9.4

保护 *IBM MQ*

注

在使用本资料及其支持的产品之前，请阅读第 607 页的『声明』中的信息。

本版本适用于 IBM® MQ V 9 发行版 4 以及所有后续发行版和修订版，直到在新版本中另有声明为止。

内容

保护 IBM MQ

对于 IBM MQ 应用程序的开发者和 IBM MQ 系统管理员而言，安全性都是一项重要注意事项。 作为绝对最小值，您应确保安全区域内和操作员工作站上的所有硬件和软件都在其支持生命周期内，具有最新的必需软件更新，并及时应用安全更新。

相关参考

IBM 安全漏洞管理

z/OS IBM Z 和 LinuxOne Security Portal

安全性概述

此主题集合介绍了 IBM MQ 安全概念。

首先介绍适用于任何计算机系统的安全概念和机制，然后讨论在 IBM MQ 中实现的这些安全机制。

常见的安全方面如下:

- 第 7 页的『标识和认证』
- 第 8 页的『授权』
- 第 9 页的『审计』
- 第 9 页的『机密性』
- 第 9 页的『数据完整性』

安全机制 是用于实现安全服务的技术工具和技术。 一种机制可单独或与其他机制一起运行，以提供特定服务。 常见安全机制的示例如下:

- 第 10 页的『密码术』
- 第 11 页的『消息摘要和数字签名』
- 第 12 页的『数字证书』
- 第 15 页的『公共密钥基础结构 (PKI)』

在规划 IBM MQ 实现时，请考虑您需要哪些安全性机制来实现对您很重要的安全性方面。 有关阅读这些主题后要考虑的内容的信息，请参阅 第 73 页的『规划安全需求』。

标识和认证

标识 是唯一标识系统或在系统中运行的应用程序的用户的能力。 认证 是证明用户或应用程序真正是该人员或该应用程序所声称的人员的能力。

例如，考虑通过输入用户标识和密码登录到系统的用户。 系统使用用户标识来标识用户。 系统在登录时通过检查提供的密码是否正确来认证用户。

IBM MQ 中的标识和认证

当应用程序连接到 IBM MQ 时，用户身份始终与该连接相关联。 用户身份最初是与应用程序进程关联的操作系统用户标识。 此身份通常足以用于与队列管理器在同一系统上托管的本地绑定应用程序。 但是，队列管理器还可以通过多种方式认证和修改与连接关联的身份。 当不一定可信的客户机应用程序通过网络连接到队列管理器时，认证与连接关联的身份很重要。

可以使用以下任何机制来建立与 IBM MQ 队列管理器的应用程序连接相关联的身份:

- 当应用程序连接到队列管理器时，它可以提供用户标识和密码。 队列管理器根据其配置验证凭证。 例如，可以将用户标识和密码传递到队列管理器的操作系统或 LDAP 服务器以进行认证。
- V 9.4.0 从 IBM MQ 9.3.4 开始，应用程序还可以提供从外部认证服务器获取的认证令牌。 有关认证令牌的更多信息，请参阅 第 287 页的『使用认证令牌』。

- 如果使用有效数字证书配置了客户机通道，那么可以将其配置为使用 TLS 相互认证。 TLS 认证可与通道认证 (CHLAUTH) 规则组合，以将相应的用户标识与连接相关联。 有关更多信息，请参阅 第 17 页的『TLS 如何提供标识，认证，机密性和完整性』。
- 通道认证 (CHLAUTH) 规则可以根据有关连接的信息来覆盖身份。 例如，通道认证规则可以根据客户机的 IP 地址设置与连接关联的用户标识。
- 定制退出代码可以根据您选择的任何条件来设置身份。

身份和认证也适用于两个队列管理器之间的通道。 这些通道称为消息通道。 当消息通道启动时，通道两端的消息通道代理程序 (MCA) 可以认证其合作伙伴。 此方法称为 相互认证。 对于发送 MCA，它提供保证它将要向其发送消息的合作伙伴是真实的。 同样，接收 MCA 也放心，即将收到来自真正合作伙伴的消息。

当已建立身份并在需要时进行认证时， IBM MQ 将以多种方式使用该身份:

- 重要的是，缺省情况下，将使用此身份进行任何后续 第 8 页的『授权』 检查。 例如，如果应用程序尝试将消息放入队列中，那么队列管理器将确认与应用程序关联的身份对队列对象具有 "put" 权限。
- 此外，每条消息都可以包含 消息上下文 信息。 此信息保存在消息描述符 (MQMD) 中。 当应用程序将消息放入队列时，队列管理器可以自动生成消息上下文。 或者，如果与应用程序关联的用户标识有权提供消息上下文，那么应用程序可以提供消息上下文。 消息中的此上下文信息提供了用于接收有关消息发起方的消息信息的应用程序。 例如，它包含放置消息的应用程序的名称以及与应用程序关联的用户标识。

不可抵赖性

不可否定服务的总体目标是能够证明特定消息与特定个人相关联。

可以将 不可抵赖性 服务视为标识和认证服务的扩展。 通常，当数据以电子方式传输时，不可抵赖性适用; 例如，向股票经纪人发出购买或出售股票的命令，或向银行发出将资金从一个账户转移到另一个账户的命令。

不可抵赖性服务可以包含多个组件，其中每个组件提供不同的功能。 如果消息的发送方拒绝发送该消息，那么具有 源证明 的不可抵赖性服务可以向接收方提供不可否认的证据，证明该消息是由该特定个人发送的。 如果消息的接收方拒绝接收该消息，那么具有 交付证明 的不可抵赖性服务可以为发件人提供不可否认的证据，证明该消息是由该特定个人接收的。

在实践中，具有几乎 100% 确定性的证明或不可否认的证据是一个困难的目标。 在现实世界中，没有什么是完全安全的。 管理安全性更关注将风险管理到企业可接受的级别。 在这种环境下，对不可否定服务的一个更现实的期待是，能够在法庭上提供可受理的证据，并支持你的案件。

不可抵赖性是 IBM MQ 环境中的相关安全服务，因为 IBM MQ 是以电子方式传输数据的方法。 例如，您可能需要与特定个人关联的应用程序发送或接收特定消息的同时期证据。

IBM MQ with Advanced Message Security 不提供不可抵赖性服务作为其基本功能的一部分。 但是，本产品文档包含有关如何通过编写您自己的出口程序在 IBM MQ 环境中提供您自己的不可抵赖性服务的建议。

授权

授权 通过将访问权仅限于授权用户及其应用程序来保护系统中的关键资源。 它可防止未经授权使用资源或以未经授权的方式使用资源。

IBM MQ 中的授权

您可以使用授权来限制特定个人或应用程序在 IBM MQ 环境中可以执行的操作。

以下是 IBM MQ 环境中授权的一些示例:

- 仅允许授权管理员发出命令来管理 IBM MQ 资源。
- 仅当与应用程序关联的用户标识有权连接到队列管理器时，才允许该应用程序连接到该队列管理器。
- 允许应用程序仅打开其功能所需的那些队列。
- 允许应用程序仅预订其功能所必需的主题。
- 允许应用程序仅对队列执行其功能所必需的那些操作。 例如，应用程序可能只需要浏览特定队列上的消息，而不需要放入或获取消息。

有关如何设置授权的更多信息，请参阅 第 76 页的『规划授权』 和关联的子主题。

审计

审计 是记录和检查事件以检测是否发生了任何意外或未经授权的活动，或者是否已尝试执行此类活动的过程。

在 IBM MQ 中进行审计

IBM MQ 可以发出事件消息以记录已发生异常活动。

以下是 IBM MQ 环境中审计的一些示例:

- 应用程序尝试打开它无权打开的队列。 发出检测事件消息。 通过检查事件消息，您发现发生了此尝试，并且可以确定需要执行的操作。
- 应用程序尝试打开通道，但尝试失败，因为不允许 TLS 连接。 发出检测事件消息。 通过检查事件消息，您发现发生了此尝试，并且可以确定需要执行的操作。

机密性

机密性 服务可防止敏感信息未经授权的泄露。

当敏感数据存储在本地时，访问控制机制可能足以保护它，前提是如果无法访问数据，那么无法读取数据。如果需要更高级别的安全性，那么可以对数据进行加密。

当敏感数据在通信网络上传输时，尤其是在不安全的网络 (例如因特网) 上传输时，加密敏感数据。 在网络环境中，访问控制机制对于拦截数据 (例如窃听) 的尝试无效。

IBM MQ 中的机密性

您可以通过加密消息在 IBM MQ 中实现机密性。

可以在 IBM MQ 环境中确保机密性，如下所示:

- 发送 MCA 从传输队列中获取消息后， IBM MQ 会在通过网络将消息发送到接收 MCA 之前使用 TLS 对消息进行加密。 在通道的另一端，将在接收 MCA 将消息放入其目标队列之前对消息进行解密。
- 当消息存储在本地队列上时， IBM MQ 提供的访问控制机制可能被视为足以保护其内容免受未经授权的泄露。 但是，为了获得更高级别的安全性，您可以使用 Advanced Message Security 对存储在队列中的消息进行加密。
- z/OS 可以使用 z/OS 数据集加密对存储在本地队列上的消息进行静态加密。

  请参阅 data set encryption for data at rest on IBM MQ for z/OS 一节。 for more information.

数据完整性

数据完整性 服务会检测是否存在未经授权的数据修改。

可以通过两种方式来更改数据: 意外地，通过硬件和传输错误，或者由于故意攻击。 许多硬件产品和传输协议都有检测和纠正硬件和传输错误的机制。 数据完整性服务的目的是检测蓄意攻击。

数据完整性服务仅旨在检测数据是否已修改。 如果已修改数据，那么它不打算将其复原到其原始状态。

访问控制机制可有助于数据完整性，因为如果访问被拒绝，那么无法修改数据。 但是，与机密性一样，访问控制机制在网络环境中并不有效。

IBM MQ 中的数据完整性

可以在 IBM MQ 环境中确保数据完整性，如下所示:

- 您可以使用 TLS 来检测在通过网络传输消息时是否有意修改了该消息的内容。 在 TLS 中，消息摘要算法提供对传输中的已修改消息的检测。

  所有 IBM MQ CipherSpecs 都提供消息摘要算法，但 TLS_RSA_WITH_NULL_NULL 除外，它不提供消息数据完整性。

IBM MQ 在接收到已修改的消息时检测到这些消息; 在接收到已修改的消息时， IBM MQ 会将 AMQ9661 错误消息写入错误日志，并且通道将停止。

- 当消息存储在本地队列上时， IBM MQ 提供的访问控制机制可能被视为足以防止故意修改消息内容。

  但是，为了获得更高级别的安全性，您可以使用 Advanced Message Security 来检测在消息放入队列的时间与从队列中检索消息的时间之间是否有意修改了消息内容。

  如果检测到已修改的消息，那么尝试接收该消息的应用程序将接收到 MQRC_SECURITY_ERROR (2063) 返回码。 如果应用程序正在使用 MQGET 调用，那么还会将消息移至 SYSTEM.PROTECTION.ERROR.QUEUE 队列。

加密概念

此主题集合描述适用于 IBM MQ 的密码术概念。

术语 实体 用于引用队列管理器， IBM MQ MQI client，单个用户或任何其他能够交换消息的系统。

密码术

密码术是在称为 *plaintext* 的可读文本与称为 *ciphertext* 的不可读格式之间进行转换的过程。

发生此情况的原因如下:

1. 发送方将明文消息转换为密文。 此过程的这一部分称为 加密 (有时称为 加密)。
2. 将密文传输到接收器。
3. 接收方将密文消息转换回其明文形式。 该过程的此部分称为 解密 (有时称为 解密)。

转换涉及在传输期间更改消息外观但不影响内容的一系列数学操作。 加密技术可确保消息的机密性并防止未经授权的查看 (窃听) ，因为加密的消息不可理解。 提供消息完整性保证的数字签名使用加密技术。 请参阅 第 19 页的『SSL/TLS 中的数字签名』以获取更多信息。

加密技术涉及一种通用算法，通过使用密钥来实现。 有两类算法:

- 要求双方使用同一密钥的密钥。 使用共享密钥的算法称为 对称 算法。 第 10 页的图 1 说明了对称密钥密码术。
- 使用一个密钥进行加密的密钥和使用另一个密钥进行解密的密钥。 其中一个必须保密，但另一个可以公开。 使用公用和专用密钥对的算法称为 非对称 算法。 第 11 页的图 2 说明非对称密钥密码术，也称为 公用密钥密码术。

所使用的加密和解密算法可以是公用的，但共享密钥和专用密钥必须保密。

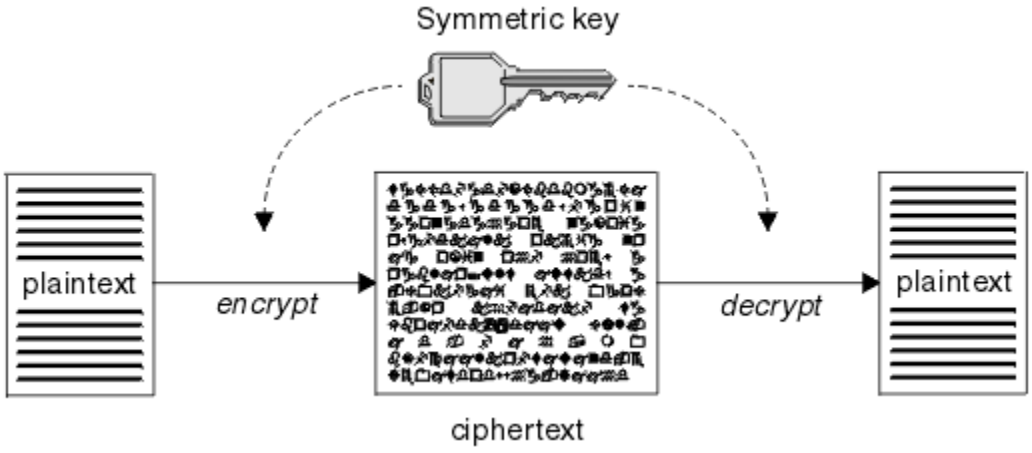


图 *1:* 对称密钥密码术 *(symmetric key cryptography)*

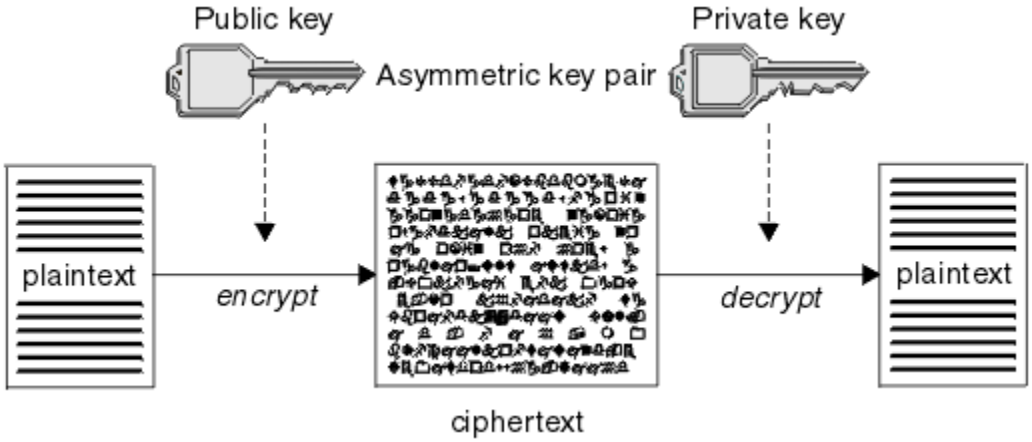


图 *2:* 非对称密钥密码术 *(asymmetric key cryptography)*

第 11 页的图 2 显示使用接收方的公用密钥加密并使用接收方的专用密钥解密的明文。 仅期望的接收方保存用于解密密文的专用密钥。 请注意，发件人还可以使用专用密钥对消息进行加密，这允许任何持有发件人公用密钥的人对消息进行解密，同时保证消息必须来自发件人。

通过非对称算法，消息使用公用密钥或专用密钥进行加密，但只能使用其他密钥进行解密。 只有私钥是秘密，公钥才能被任何人知道。 通过对称算法，共享密钥必须仅为双方所知。 这称为 密钥分发问题。 非对称算法速度较慢，但具有不存在密钥分发问题的优势。

与密码学相关的其他术语有:

强度
加密的强度由密钥大小决定。 非对称算法需要大型密钥，例如:

1024 位	低强度非对称密钥
2048 位	中等强度非对称密钥
4096 位	高强度非对称密钥

对称密钥较小: 256 位密钥为您提供强加密。

块密码算法
这些算法按块加密数据。 例如，来自 RSA Data Security Inc. 的 RC2 算法使用长度为 8 字节的块。 块算法通常比流算法慢。

流密码算法
这些算法对每个字节的数据进行操作。 流算法通常比块算法更快。

消息摘要和数字签名

消息摘要是消息内容的固定大小数字表示。 消息摘要通过散列函数计算，并可加密，形成数字签名。

用于计算消息摘要的散列函数必须满足两个条件:

- 这一定是一个办法。 除了通过测试所有可能的消息之外，不能逆转该函数来查找与特定消息摘要对应的消息。
- 要找到两个散列到同一摘要的消息，必须在计算上不可行。

消息摘要随消息本身一起发送。 接收方可以为消息生成摘要，并将其与发送方摘要进行比较。 当两个消息摘要相同时，将验证消息的完整性。 在传输期间对消息的任何篡改几乎肯定会导致不同的消息摘要。

使用密钥对称密钥创建的消息摘要称为消息认证代码 (MAC) ，因为它可以提供消息未被修改的保证。

发送者还可以生成消息摘要，然后使用非对称密钥对的专用密钥对摘要进行加密，形成数字签名。 然后，接收方必须先对签名进行解密，然后再将其与本地生成的摘要进行比较。

相关概念
第 19 页的『SSL/TLS 中的数字签名』

通过对消息的表示进行加密来形成数字签名。 加密使用签署者的专用密钥，为了提高效率，通常对消息摘要而不是消息本身进行操作。

数字证书

数字证书通过证明公用密钥属于指定实体来防止假冒。 数字证书由认证中心颁发。

数字证书提供了防止模拟的保护，因为数字证书将公用密钥绑定到其所有者，无论该所有者是个人，队列管理器还是其他某个实体。 数字证书也称为公用密钥证书，因为在您使用非对称密钥场景时，数字证书可在公用密钥所有权方面提供保证。 数字证书包含实体的公用密钥，并且是表明公用密钥属于该实体的声明：

- 当证书针对个人实体时，该证书称为个人证书或用户证书。
- 当证书针对认证中心时，该证书称为 *CA* 证书或签署者证书。

如果所有者直接将公用密钥发送给另一个实体，那么会存在消息可能被拦截且公用密钥被另一个密钥替代的风险。 这种风险称为中间人攻击。 该问题的解决方案是，通过可信的第三方交换公用密钥，为您提供强有力的保证，即确保公用密钥真正属于当前正与您通信的实体。 您将不直接发送公用密钥，而是请求可信第三方将公用密钥合并到数字证书中。 将颁发数字证书的可信第三方称为认证中心 (CA)，如第 13 页的『认证机构』中所述。

数字证书中包含的内容

数字证书包含由 X.509 标准确定的特定信息部分。

IBM MQ 所使用的数字证书遵循 X.509 标准，该标准指定所需信息以及用于发送这些信息的格式。 X.509 是 X.500 系列标准的认证框架部分。

数字证书至少包含有关所认证实体的以下信息：

- 所有者的公用密钥
- 所有者的专有名称
- 颁发证书的 CA 的专有名称
- 证书生效的起始日期
- 证书的到期日期
- X.509 中定义的证书数据格式的版本号。 X.509 标准的最新版本为 V3，大多数证书都遵循此版本。
- 序列号。 这是颁发证书的 CA 所分配的唯一标识。 序列号在颁发证书的 CA 中是唯一的：同一 CA 证书签署的两个证书不会具有相同的序列号。

X.509 V2 证书还包含颁发者标识和主题标识，X.509 V3 证书可以包含一些扩展。 一些证书扩展（如“基本约束”扩展）为标准扩展，而其他证书扩展与具体实现有关。 扩展可以是关键扩展，在此情况下，系统必须能够识别该字段；如果无法识别该字段，那么必须拒绝证书。 如果扩展不重要，那么系统可以在无法识别扩展时将其忽略。

使用签署该证书的 CA 的专用密钥生成个人证书中的数字签名。 任何需要验证个人证书的人员都可以使用 CA 的公用密钥执行此操作。 CA 的证书包含其公用密钥。

数字证书不包含专用密钥。 您必须做好专用密钥的保密工作。

个人证书的要求

IBM MQ 支持符合 X.509 标准的数字证书。 它需要客户机认证选项。

由于 IBM MQ 是对等系统，因此在 SSL/TLS 术语中将其视为客户机认证。 因此，用于 SSL/TLS 认证的任何个人证书都需要允许使用客户机认证的密钥。 并非所有服务器证书都启用了此选项，因此证书提供程序可能需要在根 CA 上为安全证书启用客户机认证。

除了规定数字证书的数据格式的标准外，还有确定证书是否有效的标准。 这些标准经过一段时间的更新，以防止某些类型的安全漏洞。 例如，较旧的 X.509 版本 1 和 2 证书未指示该证书是否可合法用于签署其他证书。 因此，恶意用户可以从合法来源获取个人证书，并创建旨在冒充其他用户的新证书。

使用 X.509 V 3 证书时， BasicConstraints 和 KeyUsage 证书扩展用于指定哪些证书可以合法地签署其他证书。 IETF RFC 5280 标准指定了一系列证书验证规则，为了防止冒充攻击，合规的应用软件必须实现这些规则。 一组证书规则称为证书验证策略。

有关 IBM MQ 中的证书验证策略的更多信息，请参阅 第 38 页的『IBM MQ 中的证书验证策略』。

认证机构

认证中心 (CA) 是颁发数字证书的可信第三方，向您保证实体的公用密钥真正属于该实体。

CA 的角色包括：

- 在收到数字证书请求后，要先验证请求者的身份，然后再构建、签署并返回个人证书
- 在 CA 证书中提供 CA 自己的公用密钥
- 发布证书撤销列表 (CRL) 中不再可信的证书列表 有关更多信息，请参阅第 299 页的『使用已撤销证书』。
- 通过操作 OCSP 响应者服务器，访问证书撤销状态

专有名称

专有名称 (DN) 将唯一地标识 X.509 证书中的实体。

注意: 在 SSLPEER 过滤器中只能使用下表中的属性。 证书 DN 可以包含其他属性，但不允许对这些属性进行过滤。

表 *1:* 可在 *SSLPEER* 过滤器中使用的 *DN* 中找到的属性类型

属性类型	描述
SERIALNUMBER	证书序列号
MAIL	电子邮件地址
Deprecated E	电子邮件地址（不推荐，最好使用 MAIL）
UID 或 USERID	用户标识
CN	公共名称
T	标题
OU	组织单元名称
DC	域组件
O	组织名称
STREET	街道/地址第一行
L	地区名称
ST（或 SP 或 S）	省/直辖市/自治区名称
PC	邮政编码
C	国家或地区
UNSTRUCTUREDNAME	主机名
UNSTRUCTUREDADDRESS	IP 地址
DNQ	专有名称限定符

X.509 标准定义通常不构成 DN 一部分但可为数字证书提供可选扩展的其他属性。

X.509 标准规定要以字符串格式指定的 DN。 例如：

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

公共名称 (CN) 可以描述单个用户或任何其他实体，例如 Web 服务器。

DN 可以包含多个 OU 和 DC 属性。 所有其他属性都只允许有一个实例。 OU 条目的顺序很重要：顺序指定组织单元名称的层次结构，最高级别单元位于最前面。 DC 条目的顺序也很重要。

IBM MQ 允许某些格式不正确的 DN。 有关更多信息，请参阅 IBM MQ SSLPEER 值的规则。

相关概念

第 12 页的『数字证书中包含的内容』
数字证书包含由 X.509 标准确定的特定信息部分。

从认证中心获取个人证书

您可以从可信的外部认证中心 (CA) 获取证书。

您可以通过向 CA 发送信息（采用证书请求格式）来获取数字证书。 X.509 标准定义此信息的格式，但某些 CA 具有自己的格式。 证书请求通常由系统使用的证书管理工具生成; 例如:

- ALW AIX, Linux, and Windows 上的 **runmqakm** 和 V 9.4.0 V 9.4.0 **runmqktool** 命令。
- z/OS z/OS 上的 RACF 。

此信息包含您的专有名称和公用密钥。 证书管理工具生成证书请求后，还会生成您必须妥善保管的专用密钥。 从不分发专用密钥。

CA 收到您的请求后，认证中心会先验证您的身份，然后再构建证书并将其作为个人证书返回给您。

第 14 页的图 3 解释了从 CA 获取数字证书的过程。

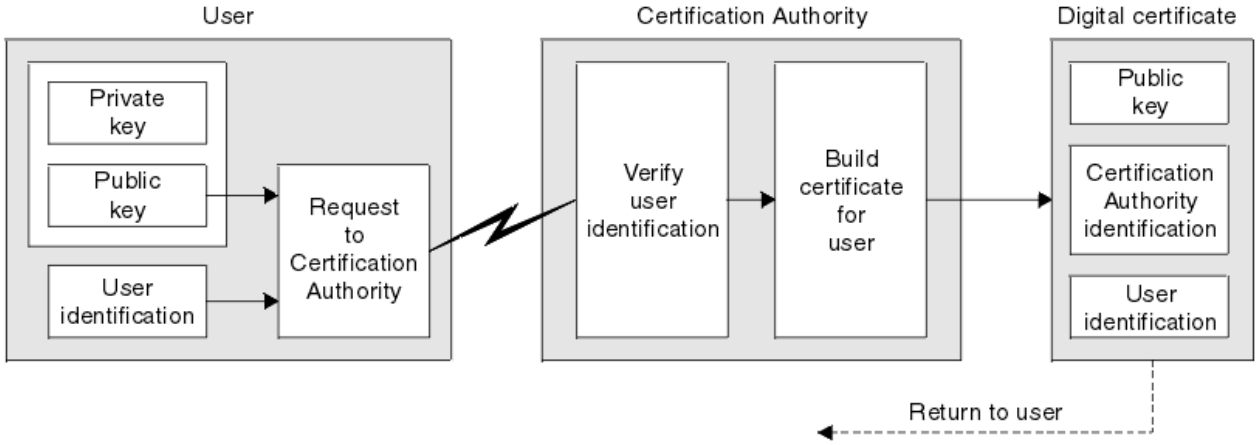


图 *3:* 获取数字证书

在该图中:

- 用户标识包含您的主题专有名称。
- 认证中心标识包含颁发证书的 CA 的专有名称。

数字证书包含图中所显示字段以外的其他字段。 有关数字证书中其他字段的更多信息，请参阅第 12 页的『数字证书中包含的内容』。

证书链的工作方式

当您接收另一个实体的证书时，可能需要使用 证书链 来获取 根 *CA* 证书。

证书链 (也称为 证书路径) 是用于认证实体的证书的列表。 链 (即路径) 以该实体的证书开头，链中的每个证书都由链中的下一个证书所标识的实体进行签名。 链使用根 CA 证书终止。 根 CA 证书始终由认证中心 (CA) 本身签署。 必须验证链中所有证书的签名，直到到达根 CA 证书为止。

第 15 页的图 4 说明了从证书所有者到根 CA 的证书路径，信任链从该路径开始。

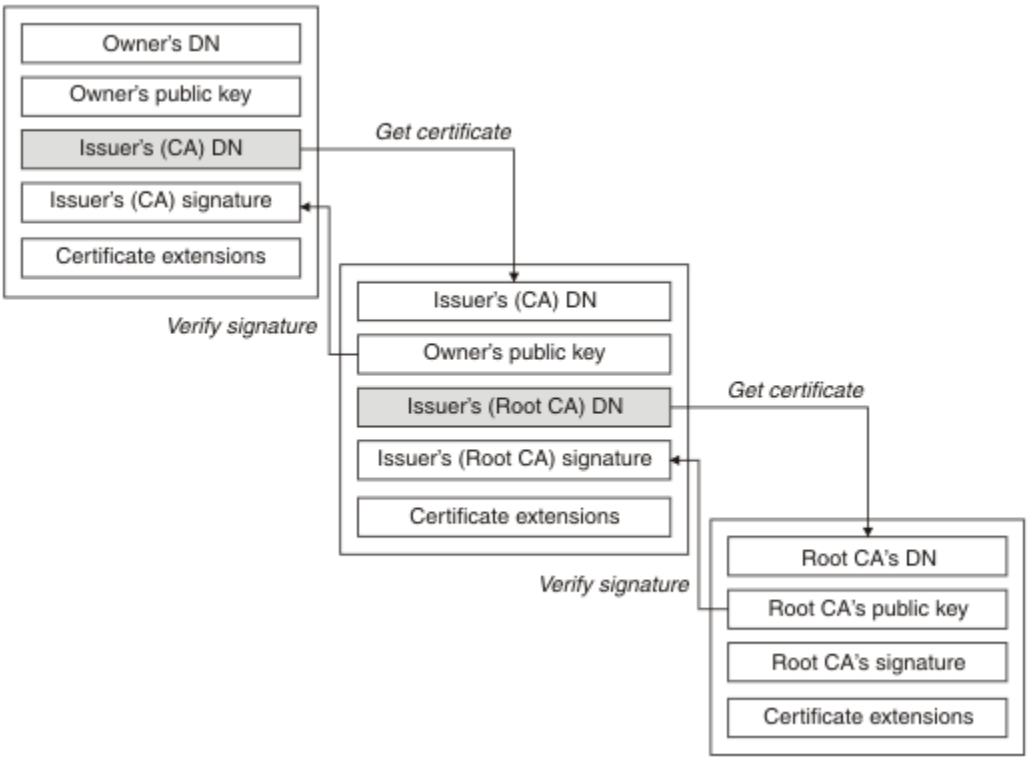


图 *4:* 信任链

每个证书都可以包含一个或多个扩展。 属于 CA 的证书通常包含设置了 isCA 标志的 BasicConstraints 扩展，以指示允许其签署其他证书。

当证书不再有效时

数字证书可以到期或撤销。

数字证书将在固定时间段内发放，并且在到期日期之后无效。

可因各种原因撤销证书，包括:

- 所有者已经移到不同的组织。
- 专用密钥不再加密。

IBM MQ 可以通过向联机证书状态协议 (OCSP) 响应程序发送请求 (仅在 AIX, Linux, and Windows 上) 来检查是否撤销了证书。 或者，他们可以访问 LDAP 服务器上的证书撤销列表 (CRL)。 OCSP 撤销和 CRL 信息由认证中心发布。 有关更多信息，请参阅 第 299 页的『使用已撤销证书』。

公共密钥基础结构 (PKI)

公用密钥基础结构 (PKI) 是一种由设施，策略和服务组成的系统，支持使用公用密钥密码术来认证交易中的参与方。

没有定义公共密钥基础结构组件的单一标准，但 PKI 通常包含认证中心 (CA) 和注册中心 (RA)。 CAs 提供以下服务:

- 发放数字证书
- 验证数字证书
- 撤销数字证书
- 分发公用密钥

X.509 标准为行业标准 "公用密钥基础结构" 提供了基础。

有关数字证书和认证中心 (CA) 的更多信息，请参阅 第 12 页的『数字证书』 。 RA 验证请求数字证书时提供的信息。 如果 RA 验证该信息，那么 CA 可以向请求者发放数字证书。

PKI 还可能提供用于管理数字证书和公用密钥的工具。 PKI 有时被描述为用于管理数字证书的 信任层次结构 ，但大多数定义包含其他服务。 一些定义包括加密和数字签名服务，但这些服务对于 PKI 的操作并不重要。

加密安全性协议：TLS

加密协议提供安全连接，使双方能够以隐私和数据完整性进行通信。 传输层安全性 (TLS) 协议由安全套接字层 (SSL) 的协议演变而来。 IBM MQ 支持 TLS。

这两种协议的主要目标是提供机密性 (有时称为 隐私) ，数据完整性，标识和使用数字证书的认证。

尽管这两个协议相似，但它们之间的差异非常显着，以至于 SSL 3.0 和 TLS 的各个版本不会进行互操作。

相关概念

第 21 页的『IBM MQ 中的 TLS 安全协议』

IBM MQ 支持传输层安全性 (TLS) 协议，以提供消息通道和 MQI 通道的链路级别安全性。

传输层安全性 (TLS) 概念

TLS 协议使双方能够相互识别和认证，并以机密性和数据完整性进行通信。 TLS 协议是从 Netscape SSL 3.0 协议演变而来的，但 TLS 和 SSL 不会互操作。

TLS 协议通过因特网提供通信安全性，并且允许客户机/服务器应用程序以是保密且可靠的方式通信。 这些协议有两层: "记录协议" 和 "握手协议" ，它们分层在诸如 TCP/IP 之类的传输协议之上。 它们都使用非对称和对称密码技术。

TLS 连接由应用程序启动，该应用程序将成为 TLS 客户机。 接收连接的应用程序将成为 TLS 服务器。 每个新会话都以握手开始，如 TLS 协议所定义。

第 368 页的『启用 CipherSpecs』 提供了 IBM MQ 支持的 CipherSpecs 的完整列表。

有关 SSL 协议的更多信息，请参阅 https://developer.mozilla.org/docs/Mozilla/Projects/NSS 中提供的信息。 有关 TLS 协议的更多信息，请参阅 TLS 工作组在因特网工程任务组 Web 站点 (https://www.ietf.org) 上提供的信息

SSL/TLS 握手概述

SSL/TLS 握手使 TLS 客户机和服务器能够建立与其通信的密钥。

本部分提供了使 TLS 客户机和服务器能够相互通信的步骤的摘要。

- 同意要使用的协议版本。
- 选择密码算法。
- 通过交换和验证数字证书来相互认证。
- 使用非对称加密技术生成共享密钥，这可避免密钥分发问题。 然后， TLS 将共享密钥用于消息的对称加密，这比非对称加密更快。

有关密码算法和数字证书的更多信息，请参阅相关信息。

在概述中， TLS 握手所涉及的步骤如下所示:

1. TLS 客户机发送 “client hello” 消息，该消息列出了加密信息，例如 TLS 版本以及客户机支持的 CipherSuites (按客户机的首选顺序)。 该消息还包含在后续计算中使用的随机字节字符串。 该协议允许 “client hello” 包含客户机支持的数据压缩方法。
2. TLS 服务器通过 “server hello” 消息进行响应，该消息包含服务器从客户机提供的列表中选择的 CipherSuite ，会话标识以及另一个随机字节字符串。 服务器还会发送其数字证书。 如果服务器需要用于客户机认证的数字证书，那么服务器将发送 “客户机证书请求” ，其中包含支持的证书类型以及可接受认证中心 (CA) 的专有名称的列表。
3. TLS 客户机验证服务器的数字证书。 有关更多信息，请参阅第 17 页的『TLS 如何提供标识，认证，机密性和完整性』 。

4. TLS 客户机发送随机字节字符串，该字符串使客户机和服务器都能够计算要用于加密后续消息数据的密钥。 随机字节字符串本身使用服务器的公用密钥进行加密。
5. 如果 TLS 服务器发送了 “客户机证书请求”，那么客户机将发送使用客户机专用密钥加密的随机字节字符串以及客户机的数字证书，或者发送 “无数字证书警报”。 此警报仅为警告，但对于某些实现，如果客户机认证是必需的，那么握手将失败。
6. TLS 服务器验证客户机的证书。 有关更多信息，请参阅第 17 页的『TLS 如何提供标识，认证，机密性和完整性』。
7. TLS 客户机向服务器发送一条 “已完成” 消息，该消息使用密钥进行加密，指示握手的客户机部分已完成。
8. TLS 服务器向客户机发送一条 “已完成” 消息，该消息使用密钥进行加密，指示握手的服务器部分已完成。
9. 在 TLS 会话期间，服务器和客户机现在可以交换使用共享密钥对称加密的消息。

第 17 页的图 5 说明 TLS 握手。

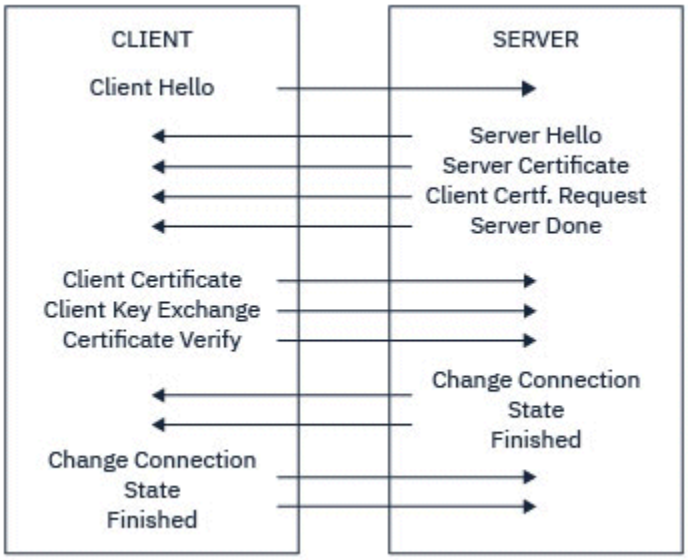


图 *5: TLS* 握手概述

TLS 如何提供标识，认证，机密性和完整性

在客户机和服务器认证期间，都有一个步骤要求使用非对称密钥对中的一个密钥对数据进行加密，并使用该对中的另一个密钥对数据进行解密。 消息摘要用于提供完整性。

有关 TLS 握手中涉及的步骤的概述，请参阅 第 16 页的『SSL/TLS 握手概述』。

TLS 如何提供认证

对于服务器认证，客户机使用服务器的公用密钥对用于计算密钥的数据进行加密。 仅当服务器可以使用正确的专用密钥对该数据进行解密时，服务器才能生成密钥。 随机字节字符串本身使用服务器的公用密钥进行加密 (概述中的步骤 第 17 页的『4』)。

对于客户机认证，服务器使用客户机证书中的公用密钥来解密客户机在握手的步骤 第 17 页的『5』 期间发送的数据。 使用密钥加密的已完成消息的交换 (概述中的步骤 第 17 页的『7』 和 第 17 页的『8』) 确认认证已完成。

如果任何认证步骤失败，那么握手将失败，并且会话将终止。

在 TLS 握手期间交换数字证书是认证过程的一部分。 有关证书如何提供防止假冒的保护的更多信息，请参阅相关信息。 所需的证书如下所示，其中 CA X 将证书发放到 TLS 客户机，而 CA Y 将证书发放到 TLS 服务器:

仅对于服务器认证， TLS 服务器需要:

- CA Y 向服务器发放的个人证书
- 服务器的专用密钥

和 TLS 客户机需要:

- CA Y 的 CA 证书

如果 TLS 服务器需要客户机认证，那么服务器将通过使用向客户机发放个人证书的 CA (在本例中为 CA X) 的公用密钥验证客户机的数字证书来验证客户机的身份。 对于服务器和客户机认证，服务器需要:

- CA Y 向服务器发放的个人证书
- 服务器的专用密钥
- CA X 的 CA 证书

客户需要:

- CA X 颁发给客户机的个人证书
- 客户机的专用密钥
- CA Y 的 CA 证书

TLS 服务器和客户机都可能需要其他 CA 证书来构成根 CA 证书的证书链。 有关证书链的更多信息，请参阅相关信息。

证书验证期间发生的情况

如概述的步骤 第 16 页的『3』 和 第 17 页的『6』 中所述， TLS 客户机验证服务器的证书， TLS 服务器验证客户机的证书。 此验证有四个方面:

1. 检查数字签名 (请参阅 第 19 页的『SSL/TLS 中的数字签名』)。
2. 已检查证书链; 您应该具有中间 CA 证书 (请参阅 第 14 页的『证书链的工作方式』)。
3. 检查到期日期和激活日期以及有效期。
4. 检查证书的撤销状态 (请参阅 第 299 页的『使用已撤销证书』)。

密钥重置

在 TLS 握手期间，将生成 密钥 以对 TLS 客户机和服务器之间的数据进行加密。 密钥在应用于数据的数学公式中使用，以将明文转换为不可读的密文，并将密文转换为明文。

密钥是从作为握手的一部分发送的随机文本生成的，用于将明文加密为密文。 密钥还用于 MAC (消息认证代码) 算法，该算法用于确定消息是否已更改。 请参阅第 11 页的『消息摘要和数字签名』以获取更多信息。

如果发现密钥，那么可以从密文中破解消息的明文，或者可以计算消息摘要，从而允许在不检测的情况下更改消息。 即使对于一个复杂的算法，也最终可以通过对密文应用每一个可能的数学变换来发现明文。 为了最大程度地减少在密钥损坏时可以解密或更改的数据量，可以定期重新协商密钥。 当已重新协商密钥时，不能再使用先前的密钥来解密使用新密钥加密的数据。

TLS 如何提供机密性

TLS 使用对称和非对称加密的组合来确保消息隐私。 在 TLS 握手期间， TLS 客户机和服务器同意仅用于一个会话的加密算法和共享密钥。 TLS 客户机和服务器之间传输的所有消息都将使用该算法和密钥进行加密，从而确保即使拦截消息，该消息也保持私有。 由于 TLS 在传输共享密钥时使用非对称加密，因此不存在密钥分发问题。 有关加密技术的更多信息，请参阅 第 10 页的『密码术』。

TLS 如何提供完整性

TLS 通过计算消息摘要来提供数据完整性。 有关更多信息，请参阅第 418 页的『消息的数据完整性』。

使用 TLS 可确保数据完整性，前提是通道定义中的 CipherSpec 使用散列算法，如 第 368 页的『启用 CipherSpecs』中的表中所述。

尤其是，如果关注数据完整性，那么应避免选择其散列算法列示为 "无" 的 CipherSpec 。 强烈建议不要使用 MD5 ，因为现在已非常旧，对于大多数实际用途而言不再安全。

CipherSpecs 和 CipherSuites

加密安全协议必须同意安全连接使用的算法。 CipherSpecs 和 CipherSuites 定义算法的特定组合。

CipherSpec 标识加密算法和消息认证代码 (MAC) 算法的组合。 TLS 连接的两端必须同意同一 CipherSpec ，才能进行通信。

IBM MQ 支持 TLS1.3 和 TLS1.2 协议以及 CipherSpecs。 但是，如果需要，可以启用不推荐的 CipherSpecs。

请参阅第 368 页的『启用 CipherSpecs』，以获取如下信息：

- IBM MQ 支持的 CipherSpecs
- 如何启用不推荐使用的 SSL 3.0 和 TLS 1.0 CipherSpecs

要点: 处理 IBM MQ 通道时，使用 CipherSpec。 处理 Java 通道， JMS 通道或 MQTT 通道时，请指定 CipherSuite。

有关 CipherSpecs 的更多信息，请参阅第 368 页的『启用 CipherSpecs』。

CipherSuite 是由 TLS 连接使用的加密算法套件。 一个套件包含三个不同的算法:

- 握手期间使用的密钥交换和认证算法
- 加密算法，用于对数据进行加密
- 用于生成消息摘要的 MAC (消息认证代码) 算法

该套件的每个组件都有多个选项，但仅当为 TLS 连接指定时，某些组合才有效。 有效 CipherSuite 的名称定义所使用算法的组合。 例如， CipherSuite TLS_RSA_WITH_AES_128_CBC_SHA 指定:

- RSA 密钥交换和认证算法
- AES 加密算法，使用 128 位密钥和密码分组链接 (CBC) 方式
- SHA-1 消息认证代码 (MAC)

SSL/TLS 中的数字签名

通过对消息的表示进行加密来形成数字签名。 加密使用签署者的专用密钥，为了提高效率，通常对消息摘要而不是消息本身进行操作。

与手写签名不同，数字签名随所签署的数据而有所不同，手写签名不取决于所签署文件的内容。 如果两个不同的消息由同一实体以数字方式进行签名，那么这两个签名不同，但这两个签名都可以使用相同的公用密钥 (即对消息进行签名的实体的公用密钥) 进行验证。

数字签名过程的步骤如下:

1. 发送方计算消息摘要，然后使用发送方的专用密钥对摘要进行加密，形成数字签名。
2. 发送者将数字签名与消息一起传输。
3. 接收方使用发送方的公用密钥对数字签名进行解密，从而重新生成发送方的消息摘要。
4. 接收方根据接收到的消息数据计算消息摘要，并验证两个摘要是否相同。

第 20 页的图 6 演示了此过程。

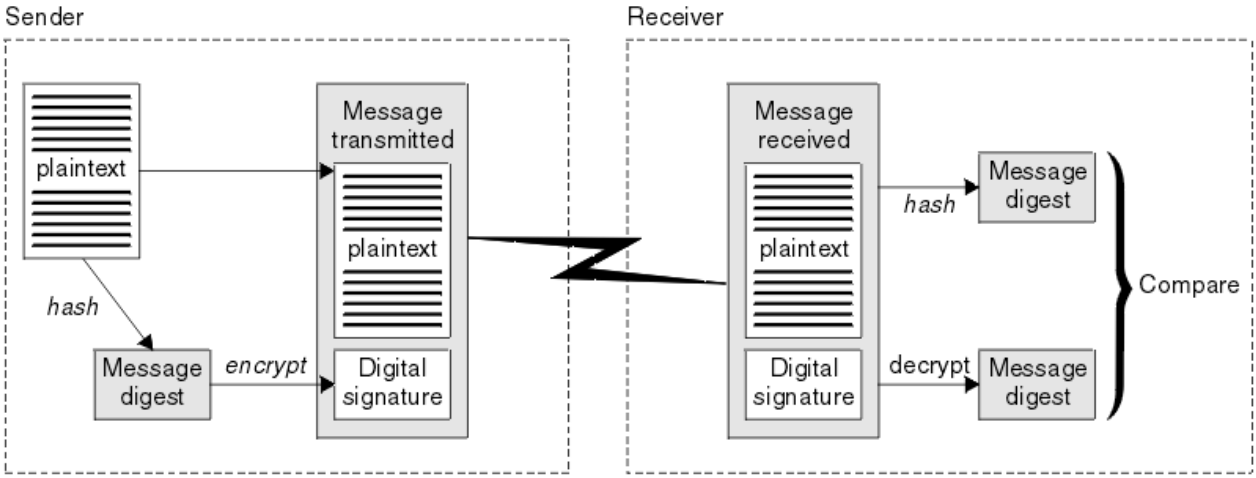


图 *6:* 数字签名过程

如果验证了数字签名，那么接收方知道:

- 在传输期间未修改消息。
- 该消息是由声称已发送该消息的实体发送的。

数字签名是完整性和认证服务的一部分。 数字签字还提供了原产地证明。 只有发件人知道专用密钥，这就提供了强有力的证据证明发件人是消息的发起方。

注: 您还可以对消息本身进行加密，这将保护消息中信息的机密性。

联邦信息处理标准

美国政府对 IT 系统和安全（包括数据加密）制定了技术建议。 美国国家标准技术研究所 (NIST) 是一个关注 IT 系统和安全的重要机构。 NIST 制定建议和标准，包括联邦信息处理标准 (FIPS)。

其中一个重要的标准是 FIPS 140-2 ，这需要使用强大的密码算法。 FIPS 140-2 还指定散列算法的要求，用于保护包在传输中不被修改。

注: 在 AIX, Linux, and Windows 上， IBM MQ 通过 IBM Crypto for C (ICC) 加密模块提供 FIPS 140-2 合规性。 此模块的证书已移至历史状态。 客户应查看 IBM Crypto for C (ICC) 证书 并了解 NIST 提供的任何建议。 当前正在进行 FIPS 140-$tag1 替换模块，可以通过在 流程列表中的 NIST CMVP 模块中搜索该模块来查看其状态。

IBM MQ Operator 3.2.0 和队列管理器容器映像 9.4.0.0 和更高版本基于 UBI 9。 FIPS 140-3 合规性当前处于暂挂状态，可通过在 流程列表中的 NIST CMVP 模块中搜索 "Red Hat Enterprise Linux 9- OpenSSL FIPS 提供程序" 来查看其状态。

IBM MQ 在配置为提供 FIPS 140-$tag2 支持时提供此支持。

随时间的变化，分析人员会开发出针对现有加密和散列算法的攻击。 此时将采用新算法来抵御这些攻击。 FIPS 140-2 将定期更新，以考虑这些变化。

相关概念

第 20 页的『国家安全局 (NSA) 套件 B 密码术』
美利坚合众国政府就 IT 系统和安全 (包括数据加密) 提供技术咨询。 美国国家安全局 (NSA) 在其 Suite B 标准中推荐了一组可互操作的密码算法。

国家安全局 (NSA) 套件 B 密码术

美利坚合众国政府就 IT 系统和安全 (包括数据加密) 提供技术咨询。 美国国家安全局 (NSA) 在其 Suite B 标准中推荐了一组可互操作的密码算法。

套件 B 标准指定仅使用一组特定安全密码算法的操作方式。 套件 B 标准指定:

- 加密算法 (AES)

- 密钥交换算法 (椭圆曲线 Diffie-Hellman ， 也称为 ECDH)
- 数字签名算法 (椭圆曲线数字签名算法，也称为 ECDSA)
- 散列算法 (SHA-256 或 SHA-384)

此外， IETF RFC 6460 标准指定符合 Suite B 的概要文件，这些概要文件定义了符合 Suite B 标准所需的详细应用程序配置和行为。 它定义了两个概要文件:

1. 与套件 B 兼容的概要文件，用于 TLS 1.2。 为符合 Suite B 的操作配置时，仅使用列出的一组受限密码算法。
2. 用于 TLS 1.0 或 TLS 1.1 的过渡概要文件。 此概要文件支持与不符合 Suite B 的服务器进行互操作性。 为 Suite B 过渡操作配置时，可使用其他加密和散列算法。

套件 B 标准在概念上类似于 FIPS 140-2 ，因为它限制了启用的密码算法集，以提供有保证的安全级别。

在 AIX, Linux, and Windows 系统上， IBM MQ 可配置为符合符合套件 B 的 TLS 1.2 概要文件，但不支持套件 B 过渡概要文件。 有关更多信息，请参阅第 35 页的『IBM MQ 中的 NSA Suite B 密码术』。

相关参考

第 20 页的『联邦信息处理标准』
美国政府对 IT 系统和安全（包括数据加密）制定了技术建议。 美国国家标准技术研究所 (NIST) 是一个关注 IT 系统和安全的重要机构。 NIST 制定建议和标准，包括联邦信息处理标准 (FIPS)。

IBM MQ 安全性机制

此主题集合描述了 IBM MQ 中实现各种安全概念的特定机制。

IBM MQ 中的 TLS 安全协议

IBM MQ 支持传输层安全性 (TLS) 协议，以提供消息通道和 MQI 通道的链路级别安全性。

消息通道和 MQI 通道可以使用 TLS 协议来提供链路级别安全性。 调用者 MCA 是 TLS 客户机，响应者 MCA 是 TLS 服务器。

IBM MQ 支持 TLS 协议的 1.2 和 1.3 版本。 缺省情况下，未启用较低版本的 TLS 以及 SSL ，但可以在需要时启用。 您可以通过在通道定义中提供 CipherSpec 来指定 TLS 协议所使用的密码算法。

请参阅 第 368 页的『启用 CipherSpecs』，以获取 IBM MQ 和 第 381 页的『不推荐使用的 CipherSpecs』支持的 CipherSpecs 列表，以获取不推荐使用的 CipherSpec。

您可以使用 SECPROT 和 SSLCIPH 参数来显示通道上正在使用的安全协议和 CipherSpec 。

在消息通道的每一端以及 MQI 通道的服务器端， MCA 代表它所连接的队列管理器执行操作。 在 TLS 握手期间， MCA 将队列管理器的数字证书发送到其在通道另一端的合作伙伴 MCA。 MQI 通道的客户机端的 IBM MQ 代码代表 IBM MQ 客户机应用程序的用户执行操作。 在 TLS 握手期间， IBM MQ 代码会将用户的数字证书发送到 MQI 通道服务器端的 MCA。

除非在通道的服务器端指定 SSLCAUTH (REQUIRED) ，否则队列管理器和 IBM MQ 客户机用户在充当 TLS 客户机时无需具有关联的个人数字证书。

数字证书存储在 密钥存储库中。 队列管理器属性 **SSLKeyRepository** 指定用于存放队列管理器的数字证书的密钥存储库的位置。 在 IBM MQ 客户机系统上， MQSSLKEYR 环境变量指定保存用户数字证书的密钥存储库的位置。 或者， IBM MQ 客户机应用程序可以在 MQCONNX 调用上的 TLS 配置选项结构 MQSCO 的 **KeyRepository** 字段中指定其位置。 请参阅相关主题，以获取有关密钥存储库以及如何指定它们所在位置的更多信息。

支持 TLS

IBM MQ 在所有平台上提供对 TLS 1.2 和 TLS 1.3 的支持。 有关 TLS 协议的更多信息，请参阅子主题中的信息。

Java 和 JMS 客户机
这些客户机使用 JVM 来提供 TLS 支持。

AIX, Linux, and Windows
TLS 支持随 IBM MQ 一起安装。

IBM i
TLS 支持是 IBM i 操作系统的组成部分。

z/OS
TLS 支持是 z/OS 操作系统的组成部分。 z/OS 上的 TLS 支持称为 系统 *SSL*。

有关 IBM MQ TLS 支持的任何先决条件的信息，请参阅 IBM MQ 的系统需求。

相关概念
第 16 页的『加密安全性协议：TLS』
加密协议提供安全连接，使双方能够以隐私和数据完整性进行通信。 传输层安全性 (TLS) 协议由安全套接字层 (SSL) 的协议演变而来。 IBM MQ 支持 TLS。

SSL/TLS 密钥存储库
相互认证的 TLS 连接在连接的每一端都需要一个密钥存储库。 密钥存储库包含数字证书和专用密钥。

此信息使用常规术语 密钥存储库 来描述数字证书及其关联专用密钥的存储库。 密钥存储库由支持 TLS 的不同平台和环境上的不同名称引用:

- IBM i 在 IBM i 上: 证书库
- 在 Java 和 JMS 上: 密钥库 和 信任库
- ALW 在 AIX, Linux, and Windows 上: 密钥数据库文件
- z/OS 在 z/OS 上: *keyring*

有关更多信息，请参阅第 12 页的『数字证书』和第 16 页的『传输层安全性 (TLS) 概念』。

相互认证的 TLS 连接在连接的每一端都需要一个密钥存储库。 密钥存储库可以包含以下证书和请求:

- 来自各种认证中心的许多 CA 证书，允许队列管理器或客户机验证它在连接的远程端从其合作伙伴处接收到的证书。 单个证书可能位于证书链中。
- 从认证中心收到的一个或多个个人证书。 您可以将单独的个人证书与每个队列管理器或 IBM MQ MQI client 相关联。 如果需要相互认证，那么个人证书在 TLS 客户机上至关重要。 如果不需要相互认证，那么客户机上不需要个人证书。 密钥存储库可能还包含对应于每个个人证书的专用密钥。
- 正在等待可信 CA 证书签署的证书请求。

有关保护密钥存储库的更多信息，请参阅第 23 页的『保护 IBM MQ 密钥存储库』。

密钥存储库的位置取决于您正在使用的平台:

IBM i **IBM i**
密钥存储库是证书库。 缺省系统证书库位于集成文件系统 (IFS) 中的 `/QIBM/UserData/ICSS/Cert/Server/Default`。 IBM MQ 将证书库的密码存储在 密码存储文件中。 例如，队列管理器 QM1 的隐藏文件为 `/QIBM/UserData/mqm/qmgrs/QM1/ssl/Stash.sth`。

或者，可以指定改为使用 IBM i 系统证书库。 要执行此操作，请将队列管理器 **SSLKEYR** 属性的值更改为 *SYSTEM。 此值指示队列管理器必须使用系统证书库，并且已注册队列管理器以用作具有数字 Certificate Manager (DCM) 的应用程序。

证书库还包含队列管理器的专用密钥。

ALW **AIX, Linux, and Windows 系统**
密钥存储库是密钥数据库文件。 例如，在 AIX and Linux 上，队列管理器 QM1 的缺省密钥数据库文件为 `/var/mqm/qmgrs/QM1/ssl/key.kdb`。 如果 IBM MQ 安装在缺省位置，那么 Windows 上的等效路径为 `C:\ProgramData\IBM\MQ\Qmgrs\QM1\ssl\key.kdb`。

要访问密钥数据库文件 IBM MQ ，必须提供密钥数据库的密码。 可以直接执行此操作，也可以通过密码隐藏文件执行此操作。 如果使用密码隐藏文件，那么该文件必须位于同一目录中，并且具有与密钥数据库相同的文件系统，并且必须以后缀 `.sth` 结尾，例如 `/var/mqm/qmgrs/QM1/ssl/key.sth`。

注: PKCS #11 加密硬件卡可以包含以其他方式保存在密钥数据库文件中的证书和密钥。 在 PKCS #11 卡上保存证书和密钥时， IBM MQ 仍需要访问密钥数据库文件和密码存储文件。

在 AIX, Linux, and Windows 系统上，密钥数据库还包含与队列管理器或 IBM MQ MQI client 相关联的个人证书的专用密钥。

z/OS **z/OS**

证书保存在 z/OS 中的密钥环中。

其他外部安全管理器 (ESM) 也使用密钥环来存储证书。

专用密钥由 RACF 管理。

保护 *IBM MQ* 密钥存储库

IBM MQ 的密钥存储库是一个文件。 请确保只有预期用户才能访问密钥存储库文件。 这将防止入侵者或其他未经授权的用户将密钥存储库文件复制到另一个系统，然后在该系统上设置相同的用户标识以模拟预期用户。

对文件的许可权取决于用户的 umask 以及使用的工具。 在 Windows 上， IBM MQ 帐户需要许可权 `BypassTraverseChecking`，这意味着文件路径中文件夹的许可权无效。

请检查密钥存储库文件的文件许可权，并确保这些文件和包含的文件夹不具有全局可读性，最好不具有组可读性。

在您使用的任何系统上，将密钥库设置为只读都是好的做法，只有管理员被允许启用写操作以执行维护。

在实践中，您必须保护所有密钥库，无论它们的位置以及它们是否受密码保护; 保护密钥存储库。

数字证书标签，了解需求

设置 TLS 以使用数字证书时，您可能必须遵循特定标签要求，具体取决于所使用的平台以及用于连接的方法。

什么是证书标签?

证书标签是表示存储在密钥存储库中的数字证书的唯一标识，并且提供了一个方便的人类可读名称，在执行密钥管理功能时使用该名称来引用特定证书。 首次向密钥存储库添加证书时，请分配证书标签。

证书标签与证书的 **Subject Distinguished Name** 或 **Subject Common Name** 字段分开。 请注意，**Subject Distinguished Name** 和 **Subject Common Name** 是证书本身中的字段。 这些是在创建证书时定义的，无法更改。 但是，如果需要，您可以更改与数字证书相关联的标签。

证书标签语法

证书标签可以包含具有以下条件的字母，数字和标点符号:

- Multi 证书标签最多可包含 64 个字符。
- z/OS 证书标签最多可包含 32 个字符。
- 证书标签可以包含空格。
- 标签区分大小写。
- 在使用 EBCDIC 片假名的系统上，不能使用小写字符。

在以下部分中指定了证书标签值的其他需求。

如何使用证书标签?

IBM MQ 使用证书标签来查找 TLS 握手期间发送的个人证书。 当密钥存储库中存在多个个人证书时，这将消除模糊性。

可以将证书标签设置为您选择的值。 如果未设置值，那么将根据您所使用的平台，使用遵循命名约定的缺省标签。 有关详细信息，请参阅以下关于特定平台的部分。

注意:

1. 不能在 Java 或 JMS 系统上自行设置证书标签。
2. 通道自动定义 (CHAD) 出口创建的自动定义通道无法设置证书标签，因为创建通道时已发生 TLS 握手。在入站通道的 CHAD 出口中设置证书标签没有任何作用。

在此上下文中， TLS 客户机是指启动握手的连接伙伴，该连接伙伴可能是 IBM MQ 客户机或其他队列管理器。

在 TLS 握手期间， TLS 客户机始终从服务器获取并验证数字证书。 通过 IBM MQ 实现， TLS 服务器始终向客户机请求证书，如果找到证书，那么客户机始终向服务器提供证书。 如果客户机无法找到个人证书，那么客户机会向服务器发送 `no certificate` 响应。

TLS 服务器始终验证客户机证书 (如果发送了客户机证书)。 如果客户机未发送证书，那么当使用 **SSLCAUTH** 参数设置为 *REQUIRED* 或 **SSLPEER** 参数值集定义充当 TLS 服务器的通道结束时，认证将失败。

请注意，仅当 IBM MQ 版本的远程同级完全支持证书标签配置，并且该通道正在使用 TLS CipherSpec 时，入站通道 (包括接收方，请求者，集群接收方，未限定的服务器和服务器连接通道) 才会发送已配置的证书。

未限定的服务器通道是未设置 CONNAME 字段的通道。

在所有其他情况下，队列管理器 **CERTLABL** 参数确定发送的证书。 尤其是，无论特定于通道的标签设置如何，以下仅接收由队列管理器的 **CERTLABL** 参数配置的证书:

- Java 和 JMS 客户机支持服务器名称指示 (SNI) ，即，基于通道的证书。
- IBM MQ 早于 IBM MQ 8.0 的版本。
- 受管 .NET 客户机

此外，通道所使用的证书必须适用于通道 CipherSpec -请参阅 第 39 页的『IBM MQ 中的数字证书和 CipherSpec 兼容性』 以获取更多信息。

IBM MQ 8.0 和更高版本支持在同一队列管理器上使用多个证书，使用每个通道的证书标签 (使用通道定义上的 **CERTLABL** 属性指定)。 队列管理器的入站通道 (例如，服务器连接或接收方) 依赖于使用 TLS 服务器名称指示 (SNI) 来检测通道名称，以便提供来自队列管理器的正确证书。 有关在队列管理器上使用多个证书的更多信息，请参阅 第 25 页的『IBM MQ 如何提供多个证书功能』。

如果通道通过 IBM MQ Internet Pass-Thru (MQIPT) 连接到目标队列管理器，并且 MQIPT 路由同时设置了 **SSLServer** 和 **SSLClient** ，那么端点之间有两个单独的 TLS 会话。 可以将 MQIPT 配置为允许目标队列管理器使用多个证书，方法是将 SNI 设置为通道名称，或者传递到路由的入站连接上接收到的 SNI。 有关多证书支持和 MQIPT 的更多信息，请参阅 IBM MQ 使用 MQIPT 的多证书支持。

有关使用单向认证 (即， TLS 客户机未发送证书时) 连接队列管理器的更多信息，请参阅 使用单向认证连接两个队列管理器。

Multiplatforms 系统

Multi

在 多平台上， TLS 服务器将证书发送到客户机。

对于队列管理器和客户机，将依次搜索以下源以查找非空值。 第一个非空值确定证书标签。 证书标签必须存在于密钥存储库中。 如果未找到与标签匹配的正确大小写和格式的匹配证书，那么将发生错误，并且 TLS 握手将失败。

队列管理器

1. 通道证书标签属性 **CERTLABL**。
2. 队列管理器证书标签属性 **CERTLABL**。
3. 缺省值，格式为: `ibmwebspheremq` ，附加队列管理器的名称，全部为小写。 例如，对于名为 `QM1` 的队列管理器，缺省证书标签为 `ibmwebspheremqqm1`。

IBM MQ Client

1. CLNTCONN 通道定义中的证书标签属性 **CERTLABL** 。
2. MQSCO 结构 **CertificateLabel** 属性。
3. 环境变量 **MQCERTLABL**。

4. 客户机 .ini 文件 (在其 SSL 部分中) **CertificateLabel** 属性
5. 缺省值，格式为: ibmwebspheremq，带有作为附加的客户机应用程序运行的用户标识，全部为小写。 例如，对于用户标识 USER1，缺省证书标签为 ibmwebspheremquser1。

z/OS 系统

z/OS

IBM MQ 客户机在 z/OS 上不受支持。 但是， z/OS 队列管理器在启动连接时可以充当 TLS 客户机的角色，在接受连接请求时可以充当 TLS 服务器的角色。 z/OS 队列管理器的证书标签需求适用于这两个角色，并且与 多平台上的需求不同。

对于队列管理器和客户机，将依次搜索以下源以查找非空值。 第一个非空值确定证书标签。 证书标签必须存在于密钥存储库中。 如果未找到与标签匹配的正确大小写和格式的匹配证书，那么将发生错误，并且 TLS 握手将失败。

1. 通道证书标签属性 **CERTLABL**。
2. 如果共享，那么队列共享组证书标签属性 **CERTQSGL**。

   如果未共享，那么队列管理器证书标签属性 **CERTLABL**。
3. 缺省值，格式为: ibmWebSphereMQ，附加队列管理器或队列共享组的名称。 请注意，此字符串区分大小写，必须按所示进行编写。 例如，对于名为 QM1 的队列管理器，缺省证书标签为 ibmWebSphereMQQM1。
4. 如果在选项 第 25 页的『3』中找不到格式的证书，那么 IBM MQ 会尝试使用在密钥环中标记为缺省值的证书。

有关如何显示密钥存储库的信息，请参阅 第 273 页的『Locating the key repository for a queue manager on z/OS』。

IBM MQ Java 和 IBM MQ JMS 客户机

IBM MQ Java 和 IBM MQ JMS 客户机使用其 Java 安全套接字扩展 (JSSE) 提供程序的工具在 TLS 握手期间选择个人证书，因此不受证书标签要求限制。

缺省行为是 JSSE 客户机通过密钥存储库中的证书进行迭代，选择找到的第一个可接受的个人证书。 但是，此行为只是缺省行为，并且取决于 JSSE 提供程序的实现。

此外，通过配置和应用程序在运行时直接访问， JSSE 接口可高度定制。 请参阅 JSSE 提供程序提供的文档以获取特定详细信息。

为了进行故障诊断，或者为了更好地了解 IBM MQ Java 客户机应用程序与特定 JSSE 提供程序组合所执行的握手，您可以通过在 JVM 环境中设置 javax.net.debug=ssl 来启用调试。

您可以在应用程序中通过配置或通过在命令行上输入 -Djavax.net.debug=ssl 来设置变量。

Linux

IBM MQ 如何提供多个证书功能

服务器名称指示 (SNI) 是 TLS 协议的扩展，允许客户机指示它需要的服务。 在 IBM MQ 术语中，这等同于通道。

SNI 扩展由 IBM MQ 使用，以允许使用通道定义上的 CERTLABL 参数在不同通道中指定多个证书。

IBM MQ 使用的 SNI 地址基于所请求的通道名称，后跟后缀 .chl.mq.ibm.com。

IBM MQ 通道名称映射为有效的 SNI 名称，如下所示:

- 将大写字母 A 到 Z 折叠为小写
- 数字 0 到 9 保持不变
- 所有其他字符 (包括小写字母 a 到 z) 将转换为其两位十六进制 ASCII 字符代码 (小写)，后跟连字符。
  - 小写字母 a 到 z 分别映射到十六进制 61- 到 7a-
  - 百分比 (%) 映射到十六进制 25-
  - 连字符 (-) 映射到十六进制 2d-

- 点 (.) 映射到十六进制 2e-
- 正斜杠 (/) 映射到十六进制 2f-
- 下划线 (_) 映射到十六进制 5f-

在 EBCDIC 平台上，在应用此映射之前，会将通道名称转换为 ASCII。

例如，通道名称 TO.QMGR1 映射到 SNI 地址 to2e-qmgr1.chl.mq.ibm.com。

相比之下，小写通道名称 to.qmgr1 映射到 74-6f-2e-71-6d-67-72-1.chl.mq.ibm.com 的 SNI 地址。

注: 在生成的 SNI URL 必须符合 URL 格式规范的环境中，例如，当客户机通过 Red Hat® OpenShift® 路由连接到在 Red Hat OpenShift 中运行的队列管理器时，通道名称不得以小写字母结尾。

SSL 节的 **OutboundSNI** 属性允许您选择应在启动 TLS 连接时将 SNI 设置为远程系统的目标 IBM MQ 通道名称还是主机名。 有关 **OutboundSNI** 属性的更多信息，请参阅 qm.ini 文件的 SSL 节和客户机配置文件的 SSL 节。

多个证书要求将 SNI 设置为 IBM MQ 通道名称。 如果使用主机名，定制或无 SNI 来连接到配置了证书标签的 IBM MQ 通道，那么将拒绝连接应用程序，并在远程队列管理器错误日志中显示一条 AMQ9673 消息。

如果通道通过 IBM MQ Internet Pass-Thru (MQIPT) 连接到目标队列管理器，那么必须将 MQIPT 配置为将 SNI 设置为通道名称，或者将入站连接上接收到的 SNI 传递到路由，以允许目标队列管理器使用多个证书。 有关多证书支持和 MQIPT 的更多信息，请参阅 IBM MQ 使用 MQIPT 的多证书支持。

有关如何使用此属性的更多信息，请参阅连接到 Red Hat OpenShift 集群中部署的队列管理器。

刷新队列管理器的密钥存储库
当您更改密钥存储库的内容时，在发出 REFRESH SECURITY TYPE (SSL) 命令或重新启动队列管理器之前，现有队列管理器进程不会选取新内容。

有关 REFRESH SECURITY TYPE (SSL) 命令的更多信息，请参阅 REFRESH SECURITY。

如果队列管理器在更改密钥库内容后创建新的通道进程 (使用 amqrmppa 或 **runmqchl**) ，那么新进程将立即开始使用新证书，而现有进程将继续使用其高速缓存的密钥库副本。 请参阅第 270 页的『当对证书或密钥存储库所作的更改在 AIX, Linux, and Windows 上生效时』以获取更多详细信息。

请注意，在发出 REFRESH SECURITY TYPE (SSL) 命令之前，多个正在运行的通道可能正在使用不同版本的密钥存储库。

您还可以使用 PCF 命令或 IBM MQ Explorer 来刷新密钥存储库。 有关更多信息，请参阅本产品文档的 IBM MQ Explorer 部分中的 MQCMD_REFRESH_SECURITY 命令 和主题 刷新 *TLS* 安全性 。

相关概念
第 26 页的『刷新客户机的 SSL/TLS 密钥存储库内容视图和 SSl/TLS 设置』
要使用密钥存储库的刷新内容更新客户机应用程序，必须停止并重新启动客户机应用程序。

刷新客户机的 *SSL/TLS* 密钥存储库内容视图和 *SSl/TLS* 设置
要使用密钥存储库的刷新内容更新客户机应用程序，必须停止并重新启动客户机应用程序。

无法在 IBM MQ 客户机上刷新安全性; 没有等效于客户机的 REFRESH SECURITY TYPE (SSL) 命令 (请参阅 REFRESH SECURITY) 以获取更多信息。

无论何时更改安全证书，都必须停止并重新启动应用程序，以使用密钥存储库的刷新内容更新客户机应用程序。

如果重新启动通道会刷新配置，并且如果应用程序具有重新连接逻辑，那么可以通过发出 STOP CHL STATUS (INACTIVE) 命令在客户机上刷新安全性。

相关概念
第 26 页的『刷新队列管理器的密钥存储库』
当您更改密钥存储库的内容时，在发出 REFRESH SECURITY TYPE (SSL) 命令或重新启动队列管理器之前，现有队列管理器进程不会选取新内容。

MQCSP 密码保护

MQCSP 结构中指定的认证凭证可以使用 IBM MQ MQCSP 密码保护功能进行保护，也可以使用 TLS 加密进行加密。

IBM MQ client 应用程序可以在连接到队列管理器时提供用户标识和密码。 V 9.4.0 从 IBM MQ 9.4.0 开始，应用程序还可以提供认证令牌作为备用认证方法。 这些凭证将以 MQCSP 结构发送到队列管理器。

如果通道正在使用 TLS 加密，那么将根据 TLS 密码规范对 MQCSP 中的凭证进行加密。 如果通道未使用 TLS 加密，那么 IBM MQ 可以在通过网络发送这些凭证之前对其进行保护，以避免通过纯文本网络发送凭证。 用于保护这些凭证的 IBM MQ 功能称为 MQCSP 密码保护。

如果使用 MQCSP 密码保护，那么将保护 MQCSP 结构中的以下数据:

- 密码 (如果 `MQCSP.AuthenticationType` 字段设置为 `MQCSP_AUTH_USER_ID_AND_PW`)。
- V 9.4.0 认证令牌 (如果 `MQCSP.AuthenticationType` 字段设置为 `MQCSP_AUTH_ID_TOKEN`)。

要点: MQCSP 密码保护对于测试和开发目的很有用，因为使用 MQCSP 密码保护比设置 TLS 加密更简单，但并不安全。 出于生产目的，使用 TLS 加密优先于 IBM MQ 密码保护，尤其是在客户机与队列管理器之间的网络不可信时，因为 TLS 加密更安全。

如果您担心正在使用什么加密，以及它提供了多少保护，那么需要使用完全 TLS 加密。 通过 TLS，算法是公开的，您可以使用 **SSLCIPH** 通道属性为企业选择相应的算法。

有关 MQCSP 结构的更多信息，请参阅 MQCSP 结构。

如果满足以下所有条件，那么将使用 IBM MQ 密码保护来保护 MQCSP 结构中的凭证:

- 连接的两端都在使用 IBM MQ 8.0 或更高版本。
- 通道未使用 TLS 加密。 如果通道具有空白 **SSLCIPH** 属性，或者 **SSLCIPH** 属性设置为不提供加密的密码规范，那么表明通道未使用 TLS 加密。 空密码 (例如， NULL_SHA) 不提供加密。
- `MQCSP.AuthenticationType` 字段设置为 `MQCSP_AUTH_USER_ID_AND_PWD` 或 `MQCSP_AUTH_ID_TOKEN`。 有关 `MQCSP.AuthenticationType` 字段的更多信息，请参阅 **AuthenticationType**。
- 如果客户机为 IBM MQ Explorer 并且未启用用户标识兼容性方式。 此方式不是 IBM MQ Explorer 用于发送用户标识和密码的缺省方式。 此条件仅适用于 IBM MQ Explorer。

如果不满足其中任何条件，那么不会使用 MQCSP 密码保护来保护凭证。 如果 **PasswordProtection** 属性的值禁止以纯文本发送凭证，并且通道未使用 TLS 加密，那么连接将失败并返回 MQRC_PASSWORD_PROTECTION_ERROR (2594) 原因码。

PasswordProtection 配置设置

客户机和队列管理器配置文件的 **Channels** 节中的 **PasswordProtection** 属性可以阻止以纯文本形式发送凭证。

注: 此属性仅与不使用 TLS 加密的连接相关。 如果连接使用 TLS 加密，那么将使用 TLS 对凭证进行加密，而不是使用 MQCSP 密码保护。

可以将该属性设置为下列其中一个值。 缺省值为 `compatible`。

兼容

如果队列管理器或客户机运行的 IBM MQ 版本低于 IBM MQ 8.0，那么将以纯文本形式发送凭证。 即，可以通过纯文本网络发送凭证，以便与不支持 MQCSP 密码保护的 IBM MQ 版本兼容。

如果队列管理器和客户机都在 IBM MQ 8.0 或更高版本上运行 IBM MQ 版本，那么凭证受 MQCSP 密码保护。

如果队列管理器和客户机都在 IBM MQ 8.0 或更高版本上运行 IBM MQ 版本，并且 `MQCSP.AuthenticationType` 字段未设置为 `MQCSP_AUTH_USER_ID_AND_PW` 或 `MQCSP_AUTH_ID_TOKEN`，那么在发送凭证之前连接失败。

始终

不得通过不受保护的网络发送凭证。

如果队列管理器和客户机都在 IBM MQ 8.0 或更高版本上运行 IBM MQ 版本，那么凭证受 MQCSP 密码保护。

在以下情况下，在发送凭证之前连接失败:

- MQCSP.AuthenticationType 字段未设置为 MQCSP_AUTH_USER_ID_AND_PW 或 MQCSP_AUTH_ID_TOKEN。
- 队列管理器或客户机正在运行低于 IBM MQ 8.0 的 IBM MQ 版本。

可选
如果队列管理器和客户机都在 IBM MQ 8.0 或更高版本上运行 IBM MQ ，并且 MQCSP.AuthenticationType 字段设置为 MQCSP_AUTH_USER_ID_AND_PW 或 MQCSP_AUTH_ID_TOKEN，那么凭证受 MQCSP 密码保护。 否则，将以纯文本形式发送凭证。

警告
允许任何客户机发送纯文本凭证。 如果接收到纯文本凭证，那么会将警告消息 AMQ9297W 写入队列管理器错误日志。

只能在队列管理器配置文件中指定此选项。

对于 Java 和 JMS 客户机， **PasswordProtection** 属性的行为会根据客户机是使用兼容性方式还是 MQCSP 方式而更改:

- 如果 Java 和 JMS 客户机以兼容性方式运行，那么在客户机连接时，不会使用 MQCSP 结构来发送用户标识和密码。 因此， **PasswordProtection** 属性的行为与针对运行版本低于 IBM MQ 8.0 的 IBM MQ 的客户机描述的行为相同。
- 如果 Java 和 JMS 客户机以 MQCSP 方式运行，那么 **PasswordProtection** 属性的行为如所述。

有关使用 Java 和 JMS 客户机进行连接认证的更多信息，请参阅 第 70 页的『使用 Java 客户机进行连接认证』。

MQCSP 密码保护和 MQIPT

V 9.4.0

如果客户机通过 IBM MQ Internet Pass-Thru (MQIPT) 连接到队列管理器，那么可以将 MQIPT 路由配置为添加或除去 TLS 加密。 即，可以使用 SSLServer=true 和 SSLClient=false 或 SSLServer=true 和 SSLClient=false 配置 MQIPT 路由。 在这种情况下，客户机和队列管理器可能无法同意密码保护算法，因为通道的一端使用 TLS 加密，而另一端不使用 TLS 加密。 这将导致连接失败，原因码为 MQRC_PASSWORD_PROTECTION_ERROR (2594)。

从 IBM MQ 9.4.0 开始， MQIPT 可以在 MQCSP 结构中添加或除去对凭证的保护，以保持添加或除去 TLS 加密的 MQIPT 路由的客户机与队列管理器之间的兼容性。 MQIPT 中的 MQCSP 密码保护是使用 **PasswordProtection** 路由属性配置的。

PasswordProtection 属性的缺省值为 required。 此值表示 MQIPT 能够添加但无法除去 MQCSP 密码保护。 与添加 TLS 加密的 MQIPT 路由的连接可能会失败，原因码为 MQRC_PASSWORD_PROTECTION_ERROR (2594) ，值为 **PasswordProtection**。 要解决此问题，请在 MQIPT 路由配置中将 **PasswordProtection** 属性的值设置为 **兼容** 。

有关 MQIPT 中的 **PasswordProtection** 属性的更多信息，请参阅 PasswordProtection。

数字证书管理器 *(Digital Certificate Manager，DCM)*

使用 DCM 来管理 IBM i 上的数字证书和专用密钥。

"数字 Certificate Manager " (DCM) 使您能够管理数字证书，并在 IBM i 服务器上的安全应用程序中使用这些证书。 通过 "数字 Certificate Manager" ，您可以请求并处理来自认证中心 (CA) 或其他第三方的数字证书。 您还可以充当本地认证中心，为用户创建和管理数字证书。

DCM 还支持使用证书撤销列表 (CRL) 来提供更强大的证书和应用程序验证过程。 您可以使用 DCM 来定义特定认证中心 CRL 驻留在 LDAP 服务器上的位置，以便 IBM MQ 可以验证是否尚未撤销特定证书。

DCM 支持并可以自动检测各种格式的证书。 当 DCM 检测到 PKCS #12 编码证书或包含加密数据的 PKCS #7 证书时，它会自动提示用户输入用于加密证书的密码。 DCM 不会提示输入不包含加密数据的 PKCS #7 证书。

DCM 提供了基于浏览器的用户界面，可用于管理应用程序和用户的数字证书。 用户界面分为两个主框架: 导航框架和任务框架。

您可以使用导航框架来选择用于管理证书的任务或使用这些证书的应用程序。 某些单独的任务直接显示在主导航框架中，但导航框架中的大多数任务都是按类别组织的。 例如， "管理证书" 是一个任务类别，其中包含各种引导式任务，例如 "查看证书"， "更新证书" 和 "导入证书"。 如果导航框架中的项是包含多个任务的类别，那么将在其左侧显示一个箭头。 箭头指示当您选择类别链接时，将显示任务的扩展列表，使您能够选择要执行的任务。

有关 DCM 的重要信息，请参阅以下 IBM Redbooks 出版物:

- *IBM i Wired Network Security: OS/400 V5R1 DCM* 和加密增强功能， SG24-6168。 具体来说，请参阅附录以获取有关将 IBM i 系统设置为本地 CA 的基本信息。
- *AS/400 Internet Security:* 开发数字证书基础结构， SG24-5659。 具体来说，请参阅 5 章。 *Digital Certificate Manager for AS/400* ，这说明了 AS/400 DCM。

联邦信息处理标准 *(FIPS)*

本主题介绍了美国国家标准与技术研究所的联邦信息处理标准 (FIPS) 加密验证程序以及可在 TLS 通道上使用的加密功能。

注: 在 AIX, Linux, and Windows 上， IBM MQ 通过 IBM Crypto for C (ICC) 加密模块提供 FIPS 140-2 合规性。 此模块的证书已移至历史状态。 客户应查看 IBM Crypto for C (ICC) 证书 并了解 NIST 提供的任何建议。 当前正在进行 FIPS 140-$tag1 替换模块，可以通过在 流程列表中的 NIST CMVP 模块中搜索该模块来查看其状态。

IBM MQ Operator 3.2.0 和队列管理器容器映像 9.4.0.0 和更高版本基于 UBI 9。 FIPS 140-3 合规性当前处于暂挂状态，可通过在 流程列表中的 NIST CMVP 模块中搜索 "Red Hat Enterprise Linux 9- OpenSSL FIPS 提供程序" 来查看其状态。

此信息适用于以下平台:

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

ALW 有关 AIX, Linux, and Windows 上的 IBM MQ TLS 连接的 FIPS 140-2 合规性的更多信息，请参阅 第 30 页的『针对 AIX, Linux, and Windows 的美国联邦信息处理标准 (FIPS)』。

z/OS 有关 z/OS 上的 IBM MQ TLS 连接的 FIPS 140-2 合规性的更多信息，请参阅 第 32 页的『Federal Information Processing Standards (FIPS) for z/OS』。

如果存在加密硬件，那么可以将 IBM MQ 使用的加密模块配置为硬件制造商提供的加密模块。 如果已完成此操作，那么仅当这些加密模块经过 FIPS 认证时，配置才符合 FIPS。

随着时间的推移，联邦信息处理标准将进行更新，以反映针对加密算法和协议的新攻击。 例如，某些 CipherSpecs 可能不再通过 FIPS 认证。 发生此类更改时，还会更新 IBM MQ 以实现最新的标准。 因此，您可能在应用维护后发现行为更改。

相关概念

第 240 页的『指定运行时在 MQI 客户机上仅使用经过 FIPS 认证的 CipherSpecs』
使用符合 FIPS 的软件创建密钥存储库，然后指定通道必须使用经 FIPS 认证的 CipherSpecs。

相关任务

在 IBM MQ classes for Java 中启用 TLS
将传输层安全性 (TLS) 与 IBM MQ classes for JMS 配合使用

相关参考

JMS 对象的 TLS 属性
第 473 页的『AIX, Linux, and Windows 上的 runmqakm 和 runmqktool 命令』
在 AIX, Linux, and Windows 系统上， 使用 **runmqakm** (GSKCapiCmd) 或 **runmqktool** (keytool) 命令来管理密钥和证书。

第 20 页的『联邦信息处理标准』

美国政府对 IT 系统和安全（包括数据加密）制定了技术建议。 美国国家标准技术研究所 (NIST) 是一个关注 IT 系统和安全的重要机构。 NIST 制定建议和标准，包括联邦信息处理标准 (FIPS)。

ALW 针对 *AIX, Linux, and Windows* 的美国联邦信息处理标准 *(FIPS)*

在 AIX, Linux, and Windows 系统上的 SSL/TLS 通道上需要密码术时， IBM MQ 将使用名为 IBM Crypto for C (ICC)的密码术包。 在 AIX, Linux, and Windows 平台上， ICC 软件已通过美国国家标准技术学会的联邦信息处理标准 (FIPS) 加密验证程序，级别为 140-2。

注: 在 AIX, Linux, and Windows 上， IBM MQ 通过 IBM Crypto for C (ICC) 加密模块提供 FIPS 140-2 合规性。 此模块的证书已移至历史状态。 客户应查看 IBM Crypto for C (ICC) 证书 并了解 NIST 提供的任何建议。 当前正在进行 FIPS 140-$tag1 替换模块，可以通过在 流程列表中的 NIST CMVP 模块中搜索该模块来查看其状态。

IBM MQ Operator 3.2.0 和队列管理器容器映像 9.4.0.0 和更高版本基于 UBI 9。 FIPS 140-3 合规性当前处于暂挂状态，可通过在 流程列表中的 NIST CMVP 模块中搜索 "Red Hat Enterprise Linux 9- OpenSSL FIPS 提供程序" 来查看其状态。

AIX, Linux, and Windows 系统上 IBM MQ TLS 连接的 FIPS 140-$tag1 合规性如下所示:

- 对于所有 IBM MQ 消息通道 (CLNTCONN 通道类型除外) ，如果满足以下条件，那么连接符合 FIPS:
  - 在已安装的操作系统版本和硬件体系结构上，已认证已安装的 IBM Global Security Kit (GSKit) ICC 版本符合 FIPS 140-2。
  - 队列管理器的 SSLFIPS 属性已设置为 YES。
  - 所有密钥存储库都是仅使用符合 FIPS 的软件 (例如带有 `-fips` 选项的 **runmqakm**) 创建和处理的。
  - 使用隐藏文件而不是队列管理器的 **KEYRPWD** 属性提供对所有密钥存储库的访问权。
- 对于所有 IBM MQ MQI client 应用程序，如果满足以下条件，那么连接将使用 GSKit 并且符合 FIPS:
  - 在已安装的操作系统版本和硬件体系结构上，已认证已安装的 GSKit ICC 版本符合 FIPS 140-2。
  - 您已指定仅使用 FIPS 认证的密码术，如 MQI 客户机的相关主题中所述。
  - 所有密钥存储库都是仅使用符合 FIPS 的软件 (例如带有 `-fips` 选项的 **runmqakm**) 创建和处理的。
  - 使用隐藏文件而不是密钥存储库密码机制提供对所有密钥存储库的访问权。
- 对于使用客户机方式的 IBM MQ classes for Java 应用程序，如果满足以下条件，那么连接将使用 JRE 的 TLS 实现并符合 FIPS:
  - 用于运行应用程序的 Java 运行时环境在已安装的操作系统版本和硬件体系结构上符合 FIPS。
  - 您已指定仅使用 FIPS 认证的密码术，如 Java 客户机的相关主题中所述。
  - 所有密钥存储库都是仅使用符合 FIPS 的软件 (例如带有 `-fips` 选项的 **runmqakm**) 创建和处理的。
- 对于使用客户机方式的 IBM MQ classes for JMS 应用程序，如果满足以下条件，那么连接将使用 JRE 的 TLS 实现并符合 FIPS:
  - 用于运行应用程序的 Java 运行时环境在已安装的操作系统版本和硬件体系结构上符合 FIPS。
  - 您已指定仅使用 FIPS 认证的密码术，如 JMS 客户机的相关主题中所述。
  - 所有密钥存储库都是仅使用符合 FIPS 的软件 (例如带有 `-fips` 选项的 **runmqakm**) 创建和处理的。
- 对于非受管 .NET 客户机应用程序，如果满足以下条件，那么连接将使用 GSKit 并且符合 FIPS:
  - 在已安装的操作系统版本和硬件体系结构上，已认证已安装的 GSKit ICC 版本符合 FIPS 140-2。
  - 您已指定仅使用 FIPS 认证的密码术，如 .NET 客户机的相关主题中所述。
  - 所有密钥存储库都是仅使用符合 FIPS 的软件 (例如带有 `-fips` 选项的 **runmqakm**) 创建和处理的。
  - 使用隐藏文件而不是密钥存储库密码机制提供对所有密钥存储库的访问权。
- 对于非受管 XMS .NET 客户机应用程序，如果满足以下条件，那么连接将使用 GSKit 并且符合 FIPS:
  - 在已安装的操作系统版本和硬件体系结构上，已认证已安装的 GSKit ICC 版本符合 FIPS 140-2。
  - 您已指定仅使用 FIPS 认证的密码术，如 XMS .NET 文档中所述。
  - 所有密钥存储库都是仅使用符合 FIPS 的软件 (例如带有 `-fips` 选项的 **runmqakm**) 创建和处理的。

– 使用隐藏文件而不是密钥存储库密码机制提供对所有密钥存储库的访问权。

所有受支持的平台都经过 FIPS 140-$tag1 认证，但每个修订包或更新包随附的自述文件中注明的情况除外。

对于使用 GSKit 的 TLS 连接， FIPS 140-$tag2 认证的组件名为 *ICC*。 它是此组件的版本，用于确定任何给定平台上的 GSKit FIPS 合规性。 要确定当前安装的 ICC 版本，请运行 **dspmqver -p 64 -v** 命令。

以下是与 ICC 相关的 **dspmqver -p 64 -v** 输出的示例抽取:

```
Icc
============
@ (#)CompanyName: IBM Corporation
@(#)LegalTrademarks: IBM
@ (#)FileDescription: IBM Crypto for C 语言
@(#)FileVersion:   8.0.0.0
@ (#)LegalCopyright: Licensed Materials-Property of IBM
@ (#) ICC
@ (#) © Copyright IBM Corp. 2002 , 2024.
@ (#) 保留所有权利。 美国政府用户
@ (#) 受限权利-使用，复制或披露
@ (#) 受与 IBM 公司的 GSA ADP 调度合同限制。
@ (#)ProductName: icc_8.0 (GoldCoast Build) 100415
@(#)ProductVersion:  8.0.0.0
@(#)ProductInfo:   10/04/15.03:32:19.10/04/15.18:41:51
@ (#) CMVCInfo:
```

可以在以下地址找到 GSKit ICC 8 的 NIST 认证声明 (包括在 GSKit 8 中): Cryptographic Module Validation Program。

如果存在加密硬件，那么可以将 IBM MQ 使用的加密模块配置为硬件制造商提供的加密模块。 如果已完成此操作，那么仅当这些加密模块经过 FIPS 认证时，配置才符合 FIPS。

在符合 FIPS 140-2 的情况下运行时实施三重 DES 限制

当 IBM MQ 配置为符合 FIPS 140-2 操作时，将对三重 DES (3DES) CipherSpecs 实施其他限制。 这些限制支持符合美国 NIST SP800-67 建议。

1. 三重 DES 密钥的所有部分都必须唯一。
2. 根据 NIST SP800-67 中的定义，三重 DES 密钥的任何部分都不能是弱密钥，半弱密钥或可能是弱密钥。
3. 在必须进行密钥重置之前，不能通过连接传输超过 32 GB 的数据。 缺省情况下， IBM MQ 不会重置密钥会话密钥，因此必须配置此重置。 使用三重 DES CipherSpec 和 FIPS 140-2 合规性时，未能启用密钥重置会导致连接在超过最大字节数后关闭，并产生错误 AMQ9288 。 有关如何配置密钥重置的信息，请参阅 第 408 页的『重置 SSL 和 TLS 密钥』。

IBM MQ 生成已符合规则 1 和 2 的三重 DES 会话密钥。 但是，要满足第三个限制，必须在 FIPS 140-2 配置中使用三重 DES CipherSpecs 时启用密钥重置。 或者，可以避免使用三重 DES。

相关概念

第 240 页的『指定运行时在 MQI 客户机上仅使用经过 FIPS 认证的 CipherSpecs』
使用符合 FIPS 的软件创建密钥存储库，然后指定通道必须使用经 FIPS 认证的 CipherSpecs。

相关任务

在 IBM MQ classes for Java 中启用 TLS

将传输层安全性 (TLS) 与 IBM MQ classes for JMS 配合使用

相关参考

JMS 对象的 TLS 属性

第 473 页的『AIX, Linux, and Windows 上的 runmqakm 和 runmqktool 命令』
在 AIX, Linux, and Windows 系统上，使用 **runmqakm** (GSKCapiCmd) 或 **runmqktool** (keytool) 命令来管理密钥和证书。

第 20 页的『联邦信息处理标准』
美国政府对 IT 系统和安全（包括数据加密）制定了技术建议。 美国国家标准技术研究所 (NIST) 是一个关注 IT 系统和安全的重要机构。 NIST 制定建议和标准，包括联邦信息处理标准 (FIPS)。

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

“联邦信息处理标准” on page 20

美国政府对 IT 系统和安全（包括数据加密）制定了技术建议。 美国国家标准技术研究所 (NIST) 是一个关注 IT 系统和安全的重要机构。 NIST 制定建议和标准，包括联邦信息处理标准 (FIPS)。

Multi 使用 *mqcertck* 验证队列管理器的 *TLS* 配置

MQCERTCK 命令是一个工具，用于查找队列管理器的 TLS 配置中的常见错误，并提供一些解决问题的建议。

介绍

mqcertck 命令将检查:

- 队列管理器的密钥存储库的存在和许可权，在队列管理器 **SSLKEYR** 属性中引用。
- 队列管理器证书的存在和有效性，在队列管理器 **CERTLABL** 属性中引用。
- 已启用 TLS 的通道的 **CERTLABL** 属性中引用的任何证书的存在和有效性。
- 客户机应用程序的密钥存储库和证书，包括检查证书是否已获得队列管理器的授权。

注: mqcertck 命令在 z/OS 或 IBM i 上不可用。

用法

要使用 **mqcertck** 命令，请从命令行运行命令 mqcertck 及其必需参数和所需的任何可选参数。

请参阅 mqcertck，以获取该命令的描述以及该命令所采用的参数。

示例

您刚刚完成设置队列管理器 QM1，以允许从客户机连接到队列管理器的 SVRCONN 通道的 TLS 连接。

您正在使用多个证书功能部件，因此队列管理器和通道都在其 **CERTLABL** 属性中指定了证书标签。 创建通道时，您在通道的 **CERTLABL** 属性中犯了错误，因此当客户机尝试连接时，队列管理器会返回 MQRC_SSL_INITIALIZATION_ERROR 的 2393 返回码。

在激活队列管理器之前，请使用 **mqcertck** 命令来验证队列管理器的 TLS 配置。

运行命令 mqcertck QM1 并接收以下输出:

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

此输出提示您检查服务器连接通道 MQCERTCK.CHANNEL。 在此，您将看到您所犯的错误，并且可以在再次运行 mqcertck 命令之前更正该错误，以验证您是否已解决问题。

验证客户机连接

mqcertck 命令能够验证客户机密钥存储库以及队列管理器的 TLS 配置。 为此， **mqcertck** 需要能够从运行队列管理器的机器访问客户机的密钥存储库。

运行 **mqcertck** 命令时，如果提供 **-clientkeyr** 参数以及客户机密钥存储库 (不包括扩展) 的位置，那么 **mqcertck** 会针对队列管理器检查此密钥存储库。

如果您知道客户机将使用哪个通道来连接到队列管理器，那么可以使用 **-clientchannel** 标志来指定此标志。

如果客户机正在使用相互认证来连接到队列管理器，那么可以使用 **-clientusername** 或 **-clientlabel** 参数，以告知 **mqcertck** 命令要在客户机密钥存储库中使用哪个证书。

如果您正在使用缺省证书，并且未向客户机应用程序提供证书标签，那么可以使用 **-clientusername** 和运行此应用程序的 **username** 参数。

在 **mqcertck** 命令的操作期间，该命令将生成证书标签 `ibmwebspheremqXXXX` ，其中 XXXX 是在 **-clientusername** 参数中传递的值。

为了完全验证客户机密钥存储库， **mqcertck** 命令使用 IBM Global Security Kit (GSKit)创建哑元连接。 要执行此操作，该命令需要具有在其客户机测试期间可以绑定到的可用端口。 使用的缺省端口为 5857，但是，如果此端口已在使用中，那么可以指定要在客户机测试期间使用的其他端口。

注: 虽然 **mqcertck** 命令绑定到端口，但 **mqcertck** 不使用任何外部通信，并且所有测试都在本地执行。

IBM MQ MQI client 上的 *SSL/TLS*

IBM MQ 在客户机上支持 TLS。 您可以通过各种方式定制 TLS 的使用。

IBM MQ 为 AIX, Linux, and Windows 系统上的 IBM MQ MQI clients 提供 TLS 支持。 如果使用的是 IBM MQ classes for Java，请参阅 使用 IBM MQ classes for Java ，如果使用的是 IBM MQ classes for JMS，请参阅 使用 IBM MQ classes for JMS。 本部分的其余部分不适用于 Java 或 JMS 环境。

您可以使用 IBM MQ 客户机配置文件中的 MQSSLKEYR 值或者在应用程序进行 MQCONNX 调用时为 IBM MQ MQI client 指定密钥存储库。 您有三个选项可用于指定通道使用 TLS:

- 使用通道定义表
- 在 MQCONNX 调用上使用 SSL 配置选项结构 MQSCO
- 使用 Active Directory (在 Windows 系统上)

不能使用 MQSERVER 环境变量来指定通道使用 TLS。

只要在通道的另一端未指定 TLS ，您就可以在没有 TLS 的情况下继续运行现有 IBM MQ MQI client 应用程序。

如果在客户机上对 TLS 密钥存储库的内容， TLS 密钥存储库的位置，认证信息或加密硬件参数进行了更改，那么需要结束所有 TLS 连接，以便在应用程序用于连接到队列管理器的客户机连接通道中反映这些更改。 所有连接结束后，重新启动 TLS 通道。 将使用所有新的 TLS 设置。 这些设置类似于队列管理器系统上的 REFRESH SECURITY TYPE (SSL) 命令刷新的设置。

当 IBM MQ MQI client 在具有加密硬件的 AIX, Linux, and Windows 系统上运行时，请使用 MQSSLCRYP 环境变量配置该硬件。 此变量等同于 ALTER QMGR MQSC 命令上的 SSLCRYP 参数。 请参阅 ALTER QMGR ，以获取 ALTER QMGR MQSC 命令上 SSLCRYP 参数的描述。 如果使用 SSLCRYP 参数的 GSK_PCS11 版本，那么必须完全以小写形式指定 PKCS #11 令牌标签。

IBM MQ MQI clients 上支持 TLS 密钥重置和 FIPS。 有关更多信息，请参阅第 408 页的『重置 SSL 和 TLS 密钥』和第 30 页的『针对 AIX, Linux, and Windows 的美国联邦信息处理标准 (FIPS)』。

请参阅 第 239 页的『设置 IBM MQ MQI client 安全性』，以获取有关 IBM MQ MQI clients 的 TLS 支持的更多信息。

相关任务

IBM MQ MQI client 配置文件， `mqclient.ini`

指定 *MQI* 通道使用 *SSL/TLS*

要使 MQI 通道使用 TLS ，客户机连接通道的 *SSLCipherSpec* 属性的值必须是客户机平台上 IBM MQ 支持的 CipherSpec 的名称。

您可以通过以下方式定义具有此属性值的客户机连接通道。 它们按优先级递减顺序列出。

1. 当 PreConnect 出口提供要使用的通道定义结构时。

   PreConnect 出口可以在通道定义结构 MQCD 的 *SSLCipherSpec* 字段中提供 CipherSpec 的名称。 此结构在 PreConnect 出口使用的 MQNXP 出口参数结构的 **ppMQCDArrayPtr** 字段中返回。

2. 当 IBM MQ MQI client 应用程序发出 MQCONNX 调用时。

   应用程序可以在通道定义结构 MQCD 的 *SSLCipherSpec* 字段中指定 CipherSpec 的名称。此结构由连接选项结构 MQCNO 引用， MQCNO 是 MQCONNX 调用上的参数。

3. 使用客户机通道定义表 (CCDT)。

   客户机通道定义表中的一个或多个条目可以指定 CipherSpec 的名称。例如，如果使用 DEFINE CHANNEL MQSC 命令创建条目，那么可以使用该命令上的 SSLCIPH 参数来指定 CipherSpec 的名称。

4. 在 Windows 上使用 Active Directory 。

   在 Windows 系统上，可以使用 **setmqscp** 控制命令在 Active Directory 中发布客户机连接通道定义。 其中一个或多个定义可以指定 CipherSpec 的名称。

例如，如果客户机应用程序在 MQCONNX 调用上的 MQCD 结构中提供客户机连接通道定义，那么此定义优先于客户机通道定义表中可由 IBM MQ 客户机访问的任何条目。

不能使用 MQSERVER 环境变量在使用 TLS 的 MQI 通道的客户机端提供通道定义。

要检查客户机证书是否已流动，请在通道的服务器端显示通道状态以显示对等名称参数值。

相关概念

第 388 页的『为 IBM MQ MQI client 指定 CipherSpec』
您有三个选项可用于为 IBM MQ MQI client 指定 CipherSpec 。

IBM MQ 中的 *CipherSpecs* 和 *CipherSuites*

IBM MQ 支持 TLS1.3 和 TLS 1.2 CipherSpecs 以及 RSA 和 Diffie-Hellman 算法。 但是，如果需要，可以启用不推荐的 CipherSpecs。

请参阅第 368 页的『启用 CipherSpecs』，以获取如下信息：

- IBM MQ 支持 CipherSpecs 。
- 如何启用不推荐的 SSL 3.0 和 TLS 1.0 CipherSpecs。

IBM MQ 支持 RSA 和 Diffie-Hellman 密钥交换和认证算法。 TLS 握手期间使用的密钥大小可能取决于您使用的数字证书，但某些 CipherSpecs 包含握手密钥大小的规范。 握手密钥大小越大，提供的认证越强。 使用较小的密钥大小时，握手会更快。

相关概念

第 19 页的『CipherSpecs 和 CipherSuites』
加密安全协议必须同意安全连接使用的算法。 CipherSpecs 和 CipherSuites 定义算法的特定组合。

IBM MQ 中的 *NSA Suite B* 密码术

本主题提供有关如何配置 IBM MQ for AIX, Linux, and Windows 以符合符合套件 B 的 TLS 1.2 概要文件的信息。

随着时间的推移， NSA 密码套件 B 标准会更新，以反映针对加密算法和协议的新攻击。 例如，某些 CipherSpecs 可能不再通过 Suite B 认证。 发生此类更改时，还会更新 IBM MQ 以实现最新的标准。 因此，您可能在应用维护后发现行为更改。 IBM MQ 自述文件列出了每个产品维护级别强制实施的 Suite B 版本。 如果配置 IBM MQ 以强制实施 Suite B 合规性，请在计划应用维护时始终查阅自述文件。 请参阅 IBM MQ, WebSphere MQ 和 MQSeries 产品自述文件。

在 AIX, Linux, and Windows 系统上，可以将 IBM MQ 配置为符合表 1 中所示的安全级别的符合套件 B 的 TLS 1.2 概要文件。

表 *3:* 具有允许的 *CipherSpecs* 和数字签名算法的套件 *B* 安全级别

安全级别	允许的 CipherSpecs	允许的数字签名算法
128 位	ECDHE_ECDSA_AES_128_GCM_SHA256 ECDHE_ECDSA_AES_256_GCM_SHA384	带有 SHA-256 的 ECDSA 带有 SHA-384 的 ECDSA
192 位	ECDHE_ECDSA_AES_256_GCM_SHA384	带有 SHA-384 的 ECDSA

表 3: 具有允许的 *CipherSpecs* 和数字签名算法的套件 *B* 安全级别 *(继续)*		
安全级别	**允许的 CipherSpecs**	**允许的数字签名算法**
两个 [1]	ECDHE_ECDSA_AES_128_GCM_SHA256 ECDHE_ECDSA_AES_256_GCM_SHA384	带有 SHA-256 的 ECDSA 带有 SHA-384 的 ECDSA

1. 可以同时配置 128 位和 192 位安全级别。 由于 Suite B 配置确定了可接受的最低密码算法，因此配置这两个安全级别等同于仅配置 128 位安全级别。 192 位安全级别的密码算法比 128 位安全级别所需的最低密码算法更强，因此即使未启用 192 位安全级别，也允许这些算法用于 128 位安全级别。

注: 用于 安全级别 的命名约定不一定表示椭圆曲线大小或 AES 加密算法的密钥大小。

CipherSpec 与 Suite B 的构造

虽然 IBM MQ 的缺省行为是不符合 Suite B 标准，但可以将 IBM MQ 配置为符合 AIX, Linux, and Windows 系统上的一个或两个安全级别。 在成功配置 IBM MQ 以使用 Suite B 之后，使用 CipherSpec 启动出站通道的任何尝试都将导致错误 AMQ9282。 此活动还会导致 MQI 客户机返回原因码 MQRC_CIPHER_SPEC_NOT_SUITE_B。 同样，尝试使用不符合 Suite B 配置的 CipherSpec 启动入站通道会导致错误 AMQ9616。

有关 IBM MQ CipherSpecs 的更多信息，请参阅 第 368 页的『启用 CipherSpecs』

套件 B 和数字证书

套件 B 限制可用于签署数字证书的数字签名算法。 套件 B 还会限制证书可以包含的公用密钥的类型。 因此，必须将 IBM MQ 配置为使用其数字签名算法和公用密钥类型为远程合作伙伴的已配置套件 B 安全级别所允许的证书。 将拒绝不符合安全级别要求的数字证书，并且连接将失败并返回错误 AMQ9633 或 AMQ9285。

对于 128 位套件 B 安全级别，证书主体集的公用密钥需要使用 NIST P-256 椭圆曲线或 NIST P-384 椭圆曲线，并使用 NIST P-256 椭圆曲线或 NIST P-384 椭圆曲线进行签名。 在 192 位 Suite B 安全级别，证书主体集的公用密钥需要使用 NIST P-384 椭圆曲线并使用 NIST P-384 椭圆曲线进行签名。

要获取适合于符合 Suite B 的操作的证书，请使用 **runmqakm** 命令并指定 **-sig_alg** 参数以请求合适的数字签名算法。 EC_ecdsa_with_SHA256 和 EC_ecdsa_with_SHA384 **-sig_alg** 参数值对应于由允许的 Suite B 数字签名算法签名的椭圆曲线键。

有关 **runmqakm** 命令的更多信息，请参阅 第 473 页的『在 AIX, Linux, and Windows 上管理密钥和证书』。

创建和请求数字证书

要为 Suite B 测试创建自签名数字证书，请参阅 第 474 页的『在 AIX, Linux, and Windows 上创建自签名个人证书』

要请求 CA 签署的数字证书以供 Suite B 生产使用，请参阅 第 476 页的『在 AIX, Linux, and Windows 上请求个人证书』。

注: 正在使用的认证中心必须生成满足 IETF RFC 6460 中描述的要求的数字证书。

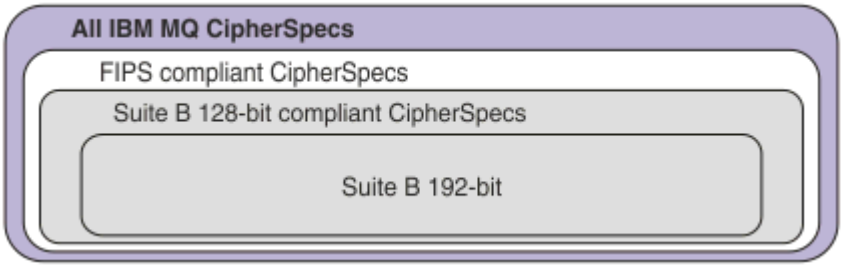
FIPS 140-$tag1 和 Suite B

注: 在 AIX, Linux, and Windows 上， IBM MQ 通过 IBM Crypto for C (ICC) 加密模块提供 FIPS 140-2 合规性。 此模块的证书已移至历史状态。 客户应查看 IBM Crypto for C (ICC) 证书 并了解 NIST 提供的任何建议。 当前正在进行 FIPS 140-$tag1 替换模块，可以通过在 流程列表中的 NIST CMVP 模块中搜索该模块来查看其状态。

IBM MQ Operator 3.2.0 和队列管理器容器映像 9.4.0.0 和更高版本基于 UBI 9。 FIPS 140-3 合规性当前处于暂挂状态，可通过在 流程列表中的 NIST CMVP 模块中搜索 "Red Hat Enterprise Linux 9- OpenSSL FIPS 提供程序" 来查看其状态。

套件 B 标准在概念上类似于 FIPS 140-2，因为它会限制启用的密码算法集，以提供有保证的安全级别。 当为符合 FIPS 140-$tag4 的操作配置了 IBM MQ 时，可以使用当前支持的 Suite B CipherSpecs 。 因此，可以同时针对 FIPS 和 Suite B 合规性配置 IBM MQ，在这种情况下，这两组限制都适用。

下图说明了这些子集之间的关系:



为符合 Suite B 的操作配置 IBM MQ

有关如何为符合 Suite B 的操作配置 IBM MQ on AIX, Linux, and Windows 的信息，请参阅 第 37 页的『为 Suite B 配置 IBM MQ』。

IBM MQ 在以下平台和客户机上不支持符合 Suite B 的操作:

- IBM i 平台
- z/OS 平台
- Java 客户机
- JMS 客户机

相关概念
第 240 页的『指定运行时在 MQI 客户机上仅使用经过 FIPS 认证的 CipherSpecs』
使用符合 FIPS 的软件创建密钥存储库，然后指定通道必须使用经 FIPS 认证的 CipherSpecs。

ALW 为 *Suite B* 配置 *IBM MQ*

可以将 IBM MQ 配置为在 AIX, Linux, and Windows 平台上按照 NSA Suite B 标准运行。

套件 B 限制启用的密码算法集，以提供有保证的安全级别。 可以将 IBM MQ 配置为与 Suite B 合规运行，以提供增强的安全性级别。 有关 Suite B 的更多信息，请参阅 第 20 页的『国家安全局 (NSA) 套件 B 密码术』。 有关 Suite B 配置及其对 TLS 通道的影响的更多信息，请参阅 第 35 页的『IBM MQ 中的 NSA Suite B 密码术』。

队列管理器

对于队列管理器，请使用带有参数 **SUITEB** 的命令 **ALTER QMGR** 来设置适合于所需安全级别的值。 有关更多信息，请参阅 ALTER QMGR。

您还可以将 PCF **MQCMD_CHANGE_Q_MGR** 命令与 **MQIA_SUITE_B_STRENGTH** 参数配合使用，为符合 Suite B 的操作配置队列管理器。

注: 如果更改队列管理器的 Suite B 设置，那么必须重新启动 MQXR 服务以使这些设置生效。

MQI 客户机

缺省情况下， MQI 客户机不会强制实施 Suite B 合规性。 您可以通过执行下列其中一个选项来启用 MQI 客户机以实现 Suite B 合规性:

1. 通过将 MQCONNX 调用上 MQSCO 结构中的 EncryptionPolicySuiteB 字段设置为以下一个或多个值:
   - `MQ_SUITE_B_NONE`
   - `MQ_SUITE_B_128_BIT`
   - `MQ_SUITE_B_192_BIT`

   将 `MQ_SUITE_B_NONE` 与任何其他值配合使用无效。

有关 MQSCO 结构的更多信息，请参阅 MQSCO-SSL 配置选项。

2. 通过将 **MQSUITEB** 环境变量设置为以下一个或多个值:
   - NONE
   - 128_BIT
   - 192_BIT

   您可以使用逗号分隔列表指定多个值。 将值 NONE 与任何其他值配合使用无效。
3. 通过将 客户机配置文件的 SSL 节 中的 **EncryptionPolicySuiteB** 属性设置为以下一个或多个值:
   - NONE
   - 128_BIT
   - 192_BIT

   您可以使用逗号分隔列表指定多个值。 将 NONE 与任何其他值配合使用无效。

注: MQI 客户机设置按优先级顺序列出。 MQCONNX 调用上的 MSCO 结构将覆盖 **MQSUITEB** 环境变量上的设置，这将覆盖 SSL 节中的属性。

.NET

对于 .NET 非受管客户机，属性 **MQC.ENCRYPTION_POLICY_SUITE_B** 指示所需的 Suite B 安全性类型。

有关在 IBM MQ classes for .NET 中使用 Suite B 的信息，请参阅 MQEnvironment .NET 类。

AMQP

队列管理器的 Suite B 属性设置适用于该队列管理器上的 AMQP 通道。 如果修改队列管理器套件 B 设置，那么必须重新启动 AMQP 服务以使更改生效。

IBM MQ 中的证书验证策略

证书验证策略确定证书链验证如何严格符合行业安全标准。

证书验证策略取决于平台和环境，如下所示:

- 对于所有平台上的 Java 和 JMS 应用程序，证书验证策略取决于 Java 运行时环境的 JSSE 组件。 有关证书验证策略的更多信息，请参阅 JRE 的文档。
- ALW 对于 AIX, Linux, and Windows 系统，证书验证策略由 IBM Global Security Kit (GSKit) 提供，并且可以进行配置。 V 9.4.0 V 9.4.0 三个 不同的证书验证策略:
  - 旧证书验证策略，用于与不符合当前 IETF 证书验证标准的旧数字证书实现最大向后兼容性和互操作性。 此策略称为基本策略。
  - 严格的符合标准的证书验证策略，强制实施 RFC 5280 标准。 此策略称为标准策略。
  - V 9.4.0 V 9.4.0 不认证 TLS 服务器证书的证书验证策略，仅可用于客户机应用程序。
- IBM i 对于 IBM i 系统，证书验证策略取决于操作系统提供的安全套接字库。 有关证书验证策略的更多信息，请参阅操作系统的文档。
- z/OS 对于 z/OS 系统，证书验证策略取决于操作系统提供的系统 SSL 组件。 有关证书验证策略的更多信息，请参阅操作系统的文档。

有关如何配置证书验证策略的信息，请参阅 第 38 页的『在 IBM MQ 中配置证书验证策略』。 有关基本证书验证策略与标准证书验证策略之间的差异的更多信息，请参阅 AIX, Linux, and Windows 上的证书验证和信任策略设计。

在 *IBM MQ* 中配置证书验证策略

您可以通过多种不同的方法来指定使用哪个 TLS 证书验证策略来验证从远程伙伴系统接收到的数字证书。

关于此任务

证书验证策略确定证书链验证如何严格符合行业安全标准。 证书验证策略取决于平台和环境。 有关证书验证策略的更多信息，请参阅 第 38 页的『IBM MQ 中的证书验证策略』。

过程

- 要在队列管理器上设置证书验证策略，请使用队列管理器属性 **CERTVPOL**。
  有关设置此属性的更多信息，请参阅 ALTER QMGR (alter queue manager settings)。
- 要在客户机上设置证书验证策略，请使用以下方法。
  如果使用多个方法来设置策略，那么客户机将按以下优先级顺序使用设置:

  1. 在客户机 MQSCO 结构中使用 `CertificateValPolicy` 字段。 将该字段设置为下列其中一个值:

     MQ_CERT_VAL_POLICY_ANY
     应用安全套接字库支持的每个证书验证策略。 如果任何策略认为证书链有效，请接受该证书链。

     MQ_CERT_VAL_POLICY_RFC5280
     仅应用符合 RFC5280 的证书验证策略。 此设置提供比 ANY 设置更严格的验证，但是会拒绝一些较旧的数字证书。

     V 9.4.0 V 9.4.0 **MQ_CERT_VAL_POLICY_NONE**
     不应用证书验证策略。 此设置仅适用于客户机应用程序，并且接受 TLS 服务器证书而不验证信任链。

     有关使用此字段的更多信息，请参阅 MQSCO-SSL 配置选项。
  2. 使用客户机环境变量 **MQCERTVPOL**。 要设置此环境变量，请使用下列其中一个命令:
     - Linux AIX 对于 AIX and Linux 系统:

       ```
       export MQCERTVPOL= value
       ```

     - Windows 对于 Windows 系统:

       ```
       SET MQCERTVPOL= value
       ```

     - IBM i 对于 IBM i 系统:

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

  3. 在客户机配置文件中使用 SSL 节的 **CertificateValPolicy** 属性。 将此属性设置为下列其中一个值:

     ANY
     使用底层安全套接字库支持的任何证书验证策略。 此设置为缺省设置。

     RFC5280
     仅使用符合 RFC 5280 标准的证书验证。

     V 9.4.0 V 9.4.0 **NONE**
     不应用证书验证策略。 此设置接受 TLS 服务器证书而不验证信任链。

     有关使用此属性的更多信息，请参阅 客户机配置文件的 SSL 节。

IBM MQ 中的数字证书和 *CipherSpec* 兼容性

本主题通过概述 CipherSpecs 与 IBM MQ 中的数字证书之间的关系，提供有关如何为安全策略选择相应 CipherSpecs 和数字证书的信息。

只有一部分受支持的 CipherSpecs 可用于所有受支持的数字证书类型。 因此，必须为数字证书选择相应的 CipherSpec 。 同样，如果贵组织的安全策略要求您使用特定的 CipherSpec ，那么必须为该 CipherSpec 获取相应的数字证书。

MD5 数字签名算法和 TLS 1.2

使用 TLS 1.2 协议时，将拒绝使用 MD5 算法签署的数字证书。 这是因为许多加密分析人员现在认为 MD5 算法很弱，通常不建议使用该算法。 要使用基于 TLS 1.2 协议的较新的 CipherSpecs ，请确保数字证书在其数字签名中不使用 MD5 算法。 使用 TLS 1.0 协议的较旧的 CipherSpecs 不受此限制，并且可以继续将证书与 MD5 数字签名配合使用。

要查看特定证书的数字签名算法，可以使用 **runmqakm** 命令:

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

其中 *cert_label* 是要显示的数字签名算法的证书标签。 请参阅 数字证书标签 以获取详细信息。

运行 **runmqakm** 命令会生成显示使用指定的签名算法的输出:

```
Label : ibmmqexample
Key Size : 1024
Version : X509 V3
Serial : 4e4e93f1
Issuer : CN=Old Certificate Authority,OU=Test,O=Example,C=US
Subject : CN=Example Queue Manager,OU=Test,O=Example,C=US
Not Before : August 19, 2011 5:48:49 PM GMT+01:00
Not After : August 18, 2012 5:48:49 PM GMT+01:00
Public Key
    30 81 9F 30 0D 06 09 2A 86 48 86 F7 0D 01 01 01
    05 00 03 81 8D 00 30 81 89 02 81 81 00 98 5A 7A
    F0 18 21 EE E4 8A 6E DE C8 01 4B 3A 1E 41 90 3D
    CE 01 3F E6 32 30 6C 23 59 F0 FE 78 6D C2 80 EF
    BC 83 54 7A EB 60 80 62 6B F1 52 FE 51 9D C1 61
    80 A5 1C D4 F0 76 C7 15 6D 1F 0D 4D 31 3E DC C6
    A9 20 84 6E 14 A1 46 7D 4C F5 79 4D 37 54 0A 3B
    A9 74 ED E7 8B 0F 80 31 63 1A 0B 20 A5 99 EE 0A
    30 A6 B6 8F 03 97 F6 99 DB 6A 58 89 7F 27 34 DE
    55 08 29 D8 A9 6B 46 E6 02 17 C3 13 D3 02 03 01
    00 01
Public Key Type : RSA (1.2.840.113549.1.1.1)
Fingerprint : SHA1 :
    09 4E 4F F2 1B CB C1 F4 4F 15 C9 2A F7 32 0A 82
    DA 45 92 9F
Fingerprint : MD5 :
    44 54 81 7C 58 68 08 3A 5D 75 96 40 D5 8C 7A CB
Fingerprint : SHA256 :
    3B 47 C6 E7 7B B0 FF 85 34 E7 48 BE 11 F2 D4 35
    B7 9A 79 53 2B 07 F5 E7 65 E8 F7 84 E0 2E 82 55
Signature Algorithm : MD5WithRSASignature (1.2.840.113549.1.1.4)
Value
    3B B9 56 E6 F2 77 94 69 5B 3F 17 EA 7B 19 D0 A2
    D7 10 38 F1 88 A4 44 1B 92 35 6F 3B ED 99 9B 3A
    A5 A4 FC 72 25 5A A9 E3 B1 96 88 FC 1E 9F 9B F1
    C5 E8 8E CF C4 8F 48 7B 0E A6 BB 13 AE 2B BD D8
    63 2C 03 38 EF DC 01 E1 1F 7A 6F FB 2F 65 74 D0
    FD 99 94 BA B2 3A D5 B4 89 6C C1 2B 43 6D E2 39
    66 6A 65 CB C3 C4 E2 CC F5 49 39 A3 8B 93 5A DD
    B0 21 0B A8 B2 59 5B 24 59 50 44 89 DC 78 19 51
Trust Status : Enabled
```

`Signature Algorithm` 行显示使用了 `MD5WithRSASignature` 算法。 此算法基于 MD5 ，因此此数字证书不能与 TLS 1.2 CipherSpecs 配合使用。

椭圆曲线与 RSA CipherSpecs 的互操作性

并非所有 CipherSpecs 都可以与所有数字证书配合使用。 CipherSpecs 由 CipherSpec 名称前缀表示。 每种类型的 CipherSpec 都对可使用的数字证书类型施加了不同的限制。 这些限制适用于所有 IBM MQ TLS 连接，但与椭圆曲线密码术的用户特别相关。

下表总结了 CipherSpecs 与数字证书之间的关系:

表 *4: CipherSpecs* 与数字证书之间的关系

类型	**CipherSpec 名称前缀**	描述	必需的公用密钥类型	数字签名加密算法	密钥建立方法
1	ECDHE_ECDSA_	使用椭圆曲线公用密钥，椭圆曲线密钥和椭圆曲线数字签名算法的 CipherSpecs 。	Elliptic Curve	ECDSA	(ECDHE)
2	ECDHE_RSA_	使用 RSA 公用密钥，椭圆曲线密钥和 RSA 数字签名算法的 CipherSpecs 。	RSA	RSA	(ECDHE)
3	(所有 TLS 1.3 CipherSpecs)	CipherSpecs ，使用椭圆曲线或 RSA 公用密钥，椭圆曲线密钥以及椭圆曲线或 RSA 数字签名算法。	椭圆曲线或 RSA	ECDSA 或 RSA	ECDHE 或 RSA
4	(所有其他)	使用 RSA 公用密钥和 RSA 数字签名算法的 CipherSpecs 。	RSA	RSA	RSA

注: IBM i 平台上的 IBM MQ 队列管理器和 MQI 客户机不支持类型 1 和 2 CipherSpecs 。

必需的公用密钥类型列显示使用每种类型的 CipherSpec 时个人证书必须具有的公用密钥类型。 个人证书是向其远程合作伙伴标识队列管理器或客户机的最终实体证书。

必须确保证书标签中指定的证书适用于通道 CipherSpec。 即，如果使用需要椭圆曲线 (EC) 证书的 CipherSpec 配置通道，那么不能在证书标签中指定 RSA 证书。 如果使用需要 RSA 证书的 CipherSpec 配置通道，那么无法在证书标签中指定 EC 证书。

假定您已正确配置 IBM MQ，那么可以执行以下操作:

- 具有混合 RSA 和 EC 证书的单个队列管理器。
- 同一队列管理器上使用 RSA 或 EC 证书的不同通道。

数字签名加密算法是指用于验证同级的加密算法。 加密算法与散列算法 (例如 MD5， SHA-1 或 SHA-256) 一起用于计算数字签名。 可以使用各种数字签名算法，例如，具有 MD5 的 RSA 或具有 SHA-256 的 ECDSA。 在表中， ECDSA 是指使用 ECDSA 的数字签名算法集 ;RSA 是指使用 RSA 的数字签名算法集。 可以使用该集合中任何受支持的数字签名算法，前提是该算法基于规定的加密算法。

类型 1 CipherSpecs 要求个人证书必须具有椭圆曲线公用密钥。 使用这些 CipherSpecs 时，将使用椭圆曲线 diffie Hellman Ephemeral 密钥协议来建立连接的密钥。

类型 2 CipherSpecs 要求个人证书具有 RSA 公用密钥。 使用这些 CipherSpecs 时，将使用椭圆曲线 diffie Hellman Ephemeral 密钥协议来建立连接的密钥。

类型 3 CipherSpecs 要求个人证书必须具有 RSA 公用密钥。 使用这些 CipherSpecs 时， RSA 密钥交换用于建立连接的密钥。

此限制列表并非详尽无遗: 根据配置，可能存在其他限制，这些限制会进一步影响互操作能力。 例如，如果 IBM MQ 配置为符合 FIPS 140-2 或 NSA Suite B 标准，那么这也将限制允许的配置范围。 请参阅以下部分以获取更多信息。

如果需要在同一队列管理器或客户机应用程序上使用不同类型的 CipherSpec ，请在客户机定义上配置相应的证书标签和 CipherSpec 组合。

三种类型的 CipherSpec 不会直接互操作: 这是当前 TLS 标准的限制。 例如，假设您已选择将 ECDHE_ECDSA_AES_128_CBC_SHA256 CipherSpec 用于名为 QM1 的队列管理器上名为 TO.QM1 的接收方通道，那么接收方应该具有具有椭圆曲线密钥和基于 ECDSA 的数字签名的个人证书。 如果接收方通道不满足这些要求，那么通道无法启动。

连接到队列管理器 QM1 的其他通道可以使用其他 CipherSpecs，前提是每个通道对该通道的 CipherSpec 使用正确类型的证书。 例如，假设 QM1 使用名为 TO.QM2 ，用于将消息发送到另一个名为 QM2 的队列管理器。 通道 TO.QM2 可以使用类型 3 CipherSpec TLS_RSA_WITH_AES_256_CBC_SHA256 ，提供了包含 RSA 公用密钥的通道使用证书的两端。 证书标签通道属性可用于为每个通道配置不同的证书。

规划 IBM MQ 网络时，请仔细考虑哪些通道需要 TLS，并确保用于每个通道的证书类型适合于与该通道上的 CipherSpec 配合使用。

要查看数字证书的数字签名算法和公用密钥类型，可以使用 **runmqakm** 命令:

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

其中 *cert_label* 是需要显示其数字签名算法的证书的标签。 请参阅 数字证书标签 以获取详细信息。

运行 **runmqakm** 命令将生成显示公用密钥类型的输出:

```
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
```

在此情况下， "公用密钥类型" 行显示证书具有椭圆曲线公用密钥。 本例中的 "签名算法" 行显示正在使用 EC_ecdsa_with_SHA384 算法: 此算法基于 ECDSA 算法。 因此，此证书仅适用于与类型 1 CipherSpecs 配合使用。

TLS 1.3 CipherSpecs

TLS 1.3 CipherSpecs 支持 ECDSA 和 RSA 证书。

椭圆曲线 CipherSpecs 和 NSA Suite B

当 IBM MQ 配置为符合符合符合套件 B 的 TLS 1.2 概要文件时，将限制允许的 CipherSpecs 和数字签名算法，如 第 35 页的『IBM MQ 中的 NSA Suite B 密码术』中所述。 此外，可接受的椭圆曲线键的范围会根据所配置的安全级别而减小。

在 128 位 Suite B 安全级别，证书主体集的公用密钥需要使用 NIST P-256 或 NIST P-384 椭圆曲线，并使用 NIST P-256 椭圆曲线或 NIST P-384 椭圆曲线进行签名。 **runmqakm** 命令可用于使用 -sig_alg 参数 EC_ecdsa_with_SHA256 或 EC_ecdsa_with_SHA384 来请求此安全级别的数字证书。

在 192 位 Suite B 安全级别，证书主体集的公用密钥需要使用 NIST P-384 椭圆曲线并使用 NIST P-384 椭圆曲线进行签名。 **runmqakm** 命令可用于使用 -sig_alg 参数 EC_ecdsa_with_SHA384 来请求此安全级别的数字证书。

支持的 NIST 椭圆曲线如下所示:

表 *5:* 支持的 *NIST* 椭圆曲线		
NIST FIPS 186-3 曲线名称	**RFC 4492 曲线名称**	**椭圆曲线键大小 (位)**
P-256	secp256r1	256
P-384	secp384r1	384
P-521	secp521r1	521

注: NIST P-521 椭圆曲线不能用于符合 Suite B 的操作。

相关概念

第 368 页的『启用 CipherSpecs』
通过在 **DEFINE CHANNEL** 或 **ALTER CHANNEL** MQSC 命令中使用 **SSLCIPH** 参数来启用 CipherSpec 。

第 240 页的『指定运行时在 MQI 客户机上仅使用经过 FIPS 认证的 CipherSpecs』
使用符合 FIPS 的软件创建密钥存储库，然后指定通道必须使用经 FIPS 认证的 CipherSpecs。

第 35 页的『IBM MQ 中的 NSA Suite B 密码术』
本主题提供有关如何配置 IBM MQ for AIX, Linux, and Windows 以符合符合套件 B 的 TLS 1.2 概要文件的信息。

第 20 页的『国家安全局 (NSA) 套件 B 密码术』
美利坚合众国政府就 IT 系统和安全 (包括数据加密) 提供技术咨询。 美国国家安全局 (NSA) 在其 Suite B 标准中推荐了一组可互操作的密码算法。

通道认证记录

要在通道级别对授予连接系统的访问权进行更为精确的控制，可以使用通道认证记录。

您可能会发现客户机尝试使用空白用户标识，或使用允许客户机执行您不准许的操作的高级别用户标识连接到您的队列管理器。 您可以使用通道认证记录来阻止这些客户机的访问。 或者，客户机可能断言了在客户机平台上有效，但在服务器平台上未知或格式无效的用户标识。 您可以使用通道认证记录将断言的用户标识映射到有效的用户标识。

您可能会发现一个客户机应用程序连接到您的队列管理器，但在某种方面行为不当。 要防止服务器出现该应用程序导致的问题，需要使用该客户机应用程序所在的 IP 地址暂时阻止该应用程序，直至出现防火墙规则更新或该客户机应用程序得到纠正之类的情况为止。 您可使用通道认证记录来阻止客户机应用程序进行从中进行连接的 IP 地址。

如果您已为该特定用途设置了某个管理工具（例如，IBM MQ Explorer）和一个通道，那么可能希望确保只有特定客户机计算机可以使用该通道。 您可以使用通道认证记录以仅允许从特定 IP 地址使用该通道。

如果您正要开始使用作为客户机运行的某些样本应用程序，请参阅准备并运行样本程序，以获取有关使用通道认证记录安全地设置队列管理器的示例。

要获取通道认证记录以控制入站通道，请使用 MQSC 命令 **ALTER QMGR CHLAUTH(ENABLED)**。

对于在新入站连接的响应中创建的通道 MCA，将应用 **CHLAUTH** 规则。 对于在本地启动的通道的响应中创建的通道 MCA，将不会应用任何 **CHLAUTH** 规则。

表 *6:* 将针对不同的通道对应用 *CHLAUTH* 规则	
通道类型	**MCA，在其中应用 CHLAUTH 规则**
SDR-RCVR	RCVR
RQSTR-SVR（在 SVR 上启动）	RQSTR
RQSTR-SVR（在 RQSTR 上启动）	SVR
RQSTR-SDR（在 SDR 上启动）	RQSTR
RQSTR-SDR（在 RQSTR 上启动）	用于初始连接的 SDR。 用于回调连接的 RQSTR。

可以创建通道认证记录以执行以下功能：

- 阻止来自特定 IP 地址的连接。
- 阻止来自特定用户标识的连接。
- 设置 MCAUSER 值，以用于任何从特定 IP 地址进行连接的通道。
- 设置 MCAUSER 值，以用于任何断言特定用户标识的通道。
- 设置 MCAUSER 值，以用于任何具有特定 SSL 或 TLS 专有名称 (DN) 的通道。
- 设置 MCAUSER 值，以用于任何从特定队列管理器进行连接的通道。
- 阻止声明来自特定队列管理器的连接，除非该连接来自特定 IP 地址。
- 阻止提供特定 SSL 或 TLS 证书的连接，除非该连接来自特定 IP 地址。

这些用法会在以下部分中进一步说明。

使用 MQSC 命令 **SET CHLAUTH** 或 PCF 命令 **Set Channel Authentication Record** 创建，修改或除去通道认证记录。

注: 大量通道认证记录会对队列管理器的性能产生负面影响。

阻止 IP 地址

阻止来自特定 IP 地址的访问通常是防火墙的角色。 但是，您可能会遇到如下情况：从某个不应该有权访问 IBM MQ 系统的 IP 地址发起了连接尝试，并且您必须临时阻止该地址才能更新防火墙。 这些连接尝试可能不是来自 IBM MQ 通道；这些连接尝试可能来自错误配置为以您的 IBM MQ 侦听器为目标的其他套接字应用程序。 请通过设置 BLOCKADDR 类型的通道认证记录来阻止 IP 地址。 您可以指定一个或多个单一地址、地址范围或包含通配符的模式。

每当入站连接由于以此方式阻止 IP 地址而遭拒时，会发出一条事件消息 MQRC_CHANNEL_BLOCKED，其中包含原因限定符 MQRQ_CHANNEL_BLOCKED_ADDRESS，前提是通道事件已启用并且队列管理器正在运行。 此外，连接会保持打开 30 秒，然后再返回错误，以确保侦听器不会因被阻止的连接尝试不断重复而疲于应付。

要仅阻止特定通道上的 IP 地址，或避免在报告错误前的延迟，请使用 USERSRC(NOACCESS) 参数设置 ADDRESSMAP 类型的通道认证记录。

每当入站连接由于该原因而遭拒时，会发出一条事件消息 MQRC_CHANNEL_BLOCKED，其中包含原因限定符 MQRQ_CHANNEL_BLOCKED_NOACCESS，前提是通道事件已启用并且队列管理器正在运行。

请参阅第 337 页的『阻止特定 IP 地址』，以获取示例。

阻止用户标识

要阻止特定用户标识通过客户机通道进行连接，请设置 BLOCKUSER 类型的通道认证记录。 该类型的通道认证记录仅适用于客户机通道，而不适用于消息通道。 您可以指定一个或多个要阻止的个别用户标识，但是不能使用通配符。

每当入站连接由于该原因而遭拒时，会发出一条事件消息 MQRC_CHANNEL_BLOCKED，其中包含原因限定符 MQRQ_CHANNEL_BLOCKED_USERID，前提是通道事件已启用。

请参阅第 339 页的『阻止特定用户标识』，以获取示例。

您还可通过使用 USERSRC(NOACCESS) 参数设置 USERMAP 类型的通道认证记录，在特定通道上阻止指定用户标识的任何访问。

每当入站连接由于该原因而遭拒时，会发出一条事件消息 MQRC_CHANNEL_BLOCKED，其中包含原因限定符 MQRQ_CHANNEL_BLOCKED_NOACCESS，前提是通道事件已启用并且队列管理器正在运行。

请参阅第 341 页的『阻止访问客户机用户标识』，以获取示例。

阻止队列管理器名称

要规定任何从指定队列管理器进行连接的通道都没有访问权，请使用 USERSRC(NOACCESS) 参数设置 QMGRMAP 类型的通道认证记录。 您可指定单个队列管理器名称或包含通配符的模式。 没有与 BLOCKUSER 功能对等的功能可用于阻止来自队列管理器的访问。

每当入站连接由于该原因而遭拒时，会发出一条事件消息 MQRC_CHANNEL_BLOCKED，其中包含原因限定符 MQRQ_CHANNEL_BLOCKED_NOACCESS，前提是通道事件已启用并且队列管理器正在运行。

请参阅第 341 页的『阻止来自远程队列管理器的访问』，以获取示例。

阻止 SSL 或 TLS DN

要规定任何提供包含指定 DN 的 SSL 或 TLS 个人证书的用户都没有访问权，请使用 USERSRC(NOACCESS) 参数设置 SSLPEERMAP 类型的通道认证记录。 您可指定单个专有名称或包含通配符的模式。 没有与 BLOCKUSER 功能对等的功能可用于阻止对 DN 的访问。

每当入站连接由于该原因而遭拒时，会发出一条事件消息 MQRC_CHANNEL_BLOCKED，其中包含原因限定符 MQRQ_CHANNEL_BLOCKED_NOACCESS，前提是通道事件已启用并且队列管理器正在运行。

请参阅第 342 页的『阻止 SSL 或 TLS 专有名称的访问』，以获取示例。

将 IP 地址映射到要使用的用户标识

要规定任何从指定 IP 地址进行连接的通道都将使用特定 MCAUSER，请设置 ADDRESSMAP 类型的通道认证记录。 您可指定单个地址、地址范围或包含通配符的模式。

如果您使用端口转发器、DMZ 会话中断或任何其他会更改提供给队列管理器的 IP 地址的设置，那么映射 IP 地址就不一定适合您的用途。

请参阅第 342 页的『将 IP 地址映射到 MCAUSER 用户标识』，以获取示例。

将队列管理器名称映射到要使用的用户标识

要规定任何从指定队列管理器进行连接的通道都将使用特定 MCAUSER，请设置 QMGRMAP 类型的通道认证记录。 您可指定单个队列管理器名称或包含通配符的模式。

请参阅第 339 页的『将远程队列管理器映射到 MCAUSER 用户标识』，以获取示例。

将客户机断言的用户标识映射到要使用的用户标识

要指定当特定用户标识由来自 IBM MQ MQI 客户机的连接使用时将使用所指定的不同 MCAUSER，请设置 USERMAP 类型的通道认证记录。 用户标识映射不使用通配符。

请参阅第 340 页的『将客户机用户标识映射到 MCAUSER 用户标识』，以获取示例。

将 SSL 或 TLS DN 映射到要使用的用户标识

要规定任何提供包含指定 DN 的 SSL/TLS 个人证书的用户将使用特定 MCAUSER，请设置 SSLPEERMAP 类型的通道认证记录。 您可指定单个专有名称或包含通配符的模式。

请参阅第 340 页的『将 SSL 或 TLS 专有名称映射到 MCAUSER 用户标识』，以获取示例。

根据 IP 地址映射队列管理器、客户机或者 SSL 或 TLS DN

在某些情况下，第三方可能会冒用队列管理器名称。 SSL 或 TLS 证书或者密钥数据库文件也可能被窃取并复用。 要针对这些威胁进行防护，您可以规定来自特定队列管理器或客户机的连接或者使用特定 DN 的连接必须从指定的 IP 地址进行连接。 设置类型为 USERMAP、QMGRMAP 或 SSLPEERMAP 的通道认证记录，并使用 ADDRESS 参数指定允许的 IP 地址或 IP 地址模式。

请参阅第 339 页的『将远程队列管理器映射到 MCAUSER 用户标识』，以获取示例。

通道认证记录之间的交互

尝试进行连接的通道有可能与多个通道认证记录匹配，从而产生冲突。 例如，某个通道可能断言了由 BLOCKUSER 通道认证记录阻止的某个用户标识，但该用户标识具有与用于设置另一个用户标识的 SSLPEERMAP 记录匹配的 SSL 或 TLS 证书。 此外，如果通道认证记录使用通配符，那么单个 IP 地址、队列管理器名称或者 SSL 或 TLS DN 可能与多个模式匹配。 例如，IP 地址 192.0.2.6 与模式 192.0.2.0-24、192.0.2.* 及 192.0.*.6 匹配。 采取的操作按如下所示来确定。

- 使用的通道认证记录按如下所示进行选择：
  - 与通道名称显式匹配的通道认证记录优先于通过使用通配符与通道名称匹配的通道认证记录。

  - 使用 SSL 或 TLS DN 的通道认证记录优先于使用用户标识、队列管理器名称或 IP 地址的记录。
  - 使用用户标识或队列管理器名称的通道认证记录优先于使用 IP 地址的记录。
- 如果找到匹配的通道认证记录并且它指定了 MCAUSER，那么会将该 MCAUSER 分配给通道。
- 如果找到匹配的通道认证记录并且它规定该通道没有访问权，那么会将 MCAUSER 值 *NOACCESS 分配给通道。 该值以后可由安全出口程序进行更改。
- 如果找不到匹配的通道认证记录，或者找到了匹配的通道认证记录但它规定将使用通道的用户标识，那么会检查 MCAUSER 字段。
  - 如果 MCAUSER 字段为空白，那么会将客户机用户标识分配给通道。
  - 如果 MCAUSER 字段不为空白，那么会将该字段分配给通道。
- 运行任何安全出口程序。 该出口程序可能设置通道用户标识或者确定将阻止访问。
- 如果连接被阻止，或者 MCAUSER 设置为 *NOACCESS，那么通道结束。
- 如果连接未被阻止，那么针对除客户机通道以外的任何其他通道，将根据被阻止用户的列表来检查先前步骤中确定的通道用户标识。
  - 如果用户标识位于被阻止用户的列表中，那么通道结束。
  - 如果用户标识不在被阻止用户的列表中，那么通道运行。

如果多个通道认证记录与通道名称、IP 地址、主机名、队列管理器名称或者 SSL 或 TLS DN 匹配，那么将使用最具体的匹配项。 此匹配项被认为是：

- 最具体是指名称没有通配符，例如：
  - 通道名称 A.B.C
  - IP 地址 192.0.2.6
  - 主机名 hursley.ibm.com
  - 队列管理器名称 192.0.2.6
- 最通用是指匹配的单个星号 (*)，例如：
  - 所有通道名称
  - 所有 IP 地址
  - 所有主机名
  - 所有队列管理器名称
- 在字符串开头使用星号的模式比在字符串开头使用定义值的模式更通用：
  - 对于通道，*.B.C 比 A.* 更通用
  - 对于 IP 地址，*.0.2.6 比 192.* 更通用
  - 对于主机名，*.ibm.com 比 hursley.* 更通用
  - 对于队列管理器名称，*QUEUEMANAGER 比 QUEUEMANAGER* 更通用
- 在字符串中特定位置使用星号的模式比在字符串中相同位置使用定义值的模式更通用，对于字符串中后续每个位置同样如此：
  - 对于通道，A.*.C 比 A.B.* 更通用
  - 对于 IP 地址，192.*.2.6 比 192.0.*. 更通用
  - 对于主机名，hursley.*.com 比 hursley.ibm.* 更通用
  - 对于队列管理器名称，Q*MANAGER 比 QUEUE* 更通用
- 如果两个或更多模式在字符串中特定位置使用星号，那么星号后的节点更少的模式更通用：
  - 对于通道，A.* 比 A.*.C 更通用
  - 对于 IP 地址，192.* 比 192.*.2.* 更通用。
  - 对于主机名，hurlsey.* 比 hursley.*.com 更通用
  - 对于队列管理器名称，Q* 比 Q*MGR 更通用

- 此外，对于 IP 地址：
  - 以连字符 (-) 表示的范围比星号更具体。 因此，192.0.2.0-24 比 192.0.2.* 更具体。
  - 如果某个范围是另一个范围的子集，那么子集范围更具体。 因此，192.0.2.5-15 比 192.0.2.0-24 更具体。
  - 不允许重叠范围。 例如，您不能同时具有针对 192.0.2.0-15 和 192.0.2.10-20 的通道认证记录。
  - 模式中包含的部分数量不能少于必需值，除非模式以单个尾部星号结束。 例如，192.0.2 无效，但 192.0.2.* 有效。
  - 尾部星号必须通过相应的部分分隔符（对于 IPv4 为点 (.)，对于 IPv6 为冒号 (:)）与地址的其余部分分隔开。 例如，192.0* 无效，因为星号未独自成为一部分。
  - 模式可以包含额外的星号，前提是没有星号与尾部星号相邻。 例如，192.*.2.* 有效，但 192.0.*.* 无效。
  - IPv6 地址模式不能包含一个双冒号和一个尾部星号，因为生成的地址会有歧义。 例如，2001::* 可以展开为 2001:0000:*、2001:0000:0000:*，等等。
- 对于 SSL 或 TLS 专有名称 (DN)，子串的优先顺序如下所示：

表 7: 子串的优先顺序

顺序	DN 子串	名称
1	SERIALNUMBER=	证书序列号
2	MAIL=	电子邮件地址
3	Deprecated E=	电子邮件地址（不推荐，最好使用 MAIL）
4	UID=, USERID=	用户标识
5	CN=	公共名称
6	T =	标题
7	OU=	组织单位
8	DC=	域组件
9	O=	组织
10	STREET=	街道/地址第一行
11	L=	地区
12	ST=, SP=, S=	省/直辖市/自治区名称
13	PC=	邮政编码
14	C=	国家或地区
15	UNSTRUCTUREDNAME=	主机名
16	UNSTRUCTUREDADDRESS=	IP 地址
17	DNQ=	专有名称限定符

因此，如果提供的 SSL 或 TLS 证书具有包含子串 O=IBM 和 C=UK 的 DN，那么 IBM MQ 在 O=IBM 和 C=UK 的通道认证记录均存在的情况下优先使用前者。

一个 DN 可包含多个 OU，这些 OU 必须按照分层顺序指定，首先指定较大的组织单位。 如果两个 DN 在除 OU 值以外的其他所有方面都相同，那么将以如下方式来确定更具体的 DN：

1. 如果它们具有不同数量的 OU 属性，那么具有最多 OU 值的 DN 更具体。 这是因为，具有更多组织单位的 DN 能够更加详细地完全限定 DN，并且提供更多匹配条件。 即使其顶级 OU 是通配符 (OU=*)，具有更多 OU 的 DN 仍被视为在总体上更具体。

2. 如果它们具有相同数量的 OU 属性，那么将根据以下规则，按照从左到右的顺序来比较相应的 OU 值对，其中最左侧的 OU 是最高级别（最不具体）。
   a. 不含通配符值的 OU 最具体，因为它只能与一个字符串精确匹配。
   b. 在开头或结尾包含单个通配符的 OU（例如，OU=ABC* 或 OU=*ABC）是次最具体的。
   c. 包含两个通配符的 OU（例如，OU=*ABC*）是再次最具体的。
   d. 只由一个星号组成的 OU (OU=*) 最不具体。
3. 如果字符串比较发现两个属性值的具体程度相同，那么较长的属性字符串更具体。
4. 如果字符串比较发现两个属性值的具体程度和长度相同，那么结果由 DN 部分（不包含任何通配符）的不区分大小写的字符串比较来确定。

如果两个 DN 在除 DC 值之外的所有方面都相同，那么适用与 OU 相同的匹配规则，只是在 DC 值中，最左侧的 DC 是最低级别（最具体），并且比较顺序也相应地有所不同。

显示通道认证记录

要显示通道认证记录，请使用 MQSC 命令 **DISPLAY CHLAUTH** 或 PCF 命令 **Inquire Channel Authentication Records**。 您可选择返回与提供的通道名称匹配的所有记录，也可选择某个显式匹配项。 显式匹配项表明，当通道尝试从特定 IP 地址、从特定队列管理器或使用特定用户标识进行连接时，以及（可选）通过提供包含指定 DN 的 SSL/TLS 个人证书进行连接时，将使用哪个通道认证记录。

相关概念

第 85 页的『远程消息传递的安全性』
本部分涉及安全性的远程消息传递方面。

CHLAUTH 和 *CONNAUTH* 的交互

通道认证记录 (CHLAUTH) 和连接认证 (CONNAUTH) 在 IBM MQ 中的交互方式 (对于通道上的单个对话)。

不同类型的绑定

IBM MQ 支持两种方法供应用程序连接:

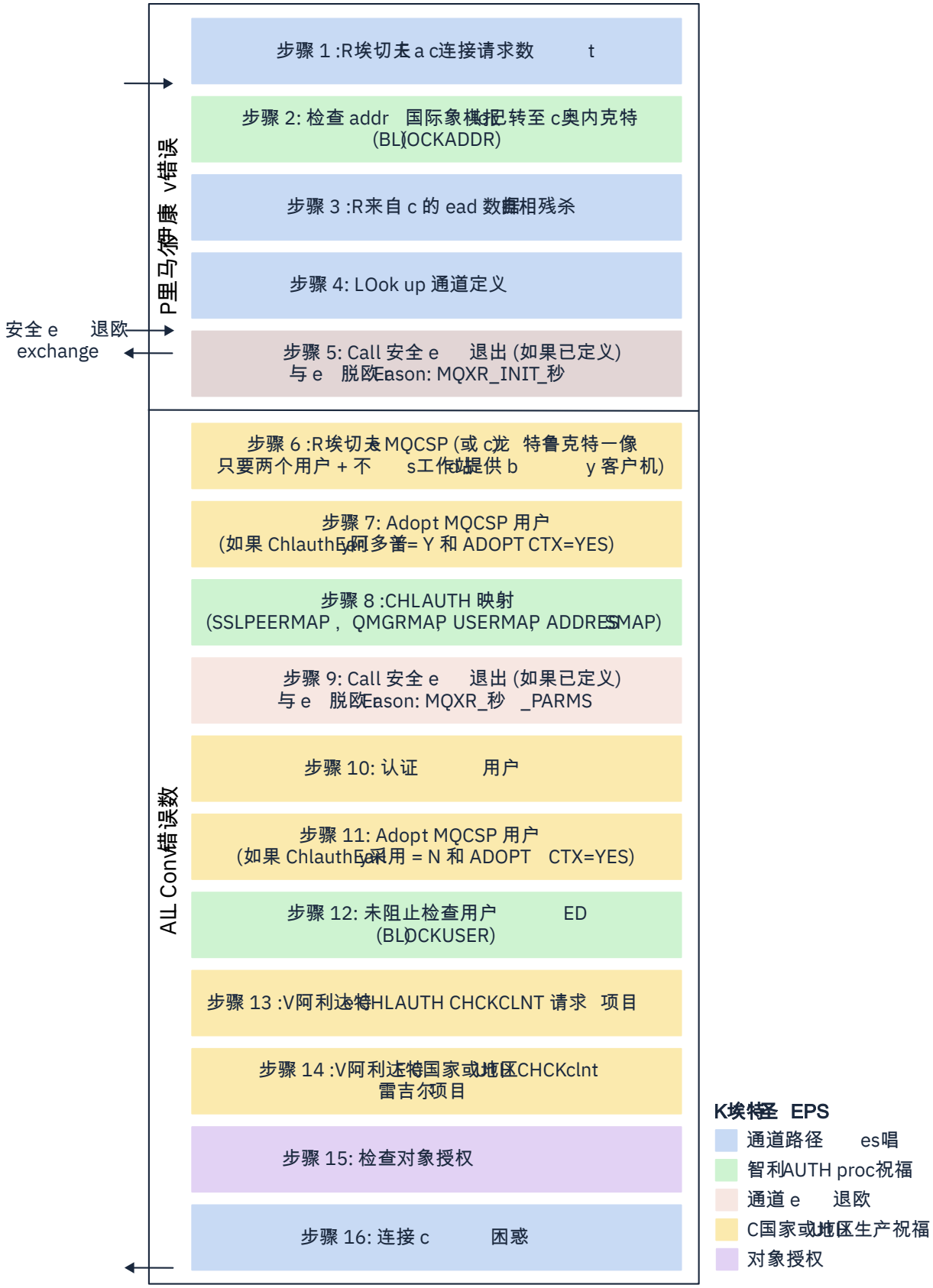
本地绑定
在应用程序和队列管理器位于同一操作映像上时应用。 CHLAUTH 与此类型的应用程序连接无关。

客户机绑定
当应用程序和队列管理器使用网络进行通信时应用。 应用程序和队列管理器可以在同一机器上运行，也可以在不同机器上运行。 在 IBM MQ 中，以服务器连接 (SVRCONN) 通道的形式处理客户机连接，在此情况下， CONNAUTH 和 CHLAUTH 都适用。

通道接收端的绑定步骤

当应用程序连接到队列管理器时，将执行大量检查以确保通道两端都了解另一端支持的内容。 通道的接收端执行一些额外的检查，包括 CHLAUTH 和 CONNAUTH ，以确保允许客户机连接，并且此过程还可能包含安全出口，因为这可能会影响结果。 此通道连接阶段也称为 绑定阶段。

下图列出了 SVRCONN 通道在服务器端 (在队列管理器处) 启动时经过的步骤:



步骤 1: 接收连接请求
通道启动程序或侦听器从网络上的某个位置接收连接请求。

步骤 2: 是否允许该地址进行连接?
在读取任何数据之前， IBM MQ 会根据 CHLAUTH 规则检查合作伙伴的 IP 地址，以查看该地址是否在 *BLOCKADDR* 规则中。 如果找不到地址，因此未阻塞，那么流将继续执行下一步。

步骤 3: 从通道读取数据
IBM MQ 现在将数据读入缓冲区，并开始处理已发送的信息。

步骤 4: 查找通道定义
在第一个数据流中， IBM MQ 除其他外发送发送端尝试启动的通道的名称。 然后，接收队列管理器可以查找通道定义，该定义具有为通道指定的所有设置。

步骤 5: 调用安全性出口 (如果已定义)
如果通道定义了安全出口 (SCYEXIT) ，那么将使用出口原因 (MQCXP.**ExitReason**) 来调用此出口 设置为 MQXR_INIT_SEC。

步骤 6: 接收 MQCSP
如果需要，请构造一个 (如果客户机提供了认证凭证)。

如果客户机是以兼容性方式运行的 Java 或 JMS 应用程序，那么客户机不会将 MQCSP 结构传递到队列管理器。 相反，如果应用程序提供了用户标识和密码，那么将在此处构造 MQCSP 结构。

步骤 7: 采用 MQCSP 用户 (如果 ChlauthEarlyAdopt 为 *Y* 且 ADOPTCTX=YES)
将对客户机提供的凭证进行认证。

如果 CONNAUTH 正在使用 LDAP 将声明的专有名称映射到简短用户标识，那么映射将在此步骤中进行。

如果认证成功，那么通道将采用用户标识并由 CHLAUTH 映射步骤使用。

注: 从 IBM MQ 9.0.4 开始， **ChlauthEarlyAdopt=** *Y* 参数将自动添加到新队列管理器的 qm.ini 文件的 channel 节中。

步骤 8 :CHLAUTH 映射
将再次检查 CHLAUTH 高速缓存以查找映射规则 *SSLPEERMAP*， *USERMAP*， *QMGRMAP* 和 *ADDRESSMAP*。

将使用与传入通道最具体匹配的规则。 如果规则具有 **USERSRC***(CHANNEL)* 或 *(MAP)*，那么通道将继续绑定。

如果 CHLAUTH 规则求值为具有 **USERSRC***(NOACCESS)*的规则，那么将阻止应用程序连接到通道，除非随后在步骤 9 中使用有效凭证覆盖凭证。

步骤 9: 调用安全性出口 (如果已定义)
如果通道定义了安全出口 (SCYEXIT) ，那么将使用出口原因 (MQCXP.**ExitReason**) 来调用此出口 设置为 MQXR_SEC_PARMS。

指向 MQCSP 的指针将出现在 MQCXP 结构的 **SecurityParms** 字段中。

MQCSP 结构具有指向用户标识 (MQCSP.**CSPUserIdPtr**) 的指针 和密码 (MQCSP.**CSPPasswordPtr**)。

V 9.4.0 从 IBM MQ 9.3.4 开始， MQCSP 结构还包含指向认证令牌 (MQCSP.**TokenPtr**) 的指针。

可以在出口中更改用户标识和密码以及认证令牌 。 以下示例显示了安全出口如何将用户标识和密码值打印到审计日志中:

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

出口可以通过在 MQCXP 中返回 *MQXCC_CLOSE_CHANNEL* 来指示 IBM MQ 关闭通道。**Exitresponse** 字段。 否则，通道处理将继续到连接认证阶段。

注: 如果安全出口更改了声明的用户，那么不会将 CHLAUTH 映射规则重新应用于新用户。

步骤 10: 认证用户

如果在队列管理器上启用了 CONNAUTH，那么将发生认证阶段。

要进行检查，请发出 MQSC 命令 "DISPLAY QMGR CONNAUTH"。

z/OS 以下示例显示了在 IBM MQ for z/OS 上运行的队列管理器中命令 **DISPLAY QMGR CONNAUTH** 的输出。

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

Multi 以下示例显示了来自在 IBM MQ for Multiplatforms 上运行的队列管理器的命令 "**DISPLAY QMGR CONNAUTH**" 的输出。

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

CONNAUTH 值是 **AUTHINFO** IBM MQ 对象的名称。

由于操作系统认证 (**AUTHTYPE**(*IDPWOS*)) 在 IBM MQ for Multiplatforms 和 IBM MQ for z/OS 上都有效，因此示例使用操作系统认证。

z/OS 以下示例显示了在 IBM MQ for z/OS 上运行的队列管理器中具有 **AUTHTYPE**(*IDPWOS*) 的缺省 AUTHINFO 对象。

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

Multi 以下示例显示了在 IBM MQ for Multiplatforms 上运行的队列管理器中具有 **AUTHTYPE**(*IDPWOS*) 的缺省 AUTHINFO 对象。

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

AUTHINFO TYPE (IDPWOS) 对象具有名为 CHCKCLNT 的属性。 如果该值更改为 *REQUIRED*，那么所有客户机应用程序都必须提供有效凭证。

如果在步骤 7 中对用户进行了认证，那么将不会执行其他认证检查，除非:

- 步骤 9 中的安全出口已更改 MQCXP 结构的 SecurityParms 字段中的用户标识和密码或认证令牌 。
- 客户机应用程序已与请求可重新连接功能的选项连接。

步骤 11: 采用 MQCSP 用户的上下文 (如果 ChlauthEarlyAdopt=*N* 且 ADOPTCTX=YES)

您可以设置 ADOPTCTX 属性，该属性控制通道是在 MCAUSER 下运行，还是在应用程序提供的用户标识下运行。

如果在 MQCSP 或 MQXCP 结构的 **SecurityParms** 字段中声明的用户标识已成功认证并且 **ADOPTCTX** 为 *YES*，那么将采用步骤 7 和 8 生成的用户上下文作为要用于此应用程序的上下文，除非步骤 9 中的安全出口已更改 MQCXP 结构的 **SecurityParms** 字段中的用户标识和密码或认证令牌 。

此声明的用户标识是要检查以获取使用 IBM MQ 资源的授权的用户标识。

例如，您在 SVRCONN 通道上未设置 MCAUSER ，并且客户机正在 Linux 机器上的 "johndoe" 下运行。应用程序在 MQCSP 中指定用户 "fred" ，因此通道开始以 "johndoe" 作为活动 MCAUSER 运行。 在 CONNAUTH 检查后，将采用用户 "fred" ，并且通道将以 "fred" 作为活动 MCAUSER 运行。

步骤 12: 检查用户是否未被阻止 (BLOCKUSER)
如果 CONNAUTH 检查成功，那么将再次检查 CHLAUTH 高速缓存，以检查活动的 MCAUSER 是否被 *BLOCKUSER* 规则阻止。 如果用户被阻止，那么通道结束。

步骤 13: 验证 CHLAUTH CHCKCLNT 需求
如果在步骤 8 中选择的 CHLAUTH 规则另外指定 CHCKCLNT 值 REQUIRED 或 REQDADM ，那么将执行验证以确保提供有效的 CONNAUTH 用户标识以满足需求。

- 如果设置了 CHCKCLNT (必需) ，那么必须已在步骤 7 或 10 中认证用户。 否则，将拒绝连接。
- 如果设置了 CHCKCLNT (REQDADM) ，那么必须在步骤 7 或 10 中认证用户 (如果确定此连接具有特权)。 否则，将拒绝连接。
- 如果设置了 CHCKCLNT (ASQMGR) ，那么将跳过此步骤。

注意:

1. 如果设置了 CHCKCLNT (必需) 或 CHCKCLNT (REQDADM) ，但在队列管理器上未启用 CONNAUTH ，那么由于配置中存在冲突，连接将失败并返回 MQRC_SECURITY_ERROR (2063) 返回码。
2. 在此步骤中未重新认证用户。

步骤 14: 验证 CONNAUTH CHCKCLNT 需求。
如果在队列管理器上启用了 CONNAUTH ，那么将发生认证阶段。

检查 CONNAUTH CHCKCLNT 值以确定为入局连接设置的需求:

- 如果设置了 CHCKCLNT (NONE) ，那么将跳过此步骤
- 如果设置了 CHCKCLNT (可选) ，那么将跳过此步骤。
- 如果设置了 CHCKCLNT (REQUIRED) ，那么必须已在步骤 7 或 10 中认证用户。 否则，将拒绝连接。
- 如果设置了 CHCKCLNT (REQDADM) ，那么必须在步骤 7 或 10 中认证用户 (如果确定此连接具有特权)。 否则，将拒绝连接。

注: 在此步骤中未重新认证用户。

Multi **步骤 15: 检查对象授权**
进行检查以确保活动的 MCAUSER 具有连接到队列管理器的相应权限。

ALW 请参阅 对象权限管理器，以获取更多信息。

IBM i 请参阅 第 134 页的『IBM i 上的对象权限管理器』，以获取更多信息。

步骤 16: 连接完成
如果上述步骤成功完成，那么连接将完成。

相关概念
CONNAUTH
可以配置队列管理器以认证应用程序在连接时提供的凭证。

相关参考
SET CHLAUTH
变更授权信息

解决 *CHLAUTH* 访问问题
使用通道认证记录 (CHLAUTH) 时解决某些访问问题的步骤和示例。

开始之前

注: 此任务中的步骤要求您运行 MQSC 命令。 如何执行此操作因平台而异。 请参阅 使用 MQSC 命令管理 IBM MQ。

关于此任务

CHLAUTH 处理有三个缺省规则:

- 任何 MQ-admin* 用户都不接受所有通道
- 不接受所有 SYSTEM.* 所有用户的通道
- ALLOW 访问 SYSTEM.ADMIN.SVRCONN 通道 (非 MQ-admin 用户)

前两个规则会阻止对所有通道的访问。 如果通道是 SYSTEM.ADMIN.SVRCONN 通道，因此允许在该通道上进行访问。

CHLAUTH 规则用于确定是否可以启动通道，并且它们允许通过 MCAUSER 映射到另一个用户标识。 如果无法启动通道，那么通常会发生以下错误:

- RC 2035 MQRC_NOT_AUTHORIZED
- RC 2059 MQRC_Q_MGR_NOT_AVAILABLE
- AMQ4036 不允许访问
- AMQ9776: 通道已被用户标识阻塞
- AMQ9777: 通道已阻塞
- MQJE001: 发生 MQException: 完成代码 2 ，原因 2035
- MQJE036: 队列管理器已拒绝连接尝试

您应该严格阻止访问，然后添加更多 CHLAUTH 规则以控制谁可以访问和启动通道。

作为临时措施，要对列出的错误进行故障诊断，请完成以下任何步骤。

过程

- **禁用 CHLAUTH 规则**

  作为临时措施，并且为了对以上错误进行故障诊断，您可以禁用 CHLAUTH 规则。 可以随时重新启用这些规则，如果禁用 CHLAUTH 规则可解决连接问题，那么您知道这是原因。

  要禁用 CHLAUTH 规则，请运行以下 MQSC 命令:

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

  请注意，您还可以将 CHLAUTH 设置为 *WARN*，这将允许访问并记录规则的结果。

- **修改或除去 CHLAUTH 规则**

  您还可以删除或修改 CHLAUTH 规则或规则，从而导致问题。

  要修改 CHLAUTH 规则，请使用带有 ACTION (REPLACE) 的 SET CHLAUTH 命令。 例如，要修改导致任何 MQ-admin 用户无法访问 WARN 的所有通道的缺省规则，而不是被阻止，请运行以下 MQSC 命令:

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

  要删除 CHLAUTH 规则，请将 SET CHLAUTH 命令与 ACTION (REMOVE) 配合使用。 例如，要删除导致任何 MQ-admin 用户无法访问所有通道的缺省规则，请运行以下 MQSC 命令:

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

- **使用 MATCH (RUNCHECK) 测试访问权**

您可以使用 CHLAUTH 规则的 MATCH (*RUNCHECK*) 选项来测试 CHLAUTH 规则的结果。 **MATCH** (*RUNCHECK*) 选项返回在运行时由特定入站通道匹配的记录 (如果该通道连接到此队列管理器)。 您必须提供:

- 通道名
- “地址”属性
- SSLPEER 属性，仅当入站通道使用 SSL 或 TLS 时
- QMNAME ，如果入站通道是队列管理器通道，或者
- CLNTUSER 属性 (如果入站通道是客户机通道)

以下示例运行 MQSC 命令以检查具有缺省规则的 CHLAUTH 规则将导致 MQ-admin 用户 johndoe 访问名为 CHAN1 的通道:

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

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

对于用户 johndoe，通道未运行，将由于 *MQADMIN 用户的 BLOCKUSER 规则而阻止用户。

以下示例运行 MQSC 命令以检查具有缺省规则的 CHLAUTH 规则将导致用户 alice (非 MQ-admin 用户) 访问名为 CHAN1 的通道:

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

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

对于用户 alice，通道将运行，并且通道将 alice 作为 MCAUSER 传入。 MCAUSER 是用于检查 IBM MQ 对象权限的用户标识。

相关参考

SET CHLAUTH

显示 CHLAUTH

为用户创建新的 *CHLAUTH* 规则

用户的一些常见场景，以及用于实现这些场景的示例 CHLAUTH 规则。

开始之前

注: 此任务中的步骤要求您运行 MQSC 命令。 如何执行此操作因平台而异。 请参阅 使用 MQSC 命令管理 IBM MQ。

关于此任务

CHLAUTH 处理有三个缺省规则:

- 任何 MQ-admin* 用户都不接受所有通道
- 不接受所有 SYSTEM.* 所有用户的通道
- ALLOW 访问 SYSTEM.ADMIN.SVRCONN 通道 (非 MQ-admin 用户)

前两个规则会阻止对所有通道的访问。 如果通道是 SYSTEM.ADMIN.SVRCONN 通道，因此允许在该通道上进行访问。

要为用户创建新的 CHLAUTH 规则，请配置以下一个或多个方案。

过程

- **控制特定 MQ-admin 用户的访问权**

a) 设置专门用于管理透视图 (即，从 IBM MQ Explorer 进行连接) 的服务器连接通道。

如果连接不是来自其中一个指定的 IP 地址，那么您具有此用途的特定通道，定义的一个或多个 IP 地址 (您希望从其中接受连接) 以及针对 'mqm' 标识的访问受阻。

b) 为 IBM MQ Explorer 和 MQ-admin 用户创建名为 ADMIN.CHAN 的 SVRCONN 通道。

运行以下 MQSC 命令:

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

c) 要进行测试，请确保您具有 MQ-admin 组中定义的用户，但没有定义的用户。

对于此方案， mqadm 位于 MQ-admin 组中，而 alice 不在此组中。

d) 确认 缺省 CHLAUTH 规则 已就绪。

e) 添加三个规则以允许特定用户访问 ADMIN.CHAN 作为 MQ-admin :

- 从任何地址设置 NOACCESS
- Set BLOCKUSER for this channel to only block user nobody, which overrides the *MQADMIN BLOCKUSER
- ALLOW 对特定地址子网上的用户 mqadm 的访问权，以及对 mqadm 用户权限的 MAP 访问权

为此，请运行以下 MQSC 命令:

```
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)
```

此时，用户 mqadm 可以访问和启动 ADMIN.CHAN 通道。

f) 可选： 您可以随时运行 MQSC 命令 MATCH (RUNCHECK) 以查看以下每个命令的结果:

```
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)
```

此时，仅允许具有 CHLAUTH 记录的用户使用 ADMIN.CHAN。

- **控制特定用户和 IBM MQ 客户机应用程序的访问权**

  对于此场景， 缺省 CHLAUTH 规则 已足够，假定应该为特定用户设置 IBM MQ 权限，以提供正确的 IBM MQ 权限 (使用 setmqaut)。

  在此场景中，将为不是 MQ-admin 用户的用户 mqapp1 设置权限。

  a) 使用以下 MQSC 命令来生成 SVRCONN 通道 APP1.CHAN，供特定应用程序和特定用户使用。

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

  b) 通过实施 缺省 CHLAUTH 规则 ，用户 mqapp1 可以启动 APP1.CHAN 频道。

来自 IBM MQ 客户机应用程序的用户标识用于 IBM MQ 对象权限检查。 在这种情况下，假定 mqapp1 用户正在运行 IBM MQ 客户机应用程序，这将用于 IBM MQ 对象权限检查。 因此，如果 mqapp1 有权访问应用程序所需的 IBM MQ 对象，那么所有操作都正常; 如果没有，您将收到权限错误。

您可以通过为 mqapp1 用户标识创建特定 CHLAUTH 规则来进一步提高安全性，但在缺省规则下，MQ-admin 组的任何成员都不能访问此通道。

运行以下 MQSC 命令:

```
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)
```

- **通过使用特定用户的证书专有名称 (DN) 来控制该用户的访问权**

  对于此场景，用户必须具有流向队列管理器的证书。 然后，将 DN 与 CHLAUTH 规则的 SSLPEER 设置进行匹配，并且 SSLPEER 可以使用通配符。

  如果匹配，那么还可以将用户映射到其他 MCAUSER 以检查 IBM MQ 对象权限。 映射 MCAUSER 可以最大限度减少需要在 IBM MQ 对象权限管理器 (OAM) 中管理的用户数。

  a) 您有一个 TLS 通道，其中包含正在使用的证书，并且需要规则来执行以下操作:

  – 阻止特定通道的所有用户

  – 仅允许具有特定 SSLPEER 的用户使用该用户的客户机进行 IBM MQ OAM 访问。

  运行以下 MQSC 命令:

```
.
# 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)
```

  通道上连接的客户机用户标识用于 IBM MQ 对象的 IBM MQ OAM 权限; 因此用户标识必须具有相应的 IBM MQ 权限。

  b) 可选： 映射到其他 IBM MQ 用户标识。

  重新运行先前的 MQSC 命令，将 USERSRC(MAP) MCAUSER('mquser1') 替换为 USERSRC(CHANNEL)。

- **将特定用户映射到 mqm 用户**

  这是对 特定 MQ-admin 用户的控制访问权的添加或修改。

  使用 MQSC 命令来添加以下 CHLAUTH 规则，以将特定用户映射到在 IBM MQ OAM 中设置了 IBM MQ 对象权限的 mqm 用户或 MQ-admin 用户标识。

```
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)
```

  这允许 johndoe 用户并将其映射到特定通道 ADMIN.CHAN 的 mqm 用户。

相关概念

第 57 页的『为通道创建新的 CHLAUTH 规则』
为了帮助您创建自己的 CHLAUTH 规则，以下是通道的一些常见方案以及用于实现这些规则的示例 CHLAUTH 规则。

相关任务

第 52 页的『解决 CHLAUTH 访问问题』
使用通道认证记录 (CHLAUTH) 时解决某些访问问题的步骤和示例。

相关参考

SET CHLAUTH

显示 CHLAUTH

为通道创建新的 *CHLAUTH* 规则

为了帮助您创建自己的 CHLAUTH 规则，以下是通道的一些常见方案以及用于实现这些规则的示例 CHLAUTH 规则。

本主题包含以下场景:

- 第 57 页的『仅允许从特定 IP 地址范围访问特定通道。』
- 第 57 页的『对于特定通道，阻止所有用户，但允许特定用户进行连接。』
- 第 58 页的『将 CHLAUTH 用于接收方和发送方通道』

仅允许从特定 IP 地址范围访问特定通道。

对于此场景，您希望:

- 设置 "无从任何位置访问通道"
- 允许从特定 IP 地址或地址范围进行访问

```
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)
```

这仅允许 APP2.CHAN 通道。

以 MCAUSER 身份连接的用户将映射到 mqapp2，因此将获取该用户的 IBM MQ OAM 权限。

对于特定通道，阻止所有用户，但允许特定用户进行连接。

CHLAUTH 处理有三个缺省规则:

- 任何 MQ-admin* 用户都不接受所有通道
- 不接受所有 SYSTEM.* 所有用户的通道
- ALLOW 访问 SYSTEM.ADMIN.SVRCONN 通道 (非 MQ-admin 用户)

前两个规则会阻止对所有通道的访问。 如果通道是 SYSTEM.ADMIN.SVRCONN 通道，因此允许在该通道上进行访问。

对于此场景，对通道 MY.SVRCONN 的访问具有缺省 CHLAUTH 规则。

您需要添加以下内容:

```
# 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)
```

此代码的第一部分阻止任何人在 MY.SVRCONN 上进行连接，那么当连接来自特定用户标识 johndoe 时，此代码仅允许启动 MY.SVRCONN 通道。

在通道 johndoe 上连接的用户用于 IBM MQ 对象的 IBM MQ OAM 权限。 因此，用户标识必须具有相应的 IBM MQ 权限。

如果要使用以下命令，可以映射到其他 IBM MQ 用户标识:

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

而不是 USERSRC(CHANNEL)。

将 CHLAUTH 用于接收方和发送方通道

您可以使用 CHLAUTH 规则向接收方和发送方通道添加额外的安全性，以限制对接收方通道的访问。 请注意，如果您正在添加或更改 CHLAUTH 规则，那么仅当启动通道时，更新后的 CHLAUTH 规则才适用，因此如果通道已在运行，那么需要停止并重新启动这些通道，以便应用 CHLAUTH 更新。

CHLAUTH 规则可以在任何通道上使用，但存在一些限制。 例如， USERMAP 规则仅适用于 SVRCONN 通道。

此示例仅允许来自特定 IP 地址的连接启动 TO.MYSVR1 通道:

```
# 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)
```

此示例仅允许来自特定队列管理器的连接:

```
# 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)
```

相关任务

第 52 页的『解决 CHLAUTH 访问问题』
使用通道认证记录 (CHLAUTH) 时解决某些访问问题的步骤和示例。

第 54 页的『为用户创建新的 CHLAUTH 规则』
用户的一些常见场景，以及用于实现这些场景的示例 CHLAUTH 规则。

相关参考

SET CHLAUTH

显示 CHLAUTH

创建 *CHLAUTH* 备份/停止规则
在考虑控制到队列管理器的入站连接时，您有两个选项。 可以尝试列出所有不允许的连接，也可以先说所有连接都不允许，然后尝试列出所有允许的连接。 此处描述了第二个选项。

关于此任务

使用第二个选项的原因是，如果您尝试列出所有不允许的连接，而所有未列出的连接都因此被允许，那么从列表中漏掉一个的结果是，应该不允许的连接能够连接，从而导致潜在的安全违规。

相反，如果相反，首先表示不允许每个连接，然后列出这些连接，那么此列表中缺少一个连接的结果不是安全违规。 如果您的企业需要添加额外的连接，这是一项相对简单的任务，但没有潜在的安全漏洞。

首先要执行的操作是创建 *back-stop* 规则，这是一条规则，用于捕获任何未以其他方式与更具体的规则匹配的连接。 此规则的作用是阻止任何远程连接完全无法连接到队列管理器。

但是，如果您关注此方法，那么可以在警告方式下设置 *back-stop* 规则; 请参阅步骤 第 59 页的『2』

过程

1. 要创建用于停止连接到队列 ` 管理器的远程连接的备份/停止规则，请发出以下命令:

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

   现在，您已关闭所有远程连接上的门，可以开始实施更具体的规则，以允许某些连接进入。 例如:

   ```
   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. 如果要以警告方式创建备份/停止规则，请发出以下命令:

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

   现在，您可以继续执行操作，并制定所有积极的规则。 当您认为已创建所需的所有规则时，请通过发出以下命令来开启通道事件:

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

   并监视 SYSTEM.ADMIN.CHANNEL.EVENT 队列， **Reason** 设置为 MQRC_CHANNEL_BLOCKED_WARNING。

   这些事件详细描述了与您的后端停止规则匹配的连接，但由于该命令正在以警告方式运行，因此实际上暂时未被阻止。

   请查看这些事件中的每个事件，并确定此连接是否应具有允许其进入的肯定规则，或者此连接是否与 *back-stop* 规则正确匹配。 您可以在此方式下运行，在创建事件时查看这些事件，直到您满意地看到了所有入站通道，并且为所有这些通道都制定了相应的肯定规则。

   此时，您可以通过发出以下命令来更改 *back-stop* 规则以启动其匹配的真正阻塞连接:

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

创建非特权 *IBM MQ* 管理员
如何使用 CHLAUTH 创建非特权 IBM MQ 管理员。

关于此任务

在此任务的上下文中，术语:

特权用户
表示有权执行操作而未被显式授予执行该操作的访问权的用户。 mqm 组中的用户是这些特权用户的示例。

***IBM MQ* 管理员**
表示需要对 IBM MQ(例如 **DEFINE QLOCAL** 或 **START CHANNEL**) 发出管理命令的用户。

以下步骤将创建非特权 IBM MQ 管理员。

过程

1. 在队列管理器机器上使用适合于您企业所使用的平台的命令来创建用户标识。
   此示例中使用了用户名 `alice` 。
2. 通过执行以下过程，授予此新用户权限以发出所有 IBM MQ 管理命令:

a) 使用特权用户启动 IBM MQ Explorer 。

b) 通过选择相应的队列管理器，然后选择 **对象权限** 和 **添加基于角色的权限**来浏览到 " 基于角色的向导 "。

c) 在弹出的向导面板中，输入您在第一步中创建的用户标识，或者如果您希望使用组，请输入要使其成为非特权 IBM MQ 管理员的用户或用户集的组名。

d) 设置向导以实现完全管理访问权。

e) 如果要允许非特权 IBM MQ 管理员能够浏览队列上的消息，请同时选中该复选框。

f) 查看向导底部的预览面板中的命令。

您可以剪切并粘贴这些命令以构建自己的脚本。

您可能更愿意使用自己的脚本执行此操作的一个原因是减少您授予此用户的访问权量。 您可能希望只授予对特定对象组的访问权，而不是授予对所有对象的访问权。

在向导上按 **确定** 将发出显示的命令。

g) 如果非特权 IBM MQ 管理员也需要远程访问，那么您需要设置一些 CHLAUTH 规则以允许对此用户标识进行远程访问。

假定您的企业正在使用 第 58 页的『创建 CHLAUTH 备份/停止规则』中的指南，那么您需要执行的操作就是添加启用规则。

您创建的规则将取决于您选择如何认证远程 IBM MQ 管理员。

如果您正在使用弱 TCP/IP 认证，那么可以设置如下所示的 CHLAUTH 规则:

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

9。如果您正在使用 TLS 认证，那么可以设置如下所示的 CHLAUTH 规则:

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

现在，当用户连接到 `admin-channel-name` (并与 CHLAUTH 规则匹配) 时，他们能够在队列管理器上以用户标识 `alice` 发出命令，因此不需要特权远程访问。

连接认证

连接认证允许应用程序在连接到队列管理器时提供认证凭证。 队列管理器将验证凭证。 还可以采用凭证中提供的用户标识，以便在应用程序访问的资源的授权检查中使用。

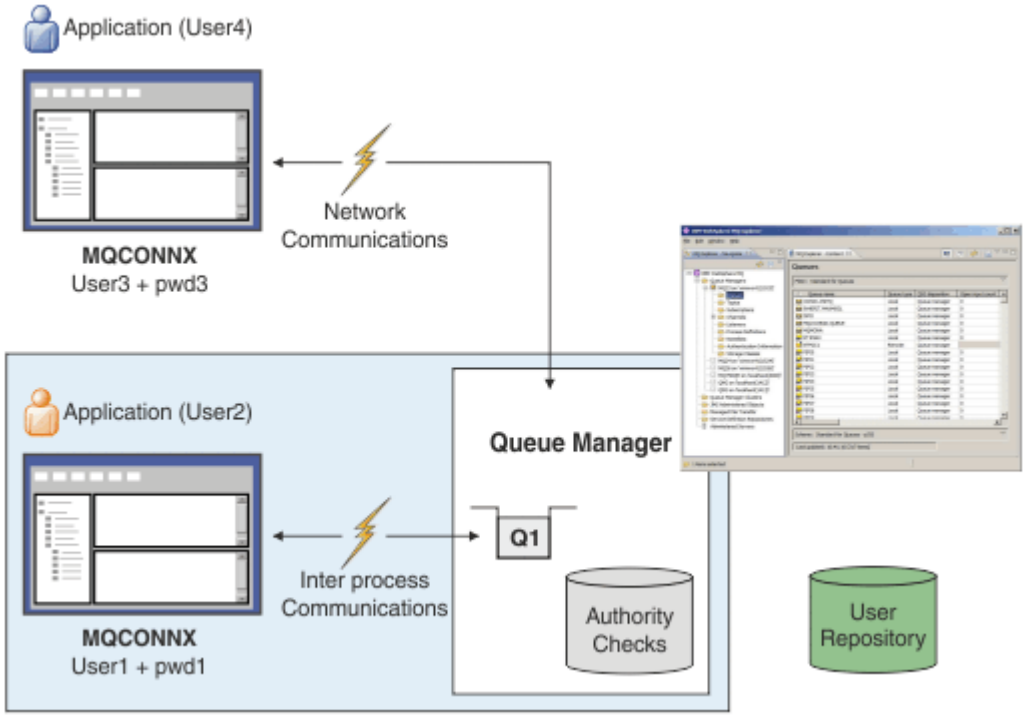
应用程序可以在连接到队列管理器时提供用于认证的用户标识和密码。

V 9.4.0 从 IBM MQ 9.3.4 开始， IBM MQ client 应用程序还可以提供认证令牌作为备用认证方法。

可以配置队列管理器以验证应用程序提供的凭证。

将使用队列管理器配置中的用户存储库来检查应用程序提供的用户标识和密码。 有关用于检查用户标识和密码的存储库的更多信息，请参阅 用户存储库。

V 9.4.0 通过使用队列管理器的令牌认证密钥库中的证书和对称密钥来验证令牌的签名，验证认证令牌。 有关使用认证令牌认证用户的更多信息，请参阅 第 287 页的『使用认证令牌』。



在图中，两个应用程序正在与队列管理器建立连接，一个应用程序作为客户机，另一个使用本地绑定。 应用程序可能使用各种 API 来连接到队列管理器，但所有应用程序都能够提供用户标识和密码。 图中正在运行应用程序的用户标识 User2 和 User4 (这是提供给 IBM MQ 的通常操作系统用户标识) 可能与应用程序 User1 和 User3 提供的用户标识不同。

队列管理器接收配置命令 (在图中，正在使用 IBM MQ Explorer)，并管理资源的打开，并检查访问这些资源的权限。 IBM MQ 中存在许多不同的资源，应用程序可能需要权限才能访问这些资源。 该图说明了打开队列以进行输出，但相同的原则也适用于其他资源。

相关概念

第 61 页的『连接认证: 配置』
可以配置队列管理器以认证应用程序在连接时提供的凭证。

第 65 页的『连接认证: 应用程序更改』

第 66 页的『连接认证: 用户存储库』
对于每个队列管理器，您可以选择不同类型的认证信息对象来认证用户标识和密码。

连接认证: 配置

可以配置队列管理器以认证应用程序在连接时提供的凭证。

在队列管理器上开启连接认证

在队列管理器对象上，可以将 **CONNAUTH** 属性设置为认证信息 (AUTHINFO) 对象的名称。 AUTHINFO 对象的 **AUTHTYPE** 属性指定对象的类型。 用于连接认证的 AUTHINFO 对象可以是以下两种类型之一:

IDPWOS
队列管理器使用本地操作系统来认证连接应用程序提供的用户标识和密码。

Linux V 9.4.0 AIX 从 IBM MQ 9.3.4 开始，此类型的 AUTHINFO 对象还允许在 AIX 或 Linux 上运行的队列管理器验证认证令牌。 除了用于配置连接认证的 AUTHINFO 对象外，还必须将队列管理器配置为接受具有 qm.ini 文件的 **AuthInfo** 节的认证令牌。 有关配置队列管理器以接受认证令牌的更多信息，请参阅 第 293 页的『配置队列管理器以使用本地密钥库接受认证令牌』。

IDPWLDAP
队列管理器使用 LDAP 服务器来认证连接应用程序提供的用户标识和密码。

注: 不能在队列管理器的 **CONNAUTH** 属性中指定任何其他类型的认证信息对象。

类型为 IDPWOS 和 IDPWLDAP 的 AUTHINFO 对象在其多个属性中相似。 此处描述的属性对于这两种类型的对象都是公共的。

以下示例 MQSC 命令通过以下操作开启连接认证:

1. 定义名为 USE.PW 的 AUTHINFO 对象。
2. 更改队列管理器 **CONNAUTH** 属性以引用此 AUTHINFO 对象。
3. 发出 **REFRESH SECURITY** 命令以刷新队列管理器的连接认证配置。 必须先发出 **REFRESH SECURITY** 命令，队列管理器才能识别对连接认证配置所作的任何更改。

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

ALTER QMGR CONNAUTH(USE.PW)

REFRESH SECURITY TYPE(CONNAUTH)
```

要控制是否检查凭证以查找本地绑定的应用程序所建立的连接，请使用 AUTHINFO 属性 **CHCKLOCL** (检查本地连接)。 要控制是否检查凭证以查找客户机应用程序所建立的连接，请使用 AUTHINFO 属性 **CHCKCLNT** (检查客户机连接)。

CHCKLOCL 接受 NONE 和 OPTIONAL 的值， **CHCKCLNT** 允许为认证需求配置 NONE 的值:

NONE
不检查应用程序提供的认证凭证。

可选
确保应用程序提供的任何凭证都有效。 但是，应用程序不需要提供认证凭证。 例如，在迁移期间，此选项可能很有用。

如果您:

- 提供用户名和密码，并对其进行认证。
- 不提供用户名和密码，允许连接。
- 请提供用户名，但不提供您收到错误的密码。

要点: 如果您还希望在通道认证 (CHLAUTH) 规则中设置更严格的选项，那么可以设置 OPTIONAL 最小值。

如果选择 **无** 并且客户机连接与 CHLAUTH 记录匹配，并且 **CHCKCLNT** 设置为 **必需** (或者在除 z/OS 以外的平台上为 REQDADM)，那么连接将失败。 在 Multiplatforms 版上接收到消息 AMQ9793，在 z/OS 上接收到消息 CSQX793E。

有关使用通道认证规则为某些客户机连接设置更严格的 **CHCKCLNT** 选项的更多信息，请参阅 第 63 页的『配置详细程度』。

Required
要求所有应用程序都提供有效的凭证。 另请参阅以下注释。

REQDADM
特权用户必须提供有效凭证，但非特权用户被视为具有 OPTIONAL 设置。 另请参阅以下注释。

z/OS (在 z/OS 系统上不允许此设置。)

注:

将 **CHCKLOCL** 设置为 REQUIRED 或 REQDADM 意味着您无法使用 **runmqsc** (错误 AMQ8135: 未授权) 本地管理队列管理器，除非用户指定 **-u** 参数以在 **runmqsc** 命令中指定用户标识。 在设置了该参数的情况下，**runmqsc** 会在控制台上提示输入用户的密码。

同样，在本地系统上运行 IBM MQ Explorer 的用户在尝试连接到队列管理器时将看到错误 AMQ4036。 要指定用户标识和密码，请右键单击本地队列管理器对象，然后选择 **连接详细信息 > 属性 ...** 从菜单中获取。 在 **用户标识** 部分中，输入要使用的用户标识和密码，然后单击 **确定**。

类似注意事项适用于与 **CHCKCLNT** 的远程连接。

对于从 IBM MQ 8.0 之前的版本迁移的队列管理器，队列管理器 **CONNAUTH** 属性为空白，但对于新创建的队列管理器，此属性设置为 *SYSTEM.DEFAULT.AUTHINFO.IDPWOS* 。 缺省情况下，此缺省 **AUTHINFO** 定义将 **CHCKCLNT** 设置为 REQDADM 。

因此，使用特权用户标识进行连接的任何现有客户机都必须提供有效凭证。

警告: 客户机应用程序的 MQCSP 结构中的凭证有时以纯文本形式通过网络发送。 要确保客户机凭证受保护，请参阅 第 27 页的『MQCSP 密码保护』。

配置详细程度

AUTHINFO 对象的 **CHCKLOCL** 和 **CHCKCLNT** 属性设置与队列管理器的所有连接的认证需求。 除这些属性外，通道认证 (CHLAUTH) 规则上的 **CHCKCLNT** 属性允许针对与 CHLAUTH 规则匹配的特定客户机连接设置更严格的认证需求。

您可以将总体 **CHCKCLNT** 值设置为 OPTIONAL(例如，在 AUTHINFO 对象上)，然后通过在 CHLAUTH 规则上将 **CHCKCLNT** 设置为 REQUIRED 或 REQDADM，将其升级为更严格的特定通道。 缺省情况下， CHLAUTH 规则是使用 **CHCKCLNT(ASQMGR)** 定义的，因此不必使用此粒度。 例如，这些 MQSC 命令定义一个覆盖 AUTHINFO 对象的 **CHCKCLNT** 属性的 CHLAUTH 规则，以及一个不执行以下操作的 CHLAUTH 规则:

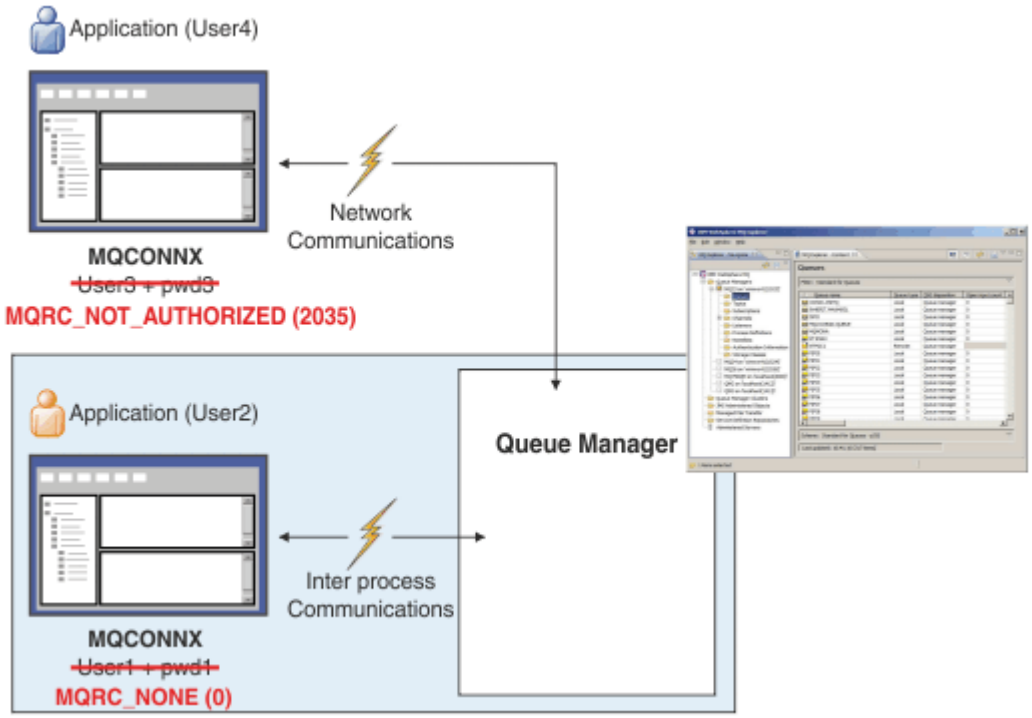
```
DEFINE AUTHINFO(USE.PW) AUTHTYPE(xxxxxx) +
CHCKCLNT(OPTIONAL)

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

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

有关 CHLAUTH 规则的更多信息，请参阅 第 43 页的『通道认证记录』。

错误通知



在以下情况下会记录错误:

- 当需要认证凭证时，应用程序不会提供这些凭证。
- 应用程序提供了无效的认证凭证。 即使配置声明应用程序提供凭证是可选的，也会将此情境视为错误。

注: 当 **CHCKLOCL** 或 **CHCKCLNT** 设置为 NONE 时，不会检测到应用程序提供的无效凭证。

在将错误返回到应用程序之前，将在 **FAILDLAY** 属性指定的秒数内保留失败的认证。 此延迟为重复尝试连接的应用程序提供了一些保护。

该错误以多种方式记录:

应用程序
MQRC_NOT_AUTHORIZED (2035) 原因码将返回到应用程序。

管理员
IBM MQ 管理员会看到错误日志中报告的事件。 错误消息显示已拒绝连接，因为凭证无效，而不是因为 (例如) 用户没有连接权限。

监测工具
如果开启权限事件，那么还可以通过 SYSTEM.ADMIN.QMGR.EVENT 队列上的事件消息来通知监视工具失败。 要开启权限事件，请发出以下 MQSC 命令:

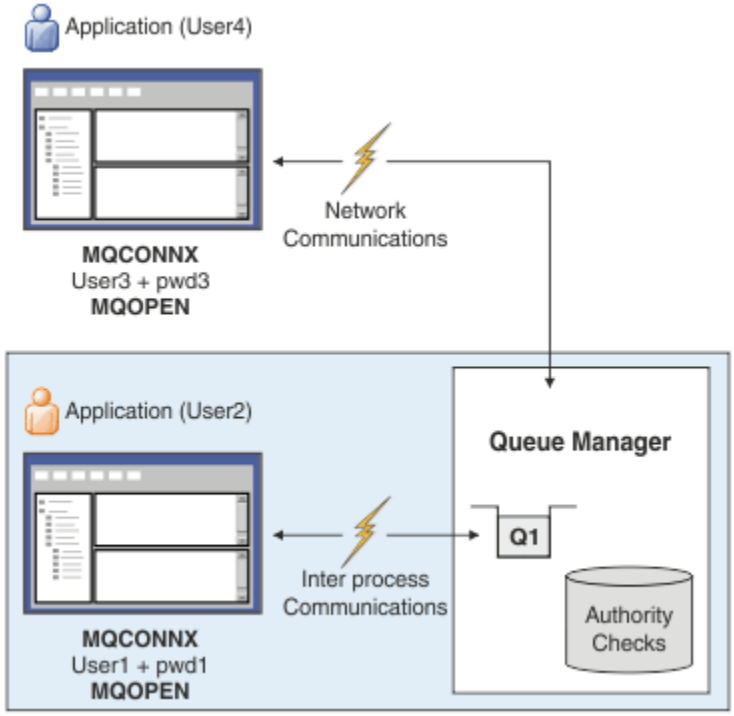
```
ALTER QMGR AUTHOREV(ENABLED)
```

此 "未授权" 事件是类型 1 连接事件，并提供与其他类型 1 事件相同的字段，以及提供的额外字段 MQCSP 用户标识。 如果应用程序提供了密码，那么该密码不会包含在事件消息中。 这意味着事件消息中有两个用户标识:

- 运行应用程序的用户标识。
- 应用程序提供的凭证中的用户标识。

有关此事件消息的更多信息，请参阅 未授权 (类型 1)。

采用用户进行授权



您可以将队列管理器配置为采用应用程序提供的凭证作为连接的上下文。 采用凭证意味着认证凭证中提供的用户标识用于授权检查，显示在管理屏幕上，并显示在消息中。 AUTHINFO 对象上的 **ADOPTCTX** 属性控制是否采用凭证作为应用程序的上下文。 例如，以下 MQSC 命令定义用于连接认证的名为 USE.PWD 的 AUTHINFO 对象，并将 **ADOPTCTX** 属性设置为 YES:

```
DEFINE AUTHINFO(USE.PWD) +
AUTHTYPE(xxxxxx) +
CHCKLOCL(OPTIONAL) +
```

```
CHCKCLNT(REQUIRED) +
ADOPTCTX(YES)

ALTER QMGR CONNAUTH(USE.PWD)
```

可以为 **ADOPTCTX** 属性指定以下值:

ADOPTCTX (是)
在连接持续时间内，将采用应用程序提供的凭证作为应用程序上下文。 将使用已认证的凭证中的用户标识对应用程序进行所有授权检查。

注意: 使用 **ADOPTCTX(YES)** 和本地操作系统用户标识时，必须确保所采用的用户标识满足 IBM MQ 中用户标识的需求。 有关更多信息，请参阅第 75 页的『用户标识』。

ADOPTCTX (NO)
应用程序提供的凭证仅用于连接时的认证。 用于运行应用程序的用户标识将继续用于将来的授权检查。 如果您计划使用其他机制 (例如通道认证记录) 来分配 消息通道代理程序用户标识 (MCAUSER)，那么您可能会发现此选项在迁移时很有用。

与通道认证交互

通道认证规则可用于根据从客户机接收到的用户标识来更改用作应用程序连接上下文的用户标识。 有关使用通道认证规则来更改与连接关联的用户标识的示例，请参阅第 340 页的『将客户机用户标识映射到 MCAUSER 用户标识』。

处理连接认证和通道认证规则的顺序是确定 IBM MQ 客户机应用程序连接的安全上下文的重要因素。 qm.ini 文件的 **channels** 节中的 **ChlauthEarlyAdopt** 参数控制队列管理器采用来自应用程序提供的凭证的上下文的顺序，并应用通道认证规则。 有关 **ChlauthEarlyAdopt** 的更多信息，请参阅 通道节的属性。

注意: 在认证信息对象上使用 **ADOPTCTX(YES)** 参数时，仅当 **ChlauthEarlyAdopt** 参数设置为 Y 时，才能通过通道认证规则来更改从应用程序提供的凭证采用的上下文。

有关连接认证和通道认证的交互以及客户机应用程序连接到队列管理器时执行检查的顺序的更多信息，请参阅第 48 页的『CHLAUTH 和 CONNAUTH 的交互』。

相关概念
第 60 页的『连接认证』
连接认证允许应用程序在连接到队列管理器时提供认证凭证。 队列管理器将验证凭证。 还可以采用凭证中提供的用户标识，以便在应用程序访问的资源的授权检查中使用。

第 65 页的『连接认证: 应用程序更改』

第 66 页的『连接认证: 用户存储库』
对于每个队列管理器，您可以选择不同类型的认证信息对象来认证用户标识和密码。

连接认证: 应用程序更改

调用 MQCONNX 时，使用消息队列接口 (MQI) 的应用程序可以在连接安全性参数 (MQCSP) 结构中提供用户标识和密码。 在其他应用程序编程接口中， MQCSP 结构通常由 IBM MQ 库代表应用程序构造。

V 9.4.0 从 IBM MQ 9.3.4 开始，连接到在 AIX 或 Linux 系统上运行的队列管理器的客户机应用程序还可以发送 MQCSP 结构中的认证令牌作为替代标识方法。

将用户标识和密码或认证令牌 传递给队列管理器随附的 对象权限管理器 (OAM) 或 z/OS 系统上随队列管理器随附的授权服务组件进行检查。 您不必编写自己的定制接口。

如果应用程序正在作为客户机，用户标识和密码或认证令牌运行，那么还会将 传递到客户机端和服务器端安全出口以进行处理。 它们还可用于设置通道实例的 消息通道代理程序用户标识 (MCAUSER) 属性 。

警告: 客户机应用程序的 MQCSP 结构中的凭证有时以纯文本形式通过网络发送。 要确保客户机应用程序凭证受保护，请参阅第 27 页的『MQCSP 密码保护』。

通过使用 XAOPEN 字符串来提供用户标识和密码，可以避免必须更改应用程序代码。

注:

从 IBM WebSphere MQ 6.0 开始，安全出口允许设置 MQCSP。 因此，此级别或更高级别的客户机不必升级。

但是，在 IBM MQ IBM MQ 8.0 之前的版本中， MQCSP 对应用程序提供的用户标识和密码没有任何限制。 将这些值与 IBM MQ 提供的功能部件配合使用时，存在适用于使用这些功能部件的限制，但如果仅将这些限制传递到您自己的出口，那么这些限制不适用。

相关概念

第 60 页的『连接认证』
连接认证允许应用程序在连接到队列管理器时提供认证凭证。 队列管理器将验证凭证。 还可以采用凭证中提供的用户标识，以便在应用程序访问的资源的授权检查中使用。

第 61 页的『连接认证: 配置』
可以配置队列管理器以认证应用程序在连接时提供的凭证。

第 66 页的『连接认证: 用户存储库』
对于每个队列管理器，您可以选择不同类型的认证信息对象来认证用户标识和密码。

连接认证: 用户存储库

对于每个队列管理器，您可以选择不同类型的认证信息对象来认证用户标识和密码。

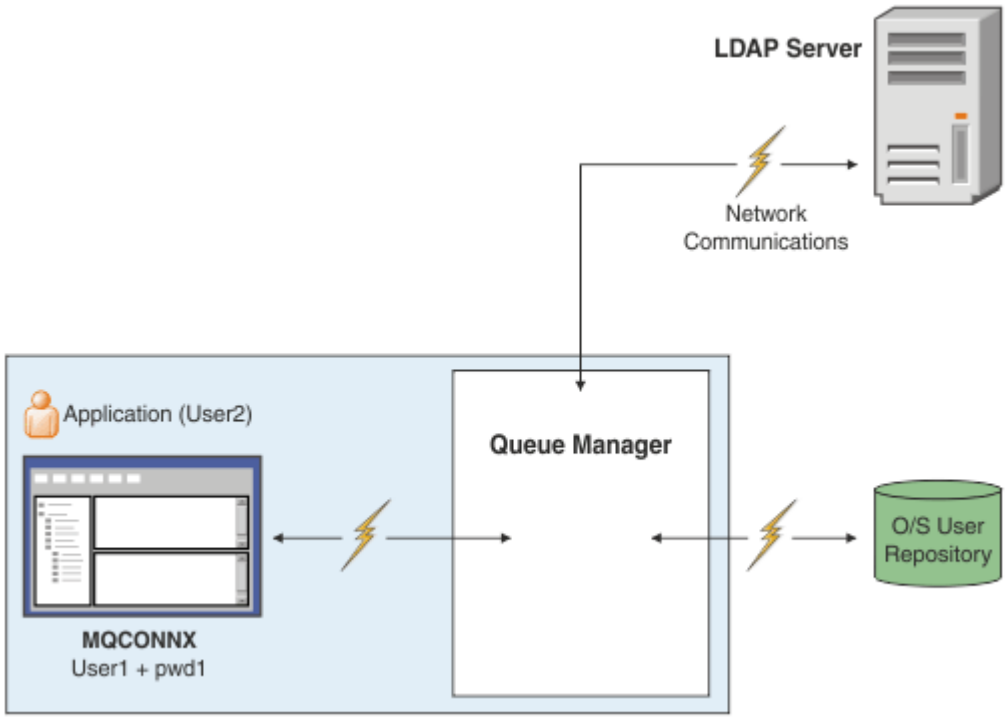


图 7: 认证信息对象的类型

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

有两种类型的认证信息对象，如图中所示:

- IDPWOS 用于指示队列管理器使用本地操作系统来认证用户标识和密码。 如果选择使用本地操作系统，那么需要设置公共属性，如上述主题中所述。
- IDPWLDAP 用于指示队列管理器使用 LDAP 服务器来认证用户标识和密码。 如果选择使用 LDAP 服务器，那么本主题中提供了更多信息。

通过在队列管理器的 **CONNAUTH** 属性中命名相应的对象，只能为每个要使用的队列管理器选择一种类型的认证信息对象。

使用 LDAP 服务器进行认证。

将 **CONNAME** 字段设置为队列管理器的 LDAP 服务器的地址。 您可以在以逗号分隔的列表中为 LDAP 服务器提供更多地址，如果 LDAP 服务器不提供此工具本身，那么这有助于实现冗余。

在 **LDAPUSER** 和 **LDAPPWD** 字段中设置必需的 LDAP 服务器标识和密码，以便队列管理器可以访问 LDAP 服务器并查找有关用户记录的信息。

与 LDAP 服务器的安全连接

与通道不同，没有 **SSLCIPH** 参数可开启使用 TLS 与 LDAP 服务器进行通信。 在这种情况下， IBM MQ 充当 LDAP 服务器的客户机，因此大部分配置是在 LDAP 服务器上完成的。 IBM MQ 中的一些现有参数用于配置该连接的工作方式。

设置 **SECCOMM** 字段以控制与 LDAP 服务器的连接是否使用 TLS。

除此属性外，队列管理器属性 **SSLFIPS** 和 **SUITEB** 还会限制所选的密码规范集。 用于向 LDAP 服务器标识队列管理器的证书是队列管理器证书 (`ibmwebspheremq` *`qmgr-name`* 或 **CERTLABL** 属性的值)。 请参阅数字证书标签 以获取详细信息。

LDAP 用户存储库

使用 LDAP 用户存储库时，要在队列管理器上执行更多配置，而不仅仅是告知队列管理器在何处查找 LDAP 服务器。

LDAP 服务器中定义的用户标识具有唯一标识它们的分层结构。 因此，应用程序可以连接到队列管理器并将其用户标识显示为标准分层用户标识。

但是，要简化应用程序必须提供的信息，可以将队列管理器配置为假定层次结构的第一部分对于所有标识是公共的，并在应用程序提供的简短标识之前自动添加此标识。 然后，队列管理器可以向 LDAP 服务器提供完整标识。

将 BASEDNU 设置为 LDAP 搜索在 LDAP 层次结构中查找标识的初始点。 设置 BASEDNU 时，必须确保在 LDAP 层次结构中搜索标识时仅返回一个结果。

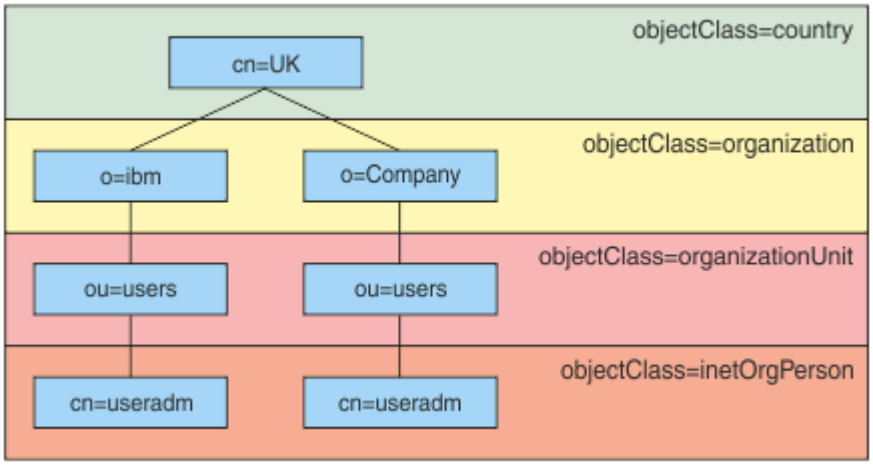


图 *8:* 示例 *LDAP* 层次结构

例如，在 第 67 页的图 8 BASEDNU 中，可以设置为 "ou=users， o=ibm， c = UK" 或 "， o=ibm， c = UK"。 但是，由于 "o = ibm" 分支和 "o=Company" 分支中都存在包含 "cn = useradm" 的专有名称，因此不能将 BASEDNU 设置为 "c = UK"。 出于性能和安全原因，请使用 LDAP 层次结构中的最高点，您可以从中引用所需的所有用户标识。 在此示例中，即 "ou=users， o=ibm， c = UK"。

您的应用程序可以向队列管理器提交用户标识，而不提供 LDAP 属性名称，例如 CN= 。 如果将 USRFIELD 设置为 LDAP 属性名，那么会将此值添加为来自应用程序的用户标识的前缀。 当您从操作系统用户标识移至 LDAP 用户标识时，这可能是一种有用的迁移辅助，因为在这两种情况下，应用程序都可以显示相同的字符串，并且可以避免更改应用程序。

因此，提供给 LDAP 服务器的完整用户标识如下所示:

```
USRFIELD = ID_from_application BASEDNU
```

相关概念

第 60 页的『连接认证』
连接认证允许应用程序在连接到队列管理器时提供认证凭证。 队列管理器将验证凭证。 还可以采用凭证中提供的用户标识，以便在应用程序访问的资源的授权检查中使用。

第 61 页的『连接认证: 配置』
可以配置队列管理器以认证应用程序在连接时提供的凭证。

第 65 页的『连接认证: 应用程序更改』

用于插入用户标识和密码的客户机端安全出口 (*mqccred*)

如果您有任何客户机应用程序需要发送用户标识或密码，但您尚无法更改源，那么可以使用 IBM MQ 8.0 随附的称为 **mqccred** 的安全出口。 **mqccred** 从 `.ini` 文件中代表客户机应用程序提供用户标识和密码。 此用户标识和密码将发送到队列管理器，如果配置为这样做，那么队列管理器将对这些用户标识和密码进行认证。

概述

mqccred 是在与客户机应用程序相同的机器上运行的安全出口。 它允许代表客户机应用程序提供用户标识和密码信息，其中该信息不是由应用程序本身提供的。 用户标识和密码信息以称为 连接安全参数 (MQCSP) 的结构提供，如果配置了 连接认证 ，那么将由队列管理器进行认证。

将从客户端机器上的 `.ini` 文件中检索用户标识和密码信息。 通过使用 **runmqccred** 命令进行模糊处理来保护文件中的密码，并确保设置对 `.ini` 文件的文件许可权，以便只有运行客户机应用程序的用户标识 (因此出口) 能够读取该文件。

位置

mqccred 已安装:

Windows 平台
在 *installation_directory*\Tools\c\Samples\mqccred\ 目录中

AIX and Linux 平台
在 *installation_directory*/samp/mqccred 目录中

注意: 出口:

1. 纯粹充当安全通道出口，并且需要是通道上定义的唯一此类出口。
2. 通常通过 "客户机通道定义表" (CCDT) 进行命名，但 Java 客户机可以直接具有 JNDI 对象中提及的出口，或者可以为手动构造 MQCD 结构的应用程序配置该出口。
3. 必须将 **mqccred** 和 **mqccred_r** 程序复制到 `var/mqm/exits` 目录。

   例如，在 64 位 AIX 或 Linux 系统上，发出以下命令:

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

   请参阅 有关如何测试 mqccred 的逐步示例 以获取更多信息。
4. 能够在先前版本的 IBM MQ 上运行，远至 IBM WebSphere MQ 7.0.1。

设置用户标识和密码

.ini 文件包含每个队列管理器的节，以及未指定的队列管理器的全局设置。 每个节都包含队列管理器的名称，用户标识以及纯文本或模糊化密码。

您必须使用所需的任何编辑器手动编辑 .ini 文件，并将纯文本密码属性添加到节中。 运行提供的 **runmqccred** 程序，该程序采用 .ini 文件，并将 **Password** 属性替换为 **OPW** 属性，这是一种模糊化形式的密码。

请参阅 runmqccred 以获取命令及其参数的描述。

mqccred.ini 文件包含您的用户标识和密码信息。

在出口所在的目录中提供了模板 .ini 文件，以提供企业的起始点。

缺省情况下，将在 $HOME/.mqs/mqccred.ini 中查找此文件。 如果要在其他位置找到它，那么可以使用环境变量 *MQCCRED* 来指向它:

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

如果使用 *MQCCRED*，那么该变量必须包含配置文件的全名，包括任何 .ini 文件类型。 由于此文件包含密码 (即使已加密) ，因此期望您使用操作系统特权来保护此文件，以确保未经授权的人员无法读取此文件。 如果您没有正确的文件许可权，那么出口将不会成功运行。

如果应用程序已提供 MQCSP 结构，那么出口通常遵循此结构，并且不会从 .ini 文件插入任何信息。 但是，您可以使用节中的 **Force** 属性来覆盖此属性。

将 **Force** 设置为值 *TRUE* 将除去应用程序提供的用户标识和密码，并将这些用户标识和密码替换为 ini 文件版本。

您还可以在文件的全局部分中设置 **Force** 属性，以设置该文件的缺省值。

Force 的缺省值为 *FALSE*。

您可以为所有队列管理器或每个单独的队列管理器提供用户标识和密码。 以下是 mqccred.ini 文件的示例:

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

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

Force=TRUE

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

注意:

1. 各个队列管理器定义优先于全局设置。
2. 属性不区分大小写。

约束

使用此出口时，运行应用程序的人员的本地用户标识不会从客户机流向服务器。 唯一可用的身份信息来自 ini 文件内容。

因此，您必须将队列管理器配置为使用 **ADOPTCTX(YES)**，或者通过其中一种可用机制 (例如 第 43 页的『通道认证记录』) 将入站连接请求映射到相应的用户标识。

要点: 如果添加新密码或更新旧密码，那么 **runmqccred** 命令仅处理任何纯文本密码，而使模糊处理的密码保持不变。

调试

如果启用了标准 IBM MQ 跟踪，那么出口将写入该跟踪。

为了协助调试配置问题，出口还可以直接写入 stdout。

无通道安全性出口数据 (**SCYDATA**) 通道通常需要配置。 但是，您可以指定:

错误
仅打印信息异常错误情况，例如无法找到配置文件。

调试
显示这些错误情况以及一些其他跟踪语句。

NOCHECKS
绕过对文件许可权的约束，以及 `.ini` 文件不应包含任何不受保护的密码的进一步约束。

您可以按任意顺序将其中一个或多个元素放入 **SCYDATA** 字段中 (以逗号分隔)。 例如，`SCYDATA=(NOCHECKS,DEBUG)`。

请注意，这些项区分大小写，并且必须以大写形式输入。

使用 mqccred

设置文件后，可以通过更新客户机连接通道定义以包含 `SCYEXIT('mqccred(ChlExit)')` 属性来调用通道出口:

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

相关参考
SCYDATA
SCYEXIT
运行 mqccred

使用 *Java* 客户机进行连接认证

连接认证是 IBM MQ 中的一项功能，使您能够配置队列管理器，以便队列管理器可以使用提供的用户标识和密码对应用程序进行认证。 当应用程序是使用客户机传输的 Java 应用程序时，可以在兼容性方式或 MQCSP 认证方式下运行连接认证。

要认证的用户标识和密码由应用程序使用下列其中一种方法指定:

- 在 IBM MQ classes for Java 应用程序中，在 `MQEnvironment` 类中，或者在传递到 `com.ibm.mq.MQQueueManager` 构造函数的属性 Hashtable 中。
- 在 IBM MQ classes for JMS 应用程序中，作为 `createConnection(String username, String Password)` 或 `createContext(String username, String password)` 方法的自变量。

MQCSP 认证方式

在此方式下，将应用程序运行所使用的客户端用户标识以及要认证的用户标识和密码发送到队列管理器。 IBM MQ classes for Java 和 IBM MQ classes for JMS 将要认证的用户标识和密码发送到 MQCSP 结构中的队列管理器。

用户标识和密码可用于 MQCSP 结构中的服务器连接安全出口。 可在通道的 MQCXP 结构的 **SecurityParms** 字段中找到 MQCSP 结构地址。

MQCSP 认证方式具有以下优点:

- 要认证的用户标识的最大长度为 1024 个字符。
- 用于认证的密码的最大长度为 256 个字符。
- 当使用 ADOPTCTX (NO) 配置了用于在队列管理器上控制连接认证的认证信息对象时，可以使用应用程序运行所使用的客户机端用户标识来执行授权检查以访问 IBM MQ 资源。

兼容性方式

在 IBM MQ 8.0 之前， Java 客户机可以通过客户机连接通道将用户标识和密码发送到服务器连接通道，并将其提供给 MQCD 结构的 **RemoteUserIdentifier** 和 **RemotePassword** 字段中的安全出口。 在兼容性方式下，将保留此行为。

您可以将此方式与连接认证结合使用，并从先前用于执行同一作业的任何安全出口进行迁移。

此方式具有以下限制:

- 用户标识和密码的长度不得超过 12 个字符。 长度超过 12 个字符的用户标识将截断为 12 个字符。 这可能会导致连接失败，原因码为 MQRC_NOT_AUTHORIZED。
- 应用程序运行所使用的客户机端用户标识不会发送到队列管理器。 必须在用于控制队列管理器上的连接认证的认证信息对象上设置 ADOPTCTX (YES) ，或者使用其他方法 (例如，基于 TLS 证书的通道认证规则) 来设置通道 MCA 用户标识，检查该用户标识以获取使用 IBM MQ 资源的授权。

缺省认证方式

IBM MQ classes for Java 或 IBM MQ classes for JMS 客户机应用程序使用的缺省认证方式根据应用程序是否指定用户标识和密码而有所不同。

- 如果指定了用户标识和密码，那么缺省情况下将使用 MQCSP 认证。
- 如果指定了用户标识，但未指定密码，那么缺省情况下将使用兼容性方式。
- 如果未指定用户标识，那么将始终使用兼容性方式。

在指定用户标识的情况下，应用程序可以为每个单独的连接选择特定的认证方式，也可以在应用程序启动之前全局设置，如 第 71 页的『选择认证方式』中所述。

注: 在 IBM MQ 9.3.0 中更改为缺省认证方式可能会影响使用 IBM MQ classes for JMS 的应用程序。 将 IBM MQ classes for JMS 升级到 IBM MQ 9.3.0 后，缺省情况下先前使用兼容性方式的应用程序将改为使用 MQCSP 认证。 这可能导致先前成功连接到队列管理器的应用程序无法与包含原因码 2035 (MQRC_NOT_AUTHORIZED) 的 JMSException 连接。 如果发生此情况，请使用 第 71 页的『选择认证方式』中描述的其中一种方法来指定应用程序使用兼容性方式。

使用本地绑定连接到队列管理器的 Java 应用程序始终使用 MQCSP 认证方式。

选择认证方式

可以使用下列其中一种方法来指定 Java 客户机应用程序使用的认证方式，这些客户机应用程序在连接到队列管理器时指定用户标识。 这些方法按优先顺序降序列出。 如果未使用这些方法中的任何方法指定认证方式，那么将使用缺省认证方式。

注: 在 IBM MQ 9.3.0 中说明了如何使用这些方法来选择认证方式。 在某些情况下，当 IBM MQ classes for Java 或 IBM MQ classes for JMS 升级到 IBM MQ 9.3.0 时， Java 客户机应用程序使用的认证方式可能会更改。 这可能导致先前成功连接到队列管理器的应用程序无法与包含原因码 2035 (MQRC_NOT_AUTHORIZED) 的 JMSException 连接。 如果发生此情况，请使用下列其中一种方法来选择所需的认证方式。

- 通过在连接到队列管理器之前在应用程序中设置相应的属性，为每个单独的连接指定认证方式。
  - 使用 IBM MQ classes for Java 时，请在传递给 com.ibm.mq.MQQueueManager 构造函数的属性 Hashtable 中设置属性 *MQConstants.USE_MQCSP_AUTHENTICATION_PROPERTY* 。
  - 使用 IBM MQ classes for JMS 时，请设置属性 *JmsConstants*。在创建连接之前，请在相应的连接工厂上使用 *USER_AUTHENTICATION_MQCSP* 。

  将这些属性的值设置为下列其中一个值:

true
向队列管理器认证时使用 MQCSP 认证方式。

false
向队列管理器认证时使用兼容性方式。

- 通过在启动应用程序时设置 *com.ibm.mq.cfg.jmqi.useMQCSPauthentication* Java 系统属性，指定应用程序所建立的所有客户机连接的认证方式。 将该属性的值设置为下列其中一个值:

  Y
  向队列管理器认证时使用 MQCSP 认证方式。

  N
  向队列管理器认证时使用兼容性方式。

  例如，以下命令设置属性以选择兼容性方式并启动 Java 应用程序:

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

- 通过在启动应用程序的环境中设置 *com.ibm.mq.jmqi.useMQCSPauthentication* 环境变量，为在同一环境中启动的应用程序所建立的所有客户机连接指定认证方式。 将环境变量的值设置为下列其中一个值:

  Y
  向队列管理器认证时使用 MQCSP 认证方式。

  N
  向队列管理器认证时使用兼容性方式。

- 通过在客户机配置文件的 JMQI 节中指定 **useMQCSPauthentication** 属性，为所有使用特定 IBM MQ MQI client 客户机配置文件的应用程序指定认证方式。 将属性值设置为下列其中一个值:

  YES
  向队列管理器认证时使用 MQCSP 认证方式。

  否
  向队列管理器认证时使用兼容性方式。

  有关 **useMQCSPauthentication** 属性的更多信息，请参阅 客户机配置文件的 JMQI 节。

在 IBM MQ Explorer 中选择认证方式

IBM MQ Explorer 是 Java 应用程序，因此这两种方式 (兼容性方式和 MQCSP 认证方式) 也适用于它。

MQCSP 认证方式是缺省方式。

在提供了用户标识的面板上，有一个用于启用或禁用兼容性方式的复选框:

- 缺省情况下，未选中此复选框。 要使用兼容性方式，请选中此复选框。

相关概念
第 60 页的『连接认证』
连接认证允许应用程序在连接到队列管理器时提供认证凭证。 队列管理器将验证凭证。 还可以采用凭证中提供的用户标识，以便在应用程序访问的资源的授权检查中使用。

第 65 页的『连接认证: 应用程序更改』

第 66 页的『连接认证: 用户存储库』
对于每个队列管理器，您可以选择不同类型的认证信息对象来认证用户标识和密码。

IBM MQ 中的消息安全性

IBM MQ 基础结构中的消息安全性由 Advanced Message Security 提供。

Advanced Message Security (AMS) 扩展 IBM MQ 安全服务以在消息级别提供数据签名和加密。 扩展服务保证在最初将消息数据放入队列与检索消息数据之间未进行修改。 此外， AMS 还会验证消息数据的发送方是否有权将已签名的消息放在目标队列上。

相关概念
第 522 页的『Advanced Message Security』

Advanced Message Security (AMS) 是 IBM MQ 的组件，可为流经 IBM MQ 网络的敏感数据提供高级别保护，同时不会影响最终应用程序。

规划安全需求

此主题集合说明在 IBM MQ 环境中规划安全性时需要考虑的事项。

您可以将 IBM MQ 用于一系列平台上的各种应用程序。 每个应用程序的安全性需求可能不同。 对一些人来说，安全将是一个关键的考虑因素。

IBM MQ 提供了一系列链路级别安全服务，包括对传输层安全性 (TLS) 的支持。

在规划安装 IBM MQ 时，必须考虑安全性的某些方面:

- Multi 在 多平台上，如果忽略这些方面而不执行任何操作，那么无法使用 IBM MQ。
- z/OS 在 z/OS 上，忽略这些方面的影响是您的 IBM MQ 资源不受保护。 即，所有用户都可以访问和更改所有 IBM MQ 资源。

管理 IBM MQ 的权限

IBM MQ 管理员需要以下权限:

- 发出命令以管理 IBM MQ
- 使用 IBM MQ Explorer
- IBM i 使用 IBM i 管理面板和命令。
- z/OS 使用 z/OS 上的操作和控制面板
- z/OS 在 z/OS 上使用 IBM MQ 实用程序 CSQUTIL
- z/OS 访问 z/OS 上的队列管理器数据集

有关更多信息，请参阅:

- ALW 第 353 页的『在 AIX, Linux, and Windows 上管理 IBM MQ 的权限』
- IBM i 第 77 页的『在 IBM i 上管理 IBM MQ 的权限』
- z/OS 第 77 页的『Authority to administer IBM MQ on z/OS』

使用 IBM MQ 对象的权限

应用程序可以通过发出 MQI 调用来访问以下 IBM MQ 对象:

- 队列管理器
- 队列
- 进程
- 名称列表
- 主题

应用程序还可以使用可编程命令格式 (PCF) 命令来访问这些 IBM MQ 对象，以及访问通道和认证信息对象。这些对象可由 IBM MQ 保护，以便与应用程序关联的用户标识需要访问这些对象的权限。

有关更多信息，请参阅 第 79 页的『应用程序使用 IBM MQ 的授权』。

通道安全性

与消息通道代理程序 (MCA) 关联的用户标识需要访问各种 IBM MQ 资源的权限。 例如， MCA 必须能够连接到队列管理器。 如果是发送 MCA，那么必须能够打开通道的传输队列。 如果它是接收 MCA，那么必须能

够打开目标队列。 与需要管理通道，通道启动程序和侦听器的应用程序相关联的用户标识需要使用相关 PCF 命令的权限。 但是，大多数应用程序不需要此类访问权。

有关更多信息，请参阅 第 96 页的『通道授权』。

其他注意事项

仅当使用某些 IBM MQ 功能或基本产品扩展时，才需要考虑安全性的以下方面:

- 第 106 页的『队列管理器集群的安全性』
- 第 106 页的『IBM MQ 发布/预订的安全性』

规划标识和认证

决定要使用的用户标识，以及要应用认证控件的方式和级别。

您必须决定如何识别 IBM MQ 应用程序的用户，同时铭记不同的操作系统支持不同长度的用户标识。 您可以使用通道认证记录从一个用户标识映射到另一个用户标识，或者根据连接的某些属性指定用户标识。 使用 TLS 的 IBM MQ 通道将数字证书用作标识和认证机制。 每个数字证书都有一个主题专有名称，可以使用通道认证记录将其映射到特定身份。 此外，密钥存储库中的 CA 证书确定哪些数字证书可用于向 IBM MQ 进行认证。 有关更多信息，请参阅:

- 第 339 页的『将远程队列管理器映射到 MCAUSER 用户标识』
- 第 340 页的『将客户机用户标识映射到 MCAUSER 用户标识』
- 第 340 页的『将 SSL 或 TLS 专有名称映射到 MCAUSER 用户标识』
- 第 342 页的『将 IP 地址映射到 MCAUSER 用户标识』

规划客户机应用程序的认证

您可以在以下四个级别应用认证控件: 通信级别，安全出口，通道认证记录以及传递到安全出口的标识。

需要考虑四个级别的安全性。 该图显示连接到服务器的 IBM MQ MQI client 。 将在四个级别应用安全性，如以下文本中所述。 MCA 是消息通道代理程序。

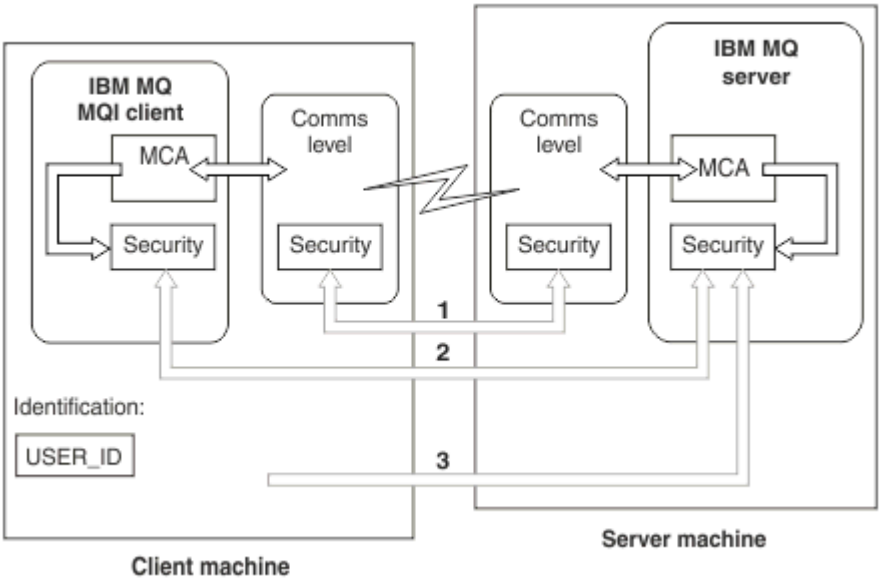


图 9: 客户机/服务器连接中的安全性

1. 通信级别

   请参阅箭头 1。 要在通信级别实现安全性，请使用 TLS。 有关更多信息，请参阅第 16 页的『加密安全性协议：TLS』。
2. 通道认证记录

See arrows 2 & 3. 可以在安全级别使用 IP 地址或 TLS 专有名称来控制认证。 还可以阻止用户标识，或者可以将已断言的用户标识映射到有效的用户标识。 第 43 页的『通道认证记录』中提供了完整的描述。

3. 连接认证

   请参阅箭头 3。 客户机发送用户标识和密码或认证令牌。 有关更多信息，请参阅 第 61 页的『连接认证: 配置』。

4. 通道安全出口

   请参阅箭头 2。 用于客户机到服务器通信的通道安全出口可以与用于服务器到服务器通信的通道安全出口以相同的方式工作。 可以编写独立于协议的出口对，以提供客户机和服务器的相互认证。 通道安全出口程序中提供了完整描述。

5. 传递到通道安全出口的标识

   请参阅箭头 3。 在客户机到服务器的通信中，通道安全出口不必作为一对操作。 可以省略 IBM MQ 客户端上的出口。 在这种情况下，用户标识放置在通道描述符 (MQCD) 中，如果需要，服务器端安全出口可以对其进行更改。

   IBM MQ MQI clients 还会发送额外的信息以帮助标识。

   - 传递到服务器的用户标识是客户机上当前已登录的用户标识。
   - 当前已登录用户的安全标识。

   服务器安全出口可以使用用户标识和安全标识 (如果可用) 的值来建立 IBM MQ MQI client 的身份。

从 IBM MQ 8.0 开始，可以发送 MQCSP 结构中包含的密码。

Linux V 9.4.0 AIX 从 IBM MQ 9.3.4 开始， IBM MQ MQI clients 连接到在 AIX 或 Linux 系统上运行的 IBM MQ 队列管理器也可以发送 MQCSP 结构中的认证令牌。

警告: 在某些情况下，客户机应用程序的 MQCSP 结构中的密码 或认证令牌 将以纯文本形式通过网络发送。 要确保相应保护客户机应用程序密码 和认证令牌 ，请参阅 第 27 页的『MQCSP 密码保护』。

用户标识

为客户机应用程序创建用户标识时，用户标识的长度不得超过允许的最大长度。 不得使用保留用户标识 UNKNOWN 和 NOBODY。 如果客户机连接到的服务器是 IBM MQ for Windows 服务器，那么必须对 at 符号 @ 的使用进行转义。 允许的用户标识长度取决于用于服务器的平台:

- Linux z/OS AIX 在 z/OS AIX and Linux 上，用户标识的最大长度为 12 个字符。
- IBM i 在 IBM i 上，用户标识的最大长度为 10 个字符。
- Windows 在 Windows 上，如果 IBM MQ MQI client 和 IBM MQ 服务器都在 Windows 上，并且服务器有权访问定义了客户机用户标识的域，那么用户标识的最大长度为 20 个字符。 但是，如果 IBM MQ 服务器不是 Windows 服务器，那么用户标识将截断为 12 个字符。
- 如果使用 MQCSP 结构来传递凭证，那么用户标识的最大长度为 1024 个字符。 MQCSP 结构用户标识不能用于规避 IBM MQ 用于授权的最大用户标识长度。 有关 MQCSP 结构的更多信息，请参阅 第 284 页的『使用 MQCSP 结构识别和认证用户』。

在 AIX and Linux 系统上，缺省情况是用户标识用于认证，组用于授权。 但是，您可以配置这些系统以针对用户标识进行授权。 有关更多信息，请参阅 第 310 页的『AIX and Linux 上基于 OAM 用户的许可权』。 Windows 系统可以同时使用用户标识进行认证和授权，并使用组进行授权。

如果创建服务帐户，而不关注组，并以不同方式授权所有用户标识，那么每个用户都可以访问其他每个用户的信息。

受限用户标识

用户标识 UNKNOWN 和组 NOBODY 对 IBM MQ 具有特殊含义。 在名为 UNKNOWN 的操作系统或名为 NOBODY 的组中创建用户标识可能会产生意外结果。

连接到 IBM MQ for Windows 服务器时的用户标识

Windows

如果客户机以包含 @ 字符的用户标识 (例如， abc@d) 运行，那么 IBM MQ for Windows 服务器不支持连接 IBM MQ MQI client 。 客户机上 MQCONN 调用的返回码为 MQRC_NOT_AUTHORIZED。

但是，您可以使用两个 @ 字符指定用户标识，例如 abc@@d。 使用 id@domain 格式是首选做法，以确保在正确的域中一致地解析用户标识; 因此 abc@@d@domain。

规划授权

规划将具有管理权限的用户，并规划如何授权应用程序用户适当使用 IBM MQ 对象，包括从 IBM MQ MQI client 连接的对象。

必须授予个人或应用程序访问权才能使用 IBM MQ。 他们需要哪些访问权取决于他们所承担的角色以及他们需要执行的任务。 IBM MQ 中的授权可以细分为两个主要类别:

- 执行管理操作的权限
- 应用程序使用 IBM MQ 的授权

这两个操作类都由同一组件控制，并且可以授予个人执行这两个操作类别的权限。

以下主题提供了有关您必须考虑的特定授权区域的更多信息:

管理 IBM MQ 的权限

IBM MQ 管理员需要权限才能执行各种功能。 此权限在不同平台上以不同方式获得。

IBM MQ 管理员需要以下权限:

- 发出命令以管理 IBM MQ。
- Windows Linux 使用 IBM MQ Explorer。
- z/OS 使用 z/OS 上的操作和控制面板。
- z/OS 在 z/OS 上使用 IBM MQ 实用程序 CSQUTIL。
- z/OS 访问 z/OS 上的队列管理器数据集。

有关更多信息，请参阅适用于您的操作系统的主题。

ALW 在 *AIX, Linux, and Windows* 系统上管理 *IBM MQ* 的权限

IBM MQ 管理员是 mqm 组的成员。 此组有权访问所有 IBM MQ 资源，并且可以发出 IBM MQ 控制命令。 管理员可针对其他用户授予特定权限。

要成为 AIX, Linux, and Windows 系统上的 IBM MQ 管理员，用户必须是 *mqm* 组的成员。 此组是在安装 IBM MQ 时自动创建的。 要允许用户发出控制命令，必须将其添加到 mqm 组。 这包括 AIX and Linux 上的 root 用户。

非 mqm 组成员的用户可以被授予管理特权，但他们无法发出 IBM MQ 控制命令，并且他们被授权仅执行他们已被授予访问权的命令。

此外，在 Windows 系统上， SYSTEM 和管理员帐户具有对 IBM MQ 资源的完全访问权。

mqm 组的所有成员都有权访问系统上的所有 IBM MQ 资源，包括能够管理在系统上运行的任何队列管理器。 只能通过从 mqm 组中除去用户来撤销此访问权。 在 Windows 系统上， Administrators 组的成员还可以访问所有 IBM MQ 资源。

管理员可以使用控制命令 **runmqsc** 来发出 IBM MQ Script (MQSC) 命令。 以间接方式使用 **runmqsc** 将 MQSC 命令发送到远程队列管理器时，每个 MQSC 命令都封装在 Escape PCF 命令中。 管理员必须具有要由远程队列管理器处理的 MQSC 命令的必需权限。

IBM MQ Explorer 发出 PCF 命令以执行管理任务。 管理员无需其他权限即可使用 IBM MQ Explorer 来管理本地系统上的队列管理器。 当 IBM MQ Explorer 用于管理另一个系统上的队列管理器时，管理员必须具有远程队列管理器要处理的 PCF 命令所需的权限。

有关处理 PCF 和 MQSC 命令时执行的权限检查的更多信息，请参阅以下主题:

- 对于在队列管理器，队列，通道，进程，名称列表和认证信息对象上运行的命令，请参阅 第 79 页的『应用程序使用 IBM MQ 的授权』。
- 对于在通道，通道启动程序，侦听器和集群上运行的命令，请参阅 通道安全性。
- z/OS 对于由 IBM MQ for z/OS 上的命令服务器处理的 MQSC 命令，请参阅 第 78 页的『Command security and command resource security on z/OS』。

有关管理 IBM MQ for AIX, Linux, and Windows 系统所需的权限的更多信息，请参阅相关信息。

IBM i 在 *IBM i* 上管理 *IBM MQ* 的权限

要成为 IBM i 上的 IBM MQ 管理员，您必须是 *QMQMADM* 组的成员。 此组具有与 AIX, Linux, and Windows 系统上 mqm 组的属性相似的属性。 尤其是在安装 IBM MQ for IBM i 时创建 QMQMADM 组，并且 QMQMADM 组的成员有权访问系统上的所有 IBM MQ 资源。 如果您具有 *ALLOBJ 权限，那么您还可以访问所有 IBM MQ 资源。

管理员可以使用 CL 命令来管理 IBM MQ。其中一个命令是 GRTMQMAUT ，用于向其他用户授予权限。 另一个命令 STRMQMMQSC 允许管理员向本地队列管理器发出 MQSC 命令。

IBM MQ for IBM i 提供了两组 CL 命令:

第 1 组
要在此类别中发出命令，用户必须是 QMQMADM 组的成员或具有 *ALLOBJ 权限。 例如， GRTMQMAUT 和 STRMQMMQSC 属于此类别。

组 2
要在此类别中发出命令，用户不需要是 QMQMADM 组的成员或具有 *ALLOBJ 权限。 相反，需要两个级别的权限:

- 用户需要 IBM i 权限才能使用该命令。 此权限是通过使用 GRTOBJAUT 命令授予的。
- 用户需要 IBM MQ 权限才能访问与该命令关联的任何 IBM MQ 对象。 此权限是通过使用 GRTMQMAUT 命令授予的。

以下示例显示此组中的命令:

- CRTMQMQ ， 创建 MQM 队列
- CHGMQMPRC ， 更改 MQM 进程
- DLTMQMNL ， 删除 MQM 名称列表
- DSPMQMAUTI ， 显示 MQM 认证信息
- CRTMQMCHL ， 创建 MQM 通道

有关这组命令的更多信息，请参阅 第 79 页的『应用程序使用 IBM MQ 的授权』。

有关组 1 和组 2 命令的完整列表，请参阅 第 135 页的『IBM i 上 IBM MQ 对象的访问权限』

有关在 IBM i 上管理 IBM MQ 所需的权限的更多信息，请参阅 管理 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.

应用程序使用 IBM MQ 的授权

当应用程序访问对象时，与应用程序关联的用户标识需要相应的权限。

应用程序可以通过发出 MQI 调用来访问以下 IBM MQ 对象:

- 队列管理器
- 队列
- 进程

- 名称列表
- 主题

应用程序还可以使用 PCF 命令来管理 IBM MQ 对象。 处理 PCF 命令时，它使用放置 PCF 消息的用户标识的权限上下文。

在此上下文中，应用程序包括用户和供应商编写的应用程序以及随 IBM MQ for z/OS 提供的应用程序。

z/OS IBM MQ for z/OS 随附的应用程序包括:

- 操作和控制面板
- IBM MQ 实用程序 CSQUTIL
- 死信队列处理程序实用程序 CSQUDLQH

使用 IBM MQ classes for Java， IBM MQ classes for JMS， IBM MQ classes for .NET 或 Message Service Client for C/C++ 和 .NET 的应用程序间接使用 MQI。

MCA 还会发出 MQI 调用，并且与 MCA 关联的用户标识需要权限才能访问这些 IBM MQ 对象。 有关这些用户标识及其所需的权限的更多信息，请参阅 第 96 页的『通道授权』。

z/OS 在 z/OS 上，应用程序还可以使用 MQSC 命令来访问这些 IBM MQ 对象，但命令安全性和命令资源安全性在这些情况下提供权限检查。 z/OS 有关更多信息，请参阅 第 78 页的『Command security and command resource security on z/OS』 和 第 78 页的『MQSC commands and the system command input queue on z/OS』。

IBM i 在 IBM i 上，在组 2 中发出 CL 命令的用户可能需要访问与该命令关联的 IBM MQ 对象的权限。 有关更多信息，请参阅第 80 页的『执行权限检查时』。

执行权限检查时

当应用程序尝试访问队列管理器，队列，进程或名称列表时，将执行权限检查。

在 IBM i 上，当用户在组 2 中发出访问任何这些 IBM MQ 对象的 CL 命令时，也可以执行权限检查。 在以下情况下执行检查:

当应用程序使用 MQCONN 或 MQCONNX 调用连接到队列管理器时

队列管理器向操作系统询问与应用程序关联的用户标识。 然后，队列管理器将检查用户标识是否有权连接到该用户标识，并保留该用户标识以供将来检查。

用户不必登录到 IBM MQ。 IBM MQ 假定用户已登录到底层操作系统并由其进行认证。

当应用程序使用 MQOPEN 或 MQPUT1 调用打开 IBM MQ 对象时

所有权限检查都在打开对象时执行，而不是在稍后访问对象时执行。 例如，当应用程序打开队列时，将执行权限检查。 当应用程序将消息放入队列或从队列中获取消息时，不会执行这些操作。

当应用程序打开对象时，它指定它需要对该对象执行的操作类型。 例如，应用程序可能会打开队列以浏览其上的消息，从中获取消息，但不会将消息放在其上。 对于每种类型的操作，队列管理器会检查与应用程序关联的用户标识是否有权执行该操作。

当应用程序打开队列时，将对对象描述符的 `ObjectName` 字段中指定的对象执行权限检查。 `ObjectName` 字段用于 `MQOPEN` 或 `MQPUT1` 调用。 如果对象是别名队列或远程队列定义，那么将对对象本身执行权限检查。 它们不会在别名队列或远程队列定义所解析的队列上执行。 这意味着用户不需要访问它的许可权。 将创建队列的权限限制为特权用户。 如果不执行此操作，那么用户可以仅通过创建别名来绕过正常访问控制。

应用程序可以显式引用远程队列。 它将对象描述符中的 `ObjectName` 和 `ObjectQMgrName` 字段设置为远程队列和远程队列管理器的名称。 将对与远程队列管理器同名的传输队列执行权限检查:

- z/OS 在 z/OS 上，将对与远程队列管理器名称匹配的 RACF 队列概要文件进行检查，并且无论是否在本地定义了此传输队列，都将执行此检查。
- Multi 在 多平台上，如果正在使用集群，那么将针对与远程队列管理器名称匹配的 RQMNAME 概要文件进行检查。

应用程序可以通过将对象描述符中的 ObjectName 字段设置为集群队列的名称来显式引用集群队列。 将对集群传输队列 SYSTEM.CLUSTER.TRANSMIT.QUEUE 执行权限检查。

对动态队列的权限基于从中派生该队列的模型队列，但不一定相同; 请参阅注释 1。

队列管理器用于权限检查的用户标识是从操作系统获取的。 用户标识是在应用程序连接到队列管理器时获取的。 适当授权的应用程序可以发出 MQOPEN 调用，指定备用用户标识; 然后对备用用户标识进行访问控制检查。 使用备用用户标识不会更改与应用程序关联的用户标识，仅更改用于访问控制检查的用户标识。

当应用程序使用 MQSUB 调用预订主题时

当应用程序预订主题时，它指定需要执行的操作类型。 它正在创建预订，更改现有预订或在不更改现有预订的情况下恢复现有预订。 对于每种类型的操作，队列管理器会检查与应用程序关联的用户标识是否有权执行该操作。

当应用程序预订主题时，将对在主题树中找到的主题对象执行权限检查。 主题对象位于或高于应用程序预订的主题树中的点。 权限检查可能涉及对多个主题对象的检查。 队列管理器用于权限检查的用户标识是从操作系统获取的。 用户标识是在应用程序连接到队列管理器时获取的。

队列管理器对订户队列执行权限检查，但不对受管队列执行权限检查。

当应用程序使用 MQCLOSE 调用删除永久动态队列时

MQCLOSE 调用上指定的对象句柄不一定与创建永久动态队列的 MQOPEN 调用所返回的对象句柄相同。 如果不同，那么队列管理器将检查与发出 MQCLOSE 调用的应用程序相关联的用户标识。 它将检查用户标识是否有权删除队列。

如果关闭预订以将其除去的应用程序未创建该预订，那么需要相应的权限来将其除去。

当命令服务器处理对 IBM MQ 对象执行的 PCF 命令时

此规则包括 PCF 命令对认证信息对象执行操作的情况。

用于权限检查的用户标识是在 PCF 命令的消息描述符中的 UserIdentifier 字段中找到的用户标识。 此用户标识必须在处理该命令的队列管理器上具有必需的权限。 封装在 Escape PCF 命令中的等效 MQSC 命令以相同的方式处理。 有关 UserIdentifier 字段及其设置方式的更多信息，请参阅 第 82 页的『消息上下文』。

IBM i **在 IBM i 上，当用户在组 2 中发出对 IBM MQ 对象执行操作的 CL 命令时**

此规则包括组 2 中的 CL 命令对认证信息对象执行操作的情况。

执行检查以确定用户是否有权对与该命令关联的 IBM MQ 对象执行操作。 除非用户是 QMQMADM 组的成员或具有 *ALLOBJ 权限，否则将执行这些检查。 所需的权限取决于命令对对象执行的操作类型。 例如，命令 **CHGMQMQ**"更改 MQM 队列" 需要更改该命令指定的队列属性的权限。 相反，命令 **DSPMQMQ**"显示 MQM 队列" 需要具有显示该命令指定的队列的属性的权限。

许多命令对多个对象执行操作。 例如，要发出命令 **DLTMQMQ**"删除 MQM 队列" ，需要以下权限:

- 连接到命令指定的队列管理器的权限
- 删除命令指定的队列的权限

某些命令完全不对任何对象执行操作。 在这种情况下，用户仅需要 IBM i 权限即可发出其中一个命令。 **STRMQMLSR** 启动 MQM 侦听器是此类命令的示例。

备用用户权限

当应用程序打开对象或预订主题时，该应用程序可以在 MQOPEN， MQPUT1 或 MQSUB 调用上提供用户标识。 它可以要求队列管理器将此用户标识用于权限检查，而不是与应用程序相关联的用户标识。

仅当同时满足以下两个条件时，应用程序才能成功打开对象:

- 与应用程序关联的用户标识具有为权限检查提供其他用户标识的权限。 该应用程序被认为具有 备用用户权限。
- 应用程序提供的用户标识具有打开所请求操作类型的对象或预订主题的权限。

消息上下文

消息上下文 信息允许检索消息的应用程序了解消息的发起方。 信息保存在消息描述符中的字段中，这些字段分为三个逻辑部分

这些部分如下:

标识上下文 (identity context)
这些字段包含有关将消息放入队列的应用程序用户的信息。

源上下文
这些字段包含有关应用程序本身以及消息何时放入队列的信息。

用户上下文
这些字段包含应用程序可用于选择队列管理器应交付的消息的消息属性。

当应用程序将消息放入队列时，应用程序可以要求队列管理器在消息中生成上下文信息。 这是缺省操作。 或者，它可以指定上下文字段不包含任何信息。 与应用程序关联的用户标识不需要任何特权即可执行这些操作。

应用程序可以在消息中设置身份上下文字段，从而允许队列管理器生成源上下文，也可以设置所有上下文字段。 应用程序还可以将身份上下文字段从它检索到的消息传递到它正在放入队列中的消息，也可以传递所有上下文字段。 但是，与应用程序关联的用户标识需要设置或传递上下文信息的权限。 应用程序指定它打算在打开将要放置消息的队列时设置或传递上下文信息，此时将检查其权限。

以下是每个上下文字段的简要描述:

身份上下文

UserIdentifier
与放置消息的应用程序关联的用户标识。 如果队列管理器设置此字段，那么它将设置为应用程序连接到队列管理器时从操作系统获取的用户标识。

AccountingToken
可用于对由于消息而完成的工作收取费用的信息。

ApplIdentityData
如果与应用程序关联的用户标识有权设置身份上下文字段或设置所有上下文字段，那么应用程序可以将此字段设置为与身份相关的任何值。 如果队列管理器设置此字段，那么它将设置为空白。

源上下文

PutApplType
放置消息的应用程序的类型; 例如， CICS 事务。

PutApplName
放置消息的应用程序的名称。

PutDate
放入消息的日期。

PutTime
放入消息的时间。

ApplOriginData
如果与应用程序关联的用户标识有权设置所有上下文字段，那么应用程序可以将此字段设置为与源相关的任何值。 如果队列管理器设置此字段，那么它将设置为空白。

用户上下文

MQINQMP 或 **MQSETMP** 支持以下值:

MQPD_USER _CONTEXT

该属性与用户上下文相关联。

无需特殊授权即可使用 MQSETMP 调用来设置与用户上下文关联的属性。

在 V7.0 或后续队列管理器上，将保存与用户上下文关联的属性，如 MQOO_SAVE_ALL_CONTEXT 所述。 指定了 MQOO_PASS_ALL_CONTEXT 的 MQPUT 会导致将属性从保存的上下文复制到新消息中。

MQPD_NO_CONTEXT

该属性未与消息上下文关联。

使用 MQRC_PD_ERROR 拒绝无法识别的值。 此字段的初始值为 **MQPD_NO_CONTEXT**。

有关每个上下文字段的详细描述，请参阅 MQMD-消息描述符。 有关如何使用消息上下文的更多信息，请参阅 消息上下文。

IBM i ALW 在 IBM i *IBM i* 和 *AIX, Linux, and Windows* 系统上使用 *IBM MQ* 对象的权限

IBM MQ 随附的授权服务组件称为 对象权限管理器 (OAM)。 它通过认证和授权检查提供访问控制。

认证。

IBM MQ 随附的 OAM 执行的认证检查是基本的，并且仅在特定情况下执行。 它不打算满足在高度安全的环境中预期的严格要求。

当应用程序连接到队列管理器时， OAM 将执行其认证检查，并且满足以下条件:

- 如果连接应用程序提供了 MQCSP 结构，并且
- MQCSP 结构中的 *AuthenticationType* 属性的值为 MQCSP_AUTH_USER_ID_AND_PWD ，并且
- 配置的 AUTHINFO 对象上的 CHCKLOCL 或 CHKCCLNT 值不是 "NONE"

OAM 中的认证步骤使用操作系统服务来验证密码，这些服务可能已配置为执行其他检查，例如，确保用户名没有太多不正确的密码测试尝试。

如果您编写新的授权服务组件，或者从供应商处获取授权服务组件，那么可以使用备用认证机制。

授权。

授权检查是全面的，旨在满足大多数正常要求。

当应用程序发出 MQI 调用以访问队列管理器，队列，进程，主题或名称列表时，将执行授权检查。 它们也在其他时间执行，例如，当命令服务器正在执行命令时。

在 IBM i IBM i AIX, Linux, and Windows 系统上， 授权服务 在应用程序发出 MQI 调用以访问作为队列管理器，队列，进程，主题或名称列表的 IBM MQ 对象时提供访问控制。 这包括检查备用用户权限以及设置或传递上下文信息的权限。

Windows 在 Windows 上，即使启用了 UAC， OAM 也会授予 Administrators 组的成员访问所有 IBM MQ 对象的权限。 此外，在 Windows 系统上， SYSTEM 帐户具有对 IBM MQ 资源的完全访问权。

当 PCF 命令对其中一个 IBM MQ 对象或认证信息对象执行操作时，授权服务还提供权限检查。 封装在 Escape PCF 命令中的等效 MQSC 命令以相同的方式处理。

IBM i 在 IBM i 上，除非用户是 QMQMADM 组的成员或具有 *ALLOBJ 权限，否则当用户在组 2 中发出对这些 IBM MQ 对象或认证信息对象进行操作的 CL 命令时，授权服务还会提供权限检查。

授权服务是 可安装服务，这意味着它由一个或多个 可安装服务组件实现。 使用记录的接口调用每个组件。 这使用户和供应商能够提供组件来扩充或替换 IBM MQ 产品提供的组件。

IBM MQ 随附的授权服务组件称为对象权限管理器 (OAM)。 将为您创建的每个队列管理器自动启用 OAM。

OAM 为其控制访问权的每个 IBM MQ 对象维护访问控制表 (ACL)。 在 AIX and Linux 系统上，只有组标识才能显示在 ACL 中。 这意味着组的所有成员都具有相同的权限。 在 IBM i IBM i 和 Windows 系统上，用户标识和组标识都可以显示在 ACL 中。 这意味着可以向个别用户和组授予权限。

12 个字符的限制同时适用于组和用户标识。 UNIX 平台通常将用户标识的长度限制为 12 个字符。 AIX 和 Linux 已提高此限制，但 IBM MQ 继续在所有 UNIX 平台上观察到 12 个字符的限制。 如果使用超过 12 个字符的用户标识，那么 IBM MQ 会将其替换为值 “UNKNOWN”。 请勿定义值为 “UNKNOWN”的用户标识。

OAM 可以认证用户并更改相应的身份上下文字段。 通过在 MQCONNX 调用上指定连接安全性参数结构 (MQCSP) 来启用此功能。 该结构将传递到 OAM Authenticate User 函数 (MQZ_AUTHENTICATE_USER) ，该函数用于设置相应的身份上下文字段。 如果来自 IBM MQ 客户机的 MQCONNX 连接，那么 MQCSP 中的信息将流向客户机通过客户机连接和服务器连接通道连接到的队列管理器。 如果在该通道上定义了安全出

口，那么 MQCSP 将传递到每个安全出口中，并且可以由该出口改变。 安全出口还可以创建 MQCSP。 有关在此上下文中使用安全出口的更多详细信息，请参阅 通道安全出口程序。

警告: 在某些情况下，客户机应用程序的 MQCSP 结构中的密码将通过纯文本网络发送。 要确保客户机应用程序密码受到适当保护，请参阅 IBM MQCSP 密码保护。

在 AIX, Linux, and Windows 系统上，控制命令 **setmqaut** 授予和撤销权限，并用于维护 ACL。 例如，命令:

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

允许组航程器的成员浏览队列 MOON.EUROPA 。 它还允许成员从队列中获取消息。 要稍后撤销这些权限，请输入以下命令:

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

该命令:

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

允许组航程器的成员将消息放在名称以字符 MOON. 开头的任何队列上。 MOON.* 是通用概要文件的名称。通用概要文件 允许您使用单个 **setmqaut** 命令授予对一组对象的权限。

控制命令 **dspmqaut** 可用于显示用户或组对指定对象具有的当前权限。控制命令 **dmpmqaut** 还可用于显示与通用概要文件关联的当前权限。

IBM i 在 IBM i 上，管理员使用 CL 命令 GRTMQMAUT 来授予权限，使用 CL 命令 RVKMQMAUT 来撤销权限。 也可以使用通用概要文件。例如， CL 命令:

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

提供与 **setmqaut** 命令的先前示例相同的功能; 它允许组航程器的成员将消息放在名称以字符 MOON. 开头的任何队列上

IBM i CL 命令 DSPMQMAUT 显示用户或组对指定对象具有的当前权限。 CL 命令 WRKMQMAUT 和 WRKMQMAUTD 也可用于处理与对象和通用概要文件关联的当前权限。

如果您不希望进行任何权限检查 (例如，在测试环境中) ，那么可以禁用 OAM。

Multi 使用 *PCF* 访问 *OAM* 命令
在 IBM i AIX, Linux, and Windows 系统上，可以使用 PCF 命令来访问 OAM 管理命令。

PCF 命令及其等效 OAM 命令如下所示:

表 *8: PCF* 命令及其等效 *OAM* 命令

PCF 命令	**OAM 命令**
查询权限记录	Dmpmqaut
查询实体权限	长石
设置权限记录	塞特 MQaut
删除权限记录	带有 -remove 选项的 setmqaut

setmqaut 和 **dmpmqaut** 命令仅限于 mqm 组的成员。 在队列管理器上已被授予 dsp 和 chg 权限的任何组中的用户都可以执行等效的 PCF 命令。

有关使用这些命令的更多信息，请参阅 可编程命令格式简介。

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 77.

远程消息传递的安全性

本部分涉及安全性的远程消息传递方面。

您必须为用户提供使用 IBM MQ 工具的权限。 这是根据要对对象和定义执行的操作来组织的。 例如:

- 队列管理器可由授权用户启动和停止
- 应用程序必须连接到队列管理器并具有使用队列的权限
- 消息通道必须由授权用户创建和控制
- 对象保留在库中，可限制对这些库的访问

远程站点上的消息通道代理程序必须检查所传递的消息是否源自有权在此远程站点上执行此操作的用户。 此外，由于可以远程启动 MCA，因此可能需要验证尝试启动 MCA 的远程进程是否有权执行此操作。 您可以通过以下四种方法来处理此问题:

1. 适当使用 RCVR， RQSTR 或 CLUSRCVR 通道定义的 PutAuthority 属性，以控制在将入局消息放入队列时用于授权检查的用户。 请参阅 "MQSC 命令参考" 中的 DEFINE CHANNEL 命令描述。
2. 实现通道认证记录以拒绝不需要的连接尝试，或基于以下内容设置 MCAUSER 值: 远程 IP 地址，远程用户标识，提供的 TLS 主题专有名称 (DN) 或远程队列管理器名称。
3. 实施 用户出口 安全性检查以确保相应的消息通道已获得授权。 托管相应通道的安装的安全性可确保所有用户都获得正确授权，因此您不需要检查个别消息。
4. 实现 用户出口 消息处理，以确保对个别消息进行审核以进行授权。

IBM i *IBM MQ for IBM i* 对象的安全性

本部分涉及安全性的远程消息传递方面。

您必须为用户提供使用 IBM MQ for IBM i 工具的权限。 此权限根据要对对象和定义执行的操作进行组织。 例如:

- 队列管理器可由授权用户启动和停止
- 应用程序需要连接到队列管理器，并且具有使用队列的权限
- 需要由授权用户创建和控制消息通道

远程站点上的消息通道代理程序必须检查所传递的消息是否派生自有权对此远程站点上的消息进行认证的用户。 此外，由于可以远程启动 MCA，因此可能需要验证尝试启动 MCA 的远程进程是否有权执行此操作。 您可以通过以下四种方法来处理此问题:

- 通道定义中的法令，即消息必须包含可接受的 上下文 权限，否则将废弃这些权限。
- 实现通道认证记录以拒绝不需要的连接尝试，或基于下列其中一项设置 MCAUSER 值: 远程 IP 地址，远程用户标识，提供的 TLS 专有名称 (DN) 或远程队列管理器名称。
- 实施用户出口安全检查，以确保相应的消息通道获得授权。 托管相应通道的安装的安全性可确保所有用户都获得正确授权，因此您不需要检查个别消息。
- 实现用户出口消息处理，以确保对个别消息进行审核以进行授权。

以下是有关 IBM MQ for IBM i 操作安全性方式的一些事实:

- 用户由 IBM i 进行标识和认证。
- 应用程序调用的队列管理器服务以队列管理器用户概要文件的权限运行，但在用户的进程中运行。
- 用户命令调用的队列管理器服务以队列管理器用户概要文件的权限运行。

Linux AIX *AIX and Linux* 上对象的安全性

如果此标识将使用 IBM MQ 管理命令，那么管理用户必须是系统 (包括 root 用户) 上 mqm 组的一部分。

您应该始终将 amqcrsta 作为 "mqm" 用户标识运行。

AIX and Linux 上的用户标识

队列管理器将所有大写或混合大小写的用户标识转换为小写。 然后，队列管理器将用户标识插入到消息的上下文部分中，或者检查其权限。 因此，授权仅基于小写标识。

Windows *Windows 系统上对象的安全性*

如果此标识将使用 IBM MQ 管理命令，那么管理用户必须同时属于 Windows 系统上的 mqm 组和管理员组。

Windows 系统上的用户标识

在 Windows 系统上， 如果未安装消息出口，那么队列管理器会将任何大写或混合大小写的用户标识转换为小写。 然后，队列管理器将用户标识插入到消息的上下文部分中，或者检查其权限。 因此，授权仅基于小写标识。

跨系统的用户标识

除 AIX, Linux, and Windows 系统以外的平台对消息中的用户标识使用大写字符。 要允许 AIX, Linux, and Windows 系统在消息中使用小写用户标识，消息通道代理程序 (MCA) 必须执行相应的字母字符转换。

为了允许 AIX, Linux, and Windows 系统在消息中使用小写用户标识，消息通道代理程序 (MCA) 在这些平台上执行以下转换:

在发送端
如果未安装消息出口，那么所有用户标识中的字母字符都将转换为大写字符。

在接收端
如果未安装消息出口，那么所有用户标识中的字母字符都将转换为小写字符。

如果出于任何其他原因在 AIX, Linux, and Windows 上提供消息出口，那么不会执行自动转换。

使用定制授权服务

IBM MQ 提供可安装的授权服务。 您可以选择安装备用服务。

随 IBM MQ 提供的授权服务组件称为对象权限管理器 (OAM)。 如果 OAM 未提供所需的授权工具，那么您可以编写自己的授权服务组件。 可安装服务接口参考信息中描述了必须由授权服务组件实现的可安装服务功能。

客户机的访问控制

访问控制基于用户标识。 可以有许多要管理的用户标识，并且用户标识可以采用不同的格式。 您可以将服务器连接通道属性 MCAUSER 设置为供客户机使用的特殊用户标识值。

IBM MQ 中的访问控制基于用户标识。 通常使用执行 MQI 调用的进程的用户标识。 对于 MQ MQI 客户机，服务器连接 MCA 代表 MQ MQI 客户机进行 MQI 调用。 您可以为要用于进行 MQI 调用的服务器连接 MCA 选择备用用户标识。 备用用户标识可以与客户机工作站相关联，也可以与您选择组织和控制客户机访问权的任何内容相关联。 用户标识需要在服务器上为其分配必要的权限才能发出 MQI 调用。 选择备用用户标识优于允许客户机使用服务器连接 MCA 的权限进行 MQI 调用。

表 9: 服务器连接通道使用的用户标识

用户标识	使用时
由安全出口设置的用户标识	除非被 **CHLAUTH TYPE(BLOCKUSER)** 规则阻止，否则将使用此属性。 请参阅以下部分 (第 88 页的『在安全出口中设置用户标识』) 以获取更多信息。
由 CHLAUTH 规则设置的用户标识	使用，除非被安全出口覆盖。 有关更多信息，请参阅 通道认证记录 。
SVRCONN 通道定义的 **MCAUSER** 属性中定义的用户标识	使用，除非被安全出口或 CHLAUTH 规则覆盖。
从客户机流出的用户标识	未通过任何其他方法设置用户标识时使用。
启动服务器连接通道的用户标识	在未通过任何其他方法设置用户标识且未流动客户机用户标识时使用。 请参阅以下部分 (第 88 页的『运行通道程序的用户标识』) 以获取更多信息。

由于服务器连接 MCA 代表远程用户进行 MQI 调用，因此考虑代表远程客户机发出 MQI 调用的服务器连接 MCA 的安全性影响以及如何管理可能大量用户的访问权非常重要。

- 一种方法是服务器连接 MCA 以自己的权限发出 MQI 调用。 但请注意，通常不希望服务器连接 MCA 以其强大的访问功能代表客户机用户发出 MQI 调用。
- 另一种方法是使用从客户机流的用户标识。 服务器连接 MCA 可以使用客户机用户标识的访问功能发出 MQI 调用。 这种方法提出了一些需要考虑的问题:
  1. 不同平台上的用户标识有不同的格式。 如果客户机上用户标识的格式与服务器上可接受的格式不同，那么这有时会导致问题。
  2. 可能有许多客户机具有不同的用户标识和正在更改的用户标识。 需要在服务器上定义和管理标识。
  3. 要信任用户标识吗? 任何用户标识都可以从客户机流出，而不一定是已登录用户的标识。 例如，客户机可能流具有完全 `mqm` 权限的标识，出于安全原因，该标识仅在服务器上有意定义。
- 首选方法是在服务器上定义客户机标识令牌，从而限制客户机连接的应用程序的功能。 这通常是通过将服务器连接通道属性 MCAUSER 设置为要由客户机使用的特殊用户标识值，并定义少量标识以供在服务器上具有不同权限级别的客户机使用来完成的。

在安全出口中设置用户标识

对于 IBM MQ MQI clients，发出 MQI 调用的进程是服务器连接 MCA。 服务器连接 MCA 使用的用户标识包含在 MQCD 的 `MCAUserIdentifier` 或 `LongMCAUserIdentifier` 字段中。 这些字段的内容由下列各项设置:

- 安全出口设置的任何值
- 来自客户机的用户标识
- MCAUSER (在服务器连接通道定义中)

当调用安全出口时，它可以覆盖对其可视的值。

- 如果服务器连接通道 MCAUSER 属性设置为非空白，那么将使用 MCAUSER 值。
- 如果服务器连接通道 MCAUSER 属性为空，那么将使用从客户机接收的用户标识。
- 如果服务器连接通道 MCAUSER 属性为空，并且未从客户机接收用户标识，那么将使用启动服务器连接通道的用户标识。

当正在使用客户机端安全性出口时， IBM MQ 客户机不会将已断言的用户标识流至服务器。

运行通道程序的用户标识

当用户标识字段派生自启动服务器连接通道的用户标识时，将使用以下值:

- z/OS 对于 z/OS，这是 z/OS 启动过程表分配给通道启动程序启动任务的用户标识。
- 对于 TCP/IP (非 z/OS)，这是 `inetd.conf` 条目中的用户标识或启动侦听器的用户标识。
- 对于 SNA (非 z/OS)，这是来自 SNA 服务器项或 (如果没有) 入局连接请求的用户标识，或者是启动侦听器的用户标识。
- 对于 NetBIOS 或 SPX，启动侦听器的用户标识。

如果存在任何将 MCAUSER 属性设置为空白的服务器连接通道定义，那么客户机可以使用此通道定义来连接到具有由客户机提供的用户标识确定的访问权限的队列管理器。 如果运行队列管理器的系统允许未经授权的网络连接，那么这可能是安全漏洞。 IBM MQ 缺省服务器连接通道 (SYSTEM.DEF.SVRCONN) 将 MCAUSER 属性设置为空白。 要防止未经授权的访问，请使用无权访问 IBM MQ MQ 对象的用户标识来更新缺省定义的 MCAUSER 属性。

用户标识的情况

使用 `runmqsc` 定义通道时，除非用户标识包含在单引号内，否则 MCAUSER 属性将更改为大写。

ALW 对于 AIX, Linux, and Windows 上的服务器，从客户机接收的 `MCAUserIdentifier` 字段的内容将更改为小写。

IBM i 对于 IBM i 上的服务器，从客户机接收的 LongMCAUserIdentifier 字段的内容将更改为大写。

Linux AIX 对于 AIX and Linux 系统上的服务器，从客户机接收的 LongMCAUserIdentifier 字段的内容将更改为小写。

缺省情况下，使用 IBM MQ JMS 绑定应用程序时传递的用户标识是运行应用程序的 JVM 的用户标识。

还可以通过 createQueueConnection 方法传递用户标识。

规划机密性

规划如何对数据保密。

您可以在应用程序级别或链接级别实现机密性。 您可以选择使用 TLS，在这种情况下，必须规划数字证书的使用。 如果标准设施不满足您的要求，您还可以使用通道出口程序。

相关概念

第 89 页的『比较链接级别安全性和应用程序级别安全性』
本主题包含有关链接级别安全性和应用程序级别安全性的各个方面的信息，并比较两个级别的安全性。

第 93 页的『通道出口程序』
通道出口程序 是在 MCA 的处理序列中定义的位置调用的程序。 用户和供应商可以编写自己的通道出口程序。 部分由 IBM 提供。

第 98 页的『使用 SSL/TLS 保护通道』
IBM MQ 中的 TLS 支持使用队列管理器认证信息对象和各种 MQSC 命令。 您还必须考虑使用数字证书。

比较链接级别安全性和应用程序级别安全性

本主题包含有关链接级别安全性和应用程序级别安全性的各个方面的信息，并比较两个级别的安全性。

第 89 页的图 10 中说明了链接级别和应用程序级别安全性。

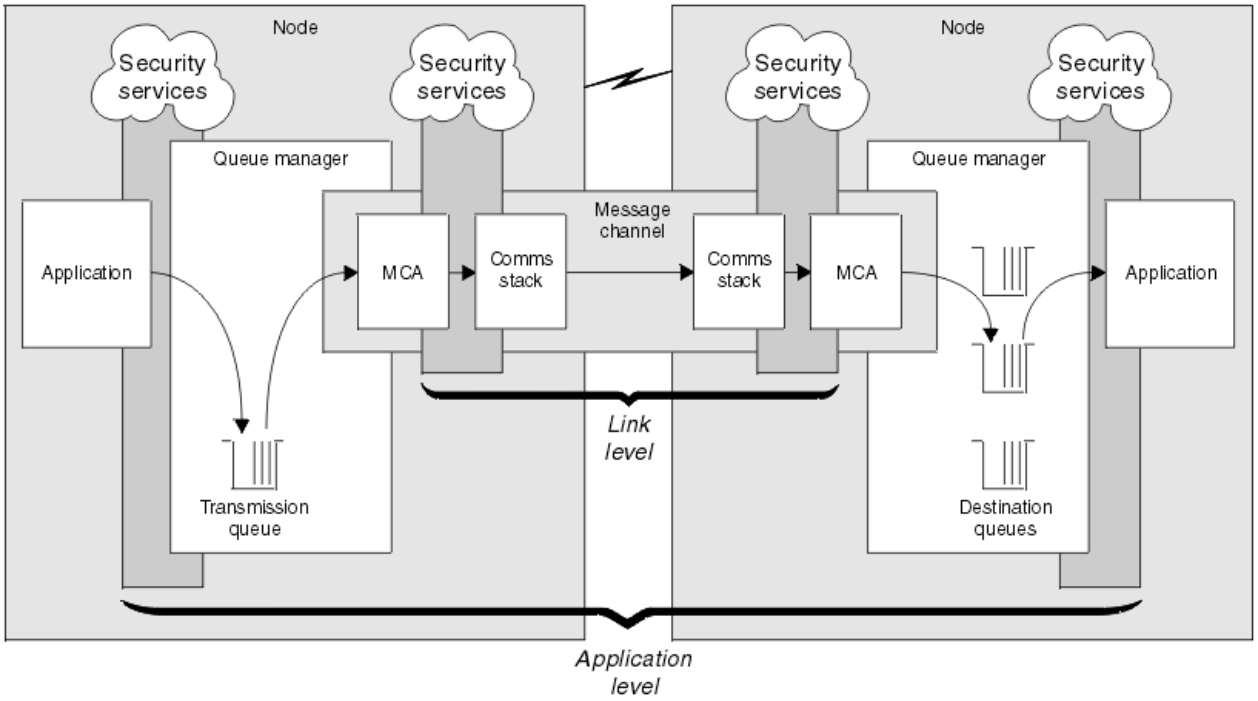


图 *10:* 链接级别安全性和应用程序级别安全性

保护队列中的消息

链路级别安全性可以在消息从一个队列管理器传输到另一个队列管理器时对其进行保护。 当通过不安全的网络传输消息时，这一点尤为重要。 但是，当消息存储在源队列管理器，目标队列管理器或中间队列管理器的队列中时，它无法保护这些消息。

z/OS z/OS 数据集加密可提供对存储在队列上的消息的某种保护，但仅针对本地队列管理器上的静态数据。 请参阅 使用数据集加密在 IBM MQ for z/OS 上静态数据的机密性 部分。 for more information.

相比之下，应用程序级别安全性可以在消息存储在队列中时对其进行保护，即使在未使用分布式排队时也适用。 这是链接级别安全性与应用程序级别安全性之间的主要区别，在 第 89 页的图 10 中进行了说明。

队列管理器未在受控环境和可信环境中运行

如果队列管理器正在受控制且可信的环境中运行，那么 IBM MQ 提供的访问控制机制可能被视为足以保护存储在其队列中的消息。 如果仅涉及本地排队并且消息从不离开队列管理器，那么尤其如此。 在这种情况下，可能认为不需要应用程序级别安全性。

如果将消息传输到另一个也在受控制和可信环境中运行的队列管理器，或者从此类队列管理器接收消息，那么也可能认为不需要应用程序级别安全性。 当消息传输到未在受控制的可信环境中运行的队列管理器或从该队列管理器接收到消息时，对应用程序级别安全性的需求将变得更大。

费用差异

在管理和性能方面，应用程序级别安全性的成本可能高于链接级别安全性。

由于配置和维护可能存在更多约束，因此管理成本可能会更高。 例如，您可能需要确保特定用户仅发送特定类型的消息，并仅将消息发送到特定目标。 相反，您可能需要确保特定用户仅接收特定类型的消息，并仅接收来自特定源的消息。 您可能需要为在单个消息通道上交换消息的每对用户配置和维护规则，而不是在该通道上管理链接级别安全服务。

如果每次应用程序放置或获取消息时都调用安全服务，那么可能会影响性能。

组织倾向于首先考虑链接级别安全性，因为实施起来可能更容易。 如果他们发现链接级别安全性不满足其所有需求，那么他们会考虑应用程序级别安全性。

组件的可用性

通常，在分布式环境中，安全服务需要至少两个系统上的组件。 例如，消息可能在一个系统上加密，而在另一个系统上解密。 这适用于链接级别安全性和应用程序级别安全性。

在异构环境中，使用的平台不同，每个平台都具有不同级别的安全功能，因此安全服务的必需组件可能无法用于需要这些组件的每个平台，也无法以易于使用的形式提供这些组件。 这可能是应用程序级别安全性的问题，而不是链接级别安全性的问题，尤其是如果您打算通过从各种来源购买组件来提供自己的应用程序级别安全性。

死信队列中的消息

如果消息受应用程序级别安全性保护，那么由于任何原因，如果消息未到达其目标并被放入死信队列中，那么可能存在问题。 如果无法确定如何处理消息描述符和死信头中的信息中的消息，那么可能需要检查应用程序数据的内容。 如果应用程序数据已加密并且只有预期的接收方可以对其进行解密，那么无法执行此操作。

应用程序级别安全性无法执行的操作

应用程序级别安全性不是完整的解决方案。 即使您实施了应用程序级别安全性，也可能仍需要一些链接级别安全性服务。 例如：

- 当通道启动时，可能仍需要两个 MCA 的相互认证。 此操作只能由链接级别安全服务完成。
- 应用程序级别安全性无法保护包含嵌入式消息描述符的传输队列头 MQXQH。 它也无法保护除消息数据以外的 IBM MQ 通道协议流中的数据。 只有链路级别安全性才能提供此保护。

- 如果在 MQI 通道的服务器端调用应用程序级别安全服务，那么这些服务无法保护通过该通道发送的 MQI 调用的参数。 特别是， MQPUT， MQPUT1 或 MQGET 调用中的应用程序数据不受保护。 在此情况下，只有链路级别安全性可以提供保护。

链路级别安全性 *(link level security)*

链路级别安全性 是指由 MCA，通信子系统或两者的组合直接或间接调用的安全服务。

链接级别安全性在 第 89 页的图 10 中进行了说明。

以下是链接级别安全服务的一些示例:

- 消息通道的每个端的 MCA 都可以认证其合作伙伴。 这是在通道启动并且已建立通信连接时，但在任何消息开始流动之前完成的。 如果在任一端认证失败，那么将关闭通道，并且不会传输任何消息。 这是标识和认证服务的示例。
- 可以在通道的发送端对消息进行加密，并在接收端对消息进行解密。 这是保密服务的示例。
- 可以在通道的接收端检查消息，以确定在通过网络传输消息时是否有意修改了其内容。 这是数据完整性服务的示例。

IBM MQ 提供的链路级别安全性

在 IBM MQ 中提供机密性和数据完整性的主要方法是使用 TLS。 有关在 IBM MQ 中使用 TLS 的更多信息，请参阅 第 21 页的『IBM MQ 中的 TLS 安全协议』。 对于认证， IBM MQ 提供了使用通道认证记录的工具。 通道认证记录在各个通道或通道组的级别提供对授予连接系统的访问权的精确控制。 有关更多信息，请参阅 第 43 页的『通道认证记录』。

提供您自己的链接级别安全性

您可以提供自己的链接级别安全服务。 编写自己的通道出口程序是提供自己的链路级别安全服务的主要方法。

在 第 93 页的『通道出口程序』中引入了通道出口程序。 同一主题还描述了随 IBM MQ for Windows 提供的通道出口程序 (SSPI 通道出口程序)。 此通道出口程序以源格式提供，以便您可以修改源代码以满足您的需求。 如果此通道出口程序或其他供应商提供的通道出口程序不符合您的要求，您可以自行设计和编写。 本主题建议通道出口程序可以提供安全服务的方式。 有关如何编写通道出口程序的信息，请参阅 编写通道出口程序。

使用安全出口的链接级别安全性

安全出口通常成对工作; 通道的每一端都有一个安全出口。 在通道启动时完成初始数据协商后，将立即调用这些参数。

安全出口可用于提供标识和认证，访问控制和机密性。

使用消息出口的链路级别安全性

只能在消息通道上使用消息出口，而不能在 MQI 通道上使用消息出口。 它可以访问传输队列头 MQXQH (包括嵌入式消息描述符) 和消息中的应用程序数据。 它可以修改消息的内容并更改其长度。

消息出口可用于需要访问整个消息而不是其一部分的任何用途。

消息出口可用于提供标识和认证，访问控制，机密性，数据完整性和不可抵赖性，以及安全以外的原因。

使用发送和接收出口的链路级别安全性

可以在消息和 MQI 通道上使用发送和接收出口。 将对在通道上流动的所有类型的数据以及双向流动调用这些数据。

发送和接收出口可访问每个传输段。 它们可以修改其内容并更改其长度。

在消息通道上，如果 MCA 需要拆分消息并将其发送到多个传输段中，那么将为包含该消息的一部分的每个传输段调用发送出口，并且在接收端将为每个传输段调用接收出口。 如果 MQI 调用的输入或输出参数太大而无法在单个传输段中发送，那么 MQI 通道上也会发生相同的情况。

在 MQI 通道上，传输段的字节 10 标识 MQI 调用，并指示传输段是否包含调用的输入或输出参数。 发送和接收出口可以检查此字节，以确定 MQI 调用是否包含可能需要保护的应用程序数据。

当第一次调用发送出口时，为了获取和初始化它需要的任何资源，它可以要求 MCA 在保存传输段的缓冲区中预留指定的空间量。 当稍后调用它来处理传输段时，它可以使用此空间来添加加密密钥或数字签名，例如。 在通道另一端的相应接收出口可以除去发送出口添加的数据，并使用它来处理传输段。

发送和接收出口最适合不需要了解它们所处理的数据结构的目的，因此可以将每个传输段视为二进制对象。

发送和接收出口可用于提供机密性和数据完整性，以及用于安全性以外的用途。

相关任务

在发送或接收出口程序中标识 API 调用

应用程序级别安全性 *(application level security)*

应用程序级别安全性 是指在应用程序与其所连接的队列管理器之间的接口上调用的那些安全服务。

当应用程序向队列管理器发出 MQI 调用时，将调用这些服务。 应用程序，队列管理器，另一个支持 IBM MQ 的产品或其中任何一个协同工作的组合可能会直接或间接调用这些服务。 第 89 页的图 10 中说明了应用程序级别安全性。

应用程序级别安全性也称为 端到端安全性 或 消息级别安全性。

以下是应用程序级别安全服务的一些示例:

- 当应用程序将消息放入队列时，消息描述符包含与应用程序关联的用户标识。 但是，不存在可用于认证用户标识的数据，例如加密密码。 安全服务可以添加此数据。 当接收应用程序最终检索消息时，服务的另一个组件可以使用随消息一起传递的数据来认证用户标识。 这是标识和认证服务的示例。
- 当消息由应用程序放入队列时，可以对该消息进行加密，当接收应用程序检索到该消息时，可以对该消息进行解密。这是保密服务的示例。
- 接收应用程序检索消息时，可以检查该消息。 此检查确定自发送应用程序首次将其放入队列后，是否有意修改其内容。这是数据完整性服务的示例。

规划 *Advanced Message Security*

Advanced Message Security (AMS) 是 IBM MQ 的组件，可为流经 IBM MQ 网络的敏感数据提供高级别保护，同时不会影响最终应用程序。

如果您正在移动高度敏感或有价值的信息，特别是患者记录或信用卡详细信息等保密或支付相关信息，您必须特别注意信息安全。 确保围绕企业移动的信息保持其完整性并保护其免受未经授权的访问是一项持续的挑战和责任。 您也可能被要求遵守安全法规，存在因不合规而受到处罚的风险。

您可以将自己的安全扩展开发到 IBM MQ。 但是，此类解决方案需要专家技能，并且可能复杂且维护成本高昂。 在几乎所有类型的商业 IT 系统之间围绕企业移动信息时， Advanced Message Security 可帮助应对这些挑战。

Advanced Message Security 通过以下方式扩展了 IBM MQ 的安全功能:

- 它使用消息的加密或数字签名，为点到点消息传递基础结构提供应用程序级别的端到端数据保护。
- 它提供全面的安全性，无需编写复杂的安全代码或修改或重新编译现有应用程序。
- 它使用公用密钥基础结构 (PKI) 技术为消息提供认证，授权，机密性和数据完整性服务。
- 它为大型机和分布式服务器提供安全策略管理。
- 它支持 IBM MQ 服务器和客户机。
- 它与 Managed File Transfer 集成，以提供端到端安全消息传递解决方案。

有关更多信息，请参阅 第 522 页的『Advanced Message Security』。

提供您自己的应用程序级别安全性

您可以提供自己的应用程序级别安全服务。 为了帮助您实现应用程序级别安全性， IBM MQ 提供了两个出口，即 API 出口和 API 交叉出口。

API 出口和 API 交叉出口可以提供标识和认证，访问控制，机密性，数据完整性和不可抵赖性服务以及其他与安全性无关的功能。

如果系统环境中不支持 API 出口或 API 交叉出口，那么您可能需要考虑使用其他方法来提供自己的应用程序级别安全性。 一种方法是开发封装 MQI 的更高级别 API。 然后，程序员使用此 API (而不是 MQI) 来编写 IBM MQ 应用程序。

使用更高级别的 API 的最常见原因是:

- 向程序员隐藏 MQI 的更高级功能。
- 在使用 MQI 时实施标准。
- 将函数添加到 MQI。 此附加功能可以是安全服务。

某些供应商产品使用此技术为 IBM MQ 提供应用程序级别安全性。

如果计划以这种方式提供安全服务，请注意以下有关数据转换的信息:

- 如果已将安全性令牌 (例如数字签名) 添加到消息中的应用程序数据，那么执行数据转换的任何代码都必须知道存在此令牌。
- 安全性令牌可能已派生自应用程序数据的二进制映像。 因此，在转换数据之前，必须对令牌执行任何检查。
- 如果消息中的应用程序数据已加密，那么必须在数据转换之前对其进行解密。

通道出口程序

通道出口程序 是在 MCA 的处理序列中定义的位置调用的程序。 用户和供应商可以编写自己的通道出口程序。 部分由 IBM 提供。

有几种类型的通道出口程序，但只有四种类型的通道出口程序在提供链路级别安全性方面具有作用:

- 安全出口
- 消息出口
- 发送出口
- 接收出口

这四种类型的通道出口程序在 第 93 页的图 11 中进行了说明，并在以下主题中进行了描述。

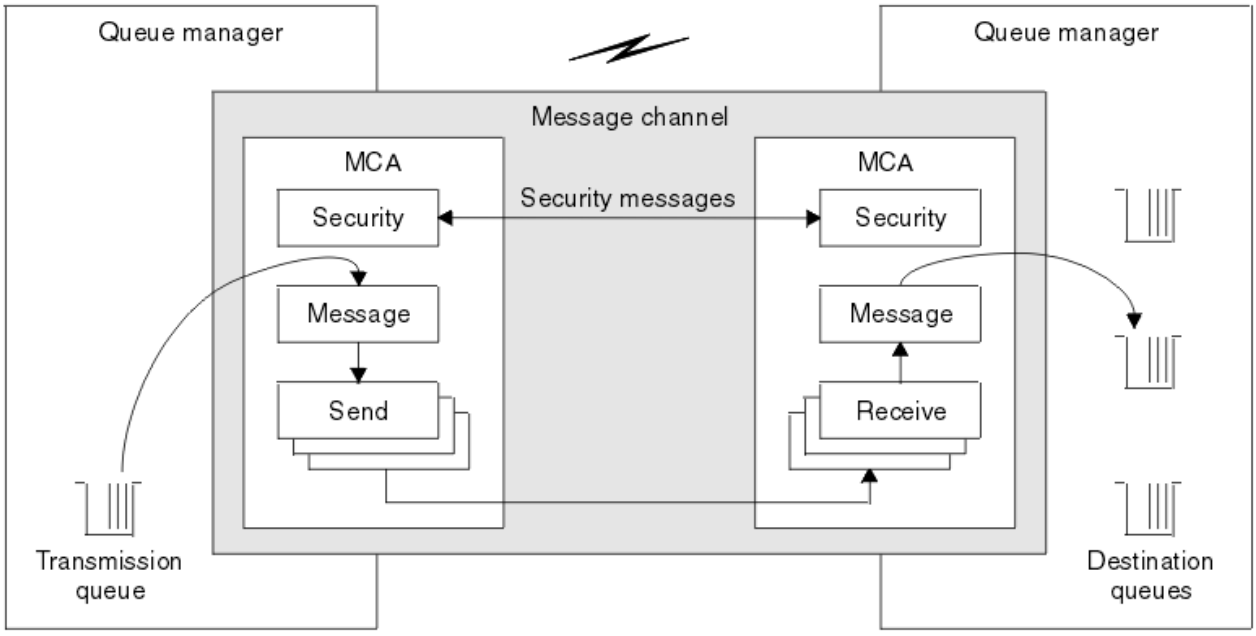


图 *11:* 消息通道上的安全性，消息，发送和接收出口

相关概念

消息传递通道的通道出口程序

安全出口概述

安全出口通常成对工作。 在消息流之前调用它们，它们的目的是允许 MCA 认证其合作伙伴。

安全出口 通常成对工作; 通道的每一端都有一个安全出口。 在通道启动时完成初始数据协商之后，但在任何消息开始流动之前，将立即调用这些消息。 安全出口的主要用途是在通道的每个端启用 MCA 以认证其合作伙伴。 但是，没有什么可以阻止安全出口执行其他功能，甚至是与安全性无关的功能。

安全出口可以通过发送 安全消息来相互通信。 未定义安全性消息的格式，它由用户确定。 交换安全消息的一个可能结果是其中一个安全出口可能决定不再继续。 在这种情况下，通道已关闭，并且消息未流动。 如果仅在通道的一端存在安全出口，那么仍会调用该出口，并且可以选择是继续还是关闭通道。

可以在消息和 MQI 通道上调用安全出口。 安全出口的名称在通道的每一端指定为通道定义中的参数。

有关安全出口的更多信息，请参阅 第 91 页的『使用安全出口的链接级别安全性』。

消息出口

消息出口仅在消息通道上运行，通常成对工作。 消息出口可以对整个消息进行操作并对其进行各种更改。

通道的发送端和接收端的 消息出口 通常成对工作。 在 MCA 从传输队列中获取消息后，将调用通道发送端的消息出口。 在通道的接收端，在 MCA 将消息放入其目标队列之前调用消息出口。

消息出口有权访问传输队列头 MQXQH (包括嵌入式消息描述符) 和消息中的应用程序数据。 消息出口可以修改消息内容并更改其长度。 长度更改可能是压缩，解压缩，加密或解密消息的结果。 这也可能是向消息添加数据或从消息中除去数据的结果。

消息出口可用于任何需要访问整个消息 (而不是其一部分) 的用途，而不一定用于安全性。

消息出口可以确定它当前正在处理的消息不应继续朝着其目标前进。 然后， MCA 将消息放在死信队列上。 消息出口也可以关闭通道。

只能在消息通道上调用消息出口，而不能在 MQI 通道上调用消息出口。 这是因为 MQI 通道的用途是允许 MQI 调用的输入和输出参数在 IBM MQ MQI client 应用程序与队列管理器之间流动。

在通道的每一端的通道定义中，将消息出口的名称指定为参数。 您还可以指定要连续运行的消息出口的列表。

有关消息出口的更多信息，请参阅 第 91 页的『使用消息出口的链路级别安全性』。

发送和接收出口

发送和接收出口通常成对工作。 它们在传输段上运行，并且最好在它们所处理的数据的结构不相关的情况下使用。

通道一端的 发送出口 和另一端的 接收出口 通常成对工作。 在 MCA 发出通信发送以通过通信连接发送数据之前，将调用发送出口。 仅当 MCA 在通信接收后重新获得控制并从通信连接接收数据之后，才会调用接收出口。 如果正在使用共享对话，那么将通过 MQI 通道为每个对话调用不同的发送和接收出口实例。

消息通道上的两个 MCA 之间的 IBM MQ 通道协议流包含控制信息以及消息数据。 同样，在 MQI 通道上，流包含控制信息以及 MQI 调用的参数。 针对所有类型的数据调用发送和接收出口。

消息数据在消息通道上仅以一个方向流动，但在 MQI 通道上， MQI 调用流的输入参数在一个方向流动，输出参数在另一个方向流动。 在消息和 MQI 通道上，控制双向信息流。 因此，可以在通道的两端调用发送和接收出口。

在两个 MCA 之间的单个流中传输的数据单元称为 传输段。 发送和接收出口可访问每个传输段。 它们可以修改其内容并更改其长度。 但是，发送出口不得更改传输段的前 8 个字节。 这些 8 字节构成 IBM MQ 通道协议头的一部分。 对于发送出口可以增加传输段的长度的大小也有一些限制。 特别是，发送出口不能将其长度增大到超过在通道启动时两个 MCA 之间协商的最大长度。

在消息通道上，如果消息过大而无法在单个传输段中发送，那么发送 MCA 将拆分该消息并在多个传输段中发送该消息。 因此，对包含消息的一部分的每个传输段调用发送出口，并且在接收端对每个传输段调用接收出口。 接收 MCA 在接收出口处理来自传输段的消息后重新构成该消息。

同样，在 MQI 通道上，如果 MQI 调用的输入或输出参数过大，那么会在多个传输段中发送这些参数。 例如，如果应用程序数据足够大，那么可能会在 MQPUT， MQPUT1 或 MQGET 调用上发生此情况。

考虑到这些因素，将发送和接收出口用于不需要了解它们正在处理的数据的结构的目的更为合适，因此可以将每个传输段视为二进制对象。

发送或接收出口可以关闭通道。

发送出口和接收出口的名称被指定为通道每一端的通道定义中的参数。 您还可以指定要连续运行的发送出口的列表。 同样，您可以指定接收出口的列表。

有关发送和接收出口的更多信息，请参阅 第 91 页的『使用发送和接收出口的链路级别安全性』。

规划数据完整性

规划如何保留数据的完整性。

您可以在应用程序级别或链接级别实现数据完整性。

在应用程序级别，如果标准设施不满足您的需求，那么可以使用 API 出口程序。 您可以选择使用 Advanced Message Security (AMS) 对消息进行数字签名，以防止未经授权的修改。

在链接级别，您可以选择使用 TLS，在这种情况下，必须规划数字证书的使用。 如果标准设施不满足您的要求，您还可以使用通道出口程序。

相关概念

第 98 页的『使用 SSL/TLS 保护通道』
IBM MQ 中的 TLS 支持使用队列管理器认证信息对象和各种 MQSC 命令。 您还必须考虑使用数字证书。

第 9 页的『数据完整性』
数据完整性 服务会检测是否存在未经授权的数据修改。

第 92 页的『规划 Advanced Message Security』
Advanced Message Security (AMS) 是 IBM MQ 的组件，可为流经 IBM MQ 网络的敏感数据提供高级别保护，同时不会影响最终应用程序。

相关参考

API 出口参考

通道出口调用和数据结构

规划审计

决定需要审计哪些数据，以及如何捕获和处理审计信息。 请考虑如何检查是否正确配置了系统。

活动监视有几个方面。 您必须考虑的方面通常由审计员需求定义，这些需求通常由监管标准 (如 HIPAA (健康保险可移植性和责任法案) 或 SOX (Sarbanes-Oxley)) 驱动。 IBM MQ 提供了旨在帮助遵守此类标准的功能。

请考虑是只对异常感兴趣，还是对所有系统行为感兴趣。

审计的某些方面也可以被视为操作监视; 审计的一个区别是，您经常查看历史数据，而不仅仅是查看实时警报。 监视在 监视和性能部分中进行了说明。

要审计的数据

请考虑需要审计哪些类型的数据或活动，如以下部分中所述:

使用 IBM MQ 接口对 IBM MQ 进行的更改
配置 IBM MQ 以发出检测事件，特别是命令事件和配置事件。

在 IBM MQ 的控制范围外对其进行的更改
某些更改可能会影响 IBM MQ 的行为方式，但无法由 IBM MQ 直接监视。 此类更改的示例包括对配置文件 `mqs.ini`， `qm.ini` 和 `mqclient.ini` 的更改，队列管理器的创建和删除，二进制文件 (例如用户出口程序) 的安装以及对文件许可权的更改。 要监视这些活动，必须使用在操作系统级别运行的工具。 不同的工具可用，适用于不同的操作系统。 您可能还具有由关联工具 (例如， *sudo*) 创建的日志。

IBM MQ 的操作控制
您可能必须使用操作系统工具来审计诸如启动和停止队列管理器之类的活动。 在某些情况下，可以将 IBM MQ 配置为发出检测事件。

IBM MQ 中的应用程序活动
要审计应用程序的操作 (例如，打开队列以及放入和获取消息) ，请配置 IBM MQ 以发出相应的事件。

入侵者警报
要审计尝试违反安全性的行为，请配置系统以发出授权事件。 通道事件对于显示活动可能也很有用，尤其是在通道意外结束时。

规划审计数据的捕获，显示和归档

您需要的许多元素报告为 IBM MQ 事件消息。 您必须选择可以读取和格式化这些消息的工具。 如果您对长期存储和分析感兴趣，那么必须将其移至辅助存储机制 (例如数据库)。 如果不处理这些消息，它们将保留在事件队列上，可能填充队列。 您可能决定实施一个根据某些事件自动执行操作的工具; 例如，在发生安全故障时发出警报。

验证是否正确配置了系统

IBM MQ Explorer 随附了一组测试。 使用这些信息来检查对象定义是否存在问题。

此外，请定期检查系统配置是否与您期望的一样。 虽然命令和配置事件可以在更改内容时报告，但转储配置并将其与已知的良好副本进行比较也很有用。

按拓扑规划安全性

本节涵盖特定情况下的安全性，即通道，队列管理器集群，发布/预订和多点广播应用程序以及使用防火墙时的安全性。

请参阅以下子主题以获取更多信息:

通道授权

通过通道发送或接收消息时，需要提供对各种 IBM MQ 资源的访问权。 消息通道代理程序 (MCA) 基本上是在队列管理器之间移动消息的 IBM MQ 应用程序，因此需要访问各种 IBM MQ 资源才能正确运行。

要在 PUT 时接收 MCA 的消息，可以使用与 MCA 关联的用户标识或与消息关联的用户标识。

在 CONNECT 时，您可以使用 **CHLAUTH** 通道认证记录将声明的用户标识映射到备用用户。

在 IBM MQ 中，通道可由 TLS 支持进行保护。

与发送和接收通道关联的用户标识 (不包括未使用 MCAUSER 属性的发送方通道) 需要访问以下资源:

- 与发送通道关联的用户标识需要访问队列管理器，传输队列，死信队列以及通道出口所需的任何其他资源。
- 接收方通道的 MCAUSER 用户标识需要 *+ setall* 权限。 原因是接收方通道必须使用从远程发送方通道接收到的数据来创建完整 MQMD (包括所有上下文字段)。 因此，队列管理器要求执行此活动的用户具有 *+ setall* 权限。 必须向以下用户授予此 *+ setall* 权限:
  - 接收方通道将消息有效放入的所有队列。
  - 队列管理器对象。 有关更多信息，请参阅 上下文授权。
- 发起方请求 COA 报告消息的接收方通道的 MCAUSER 用户标识需要返回报告消息的传输队列上的 *+* 钝化权限。 如果没有此权限，那么将记录 AMQ8077 错误消息。
- 通过与接收通道关联的用户标识，您可以打开目标队列以将消息放入队列中。 这涉及消息排队接口 (MQI) ，因此如果您未使用 IBM MQ 对象权限管理器 (OAM) ，那么可能需要进行其他访问控制检查。 您可以指定是针对与 MCA 关联的用户标识 (如本主题中所述) 进行授权检查，还是针对与消息关联的用户标识 (来自 MQMD UserIdentifier 字段) 进行授权检查。

  对于其应用的通道类型，通道定义的 **PUTAUT** 参数指定用于这些检查的用户标识。
  - 通道缺省为使用队列管理器的服务帐户，该帐户具有完整的管理权限并且不需要特殊权限。
  - 对于服务器连接通道，缺省情况下， CHLAUTH 规则会阻止管理连接，并且需要显式供应。
  - 类型为 "接收方" ， "请求者" 和 "集群接收方" 的通道允许由任何相邻队列管理器进行本地管理，除非管理员采取步骤来限制此访问权。

- 不必为接收方通道的 MCAUSER 用户标识授予 *dsp* 和 *ctrlx* 权限。
- 在 IBM MQ 8.0.0 Fix Pack 4 之前，如果您使用缺少 IBM MQ 管理特权的用户标识，那么必须将该通道的 **dsp** 和 **ctrlx** 权限授予该用户标识，以使该通道正常工作。

  从 IBM MQ 8.0.0 Fix Pack 4 开始，当通道再同步自身并更正序号时，不会进行权限检查。

  但是，手动发出 RESET CHANNEL 命令在所有发行版中仍需要 **+dsp** 和 **+ctrlx** 。

  注意: 当消息批处理确认需要通道重置时， IBM MQ 会尝试查询通道，这需要 **+dsp** 权限。
- 对于 SDR 通道类型， MCAUSER 属性未使用。
- 如果使用与消息关联的用户标识，那么该用户标识可能来自远程系统。 目标系统必须识别此远程系统用户标识。 以下命令是您可以发出以从远程系统向用户标识授予权限的命令类型的示例:

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

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

其中 *Profile* 是通道。

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

其中 *Profile* 是死信队列 (如果已设置)。

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

其中 *Profile* 是授权队列的列表。

注意: 授权用户标识将消息放入命令队列或其他敏感系统队列时，请谨慎操作。

与 MCA 关联的用户标识取决于 MCA 的类型。 有两种类型的 MCA:

调用者 MCA
启动通道的 MCA。 调用者 MCA 可以作为单个进程，作为通道启动程序的线程或作为进程池的线程来启动。 使用的用户标识是与父进程 (通道启动程序) 关联的用户标识，或者是与启动 MCA 的进程关联的用户标识。

响应者 MCA
响应者 MCA 是由于调用者 MCA 的请求而启动的 MCA。 响应者 MCA 可以作为单个进程，作为侦听器的线程或作为进程池的线程启动。 用户标识可以是下列任何一种类型 (按此首选项顺序):

1. 在 APPC 上，调用者 MCA 可以指示要用于响应者 MCA 的用户标识。 这称为网络用户标识，仅适用于作为个别进程启动的通道。 使用通道定义的 USERID 参数来设置网络用户标识。
2. 如果未使用 **USERID** 参数，那么响应程序 MCA 的通道定义可以指定 MCA 必须使用的用户标识。 使用通道定义的 **MCAUSER** 参数设置用户标识。
3. 如果先前 (两个) 方法中的任何一个未设置用户标识，那么将使用启动 MCA 的进程的用户标识或父进程 (侦听器) 的用户标识。

相关概念
第 43 页的『通道认证记录』
要在通道级别对授予连接系统的访问权进行更为精确的控制，可以使用通道认证记录。

相关参考
通道认证记录属性

保护通道启动程序定义
只有 mqm 组的成员才能处理通道启动程序。

IBM MQ 通道启动器不是 IBM MQ 对象; 对它们的访问不受 OAM 控制。 IBM MQ 不允许用户或应用程序处理这些对象，除非其用户标识是 mqm 组的成员。 如果您有发出 PCF 命令 **StartChannelInitiator** 的应用程序，那么 PCF 消息的消息描述符中指定的用户标识必须是目标队列管理器上 mqm 组的成员。

用户标识还必须是目标机器上 mqm 组的成员，才能通过 Escape PCF 命令或以间接方式使用 runmqsc 发出等效的 MQSC 命令。

传输队列

队列管理器会自动将远程消息放在传输队列上; 这不需要特殊权限。

但是，如果需要将消息直接放在传输队列上，那么这需要特殊授权; 请参阅 第 112 页的表 12。

通道出口

如果通道认证记录不合适，那么可以使用通道出口来提高安全性。 安全出口构成两个安全出口程序之间的安全连接。 一个程序用于发送消息通道代理 (MCA) ，一个程序用于接收 MCA。

请参阅 第 93 页的『通道出口程序』 ，以获取有关通道出口的更多信息。

使用 *SSL/TLS* 保护通道

IBM MQ 中的 TLS 支持使用队列管理器认证信息对象和各种 MQSC 命令。 您还必须考虑使用数字证书。

数字证书和密钥存储库

最好设置队列管理器证书标签属性 (**CERTLABL**) 到要用于大多数通道的个人证书的名称，并通过在需要不同证书的通道上设置证书标签来覆盖该个人证书以用于异常。

如果需要许多具有与队列管理器上的缺省证书集不同的证书的通道，那么应考虑在多个队列管理器之间划分这些通道，或者在队列管理器前面使用 MQIPT 代理来提供另一个证书。

您可以对每个通道使用不同的证书，但如果将太多证书存储在密钥存储库中，那么在启动 TLS 通道时可能会影响性能。 尝试将密钥存储库中的证书数保留为小于约 50 ，并将 100 视为最大值，因为 IBM Global Security Kit (GSKit) 性能会随着更大的密钥存储库而急剧下降。

允许在同一队列管理器上使用多个证书会增加在同一队列管理器上使用多个 CA 证书的可能性。 这将增加证书主体专有名称名称名称空间与由单独的认证中心发放的证书发生冲突的几率。

虽然专业认证中心可能更加谨慎，但内部认证中心通常缺乏明确的命名约定，最终可能导致一个 CA 与另一个 CA 之间的意外匹配。

除主题专有名称外，您还应检查证书颁发者专有名称。 要执行此操作，请使用通道认证 SSLPEERMAP 记录，并设置 **SSLPEER** 和 **SSLCERTI** 字段以分别与主体集 DN 和颁发者 DN 匹配。

自签名证书和 CA 签名证书

在开发和测试应用程序时以及在生产中使用数字证书时，规划数字证书的使用非常重要。 根据队列管理器和客户机应用程序的使用情况，您可以使用 CA 签名的证书或自签名证书。

CA 签名的证书

对于生产系统，请从可信认证中心 (CA) 获取证书。 从外部 CA 获取证书时，将为服务付费。

自签名证书

在开发应用程序时，可以使用本地 CA 颁发的自签名证书或证书，具体取决于平台:

ALW 在 AIX, Linux, and Windows 系统上，可以使用自签名证书。 有关指示信息，请参阅第 474 页的『在 AIX, Linux, and Windows 上创建自签名个人证书』。

IBM i 在 IBM i 系统上，可以使用本地 CA 签署的证书。 有关指示信息，请参阅第 254 页的『在 IBM i 上请求服务器证书』。

z/OS 在 z/OS 上，可以使用自签名证书或本地 CA 签名证书。 请参阅 第 275 页的『Creating a self-signed personal certificate on z/OS』 或 第 275 页的『Requesting a personal certificate on z/OS』 以获取指示信息。

自签名证书不适合生产使用，原因如下:

- 无法撤销自签名证书，这可能允许攻击者在私钥被泄露后破坏身份。 CA 可以撤销已泄密的证书，这将阻止其进一步使用。 因此， CA 签署的证书在生产环境中使用更安全，尽管自签名证书对于测试系统更方便。
- 自签名证书永不到期。 在测试环境中，这既方便又安全，但在生产环境中，这会使它们面临最终的安全漏洞。 自签名证书无法撤销，这一风险雪上加霜。
- 自签名证书既用作个人证书，也用作根 (或信任锚) CA 证书。 具有自签名个人证书的用户可能能够使用该证书来签署其他个人证书。 一般来说， CA 颁发的个人证书并不是这样，代表着重大的曝光。

CipherSpecs 和数字证书

只有一部分受支持的 CipherSpecs 可用于所有受支持的数字证书类型。 因此，必须为数字证书选择相应的 CipherSpec 。 同样，如果组织的安全策略要求使用特定 CipherSpec ，那么必须获取适当的数字证书。

有关 CipherSpecs 与数字证书之间的关系的更多信息，请参阅 第 39 页的『IBM MQ 中的数字证书和 CipherSpec 兼容性』

证书验证策略

IETF RFC 5280 标准指定了一系列证书验证规则，为了防止冒充攻击，合规的应用软件必须实现这些规则。 一组证书验证规则称为证书验证策略。 有关 IBM MQ 中的证书验证策略的更多信息，请参阅 第 38 页的『IBM MQ 中的证书验证策略』。

规划证书撤销检查

允许来自不同认证中心的多个证书可能会导致不必要的其他证书撤销检查。

特别是，如果您已显式配置使用特定 CA 中的撤销服务器 (例如，通过使用 AUTHINFO 对象或认证信息记录 (MQAIR) 结构) ，那么在提供来自其他 CA 的证书时，撤销检查将失败。

您应该避免显式证书撤销服务器配置。 而是应该启用隐式检查，其中每个证书在证书扩展 (例如， CRL 分发点或 OCSP AuthorityInfoAccess) 中包含其自己的撤销服务器位置。

有关更多信息，请参阅 OCSPCheckExtensions 和 CDPCheckExtensions。

TLS 支持的命令和属性

传输层安全性 (TLS) 协议提供通道安全性，防止窃听，篡改和冒充。 IBM MQ 对 TLS 的支持使您能够在通道定义上指定特定通道使用 TLS 安全性。 您还可以指定所需安全性类型的详细信息，例如要使用的加密算法。

- 以下 MQSC 命令支持 TLS:

  变更授权信息
  修改认证信息对象的属性。

  定义授权信息
  创建认证信息对象。

  删除授权信息
  删除认证信息对象。

  显示授权信息
  显示特定认证信息对象的属性。

- 以下队列管理器参数支持 TLS:

  CERTLABL
  定义要使用的个人证书标签。

  键盘 RPWD
  在 AIX, Linux, and Windows 系统上，定义 IBM MQ 用于访问密钥存储库的密码。 此字段使用密码保护系统进行加密。

  SSLCRLNL
  SSLCRLNL 属性指定用于提供证书撤销位置以允许增强 TLS 证书检查的认证信息对象的名称列表。

SSLCRYP
在 AIX, Linux, and Windows 系统上，设置 **SSLCryptoHardware** 队列管理器属性。 此属性是可用于配置系统上的加密硬件的参数字符串的名称。

SSLEV
确定如果使用 TLS 的通道无法建立 TLS 连接，是否报告 TLS 事件消息。

SSLFIPS
指定如果在 IBM MQ 中执行密码术，而不是在加密硬件中执行密码术，是否仅使用 FIPS 认证的算法。 如果配置了加密硬件，那么将使用硬件产品提供的加密模块，并且这些模块可能已通过 FIPS 认证到特定级别。 这取决于正在使用的硬件产品。

SSLKEYR
在 AIX, Linux, and Windows 系统上，将密钥存储库与队列管理器相关联。 GSKit 使您能够在 AIX, Linux, and Windows 系统上使用 TLS 安全性。

SSLRKEYC
在重新协商密钥之前，要在 TLS 对话中发送和接收的字节数。 此字节数包括由 MCA 发送的控制信息。

- 以下通道参数支持 TLS:

CERTLABL
定义要使用的个人证书标签。

SSLCAUTH
定义 IBM MQ 是否需要并验证来自 TLS 客户机的证书。

SSLCIPH
指定加密强度和功能 (CipherSpec) ，例如 TLS_RSA_WITH_AES_128_CBC_SHA。 CipherSpec 必须在通道两端匹配。

SSLPEER
指定允许的合作伙伴的专有名称 (唯一标识)。

本节描述用于支持认证信息对象的 **setmqaut**， **dspmqaut**， **dmpmqaut**， **rcrmqobj**， **rcdmqimg** 和 **dspmqfls** 命令。 它还描述了可用于在 AIX, Linux, and Windows 上管理密钥和证书的命令。 请参阅下列各部分:

- 设置为
- 长石 (dspmqaut)
- dmpmqaut
- rcrmqobj
- rcdmqimg
- dspmqfls
- 第 473 页的『在 AIX, Linux, and Windows 上管理密钥和证书』

有关使用 TLS 的通道安全性的概述，请参阅

- 第 21 页的『IBM MQ 中的 TLS 安全协议』

有关与 TLS 关联的 MQSC 命令的详细信息，请参阅

- ALTER AUTHINFO
- 定义的 AUTHINFO
- 删除 AUTHINFO
- 显示 AUTHINFO

有关与 TLS 关联的 PCF 命令的详细信息，请参阅

- 更改，复制和创建认证信息对象
- 删除认证信息对象
- 查询认证信息对象

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 “通道出口程序” on page 93 for more information about channel exits.

Related tasks

Writing channel exit programs on z/OS

SNA LU 6.2 安全服务

SNA LU 6.2 提供会话级密码术，会话级认证和对话级认证。

注: 此主题集合假定您已基本了解 "系统网络体系结构" (SNA)。 本节中提到的其他文件载有对相关概念和术语的简要介绍。 如果需要更全面的 SNA 技术简介，请参阅 *Systems Network Architecture Technical Overview*， GC30-3073。

SNA LU 6.2 提供三个安全服务:

- 会话级别密码术
- 会话级别认证
- 对话级别认证

对于会话级别密码术和会话级别认证， SNA 使用 数据加密标准 *(DES)* 算法。 DES 算法是一种块密码算法，它使用对称密钥对数据进行加密和解密。 块和键的长度均为 8 字节。

会话级别密码术

会话级别密码术 使用 DES 算法对会话数据进行加密和解密。 因此，它可用于在 SNA LU 6.2 通道上提供链路级别机密性服务。

逻辑单元 (LU) 可以提供必需 (或必需) 数据密码术，选择性数据密码术或无数据密码术。

在 必需加密会话上， LU 对所有出站数据请求单元进行加密，并对所有入站数据请求单元进行解密。

在 选择性加密会话上， LU 仅加密由发送事务程序 (TP) 指定的数据请求单元。 发送 LU 通过在请求头中设置指示符来表示数据已加密。 通过检查此指示符，接收 LU 可以在将哪些请求单元传递到接收 TP 之前告知这些单元要解密。

在 SNA 网络中， IBM MQ MCA 是事务程序。 MCA 不会请求对其发送的任何数据进行加密。 因此，选择性数据密码术不是一个选项; 只有强制性数据密码术或在会话上不可能进行数据密码术。

有关如何实现必需数据密码术的信息，请参阅 SNA 子系统的文档。 请参阅同一文档，以获取有关可在平台上使用的更强加密形式的信息，例如 z/OS 上的三重 DES 24 字节加密。

有关会话级密码术的更多常规信息，请参阅 系统网络体系结构 *LU 6.2 参考:* 对等协议， SC31-6808。

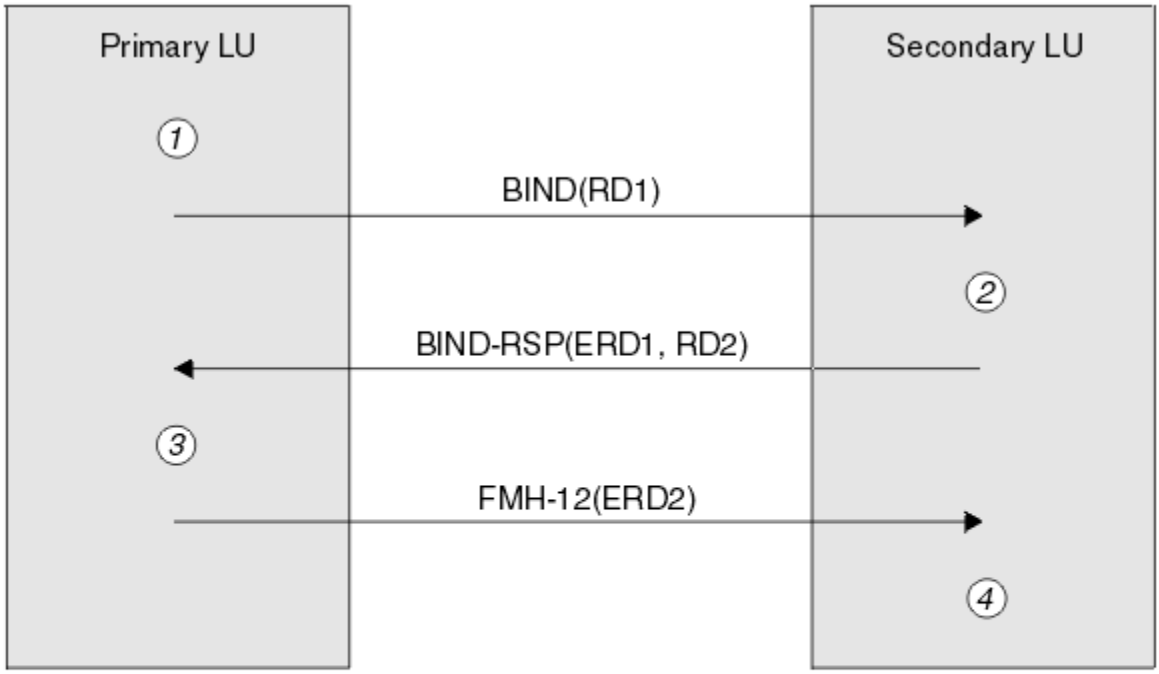
会话级别认证

会话级别认证 是一种会话级别安全协议，使两个 LU 能够在激活会话时相互认证。 也称为 *LU-LU* 验证。

由于 LU 实际上是从网络进入系统的 “网关” ，因此在某些情况下，您可能认为此认证级别已足够。 例如，如果您的队列管理器需要与在受控可信环境中运行的远程队列管理器交换消息，那么在对 LU 进行认证后，您可能已准备好信任远程系统的其余组件的身份。

会话级别认证由每个 LU 验证其合作伙伴的密码来实现。 该密码称为 *LU-LU* 密码 ，因为在每对 LU 之间建立了一个密码。 建立 LU-LU 密码的方式取决于实现，并且在 SNA 的作用域之外。

第 103 页的图 12 说明了用于会话级别认证的流。



Legend:

BIND	=	BIND request unit
BIND-RSP	=	BIND response unit
ERD	=	Encrypted random data
FMH-12	=	Function Management Header 12
RD	=	Random data

图 *12:* 用于会话级别认证的流

会话级别认证的协议如下所示。 过程中的数字与 第 103 页的图 12 中的数字相对应。

1. 主 LU 生成随机数据值 (RD1) 并将其发送到 BIND 请求中的辅助 LU。
2. 当辅助 LU 接收到包含随机数据的 BIND 请求时，它使用 DES 算法以其 LU-LU 密码副本作为密钥对数据进行加密。 然后，辅助 LU 生成第二个随机数据值 (RD2)，并将其与加密数据 (ERD1) 一起发送到 BIND 响应中的主 LU。
3. 当主 LU 接收到 BIND 响应时，它会从它最初生成的随机数据计算它自己的加密数据版本。 它通过使用 DES 算法及其 LU-LU 密码副本作为密钥来执行此操作。 然后，它将其版本与在 BIND 响应中接收到的加密数据进行比较。 如果这两个值相同，那么主 LU 知道辅助 LU 具有与其相同的密码，并且已认证辅助 LU。 如果这两个值不匹配，那么主 LU 将终止会话。

   然后，主 LU 将其在 BIND 响应中接收到的随机数据加密，并将加密数据 (ERD2) 发送到功能管理头 12 (FMH-12) 中的辅助 LU。
4. 当辅助 LU 接收到 FMH-12 时，它会从其生成的随机数据中计算其自己的加密数据版本。 然后，它将其版本与在 FMH-12 中接收到的加密数据进行比较。 如果两个值相同，那么将认证主 LU。 如果这两个值不匹配，那么辅助 LU 将终止会话。

在增强版本的协议中，通过使用其 LU-LU 密码副本作为密钥，辅助 LU 从 RD1， RD2 和辅助 LU 的标准名称计算 DES 消息认证代码 (MAC)，从而更好地防止中间攻击中的人员。 辅助 LU 将 MAC 发送到 BIND 响应中的主 LU，而不是 ERD1。

主 LU 通过计算其自己的 MAC 版本 (与 BIND 响应中接收的 MAC 进行比较) 来认证辅助 LU。 然后，主 LU 从 RD1 和 RD2 计算第二个 MAC，并将 MAC 发送到 FMH-12 中的辅助 LU，而不是 ERD2。

辅助 LU 通过计算其自己的第二个 MAC 版本 (与在 FMH-12 中接收的 MAC 进行比较) 来认证主 LU。

有关如何配置会话级别认证的信息，请参阅 SNA 子系统的文档。 有关会话级别认证的更多常规信息，请参阅 系统网络体系结构 *LU 6.2 参考:* 对等协议， SC31-6808。

对话级别认证

当本地 TP 尝试分配与伙伴 TP 的对话时，本地 LU 向伙伴 LU 发送连接请求，要求它连接伙伴 TP。 在某些情况下，连接请求可以包含安全信息，伙伴 LU 可以使用这些信息来认证本地 TP。 这称为 对话级别认证或 最终用户验证。

以下主题描述了 IBM MQ 如何为对话级别认证提供支持。

有关对话级别认证的更多信息，请参阅 系统网络体系结构 *LU 6.2 参考:* 对等协议， SC31-6808。

z/OS 有关特定于 z/OS 的信息，请参阅 z/OS MVS Planning: APPC/MVS Management。

有关 CPI-C 的更多信息，请参阅 使用 CPI 通信。

有关 APPC/MVS TP 对话可调用服务的更多信息，请参阅 APPC/MVS TP 对话可调用服务。

Multi *Multiplatforms* 版上对对话级别认证的支持

使用本主题可获取有关多平台上的对话级别认证工作方式的概述。

第 104 页的图 13 中说明了对 Multiplatforms 版上的对话级别认证的支持。 图中的数字与下面描述中的数字相对应。

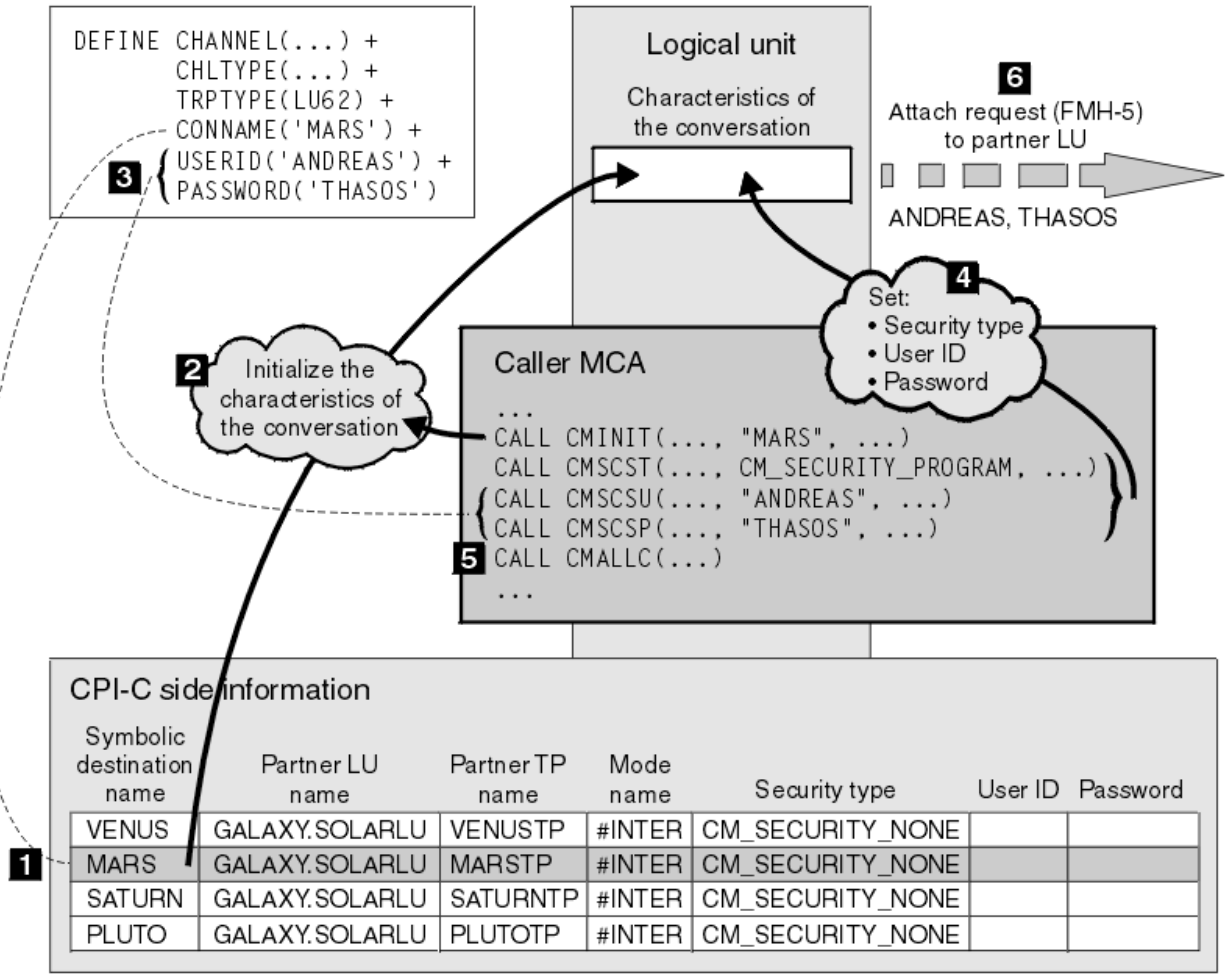


图 *13: IBM MQ* 支持对话级别认证

在多平台上， MCA 使用公共编程接口通信 (CPI-C) 调用通过 SNA 网络与合作伙伴 MCA 进行通信。 在通道调用者端的通道定义中， CONNAME 参数的值是符号目标名称，用于标识 CPI-C 端信息条目 (1)。 此条目指定:

- 伙伴 LU 的名称
- 伙伴 TP 的名称，它是响应者 MCA
- 要用于对话的方式的名称

辅助信息条目还可以指定以下安全信息:

- 安全性类型。

  通常实现的安全性类型为 CM_SECURITY_NONE， CM_SECURITY_PROGRAM 和 CM_SECURITY_SAME，但其他类型在 CPI-C 规范中定义。
- 用户标识。
- 密码。

调用者 MCA 准备通过发出 CPI-C 调用 CMINIT 来分配与响应者 MCA 的对话，使用 CONNAME 的值作为调用的其中一个参数。 为了本地 LU 的利益， CMINIT 调用标识 MCA 打算用于对话的辅助信息条目。 本地 LU 使用此条目中的值来初始化对话的特征 (2)。

然后，调用者 MCA 将检查通道定义 (3) 中 USERID 和 PASSWORD 参数的值。 如果设置了 USERID，那么调用者 MCA 将发出以下 CPI-C 调用 (4):

- CMSCST，用于将对话的安全性类型设置为 CM_SECURITY_PROGRAM。
- CMSCSU，用于将对话的用户标识设置为 USERID 的值。
- CMSCSP，用于将对话的密码设置为 PASSWORD 的值。 除非设置 PASSWORD，否则不会调用 CMSCSP。

这些调用设置的安全类型，用户标识和密码将覆盖先前从辅助信息条目获取的任何值。

然后，调用者 MCA 发出 CPI-C 调用 CMALLC 以分配对话 (5)。 作为对此调用的响应，本地 LU 向伙伴 LU (6) 发送连接请求 (功能管理头 5 或 FMH-5)。

如果伙伴 LU 将接受用户标识和密码，那么在连接请求中包含 USERID 和 PASSWORD 的值。 如果伙伴 LU 将不接受用户标识和密码，那么这些值不会包含在连接请求中。 本地 LU 发现当 LU 绑定以形成会话时，伙伴 LU 是否将接受用户标识和密码作为信息交换的一部分。

在更高版本的连接请求中，密码替换可以在 LU 之间流动，而不是在清除密码之间流动。 密码替代是由密码构成的 DES 消息认证代码 (MAC) 或 SHA-1 消息摘要。 仅当两个 LU 都支持密码替换时，才能使用密码替换。

当伙伴 LU 接收到包含用户标识和密码的入局连接请求时，它可能会将用户标识和密码用于标识和认证目的。 通过引用访问控制表，伙伴 LU 还可以确定用户标识是否有权分配对话并连接响应者 MCA。

此外，响应者 MCA 可能在连接请求中包含的用户标识下运行。 在这种情况下，用户标识将成为响应者 MCA 的缺省用户标识，并在 MCA 尝试连接到队列管理器时用于权限检查。 当 MCA 尝试访问队列管理器的资源时，它也可能用于后续的权限检查。

连接请求中的用户标识和密码可用于标识，认证和访问控制的方式取决于实现。 有关特定于 SNA 子系统的信息，请参阅相应的文档。

如果未设置 USERID，那么调用者 MCA 不会调用 CMSCST， CMSCSU 和 CMSCSP。 在这种情况下，连接请求中流的安全信息仅由辅助信息条目中指定的内容以及伙伴 LU 将接受的内容确定。

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.

队列管理器集群的安全性

虽然队列管理器集群可以方便地使用，但您必须特别注意其安全性。

队列管理器集群 是以某种方式逻辑关联的队列管理器网络。 作为集群成员的队列管理器称为 集群队列管理器。

可以使集群中的其他队列管理器知道属于集群队列管理器的队列。 这样的队列称为 集群队列。 集群中的任何队列管理器都可以将消息发送到集群队列，而不需要下列任何一项:

- 每个集群队列的显式远程队列定义
- 每个远程队列管理器之间的显式定义通道
- 每个出站通道的单独传输队列

您可以创建一个集群，在该集群中有两个或多个队列管理器是克隆的。 这意味着它们具有相同本地队列 (包括声明为集群队列的任何本地队列) 的实例，并且可以支持相同服务器应用程序的实例。

当连接到集群队列管理器的应用程序将消息发送到在每个克隆的队列管理器上具有实例的集群队列时， IBM MQ 决定要将其发送到哪个队列管理器。 当许多应用程序向集群队列发送消息时， IBM MQ 会在具有队列实例的每个队列管理器之间均衡工作负载。 如果其中一个托管克隆队列管理器的系统发生故障，那么 IBM MQ 将继续均衡其余队列管理器中的工作负载，直到重新启动失败的系统为止。

如果您正在使用队列管理器集群，那么需要考虑以下安全问题:

- 仅允许所选队列管理器将消息发送到队列管理器
- 仅允许远程队列管理器的所选用户将消息发送到队列管理器上的队列
- 允许连接到队列管理器的应用程序仅将消息发送到所选远程队列

即使您不使用集群，这些注意事项也很相关，但如果您使用集群，那么这些注意事项会变得更重要。

如果应用程序可以将消息发送到一个集群队列，那么它可以将消息发送到任何其他集群队列，而不需要其他远程队列定义，传输队列或通道。 因此，考虑是否需要限制对队列管理器上的集群队列的访问，以及限制应用程序可以向其发送消息的集群队列，这一点变得更为重要。

还有一些额外的安全注意事项，仅当您使用队列管理器集群时才相关:

- 仅允许所选队列管理器加入集群
- 强制不需要的队列管理器离开集群

有关所有这些注意事项的更多信息，请参阅 保持集群安全。 z/OS 有关特定于 IBM MQ for z/OS 的注意事项，请参阅 第 234 页的『Security in queue manager clusters on z/OS』。

相关任务

第 422 页的『阻止队列管理器接收消息』
您可以阻止集群队列管理器接收未经授权通过使用出口程序接收的消息。

IBM MQ 发布/预订的安全性

如果您正在使用 IBM MQ 发布/预订，那么还有其他安全注意事项。

在发布/预订系统中，有两种类型的应用程序: 发布者和订户。 发布者 以 IBM MQ 消息形式提供信息。 当发布者发布消息时，它指定 主题，该主题标识消息中信息的主题。

订户 是发布的信息的使用者。 订户通过预订主题来指定其感兴趣的主题。

队列管理器 是随 IBM MQ 发布/预订提供的应用程序。 它接收来自发布者的已发布消息和来自订户的预订请求，并将已发布的消息路由到订户。 订户仅在其预订的主题上发送消息。

有关更多信息，请参阅 发布/预订安全性。

多点广播安全性

使用此信息来了解 IBM MQ 多点广播可能需要安全进程的原因。

IBM MQ 多点广播没有内置安全性。 安全性检查在队列管理器中的 MQOPEN 时间处理， MQMD 字段设置由客户机处理。 网络中的某些应用程序可能不是 IBM MQ 应用程序 (例如， LLM 应用程序，请参阅 多点广播互操作性与 IBM MQ 低等待时间消息传递 以获取更多信息) ，因此您可能需要实施自己的安全过程，因为接收应用程序无法确定上下文字段的有效性。

有三个安全流程需要考虑:

访问控制

IBM MQ 中的访问控制基于用户标识。 有关此主题的更多信息，请参阅 第 87 页的『客户机的访问控制』。

保护网络安全

隔离网络可能是防止假消息的可行安全选项。 多点广播组地址上的应用程序可以使用本机通信功能发布恶意消息，这些功能与 MQ 消息无法区分，因为它们来自同一多点广播组地址上的应用程序。

多点广播组地址上的客户机也可以接收用于同一多点广播组地址上的其他客户机的消息。

隔离多点广播网络可确保只有有效的客户机和应用程序具有访问权。 这种安全预防措施可以防止恶意消息传入，以及机密信息传出。

有关多点广播组网络地址的信息，请参阅: 设置多点广播流量的相应网络

数字签名

通过对消息的表示进行加密来形成数字签名。 加密使用签署者的专用密钥，为了提高效率，通常对消息摘要而不是消息本身进行操作。 在 MQPUT 之前对消息进行数字签名是很好的安全预防措施，但如果有大量消息，此过程可能会对性能产生不利影响。

数字签名随要签名的数据不同而有所变化。 如果两个不同的消息由同一实体以数字方式进行签名，那么这两个签名不同，但这两个签名都可以使用相同的公用密钥 (即对消息进行签名的实体的公用密钥) 进行验证。

如本部分中先前所述，多点广播组地址上的应用程序可能使用本机通信功能发布恶意消息，这些功能与 MQ 消息无法区分。 数字签名提供了来源证明，只有发件人知道专用密钥，这就提供了强有力的证据证明发件人是消息的发件人。

有关此主题的更多信息，请参阅 第 10 页的『加密概念』。

防火墙和 IBM MQ Internet Pass-Thru

IBM MQ Internet Pass-Thru 可以简化通过防火墙的通信。

MQIPT 允许两个队列管理器交换消息，或者允许 IBM MQ 客户机应用程序连接到队列管理器，而不需要直接 TCP/IP 连接。 如果防火墙禁止两个系统之间的直接 TCP/IP 连接，那么此体系结构很有用。 使用 MQIPT 作为代理可使 IBM MQ 通道数据通过防火墙更简单且更易于管理。 MQIPT 还可以使用传输层安全性 (TLS) 和 HTTP 中的隧道 IBM MQ 数据来保护通过因特网发送的 IBM MQ 数据。

有关更多信息，请参阅 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 160.

Refer to other sections for details, in particular “Profiles used to control access to IBM MQ resources” on page 170.

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 165.
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 226.
   - **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 226.
   - **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 “通道认证记录” on page 43.
    - You can also determine the appropriate MCAUSER attribute value for each channel, or provide suitable channel security exits.
14. Do you want to use Transport Layer Security (TLS)?
    - **Yes**: To specify that any user presenting an TLS personal certificate containing a specified DN is to use a specific MCAUSER, set a channel authentication record of type SSLPEERMAP. You can specify a single distinguished name or a pattern including wildcards.

- Plan your TLS infrastructure. Install the System SSL feature of z/OS. In RACF, set up your certificate name filters (CNFs), if you are using them, and your digital certificates. Set up your SSL key ring. Ensure that the SSLKEYR queue manager attribute is nonblank and points to your SSL key ring. Also ensure that the value of SSLTASKS is at least 2.
- **No**: Ensure that SSLKEYR is blank, and SSLTASKS is zero.

For further details about TLS, see “IBM MQ 中的 TLS 安全协议” on page 21.

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 234

设置安全性

此主题集合包含特定于不同操作系统和客户机使用的信息。

ALW 在 AIX, Linux, and Windows 上设置安全性

特定于 AIX, Linux, and Windows 系统的安全注意事项。

IBM MQ 队列管理器传输可能有价值的信息，因此您需要使用权限系统来确保未经授权的用户无法访问您的队列管理器。 请考虑以下类型的安全控制:

谁可以管理 IBM MQ
您可以定义一组可以发出命令来管理 IBM MQ 的用户。

谁可以使用 IBM MQ 对象
您可以定义哪些用户 (通常是应用程序) 可以使用 MQI 调用和 PCF 命令来执行以下操作:

- 可以连接到队列管理器的人员。
- 可以访问对象 (队列，进程定义，名称列表，通道，客户机连接通道，侦听器，服务和认证信息对象) 的人员，以及他们对这些对象的访问类型。
- 可以访问 IBM MQ 消息的人员。
- 谁可以访问与消息关联的上下文信息。

通道安全性
您需要确保用于向远程系统发送消息的通道可以访问所需的资源。

您可以使用标准操作工具来授予对程序库， MQI 链接库和命令的访问权。 但是，包含队列和其他队列管理器数据的目录是 IBM MQ 的专用目录; 请勿使用标准操作系统命令来授予或撤销对 MQI 资源的权限。

ALW 授权如何在 AIX, Linux, and Windows 上工作

本部分主题中的授权规范表精确定义了授权的工作方式以及适用的限制。

这些表适用于以下情况:

- 发出 MQI 调用的应用程序

- 发出 MQSC 命令作为转义 PCF 的管理程序
- 发出 PCF 命令的管理程序

在此部分中，信息显示为一组指定了以下内容的表:

要执行的操作
MQI 选项， MQSC 命令或 PCF 命令。

访问控制对象
队列，进程，队列管理器，名称列表，认证信息，通道，客户机连接通道，侦听器或服务。

需要授权
表示为 MQZAO_ 常量。

在这些表中，以 MQZAO_ 为前缀的常量对应于特定实体的 `setmqaut` 命令的权限列表中的关键字。 例如，MQZAO_BROWSE 对应于关键字 `+browse`， MQZAO_SET_ALL_CONTEXT 对应于关键字 `+setall`，依此类推。 这些常量在产品随附的头文件 cmqzc.h 中定义。

ALW *MQI* 调用的授权

MQCONN， **MQOPEN**， **MQPUT1** 和 **MQCLOSE** 可能需要授权检查。 本主题中的表汇总了每个调用所需的权限。

仅当应用程序运行时所使用的用户标识 (或其能够假定的授权) 已被授予相关授权时，才允许应用程序发出特定的 MQI 调用和选项。

四个 MQI 调用可能需要授权检查: **MQCONN**， **MQOPEN**， **MQPUT1** 和 **MQCLOSE**。

对于 **MQOPEN** 和 **MQPUT1**，将对要打开的对象的名称进行权限检查，而不是对名称进行权限检查，从而在解析名称之后生成权限检查。 例如，可以授予应用程序打开别名队列的权限，而不具有打开别名所解析到的基本队列的权限。 该规则是，除非直接打开队列管理器别名定义，否则将对解析非队列管理器别名的名称过程中迂到的第一个定义执行检查; 即，其名称显示在对象描述符的 *ObjectName* 字段中。 打开的对象始终需要权限。 在某些情况下，需要通过队列管理器对象的授权获取其他独立于队列的权限。

第 111 页的表 10， 第 111 页的表 11， 第 112 页的表 12 和 第 113 页的表 13 汇总了每个调用所需的权限。 在表中， 不适用 表示授权检查与此操作无关; 不检查 表示不执行授权检查。

注: 在这些表中没有提到名称列表，通道，客户机连接通道，侦听器，服务或认证信息对象。 这是因为除了 MQOO_INQUIRE 之外，没有任何权限适用于这些对象，而 MQOO_INQUIRE 的权限与其他对象的权限相同。

特殊授权 MQZAO_ALL_MQI 包含表中与对象类型相关的所有授权，但 MQZAO_DELETE 和 MQZAO_DISPLAY 除外，它们被归类为管理授权。

要修改任何消息上下文选项，必须具有相应权限来发出调用。 例如，要使用 MQOO_SET_IDENTITY_CONTEXT 或 MQPMO_SET_IDENTITY_CONTEXT，必须具有 `+setid` 许可权。

表 *10: MQCONN* 调用所需的安全性授权

需要授权:	**队列对象 (第 113 页的『1』)**	**进程对象**	**队列管理器对象**
MQCONN	不适用	不适用	MQZAO_CONNECT

表 *11: MQOPEN* 调用所需的安全性授权

需要授权:	**队列对象 (第 113 页的『1』)**	**进程对象**	**队列管理器对象**
MQOO_INQUIRE	MQZAO_INQUIRE	MQZAO_INQUIRE	MQZAO_INQUIRE
MQOO_BROWSE	MQZAO_BROWSE	不适用	不检查
MQOO_INPUT_ *	MQZAO_INPUT	不适用	不检查

表 *11: MQOPEN* 调用所需的安全性授权 *(继续)*			
需要授权:	**队列对象 (第 113 页的『1』)**	**进程对象**	**队列管理器对象**
MQOO_SAVE_ALL_CONTEXT **(第 113 页的『2』)**	MQZAO_INPUT	不适用	不适用
MQOO_OUTPUT (正常队列) **(第 113 页的『3』)**	MQZAO_OUTPUT	不适用	不适用
MQOO_PASS_IDENTITY_CONTEXT **(第 113 页的『4』)**	MQZAO_PASS_IDENTITY_CONTEXT	不适用	不检查
MQOO_PASS_ALL_ 上下文 **(第 113 页的『4』， 第 113 页的『5』)**	MQZAO_PASS_ALL_CONTEXT	不适用	不检查
MQOO_SET_IDENTITY_CONTEXT **(第 113 页的『4』， 第 113 页的『5』)**	MQZAO_SET_IDENTITY_CONTEXT	不适用	MQZAO_SET_IDENTITY_CONTEXT **(第 113 页的『6』)**
MQOO_SET_ALL_CONTEXT **(第 113 页的『4』， 第 113 页的『7』)**	MQZAO_SET_ALL_CONTEXT	不适用	MQZAO_SET_ALL_CONTEXT **(第 113 页的『6』)**
MQOO_OUTPUT (传输队列) **(第 113 页的『8』)**	MQZAO_SET_ALL_CONTEXT	不适用	MQZAO_SET_ALL_CONTEXT **(第 113 页的『6』)**
MQOO_SET	MQZAO_SET	不适用	不检查
MQOO_ALTERNATE_USER_AUTHORITY	**(第 113 页的『9』)**	**(第 113 页的『9』)**	MQZAO_ALTERNATE_USER_AUTHORITY **(第 113 页的『9』， 第 113 页的『10』)**

表 *12: MQPUT1* 调用所需的安全性授权			
需要授权:	**队列对象 (第 113 页的『1』)**	**进程对象**	**队列管理器对象**
MQPMO_PASS_IDENTITY_CONTEXT	MQZAO_PASS_IDENTITY_CONTEXT **(第 113 页的『11』)**	不适用	不检查
MQPMO_PASS_ALL_CONTEXT	MQZAO_PASS_ALL_CONTEXT **(第 113 页的『11』)**	不适用	不检查
MQPMO_SET_IDENTITY_CONTEXT	MQZAO_SET_IDENTITY_CONTEXT **(第 113 页的『11』)**	不适用	MQZAO_SET_IDENTITY_CONTEXT **(第 113 页的『6』)**
MQPMO_SET_ALL_CONTEXT	MQZAO_SET_ALL_CONTEXT **(第 113 页的『11』)**	不适用	MQZAO_SET_ALL_CONTEXT **(第 113 页的『6』)**

表 *12: MQPUT1* 调用所需的安全性授权 *(继续)*			
需要授权:	**队列对象 (第 113 页的『1』)**	**进程对象**	**队列管理器对象**
(传输队列) **(第 113 页的『8』)**	MQZAO_SET_ ALL_CONTEXT	不适用	MQZAO_SET_ ALL_CONTEXT **(第 113 页的『6』)**
MQPMO_ALTERNATE_ USER_AUTHORITY	**(第 113 页的『12』)**	不适用	MQZAO_ALTERNATE_ USER_AUTHORITY **(第 113 页的『10』)**

表 *13: MQCLOSE* 调用所需的安全性授权			
需要授权:	**队列对象 (第 113 页的『1』)**	**进程对象**	**队列管理器对象**
MQCO_DELETE	MQZAO_DELETE **(第 113 页的『13』)**	不适用	不适用
MQCO_DELETE _PURGE	MQZAO_DELETE **(第 113 页的『13』)**	不适用	不适用

表的注释:

1. 如果打开模型队列:
   - 除了针对要打开的访问类型打开模型队列的权限外，模型队列还需要 MQZAO_DISPLAY 权限。
   - 创建动态队列不需要 MQZAO_CREATE 权限。
   - 用于打开模型队列的用户标识将自动授予所创建动态队列的所有特定于队列的权限 (相当于 MQZAO_ALL)。
2. 还必须指定 MQOO_INPUT_ *。 这对于本地队列，模型队列或别名队列有效。
3. 将对除传输队列以外的所有输出案例执行此检查 (请参阅注释 第 113 页的『8』)。
4. 还必须指定 MQOO_OUTPUT。
5. 此选项还隐含 MQOO_PASS_IDENTITY_CONTEXT。
6. 队列管理器对象和特定队列都需要此权限。
7. 此选项还包含 MQOO_PASS_IDENTITY_CONTEXT， MQOO_PASS_ALL_CONTEXT 和 MQOO_SET_IDENTITY_CONTEXT。
8. 将对具有 *Usage* 队列属性 MQUS_TRANSMISSION 的本地或模型队列执行此检查，并且将直接打开该队列以进行输出。 如果正在打开远程队列 (通过指定远程队列管理器和远程队列的名称，或者通过指定远程队列的本地定义的名称)，那么它不适用。
9. 还必须至少指定 MQOO_INQUIRE (对于任何对象类型)， MQOO_BROWSE， MQOO_INPUT_ *， MQOO_OUTPUT 或 MQOO_SET (对于队列) 中的一个。 执行的检查与指定的其他选项一样，将提供的备用用户标识用于特定指定的对象权限，并将当前应用程序权限用于 MQZAO_ALTERNATE_USER_IDENTIFIER 检查。
10. 此授权允许指定任何 *AlternateUserId* 。
11. 如果队列没有 *Usage* 队列属性 MQUS_TRANSMISSION，那么还会执行 MQZAO_OUTPUT 检查。
12. 执行的检查与指定的其他选项一样，将提供的备用用户标识用于特定指定的队列权限，并将当前应用程序权限用于 MQZAO_ALTERNATE_USER_IDENTIFIER 检查。
13. 仅当以下两个语句都为 true 时，才会执行此检查:
    - 正在关闭并删除永久动态队列。
    - 队列不是由返回所使用对象句柄的 MQOPEN 调用创建的。

    否则，将不进行检查。

ALW 对脱离 *PCF* 的 *MQSC* 命令的授权

此信息汇总了 Escape PCF 中包含的每个 MQSC 命令所需的权限。

不适用 表示此操作与此对象类型无关。

用于运行提交命令的程序的用户标识还必须具有以下权限:

- 队列管理器的 MQZAO_CONNECT 权限
- 队列管理器上的 MQZAO_DISPLAY 权限，以便执行 PCF 命令
- 在 Escape PCF 命令的文本中发出 MQSC 命令的权限

ALTER 对象

Object	需要授权
队列	MQZAO_CHANGE
Topic	MQZAO_CHANGE
进程	MQZAO_CHANGE
队列管理器	MQZAO_CHANGE
名称列表	MQZAO_CHANGE
认证信息	MQZAO_CHANGE
通道	MQZAO_CHANGE
客户机连接通道	MQZAO_CHANGE
侦听器	MQZAO_CHANGE
服务	MQZAO_CHANGE
通信信息	MQZAO_CHANGE

CLEAR 对象

Object	需要授权
队列	MQZAO_CLEAR
Topic	MQZAO_CLEAR
进程	不适用
队列管理器	不适用
名称列表	不适用
认证信息	不适用
通道	不适用
客户机连接通道	不适用
侦听器	不适用
服务	不适用
通信信息	不适用

DEFINE 对象 NOREPLACE (第 118 页的『1』)

Object	需要授权
队列	MQZAO_CREATE (第 118 页的『2』)

Object	需要授权
Topic	MQZAO_CREATE (第 118 页的『2』)
进程	MQZAO_CREATE (第 118 页的『2』)
队列管理器	不适用
名称列表	MQZAO_CREATE (第 118 页的『2』)
认证信息	MQZAO_CREATE (第 118 页的『2』)
通道	MQZAO_CREATE (第 118 页的『2』)
客户机连接通道	MQZAO_CREATE (第 118 页的『2』)
侦听器	MQZAO_CREATE (第 118 页的『2』)
服务	MQZAO_CREATE (第 118 页的『2』)
通信信息	MQZAO_CREATE (第 118 页的『2』)

DEFINE 对象 REPLACE (第 118 页的『1』， 第 118 页的『3』)

Object	需要授权
队列	MQZAO_CHANGE
Topic	MQZAO_CHANGE
进程	MQZAO_CHANGE
队列管理器	不适用
名称列表	MQZAO_CHANGE
认证信息	MQZAO_CHANGE
通道	MQZAO_CHANGE
客户机连接通道	MQZAO_CHANGE
侦听器	MQZAO_CHANGE
服务	MQZAO_CHANGE
通信信息	MQZAO_CHANGE

DELETE 对象

Object	需要授权
队列	MQZAO_DELETE
Topic	MQZAO_DELETE
进程	MQZAO_DELETE
队列管理器	不适用
名称列表	MQZAO_DELETE
认证信息	MQZAO_DELETE
通道	MQZAO_DELETE
客户机连接通道	MQZAO_DELETE
侦听器	MQZAO_DELETE

Object	需要授权
服务	MQZAO_DELETE
通信信息	MQZAO_DELETE

DISPLAY 对象

Object	需要授权
队列	MQZAO_DISPLAY
Topic	MQZAO_DISPLAY
进程	MQZAO_DISPLAY
队列管理器	MQZAO_DISPLAY
名称列表	MQZAO_DISPLAY
认证信息	MQZAO_DISPLAY
通道	MQZAO_DISPLAY
客户机连接通道	MQZAO_DISPLAY
侦听器	MQZAO_DISPLAY
服务	MQZAO_DISPLAY
通信信息	MQZAO_DISPLAY

START 对象

Object	需要授权
队列	不适用
Topic	不适用
进程	不适用
队列管理器	不适用
名称列表	不适用
认证信息	不适用
通道	MQZAO_CONTROL
客户机连接通道	不适用
侦听器	MQZAO_CONTROL
服务	MQZAO_CONTROL
通信信息	不适用

STOP 对象

Object	需要授权
队列	不适用
Topic	不适用
进程	不适用
队列管理器	不适用

Object	需要授权
名称列表	不适用
认证信息	不适用
通道	MQZAO_CONTROL
客户机连接通道	不适用
侦听器	MQZAO_CONTROL
服务	MQZAO_CONTROL
通信信息	不适用

通道命令

命令	Object	需要授权
Ping 通道	通道	MQZAO_CONTROL
重置通道	通道	MQZAO_CONTROL_EXTENDED
解析通道	通道	MQZAO_CONTROL_EXTENDED

预订命令

命令	Object	需要授权
变更 SUB	Topic	MQZAO_CONTROL
DEFINE SUB	Topic	MQZAO_CONTROL
删除 SUB	Topic	MQZAO_CONTROL
显示子项	Topic	MQZAO_DISPLAY

安全性命令

命令	Object	需要授权
SET AUTHREC	队列管理器	MQZAO_CHANGE
删除 AUTHREC	队列管理器	MQZAO_CHANGE
显示 AUTHREC	队列管理器	MQZAO_DISPLAY
显示 AUTHSERV	队列管理器	MQZAO_DISPLAY
显示 ENTAUTH	队列管理器	MQZAO_DISPLAY
SET CHLAUTH	队列管理器	MQZAO_CHANGE
显示 CHLAUTH	队列管理器	MQZAO_DISPLAY
REFRESH SECURITY	队列管理器	MQZAO_CHANGE

状态显示

命令	Object	需要授权
DISPLAY CHSTATUS	队列管理器	MQZAO_DISPLAY 请注意，如果通道类型为 CLUSSDR，那么传输队列上需要 +inq 权限 (相当于 MQZAO_INQUIRE)。
显示 lsstatus	队列管理器	MQZAO_DISPLAY
显示发布预订	队列管理器	MQZAO_DISPLAY
显示 SBSTATUS	队列管理器	MQZAO_DISPLAY
显示 SVSTATUS	队列管理器	MQZAO_DISPLAY
DISPLAY TPSTATUS	队列管理器	MQZAO_DISPLAY

集群命令

命令	Object	需要授权
DISPLAY CLUSQMGR	队列管理器	MQZAO_DISPLAY
刷新集群	需要 "mqm" 组成员资格	
Reset Cluster	需要 "mqm" 组成员资格	
已暂挂的队列管理器	需要 "mqm" 组成员资格	
恢复队列管理器	需要 "mqm" 组成员资格	

其他管理命令

命令	Object	需要授权
PING QMGR	队列管理器	MQZAO_DISPLAY
刷新队列管理器	队列管理器	MQZAO_CHANGE
重置队列管理器	队列管理器	MQZAO_CHANGE
DISPLAY CONN	队列管理器	MQZAO_DISPLAY
STOP CONN	队列管理器	MQZAO_CHANGE

注:

1. 对于 DEFINE 命令，如果指定了 LIKE 对象，那么也需要 MQZAO_DISPLAY 权限，或者在相应的 SYSTEM.DEFAULT.xxx 对象 (如果省略了 LIKE)。
2. MQZAO_CREATE 权限并非特定于特定对象或对象类型。 通过在 `setmqaut` 命令中指定对象类型 QMGR，授予对指定队列管理器的所有对象的创建权限。
3. 如果要替换的对象已存在，那么这将适用。 如果不存在，那么检查与 DEFINE 对象 NOREPLACE 相同。

相关信息

集群：使用 REFRESH CLUSTER 最佳实践

ALW *PCF* 命令的权限

本部分概述了每个 PCF 命令所需的权限。

无检查 表示不执行授权检查; 不适用 表示此操作与此对象类型无关。

用于运行提交命令的程序的用户标识还必须具有以下权限:

- 队列管理器的 MQZAO_CONNECT 权限
- 队列管理器上的 MQZAO_DISPLAY 权限，以便执行 PCF 命令

特殊授权 MQZAO_ALL_ADMIN 包含以下列表中与对象类型相关的所有权限，但 MQZAO_CREATE 除外，它并非特定于特定对象或对象类型。

更改 对象

Object	**需要授权**
队列	MQZAO_CHANGE
主题	MQZAO_CHANGE
流程	MQZAO_CHANGE
队列管理器	MQZAO_CHANGE
名称列表	MQZAO_CHANGE
认证信息	MQZAO_CHANGE
CHANNEL	MQZAO_CHANGE
客户机连接通道	MQZAO_CHANGE
侦听器	MQZAO_CHANGE
服务	MQZAO_CHANGE
通信信息	MQZAO_CHANGE

清除 对象

Object	**需要授权**
队列	MQZAO_CLEAR
主题	MQZAO_CLEAR
进程	不适用
队列管理器	不适用
名称列表	不适用
认证信息	不适用
通道	不适用
客户机连接通道	不适用
侦听器	不适用
服务	不适用
通信信息	不适用

复制 对象 (不替换) (1)

Object	**需要授权**
队列	MQZAO_CREATE **(2)**
主题	MQZAO_CREATE **(2)**
流程	MQZAO_CREATE **(2)**
队列管理器	不适用

Object	需要授权
名称列表	MQZAO_CREATE (2)
认证信息	MQZAO_CREATE (2)
CHANNEL	MQZAO_CREATE (2)
客户机连接通道	MQZAO_CREATE (2)
侦听器	MQZAO_CREATE (2)
服务	MQZAO_CREATE (2)
通信信息	MQZAO_CREATE (第 124 页的『2』)

复制 对象 (带有替换) (1, 4)

Object	需要授权
队列	MQZAO_CHANGE
主题	MQZAO_CHANGE
流程	MQZAO_CHANGE
队列管理器	不适用
名称列表	MQZAO_CHANGE
认证信息	MQZAO_CHANGE
CHANNEL	MQZAO_CHANGE
客户机连接通道	MQZAO_CHANGE
侦听器	MQZAO_CHANGE
服务	MQZAO_CHANGE
通信信息	MQZAO_CHANGE

创建 对象 (不替换) (3)

Object	需要授权
队列	MQZAO_CREATE (2)
主题	MQZAO_CREATE (2)
流程	MQZAO_CREATE (2)
队列管理器	不适用
名称列表	MQZAO_CREATE (2)
认证信息	MQZAO_CREATE (2)
CHANNEL	MQZAO_CREATE (2)
客户机连接通道	MQZAO_CREATE (2)
侦听器	MQZAO_CREATE (2)
服务	MQZAO_CREATE (2)
通信信息	MQZAO_CREATE (2)

创建 对象 (带有替换) (3, 4)

Object	需要授权
队列	MQZAO_CHANGE
主题	MQZAO_CHANGE
流程	MQZAO_CHANGE
队列管理器	不适用
名称列表	MQZAO_CHANGE
认证信息	MQZAO_CHANGE
CHANNEL	MQZAO_CHANGE
客户机连接通道	MQZAO_CHANGE
侦听器	MQZAO_CHANGE
服务	MQZAO_CHANGE
通信信息	MQZAO_CHANGE

删除 对象

Object	需要授权
队列	MQZAO_DELETE
主题	MQZAO_DELETE
流程	MQZAO_DELETE
队列管理器	不适用
名称列表	MQZAO_DELETE
认证信息	MQZAO_DELETE
CHANNEL	MQZAO_DELETE
客户机连接通道	MQZAO_DELETE
侦听器	MQZAO_DELETE
服务	MQZAO_DELETE
通信信息	MQZAO_DELETE

查询 对象

Object	需要授权
队列	MQZAO_DISPLAY
主题	MQZAO_DISPLAY
流程	MQZAO_DISPLAY
队列管理器	MQZAO_DISPLAY
名称列表	MQZAO_DISPLAY
认证信息	MQZAO_DISPLAY
CHANNEL	MQZAO_DISPLAY

Object	需要授权
客户机连接通道	MQZAO_DISPLAY
侦听器	MQZAO_DISPLAY
服务	MQZAO_DISPLAY
通信信息	MQZAO_DISPLAY

查询 *object* 名称

Object	需要授权
队列	不检查
Topic	不检查
进程	不检查
队列管理器	不检查
名称列表	不检查
认证信息	不检查
通道	不检查
客户机连接通道	不检查
侦听器	不检查
服务	不检查
通信信息	不检查

启动 对象

Object	需要授权
队列	不适用
Topic	不适用
进程	不适用
队列管理器	不适用
名称列表	不适用
认证信息	不适用
CHANNEL	MQZAO_CONTROL
客户机连接通道	不适用
侦听器	MQZAO_CONTROL
服务	MQZAO_CONTROL
通信信息	不适用

停止 对象

Object	需要授权
队列	不适用
Topic	不适用

Object	需要授权
进程	不适用
队列管理器	不适用
名称列表	不适用
认证信息	不适用
CHANNEL	MQZAO_CONTROL
客户机连接通道	不适用
侦听器	MQZAO_CONTROL
服务	MQZAO_CONTROL
通信信息	不适用

通道命令

命令	Object	需要授权
Ping 通道	通道	MQZAO_CONTROL
重置通道	通道	MQZAO_CONTROL_EXTENDED
解析通道	通道	MQZAO_CONTROL_EXTENDED

预订命令

命令	Object	需要授权
更改预订	Topic	MQZAO_CONTROL
创建预订	Topic	MQZAO_CONTROL
删除预订	Topic	MQZAO_CONTROL
查询预订	Topic	MQZAO_DISPLAY

安全性命令

命令	Object	需要授权
设置权限记录	队列管理器	MQZAO_CHANGE
删除权限记录	队列管理器	MQZAO_CHANGE
查询权限记录	队列管理器	MQZAO_DISPLAY
查询权限服务	队列管理器	MQZAO_DISPLAY
查询实体权限	队列管理器	MQZAO_DISPLAY
设置通道认证记录	队列管理器	MQZAO_CHANGE
查询通道认证记录	队列管理器	MQZAO_DISPLAY
刷新安全性	队列管理器	MQZAO_CHANGE

状态显示

命令	Object	需要授权
查询通道状态	队列管理器	MQZAO_DISPLAY 请注意，如果通道类型为 CLUSSDR，那么传输队列上需要 +inq 权限 (相当于 MQZAO_INQUIRE)。
查询通道侦听器状态	队列管理器	MQZAO_DISPLAY
查询发布/预订状态	队列管理器	MQZAO_DISPLAY
查询预订状态	队列管理器	MQZAO_DISPLAY
查询服务状态	队列管理器	MQZAO_DISPLAY
查询主题状态	队列管理器	MQZAO_DISPLAY

集群命令

命令	Object	需要授权
查询集群队列管理器	队列管理器	MQZAO_DISPLAY
刷新集群	需要 "mqm" 组成员资格	需要 "mqm" 组成员资格
Reset Cluster	需要 "mqm" 组成员资格	需要 "mqm" 组成员资格
暂挂队列管理器集群	需要 "mqm" 组成员资格	需要 "mqm" 组成员资格
恢复队列管理器集群	需要 "mqm" 组成员资格	需要 "mqm" 组成员资格

其他管理命令

命令	Object	需要授权
Ping 队列管理器	队列管理器	MQZAO_DISPLAY
刷新队列管理器	队列管理器	MQZAO_CHANGE
重置队列管理器	队列管理器	MQZAO_CHANGE
重置队列统计信息	队列	MQZAO_DISPLAY 和 MQZAO_CHANGE
查询连接	队列管理器	MQZAO_DISPLAY
停止连接	队列管理器	MQZAO_CHANGE

注:

1. 对于 "复制" 命令， "源" 对象还需要 MQZAO_DISPLAY 权限。
2. MQZAO_CREATE 权限并非特定于特定对象或对象类型。 通过在 `setmqaut` 命令中指定对象类型 QMGR，授予对指定队列管理器的所有对象的创建权限。
3. 对于 "创建" 命令，相应的 SYSTEM.DEFAULT.* 对象。
4. 如果要替换的对象已存在，那么这将适用。 如果不存在，那么此检查适用于 "复制" 或 "创建" 而不进行替换。

AIX 在 AIX 上创建和管理组

在 AIX 上，如果您未使用 NIS 或 NIS +，请使用 SMITTY 来处理组。

关于此任务

在 AIX 上，可以使用 SMITTY 来创建组，向组添加用户，显示组中用户的列表以及从组中除去用户。

过程

1. 从 SMITTY 中，选择 **安全性和用户** ，然后按 Enter 键。
2. 选择 **组** ，然后按 Enter 键。
3. 要创建组，请完成以下步骤:
   a) 选择 **添加组** ，然后按 Enter 键。
   b) 输入组的名称以及要添加到组的任何用户的名称 (以逗号分隔)。
   c) 按 Enter 键以创建组。
4. 要将用户添加到组，请完成以下步骤:
   a) 选择 **更改/显示组的特征** ，然后按 Enter 键。
   b) 输入组的名称以显示组的成员列表。
   c) 将要添加的用户的名称添加到组中，以逗号分隔。
   d) 按 Enter 键以将名称添加到组。
5. 要显示组中的人员，请完成以下步骤:
   a) 选择 **更改/显示组的特征** ，然后按 Enter 键。
   b) 输入组的名称以显示组的成员列表。
6. 要从组中除去用户，请完成以下步骤:
   a) 选择 **更改/显示组的特征** ，然后按 Enter 键。
   b) 输入组的名称以显示组的成员列表。
   c) 删除要从组中除去的用户的名称。
   d) 按 Enter 键以从组中除去该名称。

Linux 在 Linux 上创建和管理组

在 Linux 上，如果您未使用 NIS 或 NIS + ，请使用 `/etc/group` 文件来处理组。

关于此任务

在 Linux 上，组信息保存在 `/etc/group` 文件中。 您可以使用命令来创建组，向组添加用户，显示组中的用户列表以及从组中除去用户。

过程

1. 要创建新组，请使用 **groupadd** 命令。
   输入以下命令:

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

   其中 *group-ID* 是组的数字标识， *group-name* 是组的名称。
2. 要向补充组添加成员，请使用 **usermod** 命令列出用户当前是其成员的补充组以及用户要成为其成员的补充组。
   例如，如果用户已经是组 `groupa` 的成员，并且要成为 `groupb` 的成员，请使用以下命令:

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

   其中 *user-name* 是用户名。
3. 要显示属于组的人员，请使用 **getent** 命令。
   输入以下命令:

```
getent group group-name
```

其中 *group-name* 是组的名称。

4. 要从补充组中除去成员，请使用 **usermod** 命令列出您希望用户保留其成员的补充组。
例如，如果用户的主组为 `users`，并且该用户也是组 `mqm`， `groupa` 和 `groupb` 的成员，那么要从 `mqm` 组中除去该用户，请使用以下命令:

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

其中 *user-name* 是用户名。

Windows 在 Windows 上创建和管理组

在 Windows 上，使用 "计算机管理" 功能部件来管理工作站或成员服务器上的组。

关于此任务

对于域控制器，用户和组通过 Active Directory 进行管理。 有关使用 Active Directory 的更多详细信息，请参阅相应的操作系统指示信息。

在重新启动队列管理器或发出 MQSC 命令 **REFRESH SECURITY** (或 PCF 等效命令) 之前，不会识别您对主体组成员资格所作的任何更改。

使用 " Windows 计算机管理" 面板来处理用户和组。 在用户再次登录之前，对当前登录用户所作的任何更改都可能不会生效。

Windows 在 *Windows* 上创建组

使用控制面板创建组。

过程

1. 打开控制面板
2. 双击 **管理工具**。
"管理工具" 面板将打开。
3. 双击 **计算机管理**。
此时将打开 "计算机管理" 面板。
4. 展开**本地用户和组**。
5. 右键单击 **组**，然后选择 **新建组 ...**。
这样会显示 "新建组" 面板。
6. 在 "组名" 字段中输入相应的名称，然后单击 **创建**。
7. 单击**关闭**。

Windows 将用户添加到 *Windows* 上的组

使用控制面板将用户添加到组中。

过程

1. 打开控制面板
2. 双击 **管理工具**。
"管理工具" 面板将打开。
3. 双击 **计算机管理**。
此时将打开 "计算机管理" 面板。
4. 从 "计算机管理" 面板中，展开 **本地用户和组**。
5. 选择 **用户**

6. 双击要添加到组的用户。
   这样会显示 "用户属性" 面板。
7. 选择 **成员** 选项卡。
8. 选择要将用户添加到的组。 如果您想要的组不可见:
   a) 单击**添加...**。
      这样会显示 "选择组" 面板。
   b) 单击 **位置 ...**。
      这样会显示 "位置" 面板。
   c) 从列表中选择要将用户添加到的组的位置，然后单击 **确定**。
   d) 在提供的字段中输入组名。

      或者，单击 **高级 ...** 然后 **立即查找** 以列出当前所选位置中可用的组。 从此处，选择要将用户添加到的组，然后单击 **确定**。
   e) 单击**确定**。
      这样会显示用户属性面板，显示您添加的组。
   f) 选择组。
9. 单击**确定**。
   此时将显示 "计算机管理" 面板。

Windows 显示 *Windows* 上的组中的人员

使用控制面板显示组的成员。

过程

1. 打开控制面板
2. 双击 **管理工具**。
   "管理工具" 面板将打开。
3. 双击 **计算机管理**。
   此时将打开 "计算机管理" 面板。
4. 从 "计算机管理" 面板中，展开 **本地用户和组**。
5. 选择**组**。
6. 双击组。 这样会显示 "组属性" 面板。
   这样会显示 "组属性" 面板。

结果

这样会显示组成员。

Windows 从 *Windows* 上的组中除去用户

使用控制面板从组中除去用户。

过程

1. 打开控制面板
2. 双击 **管理工具**。
   "管理工具" 面板将打开。
3. 双击 **计算机管理**。
   此时将打开 "计算机管理" 面板。
4. 从 "计算机管理" 面板中，展开 **本地用户和组**。
5. 选择**用户**。

6. 双击要添加到组的用户。
   这样会显示 "用户属性" 面板。
7. 选择 **成员** 选项卡。
8. 选择要从中除去用户的组，然后单击 **除去**。
9. 单击**确定**。
   此时将显示 "计算机管理" 面板。

结果
您现在已从组中除去该用户。

Windows Windows 上安全性的特殊注意事项

某些安全功能在不同版本的 Windows 上的行为不同。

IBM MQ 安全性依赖于对操作系统 API 的调用，以获取有关用户权限和组成员资格的信息。 某些函数在 Windows 系统上的行为不相同。 此主题集合包含有关在 Windows 环境中运行 IBM MQ 时这些差异可能如何影响 IBM MQ 安全性的描述。

Windows *IBM MQ Windows* 服务的本地和域用户帐户

在运行 IBM MQ 时，必须检查是否只有授权用户才能访问队列管理器或队列。 这需要一个特殊用户帐户，IBM MQ 可以使用该帐户来查询有关尝试此类访问的任何用户的信息。

- 第 128 页的『使用 Prepare IBM MQ Wizard 配置特殊用户帐户』
- 第 128 页的『将 IBM MQ 与 Active Directory 配合使用』
- 第 129 页的『IBM MQ Windows 服务所需的用户权限』

使用 Prepare IBM MQ Wizard 配置特殊用户帐户

Prepare IBM MQ Wizard 创建特殊用户帐户，以便 Windows 服务可以由需要使用该服务的进程共享 (请参阅 配置 IBM MQ 和 PPrepare IBM MQ Wizard)。

Windows 服务在 IBM MQ 安装的客户机进程之间共享。 将为每个安装创建一个服务。 每个服务都命名为 `MQ_InstallationName`，并且显示名称为 `IBM MQ(InstallationName)`。

由于每个服务必须在非交互式和交互式登录会话之间共享，因此必须在特殊用户帐户下启动每个服务。 您可以将一个特殊用户帐户用于所有服务，或者创建不同的特殊用户帐户。 每个特殊用户帐户都必须具有 **作为服务登录**的用户权限，有关更多信息，请参阅 第 129 页的表 14。 如果用户标识没有运行该服务的权限，那么该服务不会启动，并且会在 Windows 系统事件日志中返回错误。 通常，您将运行 Prepare IBM MQ Wizard，并正确设置用户标识。 但是，如果您手动配置了用户标识，那么是否可能存在需要解决的问题。

首次安装 IBM MQ 并运行 Prepare IBM MQ Wizard 时，它将为名为 MUSR_MQADMIN 的服务创建具有必需设置和许可权 (包括 **作为服务登录**) 的本地用户帐户。

对于后续安装， Prepare IBM MQ Wizard 将创建名为 MUSR_MQADMIN*x* 的用户帐户，其中 *x* 是表示不存在的用户标识的下一个可用数字。 MUSR_MQADMIN*x* 的密码是在创建帐户时随机生成的，用于配置服务的登录环境。 生成的密码不会到期。

此 IBM MQ 帐户不受系统上设置为要求在特定时间段后更改帐户密码的任何帐户策略的影响。

密码在此一次性处理外部未知，并且由 Windows 操作系统存储在注册表的安全部分中。

将 IBM MQ 与 Active Directory 配合使用

在某些网络配置中，用户帐户是在使用 Active Directory 目录服务的域控制器上定义的，运行 IBM MQ 的本地用户帐户可能没有查询其他域用户帐户的组成员资格所需的权限。 安装 IBM MQ 时， Prepare IBM MQ Wizard 通过执行测试并询问您有关网络配置的问题来确定是否存在这种情况。

如果运行 IBM MQ 的本地用户帐户没有必需的权限，那么 Prepare IBM MQ Wizard 会提示您输入具有特定用户权限的域用户帐户的帐户详细信息。 有关如何创建和设置 Windows 域帐户的信息，请参阅 为 IBM MQ 创建和设置 Windows 域帐户。 有关域用户帐户所需的用户权限，请参阅 第 129 页的表 14。

在 Prepare IBM MQ Wizard 中输入域用户帐户的有效帐户详细信息后，向导会将 IBM MQ Windows 服务配置为在新帐户下运行。 帐户详细信息保存在注册表的安全部分中，用户无法读取。

当服务处于运行状态时，将启动 IBM MQ Windows 服务，并且只要该服务处于运行状态，就会保持该服务处于运行状态。 在 Windows 服务启动后登录到服务器的 IBM MQ 管理员可以使用 IBM MQ Explorer 来管理服务器上的队列管理器。 这会将 IBM MQ Explorer 连接到现有 Windows 服务进程。 这两个操作需要不同级别的许可权才能工作:

- 启动过程需要启动许可权。
- IBM MQ 管理员需要 "访问" 许可权。

IBM MQ Windows 服务所需的用户权限

下表列出了运行 IBM MQ 安装的 Windows 服务所使用的本地和域用户帐户所需的用户权限。

表 *14: IBM MQ Windows* 服务所需的用户权限

许可权	**描述**
以批处理作业身份登录	使 IBM MQ Windows 服务能够在此用户帐户下运行。
作为服务登录	使用户能够设置 IBM MQ Windows 服务以使用配置的帐户登录。
关闭系统	如果配置为在服务恢复失败时重新启动服务器，那么允许 IBM MQ Windows 服务重新启动服务器。
增加配额	对于操作系统 `CreateProcessAsUser` 调用是必需的。
作为操作系统的组成部分	对于操作系统 `LogonUser` 调用是必需的。
回避遍历检查	对于操作系统 `LogonUser` 调用是必需的。
替换进程层次标记	对于操作系统 `LogonUser` 调用是必需的。

注: 在运行 ASP 和 IIS 应用程序的环境中可能需要调试程序权限。

您的域用户帐户必须将这些 Windows 用户权限设置为本地安全策略应用程序中列出的有效用户权限。 如果不是，请在服务器上本地使用 "本地安全策略" 应用程序或使用 "域安全性应用程序" 域来设置它们。

Windows *Windows* 服务器安全许可权

IBM MQ 的安装在 Windows Server 上的行为有所不同，具体取决于本地用户还是域用户执行安装。

如果 本地 用户安装 IBM MQ，那么 Prepare IBM MQ Wizard 会检测到为 IBM MQ Windows 服务创建的本地用户可以检索安装用户的组成员资格信息。 Prepare IBM MQ Wizard 询问用户有关网络配置的问题，以确定在 Windows 2000 或更高版本上运行的域控制器上是否定义了其他用户帐户。 如果是这样，那么 IBM MQ Windows 服务需要在具有特定设置和权限的域用户帐户下运行。 Prepare IBM MQ Wizard 会提示用户输入此用户的帐户详细信息，如 使用 Prepare IBM MQ Wizard 配置 IBM MQ 中所述。

如果 域 用户安装 IBM MQ，那么 Prepare IBM MQ Wizard 会检测到为 IBM MQ Windows 服务创建的本地用户无法检索安装用户的组成员资格信息。 在这种情况下， Prepare IBM MQ Wizard 始终提示用户输入要使用的 IBM MQ Windows 服务的域用户帐户的帐户详细信息。

当 IBM MQ Windows 服务需要使用域用户帐户时，在使用 Prepare IBM MQ Wizard 配置此帐户之前， IBM MQ 无法正常运行。 在使用适当的帐户配置 Windows 服务之前， Prepare IBM MQ Wizard 不允许用户继续执行其他任务。

有关更多信息，请参阅 为 IBM MQ 创建和设置域帐户。

Windows 更改与 *IBM MQ* 服务关联的用户名

您可以通过创建新帐户并使用 Prepare IBM MQ Wizard 输入其详细信息来更改与 IBM MQ 服务关联的用户名。

关于此任务

首次安装 IBM MQ 并运行 Prepare IBM MQ Wizard 时，它会为名为 MUSR_MQADMIN 的服务创建本地用户帐户。 对于后续安装， Prepare IBM MQ Wizard 将创建名为 MUSR_MQADMIN*x* 的用户帐户，其中 *x* 是表示不存在的用户标识的下一个可用数字。

您可能需要将与 IBM MQ 服务关联的用户名从 MUSR_MQADMIN 或 MUSR_MQADMIN*x* 更改为其他名称。例如，如果队列管理器与 Db2 相关联，那么您可能需要执行此操作，而不接受长度超过 8 个字符的用户名。

过程

1. 创建新的用户帐户 (例如 **NEW_NAME**)
2. 使用 Prepare IBM MQ Wizard 输入新用户帐户的详细信息。

相关任务

使用 Prepare IBM MQ Wizard 配置 IBM MQ

Windows 更改 *IBM MQ Windows* 服务本地用户帐户的密码

您可以使用 "计算机管理" 面板来更改 IBM MQ Windows 服务本地用户帐户的密码。

关于此任务

要更改 IBM MQ Windows 服务本地用户帐户的密码，请执行以下步骤:

过程

1. 标识正在运行服务的用户。
2. 从 "计算机管理" 面板停止 IBM MQ 服务。
3. 更改所需密码的方式与更改个人密码的方式相同。
4. 从 "计算机管理" 面板转至 IBM MQ 服务的属性。
5. 选择 " **登录** " 页面。
6. 确认指定的帐户名称与修改了密码的用户匹配。
7. 在 **密码** 和 **确认密码** 字段中输入密码，然后单击 **确定**。

Windows 更改在域用户帐户下运行的安装的 *IBM MQ Windows* 服务的密码

作为使用 Prepare IBM MQ Wizard 输入域用户帐户的帐户详细信息的替代方法，您可以使用 "计算机管理" 面板来变更特定于安装的 IBM MQ 服务的 **登录** 详细信息。

关于此任务

如果安装的 IBM MQ Windows 服务正在域用户帐户下运行，那么您可以按如下所示更改帐户的密码:

过程

1. 更改域控制器上的域帐户的密码。 您可能需要请求域管理员为您执行此操作。
2. 完成以下步骤以修改 IBM MQ 服务的 " **登录** " 页面。
   a) 标识运行服务的用户。
   b) 从 "计算机管理" 面板停止 IBM MQ 服务。
   c) 更改所需密码的方式与更改个人密码的方式相同。
   d) 从 "计算机管理" 面板转至 IBM MQ 服务的属性。

e) 选择 " **登录** " 页面。

f) 确认指定的帐户名称与修改了密码的用户匹配。

g) 在 **密码** 和 **确认密码** 字段中输入密码，然后单击 **确定**。

IBM MQ Windows 服务在其中运行的用户帐户将执行用户界面应用程序发出的任何 MQSC 命令，或在系统启动，关闭或服务恢复时自动执行的任何 MQSC 命令。 因此，此用户帐户必须具有 IBM MQ 管理权限。 缺省情况下，会将其添加到服务器上的本地 mqm 组。 如果除去此成员资格，那么 IBM MQ Windows 服务不起作用。 有关用户权限的更多信息，请参阅 第 129 页的『IBM MQ Windows 服务所需的用户权限』。

如果运行 IBM MQ Windows 服务的用户帐户出现安全问题，那么系统事件日志中会显示错误消息和描述。

相关任务

使用 Prepare IBM MQ Wizard 配置 IBM MQ

Windows 将 *Windows* 服务器提升到域控制器时的注意事项

将 Windows 服务器提升到域控制器时，应考虑与用户和组许可权相关的安全设置是否合适。 在服务器与域控制器之间更改 Windows 机器的状态时，您应该考虑这可能会影响 IBM MQ 的操作，因为 IBM MQ 使用本地定义的 mqm 组。

与域用户和组许可权相关的安全设置

IBM MQ 依赖于组成员资格信息来实现其安全策略，这意味着执行 IBM MQ 操作的用户标识可以确定其他用户的组成员资格非常重要。

将 Windows 服务器提升到域控制器时，将向您提供与用户和组许可权相关的安全设置选项。 此选项控制任意用户是否能够从 Active Directory 中检索组成员资格。 如果设置了域控制器以使本地帐户具有查询域用户帐户的组成员资格的权限，那么 IBM MQ 在安装过程中创建的缺省用户标识可以根据需要为其他用户获取组成员资格。 但是，如果设置了域控制器以使本地帐户无权查询域用户帐户的组成员资格，那么这将阻止 IBM MQ 完成其检查，即域上定义的用户有权访问队列管理器或队列，并且访问失败。 如果您正在以这种方式设置的域控制器上使用 Windows ，那么必须使用具有所需许可权的特殊域用户帐户。

在这种情况下，您需要知道:

- Windows 版本的安全许可权的行为方式。
- 如何允许域 mqm 组成员读取组成员资格。
- 如何将 IBM MQ Windows 服务配置为在域用户下运行。

有关更多信息，请参阅 配置 IBM MQ 的用户帐户。

对本地 mqm 组的 IBM MQ 访问权

当 Windows 服务器提升到域控制器或从域控制器降级时， IBM MQ 将失去对本地 mqm 组的访问权。

当服务器提升为域控制器时，作用域将从本地更改为域本地。 将机器降级到服务器时，将除去所有域本地组。 这意味着将机器从服务器更改为域控制器并返回到服务器将失去对本地 mqm 组的访问权。 症状是指示缺少本地 mqm 组的错误，例如:

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

要解决此问题，请使用标准 Windows 管理工具重新创建本地 mqm 组。 由于所有组成员资格信息都已丢失，因此必须在新创建的本地 mqm 组中恢复特权 IBM MQ 用户。 如果机器是域成员，那么还必须将域 mqm 组添加到本地 mqm 组，以授予特权域 IBM MQ 用户标识所需的权限级别。

Windows *Windows* 上嵌套组的限制

在使用嵌套组方面存在一些限制。 这些结果部分来自域功能级别，部分来自 IBM MQ 限制。

Active Directory 可以支持 "域" 上下文中的不同组类型，具体取决于 "域" 功能级别。 缺省情况下，Windows 2003 域位于 " Windows 2000 混合 " 功能级别。 (Windows Server 2008 和 Windows Server

2012 遵循 Windows 2003 域模型。) 域功能级别确定在域环境中配置用户标识时允许的受支持组类型和嵌套级别。 请参阅 Active Directory 文档，以获取有关组作用域和包含条件的详细信息。

除 Active Directory 需求外，还对 IBM MQ 所使用的标识施加了进一步限制。 IBM MQ 使用的网络 API 不支持域功能级别支持的所有配置。 因此， IBM MQ 无法查询随后嵌套在本地组中的域本地组中存在的任何域标识的组成员资格。 此外，不支持对全局组和通用组进行多重嵌套。 但是，支持立即嵌套的全局组或通用组。

Windows 授权用户远程使用 *IBM MQ*

如果需要在远程连接到 IBM MQ 时创建和启动队列管理器，那么您必须具有 **创建全局对象** 用户访问权。

关于此任务

注: 缺省情况下，管理员具有 **创建全局对象** 用户访问权，因此如果您是管理员，那么可以在远程连接时创建和启动队列管理器，而无需更改用户权限。

如果使用 "终端服务" 或 "远程桌面连接" 连接到 Windows 机器，并且在创建，启动或删除队列管理器时迁到问题，这可能是因为您没有用户访问权 **创建全局对象**。

创建全局对象 用户访问权限制有权在全局名称空间中创建对象的用户。 为了使应用程序能够创建全局对象，它必须在全局名称空间中运行，或者运行应用程序的用户必须对其应用 **创建全局对象** 用户访问权。

使用终端服务或远程桌面连接远程连接到 Windows 机器时，应用程序将在其自己的本地名称空间中运行。 如果尝试使用 IBM MQ Explorer 或 **crtmqm** 或 **dltmqm** 命令创建或删除队列管理器，或者使用 **strmqm** 命令启动队列管理器，那么会导致授权失败。 这将创建具有探测器标识 XY132002 的 IBM MQ FDC。

使用 IBM MQ Explorer 或 **amqmdain qmgr start** 命令启动队列管理器正常工作，因为这些命令不会直接启动队列管理器。 相反，这些命令会将启动队列管理器的请求发送到在全局名称空间中运行的单独进程。

如果使用终端服务时管理 IBM MQ 的各种方法不起作用，请尝试设置 **创建全局对象** 用户权限。

过程

1. 打开 "管理工具" 面板:

   Windows Server 2008 和 Windows Server 2012
   使用 **控制面板** > **系统和维护** > **管理工具**来访问此面板。

   Windows 8.1
   使用 **管理工具** > **计算机管理** 访问此面板
2. 双击**本地安全策略**。
3. 展开 **本地策略**。
4. 单击**用户权利指派**。
5. 将新用户或组添加到 **创建全局对象** 策略。

Windows *Windows* 上的 *SSPI* 通道出口程序

IBM MQ for Windows 提供了可在消息和 MQI 通道上使用的安全出口程序。 出口作为源和对象代码提供，并提供单向和双向认证。

安全出口使用安全支持提供程序接口 (SSPI) ， 该接口提供 Windows 平台的集成安全设施。

安全出口提供以下标识和认证服务:

单向认证 (one way authentication)
这将使用 Windows NT LAN Manager (NTLM) 认证支持。 NTLM 允许服务器认证其客户机。 它不允许客户机对服务器进行认证，也不允许一个服务器对另一个服务器进行认证。 NTLM 是为一个网络环境而设计的，在该网络环境中，假定服务器是真实的。 在 IBM WebSphere MQ 7.0 支持的所有 Windows 平台上都支持 NTLM。

此服务通常在 MQI 通道上用于使服务器队列管理器能够认证 IBM MQ MQI client 应用程序。 客户机应用程序由与正在运行的进程相关联的用户标识进行标识。

要执行认证，通道客户机端的安全出口从 NTLM 获取认证令牌，并将安全消息中的令牌发送到通道另一端的合作伙伴。 伙伴安全出口将令牌传递到 NTLM ，这将检查令牌是否真实。 如果合作伙伴安全出口对令牌的真实性不满意，那么指示 MCA 关闭通道。

双向或相互认证

这将使用 Kerberos 认证服务。 Kerberos 协议不会假设网络环境中的服务器真实可靠。 服务器可以认证客户机和其他服务器，客户机可以认证服务器。 Kerberos 在 IBM WebSphere MQ 7.0 支持的所有 Windows 平台上都受支持。

可以在消息和 MQI 通道上使用此服务。 在消息通道上，它提供两个队列管理器的相互认证。 在 MQI 通道上，它使服务器队列管理器和 IBM MQ MQI client 应用程序能够相互认证。 队列管理器由以字符串 `ibmMQSeries/`为前缀的名称标识。 客户机应用程序由与正在运行的进程相关联的用户标识进行标识。

为执行相互认证，启动安全性出口从 Kerberos 安全性服务器获取认证令牌，并将安全性消息中的令牌发送给其合作伙伴。 伙伴安全出口将令牌传递到 Kerberos 服务器，这将检查令牌是否真实。 Kerberos 安全性服务器生成第二个令牌，合作伙伴将该令牌在安全性消息中发送到启动安全性出口。 然后，启动安全性出口会要求 Kerberos 服务器检查第二个令牌是否真实。 在此交换期间，如果任一安全出口对另一个安全出口发送的令牌的真实性不满意，那么将指示 MCA 关闭通道。

安全出口以源和对象格式提供。 您可以使用源代码作为编写自己的通道出口程序的起点，也可以使用提供的对象模块。 对象模块有两个入口点，一个用于使用 NTLM 认证支持的单向认证，另一个用于使用 Kerberos 认证服务的双向认证。

有关 SSPI 通道出口程序如何工作的更多信息，以及有关如何实现该程序的指示信息，请参阅 在 Windows 系统上使用 SSPI 安全出口。

Windows 在 *Windows* 上应用安全模板文件

应用模板可能会影响应用于 IBM MQ 文件和目录的安全设置。 如果使用高度安全的模板，请在安装 IBM MQ 之前应用该模板。

Windows 支持基于文本的安全性模板文件，您可以使用这些文件将统一安全性设置应用于具有安全性配置和分析 MMC 插件的一台或多台计算机。 特别是， Windows 提供了多个模板，这些模板包含一系列安全设置，目的是提供特定级别的安全性。 这些模板包括 "兼容" ， "安全" 和 "高度安全"。

应用其中一个模板可能会影响应用于 IBM MQ 文件和目录的安全设置。 如果要使用 "高度安全" 模板，请在安装 IBM MQ 之前配置机器。

如果将高度安全的模板应用于已安装 IBM MQ 的机器，那么将除去您对 IBM MQ 文件和目录设置的所有许可权。 由于已除去这些许可权，因此您将失去 管理员， *mqm* 以及 每个人 组对错误目录的访问权 (如果适用)。

Windows 为连接到 *IBM MQ* 的 *Windows* 应用程序配置额外权限

运行 IBM MQ 进程的帐户可能需要额外授权，然后才能授予对应用程序进程的同步访问权。

关于此任务

如果您有 Windows 应用程序 (例如 ASP 页面) 连接到配置为在高于通常的安全级别运行的 IBM MQ ，那么可能会迁到问题。

IBM MQ 需要对应用程序进程的同步访问权，以便协调某些操作。 当服务器应用程序首次尝试连接到队列管理器 IBM MQ 时，将修改该进程以授予 IBM MQ 管理员同步权限。 但是，运行 IBM MQ 进程的帐户可能需要额外授权，然后才能授予所请求的访问权。

要配置对运行 IBM MQ 进程的用户标识的其他权限，请完成以下步骤:

过程

1. 启动 "本地安全策略" 工具，单击 **安全设置->本地策略->用户权限分配**，然后单击 **调试程序**。
2. 双击 **调试程序**，然后将您的 IBM MQ 用户标识添加到列表中

   如果系统位于 Windows 域中，并且仍未设置有效策略设置，那么即使设置了本地策略设置，也必须使用 "域安全策略" 工具以相同方式在域级别对用户标识进行授权。

IBM i 在 IBM i 上设置安全性

IBM i 上的安全性是使用 IBM MQ 对象权限管理器 (OAM) 和 IBM i 对象级别安全性实现的。

确定对 IBM MQ 对象的访问权时必须进行的安全注意事项。

在对企业中的用户设置权限时，需要考虑以下几点:

1. 使用 IBM i GRTOBJAUT 和 RVKOBJAUT 命令授予和撤销对 IBM MQ for IBM i 命令的权限。

   在 QMQM 库中，某些非命令 (* cmd) 对象设置为具有 ***USE** 的 ***PUBLIC** 权限。 请勿更改这些对象的权限或使用权限列表来提供权限。 任何不正确的权限都可能损害 IBM MQ 功能。
2. 在 IBM MQ for IBM i 安装期间，将创建以下特殊用户概要文件:

   QMQM
   主要用于仅限内部产品的功能。 但是，它可用于使用 MQCNO_FASTPATH_BINDINGS 运行可信应用程序。 请参阅 使用 MQCONNX 调用连接到队列管理器。

   QMQMADM
   用作 IBM MQ 管理员的组概要文件。 组概要文件提供对 CL 命令和 IBM MQ 资源的访问权。

   使用 SBMJOB 提交调用 IBM MQ 命令的程序时，不得将 USER 显式设置为 QMQMADM。 请改为将 USER 设置为 QMQM 或将 QMQMADM 指定为组的其他用户概要文件。
3. 如果要向远程队列管理器发送通道命令，请确保用户概要文件是目标系统上的组 QMQMADM 的成员。 有关 PCF 和 MQSC 通道命令的列表，请参阅 IBM MQ for IBM i CL 命令。
4. 当 OAM 计算组权限时，将高速缓存与用户关联的组集。

   直到您重新启动队列管理器或执行 RFRMQMAUT 以刷新安全性之后，才会识别在高速缓存组集之后对用户组成员资格所作的任何更改。
5. 限制有权使用特别敏感的命令的用户数。 这些命令包括:
   - 创建消息队列管理器 (CRTMQM)
   - 删除消息队列管理器 (DLTMQM)
   - 启动消息队列管理器 (STRMQM)
   - 结束消息队列管理器 (ENDMQM)
   - 启动命令服务器 (STRMQMCSVR)
   - 结束命令服务器 (ENDMQMCSVR)
6. 通道定义包含安全出口程序规范。 通道创建和修改需要特殊注意事项。 第 94 页的『安全出口概述』中提供了安全出口的详细信息。
7. 可以替换通道出口和触发器监视器程序。 这种替换的安全性由程序员负责。

IBM i IBM i 上的对象权限管理器

对象权限管理器 (OAM) 管理用户处理 IBM MQ 对象 (包括队列和进程定义) 的权限。 它还提供了一个命令界面，您可以通过该界面为特定用户组授予或撤销对对象的访问权。 允许访问资源的决策由 OAM 做出，队列管理器遵循该决策。 如果 OAM 无法做出决策，那么队列管理器将阻止访问该资源。

通过 OAM ，您可以控制:

- 通过 MQI 访问 IBM MQ 对象。 当应用程序尝试访问对象时， OAM 会检查发出请求的用户概要文件是否具有所请求操作的权限。

  特别是，这意味着可以保护队列以及队列上的消息免受未经授权的访问。
- 允许使用 PCF 和 MQSC 命令。

不同的用户组可以对同一对象具有不同的访问权限。 例如，对于特定队列，一个组可以执行 put 和 get 操作;另一个组可能只允许浏览队列 (带有 browse 选项的 MQGET)。 类似地，某些组可能具有对队列的获取和放置权限，但不允许更改或删除该队列。

IBM MQ for IBM i 命令并对 IBM MQ for IBM i 对象执行操作

IBM i IBM i 上的 IBM MQ 权限

要访问 IBM MQ 对象，您需要发出命令和访问所引用对象的权限。 管理员有权访问所有 IBM MQ 资源。

对 IBM MQ 对象的访问权由以下权限控制:

1. 发出 IBM MQ 命令
2. 访问命令引用的 IBM MQ 对象

所有 IBM MQ for IBM i CL 命令都随 QMQM 的所有者一起提供，并且管理概要文件 (QMQMADM) 具有 *USE 权限，并且 *PUBLIC 访问权设置为 *EXCLUDE。

注: QSRDUPER 程序由 IBM MQ for IBM i 许可程序安装程序用于复制 QSYS 中的命令 (*CMD) 对象。 在 IBM i V5R4 和更高版本中， QSRDUPER 程序已更改，因此缺省行为是创建代理命令而不是复制原始命令。 代理命令将命令执行重定向到另一个命令，并具有属性 PRX。 如果库 QSYS 中存在与正在复制的命令同名的代理命令，那么不会向产品库中的命令授予对代理命令的专用权限。 尝试在 QSYS 中提示或运行代理命令，请检查产品库中目标命令的权限。 因此，需要在产品库 (QMQM) 中执行对 *CMD 对象的权限的任何更改，而不需要修改 QSYS 中的那些更改。 例如:

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

如果您对 IBM MQ 对象具有进行这些更改所需的 OAM 权限，那么对产品的某些 CL 命令的权限结构的更改允许公开使用这些命令。

要成为 IBM i 上的 IBM MQ 管理员，您必须是 *QMQMADM* 组的成员。 此组具有类似 AIX, Linux, and Windows 系统上 mqm 组的属性的属性。 尤其是在安装 IBM MQ for IBM i 时创建 QMQMADM 组，并且 QMQMADM 组的成员有权访问系统上的所有 IBM MQ 资源。 如果您具有 *ALLOBJ 权限，那么您还可以访问所有 IBM MQ 资源。

管理员可以使用 CL 命令来管理 IBM MQ。 其中一个命令是 GRTMQMAUT ，用于向其他用户授予权限。 另一个命令 STRMQMMQSC 允许管理员向本地队列管理器发出 MQSC 命令。

相关概念

第 77 页的『在 IBM i 上管理 IBM MQ 的权限』

IBM i *IBM i 上 IBM MQ* 对象的访问权限

运行 IBM MQ CL 命令所需的访问权限。

IBM MQ for IBM i 将产品的 CL 命令分类为两个组:

第 1 组

用户必须在 QMQMADM 用户组中，或者具有 *ALLOBJ 权限才能处理这些命令。 具有其中任一权限的用户可以处理所有类别中的所有命令，而无需任何额外权限。

注: 这些权限覆盖任何 OAM 权限。

这些命令可按如下所示进行分组:

- 命令服务器命令
  - ENDMQMCSVR ，结束 IBM MQ 命令服务器
  - STRMQMCSVR ，启动 IBM MQ 命令服务器
- 死信队列处理程序命令
  - STRMQMDLQ ，启动 IBM MQ 死信队列处理程序
- 侦听器命令
  - ENDMQMLSR ，结束 IBM MQ 侦听器
  - STRMQMLSR ，启动非对象侦听器
- 媒体恢复命令
  - RCDMQMIMG ，记录 IBM MQ 对象图像

- RCRMQMOBJ，重新创建 IBM MQ 对象
- WRKMQMTRN，使用 IBM MQ Q 事务

- 队列管理器命令
  - CRTMQM，创建消息队列管理器
  - DLTMQM，删除消息队列管理器
  - ENDMQM，结束消息队列管理器
  - STRMQM，启动消息队列管理器
- 安全性命令
  - GRTMQMAUT，授予 IBM MQ 对象权限
  - RVKMQMAUT，撤销 IBM MQ 对象权限
- 跟踪命令
  - TRCMQM，跟踪 IBM MQ 作业
- 事务命令
  - RSVMQMTRN，解析 IBM MQ 事务
- 触发器监视器命令
  - STRMQMTRM，启动触发器监视器
- IBM MQSC 命令
  - RUNMQSC，运行 IBM MQSC 命令
  - STRMQMMQSC，启动 IBM MQSC 命令

组 2

其余命令 (需要两个级别的权限):

1. 用于运行命令的 IBM i 权限。 IBM MQ 管理员使用 **GRTOBJAUT** 命令来设置此项，以覆盖用户或用户组的 *PUBLIC (*EXCLUDE) 限制。

   例如:

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

2. IBM MQ 用于处理与命令关联的 IBM MQ 对象的权限，在步骤 1 中提供了正确的 IBM i 权限。

   此权限由具有所需操作的相应 OAM 权限的用户控制，由 IBM MQ 管理员使用 **GRTMQMAUT** 命令设置

   例如:

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

这些命令可按如下所示进行分组:

- 通道命令
  - CHGMQMCHL，更改 IBM MQ 通道

    这需要对队列管理器的 * 连接权限和对通道的 * admchg 权限。
  - CPYMQMCHL，复制 IBM MQ 通道

    这需要队列管理器的 * connect 和 * admcrt 权限，要复制的缺省通道类型的 * admdsp 权限以及通道对象类的 * admcrt 权限。

    例如，复制发送方通道需要 SYSTEM.DEF.SENDER 通道
  - CRTMQMCHL，创建 IBM MQ 通道

这需要对队列管理器的 * connect 和 * admcrt 权限，对要创建的缺省通道类型的 * admdsp 权限以及对通道对象类的 * admcrt 权限。

例如，创建发送方通道需要 SYSTEM.DEF.SENDER 通道

– DLTMQMCHL，删除 IBM MQ 通道

这需要对队列管理器的 * 连接权限和对通道的 * admdlt 权限。

– RSVMQMCHL，解析 IBM MQ 通道

这需要对队列管理器的 * 连接权限和对通道的 * ctrlx 权限。

- 显示命令

要处理 DSP 命令，必须授予用户对队列管理器的 *connect 和 *admdsp 权限以及列出的任何特定选项:

– DSPMQM，显示消息队列管理器
– DSPMQMAUT，显示 IBM MQ 对象权限
– DSPMQMAUTI，向认证信息对象显示 IBM MQ 认证信息- *admdsp
– DSPMQMCHL，向通道显示 IBM MQ 通道- *admdsp
– DSPMQMCSVR，显示 IBM MQ 命令服务器
– DSPMQMNL，向名称列表显示 IBM MQ Namelist- *admdsp
– DSPMQMOBJN，显示 IBM MQ 对象名
– DSPMQMPRC，向进程显示 IBM MQ 进程- *admdsp
– DSPMQMQ，向队列显示 IBM MQ 队列- *admdsp
– DSPMQMTOP，向主题显示 IBM MQ 主题- *admdsp

- 使用命令

要处理 WRK 命令并显示选项面板，必须授予用户对队列管理器的 *connect 和 *admdsp 权限以及列出的任何特定选项:

– WRKMQM，使用消息队列管理器
– WRKMQMAUT，使用 IBM MQ 对象权限
– WRKMQMAUTD，使用 IBM MQ 对象权限数据
– WRKMQMAUTI，使用 IBM MQ 认证信息

- *admchg 表示 "更改 IBM MQ 认证信息对象" 命令。
- *admcrt，用于 "创建和复制 IBM MQ 认证信息对象" 命令。
- *admdlt 表示 "删除 IBM MQ 认证信息对象" 命令。
- *admdsp 表示 "显示 IBM MQ 认证信息对象" 命令。

– WRKMQMCHL，使用 IBM MQ 通道

这需要以下权限:

- *admchg 表示 "更改 IBM MQ 通道" 命令。
- *admclr 表示 "清除 IBM MQ 通道" 命令。
- *admcrt 表示 "创建和复制 IBM MQ 通道" 命令。
- *admdlt 表示 "删除 IBM MQ 通道" 命令。
- *admdsp 表示 "显示 IBM MQ 通道" 命令。
- *ctrl 表示 "启动 IBM MQ 通道" 命令。
- *ctrl 表示 "结束 IBM MQ 通道" 命令。
- *ctrl 表示 Ping IBM MQ 通道命令。
- *ctrlx 表示 "重置 IBM MQ 通道" 命令。
- *ctrlx 表示 "解析 IBM MQ 通道" 命令。

- WRKMQMCHST，使用 IBM MQ 通道状态

  这需要对通道具有 *admdsp 权限。
- WRKMQMCL，使用 IBM MQ 集群
- WRKMQMCLQ，使用 IBM MQ 集群队列
- WRKMQMCLQM，使用 IBM MQ 集群队列管理器
- WRKMQMLSR，使用 IBM MQ 侦听器
- WRKMQMMSG，使用 IBM MQ 消息

  这需要对队列具有 *browse 权限
- WRKMQMNL，使用 IBM MQ 名称列表

  这需要以下权限:

  - *admchg 表示 "更改 IBM MQ 名称列表" 命令。
  - *admcrt 表示 "创建和复制 IBM MQ 名称列表" 命令。
  - "删除 IBM MQ 名称列表" 命令的 *admdlt 。
  - *admdsp 表示 "显示 IBM MQ 名称列表" 命令。
- WRKMQMPRC，使用 IBM MQ 进程

  这需要以下权限:

  - *admchg 表示 "更改 IBM MQ 进程" 命令。
  - *admcrt 表示 "创建和复制 IBM MQ 进程" 命令。
  - *admdlt 表示 "删除 IBM MQ 进程" 命令。
  - *admdsp 表示 "显示 IBM MQ 进程" 命令。
- WRKMQMQ，使用 IBM MQ 队列

  这需要以下权限:

  - *admchg 表示 "更改 IBM MQ 队列" 命令。
  - *admclr 表示 "清除 IBM MQ 队列" 命令。
  - *admcrt 表示 "创建和复制 IBM MQ 队列" 命令。
  - *admdlt 表示 "删除 IBM MQ 队列" 命令。
  - *admdsp 表示 "显示 IBM MQ 队列" 命令。
- WRKMQMQSTS，使用 IBM MQ 队列状态
- WRKMQMTOP，使用 IBM MQ 主题

  这需要以下权限

  - *admchg 表示 "更改 IBM MQ 主题" 命令。
  - *admcrt 表示 "创建和复制 IBM MQ 主题" 命令。
  - *admdlt 表示 "删除 IBM MQ 主题" 命令。
  - *admdsp 表示 "显示 IBM MQ 主题" 命令。
- WRKMQMSUB，使用 IBM MQ 预订

- 其他通道命令

  要处理通道命令，必须向用户授予列出的特定权限:

  - ENDMQMCHL，结束 IBM MQ 通道

    这需要对队列管理器具有 *connect 权限，对与通道关联的传输队列具有 *allmqi 权限。
  - ENDMQMLSR，结束 IBM MQ 侦听器

    这需要对队列管理器的 *connect 权限和对指定侦听器对象的 *ctrl 权限。
  - PNGMQMCHL， Ping IBM MQ 通道

这需要对队列管理器的 *connect 和 *inq 权限以及对通道对象的 *ctrl 权限。

- RSTMQMCHL，重置 IBM MQ 通道

  这需要对队列管理器具有 *connect 权限。

- STRMQMCHL，启动 IBM MQ 通道

  这需要对队列管理器具有 *connect 权限，对通道对象具有 *ctrl 权限。

- STRMQMCHLI，启动 IBM MQ 通道启动程序

  这需要对队列管理器的 *connect 和 *inq 权限，以及对与通道的传输队列关联的启动队列的 *allmqi 权限。

- STRMQMLSR，启动 IBM MQ 侦听器

  这需要对队列管理器的 * 连接权限和对指定侦听器对象的 * ctrl 权限。

- 其他命令:

  要处理以下命令，必须向用户授予列出的特定权限:

  - CCTMQM，连接到消息队列管理器

    这不需要 IBM MQ 对象权限。

  - CHGMQM，更改消息队列管理器

    这需要对队列管理器具有 *connect 和 *admchg 权限。

  - CHGMQMAUTI，更改 IBM MQ 认证信息

    这需要对队列管理器具有 *connect 权限，对认证信息对象具有 *admchg 和 *admdsp 权限。

  - CHGMQMNL，更改 IBM MQ 名称列表

    这需要对队列管理器具有 *connect 权限，对名称列表具有 *admchg 权限。

  - CHGMQMPRC，更改 IBM MQ 进程

    这需要对队列管理器具有 *connect 权限，对进程具有 *admchg 权限。

  - CHGMQMQ，更改 IBM MQ 队列

    这需要对队列管理器的 *connect 权限和对队列的 *admchg 权限。

  - CLRMQMQ，清除 IBM MQ 队列

    这需要对队列管理器的 *connect 权限和对队列的 *admclr 权限。

  - CPYMQMAUTI，复制 IBM MQ 认证信息

    这需要对队列管理器具有 *connect 权限，对认证信息对象具有 *admdsp 权限，对认证信息对象类具有 *admcrt 权限。

  - CPYMQMNL，复制 IBM MQ 名称列表

    这需要对队列管理器具有 *connect 和 *admcrt 权限。

  - CPYMQMPRC，复制 IBM MQ 进程

    这需要对队列管理器具有 *connect 和 *admcrt 权限。

  - CPYMQMQ，复制 IBM MQ 队列

    这需要对队列管理器具有 *connect 和 *admcrt 权限。

  - CRTMQMAUTI，创建 IBM MQ 认证信息

    这需要对队列管理器具有 *connect 权限，对认证信息对象具有 *admdsp 权限，对认证信息对象类具有 *admcrt 权限。

  - CRTMQMNL，创建 IBM MQ 名称列表

    这需要对队列管理器具有 *connect 和 *admcrt 权限，对缺省名称列表具有 *admdsp 权限。

  - CRTMQMPRC，创建 IBM MQ 进程

    这需要对队列管理器具有 *connect 和 *admcrt 权限，对缺省进程具有 *admdsp 权限。

- CRTMQMQ，创建 IBM MQ 队列

  这需要对队列管理器具有 *connect 和 *admcrt 权限，对缺省队列具有 *admdsp 权限。
- CVTMQMDTA，转换 IBM MQ 数据类型命令

  这不需要 IBM MQ 对象权限。
- DLTMQMAUTI，删除 IBM MQ 认证信息

  这需要对队列管理器具有 *connect 权限，对认证信息对象具有 *ctrlx 权限。
- DLTMQMNL，删除 IBM MQ 名称列表

  这需要对队列管理器具有 *connect 权限，对名称列表具有 *admdlt 权限。
- DLTMQMPRC，删除 IBM MQ 进程

  这需要对队列管理器具有 *connect 权限，对进程具有 *admdlt 权限。
- DLTMQMQ，删除 IBM MQ 队列

  这需要对队列管理器的 *connect 权限和对队列的 *admdlt 权限。
- DSCMQM，从消息队列管理器断开连接

  这不需要 IBM MQ 对象权限。
- RFRMQMAUT，刷新安全性

  这需要对队列管理器具有 *connect 权限。
- RFRMQMCL，刷新集群

  这需要对队列管理器具有 *connect 权限。
- RSMMQMCLQM，恢复集群队列管理器

  这需要对队列管理器具有 *connect 权限。
- RSTMQMCL，重置集群

  这需要对队列管理器具有 *connect 权限。
- SPDMQMCLQM，暂挂集群队列管理器

  这需要对队列管理器具有 *connect 权限。

IBM i *IBM i 上的访问权限*

使用此信息可了解访问授权命令。

由 GRTMQMAUT 和 RVKMQMAUT 命令上的 AUT 关键字定义的权限可以分类如下:

- 与 MQI 调用相关的授权
- 与授权相关的管理命令
- 上下文授权
- 常规授权，即 MQI 调用和/或命令

下表列出了对 MQI 调用，上下文调用， MQSC 和 PCF 命令以及通用操作使用 AUT 参数的不同权限。

表 *15: MQI* 调用的授权

AUT	**描述**
*ALTUSR	允许其他用户的权限用于 MQOPEN 和 MQPUT1 调用。
*BROWSE	通过发出带有 BROWSE 选项的 MQGET 调用从队列中检索消息。
*CONNECT	通过发出 MQCONN 调用将应用程序连接到指定的队列管理器。
*GET	通过发出 MQGET 调用从队列中检索消息。
*INQ	通过发出 MQINQ 调用对特定队列进行查询。

表 *15: MQI* 调用的授权 *(继续)*	
AUT	**描述**
*PUB	打开主题以使用 MQPUT 调用发布消息。
*PUT	通过发出 MQPUT 调用将消息放入特定队列。
*RESUME	使用 MQSUB 调用恢复预订。
*SET	通过发出 MQSET 调用来设置来自 MQI 的队列上的属性。 如果为多个选项打开队列，那么必须为每个选项授权。
*SUB	使用 MQSUB 调用创建，更改或恢复对主题的预订。

表 *16:* 上下文调用的权限	
AUT	**描述**
*PASSALL	传递指定队列上的所有上下文。 将从原始请求复制所有上下文字段。
*PASSID	传递指定队列上的身份上下文。 身份上下文与请求的身份上下文相同。
*SETALL	设置指定队列上的所有上下文。 这由特殊系统实用程序使用。
*SETID	在指定队列上设置身份上下文。 这由特殊系统实用程序使用。

表 *17: MQSC* 和 *PCF* 调用的授权	
AUT	**描述**
*ADMCHG	更改指定对象的属性。
*ADMCLR	清除指定的对象 (仅 PCF 清除对象命令)。
*ADMCRT	创建指定类型的对象。
*ADMDLT	请删除指定的对象。
*ADMDSP	显示指定对象的属性。

表 *18:* 通用操作的授权	
AUT	**描述**
*ALL	使用适用于该对象的所有操作。 all 权限相当于对应于对象类型的权限 alladm, allmqi 和 system 的并集。
*ALLADM	执行适用于该对象的所有管理操作。
*ALLMQI	使用适用于该对象的所有 MQI 调用。
*CTRL	控制通道，侦听器和服务的启动和关闭。
*CTRLX	重置序号并解析不确定通道。

IBM i 在 *IBM i* 上使用访问授权命令

使用此信息来了解有关访问授权命令的信息，并使用命令示例。

使用 GRTMQMAUT 命令

如果您具有必需的权限，那么可以使用 GRTMQMAUT 命令来授予用户概要文件或用户组访问特定对象的权限。 以下示例说明如何使用 GRTMQMAUT 命令:

1.
```
GRTMQMAUT OBJ(RED.LOCAL.QUEUE) OBJTYPE(*LCLQ) USER(GROUPA) +
          AUT(*BROWSE *PUT) MQMNAME('saturn.queue.manager')
```
在该示例中:

- RED.LOCAL.QUEUE 这是对象名。
- *LCLQ (本地队列) 是对象类型。
- GROUPA 是要更改其权限的系统上的用户概要文件的名称。 此概要文件可用作其他用户的组概要文件。
- *BROWSE 和 *PUT 是对指定队列授予的权限。

  *BROWSE 添加了在队列上浏览消息的权限 (使用浏览选项发出 MQGET)。

  *PUT 添加了将 (MQPUT) 消息放入队列的权限。
- saturn.queue.manager 是队列管理器名称。

2. 以下命令将缺省队列管理器的所有进程定义的所有适用权限授予用户 JACK 和 JILL。
```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*PRC) USER(JACK JILL) AUT(*ALL)
```
3. 以下命令授予用户 GEORGE 权限以将消息放在队列管理器 TRENT 上的队列 ORDERS 上。
```
GRTMQMAUT OBJ(TRENT) OBJTYPE(*MQM) USER(GEORGE) AUT(*CONNECT) MQMNAME (TRENT)
GRTMQMAUT OBJ(ORDERS) OBJTYPE(*Q) USER(GEORGE) AUT(*PUT) MQMNAME (TRENT)
```

使用 RVKMQMAUT 命令

如果您具有必需的权限，那么可以使用 RVKMQMAUT 命令来除去先前授予的用户概要文件或用户组访问特定对象的权限。 以下示例说明如何使用 RVKMQMAUT 命令:

1.
```
RVKMQMAUT OBJ(RED.LOCAL.QUEUE) OBJTYPE(*LCLQ) USER(GROUPA) +
AUT(*PUT) MQMNAME('saturn.queue.manager')
```
对于 GROUPA，除去了先前示例中授予的将消息放入指定队列的权限。

2.
```
RVKMQMAUT OBJ(PAY*) OBJTYPE(*Q) USER(*PUBLIC) AUT(*GET) +
MQMNAME(PAYROLLQM)
```
将从名称以字符 PAY(由队列管理器 PAYROLLQM 拥有) 开头的任何队列中获取消息的权限从系统的所有用户中除去，除非他们或他们所属的组已单独获得授权。

使用 DSPMQMAUT 命令

显示 MQM 权限 (DSPMQMAUT) 对于指定的对象和用户，命令显示用户对该对象具有的权限列表。 以下示例说明如何使用该命令:

```
DSPMQMAUT OBJ(ADMINNL) OBJTYPE(*NMLIST) USER(JOE) OUTPUT(*PRINT) +
MQMNAME(ADMINQM)
```

使用 RFRMQMAUT 命令

刷新 MQM 安全性 (RFRMQMAUT) 命令使您能够立即更新 OAM 的授权组信息，反映在操作系统级别所作的更改，而无需停止并重新启动队列管理器。 以下示例说明如何使用该命令:

```
RFRMQMAUT MQMNAME(ADMINQM)
```

IBM i IBM i 上的授权规范表

使用此信息来确定在队列对象，进程对象和队列管理器对象上使用特定 API 调用以及这些调用的特定选项所需的授权。

从 第 143 页的表 19 开始的授权规范表精确定义了授权的工作方式以及适用的限制。 这些表适用于以下情况:

- 发出 MQI 调用的应用程序
- 发出 MQSC 命令作为转义 PCF 的管理程序
- 发出 PCF 命令的管理程序

在此部分中，信息显示为一组表，其中指定了以下数据:

要执行的操作
MQI 选项， MQSC 命令或 PCF 命令。

访问控制对象
队列，进程定义，队列管理器，名称列表，通道，客户机连接通道，侦听器，服务或认证信息对象。

需要授权
表示为 MQZAO_ 常量。

在表中，以 MQZAO_ 为前缀的常量对应于特定实体的 **GRTMQMAUT** 和 **RVKMQMAUT** 命令的权限列表中的关键字。 例如， MQZAO_BROWSE 对应于关键字 * 浏览 ; 同样，关键字 MQZAO_SET_ALL_CONTEXT 对应于关键字 *SETALL，依此类推。 这些常量在产品随附的头文件 cmqzc.h 中定义。

MQI 授权

仅当应用程序运行时所使用的用户标识 (或其能够假定的授权) 已被授予相关授权时，才允许应用程序发出特定的 MQI 调用和选项。

四个 MQI 调用需要授权检查 :MQCONN， MQOPEN， MQPUT1 和 MQCLOSE。

对于 MQOPEN 和 MQPUT1，将对要打开的对象的名称进行权限检查，而不是对在解析名称后生成的名称进行权限检查。 例如，可以授予应用程序打开别名队列的权限，而不具有打开别名解析到的基本队列的权限。 该规则是，除非直接打开队列管理器别名定义 (即，其名称出现在对象描述符的 *ObjectName* 字段中)，否则将对名称解析过程中迂到的不是队列管理器别名的第一个定义执行检查。 对于要打开的特定对象，始终需要权限; 在某些情况下，需要通过对队列管理器对象的授权获取其他独立于队列的权限。

第 143 页的表 19， 第 143 页的表 20， 第 144 页的表 21 和 第 145 页的表 22 汇总了每个调用所需的权限。

注: 这些表未提及名称列表，通道，客户机连接通道，侦听器，服务或认证信息对象。 这是因为除了 MQOO_INQUIRE 之外，没有任何权限适用于这些对象，而 MQOO_INQUIRE 的权限与其他对象的权限相同。

表 *19: MQCONN* 调用所需的安全性授权

需要授权:	**队列对象 (第 145 页的『1』)**	**进程对象**	**队列管理器对象**
MQCONN 选项	不适用	不适用	MQZAO_CONNECT

表 *20: MQOPEN* 调用所需的安全性授权

需要授权:	**队列对象 (第 145 页的『1』)**	**进程对象**	**队列管理器对象**
MQOO_INQUIRE	MQZAO_INQUIRE (第 145 页的『2』)	MQZAO_INQUIRE (第 145 页的『2』)	MQZAO_INQUIRE (第 145 页的『2』)
MQOO_BROWSE	MQZAO_BROWSE	不适用	不检查

表 *20: MQOPEN* 调用所需的安全性授权 *(继续)*			
需要授权:	**队列对象 (第 145 页的『1』)**	**进程对象**	**队列管理器对象**
MQOO_INPUT_ *	MQZAO_INPUT	不适用	不检查
MQOO_SAVE_ ALL_CONTEXT **(第 145 页的『3』)**	MQZAO_INPUT	不适用	不适用
MQOO_OUTPUT (正常队列) **(第 145 页的『4』)**	MQZAO_OUTPUT	不适用	不适用
MQOO_PASS_ IDENTITY_CONTEXT **(第 145 页的『5』)**	MQZAO_PASS_ IDENTITY_CONTEXT	不适用	不检查
MQOO_PASS_ALL_ 上下文 **(第 145 页的『5』， 第 145 页的『6』)**	MQZAO_PASS _ALL_CONTEXT	不适用	不检查
MQOO_SET_ IDENTITY_CONTEXT **(第 145 页的『5』， 第 145 页的『6』)**	MQZAO_SET_ IDENTITY_CONTEXT	不适用	MQZAO_SET_ IDENTITY_CONTEXT **(第 145 页的『7』)**
MQOO_SET_ ALL_CONTEXT **(第 145 页的『5』， 第 145 页的『8』)**	MQZAO_SET_ ALL_CONTEXT	不适用	MQZAO_SET_ ALL_CONTEXT **(第 145 页的『7』)**
MQOO_OUTPUT (传输队列) **(第 145 页的『9』)**	MQZAO_SET_ ALL_CONTEXT	不适用	MQZAO_SET_ ALL_CONTEXT **(第 145 页的『7』)**
MQOO_SET	MQZAO_SET	不适用	不检查
MQOO_ALTERNATE_ USER_AUTHORITY	**(第 145 页的『10』)**	**(第 145 页的『10』)**	MQZAO_ALTERNATE_ USER_AUTHORITY **(第 145 页的『10』， 第 145 页的『11』)**

表 *21: MQPUT1* 调用所需的安全性授权			
需要授权:	**队列对象 (第 145 页的『1』)**	**进程对象**	**队列管理器对象**
MQPMO_PASS_ IDENTITY_CONTEXT	MQZAO_PASS_ IDENTITY_CONTEXT **(第 145 页的『12』)**	不适用	不检查
MQPMO_PASS_ALL _CONTEXT	MQZAO_PASS_ ALL_CONTEXT **(第 145 页的『12』)**	不适用	不检查
MQPMO_SET_ IDENTITY_CONTEXT	MQZAO_SET_ IDENTITY_CONTEXT **(第 145 页的『12』)**	不适用	MQZAO_SET_ IDENTITY_CONTEXT **(第 145 页的『7』)**
MQPMO_SET_ ALL_CONTEXT	MQZAO_SET_ ALL_CONTEXT **(第 145 页的『12』)**	不适用	MQZAO_SET_ ALL_CONTEXT **(第 145 页的『7』)**

表 *21: MQPUT1* 调用所需的安全性授权 *(继续)*			
需要授权:	**队列对象 (第 145 页的『1』)**	**进程对象**	**队列管理器对象**
(传输队列) **(第 145 页的『9』)**	MQZAO_SET_ ALL_CONTEXT	不适用	MQZAO_SET_ ALL_CONTEXT **(第 145 页的『7』)**
MQPMO_ALTERNATE_ USER_AUTHORITY	**(第 145 页的『13』)**	不适用	MQZAO_ALTERNATE_ USER_AUTHORITY **(第 145 页的『11』)**

表 *22: MQCLOSE* 调用所需的安全性授权			
需要授权:	**队列对象 (第 145 页的『1』)**	**进程对象**	**队列管理器对象**
MQCO_DELETE	MQZAO_DELETE **(第 145 页的『14』)**	不适用	不适用
MQCO_DELETE _PURGE	MQZAO_DELETE **(第 145 页的『14』)**	不适用	不适用

表的注释:

1. 如果正在打开模型队列:
   - 除了针对要打开的访问类型打开模型队列的权限外，模型队列还需要 MQZAO_DISPLAY 权限。
   - 创建动态队列不需要 MQZAO_CREATE 权限。
   - 用于打开模型队列的用户标识将自动授予所创建动态队列的所有特定于队列的权限 (相当于 MQZAO_ALL)。
2. 根据打开的对象类型，检查队列，进程，名称列表或队列管理器对象。
3. 还必须指定 MQOO_INPUT_ *。 此选项对本地队列，模型队列或别名队列有效。
4. 将对所有输出个案 (注释 第 145 页的『9』中指定的个案除外) 执行此检查。
5. 还必须指定 MQOO_OUTPUT。
6. 此选项还隐含 MQOO_PASS_IDENTITY_CONTEXT。
7. 队列管理器对象和特定队列都需要此权限。
8. 此选项还包含 MQOO_PASS_IDENTITY_CONTEXT， MQOO_PASS_ALL_CONTEXT 和 MQOO_SET_IDENTITY_CONTEXT。
9. 将对具有 *Usage* 队列属性 MQUS_TRANSMISSION 的本地或模型队列执行此检查，并且将直接打开该队列以进行输出。 如果正在打开远程队列 (通过指定远程队列管理器和远程队列的名称，或者通过指定远程队列的本地定义的名称)，那么它不适用。
10. 还必须指定 MQOO_INQUIRE (对于任何对象类型) 或 (对于队列) MQOO_INQUIRE， MQOO_BROWSE， MQOO_INPUT_ *， MQOO_OUTPUT 或 MQOO_SET 中的至少一个。 执行的检查与指定的其他选项一样，将提供的备用用户标识用于特定指定的对象权限，并将当前应用程序权限用于 MQZAO_ALTERNATE_USER_IDENTIFIER 检查。
11. 此授权允许指定任何 *AlternateUserId* 。
12. 如果队列没有 *Usage* 队列属性 MQUS_TRANSMISSION，那么还会执行 MQZAO_OUTPUT 检查。
13. 执行的检查与指定的其他选项一样，使用提供的备用用户标识 (针对指定的队列权限) 和当前应用程序权限 (针对 MQZAO_ALTERNATE_USER_IDENTIFIER 检查)。
14. 仅当以下两个语句都为 true 时，才会执行此检查:
    - 正在关闭并删除永久动态队列。
    - 返回正在使用的对象句柄的 MQOPEN 未创建队列。

否则，将不进行检查。

一般说明:

1. 特殊授权 MQZAO_ALL_MQI 包含与对象类型相关的以下所有授权:
   - MQZAO_CONNECT
   - MQZAO_INQUIRE
   - MQZAO_SET
   - MQZAO_BROWSE
   - MQZAO_INPUT
   - MQZAO_OUTPUT
   - MQZAO_PASS_IDENTITY_CONTEXT
   - MQZAO_PASS_ALL_CONTEXT
   - MQZAO_SET_IDENTITY_CONTEXT
   - MQZAO_SET_ALL_CONTEXT
   - MQZAO_ALTERNATE_USER_AUTHORITY
2. MQZAO_DELETE (请参阅注释 第 145 页的『14』) 和 MQZAO_DISPLAY 被归为管理权限。 因此，它们不包含在 MQZAO_ALL_MQI 中。
3. 无检查 表示不执行授权检查。
4. 不适用 表示授权检查与此操作无关。 例如，不能对流程对象发出 MQPUT 调用。

IBM i *IBM i 上对 PCF 进行转义的 MQSC 命令的权限*

这些权限允许用户作为转义 PCF 消息发出管理命令。 这些方法允许程序将管理命令作为消息发送到队列管理器，以代表该用户执行。

本部分概述了 Escape PCF 中包含的每个 MQSC 命令所需的权限。

不适用 表示授权检查与此操作无关。

用于运行提交命令的程序的用户标识还必须具有以下权限:

- 队列管理器的 MQZAO_CONNECT 权限
- 队列管理器上的 DISPLAY 权限，以便执行 PCF 命令
- 在 Escape PCF 命令的文本中发出 MQSC 命令的权限

ALTER 对象

Object	**需要授权**
队列	MQZAO_CHANGE
Topic	MQZAO_CHANGE
进程	MQZAO_CHANGE
队列管理器	MQZAO_CHANGE
名称列表	MQZAO_CHANGE
认证信息	MQZAO_CHANGE
通道	MQZAO_CHANGE
客户机连接通道	MQZAO_CHANGE
侦听器	MQZAO_CHANGE
服务	MQZAO_CHANGE

CLEAR 对象

Object	需要授权
队列	MQZAO_CLEAR
Topic	MQZAO_CLEAR
进程	不适用
队列管理器	不适用
名称列表	不适用
认证信息	不适用
通道	不适用
客户机连接通道	不适用
侦听器	不适用
服务	不适用

DEFINE 对象 NOREPLACE (第 150 页的『1』)

Object	需要授权
队列	MQZAO_CREATE (第 150 页的『2』)
Topic	MQZAO_CREATE (第 150 页的『2』)
进程	MQZAO_CREATE (第 150 页的『2』)
队列管理器	不适用
名称列表	MQZAO_CREATE (第 150 页的『2』)
认证信息	MQZAO_CREATE (第 150 页的『2』)
通道	MQZAO_CREATE (第 150 页的『2』)
客户机连接通道	MQZAO_CREATE (第 150 页的『2』)
侦听器	MQZAO_CREATE (第 150 页的『2』)
服务	MQZAO_CREATE (第 150 页的『2』)

DEFINE 对象 REPLACE (第 150 页的『1』， 第 150 页的『3』)

Object	需要授权
队列	MQZAO_CHANGE
Topic	MQZAO_CHANGE
进程	MQZAO_CHANGE
队列管理器	不适用
名称列表	MQZAO_CHANGE
认证信息	MQZAO_CHANGE
通道	MQZAO_CHANGE
客户机连接通道	MQZAO_CHANGE
侦听器	MQZAO_CHANGE

Object	需要授权
服务	MQZAO_CHANGE

DELETE 对象

Object	需要授权
队列	MQZAO_DELETE
Topic	MQZAO_DELETE
进程	MQZAO_DELETE
队列管理器	不适用
名称列表	MQZAO_DELETE
认证信息	MQZAO_DELETE
通道	MQZAO_DELETE
客户机连接通道	MQZAO_DELETE
侦听器	MQZAO_DELETE
服务	MQZAO_DELETE

DISPLAY 对象

Object	需要授权
队列	MQZAO_DISPLAY
Topic	MQZAO_DISPLAY
进程	MQZAO_DISPLAY
队列管理器	MQZAO_DISPLAY
名称列表	MQZAO_DISPLAY
认证信息	MQZAO_DISPLAY
通道	MQZAO_DISPLAY
客户机连接通道	MQZAO_DISPLAY
侦听器	
服务	

Ping 通道

Object	需要授权
队列	不适用
Topic	不适用
进程	不适用
队列管理器	不适用
名称列表	不适用
认证信息	不适用
通道	MQZAO_CONTROL

Object	需要授权
客户机连接通道	不适用
侦听器	不适用
服务	不适用

重置通道

Object	需要授权
队列	不适用
Topic	不适用
进程	不适用
队列管理器	不适用
名称列表	不适用
认证信息	不适用
通道	MQZAO_CONTROL_EXTENDED
客户机连接通道	不适用
侦听器	不适用
服务	不适用

解析通道

Object	需要授权
队列	不适用
Topic	不适用
进程	不适用
队列管理器	不适用
名称列表	不适用
认证信息	不适用
通道	MQZAO_CONTROL_EXTENDED
客户机连接通道	不适用
侦听器	不适用
服务	不适用

START 对象

Object	需要授权
队列	不适用
Topic	不适用
进程	不适用
队列管理器	不适用
名称列表	不适用

Object	需要授权
认证信息	不适用
通道	MQZAO_CONTROL
客户机连接通道	不适用
侦听器	MQZAO_CONTROL
服务	MQZAO_CONTROL

STOP 对象

Object	需要授权
队列	不适用
Topic	不适用
进程	不适用
队列管理器	不适用
名称列表	不适用
认证信息	不适用
通道	MQZAO_CONTROL
客户机连接通道	不适用
侦听器	MQZAO_CONTROL
服务	MQZAO_CONTROL

注:

1. 对于 DEFINE 命令，如果指定了 LIKE 对象，那么也需要 MQZAO_DISPLAY 权限，或者在相应的 SYSTEM.DEFAULT.xxx 对象 (如果省略了 LIKE)。
2. MQZAO_CREATE 权限并非特定于特定对象或对象类型。 通过在 GRTMQMAUT 命令上指定对象类型 QMGR ，为指定队列管理器的所有对象授予创建权限。
3. 如果要替换的对象已存在，那么此选项适用。 如果不存在，那么检查与 DEFINE 对象 NOREPLACE 相同。

IBM i *IBM i 上 PCF 命令的权限*

这些权限允许用户作为 PCF 命令发出管理命令。 这些方法允许程序将管理命令作为消息发送到队列管理器，以代表该用户执行。

本部分概述了每个 PCF 命令所需的权限。

无检查 表示不执行授权检查; 不适用 表示授权检查与此操作无关。

用于运行提交命令的程序的用户标识还必须具有以下权限:

- 队列管理器的 MQZAO_CONNECT 权限
- 队列管理器上的 DISPLAY 权限，以便执行 PCF 命令

特殊授权 MQZAO_ALL_ADMIN 包含以下权限:

- MQZAO_CHANGE
- MQZAO_CLEAR
- MQZAO_DELETE
- MQZAO_DISPLAY

- MQZAO_CONTROL
- MQZAO_CONTROL_EXTENDED

未包含 MQZAO_CREATE ，因为它不是特定于特定对象或对象类型

更改 对象

Object	**需要授权**
队列	MQZAO_CHANGE
Topic	MQZAO_CHANGE
进程	MQZAO_CHANGE
队列管理器	MQZAO_CHANGE
名称列表	MQZAO_CHANGE
认证信息	MQZAO_CHANGE
通道	MQZAO_CHANGE
客户机连接通道	MQZAO_CHANGE
侦听器	MQZAO_CHANGE
服务	MQZAO_CHANGE

清除 对象

Object	**需要授权**
队列	MQZAO_CLEAR
Topic	MQZAO_CLEAR
进程	不适用
队列管理器	不适用
名称列表	不适用
认证信息	不适用
通道	不适用
客户机连接通道	不适用
侦听器	不适用
服务	不适用

复制 *object* (不替换) (第 156 页的『1』)

Object	**需要授权**
队列	MQZAO_CREATE (**第 156 页的『2』**)
Topic	MQZAO_CREATE (**第 156 页的『2』**)
进程	MQZAO_CREATE (**第 156 页的『2』**)
队列管理器	不适用
名称空间 MQZAO_CREATE	MQZAO_CREATE (**第 156 页的『2』**)
认证信息	MQZAO_CREATE (**第 156 页的『2』**)
通道	MQZAO_CREATE (**第 156 页的『2』**)

Object	需要授权
客户机连接通道	MQZAO_CREATE (第 156 页的『2』)
侦听器	MQZAO_CREATE (第 156 页的『2』)
服务	MQZAO_CREATE (第 156 页的『2』)

复制 对象 (带有替换) (第 156 页的『1』， 第 156 页的『4』)

Object	需要授权
队列	MQZAO_CHANGE
Topic	MQZAO_CHANGE
进程	MQZAO_CHANGE
队列管理器	不适用
名称列表	MQZAO_CHANGE
认证信息	MQZAO_CHANGE
通道	MQZAO_CHANGE
客户机连接通道	MQZAO_CHANGE
侦听器	MQZAO_CHANGE
服务	MQZAO_CHANGE

创建 对象 (不替换) (第 156 页的『3』)

Object	需要授权
队列	MQZAO_CREATE (第 156 页的『2』)
Topic	MQZAO_CREATE (第 156 页的『2』)
进程	MQZAO_CREATE (第 156 页的『2』)
队列管理器	不适用
名称列表	MQZAO_CREATE (第 156 页的『2』)
认证信息	MQZAO_CREATE (第 156 页的『2』)
通道	MQZAO_CREATE (第 156 页的『2』)
客户机连接通道	MQZAO_CREATE (第 156 页的『2』)
侦听器	MQZAO_CHANGE
服务	MQZAO_CHANGE

创建 对象 (带有替换) (第 156 页的『3』， 第 156 页的『4』)

Object	需要授权
队列	MQZAO_CHANGE
Topic	MQZAO_CHANGE
进程	MQZAO_CHANGE
队列管理器	不适用
名称列表	MQZAO_CHANGE

Object	需要授权
认证信息	MQZAO_CHANGE
通道	MQZAO_CHANGE
客户机连接通道	MQZAO_CHANGE
侦听器	MQZAO_CHANGE
服务	MQZAO_CHANGE

删除 对象

Object	需要授权
队列	MQZAO_DELETE
Topic	MQZAO_DELETE
进程	MQZAO_DELETE
队列管理器	MQZAO_DELETE
名称列表	MQZAO_DELETE
认证信息	MQZAO_DELETE
通道	MQZAO_DELETE
客户机连接通道	MQZAO_DELETE
侦听器	MQZAO_DELETE
服务	MQZAO_DELETE

查询 对象

Object	需要授权
队列	MQZAO_DISPLAY
Topic	MQZAO_DISPLAY
进程	MQZAO_DISPLAY
队列管理器	MQZAO_DISPLAY
名称列表	MQZAO_DISPLAY
认证信息	MQZAO_DISPLAY
通道	MQZAO_DISPLAY
客户机连接通道	MQZAO_DISPLAY
侦听器	MQZAO_DISPLAY
服务	MQZAO_DISPLAY

查询 *object* 名称

Object	需要授权
队列	不检查
Topic	不检查
进程	不检查

Object	需要授权
队列管理器	不检查
名称列表	不检查
认证信息	不检查
通道	不检查
客户机连接通道	不检查
侦听器	不检查
服务	不检查

Ping 通道

Object	需要授权
队列	不适用
Topic	不适用
进程	不适用
队列管理器	不适用
名称列表	不适用
认证信息	不适用
通道	MQZAO_CONTROL
客户机连接通道	不适用
侦听器	不适用
服务	不适用

重置通道

Object	需要授权
队列	不适用
Topic	不适用
进程	不适用
队列管理器	不适用
名称列表	不适用
认证信息	不适用
通道	MQZAO_CONTROL_EXTENDED
客户机连接通道	不适用
侦听器	不适用
服务	不适用

重置队列统计信息

Object	需要授权
队列	MQZAO_DISPLAY 和 MQZAO_CHANGE

Object	需要授权
Topic	不适用
进程	不适用
队列管理器	不适用
名称列表	不适用
认证信息	不适用
通道	不适用
客户机连接通道	不适用
侦听器	
服务	

解析通道

Object	需要授权
队列	不适用
Topic	不适用
进程	不适用
队列管理器	不适用
名称列表	不适用
认证信息	不适用
通道	MQZAO_CONTROL_EXTENDED
客户机连接通道	不适用
侦听器	不适用
服务	不适用

启动通道

Object	需要授权
队列	不适用
Topic	不适用
进程	不适用
队列管理器	不适用
名称列表	不适用
认证信息	不适用
通道	MQZAO_CONTROL
客户机连接通道	不适用
侦听器	不适用
服务	不适用

停止通道

Object	需要授权
队列	不适用
Topic	不适用
进程	不适用
队列管理器	不适用
名称列表	不适用
认证信息	不适用
通道	MQZAO_CONTROL
客户机连接通道	不适用
侦听器	不适用
服务	不适用

注:

1. 对于 "复制" 命令， "源" 对象还需要 MQZAO_DISPLAY 权限。
2. MQZAO_CREATE 权限并非特定于特定对象或对象类型。 通过在 GRTMQMAUT 命令上指定对象类型 QMGR ，为指定队列管理器的所有对象授予创建权限。
3. 对于 "创建" 命令，相应的 SYSTEM.DEFAULT.* 对象。
4. 如果要替换的对象已存在，那么此选项适用。 如果不存在，那么此检查适用于 "复制" 或 "创建" 而不进行替换。

IBM i IBM i 上的通用 OAM 概要文件

对象权限管理器 (OAM) 通用概要文件使您能够一次性设置用户对许多对象的权限，而不必在创建对象时针对每个单独的对象发出单独的 **GRTMQMAUT** 命令。 在 **GRTMQMAUT** 命令中使用通用概要文件使您能够为所有将来创建的适合该概要文件的对象设置通用权限。

本节的其余部分更详细地描述了通用概要文件的使用:

- 第 156 页的『使用通配符』
- 第 157 页的『概要文件优先级』

使用通配符

使概要文件通用的是在概要文件名称中使用特殊字符 (通配符)。 例如，问号 (?) 通配符与名称中的任何单个字符匹配。 因此，如果指定 ABC.?EF，那么对该概要文件的授权将应用于使用名称 ABC.DEF， ABC.CEF 和 ABC.BEF 等创建的任何对象。

可用的通配符包括:

?

使用问号（?）代替任何单个字符。 例如， AB.?D 将应用于对象 AB.CD， AB.ED 和 AB.FD。

*****

使用星号 (*) 作为:

- 概要文件名称中的 限定符 ，用于与对象名中的任何一个限定符匹配。 限定符是用句点定界的对象名的一部分。 例如，在 ABC.DEF.GHI 中，限定符是 ABC、DEF 和 GHI。

  例如， ABC.*.JKL 将应用于对象 ABC.DEF.JKL 和 ABC.GHI.JKL。 (请注意，它 **不** 适用于 ABC.JKL ; 在此上下文中使用的 * 始终指示一个限定符。)
- 概要文件名称中限定符内的字符，用于与对象名称中限定符内的零个或多个字符匹配。

例如， ABC.DE*.JKL 将应用于对象 ABC.DE.JKL， ABC.DEF.JKL 和 ABC.DEGH.JKL。

**

在概要文件名称中使用双星号 (**) ***once*** 作为:

- 要与所有对象名匹配的整个概要文件名称。 例如，如果使用关键字 OBJTYPE （*PRC）来标识进程，然后使用 ** 作为概要文件名称，那么将更改所有进程的权限。
- 作为概要文件名称中的开头，中间或结尾限定符，以匹配对象名称中的零个或多个限定符。 例如，**.ABC 标识具有最终限定符 ABC 的所有对象。

概要文件优先级

在使用通用概要文件时要了解的一个重要问题是，在确定要应用于要创建的对象的权限时，概要文件的优先级。 例如，假设您已发出以下命令:

```
GRTMQMAUT OBJ(AB.*) OBJTYPE(*Q) USER(FRED) AUT(*PUT) MQMNAME(MYQMGR)
GRTMQMAUT OBJ(AB.C*) OBJTYPE(*Q) USER(FRED) AUT(*GET) MQMNAME(MYQMGR)
```

第一个名称与概要文件 AB. * 匹配的主体 FRED 的所有队列都具有放置权限; 第二个队列授予对与概要文件 AB.C*。

假设您现在创建名为 AB.CD。 根据通配符匹配的规则， GRTMQMAUT 可以应用于该队列。 那么，它是放了还是得到了权威?

要查找答案，请应用以下规则: 每当多个概要文件可以应用于某个对象时， **只有最具体的应用**。 应用此规则的方法是将概要文件名称从左到右进行比较。 无论它们在何处不同，非通用字符都比通用字符更具体。 因此，在上一个示例中，队列 AB.CD 具有 **get** 权限 (AB.C* 比 AB. * 更具体。

比较通用字符时， 特异性 的顺序为:

1. ?
2. *
3. **

IBM i 在 IBM i 上指定已安装的授权服务

您可以指定要使用的授权服务组件。

GRTMQMAUT 和 **RVKMQMAUT** 上的参数 **Service Component name** 允许您指定已安装的授权服务组件的名称。

在初始面板上选择 **F24** ，然后在任一命令的下一个面板上选择 **F9=All 参数** ，允许您指定已安装的授权组件 (*DFT) 或在队列管理器的 qm.ini 文件的 "服务" 节中指定的所需授权服务组件的名称。

DSPMQMAUT 还具有此额外参数。 此参数允许您搜索所有已安装的授权组件 (*DFT) 或指定的授权服务组件名称，以查找指定的对象名，对象类型和用户

IBM i 在 IBM i 上使用和不使用权限概要文件

使用此信息可了解如何使用权限概要文件以及如何在没有权限概要文件的情况下工作。

您可以使用权限概要文件 (如 第 157 页的『使用权限概要文件』中所述) 或不使用权限概要文件 (如此处所述):

要在没有权限概要文件的情况下工作，请使用 *NONE 作为 **GRTMQMAUT** 上的权限参数来创建没有权限的概要文件。 这将使任何现有概要文件保持不变。

在 **RVKMQMAUT** 上，使用 *REMOVE 作为 "权限" 参数来除去现有权限概要文件。

使用权限概要文件

有两个命令与权限概要分析相关联:

- **WRKMQMAUT**
- **WRKMQMAUTD**

您可以通过以下方法直接从命令行或从 WRKMQM 面板访问这些命令:

1. 输入队列管理器名称并按 Enter 键以访问 **WRKMQM** 结果面板。
2. 在此面板上选择 F23=More options 。

选项 24 选择 **WRKMQMAUT** 命令 的结果面板，选项 25 选择 **WRKMQMAUTI** 命令 (与 SSL 绑定层配合使用)。

WRKMQMAUT

此命令允许您使用保存在权限队列中的权限数据。

注: 要运行此命令，您必须对队列管理器具有 *connect 和 *admdsp 权限。 但是，要创建或删除概要文件，您需要 QMQMADM 权限。

如果将信息输出到屏幕，那么将显示权限概要文件名称及其类型的列表。 如果打印输出，您将收到所有权限数据，注册用户及其权限的详细列表。

在此面板上输入对象或概要文件名称，然后按 ENTER 以将您转至 **WRKMQMAUT** 的结果面板。

如果选择 4=Delete，那么将转至新面板，您可以在该面板中确认要删除向您指定的通用权限概要文件名称注册的所有用户名。 此选项对所有用户运行带有选项 *REMOVE 的 **RVKMQMAUT** ，并且 **仅** 应用于通用概要文件名称。

如果选择 12=Work with profile ，请转至 **WRKMQMAUTD** 命令结果面板，如 第 158 页的『WRKMQMAUTD』中所述。

WRKMQMAUTD

此命令允许您显示使用特定权限概要文件名称和对象类型注册的所有用户。 要运行此命令，您必须对队列管理器具有 *connect 和 *admdsp 权限。 但是，要授予，运行，创建或删除概要文件，您需要 QMQMADM 权限。

从初始输入面板中选择 F24=More keys ，后跟选项 F9=All Parameters 将显示 **GRTMQMAUT** 和 **RVKMQMAUT** 的服务组件名称。

注: F11=Display Object Authorizations 键在以下类型的权限之间切换:

- 对象权限
- 上下文授权
- MQI 授权

屏幕上的选项包括:

2=Grant
将您转至 **GRTMQMAUT** 面板以添加到当前权限。

3=Revoke
将您转至 **RVKMQMAUT** 面板以除去某些当前定义

4=Delete
将您转至允许您删除指定用户的权限数据的面板。 这将使用选项 *REMOVE 运行 **RVKMQMAUT** 。

5=Display
将您转至现有 **DSPMQMAUT** 命令

F6=Create
转至允许您创建概要文件权限记录的 **GRTMQMAUT** 面板。

IBM i IBM i 上的对象权限管理器准则

有关使用对象权限管理器 (OAM) 的其他提示和技巧

限制对敏感操作的访问

某些操作很敏感; 将其限制为特权用户。 例如

- 访问某些特殊队列，例如传输队列或命令队列 SYSTEM.ADMIN.COMMAND.QUEUE
- 运行使用完整 MQI 上下文选项的程序
- 创建和复制应用程序队列

队列管理器目录

包含队列和其他队列管理器数据的目录和库是产品的专用目录和库。 请勿使用标准操作系统命令来授予或撤销对 MQI 资源的权限。

队列

对动态队列的权限基于但不一定与从中派生该动态队列的模型队列的权限相同。

对于别名队列和远程队列，授权是对象本身的授权，而不是别名或远程队列解析到的队列。 可以授权用户概要文件访问别名队列，该别名队列解析为用户概要文件没有访问许可权的本地队列。

将创建队列的权限限制为特权用户。 如果不存在，那么用户可以通过创建别名来绕过正常访问控制。

备用用户权限

备用用户权限控制一个用户概要文件在访问 IBM MQ 对象时是否可以使用另一个用户概要文件的权限。 当服务器接收来自程序的请求，并且服务器希望确保程序对该请求具有必需的权限时，此技术至关重要。 服务器可能具有必需的权限，但它需要知道程序是否具有它所请求的操作的权限。

例如:

- 在用户概要文件 PAYSERV 下运行的服务器程序从用户概要文件 USER1 放入队列的队列中检索请求消息。
- 当服务器程序获取请求消息时，它将处理请求并将应答重新放入与请求消息一起指定的应答队列中。
- 服务器可以指定其他用户概要文件 (在本例中为 USER1) ，而不是使用自己的用户概要文件 (PAYSERV) 来授权打开应答队列。 在此示例中，您可以使用备用用户权限来控制是否允许 PAYSERV 在打开应答队列时将 USER1 指定为备用用户概要文件。

在对象描述符的 *AlternateUserId* 字段上指定备用用户概要文件。

注: 您可以在任何 IBM MQ 对象上使用备用用户概要文件。 使用备用用户概要文件不会影响任何其他资源管理器使用的用户概要文件。

上下文权限

上下文是适用于特定消息的信息，包含在消息描述符 MQMD 中， MQMD 是消息的一部分。

有关与上下文相关的消息描述符字段的描述，请参阅 MQMD-消息描述符。

有关上下文选项的信息，请参阅 消息上下文。

远程安全注意事项

对于远程安全性，请考虑:

放入权限
为了确保队列管理器之间的安全性，您可以指定通道接收从另一个队列管理器发送的消息时使用的放置权限。

此参数仅对 RCVR ， RQSTR 或 CLUSRCVR 通道类型有效。 指定通道属性 PUTAUT ，如下所示:

DEF
缺省用户概要文件。 这是运行消息通道代理程序的 QMQM 用户概要文件。

CTX
消息上下文中的用户概要文件。

传输队列
队列管理器会自动将远程消息放在传输队列上; 不需要特殊权限。 但是，将消息直接放入传输队列需要特殊授权。

通道出口
通道出口可用于添加安全性。

通道认证记录
用于对授予在通道级别连接系统的访问权进行更精确的控制。

有关远程安全性的更多信息，请参阅 第 96 页的『通道授权』。

使用 SSL/TLS 保护通道

传输层安全性 (TLS) 协议提供通道安全性，防止窃听，篡改和假冒。 IBM MQ 对 TLS 的支持使您能够在通道定义上指定特定通道使用 TLS 安全性。 您还可以指定所需安全性的详细信息，例如要使用的加密算法。

IBM MQ 中的 TLS 支持使用队列管理器 认证信息对象 以及各种 CL 和 MQSC 命令以及队列管理器和通道参数，这些参数详细定义了所需的 TLS 支持。

以下 CL 命令支持 TLS:

WRKMQMAUTI
使用认证信息对象的属性。

CHGMQMAUTI
修改认证信息对象的属性。

CRTMQMAUTI
创建认证信息对象。

CPYMQMAUTI
通过复制现有认证信息对象来创建该对象。

DLTMQMAUTI
删除认证信息对象。

DSPMQMAUTI
显示特定认证信息对象的属性。

有关使用 TLS 的通道安全性的概述，请参阅

- 使用 TLS 保护通道

有关与 TLS 关联的 PCF 命令的详细信息，请参阅

- 更改，复制和创建认证信息对象
- 删除认证信息对象
- 查询认证信息对象

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 161.

Table 23. RACF classes used by IBM MQ

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

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

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

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

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

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

z/OS RACF profiles

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

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

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

```
RDEFINE MQQUEUE QSG1.QUEUE_FOR_SUBSCRIBER_LIST
```

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

```
RDEFINE MQQUEUE STCD.QUEUE_FOR_LOST_CARD_LIST
```

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

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

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

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

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

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

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

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

To use mixed-case profiles and mixed-case RACF classes you must follow the steps described in “Migrating a z/OS queue manager to mixed-case security” on page 238.

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 164. 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 164.

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 165 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.

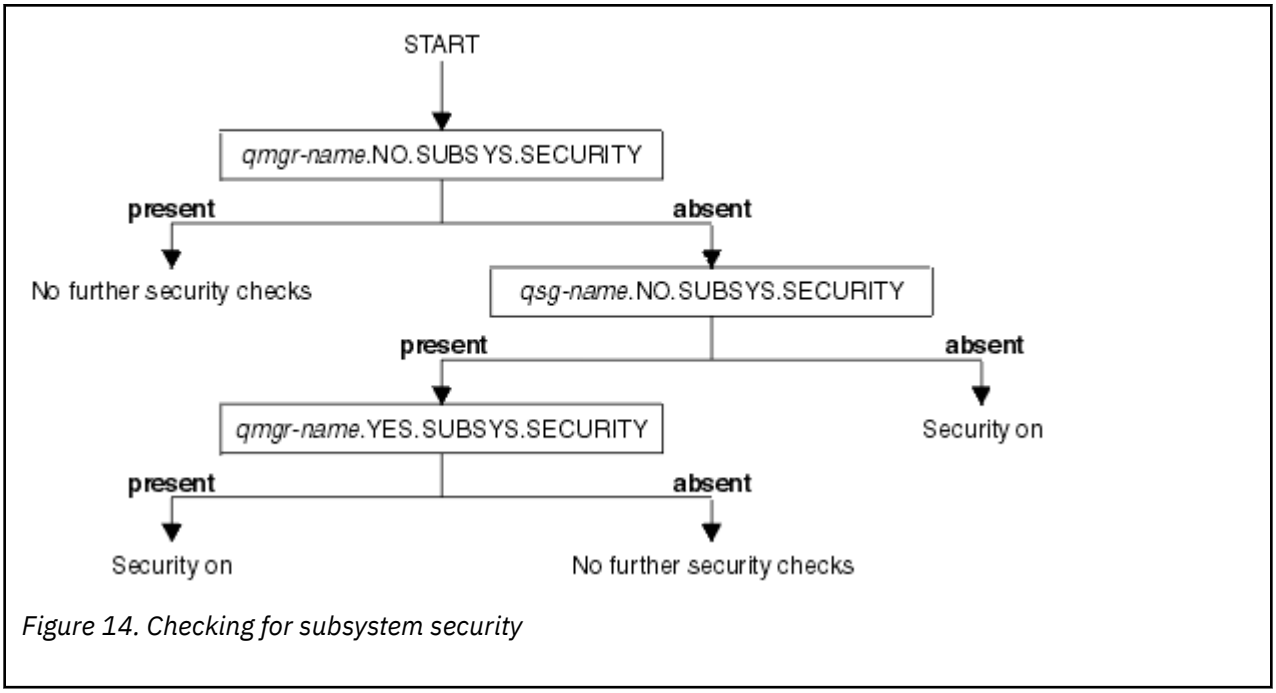


Figure 14. Checking for subsystem security

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 166 and Figure 16 on page 166 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

Table 25. Switch profiles for queue sharing group or queue manager level security (continued)	
Switch profile name	**Type of resource or checking that is controlled**
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.

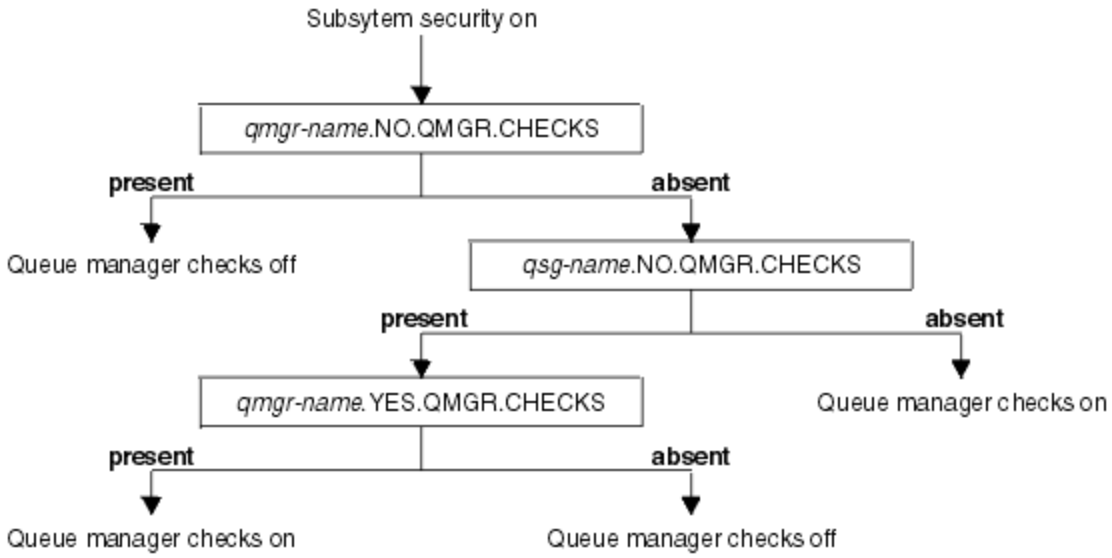


Figure 15. Checking for queue manager level security

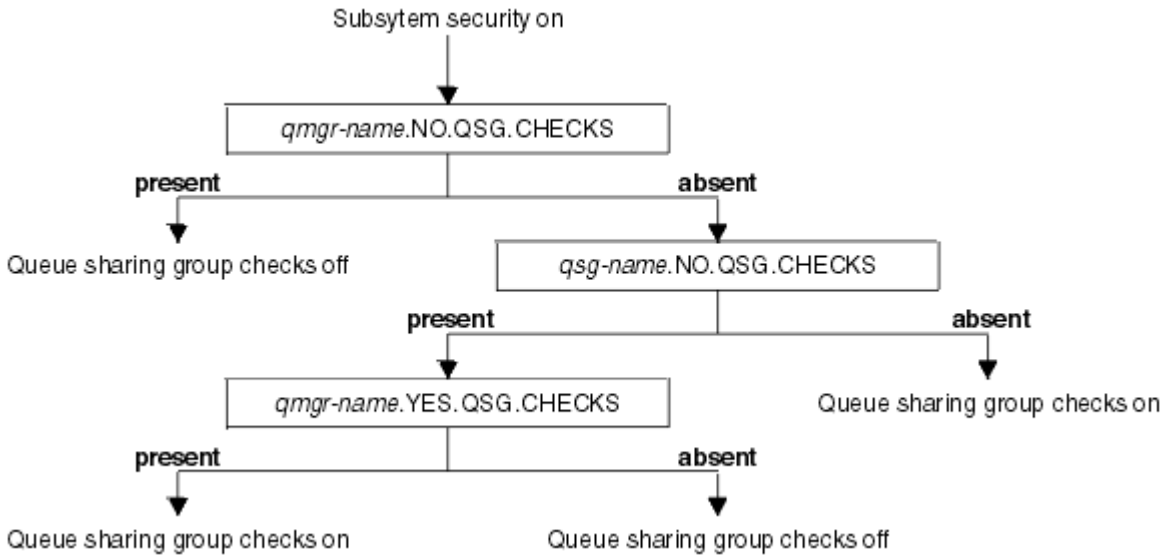


Figure 16. Checking for queue sharing group level security

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 167, Table 27 on page 167, Table 28 on page 167, and Table 29 on page 168 show the sets of combinations of switch settings that are valid for each type of security level.

Table 26. Valid security switch combinations for queue manager level security

Combinations
qmgr-name.NO.QSG.CHECKS
qsg-name.NO.QSG.CHECKS
qmgr-name.NO.QSG.CHECKS qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS
qsg-name.NO.QSG.CHECKS qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS

Table 27. Valid security switch combinations for queue sharing group level security

Combinations
qmgr-name.NO.QMGR.CHECKS
qsg-name.NO.QMGR.CHECKS
qmgr-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS

Table 28. Valid security switch combinations for queue manager and queue sharing group level security

Combinations
qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS No QSG.* profiles defined
No QMGR.* profiles defined qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS
No profiles for either switch defined

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

z/OS *Resource level checks*

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

Table 30 on page 168 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 220 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 210.
2. Resource level security (RESLEVEL) checks are also made at connection time. For details, see “RESLEVEL 安全概要文件” on page 204.

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 176 and “Considerations for model queues” on page 177 .

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 190 and "Profiles for alternate user security" on page 188. 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 181.

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 210 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 “远程消息传递的安全性” on page 85.

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 180 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 180.

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 181	-	-	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 186.

When you consider alias queues used for destination queues for PUT or GET restrictions, see “Considerations for alias queues” on page 176.

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 186.

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 188.

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 210. 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 181.

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 210 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 218.

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 213 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 174), and alternate user security (see "Profiles for alternate user security" on page 188). 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 181.

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 192.

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 193 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 198 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 198	hlq.ALTER.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
ALTER QMGR	hlq.ALTER.QMGR	ALTER	No check	-
ALTER QMODEL“5” on page 198	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 197	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 198	hlq.DEFINE.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
DEFINE QMODEL“5” on page 198	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 197	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 197	hlq.DISPLAY.CONN	READ	No check	-
DISPLAY GROUP	hlq.DISPLAY.GROUP	READ	No check	-
DISPLAY LOG “1” on page 197	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 197	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 197	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 197	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 197	-	-	-
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 “发布/预订安全性” on page 425
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 201	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 201	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 201	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 “发布/预订安全性” on page 425
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 201 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 202 shows, for each IBM MQ PCF command, the command security profiles required, and the corresponding access level for each profile in the MQCMDS class needed by the IBM MQ Console.

Table 51. IBM MQ Console PCF commands, profiles, and their access levels

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
Change Authentication Information Object	hlq.ALTER.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Change Channel	hlq.ALTER.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Change Queue	hlq.ALTER.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Change Queue Manager	hlq.ALTER.QMGR	ALTER	No check	-
Change Topic	hlq.ALTER.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Clear Queue	hlq.CLEAR.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
Create Authentication Information Object	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Create Channel	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Create Queue	hlq.DEFINE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Create Subscription	hlq.DEFINE.SUB	ALTER	No check	-
Create Topic	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Delete Authentication Information Object	hlq.DELETE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Delete Channel	hlq.DELETE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Delete Queue	hlq.DELETE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Delete Subscription	hlq.DELETE.SUB	ALTER	No check	-
Delete Topic	hlq.DELETE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Inquire Authentication Information Object	hlq.DISPLAY.AUTHINFO	READ	No check	-
Inquire Authentication Information Object Names	hlq.DISPLAY.AUTHINFO	READ	No check	-
Inquire Channel	hlq.DISPLAY.CHANNEL	READ	No check	-
Inquire Channel Authentication Records	hlq.DISPLAY.CHLAUTH	READ	No check	-
Inquire Channel Initiator	hlq.DISPLAY.CHINIT	READ	No check	-
Inquire Channel Names	hlq.DISPLAY.CHANNEL	READ	No check	-
Inquire Channel Status	hlq.DISPLAY.CHSTATUS	READ	No check	-
Inquire Queue	hlq.DISPLAY.QUEUE	READ	No check	-
Inquire Queue Manager	hlq.DISPLAY.QMGR	READ	No check	-
Inquire Queue Names	hlq.DISPLAY.QUEUE	READ	No check	-
Inquire Queue Status	hlq.DISPLAY.QSTATUS	READ	No check	-
Inquire Subscription	hlq.INQUIRE.SUB	READ	No check	-

Table 51. IBM MQ Console PCF commands, profiles, and their access levels (continued)				
Command	**Command profile for MQCMDS**	**Access level for MQCMDS**	**Command resource profile for MQADMIN or MXADMIN**	**Access level for MQADMIN or MXADMIN**
Inquire Subscription Status	hlq.INQUIRE.SBSTATUS	READ	No check	-
Inquire Topic	hlq.DISPLAY.TOPIC	READ	No check	-
Inquire Topic Names	hlq.DISPLAY.TOPIC	READ	No check	-
Inquire Topic Status	hlq.DISPLAY.TPSTATUS	READ	No check	-
Ping Channel	hlq.PING.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Refresh Cluster	hlq.REFRESH.CLUSTER	ALTER	No check	-
Refresh Security	hlq.REFRESH.SECURITY	ALTER	No check	-
Reset Channel	hlq.RESET.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Resolve Channel	hlq.RESOLVE.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Set Channel Authentication Record	hlq.SET.CHLAUTH	CONTROL	No check	-
Start Channel	hlq.START.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Stop Channel	hlq.STOP.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL

z/OS *Profiles for command resource security*

If you have not defined the command resource security switch profile, because you want security checking for resources associated with commands, you must add resource profiles for each resource to the appropriate class. The same security profiles control both MQSC and PCF commands.

If you have not defined the command resource security switch profile, `hlq.NO.CMD.RESC.CHECKS`, because you want security checking for resources associated with commands, you must:

- Add a resource profile in the **MQADMIN** class, if using uppercase profiles, for each resource.
- Add a resource profile in the **MXADMIN** class, if using mixed case profiles, for each resource.

The same security profiles control both MQSC and PCF commands.

Profiles for command resource security checking take the form:

```
hlq.type.resourcename
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name).

A profile prefixed by the queue manager name controls access to the resources associated with commands on that queue manager. A profile prefixed by the queue sharing group name controls access to the resources associated with commands on all queue managers within the queue sharing group. This access can be overridden on an individual queue manager by defining a queue manager level profile for that command resource on that queue manager.

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

For example, the RACF profile name for command resource security checking against the model queue CREDIT.WORTHY in subsystem CSQ1 is:

```
CSQ1.QUEUE.CREDIT.WORTHY
```

Because the profiles for all types of command resource are held in the MQADMIN class, the "type" part of the profile name is needed in the profile to distinguish between resources of different types that have the same name. The "type" part of the profile name can be CHANNEL, QUEUE, TOPIC, PROCESS, or NAMELIST. For example, a user might be authorized to define hlq.QUEUE.PAYROLL.ONE, but not authorized to define hlq.PROCESS.PAYROLL.ONE

If the resource type is a queue, and the profile is a queue sharing group level profile, it controls access to one or more local queues within the queue sharing group, or access to a single shared queue from any queue manager in the queue sharing group.

MQSC commands, profiles, and their access levels shows, for each IBM MQ MQSC command, the profiles required for command security checking to be carried out, and the corresponding access level for each profile in the MQCMDS class.

PCF commands, profiles, and their access levels shows, for each IBM MQ PCF command, the profiles required for command security checking to be carried out, and the corresponding access level for each profile in the MQCMDS class.

z/OS *Command resource security checking for alias queues and remote queues*

Alias queue and remote queues both provide indirection to another queue. Additional points apply when you consider security checking for these queues.

Alias queues

When you define an alias queue, command resource security checks are only performed against the name of the alias queue, not against the name of the target queue to which the alias resolves.

Alias queues can resolve to both local and remote queues. If you do not want to permit users access to certain local or remote queues, you must do both of the following:

1. Do not allow the users access to these local and remote queues.
2. Restrict the users from being able to define aliases for these queues. That is, prevent them from being able to issue DEFINE QALIAS and ALTER QALIAS commands.

Remote queues

When you define a remote queue, command resource security checks are performed only against the name of the remote queue. No checks are performed against the names of the queues specified in the RNAME or XMITQ attributes in the remote queue object definition.

z/OS RESLEVEL 安全概要文件

您可以在 MQADMIN 或 MXADMIN 类中定义特殊概要文件，以控制为 API 资源安全性检查的用户标识数。此概要文件称为 RESLEVEL 概要文件。 此概要文件如何影响 API-资源安全性取决于您访问 IBM MQ 的方式。

当应用程序尝试连接到 IBM MQ 时， IBM MQ 会检查与连接关联的用户标识对 MQADMIN 或 MXADMIN 类中的概要文件具有的访问权:

```
hlq.RESLEVEL
```

其中 `hlq` 可以是 `ssid` (子系统标识) 或 `qsg` (队列共享组标识)。

与每种连接类型关联的用户标识为:

- 批处理连接的连接任务的用户标识
- CICS 连接的 CICS 地址空间用户标识
- IMS 连接的 IMS 区域地址空间用户标识
- 通道启动程序连接的通道启动程序地址空间用户标识

注意: RESLEVEL 是一个非常强大的选项; 它可能导致绕过特定连接的所有资源安全性检查。

如果未定义 RESLEVEL 概要文件，那么必须注意 MQADMIN 类中没有其他概要文件与 hlq.RESLEVEL 匹配。 例如，如果您在 MQADMIN 中有一个名为 hlq. * * 的概要文件 并且没有 hlq.RESLEVEL 概要文件，请谨防 hlq. ** 的后果。 概要文件，因为它用于 RESLEVEL 检查。

定义 hlq.RESLEVEL 概要文件并将 UACC 设置为 NONE ， 而不是完全没有 RESLEVEL 概要文件。 在访问列表中具有尽可能少的用户或组。 有关如何审计 RESLEVEL 访问权的详细信息，请参阅 第 229 页的『Auditing considerations on z/OS』。

如果仅使用队列管理器级别安全性，那么 IBM MQ 会针对 `qmgr-name.RESLEVEL` 概要文件执行 RESLEVEL 检查。 如果仅使用队列共享组级别安全性，那么 IBM MQ 会对 `qsg-name.RESLEVEL` 概要文件执行 RESLEVEL 检查。 如果同时使用队列管理器和队列共享组级别安全性的组合，那么 IBM MQ 首先会在队列管理器级别检查是否存在 RESLEVEL 概要文件。 如果未找到一个概要文件，那么它将在队列共享组级别检查 RESLEVEL 概要文件。

如果找不到 RESLEVEL 概要文件，那么 IBM MQ 将启用对 CICS 或 IMS 连接的作业和任务 (或备用用户) 标识的检查。 对于批处理连接， IBM MQ 允许检查作业 (或备用) 用户标识。 对于通道启动程序， IBM MQ 支持检查通道用户标识和 MCA (或备用) 用户标识。

如果存在 RESLEVEL 概要文件，那么检查级别取决于该概要文件的环境和访问级别。

请记住，如果您的队列管理器是队列共享组的成员，并且您未在队列管理器级别定义此概要文件，那么可能在队列共享组级别定义了一个将影响检查级别的概要文件。 要激活两个用户标识的检查，请使用 UACC (NONE) 定义 RESLEVEL 概要文件 (以队列共享组名的队列管理器名称作为前缀) ，并确保相关用户无权访问此概要文件。

当您考虑通道启动程序的用户标识对 RESLEVEL 的访问权时，请记住通道启动程序建立的连接也是通道使用的连接。 导致绕过通道启动程序用户标识的所有资源安全性检查的设置会有效地绕过所有通道的安全性检查。 如果通道启动程序对 RESLEVEL 的用户标识访问权不是 NONE ， 那么仅检查一个用户标识 (对于 READ 或 UPDATE 访问级别) 或任何用户标识 (对于 CONTROL 或 ALTER 访问级别) 以进行访问。 如果将 NONE 以外的访问级别授予通道启动程序的用户标识到 RESLEVEL ， 请确保您了解此设置对对通道执行的安全性检查的影响。

使用 RESLEVEL 概要文件意味着不会获取正常的安全性审计记录。 例如，如果将 UAUDIT 放在用户上，那么不会审计对 MQADMIN 中的 hlq.RESLEVEL 概要文件的访问权。

如果使用 hlq.RESLEVEL 概要文件上的 RACF WARNING 选项，那么不会为 RESLEVEL 类中的概要文件生成 RACF 警告消息。

报告消息 (例如 COD) 的安全性检查由与原始应用程序关联的 RESLEVEL 概要文件控制。 例如，如果批处理作业的用户标识对 RESLEVEL 概要文件具有 CONTROL 或 ALTER 权限，那么将绕过该批处理作业执行的所有资源检查，包括报告消息的安全性检查。

如果更改 RESLEVEL 概要文件，那么用户必须在进行更改之前断开连接并再次连接。 (这包括在分布式排队地址空间用户标识对 RESLEVEL 概要文件的访问权发生更改时停止并重新启动通道启动程序。)

要关闭 RESLEVEL 审计，请使用 RESAUDIT 系统参数。

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 205. 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 180, 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 207 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 213 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 213 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 217 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 217 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.
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.

Issued from...	User ID contents
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 207 in topic "RESLEVEL and CICS connections" on page 206, two user IDs are checked if the RESLEVEL profile is set to NONE. Then, from Table 58 on page 211 on, these checks are carried out:

- The hlq.ALTERNATE.USER.userid profile is not checked.
- The hlq.CONTEXT.queuename profile is checked with both the CICS address space user ID and the CICS transaction user ID.
- The hlq.resourcename profile is checked with both the CICS address space user ID and the CICS transaction user ID.

This means that four security checks are made for this MQOPEN call.

z/OS *User IDs checked for IMS connections*

The user IDs checked for IMS connections depend on whether one or two checks are to be performed, and whether an alternate user ID is specified. If a second user ID is checked, it depends on the type of dependent region and on which user IDs are available.

Table 59. User ID checking against profile name for IMS-type user IDs

Alternate user ID specified on open?	hlq.ALTERNATE.USER.userid profile	hlq.CONTEXT.queuename profile	hlq.resourcename profile
No, 1 check	-	REG	REG
No, 2 checks	-	REG+SEC	REG+SEC
Yes, 1 check	REG	REG	REG
Yes, 2 checks	REG+SEC	REG+SEC	REG+ALT

Key:

ALT
Alternate user ID.

REG
The user ID is normally set through the STARTED class or the started procedures table or, if IMS is running, from a submitted job, by the USER JCL parameter.

SEC
The second user ID is associated with the work being done in a dependent region. It is determined according to Table 60 on page 213.

Table 60. How the second user ID is determined for the IMS connection	
Types of dependent region	**Hierarchy for determining the second user ID**
• BMP message driven and successful GET UNIQUE issued. • IFP and GET UNIQUE issued. • MPP.	User ID associated with the IMS transaction if the user is signed on. LTERM name if available. PSBNAME.
• BMP message driven and successful GET UNIQUE not issued. • BMP not message driven. • IFP and GET UNIQUE not issued.	User ID associated with the IMS dependent region address space if this is not all blanks or all zeros. PSBNAME.

z/OS *User IDs used by the channel initiator*

This collection of topics describes the user IDs used and checked for receiving channels and for client MQI requests issued over server-connection channels. Information is provided for TCP/IP and for LU6.2

You can use the PUTAUT parameter of the receiving channel definition to determine the type of security checking used. To get consistent security checking throughout your IBM MQ network, you can use the ONLYMCA and ALTMCA options.

You can use the DISPLAY CHSTATUS command to determine the user identifier used by the MCA.

z/OS *Receiving channels using TCP/IP*

The user IDs checked depend on the PUTAUT option of the channel and on whether one or two checks are to be performed.

Table 61. User IDs checked against profile name for TCP/IP channels			
PUTAUT option specified on receiver or requester channel	**hlq.ALTERNATE.USER.userid profile**	**hlq.CONTEXT.queuename profile**	**hlq.resourcename profile**
DEF, 1 check	-	CHL	CHL
DEF, 2 checks	-	CHL + MCA	CHL + MCA
CTX, 1 check	CHL	CHL	CHL
CTX, 2 checks	CHL + MCA	CHL + MCA	CHL + ALT
ONLYMCA, 1 check	-	MCA	MCA
ONLYMCA, 2 checks	-	MCA	MCA
ALTMCA, 1 check	MCA	MCA	MCA
ALTMCA, 2 checks	MCA	MCA	MCA + ALT

Key:

MCA (MCA user ID)
The user ID specified for the MCAUSER channel attribute at the receiver; if blank, the channel initiator address space user ID of the receiver or requester side is used.

CHL (Channel user ID)
On TCP/IP, security is not supported by the communication system for the channel. If Transport Layer Security (TLS) is being used and a digital certificate has been flowed from the partner, the user ID associated with this certificate (if installed), or the user ID associated with a matching filter found by

using RACF Certificate Name Filtering (CNF), is used. If no associated user ID is found, or if TLS is not being used, the user ID of the channel initiator address space of the receiver or requester end is used as the channel user ID on channels defined with the PUTAUT parameter set to DEF or CTX.

Note: The use of RACF Certificate Name Filtering (CNF) allows you to assign the same RACF user ID to multiple remote users, for example all the users in the same organization unit, who would naturally all have the same security authority. This means that the server does not have to have a copy of the certificate of every possible remote user across the world, and greatly simplifies certificate management and distribution.

If the PUTAUT parameter is set to ONLYMCA or ALTMCA for the channel, the channel user ID is ignored and the MCA user ID of the receiver or requester is used. This also applies to TCP/IP channels using TLS.

ALT (Alternate user ID)
The user ID from the context information (that is, the *UserIdentifier* field) within the message descriptor of the message. This user ID is moved into the *AlternateUserID* field in the object descriptor before an **MQOPEN** or **MQPUT1** call is issued for the target destination queue.

z/OS *Receiving channels using LU 6.2*

The user IDs checked depend on the PUTAUT option of the channel and on whether one or two checks are to be performed.

Table 62. User IDs checked against profile name for LU 6.2 channels

PUTAUT option specified on receiver or requester channel	hlq.ALTERNATE.USER.userid profile	hlq.CONTEXT.queuename profile	hlq.resourcename profile
DEF, 1 check	-	CHL	CHL
DEF, 2 checks	-	CHL + MCA	CHL + MCA
CTX, 1 check	CHL	CHL	CHL
CTX, 2 checks	CHL + MCA	CHL + MCA	CHL + ALT
ONLYMCA, 1 check	-	MCA	MCA
ONLYMCA, 2 checks	-	MCA	MCA
ALTMCA, 1 check	MCA	MCA	MCA
ALTMCA, 2 checks	MCA	MCA	MCA + ALT

Key:

MCA (MCA user ID)
The user ID specified for the MCAUSER channel attribute at the receiver; if blank, the channel initiator address space user ID of the receiver or requester side is used.

CHL (Channel user ID)

Requester-server channels
If the channel is started from the requester, there is no opportunity to receive a network user ID (the channel user ID).

If the PUTAUT parameter is set to DEF or CTX on the requester channel, the channel user ID is that of the channel initiator address space of the requester because no user ID is received from the network.

If the PUTAUT parameter is set to ONLYMCA or ALTMCA, the channel user ID is ignored and the MCA user ID of the requester is used.

Other channel types

If the PUTAUT parameter is set to DEF or CTX on the receiver or requester channel, the channel user ID is the user ID received from the communications system when the channel is initiated.

- If the sending channel is on z/OS, the channel user ID received is the channel initiator address space user ID of the sender.
- If the sending channel is on a different platform (for example, AIX), the channel user ID received is typically provided by the USERID parameter of the channel definition.

If the user ID received is blank, or no user ID is received, a channel user ID of blanks is used.

ALT (Alternate user ID)

The user ID from the context information (that is, the *UserIdentifier* field) within the message descriptor of the message. This user ID is moved into the *AlternateUserID* field in the object descriptor before an MQOPEN or MQPUT1 call is issued for the target destination queue.

z/OS *Client MQI requests*

Various user IDs can be used, depending on which user IDs and environment variables have been set. These user IDs are checked against various profiles, depending on the PUTAUT option used and whether an alternate user ID is specified.

This section describes the user IDs checked for client MQI requests issued over server-connection channels for TCP/IP and LU 6.2. The MCA user ID and channel user ID are as for the TCP/IP and LU 6.2 channels described in the previous sections.

For server-connection channels, the user ID received from the client is used if the MCAUSER attribute is blank.

See “客户机的访问控制” on page 87 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

Table 63. User IDs checked against profile name for LU 6.2 and TCP/IP server-connection channels (continued)

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
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 208 shows that two user IDs are checked because RESLEVEL is set to NONE.

Table 61 on page 213 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 222 summarizes the situations in which security information is cached and in which cached information is used.

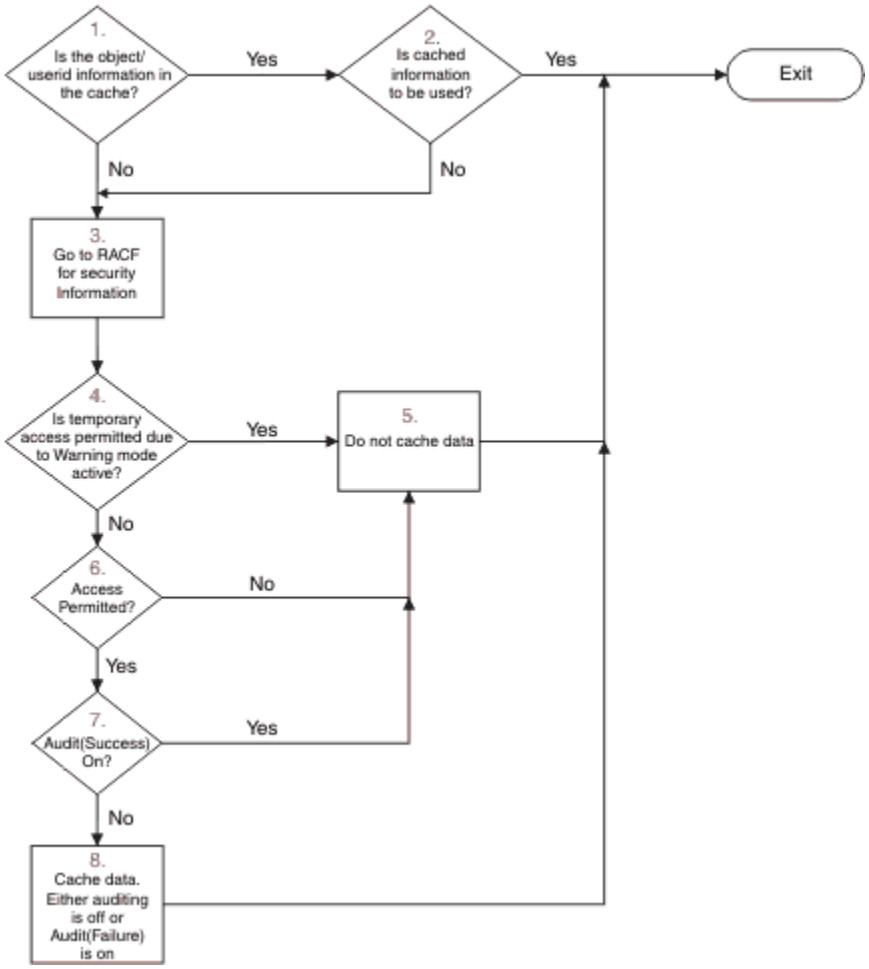


Figure 17. Logic flow for IBM MQ security caching

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 225 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 225 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 232, 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
要在通道级别对授予连接系统的访问权进行更为精确的控制，可以使用通道认证记录。

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 229.

```
       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 190.
- 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 180, and to all the user destination queues and the dead-letter queue (but see “Dead-letter queue security” on page 179).

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 190 “RESLEVEL and the channel initiator connection” on page 208 and “User IDs for security checking on z/OS” on page 210 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 174.

Data sets
The channel initiator address space needs appropriate access to queue manager data sets, see “Authorizing access to data sets” on page 225.

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 193.

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 210 for more information.

You can also use the Transport Layer Security (TLS) protocol to provide security on channels. See “IBM MQ 中的 TLS 安全协议” on page 21 for more information about using TLS with IBM MQ.

See also “客户机的访问控制” on page 87 for information about server-connection security.

User IDs
The user IDs described in “User IDs used by the channel initiator” on page 213 and “User IDs used by the intra-group queuing agent” on page 217 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 232:

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 193) 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 238.
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.

设置 IBM MQ MQI client 安全性

您必须考虑 IBM MQ MQI client 安全性，以便客户机应用程序对服务器上的资源没有不受限制的访问权。

运行客户机应用程序时，请勿使用具有比必需访问权更多的用户标识 (例如， mqm 组中的用户，甚至 mqm 用户本身) 来运行应用程序。

通过以具有过多访问权的用户身份运行应用程序，您将冒着应用程序访问和更改队列管理器部分的风险 (无论是意外访问还是恶意访问)。

客户机应用程序与其队列管理器服务器之间的安全性有两个方面: 认证和访问控制。

- 可以使用认证来确保以特定用户身份运行的客户机应用程序是他们所说的用户。 通过使用认证，您可以通过模拟某个应用程序来阻止攻击者获取对队列管理器的访问权。

  认证由以下两个选项之一提供:

  – 连接认证功能。

    有关连接认证的更多信息，请参阅 第 60 页的『连接认证』。

  – 在 TLS 中使用相互认证。

    有关 TLS 的更多信息，请参阅 第 245 页的『使用 SSL/TLS』。
- 访问控制可用于授予或除去特定用户或用户组的访问权。 通过使用专门创建的用户 (或特定组中的用户) 运行客户机应用程序，您可以使用访问控制来确保应用程序无法访问不应该访问的队列管理器部分。

设置访问控制时，必须考虑通道认证规则和通道上的 MCAUSER 字段。 这两个功能部件都能够更改用于验证访问控制权限的用户标识。

有关访问控制的更多信息，请参阅 第 309 页的『授予对对象的访问权』。

如果您已设置客户机应用程序以连接到具有受限标识的特定通道，但该通道在其 MCAUSER 字段中设置了管理员标识，那么只要客户机应用程序成功连接，该管理员标识将用于访问控制检查。 因此，客户机应用程序将具有对队列管理器的完全访问权。

有关 MCAUSER 属性的更多信息，请参阅 第 340 页的『将客户机用户标识映射到 MCAUSER 用户标识』。

通过设置要接受的连接的特定规则和条件，通道认证规则也可用作控制对队列管理器的访问的方法。

有关通道认证规则的更多信息，请参阅: 第 43 页的『通道认证记录』。

指定运行时在 MQI 客户机上仅使用经过 FIPS 认证的 CipherSpecs

使用符合 FIPS 的软件创建密钥存储库，然后指定通道必须使用经 FIPS 认证的 CipherSpecs。

注: 在 AIX, Linux, and Windows 上， IBM MQ 通过 IBM Crypto for C (ICC) 加密模块提供 FIPS 140-2 合规性。 此模块的证书已移至历史状态。 客户应查看 IBM Crypto for C (ICC) 证书 并了解 NIST 提供的任何建议。 当前正在进行 FIPS 140-$tag1 替换模块，可以通过在 流程列表中的 NIST CMVP 模块中搜索该模块来查看其状态。

IBM MQ Operator 3.2.0 和队列管理器容器映像 9.4.0.0 和更高版本基于 UBI 9。 FIPS 140-3 合规性当前处于暂挂状态，可通过在 流程列表中的 NIST CMVP 模块中搜索 "Red Hat Enterprise Linux 9- OpenSSL FIPS 提供程序" 来查看其状态。

为了在运行时符合 FIPS ，必须仅使用符合 FIPS 的软件 (例如，带有 `-fips` 选项的 **`runmqakm`**) 来创建和管理密钥存储库。

您可以指定 TLS 通道必须以三种方式仅使用 FIPS 认证的 CipherSpecs ，按优先顺序列出:

1. 将 MQSCO 结构中的 FipsRequired 字段设置为 `MQSSL_FIPS_YES`。
2. 将环境变量 **`MQSSLFIPS`** 设置为 YES。
3. 将客户机配置文件的 SSL 节中的 **`SSLFipsRequired`** 属性设置为 YES。

缺省情况下，不需要 FIPS 认证的 CipherSpecs 。

这些值与 **`ALTER QMGR SSLFIPS`** 上的等效参数值具有相同的含义 (请参阅 **`ALTER QMGR`** (改变队列管理器设置))。 如果客户机进程当前没有活动 TLS 连接，并且在 SSL MQCONNX 上有效指定了 FipsRequired 值，那么与此进程关联的所有后续 TLS 连接都必须仅使用与此值关联的 CipherSpecs 。 这在此连接和所有其他 TLS 连接停止之前适用，在此阶段，后续 MQCONNX 可以为 FipsRequired 提供新值。

如果存在加密硬件，那么可以将 IBM MQ 使用的加密模块配置为硬件产品提供的那些模块，并且这些模块可以通过 FIPS 认证到特定级别。 可配置模块以及它们是否通过 FIPS 认证取决于正在使用的硬件产品。

在可能的情况下，如果配置了仅 FIPS CipherSpecs ，那么 MQI 客户机将拒绝使用 `MQRC_SSL_INITIALIZATION_ERROR` 指定非 FIPS CipherSpec 的连接。 IBM MQ 不保证拒绝所有此类连接，您负责确定 IBM MQ 配置是否符合 FIPS 标准。

相关概念

第 30 页的『针对 AIX, Linux, and Windows 的美国联邦信息处理标准 (FIPS)』
在 AIX, Linux, and Windows 系统上的 SSL/TLS 通道上需要密码术时， IBM MQ 将使用名为 IBM Crypto for C (ICC)的密码术包。 在 AIX, Linux, and Windows 平台上， ICC 软件已通过美国国家标准技术学会的联邦信息处理标准 (FIPS) 加密验证程序，级别为 140-2。

AIX 在 AIX 上运行具有多个 GSKit 8.0 安装的 TLS 客户机应用程序

在具有多个 IBM Global Security Kit (GSKit) 8.0 安装的 AIX 系统上运行时， AIX 上的 TLS 客户机应用程序可能会迁到 `MQRC_CHANNEL_CONFIG_ERROR` 和错误 `AMQ6175` 。

在具有多个 GSKit 8.0 安装的 AIX 系统上运行客户机应用程序时，客户机连接调用可以在使用 TLS 时返回 `MQRC_CHANNEL_CONFIG_ERROR` 。 `/var/mqm/errors` 记录失败客户机应用程序的记录错误 `AMQ6175` 和 `AMQ9220` ，例如:

```
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 ----------------------------------------------------------
```

此错误的常见原因是 LIBPATH 或 LD_LIBRARY_PATH 环境变量的设置导致 IBM MQ 客户机从两个不同的 GSKit 8.0 安装装入一组混合库。 在 Db2 环境中执行 IBM MQ 客户机应用程序可能会导致此错误。

要避免此错误，请在库路径的前面包含 IBM MQ 库目录，以便 IBM MQ 库优先。 可以使用带有 **-k** 参数的 **setmqenv** 命令来实现此目标，例如:

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

有关使用 **setmqenv** 命令的更多信息，请参阅 setmqenv (set IBM MQ environment)

使用 MQSC 配置 TLS 通道

要配置 TLS 通道，请使用 **runmqsc** 和 ALTER CHANNEL 命令。 您可以选择将此通道配置为仅接受含有与给定值匹配的所有者专有名称中属性的证书。 您还可以选择配置队列管理器通道，以便在启动方不发送其自有个人证书时，此队列管理器会拒绝连接。

关于此任务

要在 IBM MQ Explorer 中配置通道，请参阅 使用 IBM MQ Explorer 配置 TLS 通道。

要使用 **runmqsc** 配置通道，请完成以下步骤。

过程

1. 调用连接到目标队列管理器的 **runmqsc** 命令。
2. 确定要为 TLS 启用的通道。
   请注意通道名称和通道类型。

3. 使用 ALTER CHANNEL 命令可更改 IBM MQ 通道的各种属性。
   除了提供命令外，还提供通道名称和通道类型。 例如，要变更名为 MQ.TEST 运行以下命令:

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR)
```

   您可以在 IBM MQ 通道定义上调整与 TLS 相关的各种通道属性。

下一步做什么

设置消息安全性

支持 TLS 的消息传递提供了两种方法来确保消息的安全性:

- 加密确保一旦此消息被拦截，它也是不可读的。
- 散列函数确保一旦此消息被改变，也会检测到这一情况。

这些方法的组合称为密码规范或 CipherSpec。 必须为通道的两端设置相同的 CipherSpec，否则支持 TLS 的消息传递将失败。 有关更多信息，请参阅第 7 页的『保护 IBM MQ』。

要变更 IBM MQ 通道启用 TLS，请在 SSLCIPH 属性中指定值。 必须将此属性设置为列表 第 368 页的『启用 CipherSpecs』中队列管理器的队列平台的有效 CipherSpec 。

要变更 IBM MQ 通道以禁用 TLS，请将 SSLCIPH 设置为空白值。 例如:

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLCIPH(ANY_TLS12_OR_HIGHER)
```

注: 您必须将通道名称包含在单引号中，以确保保留字符大小写。 如果没有单引号， IBM MQ 会将字符串转换为全大写。

根据其所有者的名称过滤证书

证书包含证书所有者的专有名称。 您可以选择将此通道配置为仅接受含有与给定值匹配的所有者专有名称中的属性的证书。

下表中列出了 IBM MQ 可以过滤的属性名称:

属性名称	含义
SERIALNUMBER	证书序列号
MAIL	电子邮件地址
Deprecated E	电子邮件地址（不推荐，最好使用 MAIL）
UID 或 USERID	用户标识
CN	公共名称
T	标题
OU	组织单元名称
DC	域组件
O	组织名称
STREET	街道/地址第一行
L	地区名称
ST（或 SP 或 S）	省/直辖市/自治区名称
PC	邮政编码
C	国家或地区
UNSTRUCTUREDNAME	主机名

属性名称	含义
UNSTRUCTUREDADDRESS	IP 地址
DNQ	专有名称限定符

可以在属性值的开头或结尾使用通配符 (*) 来代替任意数目的字符。 例如，要仅接受名称以 Smith 结尾并在 GB（英国）为 IBM 工作的任何人员的证书，请输入：

```
CN=*Smith, O=IBM, C=GB
```

例如：

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLPEER('CN=*Smith, O=IBM, C=GB')
```

注: 必须将 SSLPEER 字符串包含在单引号中，以确保保留字符大小写。 如果没有单引号， IBM MQ 会将字符串转换为全大写。

认证方启动与队列管理器的连接

当另一方启动对队列管理器的支持 TLS 的连接时，此队列管理器必须将其个人证书作为身份证明发送至启动方。 您还可以选择配置队列管理器通道，以便在启动方不发送其自己的个人证书时，此队列管理器拒绝连接。

要执行此操作，请设置 SSLCAUTH 属性。 此属性是布尔属性，可以具有值 OPTIONAL 或 REQUIRED:

- OPTIONAL 用于认证连接客户机的证书 (如果提供了连接客户机) ，但不要求客户机发送连接客户机的证书。 如果客户机发送的证书无效，那么将拒绝该客户机。
- REQUIRED 拒绝任何未提供有效 TLS 证书的连接客户机

例如：

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLCAUTH(REQUIRED)
```

IBM i 在 IBM i 上设置 SSL 或 TLS 的通信

使用 SSL 或 TLS 加密安全协议的安全通信涉及设置通信通道和管理将用于认证的数字证书。

要设置 SSL 或 TLS 安装，必须定义通道以使用 SSL 或 TLS。 您还必须创建和管理数字证书。 在某些操作系统上，可以使用自签名证书执行测试。 但是，在 IBM i 上，必须使用本地 CA 签署的个人证书。

有关创建和管理证书的完整信息，请参阅 第 245 页的『在 IBM i 上使用 SSL/TLS』。

此主题集合介绍了设置 SSL 或 TLS 通信所涉及的一些任务，并提供了有关完成这些任务的逐步指导

您可能还想要测试 SSL 或 TLS 客户机认证，这是 SSL 和 TLS 协议的可选部分。 在 SSL 或 TLS 握手期间，SSL 或 TLS 客户机始终从服务器获取并验证数字证书。 通过 IBM MQ 实现， SSL 或 TLS 服务器始终从客户机请求证书。

在 IBM i 上，仅当 SSL 或 TLS 客户机具有以正确 IBM MQ 格式标注的证书时，才会发送该证书:

- 对于队列管理器， `ibmwebspheremq` 后跟队列管理器的名称已更改为小写。 例如，对于 QM1，`ibmwebspheremqqm1`。
- 对于 IBM MQ C Client for IBM i， `ibmwebspheremq` 后跟您的登录用户标识已更改为小写，例如 `ibmwebspheremqmyuserid`。

IBM MQ 在标签上使用 `ibmwebspheremq` 前缀以避免与其他产品的证书混淆。 确保以小写形式指定整个证书标签。

SSL 或 TLS 服务器始终验证客户机证书 (如果发送了客户机证书)。 如果 SSL 或 TLS 客户机未发送证书，那么仅当使用 SSLCAUTH 参数设置为 REQUIRED 或 SSLPEER 参数值来定义充当 SSL 或 TLS 服务器的通道结束时，认证才会失败。 有关更多信息，请参阅 使用 SSL 或 TLS 连接两个队列管理器。

ALW 在 AIX, Linux, and Windows 上设置 SSL 或 TLS 的通信

使用 SSL 或 TLS 加密安全协议的安全通信涉及设置通信通道和管理将用于认证的数字证书。

要设置 SSL 或 TLS 安装，必须定义通道以使用 SSL 或 TLS。 您还必须创建和管理数字证书。 在 AIX, Linux, and Windows 系统上，可以使用自签名证书执行测试。

注意: 无法在要使用支持 TLS 的通道连接在一起的队列管理器上混合使用椭圆曲线签名证书和 RSA 签名证书。

使用支持 TLS 的通道的队列管理器必须全部使用 RSA 签名的证书，或者全部使用 EC 签名的证书，而不是两者的混合。

请参阅 第 39 页的『IBM MQ 中的数字证书和 CipherSpec 兼容性』 以获取更多信息。

无法撤销自签名证书，这可能允许攻击者在私钥被泄露后破坏身份。 CA 可以撤销已泄密的证书，这将阻止其进一步使用。 因此， CA 签署的证书在生产环境中使用更安全，尽管自签名证书对于测试系统更方便。

有关创建和管理证书的完整信息，请参阅 第 260 页的『在 AIX, Linux, and Windows 上使用 SSL/TLS』。

此主题集合介绍了设置 SSL 通信所涉及的一些任务，并提供了有关完成这些任务的逐步指导。

您可能还想要测试 SSL 或 TLS 客户机认证，这是协议的可选部分。 在 SSL 或 TLS 握手期间， SSL 或 TLS 客户机始终从服务器获取并验证数字证书。 通过 IBM MQ 实现， SSL 或 TLS 服务器始终从客户机请求证书。

在 AIX, Linux, and Windows 上，仅当 SSL 或 TLS 客户机具有以正确 IBM MQ 格式标注的证书时，才会发送该证书:

- 对于队列管理器，格式为 `ibmwebspheremq` ，后跟队列管理器的名称更改为小写。 例如，对于 `QM1`，`ibmwebspheremqqm1`
- 对于 IBM MQ 客户机， `ibmwebspheremq` 后跟您的登录用户标识已更改为小写，例如 `ibmwebspheremqmyuserid`。

IBM MQ 在标签上使用 `ibmwebspheremq` 前缀以避免与其他产品的证书混淆。 确保以小写形式指定整个证书标签。

SSL 或 TLS 服务器始终验证客户机证书 (如果发送了客户机证书)。 如果客户机未发送证书，那么仅当使用 SSLCAUTH 参数设置为 REQUIRED 或 SSLPEER 参数值定义充当 SSL 或 TLS 服务器的通道结束时，认证才会失败。 有关更多信息，请参阅 使用 SSL 或 TLS 连接两个队列管理器。

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 271.

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.

使用 SSL/TLS

这些主题提供了有关执行与将 TLS 与 IBM MQ 配合使用相关的单个任务的指示信息。

其中许多任务用作以下部分中描述的更高级别任务中的步骤:

- 第 283 页的『识别和认证用户』
- 第 309 页的『授予对对象的访问权』
- 第 368 页的『消息的机密性』
- 第 418 页的『消息的数据完整性』
- 第 419 页的『确保集群安全』

IBM i 在 IBM i 上使用 SSL/TLS

此主题集合提供有关在 IBM MQ for IBM i 中使用传输层安全性 (TLS) 的个别任务的指示信息。

对于 IBM i， TLS 支持是操作系统的组成部分。 确保已安装 IBM i 上的硬件和软件需求中列出的先决条件。

在 IBM i 上，您可以使用 "数字 Certificate Manager " (DCM) 工具来管理密钥和数字证书。

访问 *DCM*

遵循以下指示信息以访问 DCM 接口。

关于此任务

在支持框架的 Web 浏览器中执行以下步骤。

过程

1. 转至 `http://`*machine*`.domain:2001` 或 `https://`*machine*`.domain:2010`，其中 *machine* 是计算机的名称。
2. 请在请求时输入有效的用户概要文件和密码。

   确保您的用户概要文件具有 *ALLOBJ 和 *SECADM 特权，以使您能够创建新的证书库。 如果您没有特权，那么只能管理个人证书或查看您有权使用的对象的对象特征符。 如果您有权使用对象签名应用程序，那么还可以对 DCM 中的对象进行签名。

3. 在 "因特网配置" 页面上，单击 **数字 Certificate Manager**。
   这将显示 "数字 Certificate Manager " 页面。

将证书分配给 *IBM i* 上的队列管理器

使用 DCM 将证书分配给队列管理器。

使用传统 IBM i 数字证书管理将证书分配给队列管理器。 这意味着您可以指定队列管理器使用系统证书库，并且注册队列管理器以用作具有数字 Certificate Manager 的应用程序。 为此，请将队列管理器 **SSLKEYR** 属性的值更改为 *SYSTEM。

当 **SSLKEYR** 参数更改为 *SYSTEM 时， IBM MQ 将队列管理器注册为具有 QIBM_WEBSPHERE_MQ_QMGRNAME 的唯一应用程序标签和具有 Qmgrname (WMQ) 描述的标签的服务器应用程序。 请注意，如果使用 *SYSTEM 证书库，那么不会使用通道 **CERTLABL** 属性。 然后，队列管理器将显示为数字 Certificate Manager 中的服务器应用程序，并且您可以将系统存储库中的任何服务器或客户机证书分配给该应用程序。

由于队列管理器已注册为应用程序，因此可以执行 DCM 的高级功能，例如定义 CA 信任列表。

如果 **SSLKEYR** 参数更改为 *SYSTEM 以外的值，那么 IBM MQ 将队列管理器注销为具有数字 Certificate Manager 的应用程序。 如果删除了队列管理器，那么还会从 DCM 注销该队列管理器。 具有足够 *SECADM 权限的用户还可以手动在 DCM 中添加或除去应用程序。

在 *IBM i* 上设置密钥存储库

必须在连接的两端设置密钥存储库。 可以使用缺省证书库，也可以创建您自己的证书库。

TLS 连接的每一端都需要一个 密钥存储库 。 每个队列管理器和 IBM MQ MQI client 都必须有权访问密钥存储库。 如果要使用文件名和密码 (即，不使用 *SYSTEM 选项) 访问密钥存储库，请确保 QMQM 用户概要文件具有以下权限:

- 包含密钥存储库的目录的执行权限
- 包含密钥存储库的文件的读权限

请参阅第 22 页的『SSL/TLS 密钥存储库』以获取更多信息。 请注意，如果使用 *SYSTEM 证书库，那么不会使用通道 **CERTLABL** 属性。

在 IBM i 上，数字证书存储在由 DCM 管理的证书库中。 这些数字证书具有标签，这些标签将证书与队列管理器或 IBM MQ MQI client 相关联。 TLS 将证书用于认证目的。

如果设置了 **CERTLABL** 属性的值，那么该标签是该属性的值，或者附加了队列管理器名称或 IBM MQ MQI client 用户登录标识的缺省 `ibmwebspheremq` (全部为小写)。 请参阅 数字证书标签 以获取详细信息。

队列管理器或 IBM MQ MQI client 证书库名称包含路径和主干名称。 缺省路径为 `/QIBM/UserData/ICSS/Cert/Server/` ，缺省词干名称为 `Default`。 在 IBM i 上，缺省证书库 `/QIBM/UserData/ICSS/Cert/Server/Default.kdb` 也称为 `*SYSTEM`。 (可选) 您可以定义自己的路径和词干名称。

如果您定义自己的路径或文件名，请设置对该文件的许可权以严格控制对该文件的访问权。

第 249 页的『在 IBM i 上更改队列管理器的密钥存储库位置』告诉您指定证书库名称。 您可以在创建证书库之前或之后指定证书库名称。

注: 您可以使用 DCM 执行的操作可能受用户概要文件的权限限制。 例如，您需要 *ALLOBJ 和 *SECADM 权限才能创建 CA 证书。

IBM i 对 *IBM i* 上的密钥存储库密码进行加密

多个 IBM MQ 组件需要访问包含数字证书或对称密钥的密钥存储库。 密钥存储库使用密码进行保护，因为它包含敏感信息。 访问密钥存储库时，必须将密钥存储库密码存储在 IBM MQ 可以读取该密码的位置。 还必须对密码进行加密，以降低对密钥存储库进行未经授权的访问的可能性。

以下 IBM MQ 组件和功能部件支持两种不同的方法来存储密钥存储库密码:

- 队列管理器 TLS 密钥存储库。
- 使用 TLS 的 IBM MQ MQI clients 。

供这些组件使用的密钥存储库密码使用 IBM MQ 密码保护系统进行保护。 提供密码并对其进行加密的机制根据组件的不同而略有不同:

队列管理器 TLS 密钥存储库

当使用 CHGMQM (更改消息队列管理器) 命令设置 **SSLKEYRPWD** 队列管理器属性时，将对密码进行加密。

使用 AES-128 算法对密码进行加密。 此算法的详细信息是公开的，并且被认为是安全的。

密码以可能访问密钥存储库的其他软件无法理解的专有格式存储在隐藏文件中。

由一个 IBM MQ 组件加密的密码不能由另一个 IBM MQ 组件使用。

加密密钥存储库密码时，可以提供唯一的加密密钥。 唯一的加密密钥会阻止无法访问加密密钥的任何人解密密码。 通过 **INITKEY** 队列管理器属性提供此密钥，必须先设置此属性，然后才能提供要加密的密码。

有关 IBM MQ 密码保护系统的更多信息，请参阅 第 494 页的『保护 IBM MQ 组件配置文件中的密码』。

使用 TLS 的 IBM MQ MQI clients

第 258 页的『IBM MQ IBM i 的 SSL 客户机实用程序 (amqrsslc)』 可以将密钥存储库密码存储在隐藏文件中。 另请参阅 在 IBM i 上使用 MQSC 命令进行管理。

使用 AES-128 算法对密码进行加密。 此算法的详细信息是公开的，并且被认为是安全的。

密码以可能访问密钥存储库的其他软件无法理解的专有格式存储在隐藏文件中。

加密密钥存储库密码时，可以提供唯一的加密密钥。 唯一的加密密钥会阻止无法访问加密密钥的任何人解密密码。 通过 **-sf** 参数提供此密钥。

加密密码存储在与密钥存储库文件相同的目录中的隐藏文件中。

IBM MQ MQI clients 还支持通过其他机制提供的密码。 请参阅 第 250 页的『为 IBM i 上的 IBM MQ MQI client 提供密钥存储库密码』。

无论您选择何种方法对密钥存储库密码进行加密，请确保您了解对存储的密码进行加密的限制。 请参阅 第 500 页的『通过密码加密进行保护时存在的限制』。

相关概念

第 249 页的『为 IBM i 上的队列管理器提供密钥存储库密码』
因为密钥存储库包含敏感信息，所以使用密码对其进行保护。 为了能够访问密钥存储库内容以执行 TLS 操作， IBM MQ 必须能够检索密钥存储库密码。

第 250 页的『为 IBM i 上的 IBM MQ MQI client 提供密钥存储库密码』
因为密钥存储库包含敏感信息，所以使用密码对其进行保护。 为了能够访问密钥存储库内容以执行 TLS 操作， IBM MQ 必须能够检索密钥存储库密码。

第 245 页的『在 IBM i 上使用 SSL/TLS』
此主题集合提供有关在 IBM MQ for IBM i 中使用传输层安全性 (TLS) 的个别任务的指示信息。

在 *IBM i* 上创建证书库
如果您不希望使用缺省证书库，请遵循此过程来创建您自己的证书库。

关于此任务

仅当您不希望使用 IBM i 缺省证书库时，才创建新的证书库。

要指定要使用 IBM i 系统证书库，请将队列管理器的 SSLKEYR 属性的值更改为 *SYSTEM。 此值指示队列管理器使用系统证书库，并且队列管理器已注册为使用数字 Certificate Manager (DCM) 作为应用程序。

过程

1. 访问 DCM 接口，如 第 245 页的『访问 DCM』 中所述
2. 在导航面板中，单击 **新建证书库**。

"创建新证书库" 页面将显示在任务框架中。

3. 在任务框架中，选择 **其他系统证书库**，然后单击 **继续**。
   "在新证书库中创建证书" 页面将显示在任务框架中。
4. 选择 **否-不在证书库中创建证书**，然后单击 **继续**。
   "证书库名称" 和 "密码" 页面将显示在任务框架中。
5. 在 **证书库路径和文件名** 字段中，输入 IFS 路径和文件名，例如 `/QIBM/UserData/mqm/qmgrs/qm1/key.kdb`
6. 在 **密码** 字段中输入密码，然后在 **确认密码** 字段中再次输入密码。 单击**继续**。
   记下密码 (区分大小写)，因为您在存储存储库密钥时需要该密码。
7. 要退出 DCM，请关闭浏览器窗口。

下一步做什么

使用 DCM 创建证书库后，请确保隐藏密码，如 第 248 页的『在 IBM i 系统上存储证书库密码』 中所述

相关任务

第 256 页的『将证书导入到 IBM i 上的密钥存储库中』
遵循此过程以导入证书。

在 *IBM i* 系统上存储证书库密码

使用 CL 命令隐藏证书库密码。

以下指示信息适用于在 IBM i 上存储队列管理器的证书库密码。 或者，对于 IBM MQ MQI client，如果您未使用 *SYSTEM 证书库 (即 MQSSLKEYR 环境设置为 *SYSTEM 以外的值)，请遵循 第 258 页的『IBM MQ IBM i 的 SSL 客户机实用程序 (amqrsslc)』的 第 259 页的『隐藏证书库密码』 部分中描述的过程。

如果已指定将使用 *SYSTEM 证书库 (通过将队列管理器的 SSLKEYR 属性的值更改为 *SYSTEM)，那么不得遵循这些步骤。

使用 DCM 创建证书库后，请使用以下命令来隐藏密码:

```
STRMQM MQMNAME('queue_manager_name')
CHGMQM MQMNAME('queue_manager_name') SSLKEYRPWD('password')
```

密码区分大小写。 它必须与您在 第 247 页的『在 IBM i 上创建证书库』的步骤 6 中输入的内容完全一样在单引号中输入。

注: 如果您未使用缺省系统证书库，并且未隐藏密码，那么尝试启动 TLS 通道将失败，因为它们无法获取访问证书库所需的密码。

密码保护

指定密钥存储库密码时， IBM MQ 将使用 IBM MQ 密码保护系统对密码进行加密。 要对密码进行加密，将使用初始密钥; 如果未向队列管理器提供此密钥，那么将改为使用缺省密钥。

在提供密钥存储库密码之前，应该为队列管理器设置唯一的初始密钥。 您可以使用 **ALTER QMGR** MQSC 命令的 **INITKEY** 属性来执行此操作:

```
ALTER QMGR INITKEY('value')
```

在 *IBM i* 上查找队列管理器的密钥存储库

使用此过程来获取队列管理器的证书库的位置。

过程

1. 使用以下命令显示队列管理器的属性:

   ```
   DSPMQM MQMNAME('queue manager name')
   ```

2. 检查命令输出以获取证书库的路径和主干名称。

例如: /QIBM/UserData/ICSS/Cert/Server/Default，其中 /QIBM/UserData/ICSS/Cert/Server 是路径， Default 是主干名称。

在 *IBM i* 上更改队列管理器的密钥存储库位置

使用 CHGMQM 或 ALTER QMGR 更改队列管理器证书库的位置。

过程

使用 CHGMQM 命令或 ALTER QMGR MQSC 命令来设置队列管理器的密钥存储库属性。

a) 使用 CHGMQM: `CHGMQM MQMNAME('qm1') SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb')`

b) 使用 ALTER QMGR: `ALTER QMGR SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb')`

在任一情况下，证书库都具有标准文件名: /QIBM/UserData/ICSS/Cert/Server/MyKey.kdb

下一步做什么

更改队列管理器证书库的位置时，不会从旧位置传输证书。 如果创建证书库时预安装的 CA 证书不足，那么必须使用证书填充新的证书库，如 第 256 页的『将证书导入到 IBM i 上的密钥存储库中』中所述。 您还必须隐藏新位置的密码，如 第 248 页的『在 IBM i 系统上存储证书库密码』中所述。

IBM i 为 *IBM i* 上的队列管理器提供密钥存储库密码

因为密钥存储库包含敏感信息，所以使用密码对其进行保护。 为了能够访问密钥存储库内容以执行 TLS 操作， IBM MQ 必须能够检索密钥存储库密码。

IBM MQ 提供了一种机制，用于向队列管理器提供密钥存储库密码:

- **CHGMQM** 命令上的 **SSLKEYRPWD** 参数

密钥存储库密码通过使用 IBM MQ 密码保护系统进行加密。 有关保护密钥存储库密码的方法的更多信息，请参阅 第 246 页的『对 IBM i 上的密钥存储库密码进行加密』。

另请参阅 在 IBM i 上使用 MQSC 命令进行管理。

SSLKEYRPWD 属性

要直接向队列管理器提供密钥存储库密码，请运行以下 **CHGMQM** 命令，将 *queue_manager* 替换为队列管理器名称，将 *password* 替换为密钥存储库密码。

```
CHGMQM MQMNAME('queue_manager') SSLKEYRPWD('password')
```

注意: 确保使用单引号将队列管理器名称和密码括起来，否则 IBM MQ 会将字符转换为大写。

使用此方法指定密钥存储库密码时，将先使用 IBM MQ 密码保护系统对密码进行加密，然后再进行存储。

加密密钥 (称为初始密钥) 用于加密密码。 将队列管理器设置为使用唯一的初始密钥来安全地保护密码。 如果未提供初始密钥，那么将使用缺省密钥。

在设置密钥存储库密码之前，请确保使用唯一的初始密钥配置队列管理器。 您可以使用 **ALTER QMGR** 命令上的 **INITKEY** 属性来修改初始密钥。 例如:

```
ALTER QMGR INITKEY('mykey')
```

警告: 如果在设置密钥存储库密码后修改初始密钥，那么不会使用新的初始密钥对密钥存储库密码进行加密。 如果更改初始密钥，那么还必须重置密钥存储库密码。 否则， IBM MQ 无法解密密钥存储库密码，因此无法访问密钥存储库。

有关 **SSLKEYRPWD** 属性的更多信息，请参阅 **CHGMQM** 命令上的 **SSLKEYRPWD** 参数。

相关概念

第 246 页的『对 IBM i 上的密钥存储库密码进行加密』

多个 IBM MQ 组件需要访问包含数字证书或对称密钥的密钥存储库。 密钥存储库使用密码进行保护，因为它包含敏感信息。 访问密钥存储库时，必须将密钥存储库密码存储在 IBM MQ 可以读取该密码的位置。 还必须对密码进行加密，以降低对密钥存储库进行未经授权的访问的可能性。

第 250 页的『为 IBM i 上的 IBM MQ MQI client 提供密钥存储库密码』
因为密钥存储库包含敏感信息，所以使用密码对其进行保护。 为了能够访问密钥存储库内容以执行 TLS 操作， IBM MQ 必须能够检索密钥存储库密码。

IBM i 为 *IBM i* 上的 *IBM MQ MQI client* 提供密钥存储库密码

因为密钥存储库包含敏感信息，所以使用密码对其进行保护。 为了能够访问密钥存储库内容以执行 TLS 操作， IBM MQ 必须能够检索密钥存储库密码。

IBM MQ 提供了四种机制来向 IBM MQ MQI client 提供密钥存储库密码:

- 第 250 页的『MQSCO 的 KeyRepoPassword 字段 』
- 第 250 页的『MQKEYRPWD 环境变量』
- 第 251 页的『客户机配置文件的 SSLKeyRepositoryPassword 属性』
- 第 251 页的『密钥存储库隐藏文件』

如果不使用密钥存储库隐藏文件，那么可以将密钥存储库密码作为纯文本字符串或使用 IBM MQ 密码保护系统加密的字符串提供。 有关保护密钥存储库密码的方法的更多信息，请参阅 第 246 页的『对 IBM i 上的密钥存储库密码进行加密』。

MQSCO 的 KeyRepoPassword 字段

要使用 MQSCO 结构提供密钥存储库密码，必须使用以下三个变量字符串字段的组合:

KeyRepoPasswordLength
密码的长度。

KeyRepoPasswordPtr
指向内存中包含密码的位置的指针。

KeyRepoPasswordOffset
密码在内存中的位置，表示为从 MQSCO 结构开始的字节数。

注: 只能提供 **KeyRepoPasswordPtr** 或 **KeyRepoPasswordOffset** 中的一个。

例如:

```
char * pwd = "passw0rd";
MQSCO  SslConnOptions = {MQSCO_DEFAULT};

SslConnOptions.KeyRepoPasswordPtr = pwd;
SslConnOptions.KeyRepoPasswordLength = (MQLONG)strlen(SslConnOptions.KeyRepoPasswordPtr);
SslConnOptions.Version = MQSCO_VERSION_6;
```

注意: 如果使用此方法提供密码，请在将密码提供给 IBM MQ client 应用程序之前对其进行加密。 有关更多信息，请参阅 第 251 页的『加密密钥存储库密码』。

有关 MQCSO 结构的更多信息，请参阅 MQSCO-SSL/TLS 配置选项。

MQKEYRPWD 环境变量

如果未使用 MQSCO 结构向客户机提供密钥存储库密码，那么可以使用 *MQKEYRPWD* 环境变量来指定密钥存储库密码。 例如:

```
export MQKEYRPWD=passw0rd
```

或

```
set MQKEYRPWD=passw0rd
```

其中 *passw0rd* 是您的密码。

注意: 如果使用此方法提供密码，请在设置环境变量的值之前对密码进行加密。 有关更多信息，请参阅第 251 页的『加密密钥存储库密码』。

客户机配置文件的 SSLKeyRepositoryPassword 属性

如果未使用其他方法之一向客户机提供密钥存储库密码，那么可以使用客户机配置文件的 **SSL** 节中的 **SSLKeyRepositoryPassword** 属性来指定密钥存储库密码。 例如:

```
SSL:
    SSLKeyRepositoryPassword=passw0rd
```

注意: 如果使用此方法提供密码，请在设置 **SSLKeyRepositoryPassword** 属性的值之前对密码进行加密。 有关更多信息，请参阅第 251 页的『加密密钥存储库密码』。

有关客户机配置文件的 SSL 节的更多信息，请参阅客户机配置文件的 SSL 节。

密钥存储库隐藏文件

如果未使用其他方法之一向客户机提供密钥存储库密码，那么 IBM MQ 会假定存储文件与密钥存储库存在于同一目录中。 隐藏文件具有与密钥存储库相同的主干名称，但具有 `.sth` 扩展名。

密钥存储库隐藏文件是使用 **amqrsslc** 命令行工具创建的。 要创建隐藏文件，请运行以下命令:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-s' '/Path/Of/KeyDatabase/MyKey')
```

此命令提示您输入要加密的密码。 密码由 IBM MQ 密码保护系统进行加密，除非使用 **-sf** 参数提供了缺省加密密钥。

有关更多信息，请参阅第 258 页的『IBM MQ IBM i 的 SSL 客户机实用程序 (amqrsslc)』和第 251 页的『加密密钥存储库密码』。

加密密钥存储库密码

如果使用除隐藏文件以外的任何方法提供密钥存储库密码，请使用 IBM MQ 密码保护系统对密码进行加密。 要对密码进行加密，请运行 **runmqicred** 命令。 提示时输入密钥存储库密码。 该命令输出加密密码。 可以使用描述的任何方法将加密密码提供给 IBM MQ MQI client ，而不是纯文本密码。

加密密钥 (称为初始密钥) 用于加密密码。 加密密码时，请使用唯一的初始密钥来安全地保护密码。 要提供您自己的初始密钥，请对 **runmqicred** 命令使用 **-sf** 参数。 如果未提供初始密钥，那么将使用缺省密钥。

有关更多信息，请参阅 runmqicred (保护 IBM MQ 客户机密码)。

如果在加密密钥存储库密码时提供自己的初始密钥，并向 IBM MQ MQI client 提供加密密码，那么还必须确保向 IBM MQ MQI client 提供相同的初始密钥。 有关如何向 IBM MQ MQI client 提供初始密钥的更多信息，请参阅第 251 页的『为 IBM i 上的 IBM MQ MQI client 提供初始密钥』。

相关概念

第 246 页的『对 IBM i 上的密钥存储库密码进行加密』
多个 IBM MQ 组件需要访问包含数字证书或对称密钥的密钥存储库。 密钥存储库使用密码进行保护，因为它包含敏感信息。 访问密钥存储库时，必须将密钥存储库密码存储在 IBM MQ 可以读取该密码的位置。 还必须对密码进行加密，以降低对密钥存储库进行未经授权的访问的可能性。

第 249 页的『为 IBM i 上的队列管理器提供密钥存储库密码』
因为密钥存储库包含敏感信息，所以使用密码对其进行保护。 为了能够访问密钥存储库内容以执行 TLS 操作， IBM MQ 必须能够检索密钥存储库密码。

IBM i 为 *IBM i* 上的 *IBM MQ MQI client* 提供初始密钥
如果向已使用 IBM MQ 密码保护系统加密的 IBM MQ MQI client 提供变量，那么可能需要提供用于加密值的相应初始密钥。

如果在加密值时未指定初始密钥，那么不需要向 IBM MQ client 提供任何初始密钥值。 但是，如果您使用了唯一的初始密钥，那么可以使用以下方法向 IBM MQ client 提供初始密钥:

- 第 252 页的『使用 MQCSP 结构提供初始密钥』
- 第 252 页的『使用 MQS_MQI_KEYFILE 环境变量提供初始密钥』
- 第 252 页的『使用客户机配置文件提供初始密钥』

使用 MQCSP 结构提供初始密钥

要使用 MQCSP 结构提供初始键，必须使用以下三个变量字符串字段的组合:

InitialKeyLength
初始键的长度

InitialKeyPtr
指向内存中包含初始键的位置的指针

InitialKeyOffset
初始密钥在内存中的位置，表示为从 MQCSP 结构开始的字节数。

注: 只能提供 **InitialKeyPtr** 或 **InitialKeyOffset** 中的一个。

例如:

```
char * initialKey = "myInitialKey";
MQCSP  cspOptions = {MQCSP_DEFAULT};


cspOptions.InitialKeyPtr = initialKey;
cspOptions.InitialKeyLength = (MQLONG)strlen(cspOptions.InitialKeyPtr);
cspOptions.Version = MQCSP_VERSION_2;
```

使用 MQS_MQI_KEYFILE 环境变量提供初始密钥

如果未使用 MQCSP 结构向客户机提供初始密钥，那么 IBM MQ 将检查 *MQS_MQI_KEYFILE* 环境变量。 您应该将此环境变量设置为包含单行文本 (由您要使用的初始键组成) 的文件的位置。

例如，如果根目录中存在名为 mykey.key 的文件，并且该文件包含初始键，那么应按如下所示设置环境变量:

```
export MQS_MQI_KEYFILE=/mykey.key
```

或

```
set MQS_MQI_KEYFILE=C:\mykey.key
```

使用客户机配置文件提供初始密钥

如果未使用先前机制向客户机提供初始密钥，那么 IBM MQ 会检查 mqclient.ini 文件的 Security 节的 **MQIInitialKeyFile** 属性。 您应该将此属性设置为包含单行文本的文件的位置，该文本由您要使用的初始键组成。

例如，如果根目录中存在名为 mykey.key 的文件，并且该文件包含初始密钥，那么客户机配置文件应包含以下内容:

```
Security:
    MQIInitialKeyFile=/mykey.key
```

相关概念
第 246 页的『对 IBM i 上的密钥存储库密码进行加密』
多个 IBM MQ 组件需要访问包含数字证书或对称密钥的密钥存储库。 密钥存储库使用密码进行保护，因为它包含敏感信息。 访问密钥存储库时，必须将密钥存储库密码存储在 IBM MQ 可以读取该密码的位置。 还必须对密码进行加密，以降低对密钥存储库进行未经授权的访问的可能性。

第 245 页的『在 IBM i 上使用 SSL/TLS』
此主题集合提供有关在 IBM MQ for IBM i 中使用传输层安全性 (TLS) 的个别任务的指示信息。

创建认证中心和证书以在 *IBM i* 上进行测试

使用此过程可创建本地 CA 证书以签署证书请求，以及创建和安装 CA 证书。

开始之前

本主题中的指示信息假定本地认证中心 (CA) 不存在。 如果存在本地 CA，请转至 第 254 页的『在 IBM i 上请求服务器证书』。

关于此任务

安装 TLS 时提供的 CA 证书由发出 CA 签名。 在 IBM i 上，您可以生成本地认证中心，该认证中心可以签署用于在系统上测试 TLS 通信的服务器证书。 在 Web 浏览器中执行以下步骤以创建本地 CA 证书:

过程

1. 访问 DCM 接口，如 第 245 页的『访问 DCM』中所述。
2. 在导航面板中，单击 **创建认证中心**。
   "创建认证中心" 页面将显示在任务框架中。
3. 在 **证书库密码** 字段中输入密码，然后在 **确认密码** 字段中再次输入密码。
4. 在 **认证中心 (CA) 名称** 字段中输入名称，例如 `TLS Test Certificate Authority`。
5. 在 **公共名称** 和 **组织** 字段中输入相应的值，然后选择国家或地区。 对于其余可选字段，请输入所需的值。
6. 在 **有效期** 字段中输入本地 CA 的有效期。
   缺省值为 1095 天。
7. 单击**继续**。
   将创建 CA，并且 DCM 将为本地 CA 创建证书库和 CA 证书。
8. 单击 **安装证书**。
   此时将显示 "下载管理器" 对话框。
9. 输入要在其中存储 CA 证书的临时文件的完整路径名，然后单击 **保存**。
10. 下载完成后，单击 **打开**。
    此时将显示 "证书" 窗口。
11. 单击 **安装证书**。
    将显示 "证书导入" 向导。
12. 单击**下一步**。
13. 选择 **根据证书类型自动选择证书库**，然后单击 **下一步**。
14. 单击**完成**。
    这将显示确认窗口。
15. 单击**确定**。
16. 在 "证书" 窗口中，单击 **确定**。
17. 单击**继续**。
    "认证中心策略" 页面将显示在任务框架中。
18. 在 **允许创建用户证书** 字段中，选择 **是**。
19. 在 **有效期** 字段中，输入本地 CA 发放的证书的有效期。
    缺省值为 365 天。
20. 单击**继续**。
    "在新证书库中创建证书" 页面将显示在任务框架中。
21. 检查是否未选择任何应用程序。
22. 单击 **继续** 以完成本地 CA 的设置。

下一步做什么

如果需要更新现有证书，请参阅 IBM i 文档中的 更新现有证书 。

在 *IBM i* 上请求服务器证书

数字证书通过证明公用密钥属于指定实体来防止假冒。 可以使用数字 Certificate Manager (DCM) 从认证中心请求新的服务器证书。

关于此任务

在 Web 浏览器中执行以下步骤:

过程

1. 访问 DCM 接口，如 第 245 页的『访问 DCM』 中所述。
2. 在导航面板中，单击 **选择证书库**。
   "选择证书库" 页面将显示在任务框架中。
3. 选择要使用的证书库，然后单击 **继续**。
4. 可选： 如果在步骤 3 中选择了 ***SYSTEM** ，请输入系统存储密码并单击 **继续**。
5. 可选： 如果在步骤 3 中选择了 **其他系统证书库** ，请在 **证书库路径和文件名** 字段中输入创建证书库时设置的 IFS 路径和文件名。 还请在 **证书库密码** 字段中输入密码。 然后单击 **继续**
6. 在导航面板中，单击 **创建证书**。
7. 在任务框架中，选择 **服务器或客户机证书** 单选按钮，然后单击 **继续**。
   "选择认证中心 (CA)" 页面将显示在任务框架中。
8. 如果您的工作站上有本地 CA ，请选择本地 CA 或商业 CA 来签署证书。 选择所需 CA 的单选按钮，然后单击 **继续**。
   "创建证书" 页面将显示在任务框架中。
9. 可选： 对于队列管理器，在 **证书标签** 字段中，输入证书标签。
   标签是 **CERTLABL** 属性的值 (如果设置了此属性) ，或者是附加了队列管理器名称的缺省 `ibmwebspheremq` (全部为小写)。 请参阅 数字证书标签 以获取详细信息。
   例如，对于队列管理器 QM1，输入 `ibmwebspheremqqm1` 以使用缺省值。
10. 可选： 对于 IBM MQ MQI client，在 **证书标签** 字段中，输入 `ibmwebspheremq` ，后跟转换为小写的登录用户标识。
    例如，输入 `ibmwebspheremqmyuserID`
11. 在 **公共名称** 和 **组织** 字段中输入相应的值，然后选择国家或地区。 对于其余可选字段，请输入所需的值。

结果

如果选择了商业 CA 来签署证书，那么 DCM 将以 PEM (隐私增强邮件) 格式创建证书请求。 将请求转发到所选 CA。

如果您选择了本地 CA 来签署证书，那么 DCM 会通知您证书已在证书库中创建，并且可以使用该证书。

为远程系统请求服务器证书 *IBM i*

按照此过程创建由本地证书颁发机构 (CA) 签名的证书，或申请由商业 CA 签名的服务器证书以导入其他平台上的密钥存储库。

关于此任务

当数字 Certificate Manager (DCM) 在多个平台上充当 IBM MQ 的证书管理器时，必须使用用户证书。 对于分发到其他平台并导入密钥存储库的个人证书，请在 Web 浏览器中执行以下步骤:

过程

1. 访问 DCM 接口，如 第 245 页的『访问 DCM』 中所述。

2. 在 " **导航** " 窗格中，单击 **创建证书**。
   " **创建证书** " 页面将显示在任务框架中。
3. 在 " **创建证书** " 面板上，选择 **用户证书** 单选按钮，然后单击 **继续**。
   这将显示 " **创建用户证书** " 页面。
4. 在 " **创建用户证书** " 面板上，填写 **组织名称**， **状态** 或 **省/直辖市/自治区**， **国家或地区** 或 **区域**的 "证书信息" 下的必填字段。 (可选) 将值放在 **组织单元** 和 **位置** 或 **城市** 字段中。 单击**继续**。
   公共名称 将自动设置为用于登录到 iSeries 系统的用户标识。
5. 在下一个 " **创建用户证书** " 面板上，单击 **安装证书** ，然后单击 **继续**。
   将显示一条消息，说明 **您的个人证书已安装。 您应该保留此证书的备份副本。**
6. 单击**确定**。
7. 根据您用于访问 DCM 的网络浏览器，完成以下步骤之一：
   - 对于 Microsoft Edge ，请选择: **工具> Internet 选项> "内容" 选项卡> "证书" 按钮> "个人" 选项卡>**。选择证书，然后单击 **导出**。
   - 对于 Mozilla Firefox ，请选择: **工具> 选项> 高级> 加密选项卡> "查看证书" 按钮> "证书" 选项卡>**。选择证书，然后单击 **备份**。 选择路径和文件名，然后单击 **确定**。
8. 使用二进制格式的 FTP 将导出的证书传输到远程系统。
9. 导入步骤中导出的证书第 255 页的『7』到远程系统上的密钥存储库。
   - 如果使用以下方式保存证书 MicrosoftEdge，请使用第 485 页的『从 Microsoft .pfx 文件导入个人证书』文件。
   - 如果证书是使用 Mozilla Firefox 保存的，请使用 将个人证书导入密钥存储库中描述的指示信息。

   导入过程中，确保个人证书和签名者证书的标签名称更改为 IBM MQ 期望。 标签必须是 IBM MQ 队列管理器 **CERTLABL** 属性（如果已设置）或默认值 `ibmwebspheremq` 附加队列管理器的名称，全部小写。有关详细信息，请参阅 For more information, see 数字证书标签。

将服务器证书添加到 *IBM i* 上的密钥存储库

遵循此过程以将请求的证书添加到密钥存储库。

关于此任务

CA 向您发送新的服务器证书后，将其添加到从中生成请求的证书库。 如果 CA 将证书作为电子邮件消息的一部分发送，请将证书复制到单独的文件中。

注:

- 如果服务器证书由本地 CA 签署，那么无需执行此过程。
- 在将 PKCS #12 格式的服务器证书导入 DCM 之前，必须先导入相应的 CA 证书。

使用以下过程将服务器证书接收到队列管理器证书库中:

过程

1. 访问 DCM 接口，如 第 245 页的『访问 DCM』中所述。
2. 在导航面板中的 **管理证书** 任务类别中，单击 **导入证书**。
   "导入证书" 页面将显示在任务框架中。
3. 选择证书类型的单选按钮，然后单击 **继续**。
   "导入服务器" 或 "客户机证书" 页面或 "导入认证中心 (CA) 证书" 页面将显示在任务框架中。
4. 在 **导入文件** 字段中，输入要导入的证书的文件名，然后单击 **继续**。
   DCM 自动确定文件的格式。
5. 如果证书是 **服务器或客户机** 证书，请在任务框架中输入密码，然后单击 **继续**。
   DCM 通知您已导入证书。

从 *IBM i* 上的密钥存储库中导出证书

导出证书将导出公用密钥和专用密钥。 应该非常谨慎地执行此操作，因为传递专用密钥将完全损害您的安全性。

开始之前

当您与其他用户共享用户的证书时，将交换公用密钥。 此过程在 **任务 5 中进行了描述**。 **第 535 页的『AIX and Linux 上 AMS 的快速入门指南』的 共享证书 部分中的 共享证书** 。 按此处所述导出证书时，将同时导出公用密钥和专用密钥。 应该非常谨慎地执行此操作，因为传递专用密钥将完全损害您的安全性。

关于此任务

在要从中导出证书的计算机上执行以下步骤:

过程

1. 访问 DCM 接口，如 第 245 页的『访问 DCM』 中所述。
2. 在导航面板中，单击 **选择证书库**。
   "选择证书库" 页面将显示在任务框架中。
3. 选择要使用的证书库，然后单击 **继续**。
4. 可选： 如果在步骤 3 中选择了 ***SYSTEM** ，请输入系统存储密码并单击 **继续**。
5. 可选： 如果在步骤 3 中选择了 **其他系统证书库** ，请在 **证书库路径和文件名** 字段中输入创建证书库时设置的 IFS 路径和文件名，并在 **证书库密码** 字段中输入密码。 然后单击 **继续**
6. 在导航面板中的 **管理证书** 任务类别中，单击 **导出证书**。
   "导出证书" 页面将显示在任务框架中。
7. 选择证书类型的单选按钮，然后单击 **继续**。
   "导出服务器" 或 "客户机证书" 页面或 "导出认证中心 (CA) 证书" 页面将显示在任务框架中。
8. 选择要导出的证书。
9. 选择单选按钮以指定是要将证书导出到文件还是直接导出到另一个证书库。
10. 如果选择将服务器或客户机证书导出到文件，请提供以下信息:
    - 要用于存储导出的证书的位置的路径和文件名。
    - 对于个人证书，这是用于对导出的证书和目标发行版进行加密的密码。 对于 CA 证书，您不需要指定密码。
11. 如果选择将证书直接导出到另一个证书库中，请指定目标证书库及其密码。
12. 单击**继续**。

将证书导入到 *IBM i* 上的密钥存储库中

遵循此过程以导入证书。

开始之前

在将 PKCS #12 格式的个人证书导入 DCM 之前，必须首先导入相应的 CA 证书。

关于此任务

在要将证书导入到的机器上执行这些步骤。

过程

1. 访问 DCM 接口，如 第 245 页的『访问 DCM』 中所述。
2. 在导航面板中，单击 **选择证书库**。
   "选择证书库" 页面将显示在任务框架中。
3. 选择要使用的证书库，然后单击 **继续**。
4. 可选： 如果在步骤 3 中选择了 ***SYSTEM** ，请输入系统存储密码并单击 **继续**。

5. 可选： 如果在步骤 3 中选择了 **其他系统证书库** ，请在 **证书库路径和文件名** 字段中输入创建证书库时设置的 IFS 路径和文件名，并在 **证书库密码** 字段中输入密码。 然后单击 **继续**
6. 在导航面板中的 **管理证书** 任务类别中，单击 **导入证书**。
   "导入证书" 页面将显示在任务框架中。
7. 选择证书类型的单选按钮，然后单击 **继续**。
   将在任务框架中显示 "导入服务器" 或 "客户机证书" 页面或 "导入认证中心 (CA) 证书" 页面。
8. 在 **导入文件** 字段中，输入要导入的证书的文件名，然后单击 **继续**。
   DCM 自动确定文件的格式。
9. 如果证书是 **服务器或客户机** 证书，请在任务框架中输入密码，然后单击 **继续**。 DCM 通知您已导入证书。

在 *IBM i* 中除去证书

使用此过程可除去个人证书。

过程

1. 访问 DCM 接口，如 第 245 页的『访问 DCM』 中所述。
2. 在导航面板中，单击 **选择证书库**。
   "选择证书库" 页面将显示在任务框架中。
3. 选中 **其他系统证书库** 复选框，然后单击 **继续**。
   将显示 "证书库和密码" 页面。
4. 在 **证书库路径和文件名** 字段中，输入创建证书库时设置的 IFS 路径和文件名。
5. 在 **证书库密码** 字段中输入密码。 单击**继续**。
   "当前证书库" 页面将显示在任务框架中。
6. 在导航面板中的 **管理证书** 任务类别中，单击 **删除证书**。
   "确认删除证书" 页面将显示在任务框架中。
7. 选择要删除的证书。 单击**删除**。
8. 单击 **是** 以确认要删除证书。 否则，请单击**否**。
   DCM 会通知您是否已删除证书。

在 *IBM i* 上使用 **SYSTEM* 证书库进行单向认证

遵循以下指示信息来设置单向认证。

开始之前

- 创建队列管理器，通道和传输队列。
- 在服务器队列管理器上创建服务器或客户机证书。
- 将 CA 证书传输到客户机队列管理器并将其导入到密钥存储库中。
- 在服务器和客户机队列管理器上启动侦听器。

关于此任务

要使用单向认证 (使用运行 IBM i 的计算机作为 TLS 服务器) ，请将 SSL 密钥存储库 (SSLKEYR) 参数设置为 *SYSTEM。 此设置将 IBM MQ 队列管理器注册为应用程序。 然后，可以将证书分配给队列管理器以启用单向认证。

您还可以使用专用密钥库，通过在密钥存储库中为客户机队列管理器创建哑元证书来实现单向认证。

过程

1. 在服务器和客户机队列管理器上执行以下步骤:

a) 通过发出命令 CHGMQM MQMNAME(SSL) SSLKEYR(*SYSTEM)来更改队列管理器以设置 SSLKEYR 参数。

b) 通过发出命令 CHGMQM MQMNAME(SSL) SSLKEYRPWD('xxxxxxx')来隐藏缺省密钥存储库的密码。

密码必须使用单引号。

c) 更改通道以在 SSLPASSWORD 参数中具有正确的 CipherSpec 。

d) 通过发出命令 RFRMQMAUT QMNAME(QMGRNAME) TYPE(*SSL)来刷新 TLS 安全性。

2. 使用 DCM 将证书分配给服务器队列管理器，如下所示:

a) 访问 DCM 接口，如 第 245 页的『访问 DCM』中所述。

b) 在导航面板中，单击 **选择证书库**。

"选择证书库" 页面将显示在任务框架中。

c) 选择 *SYSTEM 证书库，然后单击 **继续**。

d) 在左侧面板中，展开 **管理应用程序**。

e) 选择 **查看应用程序** 定义以检查队列管理器是否已注册为应用程序。

表中列出了 SSL （WMQ）。

f) 选择 **更新证书分配**。

g) 选择 **服务器** ，然后单击 **继续**。

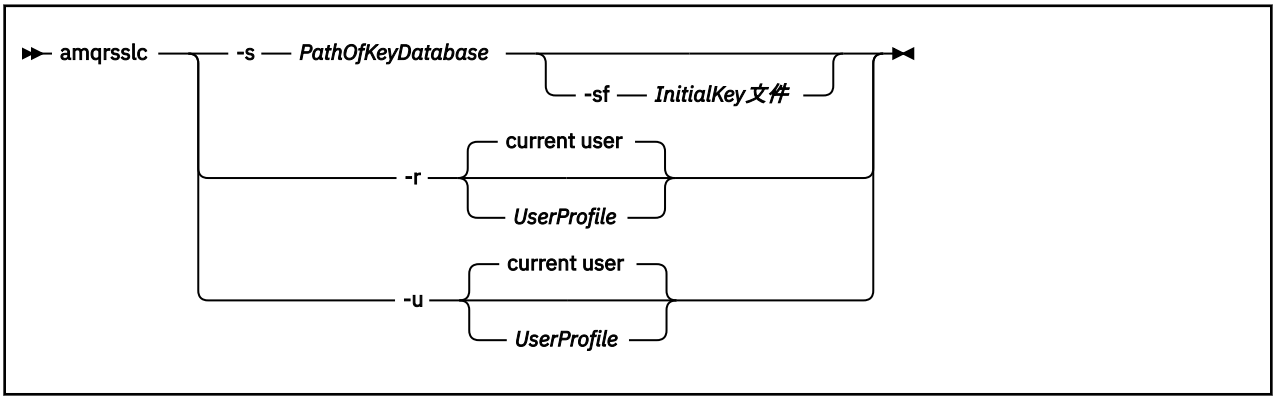
h) 选择 QMGRNAME (WMQ) ，然后单击 **更新证书分配**。

i) 选择证书，然后单击 **分配新证书**。 这将打开一个窗口，指示已将证书分配给应用程序。

IBM MQ IBM i 的 SSL 客户机实用程序 (amqrsslc)

IBM i 系统上的 IBM MQ MQI client 使用 IBM i 的 IBM MQ SSL 客户机实用程序 (amqrsslc) 来注册或注销客户机用户概要文件，或者隐藏证书库密码。 该实用程序只能由具有 *ALLOBJ 特权的概要文件的用户或具有用于在 "数字 Certificate Manager " (DCM) 中创建或删除应用程序注册的选项的 QMQMADM 成员运行。

语法图



注册客户机用户概要文件

如果 IBM MQ MQI client 正在使用 *SYSTEM 证书库，那么必须注册客户机用户概要文件 (登录用户) 以用作具有 数字 Certificate Manager (DCM)的应用程序。

如果要注册客户机用户概要文件，请使用 -r 选项和 *UserProfile* 运行 **amqrsslc** 程序。 调用 **amqrsslc** 时使用的用户概要文件必须具有 *USE 权限。 使用 -r 选项提供 *UserProfile* 将 *UserProfile* 注册为具有唯一应用程序标签 QIBM_WEBSPHERE_MQ_*UserProfile* 和描述为 *UserProfile* (WMQ) 的标签的服务器应用程序。 然后，此服务器应用程序将显示在 DCM 中，并且您可以将系统存储库中的任何服务器或客户机证书分配给此应用程序。

注: 如果未使用 -r 选项指定用户概要文件，那么将注册运行 **amqrsslc** 工具的用户的用户概要文件。

以下代码使用 **amqrsslc** 来注册用户概要文件。 在第一个示例中，已注册指定的用户概要文件; 在第二个示例中，这是已登录用户的概要文件:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-r' UserProfile)
CALL PGM(QMQM/AMQRSSLC) PARM('-r')
```

注销客户机用户概要文件

要注销客户机概要文件，请使用带有 *UserProfile* 的 -u 选项运行 **amqrsslc** 程序。 调用 **amqrsslc** 时使用的用户概要文件必须具有 *USE 权限。 使用 -u 选项提供 *UserProfile* ，将 *UserProfile* 的标签为 QIBM_WEBSPHERE_MQ_*UserProfile* 从 DCM 中注销。

注: 如果未使用 -u 选项指定用户概要文件，那么将注销运行 **amqrsslc** 工具的用户的用户概要文件。

以下代码使用 **amqrsslc** 来注销用户概要文件。 在第一个示例中，指定的用户概要文件已注销; 在第二个示例中，它是已登录用户的概要文件:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-u' UserProfile)
CALL PGM(QMQM/AMQRSSLC) PARM('-u')
```

隐藏证书库密码

如果 IBM MQ MQI client 未使用 *SYSTEM 证书库并使用另一个证书库 (即， MQSSLKEYR 设置为 *SYSTEM 以外的值) ，那么可以隐藏密钥数据库的密码，以便客户机应用程序在运行时不需要指定该密码。

使用 -s 选项来隐藏密钥数据库的密码。 指定密钥数据库的完整路径和名称。 如果未提供文件扩展名，那么假定其为 .kdb。

在以下代码中，证书库的标准文件名为 /Path/Of/KeyDatabase/MyKey.kdb:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-s' '/Path/Of/KeyDatabase/MyKey')
```

运行此代码会导致请求此密钥数据库的密码。 此密码隐藏在与具有 .sth 扩展名的密钥数据库同名的文件中。

此外，可以指定用于加密密码的初始密钥。 应将初始键作为单行文本存储在文件中，然后通过 **-sf** 标志将该文件的位置提供给程序。 如果未提供初始密钥文件，那么将使用缺省密钥对密码进行加密。

隐藏文件存储在与密钥数据库相同的路径上。 代码示例生成 /Path/Of/KeyDatabase/MyKey.sth 的隐藏文件。

QMQM 是用户所有者， QMQMADM 是此文件的组所有者。 QMQM 和 QMQMADM 具有读，写许可权，其他概要文件仅具有读许可权。

当对证书或证书库的更改在 *IBM i* 生效时

当您更改证书库中的证书或证书库的位置时，这些更改将根据通道的类型以及通道的运行方式而生效。

在以下情况下，对证书库中证书的更改以及对密钥存储库属性的更改将生效:

- 当新的出站单通道进程首次运行 TLS 通道时。
- 当新的入站 TCP/IP 单通道进程首先接收到启动 TLS 通道的请求时。
- 发出 MQSC 命令 REFRESH SECURITY TYPE (SSL) 以刷新 IBM MQ TLS 环境时。
- 对于客户机应用程序进程，当进程中的最后一个 TLS 连接关闭时。 下一个 TLS 连接将选取证书更改。
- 对于作为进程池进程 (amqrmppa) 的线程运行的通道，当进程池进程启动或重新启动并首先运行 TLS 通道时。 如果进程池进程已运行 TLS 通道，并且您希望更改立即生效，请运行 MQSC 命令 REFRESH SECURITY TYPE (SSL)。
- 对于作为通道启动程序线程运行的通道，当通道启动程序启动或重新启动时，首先运行 TLS 通道。 如果通道启动程序进程已运行 TLS 通道，并且您希望更改立即生效，请运行 MQSC 命令 REFRESH SECURITY TYPE (SSL)。

- 对于作为 TCP/IP 侦听器的线程运行的通道，当侦听器启动或重新启动时，首先接收到启动 TLS 通道的请求。 如果侦听器已运行 TLS 通道，并且您希望更改立即生效，请运行 MQSC 命令 REFRESH SECURITY TYPE (SSL)。

在 *IBM i* 上配置加密硬件

使用此过程在 IBM i 上配置加密协处理器

开始之前

确保您的用户概要文件具有 *ALLOBJ 和 *SECADM 特权，以使您能够配置协处理器硬件。

过程

1. 转至 `http://machine.domain:2001` 或 `https://machine.domain:2010`，其中 *machine* 是计算机的名称。

   将显示一个对话框，请求用户名和密码。
2. 输入有效的 IBM i 用户概要文件和密码。
3. 转至 密码术 并访问相应的链接以获取更多信息。

下一步做什么

有关配置 4767 Cryptographic Coprocessor 的更多具体信息，请参阅 4767 Cryptographic Coprocessor。

ALW 在 AIX, Linux, and Windows 上使用 SSL/TLS

在 AIX, Linux, and Windows 系统上，传输层安全性 (TLS) 支持随 IBM MQ 一起安装。

注: V 9.4.0 Deprecated 从 IBM MQ 9.4.0 开始，不推荐将 CMS 密钥存储库和隐藏文件与 IBM MQ Java 应用程序配合使用。 迁移到使用 PKCS #12 密钥存储库，并使用 IBM MQ 密码保护系统来保护密钥存储库密码。

要点: V 9.4.0 V 9.4.0 从 IBM MQ 9.4.0 开始，使用 SSL/TLS 的 AMQP 和 MQTT 通道不支持 CMS 密钥存储库和隐藏文件。 使用 PKCS #12 密钥存储库，并改为使用 IBM MQ 密码保护系统来保护密钥存储库密码。

有关证书验证策略的更多详细信息，请参阅 证书验证和信任策略设计。

有关用于在 AIX, Linux, and Windows 上管理密钥存储库和证书的命令的更多信息，请参阅 第 473 页的『AIX, Linux, and Windows 上的 runmqakm 和 runmqktool 命令』。

ALW 在 *AIX, Linux, and Windows* 上设置密钥存储库

遵循此过程以创建新的密钥存储库。

开始之前

密钥存储库使用密码进行保护，因为它包含敏感信息。 在创建密钥存储库之前，请查看 IBM MQ 提供的用于安全存储密钥存储库密码的选项。 有关更多信息，请参阅 第 263 页的『对 AIX, Linux, and Windows 上的密钥存储库密码进行加密』。

注: V 9.4.0 Deprecated 从 IBM MQ 9.4.0 开始，不推荐将 CMS 密钥存储库和隐藏文件与 IBM MQ Java 应用程序配合使用。 迁移到使用 PKCS #12 密钥存储库，并使用 IBM MQ 密码保护系统来保护密钥存储库密码。

要点: V 9.4.0 V 9.4.0 从 IBM MQ 9.4.0 开始，使用 SSL/TLS 的 AMQP 和 MQTT 通道不支持 CMS 密钥存储库和隐藏文件。 使用 PKCS #12 密钥存储库，并改为使用 IBM MQ 密码保护系统来保护密钥存储库密码。 您可以使用以下命令创建 PKCS #12 密钥存储库:

```
runmqakm -keydb -create -db filename.p12 -pw password -type pkcs12
```

此命令创建名为 *filename*.p12 的 PKCS #12 密钥存储库文件，该文件使用指定的密码进行保护。

关于此任务

TLS 连接的每一端都需要一个 密钥存储库 。 每个 IBM MQ 队列管理器和 IBM MQ MQI client 都必须有权访问密钥存储库。 有关更多信息，请参阅第 22 页的『SSL/TLS 密钥存储库』。

数字证书存储在密钥存储库中。 这些数字证书具有标签。 证书标签将个人证书与特定队列管理器或 IBM MQ MQI client 相关联。 TLS 将该证书用于认证目的。 在 AIX, Linux, and Windows 系统上， IBM MQ 将下列其中一个值用于证书标签:

- **CERTLABL** 队列管理器或通道属性的值 (如果已设置)。
- 缺省值 `ibmwebspheremq`，附加了队列管理器或 IBM MQ MQI client 用户登录标识的名称，全部为小写。

有关更多信息，请参阅 数字证书标签。

密钥存储库文件名包含路径和主干名称:

- 在 AIX and Linux 系统上，队列管理器 (在创建队列管理器时设置) 的缺省路径为 `/var/mqm/qmgrs/`*`queue_manager_name`*`/ssl`。

  在 Windows 系统上，缺省路径为 *`MQ_DATA_PATH`*`\qmgrs\`*`queue_manager_name`*`\ssl`，其中 *`MQ_DATA_PATH`* 是安装 IBM MQ 期间选择的数据路径。 例如，`C:\ProgramData\IBM\MQ\qmgrs\QM1\ssl`。

  缺省文件名为 `key.kdb`。 或者，您可以使用自己的路径和文件名。

  如果您选择自己的路径或文件名，请设置对该文件的许可权以严格控制对该文件的访问。
- 对于 IBM MQ 客户机，没有缺省路径或文件名。 严格控制对此文件的访问。

请勿在不支持文件级别锁定的文件系统上创建密钥存储库，例如 Linux 系统上的 NFS V 2。

有关检查和指定密钥数据库文件名的信息，请参阅 第 265 页的『在 AIX, Linux, and Windows 上更改队列管理器的密钥存储库位置』。 可以在创建密钥存储库之前或之后指定密钥数据库文件名。

您可以使用 **runmqakm** (GSKCapiCmd) 或 V 9.4.0 V 9.4.0 **runmqktool** (keytool) 命令来管理 IBM MQ 所使用的密钥存储库。 有关更多信息，请参阅第 473 页的『AIX, Linux, and Windows 上的 runmqakm 和 runmqktool 命令』。

运行用于管理密钥存储库的命令的用户标识必须对在其中创建或更新密钥存储库文件的目录具有写许可权。 对于使用缺省 `ssl` 目录的队列管理器，运行 **runmqakm** 或 **runmqktool** 命令的用户标识必须是 mqm 组的成员。 对于 IBM MQ MQI client，如果从与运行客户机的用户标识不同的用户标识运行 **runmqakm** 或 **runmqktool** ，那么必须变更文件许可权以允许 IBM MQ MQI client 访问密钥存储库。 有关更多信息，请参阅第 264 页的『在 Windows 上访问和保护密钥数据库文件』或第 264 页的『在 AIX and Linux 系统上访问和保护密钥数据库文件』。

您可以使用 **runmqakm** 命令创建新的空密钥存储库。 V 9.4.0 V 9.4.0 如果改为使用 **runmqktool** 命令，那么将在发出命令以创建或导入证书时创建密钥存储库。

注: 如果必须以符合 FIPS 的方式管理 TLS 证书，请使用 **runmqakm** 命令。

过程

1. 发出以下命令以使用 **runmqakm** 命令创建密钥存储库:

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

   其中:

   -db 文件名
   指定密钥存储库的标准文件名。

   -pw *password*
   指定密钥存储库的密码。

-type 类型

V 9.4.0 V 9.4.0 指定密钥存储库的类型。 对于 IBM MQ 使用的密钥存储库，可能的值为:

- pkcs12
- Deprecated 厘米

注: 从 IBM MQ 9.4.0 开始，不推荐对 IBM MQ Java 应用程序使用 CMS 密钥存储库和隐藏文件，使用 SSL/TLS 的 AMQP 和 MQTT 通道不支持这些文件。

-stash

可选。 指定此选项以将密钥存储库密码存储在隐藏文件中。 如果改为使用 IBM MQ 密码保护系统对密码进行加密，那么无需将密码存储在隐藏文件中。

-无花果

指定以 FIPS 方式运行命令。 在 FIPS 方式下， IBM Crypto for C (ICC) 组件使用经 FIPS 140-2 验证的算法。 如果 ICC 组件未以 FIPS 方式初始化，那么 **runmqakm** 命令将失败。

-强

检查输入的密码是否满足密码强度的最低要求。 密码的最低要求如下:

- 密码长度必须至少为 14 个字符。
- 密码必须至少包含一个小写字符，一个大写字符以及一个数字或特殊字符。 特殊字符包括星号 (*) ，美元符号 ($) ，数字符号 (#) 和百分号 (%)。 空格被分类为特殊字符。
- 每个字符最多可以在密码中出现三次。
- 密码中最多两个连续字符可以相同。
- 所有字符都在标准 ASCII 可打印字符集中，范围为 0x20 - 0x7E。

2. 设置密钥存储库文件的访问许可权，如 第 264 页的『在 Windows 上访问和保护密钥数据库文件』 或 第 264 页的『在 AIX and Linux 系统上访问和保护密钥数据库文件』中所述。

   在 Windows 上，缺省情况下，仅向运行用于创建密钥存储库的命令的用户标识授予读取隐藏 (.sth) 文件的访问权。 使用 **runmqakm** 命令创建隐藏文件后，请检查文件许可权，并将许可权授予运行队列管理器的服务帐户或本地 mqm 之类的组。
3. 如果未使用隐藏文件，请遵循 第 265 页的『为 AIX, Linux, and Windows 上的队列管理器提供密钥存储库密码』 或 第 267 页的『为 AIX, Linux, and Windows 上的 IBM MQ MQI client 提供密钥存储库密码』中的指示信息向队列管理器或客户机应用程序提供密钥库密码。

下一步做什么

如果需要，请将缺省认证中心 (CA) 证书添加到空密钥存储库。 有关更多信息，请参阅第 264 页的『将缺省 CA 证书添加到 AIX, Linux, and Windows 上的空密钥存储库中』。

ALW 在 *AIX, Linux, and Windows* 上生成用于密钥存储库保护的强密码

您可以使用 **runmqakm** (GSKCapiCmd) 命令生成用于密钥存储库保护的强密码。

可以使用带有以下参数的 **runmqakm** 命令来生成强密码:

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

其中 *password_length* 是要生成的密码的长度。 可以指定的最小密码长度为 14。

在后续证书管理命令的 **-pw** 参数上使用生成的密码时，请始终将密码括在双引号内。 在 AIX and Linux 系统上，如果下列字符出现在密码字符串中，那么还必须使用反斜杠字符对其进行转义:

```
!  \  "  '
```

当您输入密钥存储库密码以响应来自 **runmqakm** 或 V 9.4.0 V 9.4.0 **runmqktool** 命令的提示时，不必引用或转义密码，因为在这些情况下，操作系统 shell 不会影响数据输入。

ALW 对 *AIX, Linux, and Windows* 上的密钥存储库密码进行加密

多个 IBM MQ 组件需要访问包含数字证书或对称密钥的密钥存储库。 密钥存储库使用密码进行保护，因为它包含敏感信息。 访问密钥存储库时，必须将密钥存储库密码存储在 IBM MQ 可以读取该密码的位置。 还必须对密码进行加密，以降低对密钥存储库进行未经授权的访问的可能性。

以下 IBM MQ 组件和功能部件支持两种不同的方法来存储密钥存储库密码:

- 队列管理器 TLS 密钥存储库。
- 使用 TLS 的 IBM MQ MQI clients 。
- V 9.4.0 qm.ini 文件的 **NativeHALocalInstance** 节中的本机 HA 配置。
- V 9.4.0 qm.ini 文件的 **AuthToken** 节中的令牌认证配置。

可使用下列其中一种方法对供这些组件使用的密钥存储库密码进行加密和存储:

IBM MQ 密码保护系统。

每个 IBM MQ 组件都提供了用于加密密钥存储库密码的命令。 命令输出的加密命令存储在文件中。

对于队列管理器 TLS 密钥存储库，将在设置 **SSLKEYRPWD** 队列管理器属性时加密密码。

使用 AES-128 算法对密码进行加密。 此算法的详细信息是公开的，并且被认为是安全的。

密码以其他可能访问密钥存储库的软件无法理解的专有格式存储。

由一个 IBM MQ 组件加密的密码不能由另一个 IBM MQ 组件使用。

加密密钥存储库密码时，可以提供唯一的加密密钥。 唯一的加密密钥会阻止无法访问加密密钥的任何人解密密码。

需要纯文本密钥存储库密码来管理密钥存储库中的证书。 除了使用 IBM MQ 密码保护系统对密钥存储库密码进行加密外，您还必须将密钥存储库密码存储在安全位置，在此位置可以访问该密钥存储库密码以实现此目的。

有关 IBM MQ 密码保护系统的更多信息，请参阅 第 494 页的『保护 IBM MQ 组件配置文件中的密码』。

密钥存储库隐藏文件。

runmqakm 命令可以将密钥存储库密码存储在隐藏文件中。

密码使用特定于 IBM MQ 的加密提供程序 IBM Global Security Kit (GSKit)的专有方法进行加密。

无法提供唯一的加密密钥。

加密密码存储在与密钥存储库文件相同的目录中的隐藏文件中。

对密钥存储库和隐藏文件都具有读访问权的任何人都可以访问和管理密钥存储库的内容。

注: V 9.4.0 Deprecated 从 IBM MQ 9.4.0 开始，不推荐将隐藏文件与 IBM MQ Java 应用程序配合使用。

要点: V 9.4.0 V 9.4.0 从 IBM MQ 9.4.0 开始，使用 TLS 的 AMQP 和 MQTT 通道不支持隐藏文件。

无论您选择何种方法对密钥存储库密码进行加密，请确保您了解对存储的密码进行加密的限制。 有关更多信息，请参阅第 500 页的『通过密码加密进行保护时存在的限制』。

相关概念

第 265 页的『为 AIX, Linux, and Windows 上的队列管理器提供密钥存储库密码』
由于密钥存储库包含敏感信息，因此使用密码对其进行保护。 为了能够访问密钥存储库内容以执行 TLS 操作， IBM MQ 必须能够检索密钥存储库密码。

第 267 页的『为 AIX, Linux, and Windows 上的 IBM MQ MQI client 提供密钥存储库密码』
由于密钥存储库包含敏感信息，因此使用密码对其进行保护。 为了能够访问密钥存储库内容以执行 TLS 操作， IBM MQ 必须能够检索密钥存储库密码。

第 260 页的『在 AIX, Linux, and Windows 上使用 SSL/TLS』
在 AIX, Linux, and Windows 系统上，传输层安全性 (TLS) 支持随 IBM MQ 一起安装。

Windows 在 *Windows* 上访问和保护密钥数据库文件

密钥数据库文件可能没有相应的访问许可权。 必须设置对这些文件的相应访问权。

设置对文件 *key*.p12， *key*.kdb， *key*.sth， *key*.crl 和 *key*.rdb 的访问控制，其中 *key* 是密钥数据库的主干名称，用于向一组受限用户授予权限。

如果您使用了除 .p12 或 .kdb 以外的其他密钥存储库扩展，那么还必须确保设置此文件的许可权。

请考虑按如下所示授予访问权:

完全权限
BUILTIN\Administrators， NT AUTHORITY\SYSTEM 以及创建数据库文件的用户。

读权限
对于队列管理器，仅适用于本地 mqm 组。 这假定 MCA 正在 mqm 组中的用户标识下运行。

对于客户机，这是用于运行客户机进程的用户标识。

Linux AIX 在 *AIX and Linux* 系统上访问和保护密钥数据库文件

密钥数据库文件可能没有相应的访问许可权。 必须设置对这些文件的相应访问权。

对于队列管理器，设置对密钥数据库文件的许可权，以便队列管理器和通道进程可以在必要时读取这些文件，但其他用户无法读取或修改这些文件。 通常， mqm 用户需要读许可权。 如果您已通过以 mqm 用户身份登录来创建密钥数据库文件，那么许可权可能已足够; 如果您不是 mqm 用户，而是 mqm 组中的另一个用户，那么可能需要将读许可权授予 mqm 组中的其他用户。

同样，对于客户机，设置对密钥数据库文件的许可权，以便客户机应用程序进程可以在必要时读取这些文件，但其他用户无法读取或修改这些文件。 通常，运行客户机进程的用户需要读许可权。 如果您已通过以该用户身份登录来创建密钥数据库文件，那么许可权可能已足够; 如果您不是客户机进程用户，而是该组中的另一个用户，那么您可能需要将读许可权授予该组中的其他用户。

设置对文件 *key*.p12， *key*.kdb， *key*.sth， *key*.crl 和 *key*.rdb 的许可权，其中 *key* 是密钥数据库的主干名称，针对文件所有者设置为 read 和 write，针对 mqm 或客户机用户组设置为 read (-rw-r -----)。

如果您使用了除 .p12 或 .kdb 以外的其他密钥存储库扩展，那么还必须确保设置此文件的许可权。

ALW 将缺省 *CA* 证书添加到 *AIX, Linux, and Windows* 上的空密钥存储库中

遵循此过程将一个或多个缺省认证中心 (CA) 证书添加到空密钥存储库。

创建新的密钥存储库时，该存储库为空。 您可以使用 **runmqakm** 命令将缺省 CA 证书添加到密钥存储库。

使用 runmqakm

发出以下命令以使用 **runmqakm** 命令将缺省 CA 证书添加到密钥存储库:

```
runmqakm -cert -populate -db filename -pw password
```

其中:

-db 文件名
指定密钥存储库的标准文件名。

-pw *password*
指定密钥存储库的密码。

注: IBM MQ 信任密钥存储库中 CA 证书签署的所有证书。 请仔细考虑要信任哪些认证中心，并仅添加认证客户机和队列管理器所需的 CA 证书。 建议不要将完整的缺省 CA 证书集添加到密钥存储库。

ALW 在 *AIX, Linux, and Windows* 上查找队列管理器的密钥存储库

使用此过程可获取队列管理器的密钥数据库文件的位置

过程

1. 使用以下任一 MQSC 命令显示队列管理器的属性:

```
DISPLAY QMGR ALL
DISPLAY QMGR SSLKEYR
```

   您还可以使用 IBM MQ Explorer 或 PCF 命令来显示队列管理器的属性。
2. 检查命令输出以获取密钥数据库文件的路径和主干名称。
   例如

   a. 在 AIX and Linux: `/var/mqm/qmgrs/QM1/ssl/key` 上，其中 `/var/mqm/qmgrs/QM1/ssl` 是路径， `key` 是主干名称

   b. 在 Windows 上: *MQ_INSTALLATION_PATH*`\qmgrs\QM1\ssl\key`，其中 *MQ_INSTALLATION_PATH*`\qmgrs\QM1\ssl` 是路径， `key` 是词干名称。 *MQ_INSTALLATION_PATH* 表示 IBM MQ 安装所在的高级目录。

   注: 从 IBM MQ 9.3.0 开始， SSLKEYR 字段支持完整文件名 (包括扩展名) 和主名称 (不含扩展名)。 如果设置了主干名称，那么 IBM MQ 会自动追加 `.kdb` 并使用该密钥存储库。

ALW 在 *AIX, Linux, and Windows* 上更改队列管理器的密钥存储库位置

您可以通过各种方法 (包括 MQSC 命令 ALTER QMGR) 来更改队列管理器密钥数据库文件的位置。

通过使用 MQSC 命令 ALTER QMGR 来设置队列管理器的密钥存储库属性，可以更改队列管理器的密钥数据库文件的位置。 例如，在 AIX and Linux 上:

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

在 Windows 上:

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

注意: 在 Windows 和 Linux 上，如果使用 TLS AMQP 通道，那么密钥存储库文件的后缀必须是下列其中一项:

- `.kdb`，用于 CMS 密钥存储库
- `.p12` 或 `.pkcs12`，用于 PKCS #12 密钥存储库。

您还可以使用 IBM MQ Explorer 或 PCF 命令来更改队列管理器的属性。

当您更改队列管理器的密钥数据库文件的位置时，不会从旧位置传输证书。 如果您现在正在访问的密钥数据库文件是新的密钥数据库文件，那么必须向该文件中填充所需的 CA 和个人证书，如 第 484 页的『将个人证书导入到 AIX, Linux, and Windows 上的密钥存储库中』中所述。

为 *AIX, Linux, and Windows* 上的队列管理器提供密钥存储库密码

由于密钥存储库包含敏感信息，因此使用密码对其进行保护。 为了能够访问密钥存储库内容以执行 TLS 操作， IBM MQ 必须能够检索密钥存储库密码。

IBM MQ 提供了两种机制来向队列管理器提供密钥存储库密码:

- 第 266 页的『KEYRPWD 属性』
- 第 266 页的『密钥存储库隐藏文件』

如果不使用密钥存储库隐藏文件，那么将使用 IBM MQ 密码保护系统对密钥存储库密码进行加密。 有关保护密钥存储库密码的方法的更多信息，请参阅 第 263 页的『对 AIX, Linux, and Windows 上的密钥存储库密码进行加密』。

KEYRPWD 属性

要直接向队列管理器提供密钥存储库密码，请运行以下 MQSC 命令，将 *password* 替换为密钥存储库密码:

```
ALTER QMGR KEYRPWD('password')
```

注意: 确保使用单引号将密码括起来，否则 IBM MQ 会将字符转换为大写。

使用此方法指定密钥存储库密码时，将先使用 IBM MQ 密码保护系统对密码进行加密，然后再进行存储。

加密密钥 (称为初始密钥) 用于加密密码。 将队列管理器设置为使用唯一的初始密钥来安全地保护密码。 如果未提供初始密钥，那么将使用缺省密钥。

在设置密钥存储库密码之前，请确保使用唯一的初始密钥配置队列管理器。 您可以使用 **ALTER QMGR** 命令上的 **INITKEY** 属性来修改初始密钥。 例如:

```
ALTER QMGR INITKEY('mykey')
```

警告: 在设置密钥存储库密码之后修改初始密钥不会导致使用新的初始密钥对密钥存储库密码进行加密。 在不重置密钥存储库密码的情况下更改初始密钥将导致 IBM MQ 无法解密密钥存储库密码，因此无法访问密钥存储库。

有关 **KEYRPWD** 属性的更多信息，请参阅 KEYRPWD。

密钥存储库隐藏文件

如果未使用 **KEYRPWD** 属性向队列管理器提供密钥存储库密码，那么 IBM MQ 将假定存储文件与密钥存储库存在于同一目录中。 隐藏文件具有与密钥存储库相同的主干名称，但具有 `.sth` 扩展名。

密钥存储库隐藏文件与密钥存储库或更高版本同时创建，作为单独的 **runmqakm** 命令。

注意: 隐藏文件的格式特定于 IBM MQ 加密提供程序 IBM Global Security Kit (GSKit)，并且在使用其他加密提供程序的平台上不可用。

要在创建密钥存储库时创建隐藏文件，请指定 **-stash** 参数。 例如:

```
runmqakm -keydb -create -db key.kdb -pw passw0rd -stash
```

其中 *passw0rd* 是密钥存储库密码。

要在以后创建隐藏文件，请运行以下命令:

```
runmqakm -keydb -stashpw -db key.kdb -pw passw0rd
```

其中 *passw0rd* 是密钥存储库密码。

相关概念

第 263 页的『对 AIX, Linux, and Windows 上的密钥存储库密码进行加密』
多个 IBM MQ 组件需要访问包含数字证书或对称密钥的密钥存储库。 密钥存储库使用密码进行保护，因为它包含敏感信息。 访问密钥存储库时，必须将密钥存储库密码存储在 IBM MQ 可以读取该密码的位置。 还必须对密码进行加密，以降低对密钥存储库进行未经授权的访问的可能性。

第 267 页的『为 AIX, Linux, and Windows 上的 IBM MQ MQI client 提供密钥存储库密码』
由于密钥存储库包含敏感信息，因此使用密码对其进行保护。 为了能够访问密钥存储库内容以执行 TLS 操作， IBM MQ 必须能够检索密钥存储库密码。

ALW 在 *AIX, Linux, and Windows* 上查找 *IBM MQ MQI client* 的密钥存储库

密钥存储库的位置由 MQSSLKEYR 变量提供，或者在 MQCONNX 调用中指定。

检查 MQSSLKEYR 环境变量以查找 IBM MQ MQI client 的密钥数据库文件的位置。 例如:

```
echo $MQSSLKEYR
```

还要检查应用程序，因为还可以在 MQCONNX 调用中设置密钥数据库文件名，如第 267 页的『为 AIX, Linux, and Windows 上的 IBM MQ MQI client 指定密钥存储库位置』中所述。 MQCONNX 调用中设置的值将覆盖 MQSSLKEYR 的值。

ALW 为 *AIX, Linux, and Windows* 上的 *IBM MQ MQI client* 指定密钥存储库位置

IBM MQ MQI client 没有缺省密钥存储库。 您可以通过两种方式之一指定其位置。 确保密钥数据库文件只能由预期用户或管理员访问，以防止未经授权的复制到其他系统。

您可以通过两种方式指定 IBM MQ MQI client 的密钥数据库文件的位置:

- 设置 MQSSLKEYR 环境变量。 例如，在 AIX and Linux 上:

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

  在 Windows 上:

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

- 在应用程序进行 MQCONNX 调用时，在 MQSCO 结构的 *KeyRepository* 字段中提供密钥数据库文件的路径和系统名称。 有关在 MQCONNX 中使用 MQSCO 结构的更多信息，请参阅 MQSCO 概述。

为 *AIX, Linux, and Windows* 上的 *IBM MQ MQI client* 提供密钥存储库密码

由于密钥存储库包含敏感信息，因此使用密码对其进行保护。 为了能够访问密钥存储库内容以执行 TLS 操作， IBM MQ 必须能够检索密钥存储库密码。

IBM MQ 提供了四种机制来向 IBM MQ MQI client 提供密钥存储库密码:

- 第 267 页的『MQSCO 的 KeyRepoPassword 字段 』
- 第 268 页的『MQKEYRPWD 环境变量』
- 第 268 页的『客户机配置文件的 SSLKeyRepositoryPassword 属性』
- 第 268 页的『密钥存储库隐藏文件』

如果不使用密钥存储库隐藏文件，那么可以将密钥存储库密码作为纯文本字符串或使用 IBM MQ 密码保护系统加密的字符串提供。 有关保护密钥存储库密码的方法的更多信息，请参阅第 263 页的『对 AIX, Linux, and Windows 上的密钥存储库密码进行加密』。

MQSCO 的 KeyRepoPassword 字段

要使用 MQSCO 结构提供密钥存储库密码，必须使用以下三个变量字符串字段的组合:

KeyRepoPasswordLength
密码的长度。

KeyRepoPasswordPtr
指向内存中包含密码的位置的指针。

KeyRepoPasswordOffset
密码在内存中的位置，表示为从 MQSCO 结构开始的字节数。

注: 只能提供 **KeyRepoPasswordPtr** 或 **KeyRepoPasswordOffset** 中的一个。

例如:

```
char * pwd = "passw0rd";
MQSCO  SslConnOptions = {MQSCO_DEFAULT};

SslConnOptions.KeyRepoPasswordPtr = pwd;
SslConnOptions.KeyRepoPasswordLength = (MQLONG)strlen(SslConnOptions.KeyRepoPasswordPtr);
SslConnOptions.Version = MQSCO_VERSION_6;
```

注意: 如果使用此方法提供密码，请在将密码提供给 IBM MQ client 应用程序之前对其进行加密。 有关更多信息，请参阅第 268 页的『加密密钥存储库密码』。

有关 MQCSO 结构的更多信息，请参阅 MQSCO-SSL/TLS 配置选项。

MQKEYRPWD 环境变量

如果未使用 MQSCO 结构向客户机提供密钥存储库密码，那么可以使用 *MQKEYRPWD* 环境变量来指定密钥存储库密码。 例如：

```
export MQKEYRPWD=passw0rd
```

或

```
set MQKEYRPWD=passw0rd
```

其中 passw0rd 是您的密码。

注意: 如果使用此方法提供密码，请在设置环境变量的值之前对密码进行加密。 有关更多信息，请参阅第 268 页的『加密密钥存储库密码』。

客户机配置文件的 SSLKeyRepositoryPassword 属性

如果未使用其他方法之一向客户机提供密钥存储库密码，那么可以使用客户机配置文件的 **SSL** 节中的 **SSLKeyRepositoryPassword** 属性来指定密钥存储库密码。 例如：

```
SSL:
    SSLKeyRepositoryPassword=passw0rd
```

注意: 如果使用此方法提供密码，请在设置 **SSLKeyRepositoryPassword** 属性的值之前对密码进行加密。 有关更多信息，请参阅第 268 页的『加密密钥存储库密码』。

有关客户机配置文件的 SSL 节的更多信息，请参阅 客户机配置文件的 SSL 节。

密钥存储库隐藏文件

如果未使用其他方法之一向客户机提供密钥存储库密码，那么 IBM MQ 将假定存储文件与密钥存储库存在于同一目录中。 隐藏文件具有与密钥存储库相同的主干名称，但具有 .sth 扩展名。

使用单独的 **runmqakm** 命令在密钥存储库或更高版本的同时创建密钥存储库隐藏文件。

注意: 隐藏文件的格式特定于 IBM MQ 加密提供程序 IBM Global Security Kit (GSKit)，并且在使用其他加密提供程序的平台上不可用。

要在创建密钥存储库时创建隐藏文件，请指定 **-stash** 参数。 例如：

```
runmqakm -keydb -create -db key.kdb -pw passw0rd -stash
```

其中 *passw0rd* 是密钥存储库密码。

要在以后创建隐藏文件，请运行以下命令：

```
runmqakm -keydb -stashpw -db key.kdb -pw passw0rd
```

其中 *passw0rd* 是密钥存储库密码。

加密密钥存储库密码

如果使用除隐藏文件以外的任何方法提供密钥存储库密码，请使用 IBM MQ 密码保护系统对密码进行加密。 要对密码进行加密，请运行 **runmqicred** 命令。 提示时输入密钥存储库密码。 该命令输出加密密码。 可以使用描述的任何方法将加密密码提供给 IBM MQ MQI client ，而不是纯文本密码。

加密密钥 (称为初始密钥) 用于加密密码。 加密密码时，请使用唯一的初始密钥来安全地保护密码。 要提供您自己的初始密钥，请对 **runmqicred** 命令使用 **-sf** 参数。 如果未提供初始密钥，那么将使用缺省密钥。

有关更多信息，请参阅 runmqicred (保护 IBM MQ 客户机密码)。

如果在加密密钥存储库密码时提供自己的初始密钥，并向 IBM MQ MQI client 提供加密密码，那么还必须确保向 IBM MQ MQI client 提供相同的初始密钥。 有关如何向 IBM MQ MQI client 提供初始密钥的更多信息，请参阅 第 269 页的『为 AIX, Linux, and Windows 上的 IBM MQ MQI client 提供初始密钥』。

相关概念

第 263 页的『对 AIX, Linux, and Windows 上的密钥存储库密码进行加密』
多个 IBM MQ 组件需要访问包含数字证书或对称密钥的密钥存储库。 密钥存储库使用密码进行保护，因为它包含敏感信息。 访问密钥存储库时，必须将密钥存储库密码存储在 IBM MQ 可以读取该密码的位置。 还必须对密码进行加密，以降低对密钥存储库进行未经授权的访问的可能性。

第 265 页的『为 AIX, Linux, and Windows 上的队列管理器提供密钥存储库密码』
由于密钥存储库包含敏感信息，因此使用密码对其进行保护。 为了能够访问密钥存储库内容以执行 TLS 操作， IBM MQ 必须能够检索密钥存储库密码。

ALW 为 *AIX, Linux, and Windows* 上的 *IBM MQ MQI client* 提供初始密钥

如果向已使用 IBM MQ 密码保护系统加密的 IBM MQ MQI client 提供变量，那么可能需要提供用于加密值的相应初始密钥。

如果在加密值时未指定初始密钥，那么不需要向 IBM MQ client 提供任何初始密钥值。 但是，如果您使用了唯一的初始密钥，那么可以使用以下方法向 IBM MQ client 提供初始密钥:

- 第 269 页的『使用 MQCSP 结构提供初始密钥』
- 第 269 页的『使用 MQS_MQI_KEYFILE 环境变量提供初始密钥』
- 第 270 页的『使用客户机配置文件提供初始密钥』

使用 MQCSP 结构提供初始密钥

要使用 MQCSP 结构提供初始键，必须使用以下三个变量字符串字段的组合:

InitialKeyLength
初始键的长度

InitialKeyPtr
指向内存中包含初始键的位置的指针

InitialKeyOffset
初始密钥在内存中的位置，表示为从 MQCSP 结构开始的字节数。

注: 只能提供 **InitialKeyPtr** 或 **InitialKeyOffset** 中的一个。

例如:

```
char * initialKey = "myInitialKey";
MQCSP  cspOptions = {MQCSP_DEFAULT};


cspOptions.InitialKeyPtr = initialKey;
cspOptions.InitialKeyLength = (MQLONG)strlen(cspOptions.InitialKeyPtr);
cspOptions.Version = MQCSP_VERSION_2;
```

使用 MQS_MQI_KEYFILE 环境变量提供初始密钥

如果未使用 MQCSP 结构向客户机提供初始密钥，那么 IBM MQ 将检查 *MQS_MQI_KEYFILE* 环境变量。 您应该将此环境变量设置为包含单行文本 (由您要使用的初始键组成) 的文件的位置。

例如，如果根目录中存在名为 `mykey.key` 的文件，并且该文件包含初始键，那么应按如下所示设置环境变量:

```
export MQS_MQI_KEYFILE=/mykey.key
```

或

```
set MQS_MQI_KEYFILE=C:\mykey.key
```

使用客户机配置文件提供初始密钥

如果未使用先前机制向客户机提供初始密钥，那么 IBM MQ 会检查 `mqclient.ini` 文件的 Security 节的 **MQIInitialKeyFile** 属性。 您应该将此属性设置为包含单行文本 (由您要使用的初始键组成) 的文件的位置。

例如，如果根目录中存在名为 `mykey.key` 的文件，并且该文件包含初始密钥，那么客户机配置文件应包含以下内容:

```
Security:
    MQIInitialKeyFile=/mykey.key
```

相关概念

第 267 页的『为 AIX, Linux, and Windows 上的 IBM MQ MQI client 提供密钥存储库密码』
由于密钥存储库包含敏感信息，因此使用密码对其进行保护。 为了能够访问密钥存储库内容以执行 TLS 操作， IBM MQ 必须能够检索密钥存储库密码。

第 245 页的『使用 SSL/TLS』
这些主题提供了有关执行与将 TLS 与 IBM MQ 配合使用相关的单个任务的指示信息。

ALW 当对证书或密钥存储库所作的更改在 *AIX, Linux, and Windows* 上生效时

当您更改密钥存储库中的证书或密钥存储库的位置时，这些更改将在取决于通道类型和通道运行方式的时间生效。

在以下情况下，对密钥存储库中的证书或对密钥存储库位置所作的更改将生效:

- 当新的出站单通道进程首次运行 TLS 通道时。
- 当新的入站 TCP/IP 单通道进程首先接收到启动 TLS 通道的请求时。
- 发出 MQSC 命令 **REFRESH SECURITY TYPE(SSL)** 以刷新 TLS 环境时。
- 对于客户机应用程序进程，当进程中的最后一个 TLS 连接关闭时。 下一个 TLS 连接将获取证书更改。
- 对于作为进程池进程 (amqrmppa) 的线程运行的通道，当进程池进程启动或重新启动并首先运行 TLS 通道时。 如果进程池进程已运行 TLS 通道，并且您希望更改立即生效，请运行 MQSC 命令 **REFRESH SECURITY TYPE(SSL)**。
- 对于作为通道启动程序线程运行的通道，当通道启动程序启动或重新启动时，首先运行 TLS 通道。 如果通道启动程序进程已运行 TLS 通道，并且您希望更改立即生效，请运行 MQSC 命令 **REFRESH SECURITY TYPE(SSL)**。
- 对于作为 TCP/IP 侦听器的线程运行的通道，当侦听器启动或重新启动时，首先接收到启动 TLS 通道的请求。 如果侦听器已运行 TLS 通道，并且您希望更改立即生效，请运行 MQSC 命令 **REFRESH SECURITY TYPE(SSL)**。

您还可以使用 IBM MQ Explorer 或 PCF 命令来刷新 IBM MQ TLS 环境。

要点: 对密钥库配置文件所作的更改，或者对 Advanced Message Security (AMS) MCA 拦截器或 AMS 客户机所使用的密钥库所作的更改，将在重新启动队列管理器或应用程序时生效。

ALW 在 *AIX, Linux, and Windows* 上配置加密硬件

您可以通过多种方式为队列管理器或客户机配置加密硬件。

您可以使用以下任一方法在 AIX, Linux, and Windows 上为队列管理器配置加密硬件:

- 将 **ALTER QMGR** MQSC 命令与 SSLCRYP 参数配合使用，如 ALTER QMGR 中所述。
- 使用 IBM MQ Explorer 在 AIX, Linux, and Windows 系统上配置加密硬件。 有关更多信息，请参阅联机帮助。

您可以使用下列其中一种方法为 AIX, Linux, and Windows 上的 IBM MQ 客户机配置加密硬件:

- 设置 **MQSSLCRYP** 环境变量。 **MQSSLCRYP** 的允许值与 **SSLCRYP** 参数的允许值相同，如 ALTER QMGR 中所述。 要设置此环境变量，请使用下列其中一个命令:

– Linux AIX 在 AIX and Linux 系统上：

```
export MQSSLCRYP=string
```

– Windows 在 Windows 系统上：

```
SET MQSSLCRYP=string
```

其中 *string* 表示要用于配置系统上存在的加密硬件的参数字符串。

如果使用 GSK_PKCS11 版本的 **SSLCRYP** 参数，那么 PKCS #11 令牌标签必须与用于配置硬件的标签匹配。

- 在 IBM MQ client 配置文件的 SSL 节中设置 **SSLCryptoHardware** 属性。 允许的值与 **SSLCRYP** 参数的值相同，如 **ALTER QMGR** 中所述。

  如果使用 GSK_PKCS11 版本的 **SSLCRYP** 参数，那么 PKCS #11 令牌标签必须与用于配置硬件的标签匹配。
- 在 MQCONNX 调用上设置 SSL 配置选项结构 MQSCO 的 **CryptoHardware** 字段。 有关更多信息，请参阅 MQSCO 概述。

注意: >通过 **MQSSLCRYP** 环境变量或 **SSLCryptoHardware** 属性为加密硬件提供配置时，应在存储之前保护密码。 有关更多信息，请参阅第 497 页的『使用加密硬件的 IBM MQ clients』。

如果已使用这些方法中的任何方法配置了使用 PKCS #11 接口的加密硬件，那么必须将用于通道的个人证书存储在已配置的加密令牌的密钥数据库文件中。 在第 492 页的『管理 PKCS #11 硬件上的证书』中对此进行了描述。

MQ Appliance 在 IBM MQ Appliance 上使用 SSL/TLS

IBM MQ Appliance 具有传输层安全性 (TLS) 支持。

IBM MQ Appliance 具有用于管理证书的不同命令。 有关证书管理的详细信息，请参阅 IBM MQ Appliance 文档 TLS 证书管理

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 272.

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 *z/OS 上 TLS 的其他用户标识需求*

此信息描述了用户标识在 z/OS 上设置和使用 TLS 所需的其他需求。

确保您在系统上具有所有相应的 "高影响" 或 "普及" (HIPER) 更新。

如果密钥存储库由 CHINIT 用户标识拥有，那么此用户标识需要对 IRR.DIGTCERT.LISTRING 概要文件，否则将更新访问权，并对 IRR.DIGTCERT.LIST 概要文件。 根据需要使用带有 ACCESS (UPDATE) 或 ACCESS (READ) 的 PERMIT 命令授予访问权。

确保您已设置以下先决条件:

- *ssidCHIN* 用户标识在 RACF 中定义正确，并且 *ssidCHIN* 用户标识具有对以下概要文件的相应访问权。
  - IRR.DIGTCERT.LIST
  - IRR.DIGTCERT.LISTRING

  这些变量在 RACF FACILITY 类中定义。
- *ssidCHIN* 用户标识是密钥环的所有者。
- 队列管理器的个人证书 (如果由 RACDCERT 命令创建) 是使用与 *ssidCHIN* 用户标识相同的证书类型用户标识创建的。
- 将重新启动通道启动程序，或者发出命令 **REFRESH SECURITY TYPE(SSL)** 以获取您对密钥环所作的任何更改。
- IBM MQ 通道启动程序过程可通过链接列表， LPA 或 STEPLIB DD 语句访问系统 SSL 运行时库 *pdsname*.SIEALNKE。 此库必须经过 APF 授权。
- 在其权限下运行通道启动程序的用户标识配置为使用 z/OS UNIX System Services (z/OS UNIX) ，如 z/OS UNIX System Services Planning 文档中所述。

  不希望通道启动程序使用访客/缺省 UID 和 OMVS 段来调用 z/OS UNIX 的用户只需要根据缺省段对新的 OMVS 段进行建模，因为通道启动程序不需要特殊许可权，并且不会以超级用户身份在 UNIX 中运行。

  请参阅 第 274 页的『Giving the channel initiator the correct access rights on z/OS』 中的 PERMIT 命令，以获取有关如何为通道启动程序提供正确访问权的一些示例。

z/OS *Setting the SSLTASKS parameter on z/OS*

Use the ALTER QMGR command to set the number of server subtasks for processing TLS calls

To use TLS channels, ensure that there are at least two server subtasks by setting the SSLTASKS parameter, using the ALTER QMGR command. For example:

```
ALTER QMGR SSLTASKS(5)
```

To avoid problems with storage allocation, do not set the SSLTASKS attribute to a value greater than eight in an environment where there is no Certificate Revocation List (CRL) checking.

If CRL checking is used, an SSLTASK is held by the channel concerned for the duration of that check. This could be for a significant elapsed time while the relevant LDAP server is contacted, because each SSLTASK is a z/OS task control block.

You must restart the channel initiator if you change the value of the SSLTASKS attribute.

z/OS *Setting up a key repository on z/OS*

Set up a key repository at both ends of the connection. Associate each key repository with its queue manager.

A TLS connection requires a *key repository* at each end of the connection. Each queue manager must have access to a key repository. Use the SSLKEYR parameter on the ALTER QMGR command to associate a key repository with a queue manager. See “SSL/TLS 密钥存储库” on page 22 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 “数字证书” on page 12.

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 234

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 272.
- *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 275. 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 “数字证书标签，了解需求” on page 23 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 276.

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 272.
- *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 272.
- *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 277. 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 “通道认证记录” on page 43.

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 276.
- 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 278.

z/OS *Setting up a certificate name filter on z/OS*

Use the RACDCERT command to define a certificate name filter (CNF), which maps a Distinguished Name to a user ID.

Perform the following steps to set up a CNF.

1. Enable CNF functions using the following command. You require update authority on the class DIGTNMAP to do this.

   ```
   SETROPTS CLASSACT(DIGTNMAP) RACLIST(DIGTNMAP)
   ```

2. Define the CNF. For example:

   ```
   RACDCERT ID(USER1) MAP WITHLABEL('filter1') TRUST
   SDNFILTER('O=IBM.C=UK') IDNFILTER('O=ExampleCA.L=Internet')
   ```

   where USER1 is the user ID to be used when:

   - The DN of the subject has an Organization of `IBM` and a Country of UK.
   - The DN of the issuer has an Organization of `ExampleCA` and a Locality of `Internet`.

3. Refresh the CNF mappings:

```
SETROPTS RACLIST(DIGTNMAP) REFRESH
```

Note:

1. If the actual certificate is stored in the RACF database, the user ID under which it is installed is used in preference to the user ID associated with any CNF. If the certificate is not stored in the RACF database, the user ID associated with the most specific matching CNF is used. Matches of the subject DN are considered more specific than matches of the issuer DN.
2. Changes to CNFs do not apply until you refresh the CNF mappings.
3. A DN matches the DN filter in a CNF only if the DN filter is identical to the *least significant portion* of the DN. The least significant portion of the DN comprises the attributes that are usually listed at the right-most end of the DN, but which appear at the beginning of the certificate.

   For example, consider the SDNFILTER `'O=IBM.C=UK'`. A subject DN of `'CN=QM1.O=IBM.C=UK'` matches that filter, but a subject DN of `'CN=QM1.O=IBM.L=Hursley.C=UK'` does not match that filter.

   The least significant portion of some certificates can contain fields that do not match the DN filter. Consider excluding these certificates by specifying a DN pattern in the SSLPEER pattern on the DEFINE CHANNEL command.
4. If the most specific matching CNF is defined to RACF as NOTRUST, the entity uses the user ID under which the channel initiator is running.
5. RACF uses the `'.'` character as a separator. IBM MQ uses either a comma or a semicolon.

You can define CNFs to ensure that the entity never sets the channel user ID to the default, which is the user ID under which the channel initiator is running. For each CA certificate in the key ring associated with the entity, define a CNF with an IDNFILTER that exactly matches the subject DN of that CA certificate. This ensures that all certificates that the entity might use match at least one of these CNFs. This is because all such certificates must either be connected to the key ring associated with the entity, or must be issued by a CA for which a certificate is connected to the key ring associated with the entity.

Refer to the *z/OS Security Server RACF Security Administrator's Guide* for more information about the commands you use to manipulate CNFs.

z/OS *Defining a sender channel and transmission queue on QMA on z/OS*

Use the **DEFINE CHANNEL** and **DEFINE QLOCAL** commands to set up the required objects.

Procedure

On QMA, issue commands like the following example:

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QMB.MACH.COM) XMITQ(QMB)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256) DESCR('Sender channel using TLS from QMA to QMB')

DEFINE QLOCAL(QMB) USAGE(XMITQ)
```

Results

A sender channel, TO.QMB, and a transmission queue, QMB, are created.

z/OS *Defining a receiver channel on QMB on z/OS*

Use the **DEFINE CHANNEL** command to set up the required object.

Procedure

On QMB, issue a command like the following example:

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using TLS to QMB')
```

Results

A receiver channel, TO.QMB, is created.

z/OS *Starting the sender channel on QMA on z/OS*

If necessary, start a listener program and refresh security. Then start the channel using the **START CHANNEL** command.

Procedure

1. Optional: If you have not already done so, start a listener program on QMB.

   The listener program listens for incoming network requests and starts the receiver channel when it is needed. For information about how to start a listener, see Starting a channel listener.
2. Optional: If any SSL/TLS channels have run previously, issue the command `REFRESH SECURITY TYPE(SSL)`.

   This ensures that all the changes made to the key repository are available.
3. Start the channel on QMA, using the command `START CHANNEL(TO.QMB)`.

Results

The sender channel is started.

z/OS *Exchanging self-signed certificates on z/OS*

Exchange the certificates you previously extracted. If you use FTP, use the correct format.

Procedure

Transfer the CA part of the QM1 certificate to the QM2 system and vice versa, for example, by FTP.

If you transfer the certificates using FTP, you must do so in the correct format.

Transfer the following certificate types in *binary* format:

- DER encoded binary X.509
- PKCS #7 (CA certificates)
- PKCS #12 (personal certificates)

Transfer the following certificate types in ASCII format:

- PEM (privacy-enhanced mail)
- Base64 encoded X.509

z/OS *Defining a sender channel and transmission queue on QM1 on z/OS*

Use the **DEFINE CHANNEL** and **DEFINE QLOCAL** commands to set up the required objects.

Procedure

On QM1, issue commands like the following example:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QM1.MACH.COM) XMITQ(QM2)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA) DESCR('Sender channel using TLS from QM1 to QM2')

DEFINE QLOCAL(QM2) USAGE(XMITQ)
```

The CipherSpecs at each end of the channel must be the same.

Only the SSLCIPH parameter is mandatory if you want your channel to use TLS. See “IBM MQ 中的 CipherSpecs 和 CipherSuites” on page 35 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 280, 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.

识别和认证用户

您可以使用 X.509 证书， MQCSP 结构或多种类型的用户出口程序来标识和认证用户。

使用 X.509 证书

您可以使用带有 **SET CHLAUTH** 命令和 **SSLPEER** 参数的 X.509 证书来识别和认证用户。 **SSLPEER** 参数指定用于与来自通道另一端的对等队列管理器或客户机的证书的主题专有名称进行比较的过滤器。

有关使用 **SET CHLAUTH** 命令和 **SSLPEER** 参数的更多信息，请参阅 SET CHLAUTH。

认证中心可以撤销数字证书。 您可以根据平台，使用 OCSP 或 LDAP 服务器上的 CRL 来检查证书的撤销状态。 有关更多信息，请参阅第 299 页的『使用已撤销证书』。

使用 MQCSP 结构

MQCSP 连接安全参数结构是在 MQCONNX 调用上指定的。 此结构可以包含应用程序提供的凭证。 应用程序可以在 MQCSP 结构中提供用户标识和密码。 从 IBM MQ 9.3.4 开始，应用程序还可以提供认证令牌。 如果需要，可以在安全出口中更改 MQCSP。

警告: MQCSP 结构中的凭证有时以纯文本形式跨网络发送。 要确保客户机应用程序凭证受保护，请参阅第 27 页的『MQCSP 密码保护』。

有关更多信息，请参阅第 284 页的『使用 MQCSP 结构识别和认证用户』和第 287 页的『使用认证令牌』。

Linux AIX 在 AIX 和 Linux 上，可以使用操作系统或可插拔认证方法 (PAM) 来认证 MQCSP 结构中指定的用户标识和密码。 PAM 提供了用户认证的常规机制，用于隐藏服务中的详细信息。 有关更多信息，请参阅第 308 页的『使用可插拔认证方法 (PAM)』。

在出口中实现标识和认证

您可以使用多种类型的用户出口程序来识别和认证用户。 有关更多信息，请参阅第 285 页的『在安全出口中实现标识和认证』、第 286 页的『消息出口中的身份映射』 和 第 286 页的『API 出口和 API 交叉出口中的身份映射』。

特权用户

特权用户是对 IBM MQ 具有完全管理权限的用户。

除了下表中列出的用户外，还有某些对象和授权在授予访问权时必须格外小心，以确保队列管理器的完整性和安全性。 在授予以下任何授权时，必须应用额外的检查:

- 对 SYSTEM 对象的任何权限
- 用于创建，变更和删除对象的管理权限。

z/OS 在 z/OS 上，此授权是命令安全性和命令资源安全性权限，用于发出 DEFINE， ALTER 和 DELETE 命令。

Multi 在所有其他平台上，这些权限是管理权限，例如 `+crt`， `+chg` 和 `+dlt`。

- 用于清除队列的管理权限。

  z/OS 在 z/OS 上，此授权是命令安全性和命令资源安全性权限，用于发出 CLEAR 命令。

  Multi 在所有其他平台上，此授权为 `+clr`。
- 用于停止通道，回退或落实消息的管理权限。

  z/OS 在 z/OS 上，此授权是命令安全性和命令资源安全性权限，用于发出诸如 RESET CHANNEL， START CHANNEL 和 STOP CHANNEL 之类的命令。

  Multi 在所有其他平台上，这些权限为 `+ctrl` 和 `+ctrlx`。
- 备用用户 MQI 授权，允许应用程序升级特权以进行授权检查。

  z/OS 在 z/OS 上，此授权是授予备用用户安全概要文件的任何权限。

  Multi 在所有其他平台上，此授权为 `+altusr`。
- 允许应用程序更改消息的安全上下文的上下文授权。

  z/OS 在 z/OS 上，此授权是授予上下文安全概要文件的任何权限。

  Multi 在所有其他平台上，这些权限为 `+setall` 和 `+setid`。

作为一般主体，应该仅向消息传递应用程序授予对所需队列或主题的基本 MQI 权限。 在非特权 MCAUSER 和某些其他特殊类型的应用程序 (例如，死信队列处理程序) 下执行的 MCA 通道可能需要通常未授予应用程序的额外权限才能正常运行。

表 67: 特权用户 (按平台排列)

平台	特权用户
Windows 系统	• SYSTEM • mqm 组的成员 • Administrators 组的成员
AIX and Linux 系统	• mqm 组的成员
IBM i 系统	• 概要文件 qmqm 和 qmqmadm • qmqmadm 组的所有成员 • 使用 *ALLOBJ 设置定义的任何用户
z/OS	用于运行通道启动程序，队列管理器和高级消息安全地址空间的用户标识。 这些用户标识不会自动具有 IBM MQ 的完整管理权限，但由于通常授予这些用户标识的访问级别而被视为特权。

使用 MQCSP 结构识别和认证用户

您可以指定 MQCONNX 调用上的 MQCSP 连接安全参数结构。 MQCSP 结构是使用消息队列接口 (MQI) 来控制用于认证的凭证的应用程序的主要方法。

MQCSP 结构包含授权服务可用于标识和认证用户的凭证。

客户机或服务器端安全出口可以修改 MQCSP 结构，即使应用程序未显式提供 MQCSP 结构也是如此。 未显式提供 MQCSP 结构的应用程序的示例是使用 IBM MQ classes for JMS 的应用程序。 有关在 MQCSP 结构中插入用户标识和密码的客户机端安全出口的示例，请参阅 第 68 页的『用于插入用户标识和密码的客户机端安全出口 (mqccred)』。

V 9.4.0 MQCSP 结构包含用户标识和密码或认证令牌。 以下限制适用于 MQCSP 结构中提供的凭证:

- 应用程序或出口必须提供用户标识和密码，或者提供认证令牌，但不能同时提供两者。
- 只能使用满足特定格式和需求的认证令牌来访问 IBM MQ。 有关 IBM MQ 中认证令牌的需求的更多信息，请参阅 第 290 页的『认证令牌的需求』。
- 如果要采用认证令牌中的身份作为应用程序的上下文，那么令牌必须提供合适的用户声明，并且声明值必须是有效的 IBM MQ 用户标识。 例如，用户名必须符合最大长度和特殊字符限制。 有关采用用户标识的更多信息，请参阅 第 285 页的『MQCSP 与 ADOPTCTX 设置之间的关系』。

有关 MQCSP 结构的更多信息，请参阅 MQCSP-安全性参数。

警告: 客户机应用程序的 MQCSP 结构中的凭证有时以纯文本形式通过网络发送。 要确保客户机应用程序凭证受保护，请参阅 第 27 页的『MQCSP 密码保护』。

MQCSP 与 ADOPTCTX 设置之间的关系

如果启用了连接认证功能，那么 IBM MQ 始终会认证在 MQCSP 结构中传递的凭证。 成功认证凭证后， IBM MQ 可以采用用户标识对已连接的应用程序执行的操作进行后续授权检查。 如果使用 **ADOPTCTX(YES)** 定义了队列管理器的 **CONNAUTH** 属性所引用的认证信息 (AUTHINFO) 对象，那么将采用 MQCSP 凭证中的用户标识。

IBM MQ 对可用于授权检查的用户标识的长度有限制。 有关这些限制的更多信息，请参阅 第 75 页的『用户标识』。 采用 MQCSP 结构中传递的用户标识时， IBM MQ 的行为有所不同，具体取决于其他配置选项:

- 使用 LDAP 连接认证时， IBM MQ 将采用用户的 LDAP 记录的短用户名属性中的用户标识。 短用户名属性是使用 AUTHINFO 对象的 **SHORTUSR** 属性设置的。

  例如，如果 **SHORTUSR** 设置为 `'CN'`，并且 LDAP 记录将用户列示为 `'CN=Test,SN=MQ,O=IBM,C=UK'`，那么将使用用户标识 `Test`。
- 使用操作系统连接认证或 PAM 认证时，如果 ADOPTCTX 为 YES，那么将截断 MQCSP 结构中传递的用户标识，以便在采用作为连接上下文时满足 12 个字符的用户标识限制 IBM MQ。

  如果启用了 **ChlAuthEarlyAdopt**，那么在认证用户凭证后将发生截断。

  如果未启用 **ChlAuthEarlyAdopt**，那么会在采用之前进行截断。 在 Windows 上，如果以 `user@domain` 格式提供用户，那么这意味着截断可能导致域规范在用户少于 12 个字符时无效。

  例如，如果通过 MQCSP 提供了用户 `` `ibmmq@windowsdomain` ``，那么在此场景中将其截断为 `` `ibmmq@window` ``。 这将导致以下错误:

  ```
  AMQ8074W: 授权失败，因为 SID "SID" 与实体 "ibmmq@window" 不匹配
  ```

  在此基础上，如果通过 MQCSP 传递长度超过 12 个字符的用户标识 (例如，格式为 `user@domain` 的 Windows 域用户标识)，那么应该在 qm.ini 文件中配置 **ChlAuthEarlyAdopt**=*Y* 以避免此错误。

  或者，在 CONNAUTH AUTHINFO 配置上使用 ADOPTCTX (NO)，并使用替代方法 (例如 CHLAUTH USERMAP 规则，安全出口或通道对象 MCAUSER 设置) 来设置通道的用户标识。

在安全出口中实现标识和认证

您可以使用安全出口来实施单向或相互认证。

安全出口的主要用途是在通道的每个端启用 MCA 以认证其合作伙伴。 在消息通道的每一端以及 MQI 通道的服务器端， MCA 通常代表它所连接的队列管理器。 在 MQI 通道的客户机端， MCA 通常代表 IBM MQ MQI client 应用程序的用户执行操作。 在这种情况下，实际上在两个队列管理器之间或者在队列管理器与 IBM MQ MQI client 应用程序的用户之间进行相互认证。

提供的安全出口 (SSPI 通道出口) 说明了如何通过交换由可信认证服务器 (例如 Kerberos) 生成并检查的认证令牌来实现相互认证。 有关更多详细信息，请参阅 第 132 页的『Windows 上的 SSPI 通道出口程序』。

还可以使用公用密钥基础结构 (PKI) 技术来实现相互认证。 每个安全出口都会生成一些随机数据，使用其表示的队列管理器或用户的专用密钥对其进行签名，并在安全消息中将签名数据发送给其合作伙伴。 伙伴安全出口通过使用队列管理器或用户的公用密钥检查数字签名来执行认证。 在交换数字签名之前，如果有多个算法可供使用，那么安全出口可能需要同意用于生成消息摘要的算法。

当安全出口将已签名的数据发送给其合作伙伴时，它还需要发送一些方法来标识它所代表的队列管理器或用户。 这可能是专有名称，甚至是数字证书。 如果发送数字证书，那么合作伙伴安全出口可以通过证书链到根 CA 证书来验证证书。 这将保证用于检查数字签名的公用密钥的所有权。

仅当合作伙伴安全出口有权访问包含证书链中剩余证书的密钥存储库时，它才能验证数字证书。 如果未发送队列管理器或用户的数字证书，那么一个证书必须在合作伙伴安全出口有权访问的密钥存储库中可用。 伙伴安全出口无法检查数字签名，除非它可以找到签署者的公用密钥。

传输层安全性 (TLS) 使用类似刚才描述的 PKI 技术。 有关安全套接字层如何执行认证的更多信息，请参阅 第 16 页的『传输层安全性 (TLS) 概念』。

如果可信认证服务器或 PKI 支持不可用，那么可以使用其他方法。 一种常见的技术，可以在安全出口中实现，它使用对称密钥算法。

其中一个安全出口，出口 A，生成一个随机数，并将其在安全消息中发送到其合作伙伴安全出口，出口 B。 出口 B 使用其仅对两个安全出口已知的密钥副本对数字进行加密。 出口 B 使用出口 B 生成的第二个随机数将加密号码发送到安全消息中的出口 A。 出口 A 验证第一个随机数是否已正确加密，使用其密钥副本对第二个随机数进行加密，并将加密后的数字发送到安全消息中的出口 B。 然后，出口 B 验证第二个随机数是否已正确加密。 在此交换期间，如果任一安全出口对其他安全出口的真实性不满意，那么可以指示 MCA 关闭通道。

此技术的优点是在交换期间不会通过通信连接发送密钥或密码。 一个缺点是它没有为如何以安全的方式分发共享密钥的问题提供解决方案。 第 410 页的『在用户出口程序中实现机密性』中描述了此问题的一个解决方案。 在 SNA 中使用类似的方法在两个 LU 绑定以形成会话时进行相互认证。 此技术在 第 102 页的『会话级别认证』中进行了描述。

所有上述用于相互认证的技术都可以进行调整以提供单向认证。

消息出口中的身份映射

您可以使用消息出口来处理信息以认证用户标识，但最好在应用程序级别实现认证。

当应用程序将消息放入队列时，消息描述符中的 *UserIdentifier* 字段包含与应用程序关联的用户标识。 但是，不存在可用于认证用户标识的数据。 此数据可由通道发送端的消息出口添加，并由通道接收端的消息出口检查。 例如，认证数据可以是加密密码或数字签名。

如果在应用程序级别实施此服务，那么此服务可能更有效。 基本要求是接收消息的应用程序的用户能够识别和认证发送消息的应用程序的用户。 因此，自然会考虑在应用程序级别实现此服务。 有关更多信息，请参阅 第 286 页的『API 出口和 API 交叉出口中的身份映射』。

API 出口和 API 交叉出口中的身份映射

接收消息的应用程序必须能够识别并认证发送消息的应用程序的用户。 此服务通常最好在应用程序级别实施。 API 出口可以通过多种方式实现服务。

在单个消息级别，标识和认证是涉及消息的两个用户 (发送方和接收方) 的服务。 基本要求是接收消息的应用程序的用户能够识别和认证发送消息的应用程序的用户。 请注意，要求的是单向认证，而不是双向认证。

根据实施方式，用户及其应用程序可能需要与服务进行交互甚至交互。 此外，使用服务的时间和方式可能取决于用户及其应用程序所在的位置以及应用程序本身的性质。 因此，自然会考虑在应用程序级别而不是在链接级别实现服务。

如果您考虑在链接级别实现此服务，那么可能需要解决以下问题:

- 在消息通道上，如何将服务仅应用于需要该服务的消息?
- 如果需要，如何使用户及其应用程序能够与服务进行交互或交互?
- 在多跳跃情境中，消息在发送到其目标的途中通过多个消息通道发送，您在何处调用服务的组件?

以下是如何在应用程序级别实现标识和认证服务的一些示例。 术语 *API* 出口 表示 API 出口或 API 交叉出口。

- 当应用程序将消息放入队列时， API 出口可以从可信认证服务器 (例如 Kerberos) 获取认证令牌。 API 出口可以将此令牌添加到消息中的应用程序数据。 当接收应用程序检索消息时，第二个 API 出口可以通过检查令牌来请求认证服务器认证发送方。
- 当应用程序将消息放入队列时， API 出口可以将以下项附加到消息中的应用程序数据:
  - 发送方的数字证书
  - 发送者的数字签名

  如果可用于生成消息摘要的不同算法，那么 API 出口可以包含其已使用的算法的名称。

  当接收应用程序检索消息时，第二个 API 出口可以执行以下检查:

  - API 出口可以通过通过证书链到根 CA 证书来验证数字证书。 为此， API 出口必须有权访问包含证书链中剩余证书的密钥存储库。 此检查可保证由专有名称标识的发件人是证书中包含的公用密钥的真正所有者。
  - API 出口可以使用证书中包含的公用密钥来检查数字签名。 此检查将对发送方进行认证。

  可以发送发件人的专有名称，而不是整个数字证书。 在这种情况下，密钥存储库必须包含发送方的证书，以便第二个 API 出口可以找到发送方的公用密钥。 另一种可能是发送证书链中的所有证书。
- 当应用程序将消息放入队列时，消息描述符中的 *UserIdentifier* 字段包含与应用程序关联的用户标识。 用户标识可用于标识发送方。 要启用认证， API 出口可以将一些数据 (例如加密密码) 附加到消息中的应用程序数据。 当接收应用程序检索消息时，第二个 API 出口可以使用随消息一起传递的数据来认证用户标识。

  对于在受控可信环境中生成的消息，以及在可信认证服务器或 PKI 支持不可用的情况下，可以认为此技术已足够。

Linux V 9.4.0 AIX 使用认证令牌

从 IBM MQ 9.4.0 开始，客户机应用程序可以提供令牌以向在 AIX 或 Linux 上运行的队列管理器进行认证。 令牌中的用户标识还可用于授权访问 IBM MQ 资源。

JWT (JSON Web 令牌) 采用基于声明的身份模型。 将身份和访问控制抽象为声明和令牌发布者的构想。

- 声明是一个名称值对，其中包含有关用户的信息，并确定用户是谁，而不是他们可以执行的操作。
- 令牌发布者是可信的第三方或仅根据用户身份为用户发放令牌的服务器。 令牌发布者不关心用户可以执行的操作。

令牌是包含声明的简单结构，可轻松通过因特网在各方之间传输。 使用令牌进行认证具有集中身份管理的优点。 您可以使用一个可信令牌签发者，以便应用程序可以向许多服务进行认证，而无需单独向每个服务注册。 令牌提供了更高的安全性，因为不会将凭证发送到每个服务 (仅发送到可信签发者)。

JWT 是通过建议的因特网标准 RFC7519 定义的。

令牌如何使用 IBM MQ

与 IBM MQ 配合使用的令牌必须是已使用 IBM MQ 支持的算法进行签名的有效 JWT。 JWT 必须根据 JSON Web 签名 (JWS) 标准进行签名。 使用 JSON Web 加密 (JWE) 和 JSON Web 密钥 (JWK) JOSE 技术的令牌不能与 IBM MQ 配合使用。 有关更多信息，请参阅第 290 页的『认证令牌的需求』。

提供认证令牌的应用程序可以在支持 IBM MQ clients 的任何平台上运行。 应用程序必须以 C 或 Java 编写，并使用客户机绑定连接到队列管理器。 但是，队列管理器必须在 AIX 或 Linux 上运行。

队列管理器针对密钥存储库中的可信签发者公用密钥或对称密钥验证令牌签名。 要设置队列管理器，请遵循第 292 页的『配置队列管理器以使用 JWKS 端点接受认证令牌』 或 配置队列管理器以接受使用本地密钥库的认证令牌中的步骤。

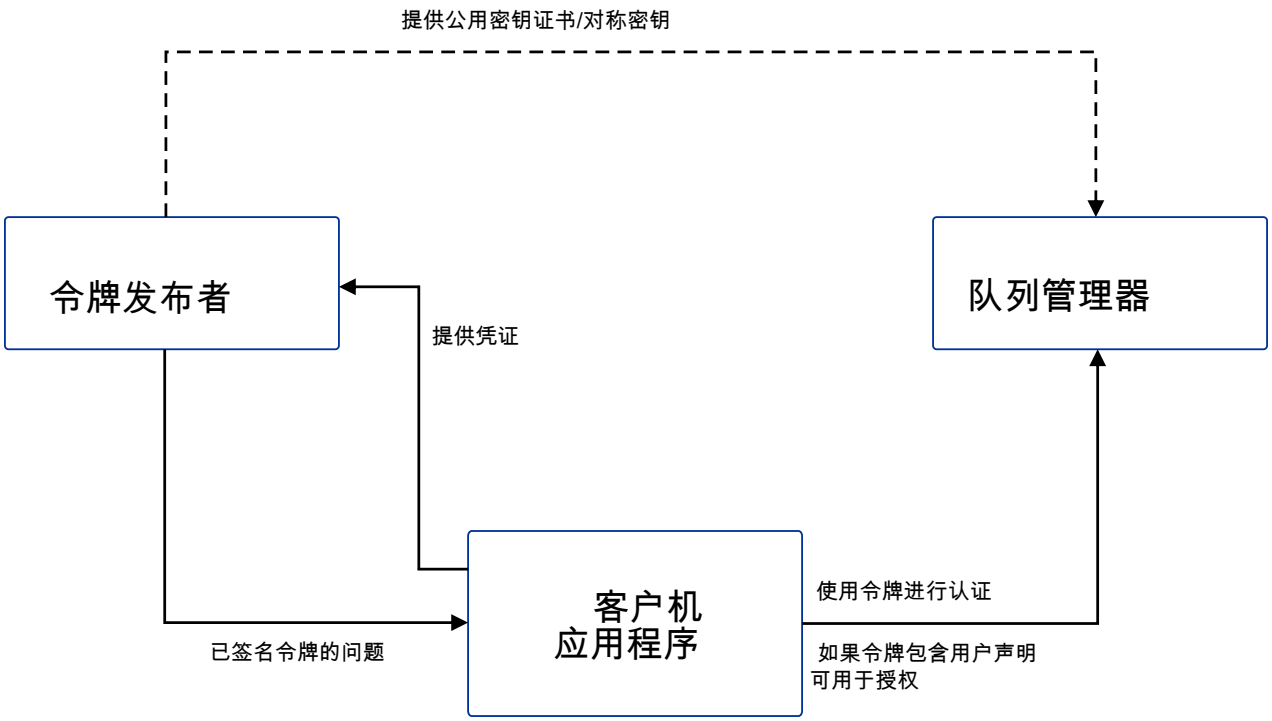
令牌签发者是具有授权安全访问权的可信方，这意味着他们验证应用程序用户的身份。 队列管理器将检查认证令牌是否有效以及已认证的用户是否有权访问 IBM MQ 对象。 队列管理器可以，但不需要先了解用户，然后再使用令牌进行连接。 IBM MQ 管理员必须为连接到队列管理器的应用程序设置认证和授权，并设置令牌必须包含的内容的需求。

客户机应用程序可以从签发者动态请求令牌，当它连接到 IBM MQ 时，会使用该令牌进行认证。 然后，应用程序使用 MQCSP 结构或所选 API 中的等效项 在连接时将令牌传递到队列管理器。

如果无法将应用程序更改为请求认证令牌，并在连接时将该令牌提供给队列管理器，那么可以使用安全出口来提供 MQCSP 结构中的令牌。

如果令牌满足认证令牌的要求，并且令牌签名有效，那么将建立连接。 如果令牌中包含可选用户声明，那么队列管理器还可以使用令牌中包含的用户标识进行授权检查以访问 IBM MQ 资源。 用户声明是令牌中包含队列管理器用于授权检查的用户标识的声明。 此用户声明名称是通过 `qm.ini` 文件的 **AuthToken** 节中的 **UserClaim** 属性指定的。

有关更多信息，请参阅 第 296 页的『在应用程序中使用认证令牌』 和 MQCSP-Security 参数。



该图显示了将令牌与 IBM MQ 配合使用的预期流程的基本示例。 期望的生命周期如下所示:

- 令牌由可信签发者发放到应用程序。 有关更多信息，请参阅 认证令牌需求。

- 应用程序在连接时将令牌传递到队列管理器。 有关更多信息，请参阅 在应用程序中使用认证令牌。
- 队列管理器针对密钥存储库中的可信签发者公用密钥或对称密钥验证令牌签名。 要设置队列管理器，请遵循 第 292 页的『配置队列管理器以使用 JWKS 端点接受认证令牌』中的步骤。
- 如果认证令牌包含有效的用户声明，那么可以采用令牌中的用户进行授权检查以访问 IBM MQ 资源。 有关更多信息，请参阅 采用用户进行授权。
- IBM MQ 管理员管理可信令牌签发者证书。 当证书到期时，必须从令牌发布者获取新证书并将其添加到密钥存储库。
- 如果您配置了队列管理器，并且应用程序正在连接但迂到令牌问题，请参阅 对认证令牌问题进行故障诊断 和 令牌认证错误代码。

IBM MQ 适用于提供符合 JWT 和 JWS 标准的令牌的任何令牌发布者。

如果您尚未使用令牌，但想要了解建立令牌服务器所涉及的内容，请参阅免费和开放式源代码 Keycloak 项目的 入门指南 。

相关参考

qm.ini 文件的 AuthToken 节

Linux V 9.4.0 AIX

认证令牌的需求

用于 IBM MQ 的认证令牌的验证需求，结构和算法。

需求

与 IBM MQ 配合使用的认证令牌必须满足以下需求。

- 令牌长度不得超过最大长度 8192 个字符。 有关更多信息，请参阅 TokenLength (MQLONG) for MQCSP。
- 令牌结构和编码有效，如 RFC7519 中的 JSON Web 令牌 (JWT) 规范和 RFC7515 中的 JSON Web 签名 (JWS) 规范所定义。
- 存在 第 291 页的表 68 中指定的必需令牌头参数，并且这些参数的值有效。
- 第 291 页的表 69 中指定的必需有效内容声明存在，并且这些声明的值有效。
- 该令牌使用 IBM MQ 支持的 第 291 页的表 70 中的算法进行签名。
- 到期 (**exp**) 声明的值晚于当前时间。
- 如果存在 not before (**nbf**) 声明，那么该值在当前时间之前。
- 如果存在用户声明，那么该值必须满足 第 292 页的『认证令牌中的用户标识』的要求。

令牌结构

IBM MQ 接受符合 RFC7519 标准的 JWT。 必须根据 RFC7515 中定义的 JWS 标准对 JWT 进行签名和编码。

IBM MQ 期望 JWS 安全令牌包含以下三个组件:

JOSE 头

一个 JSON 对象，其中包含用于描述令牌类型以及用于保护其内容的密码算法的参数。

以下头示例声明已编码的对象是 JWT，并且头和有效内容是使用 HMAC SHA-256 算法保护的。

```
{
 "typ":"JWT",
 "alg":"HS256"
}
```

JWS 有效内容

包含 JWT 标准中指定的声明的 JSON 对象。 JSON 对象的每个成员都是一个声明。 声明可以声明令牌签发者的身份或不记名的用户标识。

```
{
 "exp": 1685529153,
 "nbf": 1685528150,
 "AppUser": "MyUserName"
 }
```

JWS 签名

用于验证令牌是否由可信签发者发布。

这些组件在 JWS 安全令牌中表示为以句点 (".") 分隔的 base64url-encoded 字符串。

将对符合 JWS 标准的认证令牌进行签名，以允许对该令牌的真实性进行验证，但不会对其进行加密。 因此，具有令牌访问权的任何人都可以读取并可能复用该令牌。 配置与队列管理器的连接，以确保在通过网络 (例如，使用 TLS) 发送认证时使用加密来保护认证。 有关用于保护应用程序提供的凭证的选项的更多信息，请参阅 MQCSP 密码保护。

IBM MQ 支持认证令牌的头和有效内容中的以下参数和声明。 将忽略令牌中的任何其他参数或声明。 如果令牌包含多个同名参数或声明，那么将使用具有重复名称的最后一个参数或声明。

表 *68:* 令牌头参数描述

令牌部件	参数名	数据类型	必需	描述
头	**typ**	字符串	Yes	令牌类型。 此参数的值必须为 "JWT"。
	alg	字符串	Yes	用于保护头和有效内容的算法。 此参数的值必须是 第 291 页的表 70 中的其中一个算法。

表 *69:* 令牌有效内容声明描述

令牌部件	参数名	数据类型	必需	描述
有效内容	**exp**	整数	Yes	令牌到期时间，以自全球标准时间 1 1979 年 1 月 00:00 以来的秒数表示。 在此时间之后，将不接受该令牌。
	nbf	整数	否	时间，以自 1979 年 1 月 1 以来的秒数表示， 00:00 全球标准时间，在此时间之前不接受令牌。
	用户声明名称在 `qm.ini` 文件的 **AuthToken** 节的 **UserClaim** 字段中指定。	字符串	仅当令牌中的用户声明用于授权时才需要。	包含用于授权检查的用户标识的声明的名称。 例如，如果令牌具有用户声明 `"AppUser": "MyUserName"`，那么必须在 `qm.ini` 文件的 **AuthToken** 节中指定 **UserClaim**=AppUser 。

有关已编码和解码的令牌的良好示例，请参阅 `jwt.io` Web 站点上的 调试器 页面。

个算法

IBM MQ 支持 JWS 安全令牌的 JSON Web 算法 (JWA) 规范 中包含的算法子集。

表 *70: IBM MQ* 针对 *JWS* 安全令牌支持的 *JSON Web* 算法 *(JWA)*

alg 参数值	**数字签名或 MAC 算法**
HS256	使用 SHA-256 的 HMAC

表 70: IBM MQ 针对 JWS 安全令牌支持的 JSON Web 算法 (JWA) (继续)	
alg 参数值	**数字签名或 MAC 算法**
HS384	使用 SHA-384 的 HMAC
HS512	使用 SHA-512 的 HMAC
RS256	RSASSA-PKCS1-v1_5 (使用 SHA-256)
RS384	RSASSA-PKCS1-v1_5 使用 SHA-384
RS512	RSASSA-PKCS1-v1_5 ，使用 SHA-512

非对称密钥证书需求

如果使用非对称密钥对令牌进行签名，那么来自令牌发布者的公用密钥证书必须位于队列管理器用于令牌认证的密钥存储库中。 接收到认证令牌时，证书必须在其有效期内。 不进行检查以确保未撤销来自令牌签发者的证书。

认证令牌中的用户标识

如果队列管理器配置为采用认证令牌的用户声明中包含的用户标识作为应用程序的上下文，那么采用的用户标识必须满足以下要求:

- 最多可包含 12 个字符。
- 它必须以下列其中一个字符开头:

  A-Z a-z
- 它可以包含以下任何字符:

  0-9 A-Z a-z + ， -。 : = _
- 它不能是保留的用户标识 UNKNOWN 和 NOBODY 之一。

相关任务

配置队列管理器以接受 **AuthTokens**

相关参考

qm.ini 文件的 AuthToken 节

Linux V 9.4.0 AIX 配置队列管理器以使用 JWKS 端点接受认证令牌

配置在 AIX 或 Linux 上运行的 IBM MQ 队列管理器，以使用 JWKS 端点通过认证令牌来认证用户和应用程序。

开始之前

有关令牌如何使用 IBM MQ 的更多信息，请参阅 使用认证令牌。

在配置队列管理器之前，请检查队列管理器 **CONNAUTH** 属性中引用的 AUTHINFO 对象的类型是否为 IDPWOS。 仅当为操作系统用户标识和密码检查配置了队列管理器时，令牌认证才可用。

检查服务节的 **SecurityPolicy** 属性是否未设置为 Group。 如果 **SecurityPolicy** 显式设置为 Group，那么令牌认证不可用。 如果 **SecurityPolicy** 被设定为**团体**， 去除 **SecurityPolicy** 属性，然后重新启动队列管理器。

关于此任务

应用程序可以使用令牌向队列管理器进行认证。 IBM MQ 接受来自遵循建议因特网标准 RFC7519 的可信颁发者的 JSON Web 令牌 (*JWT*)。 您可以使用令牌来认证身份，然后可以将其用于将来的授权检查。

配置队列管理器以接受令牌的最简单方法是指向 JWKS 端点，如下所述。 如果认证服务未提供此类端点，或者由于其他原因 JWKS 不适合，请参阅 第 293 页的『配置队列管理器以使用本地密钥库接受认证令牌』。

过程

1. 请向认证服务器管理员询问以下详细信息:
   - 正确的 JWKS 端点 (URL)。
   - 此服务器用于加密 HTTP 流量的证书和/或签署此证书的权限。

     要点: 您应该始终通过 TLS/HTTPS 提供 JWKS 信息，并且需要此信息以确保队列管理器可以信任连接。
2. 通过在 qm.ini 文件中提供 **HTTPSKeyStore** ，配置队列管理器以创建出局 HTTPS 连接。
   有关更多信息，请参阅
   - qm.ini 文件中的 HTTPSKeyStore 说明。
   - 第 298 页的『创建密钥存储库以用作 TLS 信任库』.

   如果认证服务器使用定制证书 /CA ，那么您需要确保此 HTTPSKeyStore 中正确存在此证书 /CA。
3. 通过在 qm.ini 配置文件中定义 JWKS 节 来配置 JWKS 端点。
   附加节提供了以下内容:
   - **issuername**。 这必须与此权限签署的任何令牌中存在的 "iss" 声明相匹配，并且通常基于认证服务的 URL。
   - **endpoint**。 这是队列管理器从中查询用于验证令牌签名的公用密钥的地址。
   - **userclaim**。这是可选的，用于标识令牌中的定制字段，一旦验证了令牌，该定制字段应用于 IBM MQ 权限检查。

     注意: 如果您打算将 **ADOPTCTX**(YES) 用于此类连接，那么必须存在此连接。
4. 完成 .ini 文件更改后，发出命令 REFRESH SECURITY TYPE(AUTHINFO) 或重新启动队列管理器。
   如果配置成功，那么应用程序能够立即使用签名令牌进行连接。
   如果存在任何问题 (例如，无法联系认证服务以检索公用密钥)，那么将在队列管理器的 AMQERR01 日志文件中报告这些问题。

结果

您已成功配置队列管理器以接受使用 JWKS 端点的认证令牌。

注: 从认证服务器定期刷新密钥 (每 15 分钟)，如果连接应用程序提供未知密钥标识，那么会更频繁地刷新密钥。 通常，这意味着在证书到期并在服务器端被替换时，不需要进一步的 IBM MQ 配置操作来更新证书。 要强制立即刷新，请随时发出命令 REFRESH SECURITY TYPE(AUTHINFO) 。

相关概念

对认证令牌问题进行故障诊断

相关任务

在应用程序中使用认证令牌

相关参考

qm.ini 文件的 AuthToken 节

Linux V 9.4.0 AIX 配置队列管理器以使用本地密钥库接受认证令牌

配置 IBM MQ 队列管理器以使用认证令牌来认证用户和应用程序。

开始之前

如果可能，请考虑使用 JWKS 端点，请参阅 第 292 页的『配置队列管理器以使用 JWKS 端点接受认证令牌』，而不是手动配置令牌验证证书。 使用 JWKS 通常使初始配置和持续维护都更简单。

请参阅 使用认证令牌中的 "令牌如何与 IBM MQ 配合工作"。

在配置队列管理器之前，请检查队列管理器 **CONNAUTH** 属性中引用的 AUTHINFO 对象的类型是否为 IDPWOS。 仅当为操作系统用户标识和密码检查配置了队列管理器时，令牌认证才可用。

检查服务节的 **SecurityPolicy** 属性是否未设置为 Group。 如果 **SecurityPolicy** 显式设置为 Group，那么令牌认证不可用。 如果 **SecurityPolicy** 被设定为**团体**， 去除 **SecurityPolicy** 属性，然后重新启动队列管理器。

关于此任务

从 IBM MQ 9.3.4 应用程序可以使用令牌向队列管理器进行认证。 IBM MQ 接受来自遵循建议因特网标准 RFC7519 的可信颁发者的 JSON Web 令牌 (*JWT*)。 您可以使用令牌来认证身份，然后可以采用这些令牌进行将来的授权检查。

通过将可信签发者的公用密钥证书或对称密钥保存到队列管理器的密钥存储库，配置队列管理器以接受令牌。 将 AuthToken 节添加到 qm.ini 文件并刷新安全性配置，以便队列管理器选取新配置。

您可能想要配置本地密钥库，而不是在测试环境中使用 JWKS，或者当无法从队列管理器直接连接到认证服务器时。 除了任何 JWKS 端点之外，您还可以定义本地密钥库。

注: 如果 JWKS 端点和本地密钥库都为提供的令牌提供了匹配的签发者和 KID，那么首选使用 JWKS 端点提供的密钥。

在这些情况下，配置本地密钥库，如下所示:

过程

1. 创建密钥存储库。
   a) 为从可信签发者接收到的公用密钥证书或对称密钥创建密钥存储库。 您可以使用具有文件扩展名 .kdb 的 CMS 密钥存储库或具有文件扩展名 .p12 的 PKCS#12 密钥存储库。

      发出以下命令以创建 CMS 密钥存储库:

      ```
      runmqakm -keydb -create -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
      -type cms
      ```

      如果 **runmqakm** 命令返回错误，请参阅 runmqakm -keydb。 如果该命令成功完成，请使用 ls 命令列出目录的内容:

      ```
      ls -l /var/mqm/qmgrs/qm1/tokenissuer
      ```

      将显示以下文件:

      ```
      -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) 如果需要，请更改您创建的密钥存储库文件的组所有权，以便可以授予 mqm 组读访问权。 最初，只有运行该命令的管理用户才能访问创建的文件。

      ```
      chgrp mqm /var/mqm/qmgrs/qm1/tokenissuer/key.*
      ```

   c) 更改密钥存储库文件的方式以添加组 mqm 的读许可权。 例如，以下命令为文件所有者添加读/写许可权，并为组添加只读许可权。

      ```
      chmod 640 /var/mqm/qmgrs/qm1/tokenissuer/key.*
      ```

2. 使用 **runqmcred** 命令对密钥存储库密码进行加密，并将加密字符串保存到文件中。
   a) 创建文件以包含用于加密密钥存储库密码的初始密钥。

      该文件必须包含作为单行文本的初始键。 初始密钥的最大长度为 256 个字节。 如果已使用 **INITKEY** 队列管理器属性为队列管理器设置初始键，请将 **INITKEY** 属性的值复制到新文件中。 如果尚未为队列管理器设置初始密钥，请创建新的唯一加密密钥并将其添加到初始密钥文件。

      注: 有关更多信息，请参阅 INITKEY。 如果未指定初始密钥，那么将使用缺省密钥。 使用自己的初始密钥更安全。

      注: 授予对初始密钥文件的最低必要许可权，以确保文件的内容安全。 初始密钥文件仅用于加密密钥存储库密码。 因此，只有使用初始密钥对密码进行加密的管理员才需要访问读初始密钥文件。

b) 如果尚未设置队列管理器初始键，请将队列管理器 **INITKEY** 属性的值设置为您在步骤 第 294 页的『2.a』 中创建的初始键。 使用 **ALTER QMGR** 命令来设置队列管理器初始密钥。 例如：

```
ALTER QMGR INITKEY('myEncrypt10nK3y')
```

c) 发出 **runqmcred** 命令以加密密钥存储库密码。 使用 **-sf** 参数可指定包含初始密钥的文件的路径。

```
runqmcred -sf initial.key
```

提示时，输入密钥存储库密码。 加密密码由命令输出。

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
Enter password:
*************
<QM>!2!b5rb01sMzFzc1ClZeQMryruWFM3HSm8DKyEaZK7qzWY=!TrWdU57DCDXM0Qah99I/Lg==
```

复制最后一行上的字符串，并将其保存到文件中。

3. 使用下列其中一种方法将令牌发布者的公用密钥证书或对称密钥添加到密钥存储库。
   - 要将 RSA 公用密钥证书添加到密钥存储库，请发出以下命令：

```
runmqakm -cert -add -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
-label keylabel
         -file keyfile
```

   - 要将 base64 编码的对称密钥添加到密钥存储库，请发出以下命令：

```
runmqakm -secretkey -add -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
-label keylabel
         -file keyfile -format ascii
```

   其中， *keylabel* 是要附加到证书或密钥的标签，而 *keyfile* 是包含证书或 base64 编码密钥的文件的名称。
4. 将 **AuthToken** 节和以下属性添加到 `qm.ini` 文件：
   - 密钥存储库的路径，使用 **KeyStore** 属性指定。
   - 包含密钥存储库密码的文件，使用 **KeyStorePwdFile** 属性指定。
   - 在步骤 第 295 页的『3』 中添加的证书或对称密钥的标签，使用 **CertLabel** 属性指定。

   例如：

```
AuthToken:
   KeyStore=/var/mqm/qmgrs/qm1/tokenissuer/key.kdb
   KeyStorePwdFile=/var/mqm/qmgrs/qm1/tokenissuer/key.pw
   CertLabel=rsakey
```

   其中 `key.kdb` 是您在步骤 第 294 页的『1.a』 中创建的密钥存储库的名称， `key.pw` 是包含您在步骤 第 295 页的『2.c』 中创建的密钥存储库的加密密码的文件。

   有关 **AuthToken** 节的更多信息，请参阅 `qm.ini` 文件的 AuthToken 节。
5. 如果队列管理器配置为采用令牌用户声明中包含的用户标识以在后续授权检查中使用，请将 **UserClaim** 属性添加到 **AuthToken** 节。

   要确定队列管理器是否配置为采用令牌中的用户标识，请发出以下 MQSC 命令：

```
DISPLAY AUTHINFO(authinfo_name) ADOPTCTX
```

   其中 *authinfo_name* 是队列管理器 **CONNAUTH** 属性的值。 如果 **ADOPTCTX** 属性的值为 YES，那么队列管理器配置为采用令牌中的用户标识，并且必须在 **AuthToken** 节中指定 **UserClaim** 属性。

   将 **UserClaim** 属性的值设置为包含要采用的用户标识的令牌声明的名称。 例如，如果令牌包含声明 `"AppUser": "MyUserName"`，请将以下行添加到 **AuthToken** 节：

```
UserClaim=AppUser
```

6. 刷新队列管理器的安全性配置，以便从 qm.ini 文件中选取令牌配置。 发出以下命令以启动 **runmqsc** 命令:

```
runmqsc qm1
```

然后发出以下 MQSC 命令:

```
REFRESH SECURITY TYPE(CONNAUTH)
```

下一步做什么

与开发人员合作，帮助他们了解如何 在应用程序中使用令牌 向队列管理器进行认证。

相关概念

对认证令牌问题进行故障诊断

相关任务

在应用程序中使用认证令牌

相关参考

qm.ini 文件的 AuthToken 节

Linux V 9.4.0 AIX 从所选令牌签发者获取认证令牌

编写应用程序以在连接到 IBM MQ 队列管理器时从所选令牌发布者获取认证令牌。

开始之前

请参阅 第 296 页的『在应用程序中使用认证令牌』 中的信息。

过程

- 获取认证令牌的方式以及令牌的确切内容因不同令牌发布者而异。

  编写应用程序以与所选令牌发布者进行交互，以请求并获取认证令牌。 认证令牌必须符合认证令牌的 IBM MQ 需求。 有关这些需求的更多信息，请参阅 第 290 页的『认证令牌的需求』。

  如果打算采用令牌声明中包含的用户标识作为应用程序的上下文，那么认证令牌还必须满足以下需求:

  - 认证令牌必须包含与队列管理器的令牌认证配置中的用户声明名称匹配的声明。
  - 用户声明的值必须满足认证令牌中用户标识的要求。 有关更多信息，请参阅第 292 页的『认证令牌中的用户标识』。

结果

现在，您已获取格式正确的 JWT ，可将其提供给 IBM MQ 以进行验证。

相关任务

配置队列管理器以接受 **AuthTokens**

相关参考

qm.ini 文件的 AuthToken 节

MQCSP-安全性参数

Linux V 9.4.0 AIX 在应用程序中使用认证令牌

编写应用程序以在连接到 IBM MQ 队列管理器时提供认证令牌。

开始之前

从 IBM MQ 9.4.0 开始，应用程序可以在连接到队列管理器时提供认证令牌。

应用程序必须满足以下要求:

- 它必须以 C 或 Java 编写 (使用 IBM MQ classes for JMS/ Jakarta Messaging)

- 它必须作为 IBM MQ client 连接到队列管理器。 即，应用程序必须通过网络连接到队列管理器，而不是使用本地绑定。
- 它必须连接到在 AIX 或 Linux 上运行的队列管理器。

如果应用程序不满足这些需求，那么连接将失败，并且会将原因码 MQRC_FUNCTION_NOT_SUPPORTED (2298) 返回到应用程序。

提供认证令牌的应用程序可以在支持 IBM MQ MQI clients 的任何平台上运行。

使用自动客户机重新连接的客户机在连接时无法提供认证令牌。 如果应用程序提供认证令牌，并在 MQCNO 结构中指定 MQCNO_RECONNECT 或 MQCNO_RECONNECT_Q_MGR 选项，那么连接将失败并将原因码 MQRC_RECONNECT_不兼容 (2547) 返回到应用程序。 有关自动客户机重新连接的更多信息，请参阅自动客户机重新连接。

如果由于这些需求而无法编写应用程序以提供认证令牌，那么也可以通过使用客户机安全性出口来迁移应用程序以使用认证令牌。 可以写入客户机安全出口以在 MQCSP 结构中设置认证令牌。 有关安全出口的更多信息，请参阅 客户机连接上的安全出口。

从 IBM MQ 9.4.0 开始， JMS 客户机应用程序可以在连接时直接提供令牌 (请参阅 第 296 页的『从所选令牌签发者获取认证令牌』)。 在 IBM MQ 9.4.0 之前， Java 应用程序可以通过出口程序间接提供令牌。 有关更多信息，请参阅 Java 类 MQCSP。

关于此任务

注: 将对符合 JSON Web 签名 (JWS) 标准的认证令牌进行签名，以允许验证该令牌的真实性，但不会对其进行加密。 因此，具有令牌访问权的任何人都可以读取并可能复用该令牌。 配置与队列管理器的连接，以确保在通过网络 (例如，使用 TLS) 发送认证令牌时使用加密对其进行保护。 有关用于保护应用程序提供的凭证的选项的更多信息，请参阅 第 27 页的『MQCSP 密码保护』。

在修改应用程序以使用令牌进行连接之前，请确保:

- 队列管理器已配置为通过执行 第 293 页的『配置队列管理器以使用本地密钥库接受认证令牌』 中的步骤来接受认证令牌
- 应用程序可以根据需要从认证服务器获取有效令牌，请参阅 第 296 页的『从所选令牌签发者获取认证令牌』。

要在应用程序连接到 IBM MQ 队列管理器时提供认证令牌，请包含以下过程。

过程

- 要从 C (MQI) 应用程序提供认证令牌:
  应用程序必须使用 MQCONNX (而不是 MQCONN) 进行连接，并提供 MQCSP 结构:
  - **AuthenticationType** 字段必须设置为 MQCSP_AUTH_ID_TOKEN。
  - 该结构的版本必须设置为 MQCSP_VERSION_3。
  - **TokenPtr** 或 **TokenOffset** 字段必须引用您的认证令牌。
  - **TokenLength** 字段必须设置为认证令牌的长度。

  使用 MQCSP V 3 和认证令牌连接到队列管理器的示例 C 代码:

```
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         */
```

- 要从 Java 应用程序提供认证令牌:

  使用 IBM MQ classes for JMS/Jakarta Messaging 的应用程序可以通过采用用户名和密码的任何 `createContext` 或 `createConnection` 方法提供令牌。

  要提供认证令牌，请执行以下操作:

  – **UserID** 必须设置为 null 或空字符串，即，不带空格， ""

  – 令牌作为 **Password** 字符串提供。

  这适用于 `ConnectionFactory` 接口的所有 IBM MQ 实现。

  可以使用显式参数格式 (例如， `createContext`(String **userID**， String **password**)) 或隐式参数版本 (例如， `createContext`())。

  在后一种情况下，必须首先提供空的 **userID** 和令牌 **Password** 作为连接工厂上的属性。

  用于使用认证令牌连接到队列管理器的示例 Java 代码:

```
// 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
```

  如果连接失败，原因码为 MQRC_NOT_AUTHORIZED (2035) 或 MQRC_SECURITY_ERROR (2063)，请检查队列管理器错误日志以获取包含有关失败原因的更多信息的错误消息。 有关诊断认证令牌问题的更多帮助，请参阅 对认证令牌问题进行故障诊断。

结果

应用程序现在已连接到队列管理器。 它将保持连接状态，直到断开连接为止，即使用于认证的令牌到期也是如此。 如果应用程序与队列管理器断开连接并且需要重新连接，那么它可能需要获取新的认证令牌以及稍后的到期时间，然后才能重新连接。

相关任务

配置队列管理器以接受 **AuthTokens**

相关参考

`qm.ini` 文件的 AuthToken 节

MQCSP-安全性参数

Linux V 9.4.0 AIX 创建密钥存储库以用作 TLS 信任库

创建出局 TLS 连接时，您应该创建一个简单的 "信任库"，该信任库可以验证由一组公共认证中心 (CA) 签署的证书。 示例 TLS 连接是 IBM MQ 客户机通道或 HTTPS 连接，在配置 IBM MQ 的某些组件时使用。

关于此任务

注意: 决定在您的环境中信任哪些证书和认证中心是一个重要步骤，会影响端到端配置的安全性。 提供本主题是为了说明允许 IBM MQ 组件信任已为操作系统配置的同一组证书的常见步骤; 但是，如果有疑问，您应该与安全性管理员讨论此过程。

大多数基于 UNIX 和 Linux 的操作系统都具有包含 "可信" CA 集的文件系统位置。 此文件系统可能已通过操作系统安装进行配置，或者已由系统管理员进行定制 (例如，包含属于您组织的内部 CA)。 这些文件的位置各不相同，但流行操作系统的一些常用值为:

- 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`

创建和配置 IBM MQ 密钥库时，可以通过一个命令将目录中的所有证书文件 (例如 `/etc/ssl/certs`) 轻松添加到 IBM MQ 密钥数据库。

过程

1. 使用以下命令从 `/etc/ssl/certs` 目录添加证书文件:

```
runmqakm -cert -add -file /etc/ssl/certs/*.pem -db mykdb.p12 -stashed
```

2. 可选： 在某些情况下，为信任库生成一组 "缺省" 证书可能很有用。
   产品随附的 IBM MQ 安全组件提供了一组 "缺省" CA 证书。

   注: 这些证书可能不会频繁更新和/或具有相对较短的生命周期。

   如果仍要使用预先配置的 CA 证书，那么可以使用 **runmqakm** 命令中的 **populate** 和 **ibmcloudtrust** 参数来生成信任库:

```
runmqakm -keydb -create -db mqauto.p12 -genpw -stash -type pkcs12 -populate -ibmcloudtrust
```

相关概念
对认证令牌问题进行故障诊断
相关任务
在应用程序中使用认证令牌
相关参考
qm.ini 文件的 AuthToken 节

使用已撤销证书

认证中心可以撤销数字证书。 您可以根据平台，使用 OCSP 或 LDAP 服务器上的 CRL 来检查证书的撤销状态。

在 TLS 握手期间，通信伙伴使用数字证书相互认证。 认证过程中，可以检查所接收到的证书是否仍然可信。 由于各种原因，认证中心 (CA) 撤销证书，包括:

- 所有者已移至其他组织
- 专用密钥不再是私钥

CA 在证书撤销列表 (CRL) 中发布已撤销的个人证书。 已经被撤销的 CA 证书发布在权限撤销列表 (ARL) 中。

ALW 在 AIX, Linux, and Windows 平台上， IBM MQ SSL 支持使用 OCSP (联机证书状态协议) 或使用 LDAP (轻量级目录访问协议) 服务器上的 CRL 和 ARL 来检查已撤销的证书。 OCSP 是首选方法。

IBM MQ classes for Java 和 IBM MQ classes for JMS 不能使用客户机通道定义表文件中的 OCSP 信息。 但是，您可以按照使用联机证书协议中所述来配置 OCSP。

IBM i 在 IBM i 上， IBM MQ SSL 支持仅使用 LDAP 服务器上的 CRL 和 ARL 来检查已撤销的证书。

z/OS 在 z/OS 上， IBM MQ SSL 支持仅使用 LDAP 服务器上的 CRL 和 ARL 来检查已撤销的证书。

有关认证中心的更多信息，请参阅 第 12 页的『数字证书』。

OCSP/CRL 检查

对远程入局证书执行联机证书状态协议 (OCSP)/撤销列表 (CRL) 检查。 该过程会检查从远程系统的个人证书到其根证书所涉及的整个链。

使用 openSSL 验证 OCSP 验证

如果您的企业使用 openSSL 来验证 OCSP，然后尝试使用 IBM Global Security Kit (GSKit) TLS 连接，那么您将收到 UNKNOWN 状态警告。

这是因为除了根证书之外，链中的所有证书都由 GSKit 检查撤销状态。 GSKit 操作符合 RFC 5280，这在 GSKit 信任策略中进行了描述。 GSKit 算法尝试所有可用源以获取撤销信息，如 RFC 5280 和 GSKit 信任策略中所述。

OCSP/CRL 检查如何在 IBM MQ 中工作?

在针对指定的 OCSP 或 CRL 端点检查证书时， IBM MQ 支持两种机制 (在证书扩展中或在 AUTHINFO 对象中定义) 来控制行为:

- qm.ini 文件的 SSL 节的 **OCSPCheckExtensions**， **CDPCheckExtensions** 和 **OCSPAuthentication** 属性，以及
- 使用队列管理器的 SSLCRLNL 参数以及 AUTHINFO OCSP 和 CRLLDAP 配置。 请参阅 ALTER AUTHINFO 和 ALTER QMGR 以获取更多信息。

注意:

带有 **AUTHTYPE(OCSP)** 的 ALTER AUTHINFO 命令不适用于 IBM i 或 z/OS 队列管理器。 但是，可以在要复制到客户机通道定义表 (CCDT) 以供客户机使用的那些平台上指定此参数。

OCSPCheckExtensions 和 **CDPCheckExtensions** SSL 节属性控制 IBM MQ 是否将针对证书的 AIA 扩展中详细描述的 OCSP 或 CRL 服务器验证证书。

如果未启用，那么不会联系证书扩展中的 OCSP 或 CRL 服务器。

如果通过 AUTHINFO 对象详细描述了 OCSP 或 CRL 服务器，并使用 SSLCRLNL **QMGR** 属性进行引用，那么在证书撤销处理期间， IBM MQ 会尝试联系这些服务器。

要点: 在 SSLCRLNL 名称列表中只能定义一个 OCSP AUTHINFO 对象。

如果:

设置了 **OCSPCheckExtensions**= NO 和 **CDPCheckExtensions**=*NO*，并且
在 AUTHINFO 对象中未定义 OCSP 或 CRL 服务器

未执行证书撤销检查。

在验证证书的撤销状态时， IBM MQ 会联系按以下顺序命名的 OCSP 或 CRL 服务器 (如果已启用):

1. 在 **AUTHTYPE(OCSP)** 对象中详细描述并在 SSLCRLNL **QMGR** 属性中引用的 OCSP 服务器。
2. OCSP 服务器在证书的 AIA 扩展中详细说明 (如果 **OCSPCheckExtensions**=*YES*)。
3. 在证书的 **CRLDistributionPoints** 扩展中详细描述了 CRL 服务器 (如果 **CDPCheckExtensions** =*YES*)。
4. 在 **AUTHINFO(CRLLDAP)** 对象中详细描述并在 SSLCRLNL **QMGR** 属性中引用的任何 CRL 服务器。

在验证证书时，如果某个步骤导致 OCSP 服务器或 CRL 服务器返回对证书查询的最终 REVOKED 或 VALID 响应，那么不会执行进一步检查，并且所显示的证书状态将用于确定是否信任该证书。

如果 OCSP 服务器或 CRL 服务器返回结果 UNKNOWN，那么处理将继续，直到 OCSP 或 CRL 服务器返回最终结果或所有选项都已耗尽为止。

如果无法确定证书的状态，那么对于 OCSP 和 CRL 服务器，是否视为已撤销证书的行为有所不同:

- 对于 CRL 服务器，如果无法获取 CRL，那么会将证书视为 NOT_REVOKED

- 对于 OCSP 服务器，如果无法从指定的 OCSP 服务器获取撤销状态，那么将通过 qm.ini 文件的 SSL 节中的 **OCSPAuthentication** 属性来控制行为。

  您可以将此属性配置为阻止连接，允许连接或允许具有警告消息的连接。

您可以在 qm.ini 和 mqclient.ini 文件的 SSL 节中使用 **SSLHTTPProxyName**=*string* 属性进行 OCSP 检查 (如果需要)。 该字符串是要由 GSKit 用于 OCSP 检查的 HTTP 代理服务器的主机名或网络地址。

您可以在 `qm.ini` 或 `mqclient.ini` 文件的 SSL 节中设置 **OCSPTimeout** 值，该值用于设置在执行撤销检查时等待 OCSP 响应程序的秒数。

ALW 已撤销证书和 OCSP

IBM MQ 确定要使用的联机证书状态协议 (OCSP) 响应程序，并处理收到的响应。 您可能必须执行相应步骤以使 OCSP 响应程序可访问。

注: 此信息仅适用于 AIX, Linux, and Windows 系统上的 IBM MQ 。

要使用 OCSP 检查数字证书的撤销状态， IBM MQ 可以使用两种方法来确定要联系的 OCSP 响应程序:

- 通过使用要检查的证书中的 AuthorityInfoAccess (AIA) 证书扩展。
- 通过使用在认证信息对象中指定的 URL 或由客户机应用程序指定的 URL。

在认证信息对象中指定的 URL 或由客户机应用程序指定的 URL 优先于 AIA 证书扩展中的 URL。

如果 OCSP 响应程序的 URL 受防火墙保护，请重新配置防火墙以使 OCSP 响应程序可供访问，或者设置 OCSP 代理服务器。 通过在 SSL 节中使用 SSLHTTPProxyName 变量来指定代理服务器的名称。 在客户机系统中，还可以通过使用环境变量 MQSSLPROXY 来指定代理服务器的名称。 有关更多详细信息，请参阅相关信息。

如果您不关心 TLS 证书是否已撤销（可能是因为您要在测试环境中运行），那么可以在 SSL 节中将 OCSPCheckExtensions 设置为 NO。 如果设置此变量，那么将忽略任何 AIA 证书扩展。 此解决方案在生产环境中不大可行，因为在生产环境中您可能不希望允许用户访问已撤销证书。

用于访问 OCSP 响应程序的调用可能产生以下三种结果之一:

良好
证书有效。

已撤销
证书已撤销。

未知
产生此结果的可能原因有以下三种:

- IBM MQ 不能访问 OCSP 响应程序。
- OCSP 响应程序已发送响应，但 IBM MQ 无法验证该响应的数字签名。
- OCSP 响应程序已发送响应，指出它没有证书的撤销数据。

如果 IBM MQ 收到的 OCSP 结果为**未知**，那么其行为取决于 OCSPAuthentication 属性的设置。 对于队列管理器，此属性保存在下列其中一个位置:

- Linux AIX 在 AIX and Linux 上的 `qm.ini` 文件的 SSL 节中。
- Windows 在 Windows 注册表中。

可以使用 IBM MQ Explorer 来设置此属性。 对于客户机，该属性保存在客户机配置文件的 SSL 节中。

如果收到的结果为**未知**并且 OCSPAuthentication 设置为 REQUIRED（缺省值），那么 IBM MQ 将拒绝连接并且会发出类型为 AMQ9716 的错误消息。 如果启用了队列管理器 SSL 事件消息，那么将生成类型为 MQRC_CHANNEL_SSL_ERROR 的 SSL 事件消息（ReasonQualifier 设置为 MQRQ_SSL_HANDSHAKE_ERROR）。

如果收到的结果为**未知**并且 OCSPAuthentication 设置为 OPTIONAL，那么 IBM MQ 将允许启动 SSL 通道并且不会生成警告和 SSL 事件消息。

如果收到的结果为**未知**并且 OCSPAuthentication 设置为 WARN，那么 SSL 通道将启动，但 IBM MQ 会在错误日志中发出类型为 AMQ9717 的警告消息。 如果启用了队列管理器 SSL 事件消息，那么将生成类型为 MQRC_CHANNEL_SSL_WARNING 的 SSL 事件消息（ReasonQualifier 设置为 MQRQ_SSL_UNKNOWN_REVOCATION）。

OCSP 响应的数字签名

OCSP 响应程序可以采用以下三种方式之一对其响应进行签名。 响应程序将向您通知所使用的方法。

- 可以使用已发布您所检查证书的同一个 CA 证书对 OCSP 响应进行数字签名。 在这种情况下，您不需要设置任何其他证书; 您已执行的建立 TLS 连接的步骤足以验证 OCSP 响应。
- 可以使用已颁发待检查证书的认证中心 (CA) 签署的另一个证书对 OCSP 响应进行数字签名。 在这种情况下，签名证书会与 OCSP 响应一起发送。 从 OCSP 响应程序流出的证书必须将“扩展的密钥用法扩展”设置为 `id-kp-OCSPSigning`，这样就可以信任该证书以对 OCSP 响应进行数字签名。 由于 OCSP 响应是与对其进行签名的证书一起发送的 (并且该证书由已信任 TLS 连接的 CA 进行签名)，因此不需要其他证书设置。
- 可以使用与待检查证书不直接相关的另一个证书对 OCSP 响应进行数字签名。 在这种情况下，OCSP 响应由 OCSP 响应程序自身颁发的证书进行签名。 必须将 OCSP 响应程序证书的副本添加到执行 OCSP 检查的客户机或队列管理器的密钥数据库。 请参阅 第 481 页的『将 CA 证书或可信证书的公用部分添加到 AIX, Linux, and Windows 上的密钥存储库』。 在添加 CA 证书时，缺省情况下会将其添加为可信根，这在该上下文中是必需设置。 如果未添加此证书，那么 IBM MQ 无法验证 OCSP 响应上的数字签名，并且 OCSP 检查会产生未知结果，这可能会导致 IBM MQ 关闭通道，具体取决于 OCSPAuthentication 的值。

Java 和 JMS 客户机应用程序中的联机证书状态协议 (OCSP)

由于 Java API 的限制，仅当对整个 Java 虚拟机 (JVM) 进程启用了 OCSP 时， IBM MQ 才能对 TLS 安全套接字使用联机证书状态协议 (OCSP) 证书撤销检查。 可使用两种方法针对 JVM 中的所有安全套接字启用 OCSP：

- 编辑 JRE java.security 文件以包含表 1 中所示的 OCSP 配置设置，然后重新启动该应用程序。
- 使用 java.security.Security.setProperty() API，受任何 Java Security Manager 策略有效的约束。

您必须至少指定 ocsp.enable 和 ocsp.responderURL 值中的一个。

属性名	描述
ocsp.enable	该属性的值为 `true` 或 `false`。 如果为 `true`，将在执行证书撤销检查时启用 OCSP 检查；如果为 `false` 或未设置，将禁用 OCSP 检查。
ocsp.responderURL	该属性的值为用于标识 OCSP 响应程序位置的 URL。 下面是一个示例：`ocsp.responderURL=http://ocsp.example.net:80`。 缺省情况下，可通过要验证的证书间接确定 OCSP 响应程序的位置。 如果证书中缺少权限信息访问扩展（在 RFC 3280 中定义）或需要覆盖，那么将使用该属性。
ocsp.responderCertSubjectName	该属性的值为 OCSP 响应程序证书的主题名称。 下面是一个示例：`ocsp.responderCertSubjectName="CN=OCSP Responder, O=XYZ Corp"`。 缺省情况下，OCSP 响应程序的证书即是要验证的证书颁发者的证书。 该属性在缺省值不适用时标识 OCSP 响应程序的证书。 其值是一个字符串专有名称（在 RFC 2253 中定义），用于标识证书路径验证期间提供的证书集中的证书。 如果主题名称本身不足以唯一标识证书，那么必须改为同时使用 ocsp.responderCertIssuerName 和 ocsp.responderCertSerialNumber 属性。 如果设置了该属性，那么将忽略 ocsp.responderCertIssuerName 和 ocsp.responderCertSerialNumber 属性。
ocsp.responderCertIssuerName	该属性的值为 OCSP 响应程序证书的颁发者名称。 下面是一个示例：`ocsp.responderCertIssuerName="CN=Enterprise CA, O=XYZ Corp"`。 缺省情况下，OCSP 响应程序的证书即是要验证的证书颁发者的证书。 该属性在缺省值不适用时标识 OCSP 响应程序的证

属性名	描述
	书。 其值是一个字符串专有名称（在 RFC 2253 中定义），用于标识证书路径验证期间提供的证书集中的证书。 如果设置了该属性，那么还必须设置 ocsp.responderCertSerialNumber 属性。 如果设置了 ocsp.responderCertSubjectName 属性，那么将忽略该属性。
ocsp.responderCertSerialNumber	该属性的值为 OCSP 响应程序证书的序列号。 下面是一个示例：`ocsp.responderCertSerialNumber=2A:FF:00`。 缺省情况下，OCSP 响应程序的证书即是要验证的证书颁发者的证书。 该属性在缺省值不适用时标识 OCSP 响应程序的证书。 其值是一个十六进制数字字符串（可能存在冒号或空格分隔符），用于标识证书路径验证期间提供的证书集中的证书。 如果设置了该属性，那么还必须设置 ocsp.responderCertIssuerName 属性。 如果设置了 ocsp.responderCertSubjectName 属性，那么将忽略该属性。

以此方式启用 OCSP 之前，请注意以下事项：

- 设置 OCSP 配置会影响 JVM 进程中的所有安全套接字。 在某些情况下，当 JVM 与使用 TLS 安全套接字的其他应用程序代码共享时，此配置可能会产生不期望的副作用。 确保所选的 OCSP 配置适用于同一 JVM 中运行的所有应用程序。
- 对 JRE 应用维护可能会覆盖 java.security 文件。 应用 Java 临时修订和产品维护时请小心，以避免覆盖 java.security 文件。 应用维护后，可能需要重新应用 java.security 更改。 因此，您可以考虑改为使用 java.security.Security.setProperty() API 来设置 OCSP 配置。
- 仅当同时还启用撤销检查时，启用 OCSP 检查才会奏效。 可通过 `PKIXParameters.setRevocationEnabled()` 方法启用撤销检查。
- 如果您正在使用 在本机拦截器中启用 OCSP 检查中描述的 AMS Java 拦截器，请注意避免使用与密钥库配置文件中的 AMS OCSP 配置冲突的 java.security OCSP 配置。

使用证书撤销列表和权限撤销列表

IBM MQ 对 CRL 和 ARL 的支持因平台而异。

每个平台上的 CRL 和 ARL 支持如下所示：

- Multi 在多平台上， CRL 和 ARL 支持符合 PKIX X.509 V2 CRL 概要文件建议。
- z/OS 在 z/OS 上，系统 SSL 支持由 Tivoli Public Key Infrastructure 产品存储在 LDAP 服务器中的 CRL 和 ARL。

IBM MQ 维护在过去 12 小时内访问过的 CRL 和 ARL 的高速缓存。

当队列管理器或 IBM MQ MQI client 接收到证书时，它会检查 CRL 以确认该证书仍然有效。 IBM MQ 首先检入高速缓存 (如果存在高速缓存)。 如果 CRL 不在高速缓存中，那么 IBM MQ 将按照 LDAP CRL 服务器位置在由 *SSLCRLNL* 属性指定的认证信息对象的名称列表中的出现顺序来查询这些位置，直到 IBM MQ 找到可用的 CRL 为止。 如果未指定名称列表，或者使用空白值指定了名称列表，那么不会检查 CRL。

设置 *LDAP* 服务器

配置 LDAP 目录信息树结构以反映 CA 的专有名称层次结构。 使用 LDAP 数据交换格式文件执行此操作。

配置 LDAP 目录信息树 (DIT) 结构以使用对应于发放证书和 CRL 的 CA 的专有名称的层次结构。 您可以使用使用 LDAP 数据交换格式 (LDIF) 的文件来设置 DIT 结构。 您还可以使用 LDIF 文件来更新目录。

LDIF 文件是 ASCII 文本文件，其中包含在 LDAP 目录中定义对象所需的信息。 LDIF 文件包含一个或多个条目，每个条目包含一个专有名称，至少一个对象类定义以及 (可选) 多个属性定义。

`certificateRevocationList;binary` 属性包含已撤销用户证书的二进制格式列表。 `authorityRevocationList;binary` 属性包含已撤销的 CA 证书的二进制列表。 要与 IBM MQ TLS 配合使用，这些属性的二进制数据必须符合 DER (明确编码规则) 格式。 有关 LDIF 文件的更多信息，请参阅 LDAP 服务器随附的文档。

第 304 页的图 20 显示了一个样本 LDIF 文件，您可以创建该文件作为 LDAP 服务器的输入，以装入由 CA1 发出的 CRL 和 ARL，这是由测试组织在 IBM 中设置的具有专有名称 “CN=CA1， OU=Test， O=IBM，C=GB”的假想认证中心。

```
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)
```

图 *20:* 认证中心的样本 *LDIF* 文件。 这可能因实施而异。

第 304 页的图 21 显示了 LDAP 服务器在装入 第 304 页的图 20 中显示的样本 LDIF 文件时所创建的 DIT 结构以及 CA2 的类似文件 (也在 IBM 中由 PKI 组织设置的假想认证中心)。

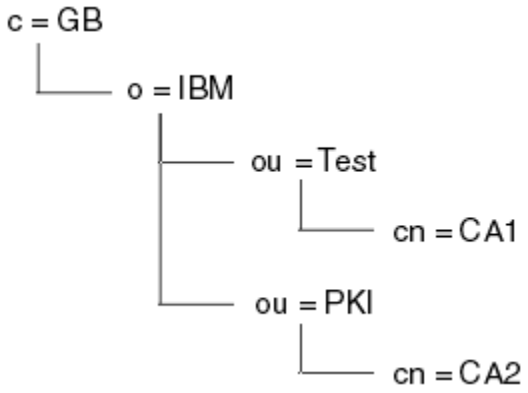


图 *21: LDAP* 目录信息树结构的示例

IBM MQ 检查 CRL 和 ARL。

注: 确保 LDAP 服务器的访问控制表允许授权用户读取，搜索和比较保存 CRL 和 ARL 的条目。 IBM MQ 使用 AUTHINFO 对象的 LDAPUSER 和 LDAPPWD 属性访问 LDAP 服务器。

配置和更新 *LDAP* 服务器

使用此过程来配置或更新 LDAP 服务器。

1. 从认证中心或权限获取 DER 格式的 CRL 和 ARL。
2. 使用文本编辑器或随 LDAP 服务器提供的工具，创建一个或多个 LDIF 文件，其中包含 CA 的专有名称和必需的对象类定义。 将 DER 格式数据作为 CRL 的 `certificateRevocationList;binary` 属性和/或 ARL 的 `authorityRevocationList;binary` 属性的值复制到 LDIF 文件中。
3. 启动 LDAP 服务器。
4. 添加您在步骤 第 304 页的『2』中创建的一个或多个 LDIF 文件中的条目。

配置 LDAP CRL 服务器后，请检查是否正确设置了该服务器。 首先，尝试使用通道上未撤销的证书，并检查通道是否正确启动。 然后使用已撤销的证书，并检查通道是否无法启动。

经常从认证中心获取更新的 CRL。 请考虑每 12 小时在 LDAP 服务器上执行此操作。

使用队列管理器访问 *CRL* 和 *ARL*

队列管理器与一个或多个认证信息对象相关联，这些对象保存 LDAP CRL 服务器的地址。 IBM i IBM i 上的 IBM MQ 的行为与其他平台不同。

请注意，在此部分中，有关证书撤销列表 (CRL) 的信息也适用于权限撤销列表 (ARL)。

您告诉队列管理器如何通过向队列管理器提供认证信息对象 (每个对象都包含 LDAP CRL 服务器的地址) 来访问 CRL。 认证信息对象保存在名称列表中，该名称列表在 *SSLCRLNL* 队列管理器属性中指定。

在以下示例中， MQSC 用于指定参数:

1. 使用 DEFINE AUTHINFO MQSC 命令定义认证信息对象，将 AUTHTYPE 参数设置为 CRLLDAP。
   IBM i 在 IBM i 上，还可以使用 CRTMQMAUTI CL 命令。

   AUTHTYPE 参数的值 CRLLDAP 指示在 LDAP 服务器上访问 CRL。 您创建的类型为 CRLLDAP 的每个认证信息对象都保存 LDAP 服务器的地址。 当您有多个认证信息对象时，它们指向的 LDAP 服务器必须包含相同的信息。 这将在一个或多个 LDAP 服务器发生故障时提供服务的连续性。

   z/OS 此外，仅在 z/OS 上，必须使用相同的用户标识和密码来访问所有 LDAP 服务器。 使用的用户标识和密码是在名称列表中的第一个 AUTHINFO 对象中指定的用户标识和密码。

   在所有平台上，用户标识和密码将以未加密方式发送到 LDAP 服务器。
2. 使用 DEFINE NAMELIST MQSC 命令，为认证信息对象的名称定义名称列表。 z/OS 在 z/OS 上，确保 NLTYPE 名称列表属性设置为 AUTHINFO。
3. 使用 ALTER QMGR MQSC 命令向队列管理器提供名称列表。 例如:

   ```
   ALTER QMGR SSLCRLNL(sslcrlnlname)
   ```

   其中 `sslcrlnlname` 是认证信息对象的名称列表。

   此命令设置名为 *SSLCRLNL* 的队列管理器属性。 此属性的队列管理器初始值为空。

IBM i 在 IBM i 上，可以指定认证信息对象，但队列管理器既不使用认证信息对象，也不使用认证信息对象的名称列表。 只有使用由 IBM i 队列管理器生成的客户机连接表的 IBM MQ 客户机才会使用为该 IBM i 队列管理器指定的认证信息。 IBM i 上的 *SSLCRLNL* 队列管理器属性确定此类客户机使用的认证信息。 有关告知 IBM i 队列管理器如何访问 CRL 的信息，请参阅 第 305 页的『在 IBM i 上访问 CRL 和 ARL』 。

您最多可以将 10 个到备用 LDAP 服务器的连接添加到名称列表，以确保在一个或多个 LDAP 服务器发生故障时服务的连续性。 请注意， LDAP 服务器必须包含相同的信息。

IBM i 在 *IBM i* 上访问 *CRL* 和 *ARL*

使用此过程可访问 IBM i 上的 CRL 或 ARL。

请注意，在此部分中，有关证书撤销列表 (CRL) 的信息也适用于权限撤销列表 (ARL)。

请遵循以下步骤在 IBM i 上为特定证书设置 CRL 位置:

1. 访问 DCM 接口，如 第 245 页的『访问 DCM』 中所述。
2. 在导航面板中的 **管理 CRL 位置** 任务类别中，单击 **添加 CRL 位置**。 "管理 CRL 位置" 页面将显示在任务框架中。
3. 在 **CRL 位置名** 字段中，输入 CRL 位置名，例如 `LDAP Server #1`
4. 在 **LDAP 服务器** 字段中，输入 LDAP 服务器名称。
5. 在 **使用安全套接字层 (SSL)** 字段中，如果要使用 TLS 连接到 LDAP 服务器，请选择 **是** 。 如果数据库中没有数据，那么选择**否**。
6. 在 **端口号** 字段中，输入 LDAP 服务器的端口号，例如 389。
7. 如果 LDAP 服务器不允许匿名用户查询目录，请在 **登录专有名称** 字段中输入服务器的登录专有名称。
8. 单击**确定**。 DCM 通知您它已创建 CRL 位置。
9. 在导航面板中，单击 **选择证书库**。 "选择证书库" 页面将显示在任务框架中。

10. 选中 **其他系统证书库** 复选框，然后单击 **继续**。 将显示 "证书库和密码" 页面。
11. 在 **证书库路径和文件名** 字段中，输入您在 第 247 页的『在 IBM i 上创建证书库』时设置的 IFS 路径和文件名。
12. 在 **证书库密码** 字段中输入密码。 单击**继续**。 "当前证书库" 页面将显示在任务框架中。
13. 在导航面板中的 **管理证书** 任务类别中，单击 **更新 CRL 位置分配**。 "CRL 位置分配" 页面将显示在任务框架中。
14. 选择要向其分配 CRL 位置的 CA 证书的单选按钮。 单击 **更新 CRL 位置分配**。 "更新 CRL 位置分配" 页面将显示在任务框架中。
15. 选择要分配给证书的 CRL 位置的单选按钮。 单击 **更新分配**。 DCM 通知您它已更新分配。

请注意， DCM 允许您通过认证中心分配其他 LDAP 服务器。

使用 *IBM MQ Explorer* 访问 *CRL* 和 *ARL*
您可以使用 IBM MQ Explorer 来告知队列管理器如何访问 CRL。

请注意，在此部分中，有关证书撤销列表 (CRL) 的信息也适用于权限撤销列表 (ARL)。

使用以下过程来设置与 CRL 的 LDAP 连接:

1. 确保已启动队列管理器。
2. 右键单击 **认证信息** 文件夹，然后单击 **新建-> 认证信息**。 在打开的属性表中:
   a. 在第一页 **创建认证信息**上，输入 CRL (LDAP) 对象的名称。
   b. 在 **更改属性**的 " **常规** " 页面上，选择连接类型。 (可选) 可以输入描述。
   c. 选择 **更改属性**的 **CRL (LDAP)** 页面。
   d. 输入 LDAP 服务器名称作为网络名或 IP 地址。
   e. 如果服务器需要登录详细信息，请提供用户标识和密码 (如果需要)。
   f. 单击**确定**。
3. 右键单击 "名称列表" 文件夹，然后单击 **新建-> 名称列表**。 在打开的属性表中:
   a. 输入名称列表的名称。
   b. 添加 CRL (LDAP) 对象的名称 (从步骤 第 306 页的『2.a』 开始) 到列表中。
   c. 单击**确定**。
4. 右键单击队列管理器，选择 **属性**，然后选择 **SSL** 页面:
   a. 选中 **根据证书撤销列表检查此队列管理器接收的证书** 复选框。
   b. 输入名称列表的名称 (从步骤 第 306 页的『3.a』 开始) 在 **CRL 名称列表** 字段中。

使用 *IBM MQ MQI client* 访问 *CRL* 和 *ARL*

您有三个选项可用于指定用于存放 CRL 以供 IBM MQ MQI client 检查的 LDAP 服务器。

请注意，在此部分中，有关证书撤销列表 (CRL) 的信息也适用于权限撤销列表 (ARL)。

指定 LDAP 服务器的三种方法如下所示:

- 使用通道定义表
- 在 MQCONNX 调用上使用 SSL 配置选项结构 MQSCO
- 使用 Active Directory (在具有 Active Directory 支持的 Windows 系统上)

有关更多详细信息，请参阅相关信息。

您最多可以包含 10 个与备用 LDAP 服务器的连接，以确保在一个或多个 LDAP 服务器发生故障时服务的连续性。 请注意， LDAP 服务器必须包含相同的信息。

无法从 Linux (zSeries 平台) 上运行的 IBM MQ MQI client 通道访问 LDAP CRL。

OCSP 响应程序的位置以及保存 *CRL* 的 *LDAP* 服务器的位置
在 IBM MQ MQI client 系统上，可以指定 OCSP 响应程序以及保存证书撤销列表 (CRL) 的轻量级目录访问协议 (LDAP) 服务器的位置。

您可以通过三种方式指定这些位置，此处按优先顺序递减顺序进行了描述。

IBM i 对于 IBM i，请参阅 在 IBM i 上访问 CRL 和 ARL。

当 IBM MQ MQI client 应用程序发出 MQCONNX 调用时

您可以指定 OCSP 响应程序或在 **MQCONNX** 调用上保存 CRL 的 LDAP 服务器。

在 **MQCONNX** 调用上，连接选项结构 MQCNO 可以引用 SSL 配置选项结构 MQSCO。 反过来， MQSCO 结构可以引用一个或多个认证信息记录结构 MQAIR。 每个 MQAIR 结构都包含 IBM MQ MQI client 访问 OCSP 响应程序或保存 CRL 的 LDAP 服务器所需的所有信息。 例如， MQAIR 结构中的其中一个字段是可与响应程序联系的 URL。 有关 MQAIR 结构的更多信息，请参阅 MQAIR-认证信息记录。

使用客户机通道定义表 (ccdt) 来访问 OCSP 响应程序或 LDAP 服务器

为了使 IBM MQ MQI client 可以访问用于存放 CRL 的 OCSP 响应程序或 LDAP 服务器，请在客户机通道定义表中包含一个或多个认证信息对象的属性。

在服务器队列管理器上，可以定义一个或多个认证信息对象。 认证对象的属性包含访问 OCSP 响应程序 (在支持 OCSP 的平台上) 或保存 CRL 的 LDAP 服务器所需的所有信息。 其中一个属性指定 OCSP 响应程序 URL，另一个属性指定运行 LDAP 服务器的系统的主机地址或 IP 地址。

z/OS IBM i 具有 AUTHTYPE (OCSP) 的认证信息对象不适用于 IBM i 或 z/OS 队列管理器，但可以在要复制到客户机通道定义表 (CCDT) 以供客户机使用的平台上指定该对象。

要使 IBM MQ MQI client 能够访问包含 CRL 的 OCSP 响应程序或 LDAP 服务器，可以在客户机通道定义表中包含一个或多个认证信息对象的属性。 您可以通过下列其中一种方式包含此类属性:

Multi

在服务器平台上: AIX， Linux， IBM i 和 Windows

您可以定义包含一个或多个认证信息对象的名称的名称列表。 然后，可以将队列管理器属性 **SSLCRLNL** 设置为此名称列表的名称。

如果您正在使用 CRL，那么可以配置多个 LDAP 服务器以提供更高的可用性。 其意图是每个 LDAP 服务器都具有相同的 CRL。 如果一个 LDAP 服务器在需要时不可用，那么 IBM MQ MQI client 可以尝试访问另一个 LDAP 服务器。

此处将名称列表所标识的认证信息对象的属性统称为 证书撤销位置。 将队列管理器属性 **SSLCRLNL** 设置为名称列表的名称时，会将证书撤销位置复制到与队列管理器关联的客户机通道定义表中。 如果可以从客户机系统作为共享文件访问 CCDT，或者如果随后将 CCDT 复制到客户机系统，那么该系统上的 IBM MQ MQI client 可以使用 CCDT 中的证书撤销位置来访问包含 CRL 的 OCSP 响应程序或 LDAP 服务器。

如果稍后更改了队列管理器的证书撤销位置，那么此更改将反映在与队列管理器相关联的 CCDT 中。 如果队列管理器属性 **SSLCRLNL** 设置为空白，那么将从 CCDT 中除去证书撤销位置。 这些更改不会反映在客户机系统上的表的任何副本中。

如果您要求 MQI 通道的客户机端和服务器端的证书撤销位置不同，并且服务器队列管理器是用于创建证书撤销位置的服务器队列管理器，那么可以执行以下操作:

1. 在服务器队列管理器上，创建要在客户机系统上使用的证书撤销位置。
2. 将包含证书撤销位置的 CCDT 复制到客户机系统。
3. 在服务器队列管理器上，将证书撤销位置更改为 MQI 通道的服务器端所需的位置。
4. 在客户机上，可以使用带有 **-n** 参数的 **runmqsc** 命令。

Multi

在客户机平台上: AIX， Linux， IBM i 和 Windows

通过将 runmqsc 命令与 CCDT 文件中的 **-n** 参数和 **DEFINE AUTHINFO** 对象配合使用，可以在客户机上构建 CCDT。 对象的定义顺序是它们在文件中的使用顺序。 您可能在 **DEFINE AUTHINFO** 对象中使用的任何名称都不会保留在文件中。 当您 **DISPLAY** CCDT 文件中的 **AUTHINFO** 对象时，仅使用位置号。

注: 如果指定 **-n** 参数，那么不得指定任何其他参数。

在 Windows 上使用 Active Directory

Windows

在 Windows 系统上，可以使用 **setmqcrl** 控制命令在 Active Directory 中发布当前 CRL 信息。

命令 **setmqcrl** 不发布 OCSP 信息。

有关此命令及其语法的信息，请参阅 setmqcrl。

使用 *IBM MQ classes for Java* 和 *IBM MQ classes for JMS* 访问 *CRL* 和 *ARL*

IBM MQ classes for Java 和 IBM MQ classes for JMS 访问 CRL 的方式与其他平台不同。

有关使用 IBM MQ classes for Java 的 CRL 和 ARL 的信息，请参阅 使用证书撤销列表

有关使用 IBM MQ classes for JMS 的 CRL 和 ARL 的信息，请参阅 SSLCERTSTORES 对象属性

处理认证信息对象

您可以使用 MQSC 或 PCF 命令或 IBM MQ Explorer 来处理认证信息对象。

以下 MQSC 命令对认证信息对象执行操作:

- 定义授权信息
- 变更授权信息
- 删除授权信息
- 显示授权信息

有关这些命令的完整描述，请参阅 MQSC 命令。

以下可编程命令格式 (PCF) 命令对认证信息对象起作用:

- 创建认证信息
- 复制认证信息
- 更改认证信息
- 删除认证信息
- 查询认证信息
- 查询认证信息名称

有关这些命令的完整描述，请参阅 可编程命令格式的定义。

在可用的平台上，您还可以使用 IBM MQ Explorer。

Linux AIX 使用可插拔认证方法 (PAM)

只能在 AIX and Linux 平台上使用 PAM。 典型 AIX 或 Linux 系统具有用于实现传统认证机制的 PAM 模块; 但是，可能有更多模块。 除了验证密码的基本任务外，还可以调用 PAM 模块来执行其他规则。

配置文件定义要用于每个应用程序的认证方法。 示例应用程序包括标准终端登录， ftp 和 telnet。

PAM 的优点是应用程序不需要知道或关心用户标识实际上是如何被认证的。 只要应用程序能向 PAM 提供正确的认证数据形式，其背后的机制就是透明的。

认证数据的形式取决于所使用的系统。 例如， IBM MQ 通过参数 (例如 MQCONNX API 调用中使用的 MQCSP 结构) 获取密码。

要点: 除非安装 IBM MQ 8.0.0 Fix Pack 3，然后使用 **-e CMDLEVEL=**级别 *802* (在 strmqm 命令上) 来设置所需的命令级别，否则无法设置 **AUTHENMD** 属性，然后重新启动队列管理器。

配置系统以使用 PAM

调用 PAM 时， IBM MQ 使用的服务名称为 *ibmmq*。

请注意， IBM MQ 安装尝试根据不同操作系统的已知缺省值来维护允许来自操作系统用户的连接的缺省 PAM 配置。

但是，系统管理员必须验证 `/etc/pam.conf` 或 `/etc/pam.d/ibmmq` 文件中定义的规则是否仍然适用。

授予对对象的访问权

本部分包含有关使用对象权限管理器和通道出口程序来控制对对象的访问的信息。

ALW 在 AIX, Linux, and Windows 系统上。 您可以使用对象权限管理器 (OAM) 来控制对对象的访问。 此主题集合包含有关使用 OAM 的命令接口的信息。

此部分还包含可用于确定要执行哪些任务以在所有平台上将安全性应用于系统的核对表，以及授予用户管理 IBM MQ 和使用 IBM MQ 对象的权限的注意事项。

如果提供的安全性机制不满足您的需求，那么您可以开发自己的通道出口程序。

确定用于授权的用户

将向用户所属的组授予访问资源的权限，或者在特定方式下向与连接关联的用户直接授予访问资源的权限。在连接过程中，尤其是对于远程 (客户机) 连接，此身份可由队列管理器的配置进行更改。 此页面列出了 IBM MQ 的不同功能部件及其配置选项，这些功能部件可能会影响连接应用程序的身份以及这些功能部件生效的优先顺序。

可修改采用的用户的功能

可以设置哪些用户应该获得授权的不同功能如下:

应用程序声明的用户
当 IBM MQ 启动远程连接时，会将进程正在运行的操作系统用户发送到接收队列管理器。 发送此用户以确保如果不存在用于修改用户的进一步配置，那么存在可用于授权检查的用户。

建议不要将此用户用作授权的基础，因为它允许连接在不进行任何服务器端验证的情况下声明其身份。这甚至可能包括管理用户 ("mqm")。

通道 MCAUSER 设置
通过网络绑定进行连接的应用程序通过使用 IBM MQ 通道定义进行连接。 通道定义支持 **MCAUSER** 属性，该属性可用于指定要用于授权的其他用户，而不是连接应用程序所声明的用户。

连接认证 ADOPTCTX
应用程序可以指定要发送到队列管理器以进行认证的用户和密码。 这些凭证使用为 "连接认证" 功能部件指定的配置进行认证。 "连接认证" 的 **ADOPTCTX** 选项控制在成功验证用户之后是否应将其用于授权。如果设置为 YES，那么将采用为认证提供的用户进行授权检查。

V 9.4.0 从 IBM MQ 9.3.4 开始，可以提供令牌以进行认证，如果 **ADOPTCTX** 设置为 YES，那么将从令牌包含的声明中采用用户。

通道认证记录 MCAUSER
在连接处理期间，队列管理器将尝试查找与连接匹配的通道认证记录。 如果通道认证记录匹配，并且其 **USERSRC** 属性值设置为 MAP，那么 IBM MQ 会将用于授权的用户更改为 **MCAUSER** 属性的值。

安全出口
安全出口是可在 IBM MQ 安全性处理期间编写和调用的定制函数。 当调用此函数时，将向其提供 MQCD 结构的副本，其中包含与将用于授权检查的连接用户相关的多个字段。 安全出口可以修改这些字段以更改将获得授权的用户。

优先顺序

下表显示了当 IBM MQ 选择要授权的用户时， 第 309 页的『可修改采用的用户的功能』 中描述的每个安全功能的优先顺序。 顺序从最低到最高，即用户在第一行的安全功能设置被任何其他行覆盖。

表 *71:* 安全功能部件的优先顺序	
顺序	**功能部件**
1 (最低)	应用程序断言标识
2	通道定义 **MCAUSER** 属性
3	使用 **ADOPTCTX(YES)** 进行连接认证
4	使用 **USERSRC(MAP)** 的通道认证记录
5 (最高)	安全出口

早期采用的影响

连接认证和通道认证记录提供了一个配置选项，用于控制何时执行连接认证用户采用。 此设置称为 "早期采用"。 如果启用早期采用，那么在处理通道认证记录之前会采用连接认证身份 (这意味着通道认证记录将覆盖任何 **CONNAUTH** 采用)。

如果禁用此选项，那么将撤销顺序-即，在 **CONNAUTH** 采用之前处理通道认证记录。 在这种情况下，采用连接认证会使通道认证记录具有更高的有效优先级。

早期采用的缺省设置为 enabled。

ALW 通过在 AIX, Linux, and Windows 上使用 OAM 来控制对对象的访问

对象权限管理器 (OAM) 提供用于授予和撤销对 IBM MQ 对象的权限的命令接口。

您必须获得适当的授权才能使用这些命令，如 第 353 页的『在 AIX, Linux, and Windows 上管理 IBM MQ 的权限』 中所述。 有权管理 IBM MQ 的用户标识对队列管理器具有 超级用户 权限，这意味着您不必授予他们进一步的许可权以发出任何 MQI 请求或命令。

Linux AIX AIX and Linux 上基于 OAM 用户的许可权

在 UNIX and Linux 系统上，对象权限管理器 (OAM) 可以使用基于用户的授权以及基于组的授权。

在 IBM MQ 8.0 之前， UNIX and Linux 上的访问控制表 (ACL) 仅基于组。 从 IBM MQ 8.0 开始， ACL 基于用户标识和组，您可以使用基于用户的模型或基于组的模型进行授权，方法是将 **SecurityPolicy** 属性设置为相应的值，如 qm.ini 文件的 Service 节中所述。

IBM MQ 8.0 和更高版本的行为更改

从 IBM MQ 8.0 开始，当使用基于用户的策略运行时，某些命令返回与产品的较早版本不同的信息:

- **dmpmqaut** 和 **dmpmqcfg** 命令显示基于用户的记录， PCF 等效操作也是如此。
- IBM MQ Explorer 的 OAM 插件显示基于用户的记录并允许基于用户的修改。
- OAM **Inquire** 函数返回显示其支持用户的结果。

在 qm.ini 文件中启用基于用户的权限时，使用 **setmqaut** 命令上的 **-p** 属性不会授予同一主组中的所有用户访问权，如 qm.ini 文件的服务节中所述。

如果您开始采用基于用户的授权并且拥有许多用户，那么与基于组的模型相比， AUTH 队列上存储的记录可能更多，并且授权过程可能需要比先前更长的时间，因为要验证的记录更多。 预计这一增幅不会很大。 如果需要，您可以混合使用用户和组许可权。

迁移考虑因素

如果将现有队列管理器的模型从组更改为用户，那么不会立即生效。 已经做出的授权继续适用。 连接到队列管理器的任何用户都将获得与之前相同的特权: 其标识所属的所有组的组合。 当针对用户标识发出新的 **setmqaut** 命令时，这些命令将立即生效。

如果使用用户策略创建新的队列管理器，那么此队列管理器仅对创建该队列管理器的用户具有许可权 (通常是但不一定是 mqm 用户标识)。 还会自动向 mqm 组授予许可权。 但是，如果您没有将 mqm 作为主组，那么 mqm 组不会包含在初始权限集中。

如果从用户移动到组策略，那么不会自动删除基于用户的权限。 但是，在许可权检查期间不再使用这些参数。 在还原策略之前，保存当前配置，更改策略，重新启动队列管理器，然后重放脚本。 因为它现在是基于组的队列管理器，所以效果是基于主组存储用户标识规则。

相关概念

对象权限管理器 (OAM)

第 356 页的『AIX, Linux, and Windows 上的主体和组』

主体可以属于组。 通过向组而不是个人授予资源访问权，可以减少所需的管理量。 访问控制表 (ACL) 基于组和用户标识。

相关参考

qm.ini 文件的服务节

crtmqm (创建队列管理器) 命令

ALW 授予对 AIX, Linux, and Windows 上的 IBM MQ 对象的访问权

使用 **setmqaut** 控制命令， **SET AUTHREC** MQSC 命令或 **MQCMD_SET_AUTH_REC** PCF 命令来授予用户和用户组对 IBM MQ 对象的访问权。 请注意，在 IBM MQ Appliance 上，只能使用 **SET AUTHREC** 命令。

有关 **setmqaut** 控制命令及其语法的完整定义，请参阅 setmqaut。

有关 **SET AUTHREC** MQSC 命令及其语法的完整定义，请参阅 SET AUTHREC。

有关 **MQCMD_SET_AUTH_REC** PCF 命令及其语法的完整定义，请参阅 设置权限记录。

队列管理器必须正在运行才能使用此命令。 当您更改了主体的访问权时， OAM 将立即反映这些更改。

要授予用户对对象的访问权，需要指定:

- 拥有您正在使用的对象的队列管理器的名称; 如果未指定队列管理器的名称，那么将采用缺省队列管理器。
- 对象的名称和类型 (用于唯一地标识对象)。 将名称指定为 概要文件 ; 这是对象的显式名称或通用名称 (包括通配符)。 有关通用概要文件的详细描述以及在这些概要文件中使用通配符的信息，请参阅 第 312 页的『在 AIX, Linux, and Windows 上使用 OAM 通用概要文件』。
- 权限应用的一个或多个主体和组名。

  如果用户标识包含空格，请在使用此命令时将其括在引号中。 在 Windows 系统上，您可以使用域名来限定用户标识。 如果实际用户标识包含 at 符号 (@) 符号，请将其替换为 @ @ ， 以显示它是用户标识的一部分，而不是用户标识与域名之间的定界符。
- 权限列表。 列表中的每个项都指定要授予该对象 (或从该对象中撤销) 的访问权类型。 列表中的每个授权都指定为关键字，前缀为加号 (+) 或减号 (-)。 使用加号来添加指定的授权，使用减号来除去授权。 + 或- 符号与关键字之间不得有空格。

  您可以在单个命令中指定任意数量的权限。 例如，允许用户或组将消息放入队列并进行浏览，但撤销获取消息的访问权的权限列表如下:

```
+browse -get +put
```

使用 setmqaut 命令的示例

以下示例显示如何使用 setmqaut 命令授予和撤销使用对象的许可权:

```
setmqaut -m saturn.queue.manager -t queue -n RED.LOCAL.QUEUE
         -g groupa +browse -get +put
```

在该示例中:

- saturn.queue.manager 是队列管理器名称
- queue 是对象类型
- RED.LOCAL.QUEUE 是对象名
- groupa 是具有要更改的权限的组的标识
- +browse -get +put 是指定队列的权限列表
  - +browse 添加用于在队列上浏览消息的权限 (使用浏览选项发出 **MQGET**)
  - -get 从队列中除去获取 (**MQGET**) 消息的权限
  - +put 添加将 (**MQPUT**) 消息放入队列的权限

以下命令从主体 fvuser 以及组 groupa 和 groupb 中撤销对队列 MyQueue 的 put 权限。 在 AIX and Linux 系统上，此命令还将撤销与 fvuser 位于同一主组中的所有主体的放置权限。

```
setmqaut -m saturn.queue.manager -t queue -n MyQueue -p fvuser
         -g groupa -g groupb -put
```

将 setmqaut 命令与其他授权服务配合使用

如果您正在使用自己的授权服务而不是 OAM，那么可以在 **setmqaut** 命令上指定此服务的名称，以将该命令定向到此服务。 如果同时运行了多个可安装组件，那么必须指定此参数; 否则，将对授权服务的第一个可安装组件进行更新。 缺省情况下，这是提供的 OAM。

SET AUTHREC 的使用说明

要添加的授权的列表和要移除的授权的列表不能重叠。 例如，不能使用同一个命令添加显示权限和移除显示权限。 即使使用不同的选项表达权限，此条规则也适用。 例如，以下命令由于 DSP 权限与 ALLADM 权限重叠而失败:

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(DSP) AUTHRMV(ALLADM)
```

此重叠行为的例外情况是 ALL 权限。 以下命令将先添加 ALL 权限，然后移除 SETID 权限:

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(ALL) AUTHRMV(SETID)
```

以下命令将先移除 ALL 权限，然后添加 DSP 权限:

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(DSP) AUTHRMV(ALL)
```

无论在命令上提供这些权限的顺序如何，都将先处理 ALL。

ALW 在 AIX, Linux, and Windows 上使用 OAM 通用概要文件

使用 OAM 通用概要文件在单个操作中设置用户对许多对象的特权; 而不必在创建对象时针对每个单独的对象发出单独的 **setmqaut** 命令或 **SET AUTHREC** 命令。 请注意，在 IBM MQ Appliance 上，只能使用 **SET AUTHREC** 命令。

通过在 setmqaut 或 SET AUTHREC 命令中使用通用概要文件，您可以为适合该概要文件的所有对象设置通用权限。

此主题集合更详细地描述了通用概要文件的使用。

在 OAM 概要文件中使用通配符

使概要文件通用的是在概要文件名称中使用特殊字符 (通配符)。 例如，问号 (?) 通配符与名称中的任何单个字符匹配。 因此，如果指定 ABC.?EF，那么您对该概要文件的授权将应用于名称为 ABC.DEF， ABC.CEF 和 ABC.BEF 等的任何对象。

可用的通配符包括:

?

使用问号（?）代替任何单个字符。 例如， AB.?D 适用于对象 AB.CD， AB.ED 和 AB.FD。

*****

使用星号 (*) 作为:

- 概要文件名称中的 限定符 ，用于与对象名中的任何一个限定符匹配。 限定符是用句点定界的对象名的一部分。 例如，在 ABC.DEF.GHI 中，限定符是 ABC、DEF 和 GHI。

  例如， ABC.*.JKL 适用于对象 ABC.DEF.JKL 和 ABC.GHI.JKL。 (请注意，它 **不** 适用于 ABC.JKL ; 在此上下文中使用的 * 始终指示一个限定符。)
- 概要文件名称中限定符内的字符，用于与对象名称中限定符内的零个或多个字符匹配。

  例如， ABC.DE*.JKL 适用于对象 ABC.DE.JKL， ABC.DEF.JKL 和 ABC.DEGH.JKL。

******

在概要文件名称中使用双星号 (**) **once** 作为:

- 要与所有对象名匹配的整个概要文件名称。 例如，如果使用 -t prcs 来标识进程，然后使用 ** 作为概要文件名称，那么将更改所有进程的权限。
- 作为概要文件名称中的开头，中间或结尾限定符，以匹配对象名称中的零个或多个限定符。 例如， **.ABC 标识具有最终限定符 ABC 的所有对象。

只能使用双星号 ** 作为完整限定符:

```
**.DEF
  ABC.**
  A*.**
```

但不像

```
A**
```

否则，您将收到消息 AMQ7226E: **概要文件名称无效。**

注: 在 AIX and Linux 系统上使用通配符时， **必须** 将概要文件名称括在单引号中。

概要文件优先级

在使用通用概要文件时要了解的一个重要问题是，在确定要应用于要创建的对象的权限时，概要文件的优先级。 例如，假设您已发出以下命令:

```
setmqaut -n AB.* -t q +put -p fred
setmqaut -n AB.C* -t q +get -p fred
```

第一个为具有与概要文件 AB. * 匹配的名称的主体 fred 的所有队列授予放置权限; 第二个队列授予对与概要文件 AB.C*。

假设您现在创建名为 AB.CD。 根据通配符匹配规则，可以将 setmqaut 应用于该队列。 那么，它是放了还是得到了权威?

要查找答案，请应用以下规则: 每当多个概要文件可以应用于某个对象时， **只有最具体的应用**。 应用此规则的方法是将概要文件名称从左到右进行比较。 无论它们有什么不同，非通用字符都比通用字符更具体。 因此，在此示例中，队列为 AB.CD 具有 **get** 权限 (AB.C* 比 AB. * 更具体。

比较通用字符时， 特异性 的顺序为:

1. ?
2. *
3. **

转储概要文件设置

有关 **dmpmqaut** 控制命令及其语法的完整定义，请参阅 dmpmqaut。

有关 **DISPLAY AUTHREC** MQSC 命令及其语法的完整定义，请参阅 DISPLAY AUTHREC。

有关 **MQCMD_INQUIRE_AUTH_RECS** PCF 命令及其语法的完整定义，请参阅 查询权限记录。

以下示例显示了如何使用 **dmpmqaut** 控制命令来转储通用概要文件的权限记录:

1. 此示例转储具有与主体 user1 的队列 a.b.c 匹配的概要文件的所有权限记录。

   ```
   dmpmqaut -m qm1 -n a.b.c -t q -p user1
   ```

   生成的转储如下所示:

   ```
   profile:     a.b.*
   object type: queue
   entity:      user1
   type:        principal
   authority:   get, browse, put, inq
   ```

   注: 虽然 AIX and Linux 上的用户可以将 -p 选项用于 **dmpmqaut** 命令，但他们在定义权限时必须改为使用 -g groupname 。

2. 此示例转储具有与队列 a.b.c 匹配的概要文件的所有权限记录。

   ```
   dmpmqaut -m qmgr1 -n a.b.c -t q
   ```

   生成的转储如下所示:

   ```
   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. 此示例转储概要文件 a.b 的所有权限记录。* 类型为队列。

   ```
   dmpmqaut -m qmgr1 -n a.b.* -t q
   ```

   生成的转储如下所示:

   ```
   profile:     a.b.*
   object type: queue
   entity:      user1
   ```

```
type:        principal
authority:   get, browse, put, inq
```

4. 此示例转储队列管理器 qmX 的所有权限记录。

```
dmpmqaut -m qmX
```

生成的转储如下所示:

```
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. 此示例转储队列管理器 qmX 的所有概要文件名称和对象类型。

```
dmpmqaut -m qmX -l
```

生成的转储如下所示:

```
profile: q1, type: queue
profile: q*, type: queue
profile: name.*, type: namelist
profile: pr1, type: process
```

注: 仅对于 IBM MQ for Windows， 显示的所有主体都包含域信息，例如:

```
profile:     a.b.*
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

ALW 在 *AIX, Linux, and Windows* 上的 *OAM* 概要文件中使用通配符

在对象权限管理器 (OAM) 概要文件名称中使用通配符以使该概要文件适用于多个对象。

使概要文件通用的是在概要文件名称中使用特殊字符 (通配符)。 例如，问号 (?) 通配符与名称中的任何单个字符匹配。 因此，如果指定 ABC.?EF，那么您对该概要文件的授权将应用于名称为 ABC.DEF， ABC.CEF 和 ABC.BEF 等的任何对象。

可用的通配符包括:

?

使用问号（?）代替任何单个字符。 例如， AB.?D 适用于对象 AB.CD， AB.ED 和 AB.FD。

*****

使用星号 (*) 作为:

- 概要文件名称中的 限定符 ，用于与对象名中的任何一个限定符匹配。 限定符是用句点定界的对象名的一部分。 例如，在 ABC.DEF.GHI 中，限定符是 ABC、DEF 和 GHI。

  例如， ABC.*.JKL 适用于对象 ABC.DEF.JKL 和 ABC.GHI.JKL。 (请注意，它 **不** 适用于 ABC.JKL ; 在此上下文中使用的 * 始终指示一个限定符。)
- 概要文件名称中限定符内的字符，用于与对象名称中限定符内的零个或多个字符匹配。

  例如， ABC.DE*.JKL 适用于对象 ABC.DE.JKL， ABC.DEF.JKL 和 ABC.DEGH.JKL。

**

在概要文件名称中使用双星号 (**) **once** 作为:

- 要与所有对象名匹配的整个概要文件名称。 例如，如果使用 -t prcs 来标识进程，然后使用 ** 作为概要文件名称，那么将更改所有进程的权限。
- 作为概要文件名称中的开头，中间或结尾限定符，以匹配对象名称中的零个或多个限定符。 例如， **.ABC 标识具有最终限定符 ABC 的所有对象。

注: 在 AIX and Linux 系统上使用通配符时， **必须** 将概要文件名称括在单引号中。

ALW *AIX, Linux, and Windows* 上的概要文件优先级

多个通用概要文件可以应用于单个对象。 在这种情况下，适用最具体的规则。

在使用通用概要文件时要了解的一个重要问题是，在确定要应用于要创建的对象的权限时，概要文件的优先级。 例如，假设您已发出以下命令:

```
setmqaut -n AB.* -t q +put -p fred
setmqaut -n AB.C* -t q +get -p fred
```

第一个为具有与概要文件 AB. * 匹配的名称的主体 fred 的所有队列授予放置权限; 第二个队列授予对与概要文件 AB.C*。

假设您现在创建名为 AB.CD。 根据通配符匹配规则，可以将 setmqaut 应用于该队列。 那么，它是放了还是得到了权威?

要查找答案，请应用以下规则: 每当多个概要文件可以应用于某个对象时， **只有最具体的应用**。 应用此规则的方法是将概要文件名称从左到右进行比较。 无论它们有什么不同，非通用字符都比通用字符更具体。 因此，在此示例中，队列为 AB.CD 具有 **get** 权限 (AB.C* 比 AB. * 更具体。

比较通用字符时， 特异性 的顺序为:

1. ?
2. *
3. **

请参阅 SET AUTHREC 以获取使用此 MQSC 命令时的等效信息。

ALW 在 *AIX, Linux, and Windows* 上转储概要文件设置

使用 **dmpmqaut** 控制命令， **DISPLAY AUTHREC** MQSC 命令或 **MQCMD_INQUIRE_AUTH_RECS** PCF 命令来转储与指定概要文件关联的当前权限。 请注意，在 IBM MQ Appliance 上，只能使用 **DISPLAY AUTHREC** 命令。

有关 **dmpmqaut** 控制命令及其语法的完整定义，请参阅 dmpmqaut。

有关 **DISPLAY AUTHREC** MQSC 命令及其语法的完整定义，请参阅 DISPLAY AUTHREC。

有关 **MQCMD_INQUIRE_AUTH_RECS** PCF 命令及其语法的完整定义，请参阅 查询权限记录。

以下示例显示了如何使用 **dmpmqaut** 控制命令来转储通用概要文件的权限记录:

1. 此示例转储具有与主体 user1 的队列 a.b.c 匹配的概要文件的所有权限记录。

```
dmpmqaut -m qm1 -n a.b.c -t q -p user1
```

生成的转储类似于以下示例:

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

注: AIX and Linux 用户不能使用 -p 选项; 他们必须改为使用 -g groupname 。

2. 此示例转储具有与队列 a.b.c 匹配的概要文件的所有权限记录。

```
dmpmqaut -m qmgr1 -n a.b.c -t q
```

生成的转储类似于以下示例:

```
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. 此示例转储概要文件 a.b 的所有权限记录。* 类型为队列。

```
dmpmqaut -m qmgr1 -n a.b.* -t q
```

生成的转储类似于以下示例:

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

4. 此示例转储队列管理器 qmX 的所有权限记录。

```
dmpmqaut -m qmX
```

生成的转储类似于以下示例:

```
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. 此示例转储队列管理器 qmX 的所有概要文件名称和对象类型。

```
dmpmqaut -m qmX -l
```

生成的转储类似于以下示例:

```
profile: q1, type: queue
profile: q*, type: queue
profile: name.*, type: namelist
profile: pr1, type: process
```

注: 仅对于 IBM MQ for Windows ， 显示的所有主体都包含域信息， 例如:

```
profile:     a.b.*
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

ALW 在 AIX, Linux, and Windows 上显示访问权设置

使用 **dspmqaut** 控制命令， **DISPLAY AUTHREC** MQSC 命令或 **MQCMD_INQUIRE_ENTITY_AUTH** PCF 命令来查看特定主体或组对特定对象具有的权限。 请注意，在 IBM MQ Appliance 上，只能使用 **DISPLAY AUTHREC** 命令。

队列管理器必须正在运行才能使用此命令。 更改主体的访问权时， OAM 会立即反映这些更改。 一次只能显示一个组或主体的授权。

有关 **dmpmqaut** 控制命令及其语法的完整定义，请参阅 dmpmqaut。

有关 **DISPLAY AUTHREC** MQSC 命令及其语法的完整定义，请参阅 DISPLAY AUTHREC。

有关 **MQCMD_INQUIRE_AUTH_RECS** PCF 命令及其语法的完整定义，请参阅 查询权限记录。

以下示例显示了如何使用 **dspmqaut** 控制命令来显示组 GpAdmin 对队列管理器 QueueMan1 上名为 Annuities 的进程定义的权限。

```
dspmqaut -m QueueMan1 -t process -n Annuities -g GpAdmin
```

ALW 在 AIX, Linux, and Windows 上更改和撤销对 IBM MQ 对象的访问权

要更改用户或组对对象的访问级别，请使用 **setmqaut** 控制命令， **DELETE AUTHREC** MQSC 命令或 **MQCMD_DELETE_AUTH_REC** PCF 命令。 MQ Appliance请注意，在 IBM MQ Appliance 上，只能使用 **DELETE AUTHREC** 命令。

以下描述了从组中除去用户的过程:

- Windows 第 126 页的『在 Windows 上创建和管理组』
- AIX 第 124 页的『在 AIX 上创建和管理组』
- Linux 第 125 页的『在 Linux 上创建和管理组』

.

创建 IBM MQ 对象的用户标识被授予对该对象的完全控制权限。 如果从本地 mqm 组 (或 Windows 系统上的 Administrators 组) 中除去此用户标识，那么不会撤销这些权限。 在从 mqm 或 Administrators 组中除去对象之后，使用 **setmqaut** 控制命令或 **MQCMD_DELETE_AUTH_REC** PCF 命令来撤销创建该对象的用户标识对该对象的访问权。

有关 setmqaut 控制命令及其语法的完整定义，请参阅 setmqaut。

有关 **DELETE AUTHREC** MQSC 命令及其语法的完整定义，请参阅 DELETE AUTHREC。

有关 **MQCMD_DELETE_AUTH_REC** PCF 命令及其语法的完整定义，请参阅 删除权限记录。

Windows 在 Windows 上，从 IBM MQ 8.0，您可以使用 **setmqaut** 的 **-u** *SID* 参数随时删除与特定 Windows 用户帐户对应的 OAM 条目。

在 IBM MQ 8.0 之前，必须先删除对应于特定 Windows 用户帐户的 OAM 条目，然后再删除用户概要文件。除去用户帐户后无法除去 OAM 条目。

ALW 阻止对 AIX, Linux, and Windows 系统进行安全访问检查

注: 本主题描述不建议启用的功能。 要关闭安全性检查，可以禁用对象权限管理器 (OAM)。 这可能适用于测试环境。 禁用后，队列管理器将无法再执行授权或连接认证检查。 仍可使用 TLS，通道认证记录和安全出口。 禁用或除去 OAM 后，无法将 OAM 添加到现有队列管理器。

如果您决定不想执行安全性检查 (例如，在测试环境中)，那么可以通过以下两种方法之一来禁用 OAM:

- 在创建队列管理器之前，请设置操作系统环境变量 **MQSNOAUT**。

  有关设置 **MQSNOAUT** 环境变量的含义以及如何在 AIX, Linux, and Windows 上设置 **MQSNOAUT** 的信息，请参阅 环境变量描述。
- 编辑队列管理器配置文件以除去服务。

警告: 除去 OAM 时，不能将其放回现有队列管理器。 这是因为 OAM 需要在对象创建时就位。 要在除去 IBM MQ OAM 后再次使用 OAM，请重建队列管理器。

如果在禁用 OAM 时使用 **setmqaut** 或 **dspmqaut** 命令，请注意以下几点:

- OAM 不会验证指定的主体或组，这意味着命令可以接受无效值。
- OAM 不执行安全性检查，并指示所有主体和组都有权执行所有适用的对象操作。
- 不会验证传递到 OAM 以进行认证检查的任何凭证。

相关概念

AIX, Linux, and Windows 的可安装服务和组件

相关任务

配置可安装服务

相关参考

可安装服务参考信息

授予对资源的必需访问权

使用本主题来确定要执行哪些任务以将安全性应用于 IBM MQ 系统。

关于此任务

在此任务期间，您将决定需要哪些操作才能将相应级别的安全性应用于 IBM MQ 安装的元素。 您引用的每个单独任务都会针对所有平台提供逐步指示信息。

过程

1. 是否需要将队列管理器的访问权限制为某些用户?
   a) 否: 不采取进一步行动。
   b) 是: 转至下一个问题。

2. 这些用户是否需要对一部分队列管理器资源进行部分管理访问?
   a) 否: 转至下一个问题。
   b) 是: 请参阅 第 320 页的『授予对队列管理器资源子集的部分管理访问权』。
3. 这些用户是否需要对队列管理器资源子集的完全管理访问权?
   a) 否: 转至下一个问题。
   b) 是: 请参阅 第 328 页的『授予对队列管理器资源子集的完全管理访问权』。
4. 这些用户是否需要对所有队列管理器资源的只读访问权?
   a) 否: 转至下一个问题。
   b) 是: 请参阅 第 333 页的『授予对队列管理器上所有资源的只读访问权』。
5. 这些用户是否需要对所有队列管理器资源的完全管理访问权?
   a) 否: 转至下一个问题。
   b) 是: 请参阅 第 334 页的『授予对队列管理器上所有资源的完全管理访问权』。
6. 您是否需要用户应用程序来连接到队列管理器?
   a) 否: 禁用连接，如 第 335 页的『除去与队列管理器的连接』 中所述
   b) 是: 请参阅 第 336 页的『允许用户应用程序连接到队列管理器』。

z/OS Multi 授予对队列管理器资源子集的部分管理访问权

您需要授予某些用户对某些队列管理器资源 (但不是全部) 的部分管理访问权。 使用此表来确定需要执行的操作。

表 *72:* 授予对队列管理器资源子集的部分管理访问权

用户需要管理此类型的对象	**执行此操作**
队列	授予对所需队列的部分管理访问权，如 第 320 页的『授予对某些队列的有限管理访问权』 中所述
主题	授予对必需主题的部分管理访问权，如 第 322 页的『授予对某些主题的有限管理访问权』 中所述
通道	授予对所需通道的部分管理访问权，如 第 323 页的『授予对某些通道的有限管理访问权』 中所述
队列管理器	授予对队列管理器的部分管理访问权，如 第 324 页的『授予对队列管理器的有限管理访问权』 中所述
进程	授予对所需流程的部分管理访问权，如 第 325 页的『授予对某些进程的有限管理访问权』 中所述
名称列表	授予对所需名称列表的部分管理访问权，如 第 326 页的『授予某些名称列表有限的管理访问权』 中所述
服务	授予对所需服务的部分管理访问权，如 第 327 页的『授予对某些服务的有限管理访问权』 中所述

授予对某些队列的有限管理访问权

将对队列管理器上某些队列的部分管理访问权授予具有业务需求的每组用户。

关于此任务

要为某些操作授予对某些队列的有限管理访问权，请对操作系统使用相应的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

注: MQ Appliance 在 IBM MQ Appliance 上，只能使用 **SET AUTHREC** 命令。

过程

- ALW
  对于 AIX, Linux, and Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName ReqdAction
```

- IBM i
  对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS 对于 z/OS，发出以下命令以授予对指定队列的访问权:

```
RDEFINE MQADMIN QMgrName.QUEUE. ObjectProfile UACC(NONE)
PERMIT QMgrName.QUEUE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  要指定用户可以在队列上执行哪些 MQSC 命令，请针对每个 MQSC 命令发出以下命令:

```
RDEFINE MQCMDS QMgrName. ReqdAction. QType UACC(NONE)
PERMIT QMgrName. ReqdAction. QType CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

  要允许用户使用 DISPLAY QUEUE 命令，请发出以下命令:

```
RDEFINE MQCMDS QMgrName.DISPLAY. QType UACC(NONE)
PERMIT QMgrName.DISPLAY. QType CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

  变量名称具有以下含义:

  QMgrName
  队列管理器的名称。

  z/OS 在 z/OS 上，此值也可以是队列共享组的名称。

  ObjectProfile
  要更改其权限的对象或通用概要文件的名称。

  GroupName
  要授予访问权的组的名称。

  ReqdAction
  允许组执行的操作:

  – ALW 在 AIX, Linux, and Windows 系统上，以下权限的任意组合: + chg， + clr， + dlt 和 + dsp。 authorization + alladm 相当于 + chg + clr + dlt + dsp。

  – IBM i 在 IBM i 上，以下权限的任意组合: *ADMCHG， *ADMCLR， *ADMDLT， *ADMDSP。 授权 *ALLADM 等同于所有这些单独的授权。

  – z/OS 在 z/OS 上，值为 ALTER， CLEAR， DELETE 或 MOVE 之一。

  注: 为队列授予 + crt 间接使用户或组成为管理员。 请勿使用 + crt 权限来授予对某些队列的有限管理访问权。

  QTYPE

  对于 DISPLAY 命令，值为 QUEUE， QLOCAL， QALIAS， QMODEL， QREMOTE 或 QCLUSTER 之一。

  对于 *ReqdAction* 的其他值，值为 QLOCAL， QALIAS， QMODEL 或 QREMOTE 之一。

授予对某些主题的有限管理访问权

将队列管理器上某些主题的部分管理访问权授予具有业务需求的每组用户。

关于此任务

要授予对某些操作的某些主题的有限管理访问权，请针对您的操作系统使用相应的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW

  对于 AIX, Linux, and Windows 系统，请发出以下命令:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName ReqdAction
  ```

- IBM i

  对于 IBM i，请发出以下命令:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS 对于 z/OS，请发出以下命令:

  ```
  RDEFINE MQADMIN QMgrName.TOPIC. ObjectProfile UACC(NONE)
  PERMIT QMgrName.TOPIC. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  这些命令授予对指定主题的访问权。 要确定用户可以对主题执行哪些 MQSC 命令，请针对每个 MQSC 命令发出以下命令:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction.TOPIC UACC(NONE)
  PERMIT QMgrName. ReqdAction.TOPIC CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  要允许用户使用 DISPLAY TOPIC 命令，请发出以下命令:

  ```
  RDEFINE MQCMDS QMgrName.DISPLAY.TOPIC UACC(NONE)
  PERMIT QMgrName.DISPLAY.TOPIC CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  变量名称具有以下含义:

  QMgrName
  队列管理器的名称。

  z/OS 在 z/OS 上，此值也可以是队列共享组的名称。

  ObjectProfile
  要更改其权限的对象或通用概要文件的名称。

  GroupName
  要授予访问权的组的名称。

  ReqdAction
  允许组执行的操作:

  – ALW 在 AIX, Linux, and Windows 系统上，以下权限的任意组合: + chg， + clr， + crt， + dlt 和 + dsp。 + ctrl。 authorization + alladm 相当于 + chg + clr + dlt + dsp。

  – IBM i 在 IBM i 上，以下权限的任意组合: *ADMCHG， *ADMCLR， *ADMCRT， *ADMDLT， *ADMDSP 和 *CTRL。 授权 *ALLADM 等同于所有这些单独的授权。

– z/OS 在 z/OS 上，值为 ALTER， CLEAR， DEFINE， DELETE 或 MOVE 之一。

授予对某些通道的有限管理访问权

向具有业务需求的每组用户授予对队列管理器上某些通道的部分管理访问权。

关于此任务

要为某些操作授予对某些通道的有限管理访问权，请对操作系统使用相应的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW
  在 AIX, Linux, and Windows 上：

  ```
  setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName ReqdAction
  ```

- IBM i
  在 IBM i 上：

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*CHL) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS 在 z/OS 上：

  ```
  RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
  PERMIT QMgrName.CHANNEL. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  这些命令授予对指定通道的访问权。 要确定用户可以在通道上执行哪些 MQSC 命令，请针对每个 MQSC 命令发出以下命令：

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction.CHANNEL UACC(NONE)
  PERMIT QMgrName. ReqdAction.CHANNEL CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  要允许用户使用 DISPLAY CHANNEL 命令，请发出以下命令：

  ```
  RDEFINE MQCMDS QMgrName.DISPLAY.CHANNEL UACC(NONE)
  PERMIT QMgrName.DISPLAY.CHANNEL CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  变量名称具有以下含义：

  QMgrName
  队列管理器的名称。

  z/OS 在 z/OS 上，此值也可以是队列共享组的名称。

  ObjectProfile
  要更改其权限的对象或通用概要文件的名称。

  GroupName
  要授予访问权的组的名称。

  ReqdAction
  允许组执行的操作：

  – ALW 在 AIX, Linux, and Windows 上，以下权限的任意组合：+ chg， + clr， + crt， + dlt 和 + dsp。 + ctrl， + ctrlx。 authorization + alladm 相当于 + chg + clr + dlt + dsp。

– IBM i 在 IBM i 上，以下权限的任意组合: *ADMCHG， *ADMCLR， *ADMCRT， *ADMDLT， *ADMDSP， *CTRL 和 *CTRLx。 授权 *ALLADM 等同于所有这些单独的授权。

– z/OS 在 z/OS 上，值为 ALTER， CLEAR， DEFINE， DELETE 或 MOVE 之一。

授予对队列管理器的有限管理访问权

将队列管理器的部分管理访问权授予具有业务需求的每组用户。

关于此任务

要授予有限的管理访问权以在队列管理器上执行某些操作，请针对您的操作系统使用相应的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW

  在 AIX, Linux, and Windows 上:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName ReqdAction
  ```

- IBM i

  在 IBM i 上:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS

  在 z/OS 上:

  要确定可以在队列管理器上执行哪些 MQSC 命令，请针对每个 MQSC 命令发出以下命令:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction.QMGR UACC(NONE)
  PERMIT QMgrName. ReqdAction.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  要允许用户使用 DISPLAY QMGR 命令，请发出以下命令:

  ```
  RDEFINE MQCMDS QMgrName.DISPLAY.QMGR UACC(NONE)
  PERMIT QMgrName.DISPLAY.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  变量名称具有以下含义:

  QMgrName
  队列管理器的名称。

  ObjectProfile
  要更改其权限的对象或通用概要文件的名称。

  GroupName
  要授予访问权的组的名称。

  ReqdAction
  允许组执行的操作:

  – ALW 在 AIX, Linux, and Windows 上，以下权限的任意组合: + chg， + clr， + crt， + dlt 和 + dsp。 authorization + alladm 相当于 + chg + clr + dlt + dsp。

    虽然 + set 是 MQI 授权，通常不被视为管理，但在队列管理器上授予 + set 可能会间接导致完全管理权限。 请勿将 + 设置授予普通用户和应用程序。

– IBM i 在 IBM i 上，以下权限的任意组合: *ADMCHG， *ADMCLR， *ADMCRT， *ADMDLT， *ADMDSP。 授权 *ALLADM 等同于所有这些单独的授权。

授予对某些进程的有限管理访问权

将对队列管理器上某些进程的部分管理访问权授予具有业务需求的每组用户。

关于此任务

要为某些操作授予对某些进程的有限管理访问权，请针对您的操作系统使用相应的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW
  在 AIX, Linux, and Windows 上:

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName ReqdAction
```

- IBM i
  在 IBM i 上:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS 在 z/OS 上:

```
RDEFINE MQADMIN QMgrName.PROCESS. ObjectProfile UACC(NONE)
PERMIT QMgrName.PROCESS. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  这些命令授予对指定通道的访问权。 要确定用户可以在通道上执行哪些 MQSC 命令，请针对每个 MQSC 命令发出以下命令:

```
RDEFINE MQCMDS QMgrName. ReqdAction.PROCESS UACC(NONE)
PERMIT QMgrName. ReqdAction.PROCESS CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

  要允许用户使用 DISPLAY PROCESS 命令，请发出以下命令:

```
RDEFINE MQCMDS QMgrName.DISPLAY.PROCESS UACC(NONE)
PERMIT QMgrName.DISPLAY.PROCESS CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

  变量名称具有以下含义:

  QMgrName
  队列管理器的名称。

  z/OS 在 z/OS 上，此值也可以是队列共享组的名称。

  ObjectProfile
  要更改其权限的对象或通用概要文件的名称。

  GroupName
  要授予访问权的组的名称。

  ReqdAction
  允许组执行的操作:

  – ALW 在 AIX, Linux, and Windows 上，以下权限的任意组合: + chg， + clr， + crt， + dlt 和 + dsp。 authorization + alladm 相当于 + chg + clr + dlt + dsp。

– IBM i 在 IBM i 上，以下权限的任意组合: *ADMCHG， *ADMCLR， *ADMCRT，*ADMDLT， *ADMDSP。 授权 *ALLADM 等同于所有这些单独的授权。

– z/OS 在 z/OS 上，值为 ALTER， CLEAR， DEFINE， DELETE 或 MOVE 之一。

授予某些名称列表有限的管理访问权

将队列管理器上某些名称列表的部分管理访问权授予具有业务需求的每组用户。

关于此任务

要授予某些名称列表对某些操作的有限管理访问权，请针对您的操作系统使用相应的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW
  在 AIX, Linux, and Windows 上:

```
setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName ReqdAction
```

- IBM i
  在 IBM i 上:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*NMLIST) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS 在 z/OS 上:

```
RDEFINE MQADMIN QMgrName.NAMELIST. ObjectProfile UACC(NONE)
PERMIT QMgrName.NAMELIST. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  这些命令授予对指定名称列表的访问权。 要确定用户可以在名称列表上执行哪些 MQSC 命令，请针对每个 MQSC 命令发出以下命令:

```
RDEFINE MQCMDS QMgrName. ReqdAction.NAMELIST UACC(NONE)
PERMIT QMgrName. ReqdAction.NAMELIST CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

  要允许用户使用 DISPLAY NAMELIST 命令，请发出以下命令:

```
RDEFINE MQCMDS QMgrName.DISPLAY.NAMELIST UACC(NONE)
PERMIT QMgrName.DISPLAY.NAMELIST CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

  变量名称具有以下含义:

  QMgrName
  队列管理器的名称。

  z/OS 在 z/OS 上，此值也可以是队列共享组的名称。

  ObjectProfile
  要更改其权限的对象或通用概要文件的名称。

  GroupName
  要授予访问权的组的名称。

  ReqdAction
  允许组执行的操作:

- ALW 在 AIX, Linux, and Windows 上，以下权限的任意组合: + chg， + clr， + crt， + dlt， + ctrl， + ctrlx 和 + dsp。 authorization + alladm 相当于 + chg + clr + dlt + dsp。
- IBM i 在 IBM i 上，以下权限的任意组合: *ADMCHG， *ADMCLR， *ADMCRT， *ADMDLT， *ADMDSP， *CTRL 和 *CTRLX。 授权 *ALLADM 等同于所有这些单独的授权。
- z/OS 在 z/OS 上，值为 ALTER， CLEAR， DEFINE， DELETE 或 MOVE 之一。

授予对某些服务的有限管理访问权

将对队列管理器上某些服务的部分管理访问权授予具有业务需求的每组用户。

关于此任务

要为某些操作授予对某些服务的有限管理访问权，请针对您的操作系统使用相应的命令。 z/OS 请注意， z/OS 上不存在服务对象。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW
  在 AIX, Linux, and Windows 上:

```
setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName ReqdAction
```

- 在 IBM i 上:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*SVC) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS 在 z/OS 上:

  这些命令授予对指定服务的访问权。 要确定用户可以对服务执行哪些 MQSC 命令，请针对每个 MQSC 命令发出以下命令:

```
RDEFINE MQCMDS QMgrName. ReqdAction.SERVICE UACC(NONE)
PERMIT QMgrName. ReqdAction.SERVICE CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

  要允许用户使用 DISPLAY SERVICE 命令，请发出以下命令:

```
RDEFINE MQCMDS QMgrName.DISPLAY.SERVICE UACC(NONE)
PERMIT QMgrName.DISPLAY.SERVICE CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

  变量名称具有以下含义:

  QMgrName
  队列管理器的名称。

  ObjectProfile
  要更改其权限的对象或通用概要文件的名称。

  GroupName
  要授予访问权的组的名称。

  ReqdAction
  允许组执行的操作:

  - ALW 在 AIX, Linux, and Windows 系统上，以下权限的任意组合: + chg， + clr， + crt， + dlt， + ctrl， + ctrlx 和 + dsp。 authorization + alladm 相当于 + chg + clr + dlt + dsp。

– IBM i 在 IBM i 上，以下权限的任意组合: *ADMCHG， *ADMCLR， *ADMCRT， *ADMDLT， *ADMDSP， *CTRL 和 *CTRLX。 授权 *ALLADM 等同于所有这些单独的授权。

授予对队列管理器资源子集的完全管理访问权

您需要授予某些用户对某些队列管理器资源 (但不是全部) 的完全管理访问权。 使用这些表来确定需要执行的操作。

表 *73:* 授予对队列管理器资源子集的完全管理访问权

用户需要管理此类型的对象	执行此操作
队列	授予对所需队列的完全管理访问权，如 第 328 页的『授予对某些队列的完全管理访问权』 中所述
主题	授予对所需主题的完全管理访问权，如 第 329 页的『授予对某些主题的完全管理访问权』 中所述
通道	授予对所需通道的完全管理访问权，如 第 329 页的『授予对某些通道的完全管理访问权』 中所述
队列管理器	授予对队列管理器的完全管理访问权，如 第 330 页的『授予对队列管理器的完全管理访问权』 中所述
进程	授予对所需流程的完全管理访问权，如 第 331 页的『授予对某些进程的完全管理访问权』 中所述
名称列表	授予对所需名称列表的完全管理访问权，如 第 332 页的『授予对某些名称列表的完全管理访问权』 中所述
服务	授予对所需服务的完全管理访问权，如 第 332 页的『授予对某些服务的完全管理访问权』 中所述

授予对某些队列的完全管理访问权

将队列管理器上某些队列的完整管理访问权授予具有业务需求的每组用户。

关于此任务

要授予对某些队列的完全管理访问权，请对操作系统使用相应的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW
  在 AIX, Linux, and Windows 上:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +alladm
```

- IBM i
  在 IBM i 上:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*ALLADM) MQMNAME(' QMgrName
')
```

- z/OS
  在 z/OS 上:

```
RDEFINE MQADMIN QMgrName.QUEUE. ObjectProfile UACC(NONE)
PERMIT QMgrName.QUEUE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

变量名称具有以下含义：

QMgrName
队列管理器的名称。

z/OS 在 z/OS 上，此值也可以是队列共享组的名称。

ObjectProfile
要更改其权限的对象或通用概要文件的名称。

GroupName
要授予访问权的组的名称。

授予对某些主题的完全管理访问权

将队列管理器上某些主题的完整管理访问权授予具有业务需求的每组用户。

关于此任务

要授予对某些操作的某些主题的完全管理访问权，请针对您的操作系统使用相应的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW
  在 AIX, Linux, and Windows 上：

  ```
  setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +alladm
  ```

- IBM i
  在 IBM i 上：

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS
  在 z/OS 上：

  ```
  RDEFINE MQADMIN QMgrName.TOPIC. ObjectProfile UACC(NONE)
  PERMIT QMgrName.TOPIC. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  变量名称具有以下含义：

  QMgrName
  队列管理器的名称。

  z/OS 在 z/OS 上，此值也可以是队列共享组的名称。

  ObjectProfile
  要更改其权限的对象或通用概要文件的名称。

  GroupName
  要授予访问权的组的名称。

授予对某些通道的完全管理访问权

将对队列管理器上某些通道的完全管理访问权授予具有业务需求的每组用户。

关于此任务

要授予对某些通道的完全管理访问权，请针对您的操作系统使用相应的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW

  在 AIX, Linux, and Windows 上：

  ```
  setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName +alladm
  ```

- IBM i

  在 IBM i 上：

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*CHL) USER(GroupName) AUT(ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS

  在 z/OS 上：

  ```
  RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
  PERMIT QMgrName.CHANNEL. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  变量名称具有以下含义：

  QMgrName
  队列管理器的名称。

  z/OS 在 z/OS 上，此值也可以是队列共享组的名称。

  ObjectProfile
  要更改其权限的对象或通用概要文件的名称。

  GroupName
  要授予访问权的组的名称。

授予对队列管理器的完全管理访问权

将队列管理器的完全管理访问权授予具有业务需求的每组用户。

关于此任务

要授予对队列管理器的完全管理访问权，请针对您的操作系统使用相应的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW

  在 AIX, Linux, and Windows 上：

  ```
  setmqaut -m QMgrName -t qmgr -g GroupName +alladm
  ```

- IBM i

  在 IBM i 上：

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(*ALLADM) MQMNAME('
QMgrName ')
```

- z/OS
  在 z/OS 上：

```
RDEFINE MQADMIN QMgrName.QMGR UACC(NONE)
PERMIT QMgrName.QMGR CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

变量名称具有以下含义：

QMgrName
队列管理器的名称。

z/OS 在 z/OS 上，此值也可以是队列共享组的名称。

ObjectProfile
要更改其权限的对象或通用概要文件的名称。

GroupName
要授予访问权的组的名称。

授予对某些进程的完全管理访问权

将队列管理器上某些进程的完全管理访问权授予具有业务需求的每组用户。

关于此任务

要授予对某些进程的完全管理访问权，请对操作系统使用相应的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW
  在 AIX, Linux, and Windows 上：

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +alladm
```

- IBM i
  在 IBM i 上：

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(*ALLADM) MQMNAME('
QMgrName ')
```

- z/OS
  在 z/OS 上：

```
RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
PERMIT QMgrName.PROCESS. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

变量名称具有以下含义：

QMgrName
队列管理器的名称。

z/OS 在 z/OS 上，此值也可以是队列共享组的名称。

ObjectProfile
要更改其权限的对象或通用概要文件的名称。

GroupName
要授予访问权的组的名称。

授予对某些名称列表的完全管理访问权

将队列管理器上某些名称列表的完整管理访问权授予具有业务需求的每组用户。

关于此任务

要授予对某些名称列表的完全管理访问权，请使用适用于您的操作系统的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW
  在 AIX, Linux, and Windows 上：

```
setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName +alladm
```

- IBM i
  在 IBM i 上：

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALLADM) MQMNAME('
QMgrName ')
```

- z/OS
  在 z/OS 上：

```
RDEFINE MQADMIN QMgrName.NAMELIST. ObjectProfile UACC(NONE)
PERMIT QMgrName.NAMELIST. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  变量名称具有以下含义：

  QMgrName
  队列管理器的名称。

  z/OS 在 z/OS 上，此值也可以是队列共享组的名称。

  ObjectProfile
  要更改其权限的对象或通用概要文件的名称。

  GroupName
  要授予访问权的组的名称。

授予对某些服务的完全管理访问权

将队列管理器上某些服务的完全管理访问权授予具有业务需求的每组用户。

关于此任务

要授予对某些服务的完全管理访问权，请对操作系统使用相应的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW
  在 AIX, Linux, and Windows 上：

```
setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName +alladm
```

- IBM i
  在 IBM i 上：

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*SVC) USER(GroupName) AUT(*ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS
  在 z/OS 上：

  ```
  RDEFINE MQADMIN QMgrName.SERVICE. ObjectProfile UACC(NONE)
  PERMIT QMgrName.SERVICE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  变量名称具有以下含义：

  QMgrName
  队列管理器的名称。

  z/OS 在 z/OS 上，此值也可以是队列共享组的名称。

  ObjectProfile
  要更改其权限的对象或通用概要文件的名称。

  GroupName
  要授予访问权的组的名称。

授予对队列管理器上所有资源的只读访问权

将队列管理器上所有资源的只读访问权授予具有业务需求的每个用户或用户组。

关于此任务

使用 "添加基于角色的权限" 向导或适用于您操作系统的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

更改任何授权详细信息后，使用 REFRESH SECURITY 命令执行安全性刷新。

过程

- 使用向导：
  a) 在 " IBM MQ Explorer Navigator " 窗格中，右键单击队列管理器，然后单击 **对象权限 > 添加基于角色的权限**
  将打开 "添加基于角色的权限" 向导。

- ALW
  对于 AIX, Linux, and Windows 系统，请发出以下命令：

  ```
  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
  ```

仅当您要使用 IBM MQ Explorer 时，才需要 SYSTEM.ADMIN.COMMAND.QUEUE 和 SYSTEM.MQEXPLORER.REPLY.MODEL 的特定权限。

- IBM i

  对于 IBM i，请发出以下命令:

```
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

  对于 z/OS，请发出以下命令:

```
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)
```

变量名称具有以下含义:

QMgrName
队列管理器的名称。

z/OS 在 z/OS 上，此值也可以是队列共享组的名称。

GroupName
要授予访问权的组的名称。

授予对队列管理器上所有资源的完全管理访问权

向每个有业务需求的用户或用户组授予对队列管理器上所有资源的完全管理访问权。

关于此任务

您可以使用 "添加基于角色的权限" 向导或适用于您操作系统的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

注意: ALW

1. 如果要使用 **runmqsc** 而不是 IBM MQ Explorer 来管理队列管理器，那么必须授予查询，获取和浏览 SYSTEM.MQSC.REPLY.QUEUE，您不需要授予 SYSTEM.MQEXPLORER.REPLY.MODEL 队列。
2. 当授予用户对队列管理器上所有资源的访问权时，存在一些用户无法运行的命令，除非该用户具有对 `qm.ini` 文件的读访问权。 这是由于对非 `mqm` 用户能够读取 `qm.ini` 文件的限制。

   除非您授予用户对 qm.ini 文件的读访问权，否则用户无法发出以下命令:

- 定义配置为使用 TLS 的通道
- 使用 qm.ini 中定义的自动配置插入变量来定义通道

过程

- 如果您正在使用该向导，请在 " IBM MQ Explorer Navigator " 窗格中，右键单击队列管理器，然后单击 **对象权限 > 添加基于角色的权限**。
  将打开 "添加基于角色的权限" 向导。
- Linux AIX
  对于 AIX and Linux 系统，请发出以下命令:

```
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
```

  请参阅 **setmqaut**，以获取有关 @class 的更多信息
- Windows
  对于 Windows 系统，发出与 AIX and Linux 系统相同的命令，但使用概要文件名称 @CLASS 而不是 @class。
- IBM i
  对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*ALL) USER(' GroupName ') AUT(*ALLADM) MQMNAME(' QMgrName ')
```

- z/OS
  对于 z/OS，请发出以下命令:

```
RDEFINE MQADMIN QMgrName.*.** UACC(NONE)
PERMIT QMgrName.*.** CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  变量名称具有以下含义:

  QMgrName
  队列管理器的名称。

  z/OS 在 z/OS 上，此值也可以是队列共享组的名称。

  GroupName
  要授予访问权的组的名称。

除去与队列管理器的连接

如果您不希望用户应用程序连接到队列管理器，请除去其连接到队列管理器的权限。

关于此任务

使用适用于您的操作系统的相应命令，撤销所有用户连接到队列管理器的权限。

在 多平台上，还可以使用 DELETE AUTHREC 命令。

注: 在 IBM MQ Appliance 上，只能使用 **DELETE AUTHREC** 命令。

过程

- ALW
  对于 AIX, Linux, and Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -t qmgr -g GroupName -connect
```

- IBM i
  对于 IBM i，请发出以下命令:

```
RVKMQMAUT OBJ ('QMgrName') OBJTYPE(*MQM) USER(*ALL) AUT(*CONNECT)
```

- z/OS
  对于 z/OS，请发出以下命令:

```
RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
RDEFINE MQCONN QMgrName.CHIN UACC(NONE)
RDEFINE MQCONN QMgrName.CICS UACC(NONE)
RDEFINE MQCONN QMgrName.IMS UACC(NONE)
```

  请勿发出任何 PERMIT 命令。

  变量名称具有以下含义:

  QMgrName
  队列管理器的名称。

  z/OS 在 z/OS 上，此值也可以是队列共享组的名称。

  GroupName
  要拒绝访问的组的名称。

允许用户应用程序连接到队列管理器

您希望允许用户应用程序连接到队列管理器。 使用本主题中的表来确定要执行的操作。

首先，确定客户机应用程序是否将连接到队列管理器。

如果将连接到队列管理器的应用程序都不是客户机应用程序，请按 第 343 页的『禁用对队列管理器的远程访问』中所述禁用远程访问。

如果将连接到队列管理器的一个或多个应用程序是客户机应用程序，请按 第 337 页的『保护与队列管理器的远程连接』中所述保护远程连接。

在这两种情况下，设置连接安全性，如 第 343 页的『设置连接安全性』 中所述

如果要控制连接到队列管理器的每个用户对资源的访问权，请参阅下表。 如果第一列中的语句为 true ，请执行第二列中列出的操作。

语句	执行此操作
您有使用队列的应用程序	请参阅 第 344 页的『控制用户对队列的访问权』
您有使用主题的应用程序	请参阅第 349 页的『控制用户对主题的访问权』。
您具有查询队列管理器对象的应用程序	请参阅第 351 页的『授予对队列管理器进行查询的权限』。
您有使用流程对象的应用程序	请参阅 第 351 页的『授予访问进程的权限』

语句	执行此操作
您有使用名称列表的应用程序	请参阅 第 352 页的『授予访问名称列表的权限』

保护与队列管理器的远程连接

您可以使用 TLS，安全出口，通道认证记录或这些方法的组合来保护与队列管理器的远程连接。

关于此任务

您可以使用客户机工作站上的客户机连接通道和服务器上的服务器连接通道将客户机连接到队列管理器。 通过下列其中一种方式保护此类连接。

过程

1. 将 TLS 与通道认证记录配合使用:
   a) 通过使用 SSLPEERMAP 通道认证记录将所有 DN 映射到 USERSRC (NOACCESS)，防止任何专有名称 (DN) 打开通道。
   b) 允许特定 DN 或 DN 集通过使用 SSLPEERMAP 通道认证记录将其映射到 USERSRC (CHANNEL) 来打开通道。
2. 将 TLS 与安全出口配合使用:
   a) 将服务器连接通道上的 MCAUSER 设置为没有特权的用户标识。
   b) 编写安全出口以分配 MCAUSER 值，具体取决于它在 SSLPeerNamePtr 和 SSLPeerName 长度字段中接收到的 TLS DN 的值，这些字段传递到 MQCD 结构中的出口。
3. 将 TLS 与固定通道定义值配合使用:
   a) 将服务器连接通道上的 SSLPEER 设置为特定值或范围较窄的值。
   b) 将服务器连接通道上的 MCAUSER 设置为运行通道应使用的用户标识。
4. 在不使用 TLS 的通道上使用通道认证记录:
   a) 通过使用具有 ADDRESS (*) 和 USERSRC (NOACCESS) 的地址映射通道认证记录，阻止任何 IP 地址打开通道。
   b) 通过使用具有 USERSRC (CHANNEL) 的地址映射通道认证记录，允许特定 IP 地址打开通道。
5. 使用安全出口:
   a) 根据您选择的任何属性 (例如，起始 IP 地址)，编写安全出口以授权连接。
6. 如果您的特定情况需要，还可以将通道认证记录与安全出口配合使用，或者使用所有这三种方法。

阻止特定 *IP* 地址

您可以通过使用通道认证记录来阻止特定通道接受来自 IP 地址的入站连接，或者阻止整个队列管理器允许从 IP 地址进行访问。

开始之前

通过运行以下命令来启用通道认证记录:

```
ALTER QMGR CHLAUTH(ENABLED)
```

关于此任务

要禁止特定通道接受入站连接并确保仅当使用正确的通道名称时才接受连接，可以使用一种类型的规则来阻止 IP 地址。 要禁止对整个队列管理器进行 IP 地址访问，通常会使用防火墙将其永久阻塞。 但是，可以使用另一种类型的规则来允许您临时阻止一些地址，例如，当您正在等待更新防火墙时。

过程

- 要阻止 IP 地址使用特定通道，请使用 MQSC 命令 **SET CHLAUTH** 或 PCF 命令 **Set Channel Authentication Record** 设置通道认证记录。

```
SET CHLAUTH(generic-channel-name) TYPE(ADDRESSMAP) ADDRESS(generic-ip-address)
USERSRC(NOACCESS)
```

该命令有三个部分:

SET CHLAUTH (*generic-channel-name*)
您可以使用该命令的此部分来控制是否要阻止整个队列管理器，单个通道或通道范围的连接。 您在此处放入的内容将确定所涵盖的区域。

例如:

- SET CHLAUTH('*') -阻止队列管理器上的每个通道，即整个队列管理器
- SET CHLAUTH ('SYSTEM. *')-阻塞以 SYSTEM 开头的每个通道。
- SET CHLAUTH ('SYSTEM.DEF.SVRCONN')-阻止通道 SYSTEM.DEF.SVRCONN

CHLAUTH 规则的类型
使用该命令的此部分来指定命令类型，并确定是要提供单个地址还是地址列表。

例如:

- TYPE(ADDRESSMAP) -如果要提供单个地址或通配符地址，请使用 ADDRESSMAP。 例如，ADDRESS('192.168.*') 会阻止来自从 192.168 开始的 IP 地址的任何连接。

  有关使用模式过滤 IP 地址的更多信息，请参阅 通用 IP 地址。
- TYPE(BLOCKADDR) -如果要提供要阻止的地址列表，请使用 BLOCKADDR。

其他参数
这些参数取决于您在命令的第二部分中使用的规则类型:

- 对于 TYPE(ADDRESSMAP)，使用 ADDRESS
- 对于 TYPE(BLOCKADDR)，使用 ADDRLIST

相关参考
SET CHLAUTH

如果队列管理器未在运行，那么临时阻止特定 *IP* 地址
当队列管理器未运行并且因此无法发出 MQSC 命令时，您可能想要阻止特定 IP 地址或地址范围。 您可以通过修改 blockaddr.ini 文件来例外地临时阻止 IP 地址。

关于此任务

blockaddr.ini 文件包含队列管理器所使用的 BLOCKADDR 定义的副本。 如果侦听器在队列管理器之前启动，那么侦听器将读取此文件。 在这些情况下，侦听器将使用您手动添加到 blockaddr.ini 文件的任何值。

但是，请注意，当启动队列管理器时，它会将 BLOCKADDR 定义集写入 blockaddr.ini 文件，从而覆盖您可能已完成的任何手动编辑。 同样，每次使用 **SET CHLAUTH** 命令添加或删除 BLOCKADDR 定义时，都会更新 blockaddr.ini 文件。 因此，仅当队列管理器正在运行时，才能使用 **SET CHLAUTH** 命令对 BLOCKADDR 定义进行永久更改。

过程

1. 在文本编辑器中打开 blockaddr.ini 文件。
   该文件位于队列管理器的数据目录中。
2. 将 IP 地址添加为简单的 "关键字/值" 对，其中关键字为 Addr。
   有关使用模式过滤 IP 地址的信息，请参阅 通用 IP 地址。
   例如:

```
Addr = 192.0.2.0
Addr = 192.0.*
Addr = 192.0.2.1-8
```

相关任务

第 337 页的『阻止特定 IP 地址』

您可以通过使用通道认证记录来阻止特定通道接受来自 IP 地址的入站连接，或者阻止整个队列管理器允许从 IP 地址进行访问。

相关参考

SET CHLAUTH

阻止特定用户标识

您可以通过指定用户标识 (如果已断言) 来阻止特定用户使用通道，从而导致通道结束。 通过设置通道认证记录来执行此操作。

开始之前

确保按如下所示启用通道认证记录:

```
ALTER QMGR CHLAUTH(ENABLED)
```

过程

使用 MQSC 命令 **SET CHLAUTH** 或 PCF 命令 **Set Channel Authentication Record** 设置通道认证记录。 例如，可以发出 MQSC 命令:

```
SET CHLAUTH(' generic-channel-name ') TYPE(BLOCKUSER) USERLIST(userID1, userID2)
```

generic-channel-name 是要控制其访问权的通道的名称，或者是包含星号 (*) 作为通配符的模式，与通道名称匹配。

TYPE（BLOCKUSER）上提供的用户列表仅适用于 SVRCONN 通道，而不适用于队列管理器通道的队列管理器。

userID1 和 *userID2* 都是要阻止使用通道的用户的标识。 您还可以指定特殊值 *MQADMIN 以引用特权管理用户。 有关特权用户的更多信息，请参阅 第 283 页的『特权用户』。 有关 *MQADMIN 的更多信息，请参阅 SET CHLAUTH。

相关参考

SET CHLAUTH

将远程队列管理器映射到 *MCAUSER* 用户标识

根据通道所连接的队列管理器，可以使用通道认证记录来设置通道的 MCAUSER 属性。

开始之前

确保按如下所示启用通道认证记录:

```
ALTER QMGR CHLAUTH(ENABLED)
```

关于此任务

(可选) 您可以限制规则适用的 IP 地址。

请注意，此方法不适用于服务器连接通道。 如果在以下命令中指定服务器连接通道的名称，那么不会产生任何影响。

过程

- 使用 MQSC 命令 **SET CHLAUTH** 或 PCF 命令 **Set Channel Authentication Record** 设置通道认证记录。 例如，可以发出 MQSC 命令:

```
SET CHLAUTH(' generic-channel-name ') TYPE (QMGRMAP) QMNAME(generic-partner-qmgr-name
) USERSRC(MAP) MCAUSER(user)
```

generic-channel-name 是要控制其访问权的通道的名称，或者是包含星号 (*) 作为通配符的模式，与通道名称匹配。

generic-partner-qmgr-name 是队列管理器的名称，或者是包含星号 (*) 符号作为与队列管理器名称匹配的通配符的模式。

user 是要用于来自指定队列管理器的所有连接的用户标识。

- 要将此命令限制为某些 IP 地址，请包括 **ADDRESS** 参数，如下所示:

```
SET CHLAUTH(' generic-channel-name ') TYPE (QMGRMAP) QMNAME(generic-partner-qmgr-name
) USERSRC(MAP) MCAUSER(user) ADDRESS(
generic-ip-address)
```

generic-channel-name 是要控制其访问权的通道的名称，或者是包含星号 (*) 作为通配符的模式，与通道名称匹配。

generic-ip-address 是单个地址，或者是包含星号 (*) 作为通配符的模式，或者是表示与地址匹配的范围的连字符 (-)。 有关通用 IP 地址的更多信息，请参阅 通用 IP 地址。

相关参考

SET CHLAUTH

将客户机用户标识映射到 *MCAUSER* 用户标识

可以使用通道认证记录根据从客户机接收到的用户标识来更改服务器连接通道的 MCAUSER 属性。

开始之前

确保按如下所示启用通道认证记录:

```
ALTER QMGR CHLAUTH(ENABLED)
```

关于此任务

请注意，此方法仅适用于服务器连接通道。 它对其他通道类型没有影响。

过程

使用 MQSC 命令 **SET CHLAUTH** 或 PCF 命令 **Set Channel Authentication Record** 设置通道认证记录。 例如，可以发出 MQSC 命令:

```
SET CHLAUTH(' generic-channel-name ') TYPE (USERMAP) CLNTUSER(client-user-name) USERSRC(MAP)
MCAUSER(
user)
```

generic-channel-name 是要控制其访问权的通道的名称，或者是包含星号 (*) 作为通配符的模式，与通道名称匹配。

client-user-name 是与客户机连接关联的用户标识，该值可由客户机应用程序声明，并通过使用早期采用或通过通道出口设置的连接认证进行更改。

user 是要使用的用户标识，而不是客户机用户名。

相关参考

SET CHLAUTH

通道节的属性 (ChlauthEarly 采用)

将 *SSL* 或 *TLS* 专有名称映射到 *MCAUSER* 用户标识

根据接收到的专有名称 (DN) ，可以使用通道认证记录来设置通道的 MCAUSER 属性。

开始之前

确保按如下所示启用通道认证记录:

```
ALTER QMGR CHLAUTH(ENABLED)
```

过程

使用 MQSC 命令 **SET CHLAUTH** 或 PCF 命令 **Set Channel Authentication Record** 设置通道认证记录。 例如，可以发出 MQSC 命令:

```
SET CHLAUTH('generic-channel-name') TYPE (SSLPEERMAP)
SSLPEER(generic-ssl-peer-name) SSLCERTI(generic-issuer-name)
USERSRC(MAP) MCAUSER(user)
```

generic-channel-name 是要控制其访问权的通道的名称，或者是包含星号 (*) 作为通配符的模式，与通道名称匹配。

generic-ssl-peer-name 是遵循 SSLPEER 值的标准 IBM MQ 规则的字符串。 请参阅 SSLPEER 值的 IBM MQ 规则。

user 是要用于使用指定 DN 的所有连接的用户标识。

generic-issuer-name 是指要匹配的证书的签发者 DN。 此参数是可选的，但您应该使用此参数，以避免在使用多个认证中心时与错误的证书进行虚假匹配。

相关参考

SET CHLAUTH

阻止来自远程队列管理器的访问

您可以使用通道认证记录来阻止远程队列管理器启动通道。

开始之前

确保按如下所示启用通道认证记录:

```
ALTER QMGR CHLAUTH(ENABLED)
```

关于此任务

请注意，此方法不适用于服务器连接通道。 如果在以下命令中指定服务器连接通道的名称，那么不会产生任何影响。

过程

使用 MQSC 命令 **SET CHLAUTH** 或 PCF 命令 **Set Channel Authentication Record** 设置通道认证记录。 例如，可以发出 MQSC 命令:

```
SET CHLAUTH(' generic-channel-name ') TYPE(QMGRMAP) QMNAME(' generic-partner-qmgr-name ')
USERSRC(NOACCESS)
```

generic-channel-name 是要控制其访问权的通道的名称，或者是包含星号 (*) 作为通配符的模式，与通道名称匹配。

generic-partner-qmgr-name 是队列管理器的名称，或者是包含星号 (*) 符号作为与队列管理器名称匹配的通配符的模式。

相关参考

SET CHLAUTH

阻止访问客户机用户标识

您可以使用通道认证记录来阻止客户机用户标识建立通道连接。

开始之前

确保按如下所示启用通道认证记录:

```
ALTER QMGR CHLAUTH(ENABLED)
```

关于此任务

请注意，此方法仅适用于服务器连接通道。 它对其他通道类型没有影响。

过程

使用 MQSC 命令 **SET CHLAUTH** 或 PCF 命令 **Set Channel Authentication Record** 设置通道认证记录。 例如，可以发出 MQSC 命令:

```
SET CHLAUTH(' generic-channel-name ') TYPE(USERMAP) CLNTUSER(' client-user-name ')
USERSRC(NOACCESS)
```

generic-channel-name 是要控制其访问权的通道的名称，或者是包含星号 (*) 作为通配符的模式，与通道名称匹配。

client-user-name 是与客户机连接关联的用户标识，该值可由客户机应用程序声明，并通过使用早期采用或通过通道出口设置的连接认证进行更改。

相关参考

SET CHLAUTH

阻止 *SSL* 或 *TLS* 专有名称的访问

您可以使用通道认证记录来阻止 TLS 专有名称 (DN) 启动通道。

开始之前

确保按如下所示启用通道认证记录:

```
ALTER QMGR CHLAUTH(ENABLED)
```

过程

使用 MQSC 命令 **SET CHLAUTH** 或 PCF 命令 **Set Channel Authentication Record** 设置通道认证记录。 例如，可以发出 MQSC 命令:

```
SET CHLAUTH('generic-channel-name') TYPE(SSLPEERMAP)
SSLPEER('generic-ssl-peer-name') SSLCERTI(generic-issuer-name)
USERSRC(NOACCESS)
```

generic-channel-name 是要控制其访问权的通道的名称，或者是包含星号 (*) 作为通配符的模式，与通道名称匹配。

generic-ssl-peer-name 是遵循 SSLPEER 值的标准 IBM MQ 规则的字符串。 请参阅 SSLPEER 值的 IBM MQ 规则。

generic-issuer-name 是指要匹配的证书的签发者 DN。 此参数是可选的，但您应该使用此参数，以避免在使用多个认证中心时与错误的证书进行虚假匹配。

相关参考

SET CHLAUTH

将 *IP* 地址映射到 *MCAUSER* 用户标识

您可以使用通道认证记录根据从中接收连接的 IP 地址来设置通道的 MCAUSER 属性。

开始之前

确保按如下所示启用通道认证记录:

```
ALTER QMGR CHLAUTH(ENABLED)
```

过程

使用 MQSC 命令 **SET CHLAUTH** 或 PCF 命令 **Set Channel Authentication Record** 设置通道认证记录。 例如，可以发出 MQSC 命令:

```
SET CHLAUTH(' generic-channel-name ') TYPE(ADDRESSMAP) ADDRESS(' generic-ip-address ')
USERSRC(MAP) MCAUSER(user)
```

generic-channel-name 是要控制其访问权的通道的名称，或者是包含星号 (*) 作为通配符的模式，与通道名称匹配。

user 是要用于使用指定 DN 的所有连接的用户标识。

generic-ip-address 是从中建立连接的地址，或者包含星号 (*) 作为通配符或连字符 (-) 以指示与地址匹配的范围的模式。

相关参考

SET CHLAUTH

禁用对队列管理器的远程访问

如果您不希望客户机应用程序连接到队列管理器，请禁用对其的远程访问。

关于此任务

通过下列其中一种方式阻止客户机应用程序连接到队列管理器:

过程

- 使用 MQSC 命令 **DELETE CHANNEL** 删除所有服务器连接通道。
- 使用 MQSC 命令 **ALTER CHANNEL** 将通道的消息通道代理程序用户标识 (MCAUSER) 设置为没有访问权的用户标识。

设置连接安全性

将连接到队列管理器的权限授予每个有业务需要的用户或用户组。

关于此任务

要设置连接安全性，请针对操作系统使用相应的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW
  在 AIX, Linux, and Windows 上:

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

- IBM i
  在 IBM i 上:

  ```
  GRTMQMAUT OBJ('QMgrName') OBJTYPE(*MQM) USER('GroupName') AUT(*CONNECT)
  ```

- z/OS
  在 z/OS 上：

```
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)
```

  这些命令授予针对批处理， CICS， IMS 和通道启动程序 (CHIN) 进行连接的权限。 如果不使用特定类型的连接，请省略相关命令。

  变量名称具有以下含义：

  QMgrName
  队列管理器的名称。 在 z/OS 上，此值也可以是队列共享组的名称。

  ObjectProfile
  要更改其权限的对象或通用概要文件的名称。

  GroupName
  要授予访问权的组的名称。

相关概念

第 174 页的『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.

控制用户对队列的访问权

您希望控制应用程序对队列的访问权。 使用本主题来确定要执行的操作。

对于第一列中的每个 true 语句，执行第二列中指示的操作。

语句	操作
应用程序从队列获取消息	请参阅 第 344 页的『授予从队列获取消息的权限』
应用程序集上下文	请参阅 第 345 页的『授予设置上下文的权限』
应用程序传递上下文	请参阅 第 346 页的『授予传递上下文的权限』
应用程序将消息放在集群队列上	请参阅 第 420 页的『授权将消息放入远程集群队列』
应用程序将消息放入本地队列	请参阅 第 347 页的『授予将消息放入本地队列的权限』
应用程序将消息放入模型队列	请参阅 第 348 页的『授予将消息放入模型队列的权限』
应用程序将消息放在远程队列上	请参阅 第 348 页的『授予将消息放入远程集群队列的权限』

授予从队列获取消息的权限
将从队列或队列集获取消息的权限授予具有业务需求的每组用户。

关于此任务

要授予从某些队列获取消息的权限，请使用适用于您的操作系统的相应命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- Windows
  对于 AIX, Linux, and Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +get
```

- IBM i
  对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*GET) MQMNAME(' QMgrName ')
```

- z/OS
  对于 z/OS，请发出以下命令:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(UPDATE)
```

  变量名称具有以下含义:

  QMgrName
  队列管理器的名称。 在 z/OS 上，此值也可以是队列共享组的名称。

  ObjectProfile
  要更改其权限的对象或通用概要文件的名称。

  GroupName
  要授予访问权的组的名称。

授予设置上下文的权限
向具有业务需求的每组用户授予对要放入的消息设置上下文的权限。

关于此任务

要授予在某些队列上设置上下文的权限，请对操作系统使用相应的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW
  对于 AIX, Linux, and Windows 系统，请发出下列其中一个命令:
  - 要仅设置身份上下文:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +setid
```

  - 要设置所有上下文:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +setall
```

  注: 要使用 `setid` 或 `setall` 权限，必须对相应的队列对象以及队列管理器对象授予权限。

- IBM i
  对于 IBM i，发出下列其中一个命令:
  - 要仅设置身份上下文:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*SETID) MQMNAME('
QMgrName ')
```

- 要设置所有上下文:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*SETALL) MQMNAME('
QMgrName ')
```

- z/OS

对于 z/OS，发出下列其中一组命令:

- 要仅设置身份上下文:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

- 要设置所有上下文:

```
RDEFINE MQQUEUE QMgrName. ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(CONTROL)
```

变量名称具有以下含义:

QMgrName
队列管理器的名称。 在 z/OS 上，此值也可以是队列共享组的名称。

ObjectProfile
要更改其权限的对象或通用概要文件的名称。

GroupName
要授予访问权的组的名称。

授予传递上下文的权限
授予将上下文从检索到的消息传递到正在放置的消息的权限，以传递给具有业务需求的每组用户。

关于此任务

要授予在某些队列上传递上下文的权限，请使用适用于您的操作系统的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW

对于 AIX, Linux, and Windows 系统，请发出下列其中一个命令:

- 仅传递身份上下文:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +passid
```

- 要传递所有上下文:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +passall
```

- IBM i

对于 IBM i，发出下列其中一个命令:

- 仅传递身份上下文:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PASSID) MQMNAME('
QMgrName ')
```

- 要传递所有上下文:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PASSALL) MQMNAME('
QMgrName ')
```

- z/OS

对于 z/OS，发出以下命令以传递身份上下文或所有上下文:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

变量名称具有以下含义:

QMgrName
队列管理器的名称。 在 z/OS 上，此值也可以是队列共享组的名称。

ObjectProfile
要更改其权限的对象或通用概要文件的名称。

GroupName
要授予访问权的组的名称。

授予将消息放入本地队列的权限
将将消息放入本地队列或一组队列的权限授予具有业务需求的每组用户。

关于此任务

要授予将消息放入某些本地队列的权限，请对操作系统使用相应的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW

对于 AIX, Linux, and Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +put
```

- IBM i

对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
```

- z/OS

对于 z/OS，请发出以下命令:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

变量名称具有以下含义:

QMgrName
队列管理器的名称。 在 z/OS 上，此值也可以是队列共享组的名称。

ObjectProfile
要更改其权限的对象或通用概要文件的名称。

GroupName
要授予访问权的组的名称。

授予将消息放入模型队列的权限
向业务需要的每组用户授予将消息放入模型队列或模型队列集合的权限。

关于此任务

模型队列用于创建动态队列。 因此，必须授予对模型和动态队列的权限。 要授予这些权限，请对操作系统使用相应的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW
  对于 AIX, Linux, and Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ModelQueueName -t queue -g GroupName +put
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +put
```

- IBM i
  对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ(' ModelQueueName ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
```

- z/OS
  对于 z/OS，请发出以下命令:

```
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)
```

  变量名称具有以下含义:

  QMgrName
  队列管理器的名称。 在 z/OS 上，此值也可以是队列共享组的名称。

  modelQueueName
  动态队列所基于的模型队列的名称。

  ObjectProfile
  要更改其权限的动态队列或通用概要文件的名称。

  GroupName
  要授予访问权的组的名称。

授予将消息放入远程集群队列的权限
将将消息放入远程集群队列或一组队列的权限授予具有业务需求的每组用户。

关于此任务

要将消息放在远程集群队列上，可以将其放在远程队列的本地定义上，也可以放在标准远程队列上。 如果您正在使用远程队列的本地定义，那么需要权限才能放入本地对象: 请参阅 第 347 页的『授予将消息放入本地队列的权限』。 如果您正在使用标准远程队列，那么需要权限才能将其放入远程队列。 使用适用于您的操作系统的相应命令授予此权限。

缺省行为是对 SYSTEM.CLUSTER.TRANSMIT.QUEUE 执行访问控制。 请注意，即使您正在使用多个传输队列，此行为也适用。

仅当您在 qm.ini 文件中将 **ClusterQueueAccessControl** 属性配置为 *RQMName*(如 安全性节 主题中所述) 并重新启动队列管理器时，本主题中描述的特定行为才适用。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW
  对于 AIX, Linux, and Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -t rqmname -n
ObjectProfile -g GroupName +put
```

  请注意，只能将 *rqmname* 对象用于远程集群队列。

- IBM i
  对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJTYPE(*RMTMQMNAME) OBJ('
ObjectProfile') USER(GroupName) AUT(*PUT) MQMNAME('
QMgrName')
```

  请注意，只能将 RMTMQMNAME 对象用于远程集群队列。

- z/OS
  对于 z/OS，请发出以下命令:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

  请注意，只能将远程队列管理器 (或队列共享组) 的名称用于远程集群队列。

  变量名称具有以下含义:

  QMgrName
  队列管理器的名称。 在 z/OS 上，此值也可以是队列共享组的名称。

  ObjectProfile
  要更改其权限的远程队列管理器或通用概要文件的名称。

  GroupName
  要授予访问权的组的名称。

控制用户对主题的访问权

您需要控制应用程序对主题的访问权。 使用本主题来确定要执行的操作。

对于第一列中的每个 true 语句，执行第二列中指示的操作。

表 *74:* 控制用户对主题的访问权

语句	**操作**
应用程序将消息发布到主题	请参阅 第 349 页的『授予向主题发布消息的权限』
应用程序预订主题	请参阅 第 350 页的『授予预订主题的权限』

授予向主题发布消息的权限

将将消息发布到主题或主题集的权限授予具有业务需求的每组用户。

关于此任务

要授予将消息发布到某些主题的权限，请使用适用于您的操作系统的相应命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW
  对于 AIX, Linux, and Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +pub
```

- IBM i
  对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(*PUB) MQMNAME('
QMgrName ')
```

- z/OS
  对于 z/OS，请发出以下命令:

```
RDEFINE MQTOPIC QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQTOPIC) ID(GroupName) ACCESS(UPDATE)
```

  变量名称具有以下含义:

  QMgrName
  队列管理器的名称。 在 z/OS 上，此值也可以是队列共享组的名称。

  ObjectProfile
  要更改其权限的对象或通用概要文件的名称。

  GroupName
  要授予访问权的组的名称。

授予预订主题的权限
将预订主题或主题集的权限授予具有业务需求的每组用户。

关于此任务

要授予预订某些主题的权限，请针对您的操作系统使用相应的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW
  对于 AIX, Linux, and Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +sub
```

- IBM i
  对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(*SUB) MQMNAME('
QMgrName ')
```

- z/OS
  对于 z/OS，请发出以下命令:

```
RDEFINE MQTOPIC QMgrName.SUBSCRIBE.ObjectProfile UACC(NONE)
PERMIT QMgrName.SUBSCRIBE.ObjectProfile CLASS(MQTOPIC) ID(GroupName) ACCESS(UPDATE)
```

变量名称具有以下含义:

QMgrName
队列管理器的名称。 在 z/OS 上，此值也可以是队列共享组的名称。

ObjectProfile
要更改其权限的对象或通用概要文件的名称。

GroupName
要授予访问权的组的名称。

授予对队列管理器进行查询的权限

将查询队列管理器的权限授予具有业务需求的每组用户。

关于此任务

要授予对队列管理器进行查询的权限，请对操作系统使用相应的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW
  对于 AIX, Linux, and Windows 系统，请发出以下命令:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName +inq
  ```

- IBM i
  对于 IBM i，请发出以下命令:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(*INQ) MQMNAME(' QMgrName
  ')
  ```

- z/OS
  对于 z/OS，请发出以下命令:

  ```
  RDEFINE MQCMDS QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  这些命令授予对指定队列管理器的访问权。 要允许用户使用 MQINQ 命令，请发出以下命令:

  ```
  RDEFINE MQCMDS QMgrName.MQINQ.QMGR UACC(NONE)
  PERMIT QMgrName.MQINQ.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  变量名称具有以下含义:

  QMgrName
  队列管理器的名称。 在 z/OS 上，此值也可以是队列共享组的名称。

  ObjectProfile
  要更改其权限的对象或通用概要文件的名称。

  GroupName
  要授予访问权的组的名称。

授予访问进程的权限

将访问流程或流程集的权限授予具有业务需求的每组用户。

关于此任务

要授予访问某些进程的权限，请对操作系统使用相应的命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW

  对于 AIX, Linux, and Windows 系统，请发出以下命令:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +all
  ```

- IBM i

  对于 IBM i，请发出以下命令:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(*ALL) MQMNAME(' QMgrName
  ')
  ```

- z/OS

  对于 z/OS，请发出以下命令:

  ```
  RDEFINE MQPROC QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQPROC) ID(GroupName) ACCESS(READ)
  ```

  变量名称具有以下含义:

  QMgrName
  队列管理器的名称。 在 z/OS 上，此值也可以是队列共享组的名称。

  ObjectProfile
  要更改其权限的对象或通用概要文件的名称。

  GroupName
  要授予访问权的组的名称。

授予访问名称列表的权限

将访问名称列表或名称列表集的权限授予具有业务需求的每组用户。

关于此任务

要授予访问某些名称列表的权限，请使用适用于您的操作系统的相应命令。

Multi 在 Multiplatforms 版上，还可以使用 SET AUTHREC 命令。

过程

- ALW

  对于 AIX, Linux, and Windows 系统，请发出以下命令:

  ```
  setmqaut -m QMgrName -n
  ObjectProfile -t namelist -g GroupName
  +all
  ```

- IBM i

  对于 IBM i，请发出以下命令:

  ```
  GRTMQMAUT OBJ('ObjectProfile
  ```

```
') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALL) MQMNAME('
QMgrName')
```

- z/OS

  对于 z/OS，请发出以下命令:

```
RDEFINE MQNLIST
QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile
CLASS(MQNLIST) ID(GroupName) ACCESS(READ)
```

  变量名称具有以下含义:

  QMgrName
  队列管理器的名称。 在 z/OS 上，此值也可以是队列共享组的名称。

  ObjectProfile
  要更改其权限的对象或通用概要文件的名称。

  GroupName
  要授予访问权的组的名称。

ALW 在 AIX, Linux, and Windows 上管理 IBM MQ 的权限

IBM MQ 管理员可以使用所有 IBM MQ 命令并授予其他用户权限。 当管理员向远程队列管理器发出命令时，他们必须在远程队列管理器上具有必需的权限。 进一步的注意事项适用于 Windows 系统。

IBM MQ 管理员有权使用所有 IBM MQ 命令 (包括用于向其他用户授予 IBM MQ 权限的命令)。

要成为 IBM MQ 管理员，您必须是名为 **mqm** 组的特殊组的成员。

Windows 或者，仅在 Windows 上，本地帐户可以管理 IBM MQ (如果它们是 Windows 系统上的 Administrators 组的成员)。

注意: 您可以使用管理员命令将 Azure AD 用户添加到 mqm 组。 例如，使用命令 `net localgroup mqm AzureAD\<your userID> /add`。 然后运行 IBM MQ 管理命令或使用 IBM MQ Explorer。

安装 IBM MQ 时，将自动创建 **mqm** 组。 您可以将更多用户添加到组以允许他们执行管理。 此组的所有成员都有权访问所有资源。 只能通过从 **mqm** 组中除去用户并发出 **REFRESH SECURITY** 命令来撤销此访问权。

管理员可以使用控制命令来管理 IBM MQ。 其中一个控制命令是 **setmqaut**，用于向其他用户授予访问或控制 IBM MQ 资源的权限。 用于管理权限记录的 PCF 命令可供在队列管理器上被授予 dsp 和 chg 权限的非管理员使用。 有关使用 PCF 命令管理权限的更多信息，请参阅 可编程命令格式。

管理员必须具有要由远程队列管理器处理的 MQSC 命令的必需权限。 IBM MQ Explorer 发出 PCF 命令以执行管理任务。 管理员无需其他权限即可使用 IBM MQ Explorer 来管理本地系统上的队列管理器。 当 IBM MQ Explorer 用于管理另一个系统上的队列管理器时，管理员必须具有远程队列管理器要处理的 PCF 命令所需的权限。

注意: 您不必是管理员才能使用发出 IBM MQ 脚本 (MQSC) 命令的控制命令 **runmqsc**。

当以间接方式使用 **runmqsc** 将 MQSC 命令发送到远程队列管理器时，每个 MQSC 命令都封装在 Escape PCF 命令中。

有关处理 PCF 和 MQSC 命令时的权限检查的更多信息，请参阅以下主题:

- 对于在队列管理器，队列，进程，名称列表和认证信息对象上运行的 PCF 命令，请参阅 使用 IBM MQ 对象的权限。 请参阅本节以了解封装在 Escape PCF 命令中的等效 MQSC 命令。
- 对于在通道，通道启动器，侦听器和集群上运行的 PCF 命令，请参阅 通道安全性。
- 对于对权限记录执行操作的 PCF 命令，请参阅 针对 PCF 命令的权限检查
- z/OS 对于由 IBM MQ for z/OS 上的命令服务器处理的 MQSC 命令，请参阅 z/OS 上的命令安全性和命令资源安全性 。

此外，在 Windows 系统上， SYSTEM 帐户具有对 IBM MQ 资源的完全访问权。

在 AIX and Linux 平台上，还会创建 **mqm** 的特殊用户标识，仅供产品使用。 它必须永远不可供非特权用户使用。 所有 IBM MQ 对象都由用户标识 **mqm** 拥有。

在 Windows 系统上， Administrators 组的成员还可以管理任何队列管理器，也可以管理 SYSTEM 帐户。 您还可以在域控制器上创建包含域中所有活动的特权用户标识的域 **mqm** 组，并将其添加到本地 **mqm** 组。 某些命令 (例如 **crtmqm**) 处理 IBM MQ 对象的权限，因此需要使用这些对象的权限 (如以下部分中所述)。 **mqm** 组的成员有权使用所有对象，但如果您具有本地用户和同名的域认证用户，那么在 Windows 系统上可能存在权限被拒绝的情况。 在第 356 页的『AIX, Linux, and Windows 上的主体和组』中对此进行了描述。

具有用户帐户控制 (UAC) 功能的 Windows 版本限制用户可以在特定操作系统工具上执行的操作，即使他们是 Administrators 组的成员也是如此。 如果您的用户标识位于 Administrators 组而不是 **mqm** 组中，那么必须使用提升的命令提示符来发出 IBM MQ 管理命令，例如 **crtmqm**，否则将生成错误 AMQ7077： **您无权执行请求的操作** 。 要打开提升的命令提示符，请右键单击命令提示符的开始菜单项或图标，然后选择 **以管理员身份运行**。

您无需是 **mqm** 组的成员即可执行以下操作:

- 从发出 PCF 命令的应用程序发出命令，或在 Escape PCF 命令中发出 MQSC 命令，除非这些命令处理通道启动程序。 (第 97 页的『保护通道启动程序定义』 中描述了这些命令)。
- 从应用程序发出 MQI 调用 (除非要在 MQCONNX 调用上使用快速路径绑定)。
- 使用 crtmqcvx 命令可创建对数据类型结构执行数据转换的代码片段。
- 使用 dspmq 命令显示队列管理器。
- 使用 dspmqtrc 命令显示 IBM MQ 格式化的跟踪输出。

12 个字符的限制同时适用于组标识和用户标识。

UNIX and Linux 平台通常将用户标识的长度限制为 12 个字符。 AIX 5.3 已提高此限制，但 IBM MQ 继续在所有 UNIX and Linux 平台上观察到 12 个字符的限制。 如果使用大于 12 个字符的用户标识，那么 IBM MQ 会将其替换为值 UNKNOWN。 请勿定义值为 UNKNOWN 的用户标识。

ALW 在 AIX, Linux, and Windows 上管理 mqm 组

mqm 组中的用户被授予基于 IBM MQ 的完整管理特权。 因此，您不应在 mqm 组中注册应用程序和普通用户。 mqm 组应仅包含 IBM MQ 管理员的帐户。

以下内容中描述了这些任务:

- Windows 在 Windows 上创建和管理组
- AIX 在 AIX 上创建和管理组
- Linux 在 Linux 上创建和管理组

Windows 如果域控制器在 Windows 2000 或 Windows 2003 或更高版本上运行，那么域管理员可能必须设置一个特殊帐户以供 IBM MQ 使用。 有关更多信息，请参阅 使用 Prepare IBM MQ Wizard 配置 IBM MQ 和 为 IBM MQ 创建和设置 Windows 域帐户。

ALW 在 AIX, Linux, and Windows 上使用 IBM MQ 对象的权限

所有对象都受 IBM MQ 保护，并且必须为主体授予访问这些对象的相应权限。 不同主体需要对不同对象的不同访问权。

队列管理器，队列，进程定义，名称列表，通道，客户机连接通道，侦听器，服务和认证信息对象都可从使用 MQI 调用或 PCF 命令的应用程序进行访问。 这些资源都受 IBM MQ 保护，需要授予应用程序访问这些资源的许可权。 发出请求的实体可以是用户，发出 MQI 调用的应用程序或发出 PCF 命令的管理程序。 请求者的标识称为 主体。

可以向不同主体组授予对同一对象的不同类型的访问权限。 例如，对于特定队列，可能允许一个组同时执行 put 和 get 操作; 可能只允许另一个组浏览队列 (带有浏览选项的 MQGET)。 同样，某些组可能具有对队列的放置和获取权限，但不允许更改该队列的属性或将其删除。

某些操作特别敏感，应该仅限于特权用户。 例如:

- 访问一些特殊队列，例如传输队列或命令队列 SYSTEM.ADMIN.COMMAND.QUEUE
- 运行使用完整 MQI 上下文选项的程序
- 创建和删除应用程序队列

将自动向创建对象的用户标识和 mqm 组的所有成员 (以及 Windows 系统上本地 Administrators 组的成员) 授予对对象的完全访问许可权。

相关概念

第 353 页的『在 AIX, Linux, and Windows 上管理 IBM MQ 的权限』

IBM MQ 管理员可以使用所有 IBM MQ 命令并授予其他用户权限。 当管理员向远程队列管理器发出命令时，他们必须在远程队列管理器上具有必需的权限。 进一步的注意事项适用于 Windows 系统。

ALW 在 AIX, Linux, and Windows 上进行安全性检查时

通常会在连接到队列管理器，打开或关闭对象以及放置或获取消息时进行安全性检查。

针对典型应用程序进行的安全性检查如下所示:

连接到队列管理器 (MQCONN 或 MQCONNX 调用)

这是应用程序首次与特定队列管理器相关联。 队列管理器询问操作环境以发现与应用程序关联的用户标识。 然后， IBM MQ 将验证该用户标识是否有权连接到队列管理器，并保留该用户标识以供将来检查。

用户不必登录到 IBM MQ; IBM MQ 假定用户已登录到底层操作系统并已通过该系统认证。

打开对象 (MQOPEN 或 MQPUT1 调用)

可通过打开对象并对其发出命令来访问 IBM MQ 对象。 所有资源检查都在打开对象时执行，而不是在实际访问对象时执行。 这意味着 **MQOPEN** 请求必须指定所需的访问类型 (例如，用户是只想浏览对象还是像将消息放入队列一样执行更新)。

IBM MQ 将检查 **MQOPEN** 请求中指定的资源。 对于别名或远程队列对象，使用的授权是对象本身的授权，而不是别名或远程队列解析到的队列。 这意味着用户不需要访问它的许可权。 将创建队列的权限限制为特权用户。 如果不执行此操作，那么用户可以仅通过创建别名来绕过正常访问控制。 如果使用队列和队列管理器名称显式地引用了远程队列，那么将检查与远程队列管理器相关联的传输队列。

对动态队列的权限基于派生自的模型队列的权限，但不一定相同。 这在注释 第 113 页的『1』 中进行了描述。

队列管理器用于访问检查的用户标识是从连接到队列管理器的应用程序的操作环境中获取的用户标识。 适当授权的应用程序可以发出 **MQOPEN** 调用，指定备用用户标识; 然后对备用用户标识进行访问控制检查。 这不会更改与应用程序关联的用户标识，仅更改用于访问控制检查的用户标识。

放置和获取消息 (MQPUT 或 MQGET 调用)

不执行访问控制检查。

关闭对象 (MQCLOSE)

除非 **MQCLOSE** 导致删除动态队列，否则不会执行访问控制检查。 在这种情况下，将检查用户标识是否有权删除队列。

预订主题 (MQSUB)

当应用程序预订主题时，它指定需要执行的操作类型。 它正在创建新预订，更改现有预订或在不更改现有预订的情况下恢复现有预订。 对于每种类型的操作，队列管理器会检查与应用程序关联的用户标识是否有权执行该操作。

当应用程序预订主题时，将对主题树中的主题对象执行权限检查，这些对象位于应用程序预订的主题树中的点或上方。 权限检查可能涉及对多个主题对象的检查。

队列管理器用于权限检查的用户标识是应用程序连接到队列管理器时从操作系统获取的用户标识。

队列管理器对订户队列执行权限检查，但不对受管队列执行权限检查。

ALW IBM MQ 在 AIX, Linux, and Windows 上如何实现访问控制

IBM MQ 使用底层操作系统提供的安全服务，使用对象权限管理器。 IBM MQ 提供了用于创建和维护访问控制表的命令。

称为 "授权服务接口" 的访问控制接口是 IBM MQ 的一部分。 IBM MQ 提供称为 对象权限管理器 *(OAM)*的访问控制管理器 (符合授权服务接口) 的实现。 除非您另行指定 (如 第 319 页的『阻止对 AIX, Linux, and Windows 系统进行安全访问检查』 中所述) ，否则将自动为您创建的每个队列管理器安装并启用此功能。 可以将 OAM 替换为符合授权服务接口的任何用户或供应商编写的组件。

OAM 利用底层操作系统的安全功能，使用操作系统用户和组标识。 仅当用户具有正确的权限时，他们才能访问 IBM MQ 对象。 第 310 页的『通过在 AIX, Linux, and Windows 上使用 OAM 来控制对对象的访问』 描述了如何授予和撤销此权限。

OAM 为其控制的每个资源维护一个访问控制表 (ACL)。 授权数据存储在名为 SYSTEM.AUTH.DATA.QUEUE。此队列的访问权仅限于 mqm 组中的用户，此外在 Windows 上仅限于 Administrators 组中的用户以及使用 SYSTEM 标识登录的用户。 无法更改用户对队列的访问权。

IBM MQ 提供了用于创建和维护访问控制表的命令。 有关这些命令的更多信息，请参阅 第 310 页的『通过在 AIX, Linux, and Windows 上使用 OAM 来控制对对象的访问』。

IBM MQ 向 OAM 传递包含主体，资源名称和访问类型的请求。 OAM 根据其维护的 ACL 授予或拒绝访问权。IBM MQ 遵循 OAM 的决策; 如果 OAM 无法做出决策，那么 IBM MQ 不允许访问。

ALW 在 AIX, Linux, and Windows 上标识用户标识

对象权限管理器标识请求访问资源的主体。 用作主体的用户标识因上下文而异。

对象权限管理器 (OAM) 必须能够识别请求访问特定资源的人员。 IBM MQ 使用术语 主体 来引用此标识。 主体是在应用程序首次连接到队列管理器时建立的; 它由队列管理器根据与连接应用程序相关联的用户标识确定。 (如果应用程序在未连接到队列管理器的情况下发出 XA 调用，那么与发出 xa_open 调用的应用程序相关联的用户标识将用于队列管理器的权限检查。)

在 AIX and Linux 系统上，授权例程检查与应用程序关联的实际 (登录) 用户标识或有效用户标识。 检查的用户标识可能依赖于绑定类型，有关详细信息，请参阅 可安装服务。

IBM MQ 将从系统接收的用户标识作为用户标识传播到每条消息的消息头 (MQMD 结构) 中。 此标识是消息上下文信息的一部分，在 第 358 页的『AIX, Linux, and Windows 上的上下文权限』中进行了描述。 除非已授权应用程序更改上下文信息，否则应用程序无法更改此信息。

ALW *AIX, Linux, and Windows* 上的主体和组

主体可以属于组。 通过向组而不是个人授予资源访问权，可以减少所需的管理量。 访问控制表 (ACL) 基于组和用户标识。

例如，您可以定义由想要运行特定应用程序的用户组成的组。 通过将其他用户的用户标识添加到相应的组，可以授予这些用户对其所需的所有资源的访问权。

针对特定平台描述了定义和管理组的此过程:

- AIX 在 AIX 上创建和管理组
- Linux 在 Linux 上创建和管理组
- Windows 在 Windows 上创建和管理组

主体可以属于多个组 (其组集)。 它具有向其组集中的每个组授予的所有权限的聚集。 将对这些权限进行高速缓存，因此除非您发出 MQSC 命令 **REFRESH SECURITY** (或其 PCF 等效命令) ，否则直到重新启动队列管理器之后才会识别您对主体的组成员资格所作的任何更改。

Linux AIX AIX and Linux 系统

访问控制列表 (ACL) 基于用户 ID 和组，您可以通过设置 **SecurityPolicy** 属性为适当的值，如中所述 服务节 qm.ini 文件。

您可以使用 基于用户的模型 进行授权，这允许您同时使用用户和组。 但是，在 setmqaut 命令中指定用户时，新许可权仅适用于该用户，而不适用于该用户所属的任何组。 有关更多信息，请参阅第 310 页的『AIX and Linux 上基于 OAM 用户的许可权』。

使用 基于组的模型 进行授权时， ACL 中包含用户标识所属的主组。 未包含个人用户标识，并且已向该组的所有成员授予权限。 因此，请注意，您可以通过更改同一组中另一个主体的权限来不经意地更改该主体的权限。

所有用户名义上都分配给缺省用户组 no，缺省情况下，不会向该组授予任何权限。 您可以更改 **没人** 组中的权限，以向没有特定权限的用户授予对 IBM MQ 资源的访问权。

从 IBM MQ 9.3.0 开始，您可以使用 **SecurityPolicy** 属性的 `UserExternal` 选项来创建非操作系统用户名。 如果创建非操作系统用户名，那么该用户将被视为不属于除 nobody 组以外的任何组。 有关此选项的更多信息，请参阅 crtmqm 和 qm.ini 文件的服务节。

请勿使用值 `UNKNOWN` 定义用户标识。 当用户标识太长时，将使用值 `UNKNOWN`，因此任意用户标识将使用 `UNKNOWN` 的访问权限。

有关使用 LDAP 的信息，请参阅 第 363 页的『设置权限』。

用户标识最多可以包含 12 个字符，组名最多可以包含 12 个字符。

Windows Windows 系统

ACL 基于用户标识和组。 检查与 AIX and Linux 相同。 您可以在具有相同用户标识的不同域上具有不同的用户。 IBM MQ 允许用户标识由域名限定，以便为这些用户提供不同级别的访问权。

组名可以选择包含以以下格式指定的域名:

```
GroupName@domain domain_name\group_name
```

全局组仅在两种情况下由 OAM 检查:

1. 队列管理器安全性节包含以下设置: `GroupModel=GlobalGroups`。 请参阅 保护。
2. 队列管理器正在使用备用安全访问组。 请参阅 **crtmqm**。

用户标识最多可包含 20 个字符，域名最多可包含 15 个字符，组名最多可包含 64 个字符。

OAM 首先检查本地安全数据库，然后检查主域的数据库，最后检查任何可信域的数据库。 迁到的第一个用户标识由 OAM 用于检查。 其中每个用户标识在特定计算机上可能具有不同的组成员资格。

某些控制命令 (例如， **crtmqm**) 使用对象权限管理器 (OAM) 更改对 IBM MQ 对象的权限。 OAM 按照上一段中给出的顺序搜索安全数据库，以确定特定用户标识的权限。 因此，由 OAM 确定的权限可能会覆盖用户标识是本地 mqm 组的成员这一事实。 例如，如果通过全局组从具有本地 mqm 组成员资格的域控制器认证的用户标识发出 **crtmqm** 命令，那么如果系统具有不在本地 mqm 组中的同名本地用户，那么该命令将失败。

有关设置的更多信息 **SecurityPolicy** 归因于 Windows， 看服务节 qm.ini 文件。

Windows *Windows 安全标识 (SID)*

Windows 上的 IBM MQ 使用其中可用的 SID。 如果未随授权请求一起提供 Windows SID，那么 IBM MQ 仅根据用户名来标识用户，但这可能会导致授予错误的权限。

在 Windows 系统上，安全标识 (SID) 用于补充用户标识。 SID 包含用于标识定义用户的 Windows 安全帐户管理器 (SAM) 数据库上的完整用户帐户详细信息的信息。 在 IBM MQ for Windows 上创建消息时， IBM MQ 会将 SID 存储在消息描述符中。 当 IBM MQ on Windows 执行授权检查时，它将使用 SID 从 SAM 数据库中查询完整信息。 (定义了用户的 SAM 数据库必须可访问才能成功执行此查询。)

缺省情况下，如果未随授权请求提供 Windows SID，那么 IBM MQ 将根据用户名单独标识用户。 它通过按以下顺序搜索安全数据库来执行此操作:

1. 本地安全数据库
2. 主域的安全数据库
3. 可信域的安全数据库

如果用户名不唯一，那么可能会授予不正确的 IBM MQ 权限。 要防止此问题，请在每个授权请求中包含 SID; SID 由 IBM MQ 用于建立用户凭证。

要指定所有授权请求都必须包含 SID，请使用 **regedit**。 将 SecurityPolicy 设置为 NTSIDsRequired。

ALW AIX, Linux, and Windows 上的备用用户权限

您可以指定用户标识在访问 IBM MQ 对象时可以使用另一个用户的权限。 这称为 备用用户权限，您可以在任何 IBM MQ 对象上使用此权限。

当服务器接收来自程序的请求并希望确保该程序具有请求的必需权限时，备用用户权限至关重要。 服务器可能具有必需的权限，但它需要知道程序是否具有它所请求的操作的权限。

例如，假定在用户标识 PAYSERV 下运行的服务器程序从用户标识 USER1 放入队列的队列中检索请求消息。 当服务器程序获取请求消息时，它将处理请求并将应答重新放入与请求消息一起指定的应答队列中。 服务器可以指定其他用户标识 (在本例中为 USER1)，而不是使用自己的用户标识 (PAYSERV) 来授权打开应答队列。 在此示例中，您可以使用备用用户权限来控制是否允许 PAYSERV 在打开应答队列时将 USER1 指定为备用用户标识。

在对象描述符的 **AlternateUserId** 字段上指定备用用户标识。

Linux 解决 Linux 上的特定组成员资格问题

某些系统通过正常系列的 **getgrent** 操作系统 API 调用返回组信息的速度很慢，如果您的企业有数千个组要搜索，正在查找 mqm 用户所在的组，那么响应缓慢可能会导致内部队列管理器超时。 为了避免此问题，提供了备用操作系统 API。

要使用更快的备用 API 并从一个调用返回所有组，请设置环境变量 MQS_GETGROUPLIST_API。

当授予对用户辅助组的连接访问权并启用 MQS_GETGROUPLIST_API 变量时，您可能收到 RC2035 错误。

然后， IBM MQ 将使用 **getgrouplist** API 而不是 **getgrent** API。

要启用 **getgrouplist**:

1. 停止队列管理器
2. 发出命令导出 MQS_GETGROUPLIST_API=1
3. 重新启动队列管理器

重试失败的方案，如果问题已解决，那么您可以考虑修改用户 mqm 的 .bashrc / .profile 文件以添加此环境变量，或者将此环境变量添加到用于启动队列管理器的脚本中。

如果系统从多个存储库 (例如 NIS 或 LDAP) 合并操作系统的用户或组信息，请确保组或用户标识在所有存储库 (包括本地存储库) 之间一致，因为这些存储库用于安装和设置操作系统级别许可权。

ALW AIX, Linux, and Windows 上的上下文权限

上下文是适用于特定消息的信息，包含在消息描述符 MQMD 中， MQMD 是消息的一部分。 在进行 MQOPEN 或 MQPUT 调用时，应用程序可以指定上下文数据。

上下文信息有两个部分:

身份部分
消息来自谁。 它由 UserIdentifier， AccountingToken 和 ApplIdentityData 字段组成。

"源" 部分
消息来自何处，以及何时将其放入队列。 它由 PutApplType， PutApplName， PutDate， PutTime 和 ApplOriginData 字段组成。

在进行 MQOPEN 或 MQPUT 调用时，应用程序可以指定上下文数据。 此数据可能由应用程序生成，从另一条消息传递，或者缺省情况下由队列管理器生成。 例如，服务器程序可以使用上下文数据来检查请求者的身份，从而测试消息是否来自使用授权用户标识运行的应用程序。

服务器程序可以使用 UserIdentifier 来确定备用用户的用户标识。 您可以使用上下文授权来控制用户是否可以在任何 MQOPEN 或 MQPUT1 调用上指定任何上下文选项。

请参阅 控制上下文信息 以获取有关上下文选项的信息，并参阅 MQMD-消息描述符 以获取与上下文相关的消息描述符字段的描述。

在安全出口中实现访问控制

您可以使用 MCAUserIdentifier 或对象权限管理器在安全出口中实现访问控制。

MCAUserIdentifier

当前通道的每个实例都具有关联的通道定义结构 MQCD。 MQCD 中字段的初始值由 IBM MQ 管理员创建的通道定义确定。 特别是，其中一个字段的初始值 *MCAUserIdentifier* 由 DEFINE CHANNEL 命令中的 MCAUSER 参数值确定，或者由以其他方式创建通道定义的等价于 MCAUSER 的值确定。

MQCD 结构在由 MCA 调用时传递到通道出口程序。 当 MCA 调用安全出口时，该安全出口可以更改 *MCAUserIdentifier* 的值，从而替换在通道定义中指定的任何值。

Multi 在 多平台上，除非 *MCAUserIdentifier* 的值为空，否则当 MCA 在连接到队列管理器后尝试访问队列管理器的资源时，队列管理器将使用 *MCAUserIdentifier* 的值作为权限检查的用户标识。 如果 *MCAUserIdentifier* 的值为空，那么队列管理器将改为使用 MCA 的缺省用户标识。 这适用于 RCVR，RQSTR， CLUSRCVR 和 SVRCONN 通道。 对于发送 MCA，缺省用户标识始终用于权限检查，即使 *MCAUserIdentifier* 的值不是空白也是如此。

z/OS 在 z/OS 上，队列管理器可以使用 *MCAUserIdentifier* 的值进行权限检查，前提是该值不为空。 对于接收 MCA 和服务器连接 MCA，队列管理器是否使用 *MCAUserIdentifier* 的值进行权限检查取决于:

- 通道定义中 PUTAUT 参数的值
- 用于检查的 RACF 概要文件
- 通道启动程序地址空间用户标识对 RESLEVEL 概要文件的访问级别

对于发送 MCA，它取决于:

- 发送 MCA 是调用者还是响应者
- 通道启动程序地址空间用户标识对 RESLEVEL 概要文件的访问级别

可以通过各种方式获取安全出口存储在 *MCAUserIdentifier* 中的用户标识。 以下是一些示例:

- 如果 MQI 通道的客户机端没有安全出口，那么当客户机应用程序发出 MQCONN 调用时，与 IBM MQ 客户机应用程序关联的用户标识将从客户机连接 MCA 流向服务器连接 MCA。服务器连接 MCA 将此用户标识存储在通道定义结构 MQCD 中的 *RemoteUserIdentifier* 字段中。 如果此时 *MCAUserIdentifier* 的值为空，那么 MCA 会将同一用户标识存储在 *MCAUserIdentifier* 中。 如果 MCA 未将用户标识存储在 *MCAUserIdentifier* 中，那么安全出口稍后可以通过将 *MCAUserIdentifier* 设置为 *RemoteUserIdentifier* 的值来执行此操作。

  如果来自客户机系统的用户标识正在进入新的安全域并且在服务器系统上无效，那么安全出口可以将该用户标识替换为有效的用户标识，并将替换的用户标识存储在 *MCAUserIdentifier* 中。
- 用户标识可由合作伙伴安全出口在安全消息中发送。

  在消息通道上，由发送 MCA 调用的安全出口可以发送正在运行发送 MCA 的用户标识。 然后，由接收 MCA 调用的安全出口可以将用户标识存储在 *MCAUserIdentifier* 中。 类似地，在 MQI 通道上，通道客户机端的安全出口可以发送与 IBM MQ MQI client 应用程序关联的用户标识。 然后，通道服务器端的安全出口可以将用户标识存储在 *MCAUserIdentifier* 中。 与上一个示例一样，如果用户标识在目标系统上无效，那么安全出口可以将用户标识替换为有效的用户标识，并将替换后的用户标识存储在 *MCAUserIdentifier* 中。

  如果作为标识和认证服务的一部分接收数字证书，那么安全出口可以将证书中的专有名称映射到在目标系统上有效的用户标识。 然后，它可以将用户标识存储在 *MCAUserIdentifier* 中。
- 如果在通道上使用 TLS，那么会将合作伙伴的专有名称 (DN) 传递到 MQCD 的 SSLPeerNamePtr 字段中的出口，并将该证书的签发者的 DN 传递到 MQCXP 的 SSLRemCertIssNamePtr 字段中的出口。

有关 *MCAUserIdentifier* 字段，通道定义结构， MQCD 和通道出口参数结构 MQCXP 的更多信息，请参阅 通道出口调用和数据结构。 有关 MQI 通道上从客户机系统流出的用户标识的更多信息，请参阅 访问控制。

注: 在 IBM WebSphere MQ 7.1 发行版之前构造的安全出口应用程序可能需要更新。 有关更多信息，请参阅 通道安全出口程序。

IBM MQ 对象权限管理器用户认证

在 IBM MQ MQI client 连接上，可以使用安全出口来修改或创建在对象权限管理器 (OAM) 用户认证中使用的 MQCSP 结构。 这在 消息传递通道的通道出口程序 中进行了描述

在消息出口中实现访问控制

您可能需要使用消息出口将一个用户标识替换为另一个用户标识。

请考虑将消息发送到服务器应用程序的客户机应用程序。 服务器应用程序可以从消息描述符中的 *UserIdentifier* 字段中抽取用户标识，如果它具有备用用户权限，请队列管理器在代表客户机访问 IBM MQ 资源时使用此用户标识进行权限检查。

如果在通道定义中将 PUTAUT 参数设置为 CTX (或 z/OS 上的 ALTMCA)，那么当 MCA 打开目标队列时，将使用每条入局消息的 *UserIdentifier* 字段中的用户标识进行权限检查。

在某些情况下，生成报告消息时，将使用导致报告的消息的 *UserIdentifier* 字段中用户标识的权限进行放置。 特别是，交付时确认 (COD) 报告和到期报告始终具有此权限。

由于这些情况，在消息进入新的安全域时，可能需要在 *UserIdentifier* 字段中将一个用户标识替换为另一个用户标识。 这可以通过通道接收端的消息出口来完成。 或者，您可以确保在新的安全域中定义入局消息的 *UserIdentifier* 字段中的用户标识。

如果入局消息包含发送消息的应用程序用户的数字证书，那么消息出口可以验证证书并将证书中的专有名称映射到在接收系统上有效的用户标识。 然后，可以将消息描述符中的 *UserIdentifier* 字段设置为此用户标识。

如果消息出口需要更改入局消息中 *UserIdentifier* 字段的值，那么该消息出口可能适合同时认证消息的发送方。 有关更多详细信息，请参阅第 286 页的『消息出口中的身份映射』。

在 API 出口和 API 交叉出口中实施访问控制

API 或 API 交叉出口可以提供访问控制，以补充 IBM MQ 提供的访问控制。 尤其是，出口可以在消息级别提供访问控制。 该出口可确保应用程序只将满足特定条件的消息放入队列或从队列中获取这些消息。

请考虑以下示例：

- 消息包含有关订单的信息。 当应用程序尝试将消息放入队列时， API 或 API 交叉出口可以检查订单的总值是否小于某些规定的限制。
- 消息从远程队列管理器到达目标队列。 当应用程序尝试从队列中获取消息时， API 或 API 交叉出口可以检查消息的发送方是否有权将消息发送到队列。

Multi 流式队列安全性

流式队列功能允许管理员配置具有辅助队列的本地 (或模型) 队列，只要将消息放入原始队列，就会放置重复的消息。 关于队列流权限有两个方面需要考虑。

用于配置队列以流式处理重复消息的权限

如果要启用从一个队列到辅助队列的重复消息的消息流，那么必须具有执行此操作的许可权。 配置队列的 **STREAMQ** 属性的许可权要求您具有以下权限：

1. 他们正在更改其 **STREAMQ** 属性的队列的 CHG 权限
2. 要将重复消息放入的队列的 CHG 权限

这两种权限检查在配置时的组合可确保仅在原始队列上具有 CHG 权限的用户无法将消息放入他们没有许可权的另一个队列。

打开一个或多个队列并放置消息的权限

当应用程序打开已配置了辅助队列的队列时，将通过其 **STREAMQ** 属性进行权限检查，确保应用程序用户对原始队列具有 PUT 权限。

注: 不会对辅助队列上的应用程序用户进行其他权限检查，这类似于用于别名队列的权限模型。

使用来自原始队列或辅助队列的消息的应用程序需要 GET 或 BROWSE 权限，仅在使用这些消息的队列上。

不会在放置或获取时间进行其他权限检查。

示例

以下示例显示了为允许用户 admin 配置原始队列 INQUIRIES.QUEUE，用于将其重复消息流至本地队列 ANALYTICS.QUEUE，但阻止 admin 将消息复制到 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)
```

然后，用户 admin 可以发出以下命令:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(ANALYTICS.QUEUE)
```

但如果同一用户发出以下命令:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(PURCHASES.QUEUE)
```

以配置 INQUIRIES.QUEUE ，用于将重复消息放入 PURCHASES.QUEUE，它们接收到以下错误:

```
AMQ8135E 未授权
```

使用 INQUIRIES.QUEUE 已配置为将消息复制到 ANALYTICS.QUEUE，以下权限记录用于允许作为用户 appuser 运行的应用程序将消息放入 INQUIRIES.QUEUE，以及到 ANALYTICS.QUEUE:

```
SET AUTHREC PROFILE(INQUIRIES.QUEUE) PRINCIPAL('appuser') AUTHADD(PUT)
```

注: appuser 不需要 ANALYTICS.QUEUE。 队列管理器将重复的消息放入队列。

相关概念

流式队列

z/OS Streaming queues security on z/OS

The streaming queues feature allows an administrator to configure a local (or model) queue with a secondary queue, where duplicate messages are placed, whenever a message is put to the original queue. There are two aspects to consider regarding queue streaming authorities.

Authority to configure a queue for streaming duplicate messages

If you want to enable message streaming of duplicate messages from one queue to a secondary queue, you must have permission to do so. Permission to configure the **STREAMQ** attribute of a queue requires that you have the following profiles setup:

1. ALTER access level to MQADMIN or MXADMIN for the queue they are altering the **STREAMQ** attribute for
2. ALTER access level to MQADMIN or MXADMIN for the queue you want to stream messages to

The combination of these security checks at configuration time ensures that a user, who only has ALTER access on the original queue, cannot cause messages to be put to another queue on which they have no permissions.

Authority to open the queue or queues and put messages

When an application opens a queue that has been configured with a secondary queue, through its **STREAMQ** attribute, an authority check is made that the application user has UPDATE authority on the original queue.

Note: No additional authority check is made for the application user on the secondary queue, which is similar to the authority model used for alias queues.

Applications consuming messages from either the original or the secondary queue require UPDATE or READ authority, only on the queue they are consuming from.

No additional authority checks are made at put or get time.

Example

The following example shows the correct profiles being set to allow user ADMIN to configure an original queue, INQUIRIES.QUEUE, to stream messages to local queue ANALYTICS.QUEUE using RACF:

```
RDEFINE MQCMDS <QMGR>.ALTER.QLOCAL UACC(NONE) OWNER(<OWNER>)
PERMIT <QMGR>.ALTER.QLOCAL CLASS(MQCMDS) ID(ADMIN) ACCESS(ALTER)

RDEFINE MQADMIN <QMGR>.QUEUE.INQUIRIES.QUEUE UACC(NONE) OWNER(<OWNER>)
PERMIT <QMGR>.QUEUE.INQUIRIES.QUEUE CLASS(MQADMIN) ID(ADMIN) ACCESS(ALTER)

RDEFINE MQADMIN <QMGR>.QUEUE.ANALYTICS.QUEUE UACC(NONE) OWNER(<OWNER>)
PERMIT <QMGR>.QUEUE.ANALYTICS.QUEUE CLASS(MQADMIN) ID(ADMIN) ACCESS(ALTER)
```

User ADMIN is then able to issue the following command:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(ANALYTICS.QUEUE)
```

but if the same user issues the following command without setting up the correct security profiles:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(PURCHASES.QUEUE)
```

to configure INQUIRIES.QUEUE to put duplicate messages to PURCHASES.QUEUE, they receive the following error:

```
CSQM166I <QMGR> CSQMAQLC QLOCAL(INQUIRIES.QUEUE) NOT AUTHORIZED
```

Related concepts

Streaming queues

Multi LDAP 授权

您可以使用 LDAP 授权来消除对本地用户标识的需求。

受支持平台上 LDAP 授权的可用性

多平台上提供了 LDAP 授权:

注意:

从 IBM MQ 9.0 一般可用性开始，此功能在所有队列管理器上都可用，无论是新队列管理器还是从较早发行版迁移的队列管理器。

LDAP 授权概述

通过 LDAP 授权，处理授权配置的命令 (例如 **setmqaut** 和 **DISPLAY AUTHREC**) 可以处理专有名称。 先前，用户通过将其凭证与本地操作系统上用户和组的最大可用字符进行比较来进行认证。

注意: 如果已运行 **DEFINE AUTHINFO** 命令，那么必须重新启动队列管理器。 如果不重新启动队列管理器，那么 setmqaut 命令不会返回正确的结果。

如果用户提供的是用户标识，而不是专有名称，那么将处理该用户标识。 例如，在具有 PUTAUT (CTX) 的通道上存在入局消息时，会将用户标识中的字符映射到 LDAP 专有名称，并进行相应的授权检查。

其他命令 (例如 **DISPLAY CONN**) 继续使用并显示用户标识的实际值，即使该用户标识可能实际不存在于本地操作系统上也是如此。

Linux AIX 当 LDAP 授权到位时，队列管理器始终使用 AIX and Linux 平台上的安全性用户模型，而不考虑 qm.ini 文件中的 **SecurityPolicy** 属性。 因此，为单个用户设置许可权只会影响该用户，而不会影响属于该用户的任何组的任何其他用户。

与操作系统模型一样，用户仍具有已分配给个人和用户所属的所有组 (如果有) 的组合权限。

例如，假定已在 LDAP 存储库中定义以下记录。

- 在 **inetOrgPerson** 类中:

```
dn="cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry"
        email=JohnDoe1@yourcompany.com [longer than 12 characters]
        shortu=jodoe
        Phone=1234567
```

- 在 **groupOfNames** 类中:

```
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"
```

出于认证目的，必须定义使用此 LDAP 服务器的队列管理器，以使其 **CONNAUTH** 值指向类型为 IDPWLDAP 的 **AUTHINFO** 对象，并且其相关名称解析属性可能设置为如下所示:

```
USRFIELD(email) SHORTUSR(shortu)
BASEDNU(ou=users,o=yourcompany,c=yourcountry) CLASSUSR(inetOrgPerson)
```

给定此用于认证的配置，应用程序可以使用以下任一组值完成 MQCNO 调用中使用的 CSPUserID 字段:

```
" cn=JohnDoe ", " JohnDoe1@yourcompany.com ", " email=JohnDoe1@yourcompany.com "
```

或

```
" cn=JohnDoe, ou=users, o=ibm, c=uk ", " shortu=jodoe "
```

在任一情况下，系统都可以使用提供的值来认证 " jodoe"。

Multi 设置权限

如何使用短名称或 **USRFIELD** 来设置权限。

使用多种格式的方法 (如 第 362 页的『LDAP 授权』 中所述) 继续进入授权命令中，并具有可以以未修饰的方式使用 shortname 或 USRFIELD 的进一步扩展。

当为授权的用户 (主体) 命名时，字符串指定 LDAP 记录中的特定属性。

要点: 字符串不得包含 = 字符，因为不能在操作系统用户标识中使用此字符。

如果将主体名称传递到 OAM 以进行可能是 shortname 的授权，那么字符串必须适合 12 个字符。 映射算法首先尝试在其 LDAP 查询中使用 SHORTUSR 属性将其解析为 DN。

如果由于 UNKNOWN_ENTITY 错误而失败，或者如果给定的字符串不能是 shortname，那么将使用 USRFIELD 属性进一步尝试构造 LDAP 查询。

注意: 如果已运行 DEFINE AUTHINFO 命令，那么必须重新启动队列管理器。 如果不重新启动队列管理器，那么 setmqaut 命令不会返回正确的结果。

对于处理用户权限，以下 setmqaut 命令设置都是等效的。

表 75: 用户授权设置	
命令	**注**
`setmqaut -m QM -t qmgr -p jodoe +connect`	这是一个平面的非限定名，通过 SHORTUSR 解析。
`setmqaut -m QM -t qmgr -pJohnDoe1@yourcompany.com +connect`	也是一个平面的非限定名，通过 USRFIELD 解析到同一个实体。
`setmqaut -m QM -t qmgr -p email=JohnDoe1@yourcompany.com +connect`	使用指定的属性。
`setmqaut -m QM -t qmgr -p "phone=1234567" +connect`	使用不必是 AUTHINFO 对象上配置的任何属性的另一个指定属性。

可以使用 SET AUTHREC MQSC 命令作为 **setmqaut** 命令的替代方法:

```
SET AUTHREC OBJTYPE(QMGR) PRINCIPAL('JohnDoe1@yourcompany.com') AUTHADD(connect)
```

或带有包含字符串的 MQCACF_PRINCIPAL_ENTITY_NAMES 元素的 Set Authority Record (MQCMD_SET_AUTH_REC) PCF 命令:

```
 "cn=JohnDoe,ou=users,o=yourcompany,c=yourcountry"
```

处理组时，对于 shortname 处理没有任何歧义，因为不需要将任何形式的组名拟合为 12 个字符。 因此，没有等价于组的 SHORTUSR 属性。

这意味着 第 364 页的表 76 中描述的语法示例有效，假定您已使用扩展属性配置 AUTHINFO 对象，并设置为:

```
GRPFIELD(longname)
BASEDNG(ou=groups,o=yourcompany,c=yourcountry ) CLASSGRP(groupOfNames)
```

表 76: 组授权设置	
命令	**注**
`setmqaut -m QM -t qmgr -g ApplicationGroupA +connect`	使用 GRPFIELD 进行解析
`setmqaut -m QM -t qmgr -g longname=ApplicationGroupA +connect`	对单个属性进行命名
`setmqaut -m QM -t qmgr -g "cn=Application Group A,ou=groups,o=yourcompany,c=yourcountry" +connect`	使用完整 DN

可以使用 SET AUTHREC MQSC 命令作为上述 **setmqaut** 命令的替代方法:

```
SET AUTHREC OBJTYPE(QMGR) GROUP('ApplicationGroupA')
      AUTHADD(connect)
```

或带有包含字符串的 MQCACF_GROUP_ENTITY_NAMES 元素的 Set Authority Record (MQCMD_SET_AUTH_REC) PCF 命令:

```
 "ApplicationGroupA"
```

要点:

无论您使用哪种格式来引用名称 (无论是用户还是组) ，都必须能够派生唯一的 DN。

因此，例如，您不得有两个具有 "shortu=jodoe" 的不同记录。

如果无法确定单个唯一 DN ，那么 OAM 将返回 MQRC_UNKNOWN_ENTITY。

Multi 显示权限

显示用户或组的授权的各种方法。

dspmqaut 命令

显示可用于用户或组的权限的最简单方法是使用 dspmqaut 命令。

您可以对任何语法变体使用查询来标识用户或组。 请注意，命令输出将以命令行上给出的格式重复标识。 输出不会报告完整的已解析 DN。

例如:

```
dspmqaut -m QM -t qmgr -p johndoe
Entity johndoe has the following authorizations for object QM:
    connect
```

或

```
dspmqaut -m QM -t qmgr -p email=JohnDoe1@yourcompany.com
Entity email=JohnDoe1@yourcompany.com has the following authorizations for object QM:
    connect
```

dmpmqaut 和 dmpmqcfg 命令

dmpmqaut 命令及其 MQSC 或 PCF 等效命令可以以任何受支持的格式 (如 第 363 页的『设置权限』中描述的 **setmqaut** 表) 指定主体或组。 但是，与 **dspmqaut** 不同， **dmpmqaut** 命令始终报告完整 DN。

```
dmpmqaut -m QM -t qmgr -p jodoe
------------------------------------
profile: self
object type:qmgr
entity:cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry
entity type: principal
authority: connect
```

同样，对所选记录没有任何过滤的 dmpmqcfg 命令始终以稍后可重放的格式显示完整 DN。

```
dmpmqcfg -m QM -x authrec
------------------------------------
SET AUTHREC PROFILE(SELF) +
    PRINCIPAL('cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry') +
    OBJTYPE(QMGR)
    AUTHADD(CONNECT)
```

Multi 使用 LDAP 授权时的其他注意事项

对消息队列接口 (MQI) 以及使用来自 IBM MQ 9.0.0 的 LDAP 授权时需要注意的其他 MQSC 和 PCF 命令的更改的简要描述。

ADOPTCTX

应用程序不需要提供认证信息，也不需要将 ADOPTCTX 属性设置为 YES。

如果应用程序未显式认证，或者如果针对活动 CONNAUTH 对象将 **ADOPTCTX** 设置为 NO ，那么将从操作系统用户标识获取与该应用程序相关联的身份上下文。

需要应用授权时，将使用与 setmqaut 命令相同的规则将该上下文映射到 LDAP 身份。

MQI 调用的输入参数

MQOPEN， MQPUT1 和 MQSUB 具有允许指定备用用户标识的结构。

如果使用这些字段，那么将使用与 **setmqaut**， **dmpmqaut** 和 **dspmqaut** 命令相同的规则将 12 个字符的用户标识映射到 DN。

MQPUT 和 MQPUT1 还允许适当的授权程序设置 MQMD UserIdentifier 字段。 此字段的值在 PUT 过程中未被管理，可以设置为任何值。

但是，与往常一样， **UserIdentifier** 值可以在消息处理的稍后阶段用于授权，例如，在接收通道上定义 PUTAUT (CTX) 时。

此时，将使用该接收队列管理器 (可以是 LDAP 或基于操作系统) 的配置来检查标识以获取授权。

MQI 调用的输出参数

无论在 MQI 结构中向程序提供用户标识的位置，它都是与连接关联的 12 个字符的短名称版本。

例如， API 出口的 **MQAXC.UserId** 值是从 LDAP 映射返回的短名称。

其他管理 MQSC 和 PCF 命令

用于显示处于对象状态的用户信息 (例如， DISPLAY CONN USERID) 的命令返回与上下文关联的 12 个字符的短名称。 未显示完整 DN。

允许断言身份的命令 (例如通道的 CHLAUTH 映射规则或 MCAUSER 值) 可以采用最多为这些属性定义的最大长度 (当前为 64 个字符) 的值。

对语法没有任何更改。 当该身份需要授权时，会使用与 **setmqaut**， **dmpmqaut** 和 **dspmqaut** 命令相同的规则将其内部映射到 DN。

这意味着通道定义上的 MCAUSER 值可能不会显示为与 DISPLAY CHSTATUS 相同的字符串，但它们确实引用了相同的标识。

例如：

```
DEFINE CHL(SV1) CHLTYPE(SVRCONN) MCAUSER('cn=JohnDoe')
DEFINE CHL(SV2) CHLTYPE(SVRCONN) MCAUSER('jodoe')
DEFINE CHL(SV3) CHLTYPE(SVRCONN) MCAUSER('JohnDoe1@yourcompany.com')
```

然后， DISPLAY CHSTATUS (*) ALL 显示所有连接的 SHORTUSR 值 *MCAUSER (jodoe)* 。

Multi 在操作系统和 LDAP 授权模型之间切换

如何在不同平台上的不同授权方法之间切换。

队列管理器的 CONNAUTH 属性指向 AUTHINFO 对象。 当对象的类型为 IDPWLDAP 时，将使用 LDAP 存储库进行认证。

现在，您可以将授权方法应用于同一对象，这允许您继续进行基于操作系统的授权，或者使用 LDAP 授权

IBM i, AIX and Linux

Linux IBM i AIX

可以随时在操作系统和 LDAP 模型之间切换队列管理器。 您可以更改配置，并使用 REFRESH SECURITY TYPE (CONNAUTH) 命令使该配置处于活动状态。

例如，如果已使用用于认证的连接信息配置此对象：

```
ALTER AUTHINFO(MYLDAP) AUTHTYPE(IDPWLDAP) +
        AUTHORMD(SEARCHGRP) +
        BASEDNG('ou=groups,o=ibm,c=uk') +
        <other attributes>
ALTER QMGR CONNAUTH(MYLDAP)
REFRESH SECURITY
```

Windows

Windows

如果权限配置更改涉及在操作系统和 LDAP 模型之间进行切换，那么必须重新启动队列管理器才能使更改生效。 否则，可以使用 REFRESH SECURITY TYPE (CONNAUTH) 命令使更改处于活动状态。

处理规则

从操作系统授权切换到 LDAP 授权时，任何已设置的现有操作系统权限规则都将变为不活动且不可见。

dmpmqaut 之类的命令不会显示这些操作系统规则。 同样，当从 LDAP 切换回操作系统时，任何已定义的 LDAP 权限都将变为不活动且不可见，从而恢复原始操作系统规则。

如果由于任何原因要使用 **dmpmqcfg** 命令来备份队列管理器的定义，那么该备份将仅包含针对备份时生效的授权方法定义的规则。

Multi LDAP 管理

每个平台如何管理 LDAP 的概述。

使用 LDAP 授权时，操作系统中 mqm 组 (或同等组) 的成员资格并不重要。 作为该组的成员仅控制是否可以处理某些命令行命令。

特别是，您必须在该组中才能发出 strmqm 和 endmqm 命令。

队列管理器运行后，现在会对完全特权帐户进行限制。 除发出 **strmqm** 命令的人员的用户标识外，属于操作系统 mqm (或同等) 组的其他用户不会获得特殊特权。

其他用户的权限基于他们所属的 LDAP 组。 不允许在诸如 **setmqaut** 之类的命令中不限定地使用 mqm 组名来映射到任何 LDAP 组。

AIX and Linux

Linux AIX

队列管理器运行后，唯一自动具有完全特权的帐户是启动队列管理器的实际用户。

mqm 标识仍然存在，并且用作操作系统资源 (例如文件) 的所有者，因为 mqm 是运行队列管理器的有效标识。 但是， mqm 用户将无法自动执行由 OAM 控制的管理任务。

Windows

Windows

在 Windows 上，自动完全特权帐户是启动队列管理器的操作系统用户，也是运行核心队列管理器进程的用户，例如 MUSR_MQADMIN (如果队列管理器已作为 Windows 服务启动)。

在 LDAP 授权方式下运行时， Windows 的行为与 AIX and Linux 平台非常相似。 它处理 12 个字符的短名称和完整 DN。

IBM i

IBM i

在 IBM i 上，自动特权帐户是启动队列管理器和 QMQM 标识的帐户。

您需要这两个标识，因为仅需要启动队列管理器的用户标识才能启动系统。 运行后，队列管理器进程仅具有 QMQM 权限。

用于提供 MQADMIN 特权的样本脚本

Linux AIX

由于具有能够对队列管理器执行完全管理的组很有用，因此在 AIX and Linux 平台上提供了样本脚本，如下所示:

```
MQ_INSTALLATION_PATH/samp/bin/amqauthg.sh
```

此样本采用两个参数:

- 队列管理器名称
- LDAP 组名

样本将处理 setmqaut 命令，授予所有对象的完整权限。 这是 IBM MQ Explorer OAM 向导为管理角色生成的相同脚本。 例如，代码启动:

```
setmqaut -t q -m qmgr -n "**" +alladm -g
      groupname
```

消息的机密性

加密消息可确保消息内容保持机密。 根据您的需要，可以使用各种方法对 IBM MQ 中的消息进行加密。

如果需要针对点到点消息传递基础结构的应用程序级别端到端数据保护，那么可以 使用 Advanced Message Security 对消息进行加密，或者 编写您自己的 API 出口或 API 交叉出口。

最安全的解决方案是提供端到端加密，方法是将消息从应用程序放置的位置加密到消费应用程序获取的位置。 这可以使用 第 92 页的『规划 Advanced Message Security』 (AMS) 来完成，也可以通过编写您自己的 API 出口或 API 交叉出口来完成; 请参阅 第 410 页的『在用户出口程序中实现机密性』 以获取更多信息。

如果仅需要在通过网络传输消息时对其进行加密，那么可以使用 TLS; 请参阅 第 21 页的『IBM MQ 中的 TLS 安全协议』 以获取更多信息，或者您可以编写自己的安全出口，消息出口或发送和接收出口程序以执行加密。

z/OS 如果需要对队列管理器上的静态消息进行加密，那么可以在该队列管理器上使用 z/OS 数据集加密; 请参阅 第 411 页的『Confidentiality for data at rest on IBM MQ for z/OS with data set encryption』 以获取更多信息。

相关任务

使用 TLS 连接两个队列管理器

安全地将客户机连接到队列管理器

启用 CipherSpecs

通过在 **DEFINE CHANNEL** 或 **ALTER CHANNEL** MQSC 命令中使用 **SSLCIPH** 参数来启用 CipherSpec 。

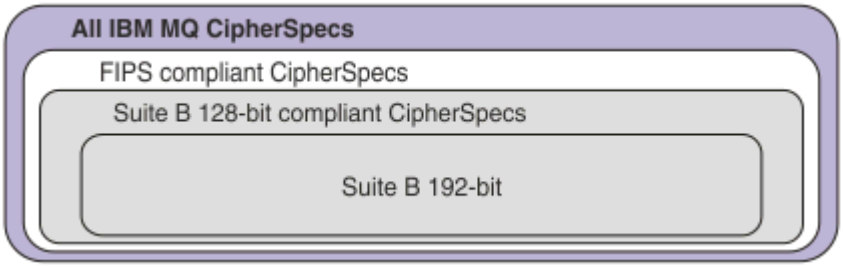
注: 在 AIX, Linux, and Windows 上， IBM MQ 通过 IBM Crypto for C (ICC) 加密模块提供 FIPS 140-2 合规性。 此模块的证书已移至历史状态。 客户应查看 IBM Crypto for C (ICC) 证书 并了解 NIST 提供的任何建议。 当前正在进行 FIPS 140-$tag1 替换模块，可以通过在 流程列表中的 NIST CMVP 模块中搜索该模块来查看其状态。

IBM MQ Operator 3.2.0 和队列管理器容器映像 9.4.0.0 和更高版本基于 UBI 9。 FIPS 140-3 合规性当前处于暂挂状态，可通过在 流程列表中的 NIST CMVP 模块中搜索 "Red Hat Enterprise Linux 9- OpenSSL FIPS 提供程序" 来查看其状态。

可以与 IBM MQ 配合使用的某些 CipherSpecs 符合 FIPS。 某些符合 FIPS 的 CipherSpecs 也符合 Suite B , 但其他的 (例如 TLS_RSA_WITH_AES_256_CBC_SHA) 则不符合 Suite B。

所有符合套件 B 的 CipherSpecs 也符合 FIPS。 所有符合 Suite B 的 CipherSpecs 分为两个组: 128 位 (例如， ECDHE_ECDSA_AES_128_GCM_SHA256) 和 192 位 (例如， ECDHE_ECDSA_AES_256_GCM_SHA384) ，

下图说明了这些子集之间的关系:



该产品在所有平台上都支持 TLS 1.3 安全协议。

第 369 页的表 77 中列出了可用于其中每个平台的 CipherSpecs 。 有关使用这些 CipherSpecs 的信息，请参阅 第 372 页的『在 IBM MQ 中使用 TLS 1.3』 和 第 372 页的『IBM MQ MQI client 和 TLS 1.3』 。

为了便于配置和将来进行迁移， IBM MQ 还提供了一组别名 CipherSpecs。 迁移现有安全配置以使用别名 CipherSpec 意味着您可以适应密码添加和弃用，而无需将来进行进一步的侵入性配置更改。 这些别名 CipherSpecs 在 第 369 页的表 77 的 "别名" CipherSpecs 部分中列出。 有关迁移以使用别名 CipherSpec 的更多信息，请参阅 迁移现有安全性配置以使用别名 CipherSpec。

您可以配置缺省 CipherSpecs ，如 第 372 页的『IBM MQ 中启用的缺省 CipherSpec 值』 中所述。 您还可以提供一组备用 CipherSpecs ，这些 CipherSpec 支持在以下位置与通道配合使用:

- Multi IBM MQ for Multiplatforms，如 第 380 页的『在 IBM MQ for Multiplatforms 上提供已订购和已启用的 CipherSpecs 的定制列表』 中所述。
- z/OS IBM MQ for z/OS，如 第 381 页的『在 IBM MQ for z/OS 上提供已订购和已启用的 CipherSpecs 的定制列表』 中所述。

第 381 页的『不推荐使用的 CipherSpecs』 中列出了您可以重新启用以用于 IBM MQ (如果需要) 的不推荐使用的 CipherSpecs 。

可用于 IBM MQ TLS 支持的 CipherSpecs

下表中列出了可以自动与 IBM MQ 队列管理器配合使用的 CipherSpecs 。 当您请求个人证书时，您为公用和专用密钥对指定密钥大小。 TLS 握手期间使用的密钥大小是证书中存储的大小，除非由 CipherSpec 确定 (如表中所述)。

表 77: *CipherSpec*，您可以用于 *IBM MQ TLS* 支持

平台支持 第 371 页的『1』	CipherSpec 名称	十六进制代码	使用的协议	MAC 算法	加密算法 (加密位)	FIPS 第 371 页的『2』	套件 B
别名 CipherSpec							
全部	ANY_TLS13_OR_HIGHER 第 371 页的『3』 第 371 页的『4』	不适用	已协商	已协商	已协商	已协商	已协商
全部	ANY_TLS13 第 371 页的『4』 第 371 页的『5』	不适用	TLS 1.3	已协商	已协商	已协商	已协商
全部	ANY_TLS12_OR_HIGHER 第 371 页的『4』 第 371 页的『6』	不适用	已协商	已协商	已协商	已协商	已协商
全部	ANY_TLS12 第 371 页的『7』	不适用	TLS 1.2	已协商	已协商	已协商	已协商

表 77: CipherSpec，您可以用于 IBM MQ TLS 支持 (继续)

平台支持 第 371 页的『1』	CipherSpec 名称	十六进制代码	使用的协议	MAC 算法	加密算法 (加密位)	FIPS 第 371 页的『2』	套件 B
全部	ANY 第 371 页的『8』	不适用	已协商	已协商	已协商	已协商	已协商
适用于 TLS 1.3 的 CipherSpec							
全部	TLS_AES_128_GCM_SHA256	1301	TLS 1.3	GCM	AES-128，带有 GCM (128)	Yes	否
全部	TLS_AES_256_GCM_SHA384	1302	TLS 1.3	GCM	AES-256，带有 GCM (256)	Yes	否
全部	TLS_CHACHA20_POLY1305_SHA256	1303	TLS 1.3	POLY1305	CHACHA20 (256)	否	否
ALW	TLS_AES_128_CCM_SHA256	1304	TLS 1.3	CBC-MAC	带有 CTR (128) 的 AES-128	Yes	否
ALW	TLS_AES_128_CCM_8_SHA256 第 371 页的『10』	1305	TLS 1.3	CBC-MAC	带有 CTR (128) 的 AES-128	Yes	否
适用于 TLS 1.2 的 CipherSpec							
全部	TLS_RSA_WITH_AES_128_CBC_SHA256 第 371 页的『9』	003C	TLS 1.2	SHA-256	AES (128)	Yes	否
全部	TLS_RSA_WITH_AES_256_CBC_SHA256 第 371 页的『9』 第 371 页的『11』	003D	TLS 1.2	SHA-256	AES (256)	Yes	否
全部	TLS_RSA_WITH_AES_128_GCM_SHA256 第 371 页的『9』 第 371 页的『12』	009C	TLS 1.2	SHA-256 和 AEAD GCM	AES (128)	Yes	否
全部	TLS_RSA_WITH_AES_256_GCM_SHA384 第 371 页的『9』 第 371 页的『11』 第 371 页的『12』	009D	TLS 1.2	SHA-384 和 AEAD GCM	AES (256)	Yes	否
全部	ECDHE_ECDSA_AES_128_CBC_SHA256 第 371 页的『9』	C023	TLS 1.2	SHA-256	AES (128)	Yes	否
全部	ECDHE_ECDSA_AES_256_CBC_SHA384 第 371 页的『9』 第 371 页的『11』	C024	TLS 1.2	SHA-384	AES (256)	Yes	否
全部	ECDHE_RSA_AES_128_CBC_SHA256 第 371 页的『9』	C027	TLS 1.2	SHA-256	AES (128)	Yes	否
全部	ECDHE_RSA_AES_256_CBC_SHA384 第 371 页的『9』 第 371 页的『11』	C028	TLS 1.2	SHA-384	AES (256)	Yes	否
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256 第 371 页的『11』 第 371 页的『12』	C02B	TLS 1.2	SHA-256 和 AEAD GCM	AES (SHA384)	Yes	128 位

表 77: *CipherSpec*，您可以用于 *IBM MQ TLS* 支持 (继续)

平台支持 第 371 页的『1』	CipherSpec 名称	十六进制代码	使用的协议	MAC 算法	加密算法 (加密位)	FIPS 第 371 页的『2』	套件 B
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384 第 371 页的『11』 第 371 页的『12』	C02C	TLS 1.2	SHA-384 和 AEAD GCM	AES (SHA384)	Yes	192 位
全部	ECDHE_RSA_AES_128_GCM_SHA256 第 371 页的『12』	C02F	TLS 1.2	SHA-256 和 AEAD GCM	AES (128)	Yes	否
全部	ECDHE_RSA_AES_256_GCM_SHA384 第 371 页的『11』 第 371 页的『12』	C030	TLS 1.2	AEAD AES-128 GCM	AES (SHA384)	Yes	否

注意:

1. 有关每个平台图标所涵盖的平台的列表，请参阅 产品文档中使用的图标。
2. 指定 CipherSpec 是否在经 FIPS 认证的平台上经 FIPS 认证。 请参阅美国联邦信息处理标准 (FIPS)，以获取 FIPS 的解释。
3. ALW ANY_TLS13_OR_HIGHER 别名 CipherSpec 协调远程端将允许但是仅使用 TLS 1.3 或更高版本协议进行连接的最高级别的安全性。
4. IBM i 要在 IBM i 上使用 TLS 1.3 或 ANY CipherSpec，底层操作系统版本必须支持 TLS 1.3。 请参阅 TLSv1.3 的系统 TLS 支持以获取更多信息。
5. ALW ANY_TLS13 别名 CipherSpec 表示使用 TLS 1.3 协议的可接受 CipherSpecs 的子集（如下表中针对每个平台所列示）。
6. ALW ANY_TLS12_OR_HIGHER 别名 CipherSpec 协调远程端将允许但是仅使用 TLS 1.2 或更高版本协议进行连接的最高级别的安全性。
7. ANY_TLS12 CipherSpec 表示使用 TLS 1.2 协议的可接受 CipherSpecs 的子集（如下表中针对每个平台所列示）。
8. ALW ANY 别名 CipherSpec 协调远程端将允许的最高级别的安全性。
9. IBM i 在系统值 QSSLCSLCTL 设置为 *OPSSYS 的 IBM i 7.4 系统上，未启用这些 CipherSpec。
10. ALW 这些 CipherSpec 使用 8-octet 完整性检查值 (ICV)，而不是 16-octet ICV。
11. 除非将相应的不受限策略文件应用于资源管理器使用的 JRE，否则无法使用此 CipherSpec 来保护从 IBM MQ Explorer 到队列管理器的连接。
12. ALW 根据 GSKit 的建议， TLS 1.2 GCM CipherSpecs 具有以下限制: 在使用同一会话密钥发送 2 个 24.5 TLS 记录后，将终止连接并返回消息 AMQ9288E。 此 GCM 限制处于活动状态，而不考虑所使用的 FIPS 方式。

    要防止发生此错误，请避免使用 TLS 1.2 GCM 密码，启用密钥重置，或者在设置了环境变量 GSK_ENFORCE_GCM_RESTRICTION=GSK_FALSE 的情况下启动 IBM MQ 队列管理器或客户机。 对于 GSKit 库，必须在连接的两侧设置此环境变量，并将其应用于队列管理器连接的客户机和队列管理器连接的队列管理器。 请注意，此设置会影响非受管 .NET 客户机，但不会影响 Java 或受管 .NET 客户机。 有关更多信息，请参阅 AES-GCM 密码限制。

    z/OS 此限制不适用于 IBM MQ for z/OS。

在 IBM MQ 中使用 TLS 1.3

该产品在所有平台上都支持 TLS 1.3 。

缺省情况下，在 IBM MQ 9.2.0 或更高版本上创建的队列管理器支持 TLS 1.3 。 从较早版本的 IBM MQ 迁移的队列管理器需要启用 TLS 1.3 。 您可以通过设置 **AllowTLSV13**=TRUE 属性在迁移的队列管理器上启用 TLS 1.3 :

- Multi 对于 IBM MQ for Multiplatforms 队列管理器，编辑 qm.ini 文件，并在 SSL 节下添加 **AllowTLSV13**=TRUE 属性 (链接到

```
SSL:
  AllowTLSV13=TRUE
```

- z/OS 对于 IBM MQ for z/OS 队列管理器，编辑 队列管理器启动 JCL 中指定的 QMINI 数据集 ，并在 TransportSecurity 节下添加 **AllowTLSV13**=TRUE 属性

```
TransportSecurity:
    AllowTLSV13=TRUE
```

启用 TLS 1.3 时，根据 TLS 1.3 规范，将拒绝与弱 CipherSpec 通信的任何尝试，无论是否在 IBM MQ 中启用这些尝试。 TLS 1.3 认为较弱的 CipherSpecs 是满足以下一个或多个条件的 CipherSpecs :

- 使用 SSL 3.0 协议。
- 使用 RC4 或 RC2 作为加密算法。
- 加密密钥大小 (位) 等于或小于 112。

这些限制通过 不推荐 CipherSpecs 的表 1 中的注释 [3] 进行标记。

如果需要继续使用此类 CipherSpecs，那么必须禁用 TLS 1.3 方式:

- ALW 编辑队列管理器的 qm.ini 文件，并将 **AllowTLSV13** 属性的设置更改为:

```
SSL:
  AllowTLSV13=FALSE
```

- z/OS 编辑队列管理器的 QMINI 数据集 ，并将 **AllowTLSV13** 属性的设置更改为:

```
TransportSecurity:
    AllowTLSV13=FALSE
```

IBM MQ MQI client 和 TLS 1.3

ALW

使用 IBM MQ MQI client 时，将推断 **AllowTLSV13** 的值，除非在应用程序正在使用的 mqclient.ini 文件的 SSL 节中显式指定该值。

- 如果启用了任何弱 CipherSpecs ，那么 **AllowTLSV13** 将设置为 FALSE ，并且不能使用 TLS 1.3 CipherSpecs 。
- 否则， **AllowTLSV13** 设置为 TRUE ，并且可以使用新的 TLS 1.3 CipherSpecs 和别名 CipherSpecs 。

IBM MQ 中启用的缺省 CipherSpec 值

在新的 IBM MQ 队列管理器的缺省配置中， IBM MQ 支持 TLS 1.2 和 TLS 1.3 协议以及使用 CipherSpecs 的各种密码算法。 出于兼容性目的，还可以将 IBM MQ 配置为使用 SSL 3.0 和 TLS 1.0 协议以及一些已知较弱或易受安全漏洞影响的密码算法。 在缺省配置中启用的 CipherSpecs 列表可能会通过应用维护来更改。

可以使用以下控件配置 IBM MQ 以限制或允许使用 CipherSpecs :

- 仅允许使用 SSLFIPS 的符合 FIPS 140-2 的 CipherSpecs 。

- ALW 仅允许使用 SUITEB 符合 NSA Suite B 的 CipherSpecs 。
- Multi 使用 **AllowedCipherSpecs** 允许 CipherSpecs 的定制列表。
- ALW 使用 **AMQ_ALLOWED_CIPHERS** 环境变量来允许 CipherSpecs 的定制列表。
- ALW 允许使用不推荐使用的 CipherSpecs (使用 **AllowWeakCipher** 或 **AMQ_SSL_WEAK_CIPHER_ENABLE** 环境变量)。
- z/OS 允许在 CHINIT JCL 中使用不推荐的 CipherSpecs 使用 DD 语句。

注: 如果使用 **AllowedCipherSpecs** 或 **AMQ_ALLOWED_CIPHERS** 指定 CipherSpecs 的定制列表，那么这将覆盖对任何不推荐使用的 CipherSpecs 的启用。 请注意，将 NSA Suite B 或 FIPS 140-2 限制与定制 CipherSpec 列表结合使用时，必须确保定制列表仅包含 Suite B 或 FIPS 140-2 设置所允许的 CipherSpecs 。

相关概念

第 39 页的『IBM MQ 中的数字证书和 CipherSpec 兼容性』
本主题通过概述 CipherSpecs 与 IBM MQ 中的数字证书之间的关系，提供有关如何为安全策略选择相应 CipherSpecs 和数字证书的信息。

第 19 页的『CipherSpecs 和 CipherSuites』
加密安全协议必须同意安全连接使用的算法。 CipherSpecs 和 CipherSuites 定义算法的特定组合。

第 37 页的『为 Suite B 配置 IBM MQ』
可以将 IBM MQ 配置为在 AIX, Linux, and Windows 平台上按照 NSA Suite B 标准运行。

第 29 页的『联邦信息处理标准 (FIPS)』
本主题介绍了美国国家标准与技术研究所的联邦信息处理标准 (FIPS) 加密验证程序以及可在 TLS 通道上使用的加密功能。

相关任务

迁移现有安全配置以使用别名 CipherSpe

相关参考

DEFINE CHANNEL
ALTER CHANNEL
更改，复制和创建通道

ALW AES-GCM 密码限制

用于 TLS 密码术时对 AES-GCM 密码施加的限制的指南。 这些限制由 IETF 和 NIST 组织实施，并且要求在使用 AES-GCM 密码时，不得使用相同的会话密钥来安全地传输超过 $2^{24.5}$ TLS 记录。

有关这些限制的更多信息，请参阅 RFC 9325 部分 4.4 限制密钥使用 和 RFC 8446 部分 5.5。

IBM MQ 不会直接实现加密功能。 而是使用多个不同的加密库来提供 TLS 和 Advanced Message Security 功能。 在 Windows， Linux 和 AIX 操作系统上， IBM MQ 使用的加密库为 IBM Global Security Kit (GSKit)。 对于应用程序， C 和非受管 .NET 库将 GSKit 用于加密功能。 GSKit 对 AES-GCM 加密算法的实现包括标准组指定的限制。 此外，缺省情况下还会启用这些限制。 因此，使用 AES-GCM 密码时，如果使用同一会话密钥传输超过 $2^{24.5}$ TLS 记录，那么 IBM MQ TLS 通信将终止。

注: 在 IBM i， IBM Z 或 IBM MQ for HPE NonStop 平台或 Java/JMS 受管 .NET 应用程序上不存在此限制，因为使用了不同的加密库，并且这些库未实现相同的限制。

如果 IBM MQ 通道保持运行足够长的时间，以便使用同一会话密钥传输超过 $2^{24.5}$ TLS 记录，那么底层加密库将终止连接。 这将导致通道终止，并生成 AMQ9288E 错误消息。 以这种方式终止其通信的应用程序从执行的任何 IBM MQ 操作中接收到 MQRC_CONNECTION_BROKEN 返回码。

连接的终止可以在通信的任一端执行，但只能在使用 GSKit 进行加密功能的一端执行。

缓解限制的建议

有关如何防止或处理由于此限制而终止的通信的一些选项如下所示:

使用可重新连接的客户机

可以将应用程序配置为在连接失败时自动尝试重新连接。 这包括由于 GCM 限制而终止的连接。 配置为重新连接时，将在任何故障点自动复原客户机应用程序，并且将复原用于打开对象的任何句柄。 这将在不返回到应用程序代码的情况下完成。

有关更多信息，请参阅客户机自动重新连接。

设置密钥重置值

可以将 IBM MQ 配置为在通过通道传输可配置的字节数之后请求会话密钥重置。 达到此限制时， IBM MQ 会请求加密层执行会话密钥重置，从而生成新的会话密钥。

需要注意的是，指定的值是传输的字节数，这与 IBM MQ 发送的消息的大小相关。 此限制是对发送的 TLS 记录数的限制。 消息字节与 TLS 记录之间没有直接映射，因为 TLS 记录可以发送依赖于网络的最大传输单元 (MTU) 的最大字节数。 发送的任何大于此值的消息都将作为多条 TLS 记录进行传输。 MTU 值因网络而异。 此外，还有其他原因导致可能需要在传输 IBM MQ 消息数据之外发送 TLS 记录，例如 IBM MQ 脉动信号检查， TLS 警报和其他 IBM MQ 协议消息。 这些额外的 TLS 记录将计入最大 TLS 记录数，但不会计入 IBM MQ 密钥重置值中。

使用密钥重置定期重置会话密钥可防止由于 AES-GCM 限制而终止通道。

有关更多信息，请参阅 重置 SSL 和 TLS 密钥。

使用 TLS 1.3 密码规范

虽然使用 TLS 1.3 协议时仍存在 AES-GCM 限制，但 TLS 1.3 协议支持自动执行会话密钥重置，而无需中断 TLS 通信。 这允许 GSKit 在需要时管理会话密钥的重置，而无需 IBM MQ 请求密钥重置。

有关更多信息，请参阅 第 368 页的『启用 CipherSpecs』中的 在 IBM MQ 中使用 TLS 1.3 。

禁用 AES-GCM 限制

如果需要，可以通过设置环境变量 **GSK_ENFORCE_GCM_RESTRICTION=GSK_FALSE** 来禁用 AES-GCM 限制来禁用该限制。 这样做将允许使用同一会话密钥发送任意数量的 TLS 记录。 如果选择此缓解，那么必须在使用 GSKit 进行安全通信的每个通信端设置环境变量。

警告: 建议不要使用此选项，因为在发送了超过 $2^{24.5}$ 条 TLS 记录之后，攻击者可以对已发送的记录执行分析以确定正在使用的会话密钥。 一旦确定了会话密钥，使用该会话密钥的所有现有和将来的通信都将受到损害。

TLS 握手中的 CipherSpec 顺序

在多个可能的 CipherSpecs 之间进行选择时，将使用 CipherSpecs 的顺序，例如，使用其中一个 ANY* CipherSpecs 时。

在 TLS 握手期间，客户机和服务器按其首选项顺序交换它们支持的 CipherSpecs 和协议。 选择了双方都优先使用的公共 CipherSpec ，并将其用于 TLS 通信。 在选择 CipherSpec 协议时，还会考虑版本，例如，如果服务器在 TLS 1.3 CipherSpecs 之前列出 TLS 1.2 CipherSpecs ，那么只要客户机能够支持 TLS 1.3 并具有可使用的公共 TLS 1.3 CipherSpec ，它仍将优先使用 TLS 1.3。

当为 TLS 配置 IBM MQ 时，它会将 CipherSpecs 设置为下表中显示的顺序，从最首选到最不首选。

注: 如果未通过 **AllowedCipherSpecs** 属性启用 CipherSpec ，那么不会将其配置为在 TLS 握手期间使用。

在未指定 **AllowedCipherSpecs** 属性的情况下，将使用由下表指示的已启用密码的缺省列表。

表 *78:* 来自 *IBM MQ 9.2.0* 的 *CipherSpecs* 订单

平台	CipherSpec	协议	十六进制代码	在缺省情况下启用
全部	TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Yes
全部	TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Yes
全部	TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Yes

表 78: 来自 *IBM MQ 9.2.0* 的 *CipherSpecs* 订单 *(继续)*				
平台	**CipherSpec**	**协议**	**十六进制代码**	**在缺省情况下启用**
ALW	TLS_AES_128_CCM_SHA256	TLS 1.3	1304	Yes
ALW	TLS_AES_128_CCM_8_SHA256	TLS 1.3	1305	Yes
全部	TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Yes
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	C02C	Yes
全部	ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Yes
全部	TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Yes
全部	ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Yes
全部	ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Yes
全部	TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Yes
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	C02B	Yes
全部	ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Yes
全部	TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Yes
全部	ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Yes
全部	ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Yes
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	C008	否
Multi	ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	C012	否
ALW	TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	0005	否

表 78: 来自 *IBM MQ 9.2.0* 的 *CipherSpecs* 订单 *(继续)*				
平台	**CipherSpec**	**协议**	**十六进制代码**	**在缺省情况下启用**
ALW	ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	C007	否
Multi	ECDHE_RSA_RC4_128_SHA256	TLS 1.2	C011	否
全部	TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	否
ALW	ECDHE_ECDSA_NULL_SHA256	TLS 1.2	C006	否
Multi	ECDHE_RSA_NULL_SHA256	TLS 1.2	C010	否
ALW	TLS_RSA_WITH_NULL_NULL	TLS 1.2	0000	否
ALW z/OS	TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	否
ALW z/OS	TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	否
IBM i	AES_SHA_US	TLS 1.0	002E	否
全部	TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	否
全部	TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	否
IBM i	TLS_RSA_WITH_RC4_128_MD5	TLS 1.0	0004	否
全部	TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	否
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5	TLS 1.0	0003	否
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5	TLS 1.0	0006	否
IBM i	TLS_RSA_WITH_NULL_SHA	TLS 1.0	0002	否
IBM i	TLS_RSA_WITH_NULL_MD5	TLS 1.0	0001	否
全部	TRIPLE_DES_SHA_US	SSL V3	000A	否
全部	RC4_SHA_US	SSL V3	0005	否

表 78: 来自 IBM MQ 9.2.0 的 CipherSpecs 订单 (继续)				
平台	**CipherSpec**	**协议**	**十六进制代码**	**在缺省情况下启用**
全部	RC4_MD5_US	SSL V3	0004	否
全部	DES_SHA_EXPORT	SSL V3	0009	否
全部	RC4_MD5_EXPORT	SSL V3	0003	否
全部	RC2_MD5_EXPORT	SSL V3	0006	否
全部	NULL_SHA	SSL V3	0002	否
全部	NULL_MD5	SSL V3	0001	否
ALW	FIPS_WITH_3DES_EDE_CBC_SHA	SSL V3	FEFF	否
ALW	RC4_56_SHA_EXPORT1024	SSL V3	0064	否
ALW	DES_SHA_EXPORT1024	SSL V3	0062	否
ALW	FIPS_WITH_DES_CBC_SHA	SSL V3	FEFE	否

此列表是通过使用 IBM MQ 在 z/OS 上使用的加密库提供的缺省列表对协议进行排序而构造的，并且在 z/OS 和分布式平台之间保持一致。

更改顺序

如果需要其他顺序，那么可以使用 IBM MQ for Multiplatforms z/OS 或 IBM MQ for z/OS 上的 TransportSecurity 节， 上 SSL 节的 **AllowedCipherSpecs** 属性以及以下规则来提供 CipherSpecs 的新顺序:

- 无论它们在列表中的位置如何，都将始终使用更高的协议版本。
- 如果在列表中提供，那么将重新启用任何已禁用的 CipherSpecs 。
- TLS 服务器的列表顺序具有比 TLS 客户机更高的优先级。
- 启用 TLS 1.3 时，不支持某些 CipherSpecs 。

例如，在 IBM MQ for Multiplatforms 上，如果在队列管理器上配置了以下内容:

```
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 在 IBM MQ for z/OS 上，如果在队列管理器上配置了以下内容:

```
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
```

然后:

- 与 ANY_TLS12 连接的客户机可能会使用 TLS 1.2 CipherSpec TLS_RSA_WITH_AES_128_GCM_SHA256。
- 与 ANY_TLS12_OR_HIGHER 连接的客户机可能使用 TLS 1.3 CipherSpec TLS_AES_128_GCM_SHA256 (假定客户机支持 TLS 1.3)。
- 使用 TLS 1.0 CipherSpec TLS_RSA_WITH_AES_256_CBC_SHA 连接的客户机将使用该 CipherSpec。

IBM MQ 的先前版本

在 IBM MQ 9.2.0 之前，使用了 CipherSpecs 的以下顺序:

表 *79: CipherSpecs* 订购前 *IBM MQ 9.2.0*

平台	CipherSpec	协议	在缺省情况下启用
ALW z/OS	TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	否
IBM i	AES_SHA_US	TLS 1.0	否
ALW z/OS	TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	否
全部	RC4_SHA_US	SSL V3	否
全部	TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	否
全部	RC4_MD5_US	SSL V3	否
IBM i	TLS_RSA_WITH_RC4_128_MD5	TLS 1.0	否
全部	TRIPLE_DES_SHA_US	SSL V3	否
全部	TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	否
ALW	DES_SHA_EXPORT1024	SSL V3	否
全部	RC4_56_SHA_EXPORT1024	SSL V3	否
全部	RC4_MD5_EXPORT	SSL V3	否
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5	TLS 1.0	否
全部	RC2_MD5_EXPORT	SSL V3	否
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5	TLS 1.0	否
全部	DES_SHA_EXPORT	SSL V3	否
全部	TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	否
全部	NULL_SHA	SSL V3	否
IBM i	TLS_RSA_WITH_NULL_SHA	TLS 1.0	否
全部	NULL_MD5	SSL V3	否
IBM i	TLS_RSA_WITH_NULL_MD5	TLS 1.0	否

表 79: CipherSpecs 订购前 IBM MQ 9.2.0 (继续)

平台	CipherSpec	协议	在缺省情况下启用
ALW	FIPS_WITH_DES_CBC_SHA	SSL V3	否
ALW	FIPS_WITH_3DES_EDE_CBC_SHA	SSL V3	否
全部	TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	Yes
全部	TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	Yes
全部	TLS_RSA_WITH_NULL_SHA256	TLS 1.2	否
全部	TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	Yes
全部	TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	Yes
ALW	ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	否
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	否
Multi	ECDHE_RSA_RC4_128_SHA256	TLS 1.2	否
Multi	ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	否
全部	ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	Yes
全部	ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	Yes
全部	ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	Yes
全部	ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	Yes
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	Yes
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	Yes
全部	ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	Yes
全部	ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	Yes
Multi	ECDHE_RSA_NULL_SHA256	TLS 1.2	否
ALW	ECDHE_ECDSA_NULL_SHA256	TLS 1.2	否

表 79: CipherSpecs 订购前 IBM MQ 9.2.0 (继续)			
平台	**CipherSpec**	**协议**	**在缺省情况下启用**
ALW	TLS_RSA_WITH_NULL_NULL	TLS 1.2	否
ALW	TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	否
Multi	TLS_AES_128_GCM_SHA256	TLS 1.3	Yes
Multi	TLS_AES_256_GCM_SHA384	TLS 1.3	Yes
Multi	TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	Yes
ALW	TLS_AES_128_CCM_SHA256	TLS 1.3	Yes
ALW	TLS_AES_128_CCM_8_SHA256	TLS 1.3	Yes

要点: 从 23rd 开始，以下 AllowedCipherSpecs 属性仅启用缺省情况下当前启用的 CipherSpecs 。 但是，您应该使用当前数据验证以下 AllowedCipherSpecs 属性启用的 CipherSpecs ，以确保自此日期以来已弃用的 CipherSpecs 不会无意中重新启用。

如果需要返回到 CipherSpecs 的此顺序，那么可以使用以下 **AllowedCipherSpecs** SSL/TransportSecurity 节属性值来执行此操作:

```
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
```

在 IBM MQ for Multiplatforms 上提供已订购和已启用的 CipherSpecs 的定制列表

Multi

您可以使用 ALW **AMQ_ALLOWED_CIPHERS** 环境变量 或 `.ini` 文件的 **AllowedCipherSpecs** SSL 节属性，提供一组已启用的 CipherSpecs (按您的首选顺序) ，以便与 IBM MQ 通道配合使用。 出于以下任一原因，您可能希望使用此设置:

- 要限制 IBM MQ 侦听器接受入局通道启动请求，除非它们使用其中一个指定的 CipherSpecs。
- 更改 TLS 握手中使用的 CipherSpecs 的优先级顺序。

此功能可用于控制 ANY* CipherSpecs 中包含的 CipherSpecs 。

AMQ_ALLOWED_CIPHERS 环境变量或 **AllowedCipherSpecs** SSL 节属性接受:

- 单个 CipherSpec 名称。
- 要重新启用的 CipherSpec 名称的逗号分隔列表。
- ALL 的特殊值，表示所有 CipherSpecs。

注: 不应启用 **ALL** CipherSpecs，因为这将启用 SSL 3.0 和 TLS 1.0 协议以及大量弱加密算法。

如果配置了此设置，那么它将覆盖缺省 CipherSpec 列表，并导致 IBM MQ 忽略弱密码弃用设置 (请参阅以下内容):

- IBM MQ 侦听器仅接受使用其中一个指定的 CipherSpecs 的 SSL/TLS 建议。
- IBM MQ 通道仅允许空白 SSLCIPH 值或其中一个指定的 CipherSpecs。

- **runmqsc** 选项卡完成 SSLCIPH 值将完成值限制为名称 CipherSpecs 之一。

例如，如果只想允许定义/更改通道以及允许侦听器接受 ECDHE_RSA_AES_128_GCM_SHA256 或 ECDHE_ECDSA_AES_256_GCM_SHA384，那么可以在 qm.ini 文件中设置以下内容:

```
SSL:
  AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256, ECDHE_ECDSA_AES_256_GCM_SHA384
```

此外，此列表中的 CipherSpecs 将用于确定 TLS 握手期间使用的 CipherSpecs 的优先级。 例如，如果指定 TLS_RSA_WITH_AES_128_CBC_SHA256，TLS_RSA_WITH_AES_256_CBC_SHA256 列表，那么在握手期间可能会选择 TLS_RSA_WITH_AES_128_CBC_SHA256 CipherSpec 通过 TLS_RSA_WITH_AES_256_CBC_SHA256 CipherSpec (如果客户机连接同时指定了这两个 CipherSpecs，即使用 ANY_TLS12 连接的客户机)。

请注意，可以使用 java.security 文件设置来限制 AMQP 或 MQTT 通道使用的密码。

在 IBM MQ for z/OS 上提供已订购和已启用的 CipherSpecs 的定制列表

z/OS

您可以使用 QMINI 数据集的 **AllowedCipherSpecs** TransportSecurity 节属性提供一组已启用的备用 CipherSpecs，并按您的首选顺序与 IBM MQ 通道配合使用。 出于以下任一原因，您可能希望执行此操作:

- 要限制 IBM MQ 侦听器接受入局通道启动请求，除非它们使用其中一个指定的 CipherSpecs。
- 更改 TLS 握手中使用的 CipherSpecs 的优先级顺序。

您可以使用此功能来控制 ANY* CipherSpecs 中包含的 CipherSpecs。 **AllowedCipherSpecs** 属性接受:

- 单个 CipherSpec 名称。
- 要重新启用的 CipherSpec 名称的逗号分隔列表。
- ALL 的特殊值，表示所有 CipherSpecs。

  注: 不应启用 **ALL** CipherSpecs，因为这将启用 SSL 3.0 和 TLS 1.0 协议以及大量弱加密算法。 如果您确实配置了此设置，那么它将覆盖缺省 CipherSpec 列表，并导致 IBM MQ 忽略弱密码弃用设置; 请参阅 第 385 页的『在 z/OS 上启用不推荐的 CipherSpecs』。

IBM MQ 侦听器仅接受使用其中一个指定的 CipherSpecs 和 IBM MQ 通道的 SSL/TLS 建议，仅允许空白 SSLCIPH 值或其中一个指定的 CipherSpecs。

例如，如果您只想允许定义/改变通道并允许侦听器接受 ECDHE_RSA_AES_128_GCM_SHA256 或 ECDHE_RSA_AES_256_GCM_SHA384，那么可以设置以下内容:

```
TransportSecurity:
  AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256,
                     ECDHE_RSA_AES_256_GCM_SHA384
```

此外，此列表中的 CipherSpecs 用于确定 TLS 握手期间使用的 CipherSpecs 的优先级。 例如，如果指定 TLS_RSA_WITH_AES_128_CBC_SHA256，TLS_RSA_WITH_AES_256_CBC_SHA256 的列表，那么在握手期间，可能会选择 TLS_RSA_WITH_AES_128_CBC_SHA256 CipherSpec 通过 TLS_RSA_WITH_AES_256_CBC_SHA256 CipherSpec (如果客户机连接时指定了这两个 CipherSpecs

Deprecated 不推荐使用的 CipherSpecs

不推荐使用的 CipherSpecs 列表，必要时可以与 IBM MQ 配合使用。

下表列出了可用于 IBM MQ TLS 支持的不推荐使用的 CipherSpecs。

表 80: 不推荐使用的 *CipherSpec*，您可以重新启用以用于 *IBM MQ*

平台支持 第 384 页的『1』	CipherSpec 名称	十六进制代码	使用的协议	数据完整性	加密算法(加密位)	FIPS 第 384 页的『2』	套件 B	不推荐使用时更新
适用于 SSL 3.0 的 CipherSpec								
IBM i	AES_SHA_US 第 384 页的『3』	002F	SSL 3.0	SHA-1	AES (128)	否	否	9.0.0.0
全部	DES_SHA_EXPORT 第 384 页的『3』 第 384 页的『4』 第 384 页的『5』	0009	SSL 3.0	SHA-1	DES (56)	否	否	9.0.0.0
ALW	DES_SHA_EXPORT1024 第 384 页的『3』 第 384 页的『6』	0062	SSL 3.0	SHA-1	DES (56)	否	否	9.0.0.0
ALW	FIPS_WITH_DES_CBC_SHA 第 384 页的『3』	FEFE	SSL 3.0	SHA-1	DES (56)	否第 384 页的『7』	否	9.0.0.0
ALW	FIPS_WITH_3DES_EDE_CBC_SHA 第 384 页的『3』	FEFF	SSL 3.0	SHA-1	3DES (168)	否第 384 页的『8』	否	9.0.0.1 和 9.0.1
全部	NULL_MD5 第 384 页的『3』	0001	SSL 3.0	MD5	None	否	否	9.0.0.1
全部	NULL_SHA 第 384 页的『3』	0002	SSL 3.0	SHA-1	None	否	否	9.0.0.1
全部	RC2_MD5_EXPORT 第 384 页的『3』 第 384 页的『4』 第 384 页的『5』	0006	SSL 3.0	MD5	RC2 (40)	否	否	9.0.0.0
全部	RC4_MD5_EXPORT 第 384 页的『4』 第 384 页的『3』	0003	SSL 3.0	MD5	RC4 (40)	否	否	9.0.0.0
全部	RC4_MD5_US 第 384 页的『3』	0004	SSL 3.0	MD5	RC4 (128)	否	否	9.0.0.0
全部	RC4_SHA_US 第 384 页的『3』 第 384 页的『5』	0005	SSL 3.0	SHA-1	RC4 (128)	否	否	9.0.0.0
ALW	RC4_56_SHA_EXPORT1024 第 384 页的『3』 第 384 页的『6』	0064	SSL 3.0	SHA-1	RC4 (56)	否	否	9.0.0.0
全部	TRIPLE_DES_SHA_US 第 384 页的『3』 第 384 页的『5』	000A	SSL 3.0	SHA-1	3DES (168)	否	否	9.0.0.1 和 9.0.1
TLS 1.0 的 CipherSpec								
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5 第 384 页的『3』	0006	TLS 1.0	MD5	RC2 (40)	否	否	9.0.0.0
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5 第 384 页的『3』 第 384 页的『4』	0003	TLS 1.0	MD5	RC4 (40)	否	否	9.0.0.0
全部	TLS_RSA_WITH_DES_CBC_SHA 第 384 页的『3』	0009	TLS 1.0	SHA-1	DES (56)	否第 384 页的『9』	否	9.0.0.0

表 80: 不推荐使用的 CipherSpec，您可以重新启用以用于 IBM MQ (继续)								
平台支持 第 384 页的『1』	**CipherSpec 名称**	**十六进制代码**	**使用的协议**	**数据完整性**	**加密算法 (加密位)**	**FIPS** 第 384 页的『2』	**套件 B**	**不推荐使用时更新**
IBM i	TLS_RSA_WITH_NULL_MD5 第 384 页的『3』	0001	TLS 1.0	MD5	None	否	否	9.0.0.1
IBM i	TLS_RSA_WITH_NULL_SHA 第 384 页的『3』	0002	TLS 1.0	SHA-1	None	否	否	9.0.0.1
IBM i	TLS_RSA_WITH_RC4_128_MD5 第 384 页的『3』	0004	TLS 1.0	MD5	RC4 (128)	否	否	9.0.0.0
ALW z/OS	TLS_RSA_WITH_AES_128_CBC_SHA 第 384 页的『10』	002F	TLS 1.0	SHA-1	AES (128)	Yes	否	9.0.5
ALW z/OS	TLS_RSA_WITH_AES_256_CBC_SHA 第 384 页的『6』 第 384 页的『10』	0035	TLS 1.0	SHA-1	AES (256)	Yes	否	9.0.5
全部	TLS_RSA_WITH_3DES_EDE_CBC_SHA	000A	TLS 1.0	SHA-1	3DES (168)	Yes	否	9.0.0.1 和 9.0.1
适用于 TLS 1.2 的 CipherSpec								
ALW	ECDHE_ECDSA_NULL_SHA256 第 384 页的『3』	C006	TLS 1.2	SHA-1	None	否	否	9.0.0.1
ALW	ECDHE_ECDSA_RC4_128_SHA256 第 384 页的『3』	C007	TLS 1.2	SHA-1	RC4 (128)	否	否	9.0.0.0
ALW IBM i	ECDHE_RSA_NULL_SHA256 第 384 页的『3』	C010	TLS 1.2	SHA-1	None	否	否	9.0.0.1
ALW IBM i	ECDHE_RSA_RC4_128_SHA256 第 384 页的『3』	C011	TLS 1.2	SHA-1	RC4 (128)	否	否	9.0.0.0
ALW	TLS_RSA_WITH_NULL_NULL 第 384 页的『3』	0000	TLS 1.2	None	None	否	否	9.0.0.1
全部	TLS_RSA_WITH_NULL_SHA256 第 384 页的『3』	003B	TLS 1.2	SHA-256	None	否	否	9.0.0.1
ALW	TLS_RSA_WITH_RC4_128_SHA256 第 384 页的『3』	0005	TLS 1.2	SHA-1	RC4 (128)	否	否	9.0.0.0
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	C0008	TLS 1.2	SHA-1	3DES (168)	Yes	否	9.0.0.1 和 9.0.1
ALW IBM i	ECDHE_RSA_3DES_EDE_CBC_SHA256	C012	TLS 1.2	SHA-1	3DES (168)	Yes	否	9.0.0.1 和 9.0.1

表 80: 不推荐使用的 *CipherSpec*，您可以重新启用以用于 *IBM MQ (继续)*

平台支持 第 384 页的『1』	CipherSpec 名称	十六进制代码	使用的协议	数据完整性	加密算法 (加密位)	FIPS 第 384 页的『2』	套件 B	不推荐使用时更新

注意:

1. 有关每个平台图标所涵盖的平台的列表，请参阅 产品文档中使用的图标。
2. 指定 CipherSpec 是否在经 FIPS 认证的平台上经 FIPS 认证。 请参阅美国联邦信息处理标准 (FIPS)，以获取 FIPS 的解释。
3. ALW 启用 TLS 1.3 时，将禁用这些 CipherSpecs（通过 `qm.ini` 中的 AllowTLSV13 属性）。

   z/OS 在 IBM MQ for z/OS 9.2.0 或更高版本上创建的队列管理器缺省情况下启用了 TLS 1.3，而这将禁用这些 CipherSpec。 如果需要，您可以通过关闭 TLS V1.3 来启用这些 CipherSpec。 通过将 **AllowTLSV13**=*FALSE* 添加到队列管理器 JCL 中 QMINI 数据集的 TransportSecurity 节来完成此操作。 缺省情况下，从较早版本迁移至 IBM MQ for z/OS 9.2.0 的队列管理器未启用 TLS 1.3，因此已启用这些 CipherSpec。
4. 最大握手密钥大小是 512 位。 如果在 SSL 握手期间交换的两个证书中有一个密钥大小超出 512 位，那么在握手期间会生成一个临时的 512 位密钥以供使用。
5. 这些 CipherSpec 不再受 IBM MQ classes for Java 或 IBM MQ classes for JMS 支持。 有关更多信息，请参阅 IBM MQ classes for Java 中的 SSL/TLS CipherSpec 和 CipherSuite 或 IBM MQ classes for JMS 中的 SSL/TLS CipherSpec 和 CipherSuite。
6. 握手密钥大小是 1024 位。
7. Deprecated 在 2007 年 5 月 19 日之前，该 CipherSpec 经 FIPS 140-2 认证。 名称 `FIPS_WITH_DES_CBC_SHA` 是历史记录，反映了此 CipherSpec 先前（但已不再）符合 FIPS 标准的事实。 不推荐使用此 CipherSpec。
8. Deprecated 名称 `FIPS_WITH_3DES_EDE_CBC_SHA` 是历史记录，反映了此 CipherSpec 先前（但已不再）符合 FIPS 标准的事实。 不推荐使用此 CipherSpec。
9. 在 2007 年 5 月 19 日之前，该 CipherSpec 经 FIPS 140-2 认证。
10. 重新启用仅这些密码规范不需要使用 CSQXWEAK DD 语句。

在 IBM MQ for Multiplatforms 上启用不推荐的 CipherSpecs

Multi

缺省情况下，不允许您在通道定义上指定不推荐的 CipherSpec 。 如果尝试在 IBM MQ for Multiplatforms 上指定不推荐的 CipherSpec ，那么会收到消息 AMQ8242: SSLCIPH 定义错误，并且 PCF 返回 `MQRCCF_SSL_CIPHER_SPEC_ERROR`。

不能使用不推荐的 CipherSpec 启动通道。 如果尝试使用不推荐的 CipherSpec 执行此操作，那么系统会将 `MQCC_FAILED` (2)与 **Reason** `MQRC_SSL_INITIALIZATION_ERROR` (2393) 一起返回到客户机。

您可以通过设置环境变量 **AMQ_SSL_WEAK_CIPHER_ENABLE**，在服务器上的运行时重新启用一个或多个不推荐使用的 CipherSpecs 以定义通道。

AMQ_SSL_WEAK_CIPHER_ENABLE 环境变量接受:

- 单个 CipherSpec 名称，或
- 要重新启用的 CipherSpec 名称的逗号分隔列表，或者
- `ALL` 的特殊值，表示所有 CipherSpecs。

注意: 虽然 `ALL` 是有效选项，但您应该 **仅** 在企业需要的特定情况下使用此选项，因为重新启用 `ALL` CipherSpecs 会启用 SSL 3.0 和 TLS 1.0 协议以及大量弱加密算法。

例如，如果要重新启用 ECDHE_RSA_RC4_128_SHA256，请设置以下环境变量:

```
export AMQ_SSL_WEAK_CIPHER_ENABLE=ECDHE_RSA_RC4_128_SHA256
```

或者，通过设置以下内容来更改 qm.ini 文件中的 SSL 节:

```
SSL:
  AllowTLSV1=Y
  AllowWeakCipherSpec=ECDHE_RSA_RC4_128_SHA256
```

在 z/OS 上启用不推荐的 CipherSpecs

z/OS

缺省情况下，不允许您在通道定义上指定不推荐的 CipherSpec 。 如果尝试在 z/OS 上指定不推荐的 CipherSpec ，那么将收到消息 CSQM102E，消息 CSQX616E 或 CSQX674E。

如果接收到其中任何一条消息，并且您的企业需要重新启用弱 CipherSpecs 的使用，请遵循本节中列出的指示信息。

注意: 在以下指示信息中，要使哑元定义 (DD) 语句生效， SSLTASKS 必须是非零值。 如果这需要对 SSLTASKS 进行更改，那么必须重新启动通道启动程序。

在 IBM MQ for z/OS 上，当前控制弱或弱 CipherSpecs 的方法如下所示:

- 如果要重新启用弱 CipherSpecs 的使用，请通过将名为 CSQXWEAK 的哑元数据定义 (DD) 语句添加到通道启动程序 JCL 来执行此操作。 如果单独指定，那么仅启用与 TLS 1.2 协议关联的弱 CipherSpecs ; 例如:

  ```
  //CSQXWEAK DD DUMMY
  ```

  注: 并非所有不推荐使用的 CipherSpecs 都需要使用此 DD 语句，请参阅上表中的注释 10。
- 如果要重新启用 SSLv3 CipherSpecs 的使用，那么还可以通过将名为 CSQXSSL3 的伪 DD 语句添加到通道启动程序 JCL 来执行此操作。 所有 SSLv3 CipherSpecs 都被视为 **Weak**，因此您还必须指定 CSQXWEAK:

  ```
  //CSQXSSL3 DD DUMMY
  ```
- 如果要重新启用不推荐使用的 TLS V1 CipherSpecs，请通过将名为 TLS10ON (将 TLS V1.0 ON) 的伪 DD 语句添加到通道启动程序 JCL 来执行此操作。 如果单独指定，那么这将启用与 TLS 1.0 协议关联的强 CipherSpecs :

  ```
  //TLS10ON DD DUMMY
  ```

  如果与 CSQXWEAK 一起指定，那么还会启用与 TLS 1.0 关联的 **Weak** CipherSpecs 。
- 如果要显式关闭不推荐使用的 TLS V1 CipherSpecs，请通过向通道启动程序 JCL 添加名为 TLS10OFF (关闭 TLS V1.0 OFF) 的伪 DD 语句来执行此操作; 例如:

  ```
  //TLS10OFF DD DUMMY
  ```

如果只想使用 **System SSL** 缺省密码规范列表中列出的密码规范与侦听器协商，那么需要在 CHINIT JCL 中定义以下 DD 语句:

```
JCL: //GSKDCIPS DD DUMMY
```

要点: 对于 IBM MQ for z/OS 9.2.0 和更高版本，在通道启动程序启动期间显示消息时，将考虑先前列示的 DD 卡和 **AllowTLSV13** 的值，以指示哪些协议已启用，哪些协议未启用。 因此，即使指定了先前列出的某个 DD 卡，也可能意味着由于这些设置的组合，无法使用另一个协议来启用某个协议。 例如，如果启用了 TLS 1.3 ，那么不允许协议 SSL 3.0 。

有一些备用机制可用于强制重新启用弱 CipherSpecs 和 SSLv3 支持 (如果数据定义更改不合适)。 请联系 IBM 服务人员以获取更多信息。

相关概念

第 39 页的『IBM MQ 中的数字证书和 CipherSpec 兼容性』

本主题通过概述 CipherSpecs 与 IBM MQ 中的数字证书之间的关系，提供有关如何为安全策略选择相应 CipherSpecs 和数字证书的信息。

相关参考

DEFINE CHANNEL

ALTER CHANNEL

别名 CipherSpec 设置之间的关系

此信息描述客户机和服务器配置中别名 CipherSpecs 的不同组合的预期行为。 其中，客户机是指发起通信的实体，例如客户机应用程序或队列管理器发送方通道，而服务器是指从客户机接收通信的实体，例如服务器连接通道或接收方通道。

最小协议与固定协议 CipherSpecs

IBM MQ 支持两种不同类型的 CipherSpecs:

最小协议
: 最低协议 CipherSpecs 是未设置上限的协议，例如 ANY， ANY_TLS12_OR_HIGHER 或 ANY_TLS13_OR_HIGHER。

固定协议
: 固定协议 CipherSpecs 是标识特定协议 (例如 ANY_TLS12 和 ANY_TLS13) 或特定算法 (例如 ECDHE_ECDSA_3DES_EDE_CBC_SHA256) 的协议。

在所有平台上都支持最低协议和固定协议 CipherSpecs 。

为了在维护安全性的同时最大程度地简化配置，建议在通道两侧使用 **最低协议** CipherSpecs 。 这允许您的通信在双方都支持新版本时自动支持并使用更高的 TLS 协议版本，而无需更改任何一方的配置。

在启动端使用 **最低协议** CipherSpec ，但在接收端使用 **固定协议** CipherSpec 可能会导致连接被拒绝，并且

- Multi 正在发出消息 AMQ9631 和 AMQ9641 。
- z/OS 正在发出消息 CSQX631E 和 CSQX641E 。

下表显示了不同别名 CipherSpec 设置与预期结果之间的关系。 第 386 页的表 81 显示了在客户机和/或服务器上未启用 TLS 1.3 时的预期行为。 第 387 页的表 82 显示了在客户机和服务器上启用 TLS 1.3 时的预期行为。 在这两种情况下，客户机的 CipherSpecs 显示在表的 Y 轴上，服务器的 CipherSpecs 显示在表的 X 轴上。

注: 在下表中，标记为 可能失败 的单元格指示当您为连接的一部分指定 **最低协议** CipherSpec ，为另一部分指定特定 (**固定协议**) CipherSpec 时可能会发生冲突。

例如，假设客户机和服务器设置为使用 ANY CipherSpec，而服务器通道设置为使用特定的 CipherSpec:

- 如果客户机和服务器支持的最强 CipherSpec 与通道上配置的特定 CipherSpec 匹配，那么 TLS 握手会成功解析。
- 但是，如果客户机和服务器都支持更强的 CipherSpec ，那么 TLS 握手会解析为使用此功能，即使它与通道上指定的 CipherSpec 不匹配， TLS 握手也会失败。

表 *81:* 在客户机和/或服务器上未启用 *TLS 1.3* 时的预期行为

	服务器			
客户机	**特定 TLS 1.2 CipherSpec**	**ANY**	**ANY_ TLS12**	**ANY_TLS12_ OR_ 上级**
特定 TLS 1.2 CipherSpec	连接数	连接数	连接数	连接数
any	可能失败	连接数	连接数	连接数

表 *81:* 在客户机和/或服务器上未启用 *TLS 1.3* 时的预期行为 *(继续)*				
	服务器			
客户机	**特定 TLS 1.2 CipherSpec**	**ANY**	**ANY_ TLS12**	**ANY_TLS12_ OR_ 上级**
ANY_ TLS12	可能失败	连接数	连接数	连接数
ANY_TLS12_ OR_ 上级	可能失败	连接数	连接数	连接数

表 *82:* 在客户机和服务器上启用 *TLS 1.3* 时的预期行为							
	服务器						
客户机	**特定 TLS 1.2 CipherSpec**	**特定 TLS 1.3 CipherSpec**	**ANY**	**ANY_TLS 12**	**ANY_TLS 13**	**ANY_TLS12_ 或更高版本**	**ANY_TLS13_ 或更高版本**
特定 TLS 1.2 CipherSpec	连接数	**失败**	连接数	连接数	**失败**	连接数	**失败**
特定 TLS 1.3 CipherSpec	**失败**	连接数	连接数	**失败**	连接数	连接数	连接数
any	**失败**	可能失败	连接数	**失败**	连接数	连接数	连接数
ANY_TLS12	可能失败	**失败**	连接数	连接数	**失败**	连接数	**失败**
ANY_TLS13	**失败**	可能失败	连接数	**失败**	连接数	连接数	连接数
ANY_TLS12_ OR_上级	**失败**	可能失败	连接数	**失败**	连接数	连接数	连接数
ANY_TLS13_ OR_上级	**失败**	可能失败	连接数	**失败**	连接数	连接数	连接数

相关概念

第 39 页的『IBM MQ 中的数字证书和 CipherSpec 兼容性』
本主题通过概述 CipherSpecs 与 IBM MQ 中的数字证书之间的关系，提供有关如何为安全策略选择相应 CipherSpecs 和数字证书的信息。

第 19 页的『CipherSpecs 和 CipherSuites』
加密安全协议必须同意安全连接使用的算法。 CipherSpecs 和 CipherSuites 定义算法的特定组合。

第 368 页的『启用 CipherSpecs』
通过在 **DEFINE CHANNEL** 或 **ALTER CHANNEL** MQSC 命令中使用 **SSLCIPH** 参数来启用 CipherSpec 。

相关任务

迁移现有安全性配置以使用 ANY_TLS12_OR_HIGHER CipherSpec

使用 IBM MQ Explorer 获取有关 CipherSpecs 的信息

您可以使用 IBM MQ Explorer 来显示 CipherSpecs 的描述。

使用以下过程来获取有关 第 368 页的『启用 CipherSpecs』中的 CipherSpecs 的信息:

1. 打开 IBM MQ Explorer 并展开**队列管理器**文件夹。
2. 确保已启动队列管理器。
3. 选择要使用的队列管理器，然后单击 **通道**。
4. 右键单击要使用的通道，然后选择 **属性**。

5. 选择 **SSL** 属性页面。
6. 从列表中选择要使用的 CipherSpec 。 描述将显示在列表下方的窗口中。

z/OS IBM i 指定 CipherSpecs 的替代方法

对于操作系统提供 TLS 支持的平台，您的系统可能支持 第 368 页的『启用 CipherSpecs』中未包含的新 CipherSpecs 。

您可以使用 SSLCIPH 参数指定新的 CipherSpec ，但您提供的值取决于您的平台。 在所有情况下，该规范都必须对应于 TLS CipherSpec ，该规范有效且受系统正在运行的 TLS 版本支持。

注: 此部分不适用于 AIX, Linux, and Windows 系统，因为 IBM MQ 产品随附了 CipherSpecs ，因此新的 CipherSpecs 在装运后不可用。

IBM i IBM i

表示十六进制值的双字符字符串。

有关允许的值的更多信息，请参阅 设置安全会话的字符信息的 "使用说明" 部分中的第 3 点。

注意: 您不应在 **SSLCIPH** 中指定十六进制密码值，因为从将使用的密码的值看不清楚，而要使用的协议的选择是不确定的。 使用十六进制密码值可能会导致 CipherSpec 不匹配错误。

可以使用 **CHGMQMCHL** 或 **CRTMQMCHL** 命令来指定值，例如:

```
CRTMQMCHL CHLNAME(' channel name ') SSLCIPH(' hexadecimal value ')
```

您还可以使用 **ALTER QMGR** MQSC 命令来设置 **SSLCIPH** 参数。

z/OS z/OS

表示十六进制值的四字符字符串。 十六进制代码对应于 TLS 协议中定义的值。

有关更多信息，请参阅 密码套件定义 ，其中包含所有受支持的 TLS 1.0， TLS 1.2 和 TLS 1.3 密码规范的列表，格式为 4 位十六进制代码。

注: Deprecated 要使用弱 CipherSpec 或属于不推荐使用的协议 (例如 SSL V3.0 或 TLS 1.0) 的 CipherSpec ，必须在通道启动程序启动 JCL 中指定相关 DD 卡。 请参阅 第 381 页的『不推荐使用的 CipherSpecs』 以获取更多信息。

IBM MQ 集群的注意事项

对于 IBM MQ 集群，在 第 368 页的『启用 CipherSpecs』中使用 CipherSpec 名称是最安全的。 如果使用备用规范，请注意该规范可能在其他平台上无效。 有关更多信息，请参阅第 423 页的『SSL/TLS 和集群』。

为 IBM MQ MQI client 指定 CipherSpec

您有三个选项可用于为 IBM MQ MQI client 指定 CipherSpec 。

这些选项如下:

- 使用通道定义表
- 在 MQCONNX 调用上使用 MQCD 结构中 MQCD_VERSION_7 或更高版本的 SSLCipherSpec 字段。
- 使用 Active Directory (在具有 Active Directory 支持的 Windows 系统上)

使用 IBM MQ classes for Java 和 IBM MQ classes for JMS 指定 CipherSuite

IBM MQ classes for Java 和 IBM MQ classes for JMS 以不同于其他平台的方式指定 CipherSuites 。

有关使用 IBM MQ classes for Java 指定 CipherSuite 的信息，请参阅 Java 的传输层安全性 (TLS) 支持

有关将 CipherSuite 与 IBM MQ classes for JMS 一起指定的信息，请参阅 将传输层安全性 (TLS) 与 IBM MQ classes for JMS 配合使用

为 IBM MQ.NET 指定 CipherSpec

对于 IBM MQ.NET，可以使用 MQEnvironment 类或使用连接属性散列表中的 MQC.SSL_CIPHER_SPEC_PROPERTY 来指定 CipherSpec。

有关为 .NET 非受管客户机指定 CipherSpec 的信息，请参阅 为非受管 .NET 客户机启用 TLS

有关为 .NET 受管客户机指定 CipherSpec 的信息，请参阅 对受管 .NET 客户机的 CipherSpec 支持

z/OS 将 AT-TLS 与 IBM MQ for z/OS 配合使用

应用程序透明传输层安全性 (AT-TLS) 为 z/OS 应用程序提供 TLS 支持，而无需这些应用程序实施 TLS 支持，甚至可以知道正在使用 TLS。 AT-TLS 仅在 z/OS 上可用。

AT-TLS 可与所有版本的 IBM MQ for z/OS 配合使用。

在将 AT-TLS 与 IBM MQ for z/OS 配合使用之前，请确保您了解所涉及的 第 392 页的『限制』。

要使用 应用程序透明传输层安全性，请定义包含一组规则的策略语句， z/OS Communications Server 使用这些规则来确定哪些 TCP/IP 连接以透明方式启用了 TLS。

IBM MQ for z/OS 具有自己的 TLS 实现，这要求通道使用受支持的 CipherSpec 配置 SSLCIPH 参数。

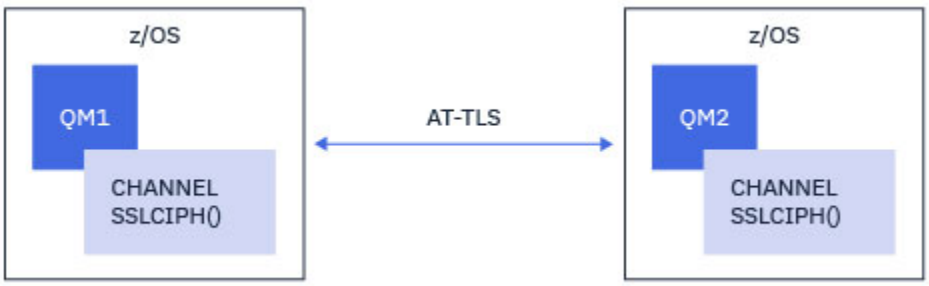
当决定在通道上启用 TLS 时， IBM MQ 管理员可以决定使用 AT-TLS 或 IBM MQ TLS。 决策通常基于 AT-TLS 是用于其他中间件，还是基于性能影响。 有关 AT-TLS 和 IBM MQ TLS 性能的基本比较，请参阅 MP16: Capacity Planning and Tuning for IBM MQ for z/OS。

方案

在以下场景中支持将 AT-TLS 与 IBM MQ 配合使用:

方案 1

在通道两侧使用 AT-TLS 的两个 IBM MQ for z/OS 队列管理器之间。 即，这两个通道都未指定 SSLCIPH 属性。 此方法可与任何消息通道配合使用。



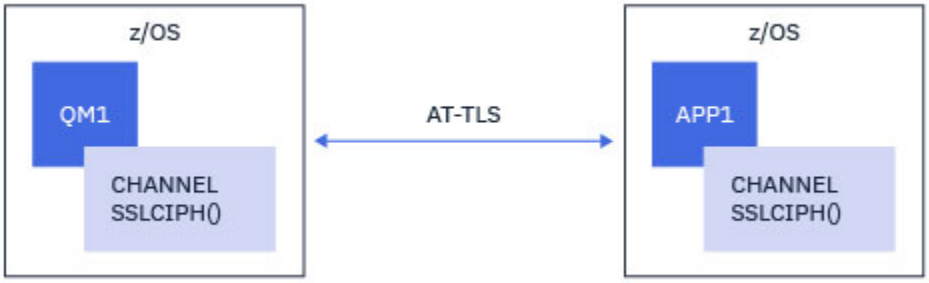
此方案的实现包括定义两个 AT-TLS 策略，一个用于通道的每一端。 这些策略与用于 方案 3 或 方案 4 的策略相同。

例如，如果正在将通道从使用名为 CipherSpec 的单个通道更改为使用 AT-TLS，那么出站通道将使用来自 第 393 页的『在出站通道上使用单个名为 CipherSpec 的 IBM MQ for Multiplatforms 队列管理器配置 AT-TLS』 的策略，入站通道将使用来自 第 400 页的『使用单个名为 CipherSpec 的队列管理器在来自 IBM MQ for Multiplatforms 队列管理器的入站通道上配置 AT-TLS』的策略。

如果正在将通道从使用别名 CipherSpec 更改为使用 AT-TLS，那么出站通道将使用来自 第 396 页的『在出站通道上使用别名 CipherSpecs 配置到 IBM MQ for Multiplatforms 队列管理器的 AT-TLS』 的策略，入站通道将使用来自 第 404 页的『使用别名 CipherSpec 在来自 IBM MQ for Multiplatforms 队列管理器的入站通道上配置 AT-TLS』的策略。

方案 2

在 IBM MQ for z/OS 队列管理器与在 z/OS 上运行的 IBM MQ Java 客户机应用程序之间，通道两侧都使用 AT-TLS。 即，服务器连接通道或客户机连接通道都未指定 SSLCIPH 属性。



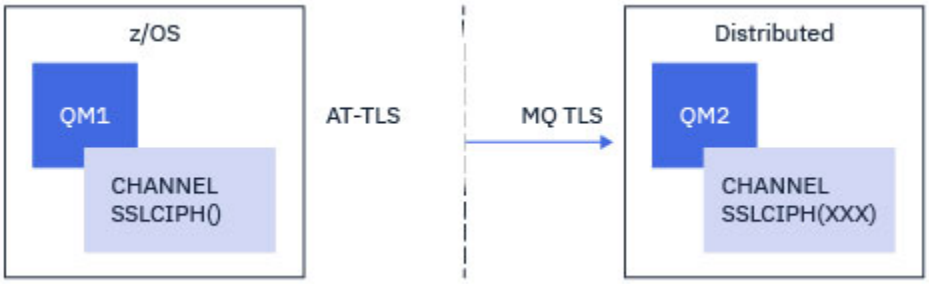
此方案的实现包括定义两个 AT-TLS 策略，一个用于通道的每一端。 这些策略与用于 方案 3 或 方案 4 的策略相同。

例如，如果正在将通道从使用名为 CipherSpec 的单个通道更改为使用 AT-TLS，那么客户机连接通道将使用来自 第 393 页的『在出站通道上使用单个名为 CipherSpec 的 IBM MQ for Multiplatforms 队列管理器配置 AT-TLS』 的策略，而服务器连接通道将使用来自 第 400 页的『使用单个名为 CipherSpec 的队列管理器在来自 IBM MQ for Multiplatforms 队列管理器的入站通道上配置 AT-TLS』 的策略。

如果正在将通道从使用别名 CipherSpec 更改为使用 AT-TLS，那么客户机连接通道将使用来自 第 396 页的『在出站通道上使用别名 CipherSpecs 配置到 IBM MQ for Multiplatforms 队列管理器的 AT-TLS』 的策略，而服务器连接通道将使用来自 第 404 页的『使用别名 CipherSpec 在来自 IBM MQ for Multiplatforms 队列管理器的入站通道上配置 AT-TLS』 的策略。

方案 3

在 IBM MQ for z/OS 队列管理器和正在运行的队列管理器 IBM MQ for Multiplatforms，其中 IBM MQ for z/OS 队列管理器使用 AT-TLS 和 IBM MQ for Multiplatforms 队列管理器使用 IBM MQTLS，通过使用单个命名的 SSLCIPH 属性指定 CipherSpec。 这适用于除集群发送方和集群接收方以外的所有消息通道类型。

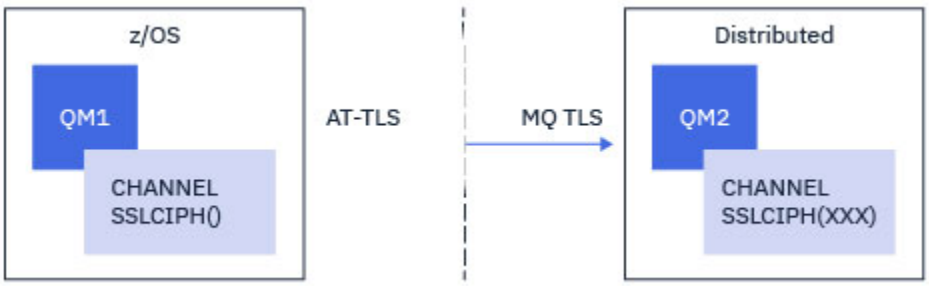


请参阅 第 393 页的『在出站通道上使用单个名为 CipherSpec 的 IBM MQ for Multiplatforms 队列管理器配置 AT-TLS』 以获取从 IBM MQ for z/OS 队列管理器到 IBM MQ for Multiplatforms 队列管理器的出站通道的 AT-TLS 配置示例，并参阅 第 400 页的『使用单个名为 CipherSpec 的队列管理器在来自 IBM MQ for Multiplatforms 队列管理器的入站通道上配置 AT-TLS』 以获取从 IBM MQ for Multiplatforms 队列管理器到 IBM MQ for z/OS 队列管理器的入站通道的 AT-TLS 配置示例。

当两个队列管理器都在 z/OS 上，但右侧的队列管理器尚未配置为使用 AT-TLS 时，可以使用相同的 AT-TLS 配置。

情况 4

在 IBM MQ for z/OS 队列管理器与 IBM MQ for Multiplatforms 上运行的队列管理器之间，其中 IBM MQ for z/OS 队列管理器使用 AT-TLS，而 IBM MQ for Multiplatforms 队列管理器使用 IBM MQ TLS，方法是指定具有别名 CipherSpec 的 SSLCIPH 属性。 这适用于除集群发送方和集群接收方以外的所有消息通道类型。

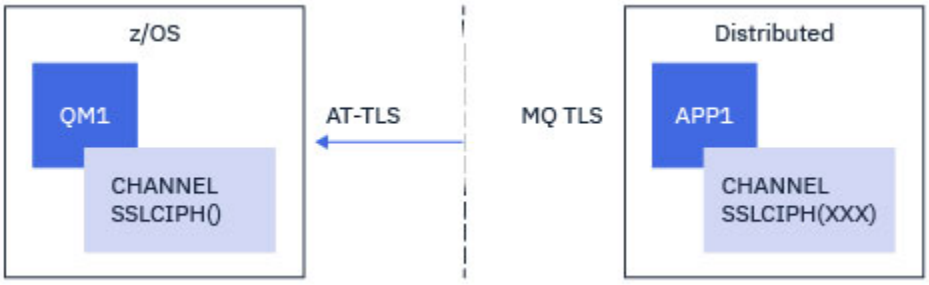


请参阅 第 396 页的『在出站通道上使用别名 CipherSpecs 配置到 IBM MQ for Multiplatforms 队列管理器的 AT-TLS』 以获取从 IBM MQ for z/OS 队列管理器到 IBM MQ for Multiplatforms 队列管理器的出站通道的 AT-TLS 配置示例，以及 第 404 页的『使用别名 CipherSpec 在来自 IBM MQ for Multiplatforms 队列管理器的入站通道上配置 AT-TLS』和 第 404 页的『使用别名 CipherSpec 在来自 IBM MQ for Multiplatforms 队列管理器的入站通道上配置 AT-TLS』 以获取从 IBM MQ for Multiplatforms 队列管理器到 IBM MQ for z/OS 队列管理器的入站通道的 AT-TLS 配置示例。

当两个队列管理器都在 z/OS 上，但右侧的队列管理器尚未配置为使用 AT-TLS 时，可以使用相同的 AT-TLS 配置。

情况 5

在 IBM MQ for z/OS 队列管理器与在 IBM MQ for Multiplatforms 上运行的客户机应用程序之间，其中 IBM MQ for z/OS 队列管理器使用 AT-TLS，而客户机应用程序通过指定名为 CipherSpec 的单个 SSLCIPH 属性来使用 IBM MQ TLS。

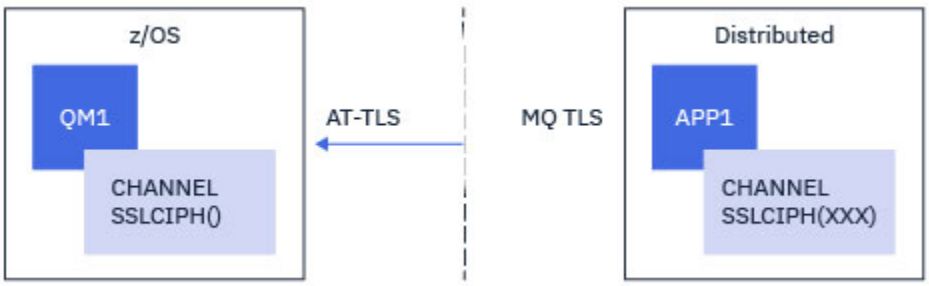


此方案需要单个 AT-TLS 策略，该策略满足与入站消息通道所使用的需求相同的需求; 请参阅 第 400 页的『使用单个名为 CipherSpec 的队列管理器在来自 IBM MQ for Multiplatforms 队列管理器的入站通道上配置 AT-TLS』。

当客户机应用程序是 Java 应用程序，并且也在 z/OS 上运行，但尚未配置为使用 AT-TLS 时，可以使用相同的 AT-TLS 配置。

情况 6

在 IBM MQ for z/OS 队列管理器与在 IBM MQ for Multiplatforms 上运行的客户机应用程序之间，其中 IBM MQ for z/OS 队列管理器使用 AT-TLS，而客户机应用程序通过指定带有别名 CipherSpec 的 SSLCIPH 属性来使用 IBM MQ TLS。



此方案需要单个 AT-TLS 策略，该策略满足与入站消息通道所使用的需求相同的需求; 请参阅 第 404 页的『使用别名 CipherSpec 在来自 IBM MQ for Multiplatforms 队列管理器的入站通道上配置 AT-TLS』。

当客户机应用程序是 Java 应用程序，并且也在 z/OS 上运行，但尚未配置为使用 AT-TLS 时，可以使用相同的 AT-TLS 配置。

限制

IBM MQ for z/OS 不知道 AT-TLS，因此在上述场景中存在一些适用的限制:

- AT-TLS 与 IBM MQ TLS 组合不适用于集群发送方和集群接收方通道。
- IBM MQ for z/OS 队列管理器不知道它们正在使用 AT-TLS，并且不会从其伙伴队列管理器或客户机接收任何证书信息。 因此，以下属性对使用 AT-TLS 的通道的 z/OS 端没有任何影响:
  - SSLCAUTH 和 SSLPEER 通道属性
  - SSLRKEYC 队列管理器属性
  - CHLAUTH 规则的 SSLPEERMAP 属性
- 使用 TLS 密钥重新协商要求通道两侧都使用 IBM MQ TLS。 因此，如果使用 AT-TLS 连接到 IBM MQ for z/OS 队列管理器，那么 IBM MQ for Multiplatforms 队列管理器或客户机不应启用 TLS 密钥重新协商。

  要对队列管理器禁用 TLS 密钥重新协商，请将队列管理器 SSLRKEYC 参数设置为 0。 对于客户机，根据客户机类型，将相关参数设置为 0。 有关如何执行此操作的详细信息，请参阅 第 408 页的『重置 SSL 和 TLS 密钥』。

AT-TLS 配置语句

AT-TLS 使用一组语句进行配置。 本主题中记录的场景中使用的场景包括:

TTLSRule
指定用于将 TCP/IP 连接与 TLS 配置匹配的一组条件。 这又引用了其他语句类型。

TTLSGroupAction
指定是否启用引用 `TTLSRule` 。

TTLSEnvironmentAction
指定引用 `TTLSRule` 的详细配置，并引用许多其他语句。

TTLSKeyringParms
引用要由 AT-TLS 使用的密钥环。

TTLSCipherParms
定义要使用的密码套件。

TTLSEnvironmentAdvancedParms
定义已启用哪些 TLS 或 SSL 协议。

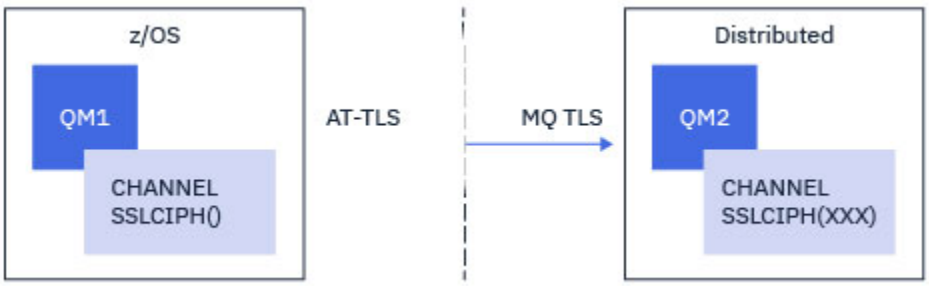
注意: 此处未记录使用 AT-TLS 的其他 AT-TLS 策略语句，可根据需要将其与 IBM MQ 配合使用。 但是，仅使用本主题中描述的策略对 IBM MQ 进行了测试。

z/OS 在出站通道上使用单个名为 *CipherSpec* 的 *IBM MQ for Multiplatforms* 队列管理器配置 *AT-TLS*

如何在从 IBM MQ for z/OS 队列管理器到 IBM MQ for Multiplatforms 队列管理器的出站通道上设置 AT-TLS。 在这种情况下， z/OS 队列管理器上的通道是未设置 SSLCIPH 属性的发送方通道，而非 z/OS 队列管理器上的通道是 SSLCIPH 属性设置为单个名为 CipherSpec 的接收方通道。

请参阅 第 396 页的『在出站通道上使用别名 CipherSpecs 配置到 IBM MQ for Multiplatforms 队列管理器的 AT-TLS』 以获取使用别名 CipherSpec 的示例。

在此示例中，将调整使用 TLS 1.3 TLS_AES_256_GCM_SHA384 CipherSpec 的现有发送方/接收方通道对，以便发送方通道使用 AT-TLS 而不是 IBM MQ TLS。



可以通过对配置进行轻微调整来使用其他 TLS 协议和 CipherSpecs 。 除了集群发送方和集群接收方通道之外，其他消息通道类型可以在不更改 AT-TLS 配置的情况下使用。

过程

步骤 1: 停止通道

步骤 2: 创建并应用 AT-TLS 策略

您需要为此方案创建以下 AT-TLS 语句:

1. TTLSRule 语句，用于将从通道启动程序地址空间到目标接收方通道的 IP 地址和端口号的出站连接进行匹配。 这些值应该与发送方通道的 CONNAME 中使用的信息匹配。 在此，已包含进一步的过滤以与特定通道启动程序作业名匹配。

```
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
}
```

   上述规则与从 CSQ1CHIN 作业到端口 1414 上的 IP 地址 123.456.78.9 的连接相匹配。

   TTLSRule 中描述了更高级的过滤选项。

2. 用于启用规则的 TTLSGroupAction 语句。 TTLSRule 使用 **TTLSGroupActionRef** 属性引用 TTLSGroupAction 。

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. 通过 **TTLSEnvironmentActionRef** 属性与 TTLSRule 关联的 TTLSEnvironmentAction 语句。TTLSEnvironmentAction 配置 TLS 环境并指定要使用的密钥环。

```
TTLSEnvironmentAction               CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                     CLIENT
  TTLSKeyringParmsRef               CSQ1-KEYRING
  TTLSCipherParmsRef                CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef   CSQ1-ENVIRONMENT-ADVANCED
}
```

4. TTLSKeyringParms 语句通过 **TTLSKeyringParmsRef** 属性与 TTLSEnvironmentAction 相关联，并定义 AT-TLS 所使用的密钥环。

   密钥环应包含远程非 z/OS 队列管理器所信任的证书。 可以使用与通道启动程序使用的密钥环相同的方式定义此密钥环; 请参阅 第 226 页的『Configuring your z/OS system to use TLS』。

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. 通过 **TTLSCipherParmsRef** 属性与 TTLSEnvironmentAction 关联的 TTLSCipherParms 语句。

   此语句必须包含单个密码套件名称，该名称必须与目标接收方通道上使用的 IBM MQ CipherSpec 名称等效。

   注: AT-TLS 密码套件名称不一定与 IBM MQ CipherSpec 名称匹配。 但是，通过在下表中查找 IBM MQ CipherSpec 名称，并在 TTLSCipherParms 语句主题中使用表 2 中的扩展字符列交叉引用十六进制代码列，可以找到与 IBM MQ CipherSpec 名称匹配的 AT-TLS 密码套件名称。

表 *83: z/OS* 上来自 *IBM MQ for z/OS 9.2.0* 的 *CipherSpecs*

CipherSpec	**协议**	**十六进制代码**	**在缺省情况下启用**
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Yes
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Yes
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Yes
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Yes
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Yes
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Yes
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Yes
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Yes
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Yes
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Yes

表 83: z/OS 上来自 IBM MQ for z/OS 9.2.0 的 CipherSpecs (继续)

CipherSpec	协议	十六进制代码	在缺省情况下启用
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Yes
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Yes
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Yes
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	否
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	否
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	否
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	否
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	否
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	否
TRIPLE_DES_SHA_US	SSL V3	000A	否
RC4_SHA_US	SSL V3	0005	否
RC4_MD5_US	SSL V3	0004	否
DES_SHA_EXPORT	SSL V3	0009	N
RC4_MD5_EXPORT	SSL V3	0003	否
RC2_MD5_EXPORT	SSL V3	0006	否
NULL_SHA	SSL V3	0002	否
NULL_MD5	SSL V3	0001	否

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_AES_256_GCM_SHA384
}
```

6. TTLSEnvironmentAdvancedParms 语句通过 **TTLSEnvironmentAdvancedParmsRef** 属性与 TTLSEnvironmentAction 相关联。

   此语句可用于指定启用哪些 SSL 和 TLS 协议。 使用 IBM MQ 时，应仅启用与 TTLSCipherParms 语句中使用的密码套件名称匹配的单个协议。

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

完整的语句集如下所示，应该应用于策略代理程序:

```
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
}
```

步骤 3: 从 z/OS 通道中除去 SSLCIPH

使用以下命令从 z/OS 通道中除去 CipherSpec :

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

步骤 4: 启动通道

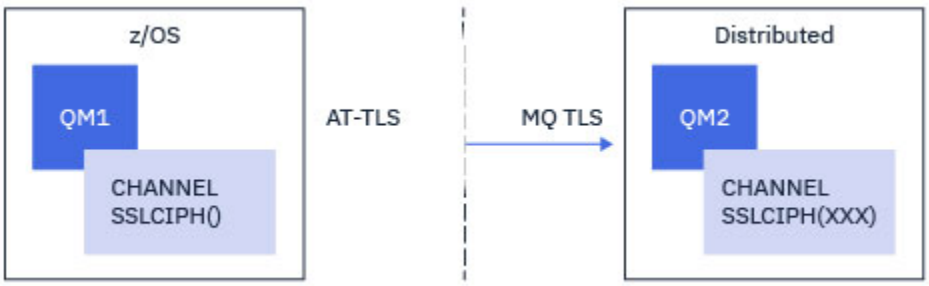
通道启动后，将使用 AT-TLS 和 IBM MQ TLS 的组合。

注意: 先前的 AT-TLS 语句仅是最低配置。 此处未记录使用 AT-TLS 的其他 AT-TLS 策略语句 ，可根据需要将其与 IBM MQ 配合使用。 但是，仅使用描述的策略对 IBM MQ 进行了测试。

z/OS 在出站通道上使用别名 *CipherSpecs* 配置到 *IBM MQ for Multiplatforms* 队列管理器的 *AT-TLS*

如何在从 IBM MQ for z/OS 队列管理器到 IBM MQ for Multiplatforms 队列管理器的出站通道上设置 AT-TLS。 在这种情况下， z/OS 队列管理器上的通道是未设置 SSLCIPH 属性的发送方通道，而非 z/OS 队列管理器上的通道是 SSLCIPH 属性设置为别名 CipherSpec 的接收方通道

在此示例中，将调整使用 ANY_TLS13 别名 CipherSpec 的现有发送方/接收方通道对，以便发送方通道使用 AT-TLS 而不是 IBM MQ TLS。



可以通过对配置进行轻微调整来使用其他 TLS 协议和 CiperSpecs 。 除了集群发送方和集群接收方通道之外，其他消息通道类型可以在不更改 AT-TLS 配置的情况下使用。

过程

步骤 1: 停止通道

步骤 2: 创建并应用 AT-TLS 策略

您需要为此方案创建以下 AT-TLS 语句:

1. TTLSRule 语句，用于将从通道启动程序地址空间到目标接收方通道的 IP 地址和端口号的出站连接进行匹配。 这些值应该与发送方通道的 CONNAME 中使用的信息匹配。 在此，已包含进一步的过滤以与特定通道启动程序作业名匹配。

```
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
}
```

上述规则与从 CSQ1CHIN 作业到端口 1414 上的 IP 地址 123.456.78.9 的连接相匹配。

TTLSRule 中描述了更高级的过滤选项。

2. 用于启用规则的 TTLSGroupAction 语句。 TTLSRule 使用 **TTLSGroupActionRef** 属性引用 TTLSGroupAction 。

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. 通过 **TTLSEnvironmentActionRef** 属性与 TTLSRule 关联的 TTLSEnvironmentAction 语句。 TTLSEnvironmentAction 配置 TLS 环境并指定要使用的密钥环。

```
TTLSEnvironmentAction               CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                     CLIENT
  TTLSKeyringParmsRef               CSQ1-KEYRING
  TTLSCipherParmsRef                CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef   CSQ1-ENVIRONMENT-ADVANCED
}
```

4. TTLSKeyringParms 语句通过 **TTLSKeyringParmsRef** 属性与 TTLSEnvironmentAction 相关联，并定义 AT-TLS 所使用的密钥环。

密钥环应包含远程非 z/OS 队列管理器所信任的证书。 可以使用与通道启动程序使用的密钥环相同的方式定义此密钥环; 请参阅 第 226 页的『Configuring your z/OS system to use TLS』。

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. 通过 **TTLSCipherParmsRef** 属性与 TTLSEnvironmentAction 关联的 TTLSCipherParms 语句。

   此语句必须包含一个或多个密码套件名称，其中至少一个应该与目标接收方通道上使用的别名 CipherSpec 所隐含的 CipherSpecs 集合兼容。

   注: AT-TLS 密码套件名称不一定与 IBM MQ CipherSpec 名称匹配。 但是，可以找到与 IBM MQ CipherSpec 名称匹配的 AT-TLS 密码套件名称，方法是在下表中找到 IBM MQ CipherSpec 名称，并将十六进制代码列与 TTLSCipherParms 主题中的表 2 中的扩展字符列进行交叉引用。

表 *84: z/OS* 上来自 *IBM MQ for z/OS 9.2.0* 的 *CipherSpecs*

CipherSpec	**协议**	**十六进制代码**	**在缺省情况下启用**
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Yes
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Yes
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Yes
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Yes
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Yes
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Yes
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Yes
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Yes
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Yes
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Yes
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Yes
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Yes
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Yes
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	否
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	否

表 84: z/OS 上来自 IBM MQ for z/OS 9.2.0 的 CipherSpecs (继续)			
CipherSpec	**协议**	**十六进制代码**	**在缺省情况下启用**
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	否
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	否
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	否
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	否
TRIPLE_DES_SHA_US	SSL V3	000A	否
RC4_SHA_US	SSL V3	0005	否
RC4_MD5_US	SSL V3	0004	否
DES_SHA_EXPORT	SSL V3	0009	N
RC4_MD5_EXPORT	SSL V3	0003	否
RC2_MD5_EXPORT	SSL V3	0006	否
NULL_SHA	SSL V3	0002	否
NULL_MD5	SSL V3	0001	否

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256
}
```

注意: 如果队列管理器和 AT-TLS 策略都支持 TLS 1.3，那么仅包含至少一个 TLS 1.3 CipherSpec 的别名 CipherSpecs 允许通道启动。 例如，使用 ANY_TLS12 会导致通道无法启动，即使 TTLSCipherParms 包含 TLS 1.2 CipherSpecs，但使用 ANY_TLS12_OR_HIGHER 或 ANY_TLS13 会允许通道启动。 请参阅 第 386 页的『别名 CipherSpec 设置之间的关系』 以获取说明。

6. TTLSEnvironmentAdvancedParms 语句通过 **TTLSEnvironmentAdvancedParmsRef** 属性与 TTLSEnvironmentAction 相关联。

   此语句可用于指定启用了哪些 SSL 和 TLS 协议，并且应该与 TTLSCipherParms 语句中的密码套件一致。

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3           OFF
  TLSv1           OFF
  TLSv1.1         OFF
  SecondaryMap    OFF
  TLSv1.2         OFF
  TLSv1.3         ON
}
```

完整的语句集如下所示，应该应用于策略代理程序:

```
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
}
```

步骤 3: 从 z/OS 通道中除去 SSLCIPH

使用以下命令从 z/OS 通道中除去 CipherSpec :

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

步骤 4: 启动通道

通道启动后，将使用 AT-TLS 和 IBM MQ TLS 的组合。

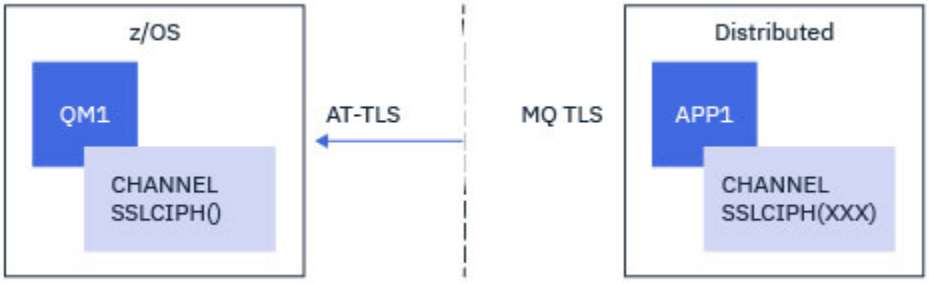
注意: 先前的 AT-TLS 语句仅是最低配置。 此处未记录使用 AT-TLS 的其他 AT-TLS 策略语句，可根据需要将其与 IBM MQ 配合使用。 但是，仅使用描述的策略对 IBM MQ 进行了测试。

z/OS 使用单个名为 *CipherSpec* 的队列管理器在来自 *IBM MQ for Multiplatforms* 队列管理器的入站通道上配置 *AT-TLS*

如何在从 IBM MQ for Multiplatforms 队列管理器到 IBM MQ for z/OS 队列管理器的入站通道上设置 AT-TLS。 在这种情况下， z/OS 队列管理器上的通道是未设置 SSLCIPH 属性的接收方通道，而非 z/OS 队列管理器上的通道是将 SSLCIPH 属性设置为单个名为 CipherSpec 的发送方通道。

请参阅 第 404 页的『使用别名 CipherSpec 在来自 IBM MQ for Multiplatforms 队列管理器的入站通道上配置 AT-TLS』 以获取使用别名 CipherSpec 的示例。

在此示例中，将调整使用 TLS 1.3 TLS_AES_256_GCM_SHA384 CipherSpec 的现有发送方/接收方通道对，以便接收方通道使用 AT-TLS 而不是 IBM MQ TLS。



可以通过对配置进行轻微调整来使用其他 TLS 协议和 CipherSpecs 。 除了集群发送方和集群接收方通道之外，其他消息通道类型可以在不更改 AT-TLS 配置的情况下使用。

过程

步骤 1: 停止通道

步骤 2: 创建并应用 AT-TLS 策略

您需要为此方案创建以下 AT-TLS 语句:

1. TTLSRule 语句，用于将入站连接与来自发送方通道的 IP 地址的通道启动程序地址空间相匹配。 在此，已包含进一步的过滤以与特定通道启动程序作业名匹配。

```
TTLSRule                        REMOTE-TO-CSQ1
{
  LocalAddr                     ALL
  LocalPortRange                1414
  RemoteAddr                    123.456.78.9
  Jobname                       CSQ1CHIN
  Direction                     INBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-INBOUND-ENVIRONMENT-ACTION
}
```

   上述规则与从远程 IP 地址 123.456.78.9 进入本地端口 1414 上的 CSQ1CHIN 作业的连接相匹配。

   TTLSRule 中描述了更高级的过滤选项。

2. 用于启用规则的 TTLSGroupAction 语句。 TTLSRule 使用 **TTLSGroupActionRef** 属性引用 TTLSGroupAction 。

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. TTLSEnvironmentAction 语句通过 **TTLSEnvironmentActionRef** 属性与 TTLSRule 相关联。 TTLSEnvironmentAction 配置 TLS 环境并指定要使用的密钥环。

```
TTLSEnvironmentAction                 CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                       SERVER
  TTLSKeyringParmsRef                 CSQ1-KEYRING
  TTLSCipherParmsRef                  CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef     CSQ1-ENVIRONMENT-ADVANCED
}
```

AT-TLS 提供了提供相互认证的功能，这相当于使用 SSLCAUTH 通道属性。 通过将 **HandshakeRole** 值为 *ServerWithClientAuth* 的 TTLSEnvironmentAction 语句用于入站 TTLSEnvironmentAction 语句来完成此操作。

4. TTLSKeyringParms 语句通过 **TTLSKeyringParmsRef** 属性与 TTLSEnvironmentAction 相关联，并定义 AT-TLS 所使用的密钥环。

   密钥环应包含远程非 z/OS 队列管理器所信任的证书。 可以使用与通道启动程序使用的密钥环相同的方式定义此密钥环; 请参阅 第 226 页的『Configuring your z/OS system to use TLS』。

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. 通过 **TTLSCipherParmsRef** 属性与 TTLSEnvironmentAction 关联的 TTLSCipherParms 语句。

   此语句必须包含单个密码套件名称，该名称必须与远程发送方通道上使用的 IBM MQ CipherSpec 名称等效。

   注: AT-TLS 密码套件名称不一定与 IBM MQ CipherSpec 名称匹配。 但是，通过在下表中查找 IBM MQ CipherSpec 名称，并在 TTLSCipherParms 语句主题中使用表 2 中的扩展字符列交叉引用十六进制代码列，可以找到与 IBM MQ CipherSpec 名称匹配的 AT-TLS 密码套件名称。

表 *85: z/OS* 上来自 *IBM MQ for z/OS 9.2.0* 的 *CipherSpecs*

CipherSpec	**协议**	**十六进制代码**	**在缺省情况下启用**
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Yes
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Yes
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Yes
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Yes
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Yes
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Yes
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Yes
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Yes
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Yes
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Yes
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Yes
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Yes
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Yes

表 85: z/OS 上来自 IBM MQ for z/OS 9.2.0 的 CipherSpecs (继续)			
CipherSpec	**协议**	**十六进制代码**	**在缺省情况下启用**
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	否
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	否
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	否
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	否
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	否
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	否
TRIPLE_DES_SHA_US	SSL V3	000A	否
RC4_SHA_US	SSL V3	0005	否
RC4_MD5_US	SSL V3	0004	否
DES_SHA_EXPORT	SSL V3	0009	N
RC4_MD5_EXPORT	SSL V3	0003	否
RC2_MD5_EXPORT	SSL V3	0006	否
NULL_SHA	SSL V3	0002	否
NULL_MD5	SSL V3	0001	否

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_AES_256_GCM_SHA384
}
```

6. TTLSEnvironmentAdvancedParms 语句通过 **TTLSEnvironmentAdvancedParmsRef** 属性与 TTLSEnvironmentAction 相关联。

   此语句可用于指定启用哪些 SSL 和 TLS 协议。 使用 IBM MQ 时，应仅启用与 TTLSCipherParms 语句中使用的密码套件名称匹配的单个协议。

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

完整的语句集如下所示，应该应用于策略代理程序:

```
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
}
```

步骤 3: 从 z/OS 通道中除去 SSLCIPH

使用以下命令从 z/OS 通道中除去 CipherSpec :

```
ALTER CHANNEL(channel-name) CHLTYPE(RCVR) SSLCIPH(' ')
```

步骤 4: 启动通道

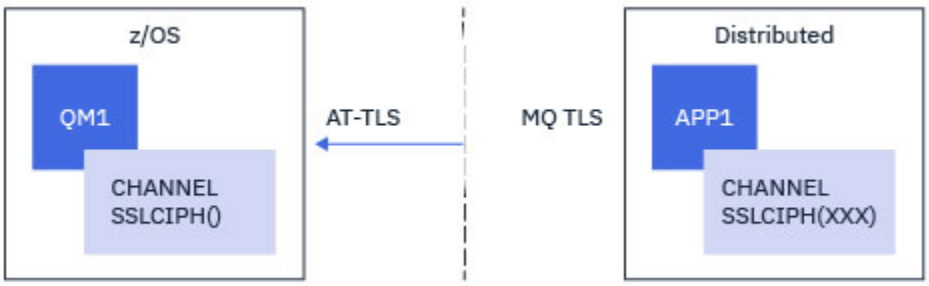
通道启动后，将使用 AT-TLS 和 IBM MQ TLS 的组合。

注意: 先前的 AT-TLS 语句仅是最低配置。 此处未记录使用 AT-TLS 的其他 AT-TLS 策略语句，可根据需要将其与 IBM MQ 配合使用。 但是，仅使用描述的策略对 IBM MQ 进行了测试。

z/OS 使用别名 *CipherSpec* 在来自 *IBM MQ for Multiplatforms* 队列管理器的入站通道上配置 *AT-TLS*

如何在从 IBM MQ for Multiplatforms 队列管理器到 IBM MQ for z/OS 队列管理器的入站通道上设置 AT-TLS。 在这种情况下， z/OS 队列管理器上的通道是未设置 SSLCIPH 属性的接收方通道，而非 z/OS 队列管理器上的通道是 SSLCIPH 属性设置为别名 CipherSpec 的发送方通道。

在此示例中，将调整使用任何 TLS 1.3 CipherSpec 的现有发送方/接收方通道对，以便接收方通道使用 AT-TLS 而不是 IBM MQ TLS。



可以通过对配置进行轻微调整来使用其他 TLS 协议和 CiperSpecs 。 除了集群发送方和集群接收方通道之外，其他消息通道类型可以在不更改 AT-TLS 配置的情况下使用。

过程

步骤 1: 停止通道

步骤 2: 创建并应用 AT-TLS 策略

您需要为此方案创建以下 AT-TLS 语句:

1. TTLSRule 语句，用于将入站连接与来自发送方通道的 IP 地址的通道启动程序地址空间相匹配。 在此，已包含进一步的过滤以与特定通道启动程序作业名匹配。

```
TTLSRule                        REMOTE-TO-CSQ1
{
  LocalAddr                     ALL
  LocalPortRange                1414
  RemoteAddr                    123.456.78.9
  Jobname                       CSQ1CHIN
  Direction                     INBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-INBOUND-ENVIRONMENT-ACTION
}
```

上述规则与从远程 IP 地址 123.456.78.9 进入本地端口 1414 上的 CSQ1CHIN 作业的连接相匹配。

TTLSRule 中描述了更高级的过滤选项。

2. 用于启用规则的 TTLSGroupAction 语句。 TTLSRule 使用 **TTLSGroupActionRef** 属性引用 TTLSGroupAction 。

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. TTLSEnvironmentAction 语句通过 **TTLSEnvironmentActionRef** 属性与 TTLSRule 相关联。 TTLSEnvironmentAction 配置 TLS 环境并指定要使用的密钥环。

```
TTLSEnvironmentAction                 CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                       SERVER
  TTLSKeyringParmsRef                 CSQ1-KEYRING
  TTLSCipherParmsRef                  CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef     CSQ1-ENVIRONMENT-ADVANCED
}
```

AT-TLS 提供了提供相互认证的功能，这相当于使用 SSLCAUTH 通道属性。 通过将 **HandshakeRole** 值为 *ServerWithClientAuth* 的 TTLSEnvironmentAction 语句用于入站 TTLSEnvironmentAction 语句来完成此操作。

4. TTLSKeyringParms 语句通过 **TTLSKeyringParmsRef** 属性与 TTLSEnvironmentAction 相关联，并定义 AT-TLS 所使用的密钥环。

   密钥环应包含远程非 z/OS 队列管理器所信任的证书。 可以使用与通道启动程序使用的密钥环相同的方式定义此密钥环; 请参阅 第 226 页的『Configuring your z/OS system to use TLS』。

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. 通过 **TTLSCipherParmsRef** 属性与 TTLSEnvironmentAction 关联的 TTLSCipherParms 语句。

   此语句必须至少包含一个包含在远程发送方通道上设置的别名 CipherSpec 中的密码套件名称。

   注: AT-TLS 密码套件名称不一定与 IBM MQ CipherSpec 名称匹配。 但是，通过在下表中查找 IBM MQ CipherSpec 名称，并在 TTLSCipherParms 语句主题中使用表 2 中的扩展字符列交叉引用十六进制代码列，可以找到与 IBM MQ CipherSpec 名称匹配的 AT-TLS 密码套件名称。

表 *86: z/OS* 上来自 *IBM MQ for z/OS 9.2.0* 的 *CipherSpecs*

CipherSpec	**协议**	**十六进制代码**	**在缺省情况下启用**
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Yes
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Yes
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Yes
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Yes
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Yes
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Yes
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Yes
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Yes
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Yes
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Yes
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Yes
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Yes
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Yes
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	否

表 86: z/OS 上来自 IBM MQ for z/OS 9.2.0 的 CipherSpecs (继续)			
CipherSpec	**协议**	**十六进制代码**	**在缺省情况下启用**
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	否
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	否
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	否
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	否
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	否
TRIPLE_DES_SHA_US	SSL V3	000A	否
RC4_SHA_US	SSL V3	0005	否
RC4_MD5_US	SSL V3	0004	否
DES_SHA_EXPORT	SSL V3	0009	N
RC4_MD5_EXPORT	SSL V3	0003	否
RC2_MD5_EXPORT	SSL V3	0006	否
NULL_SHA	SSL V3	0002	否
NULL_MD5	SSL V3	0001	否

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256
}
```

注意: 如果队列管理器和 AT-TLS 策略都支持 TLS 1.3，那么仅包含至少一个 TLS 1.3 CipherSpec 的别名 CipherSpecs 允许通道启动。 例如，使用 ANY_TLS12 会导致通道无法启动，即使 TTLSCipherParms 包含 TLS 1.2 CipherSpecs，但使用 ANY_TLS12_OR_HIGHER 或 ANY_TLS13 会允许通道启动。 请参阅 第 386 页的『别名 CipherSpec 设置之间的关系』 以获取说明。

6. TTLSEnvironmentAdvancedParms 语句通过 **TTLSEnvironmentAdvancedParmsRef** 属性与 TTLSEnvironmentAction 相关联。

   此语句可用于指定启用了哪些 SSL 和 TLS 协议，并且应该与 TTLSCipherParms 语句中的密码套件一致。

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3           OFF
  TLSv1           OFF
  TLSv1.1         OFF
  SecondaryMap    OFF
  TLSv1.2         OFF
  TLSv1.3         ON
}
```

完整的语句集如下所示，应该应用于策略代理程序:

```
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
}
```

步骤 3: 从 z/OS 通道中除去 SSLCIPH

使用以下命令从 z/OS 通道中除去 CipherSpec :

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

步骤 4: 启动通道

通道启动后，将使用 AT-TLS 和 IBM MQ TLS 的组合。

注意: 先前的 AT-TLS 语句仅是最低配置。 此处未记录使用 AT-TLS 的其他 AT-TLS 策略语句，可根据需要将其与 IBM MQ 配合使用。 但是，仅使用描述的策略对 IBM MQ 进行了测试。

重置 SSL 和 TLS 密钥

IBM MQ 支持在队列管理器和客户机上重置密钥。

当指定数量的加密数据字节流经通道时，将重置密钥。 如果启用了通道脉动信号，那么将在发送数据或在通道脉动信号之后接收数据之前重置密钥。

密钥重置值始终由 IBM MQ 通道的启动端设置。

队列管理器

对于队列管理器，请使用带有参数 **SSLRKEYC** 的命令 **ALTER QMGR** 来设置密钥重新协商期间使用的值。

在 IBM i 上，将 **CHGMQM** 与 **SSLRSTCNT** 参数配合使用。

MQI 客户机

缺省情况下， MQI 客户机不会重新协商密钥。 您可以通过三种方式中的任何一种来使 MQI 客户机重新协商密钥。 在以下列表中，将按优先级顺序显示方法。 如果指定多个值，那么将使用最高优先级值。

1. 通过在 MQCONNX 调用上的 MQSCO 结构中使用 KeyResetCount 字段。
2. 通过使用环境变量 **MQSSLRESET**。
3. 通过在 客户机配置文件的 SSL 节中设置 **SSLKeyResetCount** 属性。

这些变量可以设置为 0 到 999 999 999 范围内的整数，表示在重新协商 TLS 密钥之前在 TLS 对话中发送和接收的未加密字节数。 指定值 0 指示从不重新协商 TLS 密钥。 如果指定 1 字节到 32 KB 范围内的 TLS 密钥重置计数，那么 TLS 通道将使用 32 KB 的密钥重置计数。 这是为了避免对小型 TLS 密钥重置值进行过多的密钥重置。

如果指定了大于零的值，并且为通道启用了通道脉动信号，那么在发送消息数据或在通道脉动信号之后接收消息数据之前，还会重新协商密钥。

每次成功重新协商后重置下一个密钥重新协商之前的字节计数。

Java

对于 IBM MQ classes for Java，应用程序可以通过以下任一方式重置密钥:

- 通过在 MQEnvironment 类中设置 sslResetCount 字段。
- 通过在 Hashtable 对象中设置环境属性 MQC.SSL_RESET_COUNT_PROPERTY。 应用程序之后会向 MQEnvironment 类中的 `properties` 字段分配散列表，或者将散列表传递到其构造函数上的 MQQueueManager 对象。

如果应用程序使用这些方法中的多种方法，那么将应用通常的优先顺序规则。 请参阅 类 com.ibm.mq.MQEnvironment 以获取优先顺序规则。

sslResetCount 字段的值或环境属性 MQC.SSL_RESET_COUNT_PROPERTY，表示 IBM MQ classes for Java 客户机代码在重新协商密钥前发送和接收的字节总数。 发送的字节数是加密之前的字节数，接收的字节数是解密之后的字节数。 字节数也包括 IBM MQ classes for Java 客户机发送和接收的控制信息。

如果重置计数是零（即缺省值），那么始终不会重新协商密钥。 如果没有指定 CipherSuite，将忽略重置计数。

JMS

对于 IBM MQ classes for JMS， SSLRESETCOUNT 属性表示在重新协商用于加密的密钥之前，连接发送和接收的总字节数。 发送的字节数是加密之前的字节数，接收的字节数是解密之后的字节数。 字节数还包含 IBM MQ classes for JMS 发送和接收的控制信息。 例如，要配置 ConnectionFactory 对象，该对象可用于通过启用 TLS 的 MQI 通道创建连接，该通道具有在 4 MB 数据流后重新协商的密钥，请向 JMSAdmin 发出以下命令:

```
ALTER CF(my.cf) SSLRESETCOUNT(4194304)
```

如果 SSLRESETCOUNT 的值为零 (缺省值)，那么永远不会重新协商密钥。 如果未设置 SSLCIPHERSUITE，那么将忽略 SSLRESETCOUNT 属性。

.NET

对于 .NET 非受管客户机，整数属性 **SSLKeyResetCount** 指示在重新协商密钥之前在 TLS 对话中发送和接收的未加密字节数。 有关在 IBM MQ classes for .NET 中使用对象属性的更多信息，请参阅 获取和设置属性值。

对于 .NET 受管客户机， SSLStream 类不支持密钥重置/重新协商。 但是，为了与其他 IBM MQ 客户机保持一致， IBM MQ 受管 .NET 客户机允许应用程序设置 **SSLKeyResetCount**。 有关更多信息，请参阅 密钥重置或重新协商。

XMS .NET

对于 XMS .NET 非受管客户机，请参阅 与 IBM MQ 队列管理器的安全连接。

相关参考

ALTER QMGR
显示队列管理器
更改消息队列管理器 (CHGMQM)
显示消息队列管理器 (DSPMQM)

在用户出口程序中实现机密性

在安全出口中实现机密性

安全出口可以通过生成和分发对称密钥来对通道上流动的数据进行加密和解密，从而在机密性服务中发挥作用。 用于执行此操作的常见技术使用 PKI 技术。

一个安全出口生成一个随机数据值，使用合作伙伴安全出口所表示的队列管理器或用户的公用密钥对其进行加密，并将加密后的数据以安全消息形式发送给其合作伙伴。 伙伴安全出口使用其表示的队列管理器或用户的专用密钥对随机数据值进行解密。 现在，每个安全出口都可以使用随机数据值，通过使用两者已知的算法，独立于另一个安全出口来派生对称密钥。 或者，他们可以使用随机数据值作为键。

如果第一个安全出口此时尚未认证其合作伙伴，那么合作伙伴发送的下一条安全消息可能包含使用对称密钥加密的期望值。 第一个安全出口现在可以通过检查合作伙伴安全出口是否能够正确加密期望值来认证其合作伙伴。

如果有多个算法可供使用，那么安全出口还可以利用此机会来商定用于对通道上流动的数据进行加密和解密的算法。

在消息出口中实现机密性

通道发送端的消息出口可以对消息中的应用程序数据进行加密，通道接收端的另一个消息出口可以对数据进行解密。 出于性能原因，通常使用对称密钥算法来实现此目的。 有关如何生成和分发对称密钥的更多信息，请参阅 第 410 页的『在用户出口程序中实现机密性』。

消息中的头 (例如，包含嵌入式消息描述符的传输队列头 MQXQH) 不得由消息出口加密。 这是因为消息头的数据转换是在发送端调用消息出口之后或在接收端调用消息出口之前进行的。 如果对头进行了加密，那么数据转换将失败，并且通道将停止。

在发送和接收出口中实现机密性

发送和接收出口可用于对通道上流动的数据进行加密和解密。 它们比消息出口更适合提供此服务，原因如下:

- 在消息通道上，可以对消息头以及消息中的应用程序数据进行加密。
- 可以在 MQI 通道以及消息通道上使用发送和接收出口。 MQI 调用上的参数可能包含在 MQI 通道上流动时需要保护的敏感应用程序数据。 因此，您可以在两种通道上使用相同的发送和接收出口。

在 API 出口和 API 交叉出口中实现机密性

当消息由发送应用程序放入时，消息中的应用程序数据可由 API 或 API 交叉出口加密，当接收应用程序检索消息时，消息中的应用程序数据可由第二个出口解密。 出于性能原因，对称密钥算法通常用于此目的。 但是，在应用程序级别，许多用户可能正在相互发送消息，问题是如何确保只有消息的预期接收方才能够解密消息。 一种解决方案是对相互发送消息的每对用户使用不同的对称密钥。 但是，此解决方案可能难以管理，并且需要花费大量时间，尤其是在用户属于不同组织的情况下。 解决此问题的标准方法称为 数字封包 ，并使用 PKI 技术。

当应用程序将消息放入队列时， API 或 API 交叉出口会生成随机对称密钥，并使用该密钥对消息中的应用程序数据进行加密。 出口使用预期接收方的公用密钥对对称密钥进行加密。 然后，它将消息中的应用程序数据替换为加密的应用程序数据和加密的对称密钥。 这样，只有预期的接收方才能解密对称密钥，从而解密应用程序数据。 如果加密消息具有多个可能的预期接收方，那么出口可以对每个预期接收方的对称密钥副本进行加密。

如果可用于对应用程序数据进行加密和解密的不同算法，那么出口可以包含其已使用的算法的名称。

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 411
- Archive log data sets; see note “2” on page 411
- Page sets; see note “1” on page 411
- BSDS; see note “2” on page 411
- CSQINP* data sets; see note “2” on page 411
- SMDS; see note “1” on page 411

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 412 to “6” on page 412 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 412
2. “Configuring data set encryption for the log data sets” on page 413

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 413.

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 413

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 412

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 414 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 412
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 412.
   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 416.

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 417 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 416.

   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 416 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 416 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).

消息的数据完整性

为了保持数据完整性，您可以使用各种类型的用户出口程序为消息提供消息摘要或数字签名。

数据完整性

在消息中实现数据完整性

使用 TLS 时，您选择的 CipherSpec 将确定企业中的数据完整性级别。 如果使用 IBM MQ Advanced Message Service (AMS)，那么可以指定唯一消息的完整性。

在消息出口中实现数据完整性

消息可以由消息出口在通道发送端进行数字签名。 然后，可以通过消息出口在通道的接收端检查数字签名，以检测消息是否已被故意修改。

可以通过使用消息摘要而不是数字签名来提供某些保护。 消息摘要可能对随意或不分青红皂白的篡改有效，但它不会阻止更知情的个人更改或替换消息，并为其生成全新的摘要。 如果用于生成消息摘要的算法是众所周知的算法，那么情况尤其如此。

在发送和接收出口中实现数据完整性

在消息通道上，消息出口更适合提供此服务，因为消息出口有权访问整个消息。 在 MQI 通道上， MQI 调用上的参数可能包含需要保护的应用程序数据，只有发送和接收出口才能提供此保护。

在 API 出口或 API 交叉出口中实现数据完整性

当消息由发送应用程序放入时，可以通过 API 或 API 交叉出口对消息进行数字签名。 然后，当接收应用程序检索消息以检测消息是否已被有意修改时，可通过第二出口来检查数字签名。

可以通过使用消息摘要而不是数字签名来提供某些保护。 消息摘要可能对随意或不分青红皂白的篡改有效，但它不会阻止更知情的个人更改或替换消息，并为其生成全新的摘要。 如果用于生成消息摘要的算法是众所周知的算法，那么尤其如此。

更多信息

请参阅 第 368 页的『启用 CipherSpecs』 部分，以获取有关确保数据完整性的更多信息。

相关任务

使用 TLS 连接两个队列管理器

安全地将客户机连接到队列管理器

审计

您可以使用事件消息来检查安全性入侵或尝试的入侵。 您还可以使用 IBM MQ Explorer 来检查系统的安全性。

要检测是否尝试执行未经授权的操作 (例如，连接到队列管理器或将消息放入队列)，请检查队列管理器生成的事件消息，特别是权限事件消息。 有关队列管理器事件消息的更多信息，请参阅 队列管理器事件，有关一般事件监视的更多信息，请参阅 事件监视。

确保集群安全

授权或阻止队列管理器加入集群或将消息放入集群队列。 强制队列管理器离开集群。 为集群配置 TLS 时，请考虑一些其他注意事项。

停止未经授权的队列管理器发送消息

防止未经授权的队列管理器使用通道安全出口将消息发送到队列管理器。

开始之前

集群对安全出口的工作方式没有影响。 您可以像在分布式排队环境中一样限制对队列管理器的访问。

关于此任务

阻止所选队列管理器向队列管理器发送消息:

过程

1. 在 CLUSRCVR 通道定义上定义通道安全出口程序。
2. 编写一个程序，用于对尝试在集群接收方通道上发送消息的队列管理器进行认证，并在这些队列管理器未经授权时拒绝其访问。

下一步做什么

在 MCA 启动和终止时调用通道安全出口程序。

停止将消息放入队列的未经授权的队列管理器

使用集群接收方通道上的通道放置权限属性来停止未经授权的队列管理器将消息放置到队列中。 通过使用 z/OS 上的 RACF ， 或 OAM on Multiplatforms 检查消息中的用户标识来授权远程队列管理器。

关于此任务

使用平台的安全设施和 IBM MQ 中的访问控制机制来控制对队列的访问。

过程

1. 要防止某些队列管理器将消息放入队列中，请使用平台上可用的安全设施。

   例如:

   - z/OS RACF 或 IBM MQ for z/OS 上的其他外部安全性管理器
   - Multi 其他 Multiplatforms 版上的对象权限管理器 (OAM)。
2. 使用 CLUSRCVR 通道定义上的 put 权限 PUTAUT 属性。

   PUTAUT 属性允许您指定要用于建立将消息放入队列的权限的用户标识。

   PUTAUT 属性上的选项包括:

DEF
使用缺省用户标识。

z/OS 在 z/OS 上，检查可能涉及使用从网络接收到的用户标识以及从 MCAUSER 派生的用户标识。

CTX
在与消息关联的上下文信息中使用用户标识。

z/OS 在 z/OS 上，检查可能涉及使用从网络接收的用户标识和/或从 MCAUSER 派生的用户标识。 如果链接可信且已认证，请使用此选项。

z/OS **ONLYMCA (仅限 z/OS)**
对于 DEF，但不使用从网络接收的任何用户标识。 如果链接不可信，请使用此选项。 您希望仅允许对其执行针对 MCAUSER 定义的一组特定操作。

z/OS **ALTMCA (仅限 z/OS)**
对于 CTX，但不使用从网络接收的任何用户标识。

授权将消息放入远程集群队列

在 z/OS 上，使用 RACF 设置用于放入集群队列的授权。 在多平台上，授权访问以连接到队列管理器，并将其放入这些队列管理器上的队列。

关于此任务

缺省行为是对 SYSTEM.CLUSTER.TRANSMIT.QUEUE 执行访问控制。 请注意，即使您正在使用多个传输队列，此行为也适用。

仅当您在 qm.ini 文件中将 **ClusterQueueAccessControl** 属性配置为 *RQMName*(如 安全性节 主题中所述) 并重新启动队列管理器时，本主题中描述的特定行为才适用。

过程

- z/OS
  对于 z/OS，请发出以下命令:

```
RDEFINE MQQUEUE QMgrName.QUEUE. QueueName UACC(NONE)
PERMIT QMgrName.QUEUE. QueueName CLASS(MQADMIN) ID(GroupName) ACCESS(UPDATE)
```

- ALW
  对于 AIX, Linux, and Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -t qmgr -g GroupName +connect
setmqaut -m QMgrName -t queue -n QueueName -g GroupName -all +put
```

- IBM i
  对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ(' QMgrName ') OBJTYPE(*MQM) USER(GroupName) AUT(*CONNECT)
GRTMQMAUT OBJ(' QueueName ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
```

用户只能将消息放入指定的集群队列，而不能放入其他集群队列。

变量名称具有以下含义:

QMgrName
队列管理器的名称。 在 z/OS 上，此值也可以是队列共享组的名称。

GroupName
要授予访问权的组的名称。

QueueName
要更改其权限的队列或通用概要文件的名称。

下一步做什么

如果在将消息放入集群队列时指定应答队列，那么使用应用程序必须具有发送应答的权限。 通过遵循 第 348 页的『授予将消息放入远程集群队列的权限』中的指示信息来设置此权限。

相关概念
qm.ini 中的安全性节

阻止队列管理器加入集群

如果流氓队列管理器加入集群，那么很难阻止它接收您不希望它接收的消息。

过程

如果要确保只有某些授权队列管理器加入集群，您可以选择三种方法:

- 通过使用通道认证记录，您可以根据远程 IP 地址，远程队列管理器名称或远程系统提供的 TLS 专有名称来阻止集群通道连接。
- 编写出口程序以防止未经授权的队列管理器写入 SYSTEM.CLUSTER.COMMAND.QUEUE。 请勿限制对 SYSTEM.CLUSTER.COMMAND.QUEUE 的访问，以使任何队列管理器都无法对其进行写操作，否则将阻止任何队列管理器加入集群。
- CLUSRCVR 通道定义上的安全出口程序。

集群通道上的安全出口

在集群通道上使用安全出口时的额外注意事项。

关于此任务

首次启动集群发送方通道时，它将使用系统管理员手动定义的属性。 当通道停止并重新启动时，它将从相应的集群接收方通道定义中选取属性。 将使用新属性 (包括 SecurityExit 属性) 覆盖原始集群发送方通道定义。

过程

1. 必须在通道的集群发送方端和集群接收方端定义安全出口。

   初始连接必须使用安全出口握手进行，即使安全出口名称是从集群接收方定义发送过来的。
2. 验证安全出口中 MQCXP 结构中的 PartnerName 。

   仅当伙伴队列管理器已获得授权时，出口才能允许通道启动
3. 将集群接收方定义上的安全出口设计为接收方启动。
4. 如果将其设计为发送方启动，那么没有安全出口的未经授权的队列管理器可以加入集群，因为不会执行安全检查。

   直到通道停止并重新启动后，才能从集群接收方定义中发送 SCYEXIT 名称并进行完全安全性检查。
5. 要查看当前正在使用的集群发送方通道定义，请使用以下命令:

   ```
   DISPLAY CLUSQMGR( queue manager ) ALL
   ```

   此命令显示从集群接收方定义发送的属性。
6. 要查看原始定义，请使用以下命令:

   ```
   DISPLAY CHANNEL( channel name ) ALL
   ```

7. 如果队列管理器位于不同的平台上，那么您可能需要在集群发送方队列管理器上定义通道自动定义出口 CHADEXIT。

   使用通道自动定义出口将 SecurityExit 属性设置为适合于目标平台的格式。
8. 部署并配置安全性出口。

   z/OS

   安全出口装入模块必须位于通道启动程序地址空间过程的 CSQXLIB DD 语句中指定的数据集中。

   AIX, Linux, and Windows 系统

   - 安全出口动态链接库必须位于通道定义的 SCYEXIT 属性中指定的路径中。
   - 通道自动定义出口动态链接库必须位于队列管理器定义的 CHADEXIT 属性中指定的路径中。

强制不需要的队列管理器离开集群

通过在完整存储库队列管理器上发出 RESET CLUSTER 命令，强制不需要的队列管理器离开集群。

关于此任务

您可以强制不需要的队列管理器离开集群。 例如，如果删除了队列管理器，但仍向集群定义了其集群接收方通道。 你可能想整理一下

只有完整存储库队列管理器才有权从集群中弹出队列管理器。

注: 虽然使用 RESET CLUSTER 命令会强制从集群中除去队列管理器，但使用 RESET CLUSTER 本身并不会阻止队列管理器稍后重新加入集群。 要确保队列管理器不重新加入集群，请遵循 第 421 页的『阻止队列管理器加入集群』中详细描述的步骤。

遵循以下过程从集群 NORWAY 中弹出队列管理器 OSLO :

过程

1. 在完整存储库队列管理器上，发出以下命令:

   ```
   RESET CLUSTER(NORWAY) QMNAME(OSLO) ACTION(FORCEREMOVE)
   ```
2. 在命令中替代使用 QMID 而不是 QMNAME :

   ```
   RESET CLUSTER(NORWAY) QMID(qmid) ACTION(FORCEREMOVE)
   ```

   注: QMID 是一个字符串，因此 qmid 的值应该用单引号括起来，例如 QMID('FR01_2019-07-15_14.42.42')。

结果

强制除去的队列管理器不会更改; 其本地集群定义显示该队列管理器在集群中。 所有其他队列管理器中的定义都不会显示在集群中。

阻止队列管理器接收消息

您可以阻止集群队列管理器接收未经授权通过使用出口程序接收的消息。

关于此任务

很难阻止作为集群成员的队列管理器定义队列。 存在流氓队列管理器加入集群的危险，并定义其自己的集群中某个队列的实例。 现在，它可以接收它无权接收的消息。 要阻止队列管理器接收消息，请使用过程中提供的下列其中一个选项。

过程

- 每个集群发送方通道上的通道出口程序。 出口程序使用连接名称来确定要发送消息的目标队列管理器是否合适。
- 集群工作负载出口程序，它使用目标记录来确定要发送消息的目标队列和队列管理器的适用性。

SSL/TLS 和集群

为集群配置 TLS 时，请注意 CLUSRCVR 通道定义将作为自动定义的 CLUSSDR 通道传播到其他队列管理器。 如果 CLUSRCVR 通道使用 TLS ，那么必须在使用该通道进行通信的所有队列管理器上配置 TLS。

有关 TLS 的更多信息，请参阅第 21 页的『IBM MQ 中的 TLS 安全协议』。 这里的建议通常适用于集群通道，但您可能希望对以下内容给予一些特殊考虑:

在 IBM MQ 集群中，特定 CLUSRCVR 通道定义经常传播到许多其他队列管理器，在这些队列管理器中，它将变换为自动定义的 CLUSSDR。 随后，自动定义的 CLUSSDR 用于启动到 CLUSRCVR 的通道。 如果为 TLS 连接配置了 CLUSRCVR ，那么以下注意事项适用:

- 要与此 CLUSRCVR 通信的所有队列管理器都必须有权访问 TLS 支持。 此 TLS 供应必须支持通道的 CipherSpec 。
- 自动定义的集群发送方通道所传播到的不同队列管理器将具有不同的关联专有名称。 如果要在 CLUSRCVR 上使用专有名称对等检查，那么必须进行设置，以便成功匹配可接收的所有专有名称。

  例如，假设将托管将连接到特定 CLUSRCVR 的集群发送方通道的所有队列管理器都具有关联的证书。 我们还假设所有这些证书中的专有名称都将国家或地区定义为英国，将组织定义为 IBM，将组织单元定义为 IBM MQ 开发，并且所有这些证书都具有格式为 DEVT.QMnnn 的公共名称，其中 nnn 是数字。

  在这种情况下， CLUSRCVR 上的 SSLPEER 值 C=UK, O=IBM, OU=IBM MQ Development, CN=DEVT.QM* 将允许所有必需的集群发送方通道成功连接，但将阻止不需要的集群发送方通道连接。
- 如果使用定制 CipherSpec 字符串，请注意在所有平台上都不允许使用定制字符串格式。 例如，CipherSpec string RC4_SHA_US 在 IBM i 上具有值 05 ，但在 AIX, Linux, and Windows 系统上不是有效的规范。 因此，如果在 CLUSRCVR 上使用定制 SSLCIPH 参数，那么所有生成的自动定义集群发送方通道都应该驻留在底层 TLS 支持实现此 CipherSpec 的平台上，并且可以在这些平台上使用定制值对其进行指定。 如果无法为将在整个集群中理解的 SSLCIPH 参数选择值，那么将需要通道自动定义出口以将其更改为正在使用的平台将理解的内容。 尽可能使用文本 CipherSpec 字符串 (例如 TLS_RSA_WITH_AES_128_CBC_SHA)。

SSLCRLNL 参数适用于单个队列管理器，并且不会传播到集群中的其他队列管理器。

将集群队列管理器和通道升级到 SSL/TLS

一次升级一个集群通道，在 CLUSSDR 通道之前更改所有 CLUSRCVR 通道。

开始之前

请考虑以下注意事项，因为这些可能会影响您为集群选择的 CipherSpec :

- 某些 CipherSpecs 并非在所有平台上都可用。 请注意选择集群中所有队列管理器都支持的 CipherSpec 。
- 某些 CipherSpecs 可能是当前 IBM MQ 发行版中的新增功能，在较旧发行版中不受支持。 包含在不同 MQ 发行版上运行的队列管理器的集群只能使用每个发行版支持的 CipherSpecs 。

  要在集群中使用新的 CipherSpec ，必须首先将所有集群队列管理器迁移到当前发行版。
- 某些 CipherSpecs 要求使用特定类型的数字证书，尤其是那些使用椭圆曲线密码术的证书。

  注意: 无法在要作为集群的一部分连接在一起的队列管理器上混合使用椭圆曲线签名证书和 RSA 签名证书。

  集群中的队列管理器必须全部使用 RSA 签名的证书，或者全部使用 EC 签名的证书，而不是两者的混合。

  请参阅第 39 页的『IBM MQ 中的数字证书和 CipherSpec 兼容性』以获取更多信息。

将集群中的所有队列管理器升级到 IBM MQ V8 或更高版本 (如果它们尚未处于这些级别)。 分发证书和密钥，以便 TLS 从其中每个证书和密钥工作。

必须先升级队列管理器，然后才能升级到或使用任何别名 CipherSpecs (ANY_TLS13，ANY_TLS13_OR_HIGHER， ANY_TLS12， ANY_TLS12_OR_HIGHER 等):

- Multi 将集群中的所有 IBM MQ for Multiplatforms 队列管理器升级到 IBM MQ 9.1.4 或更高版本。
- z/OS 将集群中的所有 IBM MQ for z/OS 队列管理器升级到 IBM MQ for z/OS 9.2.0 或更高版本。

你必须

关于此任务

在 CLUSSDR 通道之前更改 CLUSRCVR 通道。

过程

1. 以您喜欢的任何顺序将 CLUSRCVR 通道切换到 TLS， 一次更改一个 CLUSRCVR，并允许更改在更改下一个之前流经集群。

   要点: 确保在将当前通道的更改分布到集群中之前，不会更改反向路径。
2. 可选： 将所有手动 CLUSSDR 通道切换到 TLS。

   除非将 REFRESH CLUSTER 命令与 REPOS(YES) 选项配合使用，否则这不会对集群的操作产生任何影响。

   注: 对于大型集群，使用 **REFRESH CLUSTER** 命令可能会在集群进行时对其造成干扰，并且在集群对象自动将状态更新发送到所有相关队列管理器之后的 27 天时间间隔再次中断。 请参阅在大型集群中刷新可能会影响集群的性能和可用性。
3. 使用 DISPLAY CLUSQMGR 命令来确保新的安全性配置已在整个集群中传播。
4. 重新启动通道以使用 TLS 并运行 REFRESH SECURITY (SSL)。

相关概念

第 368 页的『启用 CipherSpecs』
通过在 **DEFINE CHANNEL** 或 **ALTER CHANNEL** MQSC 命令中使用 **SSLCIPH** 参数来启用 CipherSpec 。

第 39 页的『IBM MQ 中的数字证书和 CipherSpec 兼容性』
本主题通过概述 CipherSpecs 与 IBM MQ 中的数字证书之间的关系，提供有关如何为安全策略选择相应 CipherSpecs 和数字证书的信息。

相关信息

集群：使用 REFRESH CLUSTER 最佳实践

在集群队列管理器和通道上禁用 SSL/TLS

要关闭 TLS， 请将 SSLCIPH 参数设置为 ' '。 在集群通道上单独禁用 TLS， 在集群发送方通道之前更改所有集群接收方通道。

关于此任务

一次更改一个集群接收方通道，并允许更改流经集群，然后再更改下一个集群。

要点: 确保在将当前通道的更改分布到集群中之前，不会更改反向路径。

过程

1. 将 SSLCIPH 参数的值设置为 ' '，这是单引号 IBM i 或 *NONE on IBM i 中的空字符串。

   可以按您喜欢的任何顺序关闭集群接收方通道上的 TLS。

   请注意，这些更改在您使 TLS 处于活动状态的通道上以相反方向流动。
2. 使用命令 **DISPLAY CLUSQMGR(*)** ALL 检查新值是否反映在所有其他队列管理器中。

3. 关闭所有手动集群发送方通道上的 TLS。

   除非将 **REFRESH CLUSTER** 命令与 REPOS（YES）选项配合使用，否则这不会对集群的操作产生任何影响。

   对于大型集群，当集群对象自动向所有相关队列管理器发送状态更新时，使用 **REFRESH CLUSTER** 命令可能会在集群进行中时对集群造成干扰，此后还会定期执行此操作。 请参阅 在大型集群中刷新可能会影响集群的性能和可用性 以获取更多信息。
4. 停止并重新启动集群发送方通道。

发布/预订安全性

发布/预订中涉及的组件和交互描述为后续更详细的说明和示例的简介。

发布和预订主题涉及多个组件。 在 第 425 页的图 22 中说明了它们之间的某些安全关系，并在以下示例中进行了描述。

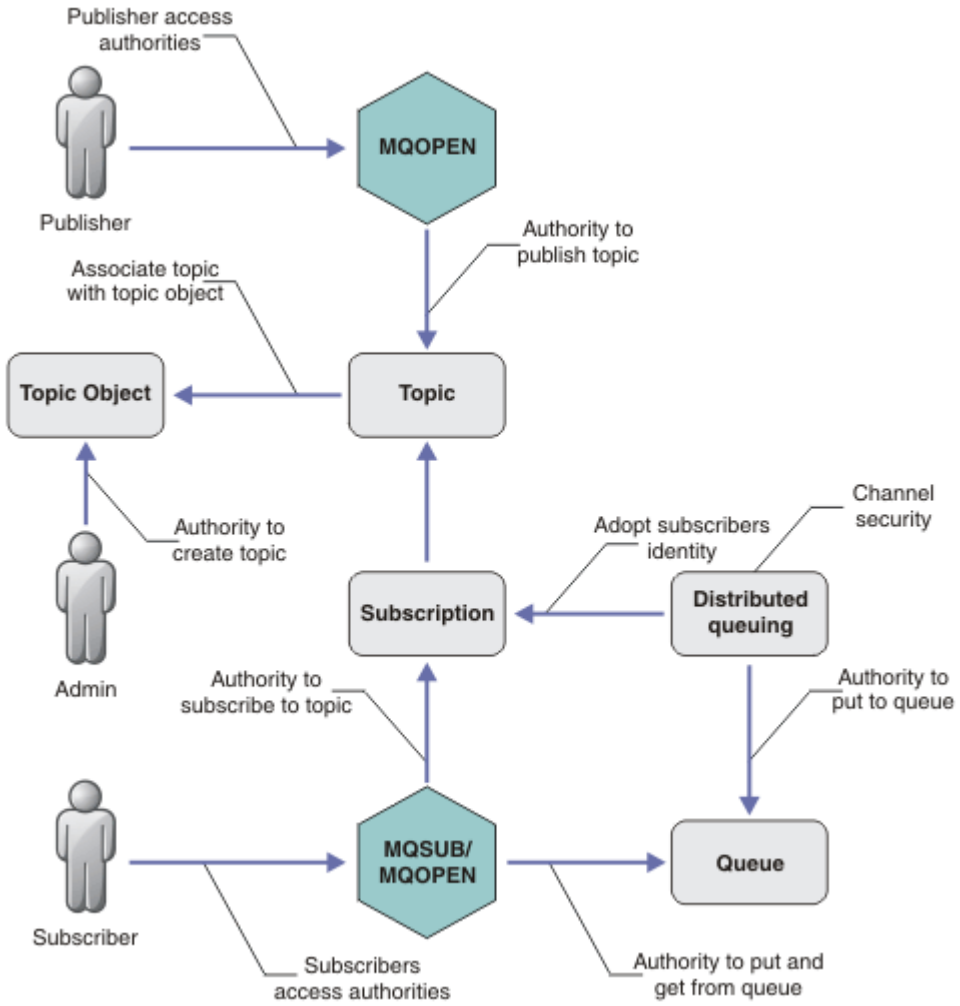


图 *22:* 发布/预订各种组件之间的安全关系

主题

主题由主题字符串标识，并且通常组织到树中，请参阅 主题树。 您需要将主题与主题对象相关联，以控制对该主题的访问权。 第 427 页的『主题安全模型』 说明了如何使用主题对象保护主题。

管理主题对象

您可以通过将命令 **setmqaut** 与管理主题对象的列表配合使用来控制谁有权访问某个主题以及出于何种目的。 请参阅示例 第 431 页的『授予用户预订主题的访问权』 和 第 437 页的『授予用户访问权以发布到主题』。

z/OS 要控制对 z/OS 上主题对象的访问，请参阅 主题安全性的概要文件。

预订

通过创建预订来预订一个或多个主题，该预订提供可包含通配符的主题字符串，以与发布的主题字符串相匹配。 有关更多详细信息，请参阅:

使用主题对象进行预订
第 428 页的『使用主题对象名称进行预订』

使用主题进行预订
第 429 页的『使用主题节点不存在的主题字符串进行预订』

使用带有通配符的主题进行预订
第 429 页的『使用包含通配符的主题字符串进行预订』

预订包含有关订户的标识以及要将发布的目标队列的标识的信息。 它还包含有关如何将发布放在目标队列上的信息。

除了定义哪些订户有权预订特定主题外，您还可以将预订限制为由单个订户使用。 您还可以控制将发布放在目标队列上时队列管理器使用的有关订户的信息。 请参阅第 442 页的『预订安全性』。

队列

目标队列是安全的重要队列。 它是订户的本地出版物，与预订匹配的出版物将放置在该出版物上。 您需要从两个角度考虑对目标队列的访问:

1. 将发布放在目标队列上。
2. 正在将发布从目标队列中获取。

队列管理器使用订户提供的标识将发布放到目标队列上。 订户或已委派用于获取发布的任务的程序会将消息从队列中取出。 请参阅第 429 页的『对目标队列的权限』。

没有主题对象别名，但您可以使用别名队列作为主题对象的别名。 如果执行此操作，以及检查使用发布或预订主题的权限，那么队列管理器将检查使用队列的权限。

第 444 页的『队列管理器之间的发布/预订安全性』

将使用本地身份和权限在本地队列管理器上检查您发布或预订主题的许可权。 授权不取决于是否定义了主题，也不取决于定义了主题的位置。 因此，使用集群主题时，需要对集群中的每个队列管理器执行主题授权。

注: 主题的安全模型与队列的安全模型不同。 您可以通过在本地为每个集群队列定义队列别名来实现队列的相同结果。

队列管理器在集群中交换预订。 在大多数 IBM MQ 集群配置中，使用 PUTAUT=DEF 配置通道以使用通道进程的权限将消息放入目标队列。 您可以修改通道配置以使用 PUTAUT=CTX，从而要求预订用户有权将预订传播到集群中的另一个队列管理器。

第 444 页的『队列管理器之间的发布/预订安全性』 描述了如何更改通道定义，以控制允许谁将预订传播到集群中的其他服务器。

授权

您可以对主题对象应用授权，就像队列和其他对象一样。 有三个授权操作 (pub， sub 和 resume) 只能应用于主题。 在 指定不同对象类型的权限中描述了详细信息。

函数调用

在发布和预订程序 (例如在排队的程序中) 中，将在打开，创建，更改或删除对象时进行授权检查。 当执行 MQPUT 或 MQGET MQI 调用以放置和获取发布时，不会执行检查。

要发布主题，请对主题执行 MQOPEN，这将执行授权检查。 使用不执行授权检查的 MQPUT 命令将消息发布到主题句柄。

要预订主题，通常可以执行 MQSUB 命令来创建或恢复预订，还可以打开目标队列以接收发布。 或者，执行单独的 MQOPEN 以打开目标队列，然后执行 MQSUB 以创建或恢复预订。

无论您使用哪种调用，队列管理器都会检查您是否可以预订主题，并从目标队列中获取生成的发布内容。 如果目标队列是非受管队列，那么还会进行授权检查，以确保队列管理器能够将发布放在目标队列上。 它使用从匹配预订中采用的身份。 假定队列管理器始终能够将发布放在受管目标队列上。

角色

用户在运行发布/预订应用程序时涉及四个角色:

1. 发布者
2. 订户
3. 主题管理员
4. IBM MQ 管理员-组的成员 mqm

定义具有对应于发布，预订和主题管理角色的相应权限的组。 然后，您可以将主体分配给这些组，以授权它们执行特定的发布和预订任务。

此外，您需要将管理操作权限扩展至负责移动发布和预订的队列和通道的管理员。

主题安全模型

只有已定义的主题对象才能具有关联的安全性属性。 有关主题对象的描述，请参阅 管理主题对象。 安全性属性指定是否允许指定的用户标识或安全组对每个主题对象执行预订或发布操作。

安全性属性与主题树中的相应管理节点相关联。 在预订或发布操作期间对特定用户标识进行权限检查时，授予的权限基于关联主题树节点的安全性属性。

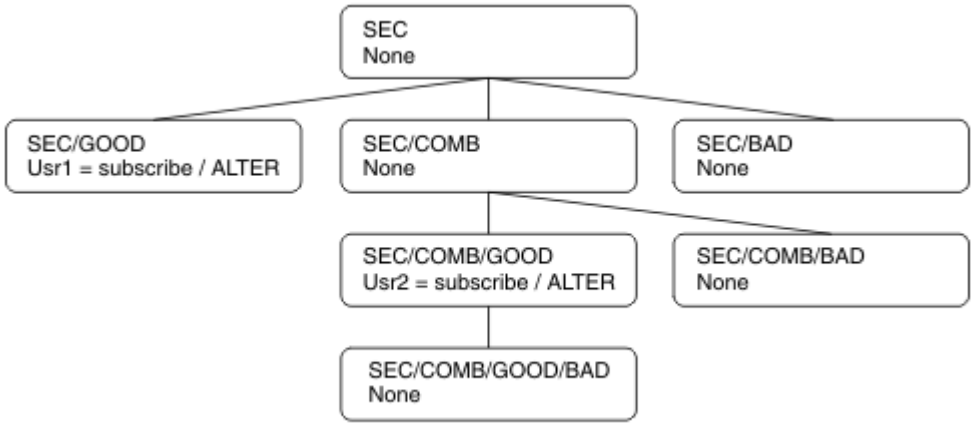
安全性属性是访问控制表，指示特定操作系统用户标识或安全组对主题对象具有的权限。

请考虑以下示例，其中主题对象是使用所显示的安全性属性或权限定义的:

表 *87:* 示例主题对象权限

主题名称	**主题字符串**	**权限-多平台**	**z/OS 权限**
SECROOT	SEC	None	None
SECGOOD	SEC/GOOD	usr1+subscribe	ALTER HLQ.SUBSCRIBE.SECGOOD
SECBAD	SEC/BAD	None	None HLQ.SUBSCRIBE.SECBAD
SECCOMB	SEC/COMB	None	None HLQ.SUBSCRIBE.SECCOMB
SECCOMBB	SEC/COMB/ GOOD/BAD	None	None HLQ.SUBSCRIBE.SECCOMBB
SECCOMBG	SEC/COMB/GOOD	usr2+subscribe	ALTER HLQ.SUBSCRIBE.SECCOMBG
SECCOMBN	SEC/COMB/BAD	None	None HLQ.SUBSCRIBE.SECCOMBN

每个节点上具有关联安全属性的主题树可以如下所示:



列出的示例提供了以下权限:

- 在树 /SEC 的根节点上，没有用户对该节点具有权限。
- usr1 已被授予对对象 /SEC/GOOD 的预订权限
- usr2 已被授予对对象 /SEC/COMB/GOOD 的预订权限

使用主题对象名称进行预订

通过指定 MQCHAR48 名称预订主题对象时，将找到主题树中的相应节点。 如果与节点关联的安全性属性指示用户具有预订权限，那么将授予访问权。

如果未授予用户访问权，那么树中的父节点将确定用户是否有权在父节点级别进行预订。 如果是，那么将授予访问权。 如果没有，那么将考虑该节点的父代。 递归将继续，直到找到向用户授予预订权限的节点为止。 当在没有授予权限的情况下考虑根节点时，递归将停止。 在后一种情况下，拒绝访问。

简而言之，如果路径中的任何节点授予预订该用户或应用程序的权限，那么允许订户在该节点或主题树中该节点下方的任何位置进行预订。

示例中的根节点为 SEC。

如果访问控制表指示用户标识本身具有权限，或者用户标识所属的操作系统安全组具有权限，那么将授予用户预订权限。

因此，例如:

- 如果 usr1 尝试使用主题字符串 SEC/GOOD 进行预订，那么将允许该预订，因为用户标识有权访问与该主题关联的节点。 但是，如果 usr1 尝试使用主题字符串 SEC/COMB/GOOD 进行预订，那么将不允许进行预订，因为用户标识无权访问与其关联的节点。
- 如果 usr2 尝试预订，那么将允许使用主题字符串 SEC/COMB/GOOD 进行预订，因为用户标识有权访问与该主题关联的节点。 但是，如果 usr2 尝试预订 SEC/GOOD ，那么将不允许预订，因为用户标识无权访问与其关联的节点。
- 如果 usr2 尝试使用主题字符串 SEC/COMB/GOOD/BAD 进行预订，那么将允许进行预订，因为用户标识有权访问父节点 SEC/COMB/GOOD。
- 如果 usr1 或 usr2 尝试使用主题字符串 /SEC/COMB/BAD 进行预订，那么将不允许进行预订，因为它们无权访问与其关联的主题节点或该主题的父节点。

指定不存在的主题对象的名称的预订操作将导致 MQRC_UNKNOWN_OBJECT_NAME 错误。

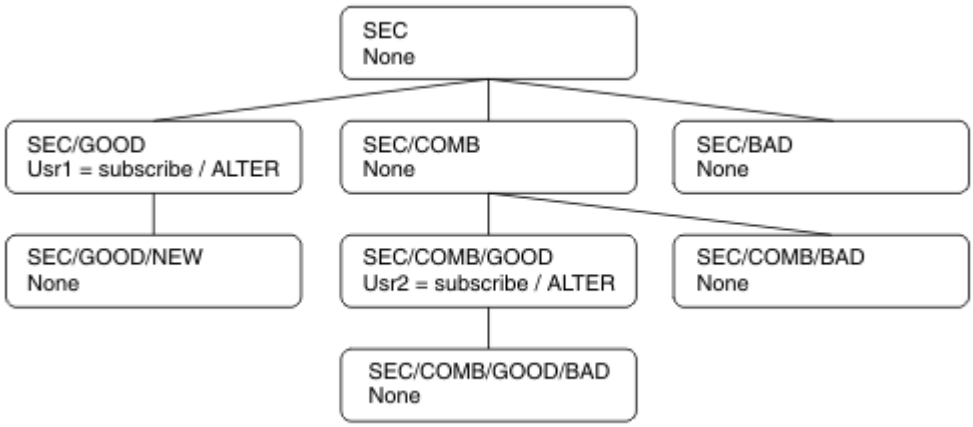
使用存在主题节点的主题字符串进行预订

此行为与通过 MQCHAR48 对象名指定主题时的行为相同。

使用主题节点不存在的主题字符串进行预订

请考虑应用程序预订的情况，指定表示主题树中当前不存在的主题节点的主题字符串。 将按照上一部分中的概述来执行权限检查。 此检查从主题字符串所表示的父节点开始。 如果授予了权限，那么将在主题树中创建表示主题字符串的新节点。

例如， usr1 尝试预订主题 SEC/GOOD/NEW。 已授予权限，因为 usr1 具有对父节点 SEC/GOOD 的访问权。 将在树中创建新的主题节点，如下图所示。 新主题节点不是主题对象，它没有任何直接关联的安全属性; 这些属性继承自其父代。



使用包含通配符的主题字符串进行预订

请考虑使用包含通配符的主题字符串进行预订的情况。 将对主题树中与主题字符串的标准部分匹配的节点执行权限检查。

因此，如果应用程序预订了 SEC/COMB/GOOD/*，那么将按照主题树中节点 SEC/COMB/GOOD 上的前两个部分中概述的那样执行权限检查。

同样，如果应用程序需要预订 SEC/COMB/*/GOOD，那么将在节点 SEC/COMB 上执行权限检查。

对目标队列的权限

预订主题时，其中一个参数是已打开用于输出以接收发布的队列的句柄 hobj 。

如果未指定 hobj ，但该值为空白，那么将在满足以下条件时创建受管队列:

- 已指定 MQSO_MANAGED 选项。
- 该预订不存在。
- 指定了 create。

如果 hobj 为空，并且您正在变更或恢复现有预订，那么先前提供的目标队列可以是受管队列，也可以是非受管队列。

发出 MQSUB 请求的应用程序或用户必须具有将消息放入其提供的目标队列的权限; 实际上具有将已发布的消息放入该队列的权限。 权限检查遵循队列安全性检查的现有规则。

安全性检查包括备用用户标识和上下文安全性检查 (如果需要)。 要能够设置任何 "身份" 上下文字段，必须指定 MQSO_SET_IDENTITY_CONTEXT 选项以及 MQSO_CREATE 或 MQSO_ALTER 选项。 不能对 MQSO_RESUME 请求设置任何 "身份" 上下文字段。

如果目标是受管队列，那么不会对该受管目标执行安全性检查。 如果允许您预订主题，那么假定您可以使用受管目标。

使用主题节点所在的主题名称或主题字符串进行发布

用于发布的安全模型与用于预订的安全模型相同，但通配符除外。 发布不包含通配符; 因此不存在包含要考虑的通配符的主题字符串。

用于发布和预订的权限不同。 用户或组可以具有执行一个操作的权限，而不必执行另一个操作。

通过指定 MQCHAR48 名称或主题字符串发布到主题对象时，将找到主题树中的相应节点。 如果与主题节点关联的安全性属性指示用户具有发布权限，那么将授予访问权。

如果未授予访问权，那么树中的父节点将确定用户是否具有在该级别发布的权限。 如果是，那么将授予访问权。 如果不存在，那么递归将继续，直到找到向用户授予发布权限的节点为止。 当在没有授予权限的情况下考虑根节点时，递归将停止。 在后一种情况下，拒绝访问。

简而言之，如果路径中的任何节点授予向该用户或应用程序发布的权限，那么允许发布者在该节点或该节点下的主题树中的任何位置发布。

使用主题节点不存在的主题名称或主题字符串进行发布

与预订操作一样，当应用程序发布，指定表示主题树中当前不存在的主题节点的主题字符串时，将从该主题字符串所表示的节点的父代开始执行权限检查。 如果授予了权限，那么将在主题树中创建表示主题字符串的新节点。

使用解析为主题对象的别名队列进行发布

如果使用解析为主题对象的别名队列进行发布，那么将在别名队列及其解析为的底层主题上进行安全性检查。

别名队列上的安全性检查将验证用户是否有权将消息放在该别名队列上，而主题上的安全性检查将验证用户是否可以发布到该主题。 当别名队列解析为另一个队列时，不会对底层队列进行检查。 对于主题和队列，将以不同方式执行权限检查。

关闭预订

如果未在此句柄下创建预订，那么如果使用 MQCO_REMOVE_SUB 选项关闭预订，那么还会进行其他安全性检查。

执行安全性检查以确保您具有正确的权限来执行此操作，因为该操作会导致除去预订。 如果与主题节点关联的安全性属性指示用户具有权限，那么将授予访问权。 如果没有，那么将考虑树中的父节点，以确定用户是否有权关闭预订。 递归将继续，直到授予权限或到达根节点为止。

定义，变更和删除预订

当以管理方式创建预订 (而不是使用 MQSUB API 请求) 时，不会执行预订安全性检查。 管理员已通过命令获得此权限。

执行安全性检查以确保可以将发布放在与预订关联的目标队列上。 执行检查的方式与执行 MQSUB 请求的方式相同。

用于这些安全性检查的用户标识取决于发出的命令。 如果指定了 **SUBUSER** 参数，那么将影响执行检查的方式，如 第 430 页的表 88 中所示:

表 *88:* 用于对命令进行安全性检查的用户标识

命令	指定了 SUBUSER 且为空	指定了 SUBUSER 并已完成	未指定 SUBUSER
	使用管理员标识		使用 LIKE 预订中的用户标识
	使用管理员标识		使用.DEFAULT.SUSYSTEMB 预订-如果为空，请使用管理员标识

表 88: 用于对命令进行安全性检查的用户标识 (继续)			
命令	**指定了 SUBUSER 且为空**	**指定了 SUBUSER 并已完成**	**未指定 SUBUSER**
	使用管理员标识		使用现有预订中的用户标识

使用 DELETE SUB 命令删除预订时执行的唯一安全性检查是命令安全性检查。

示例发布/预订安全性设置

本部分描述了以允许根据需要应用安全控制的方式在主题上设置访问控制的方案。

授予用户预订主题的访问权

本主题是任务列表中的第一个主题，用于告诉您如何授予多个用户对主题的访问权。

关于此任务

此任务假定不存在任何管理主题对象，也没有为预订或发布定义任何概要文件。 应用程序正在创建新预订，而不是恢复现有预订，并且仅使用主题字符串来执行此操作。

应用程序可以通过提供主题对象，主题字符串或两者的组合来进行预订。 无论应用选择哪种方式，效果都是在主题树中的某个点进行订阅。 如果主题树中的此点由管理主题对象表示，那么将根据该主题对象的名称来检查安全概要文件。

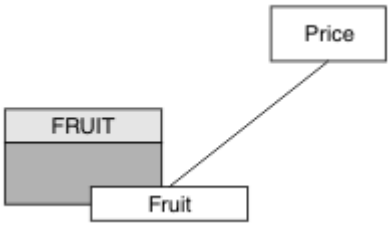


图 23: 主题对象访问示例

表 89: 主题对象访问示例		
Topic	**需要预订访问权**	**主题对象**
价格	无用户	None
价格/水果	USER1	水果

定义新主题对象，如下所示:

过程

1. 发出 MQSC 命令 `DEF TOPIC(FRUIT) TOPICSTR('Price/Fruit')`。
2. 授予访问权，如下所示:

- z/OS **z/OS** :

  通过授予用户对 `hlq.SUBSCRIBE.FRUIT` 概要文件的访问权，授予对 USER1 的访问权以预订主题 “Price/Fruit”。 使用以下 RACF 命令执行此操作:

```
RDEFINE MXTOPIC hlq.SUBSCRIBE.FRUIT UACC(NONE)
PERMIT hlq.SUBSCRIBE.FRUIT CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
```

- Multi **多平台**:

  通过授予用户对 FRUIT 对象的访问权，授予对 USER1 的访问权以预订主题 “Price/Fruit”。 执行此操作，对平台使用授权命令:

  ALW **AIX, Linux, and Windows 系统**

  ```
  setmqaut -t topic -n FRUIT -p USER1 +sub
  ```

  IBM i **IBM i**

  ```
  GRTMQAUT OBJ(FRUIT) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
  ```

结果

当 USER1 尝试预订主题 “Price/Fruit” 时，结果为成功。

当 USER2 尝试预订主题 “Price/Fruit” 时，结果是失败，并显示 MQRC_NOT_AUTHORIZED 消息，以及:

- z/OS 在 z/OS 上，在控制台上看到以下消息，这些消息显示通过已尝试的主题树的完整安全路径:

  ```
  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.FRUIT ...

  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
  ```

- ALW 在 AIX, Linux, and Windows 上，发生以下授权事件:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit"
  ```

- IBM i 在 IBMi 上，发生以下授权事件:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit"
  ```

请注意，这是您所看到的内容的说明; 而不是所有字段。

授予用户访问权以预订树中更深层的主题

此主题是任务列表中的第二个主题，用于告诉您如何授予多个用户对主题的访问权。

开始之前

本主题使用 第 431 页的『授予用户预订主题的访问权』中描述的设置。

关于此任务

如果应用程序在其中进行预订的主题树中的点未由管理主题对象表示，请将该树向上移动，直到找到最接近的父管理主题对象为止。 将根据该主题对象的名称来检查安全概要文件。

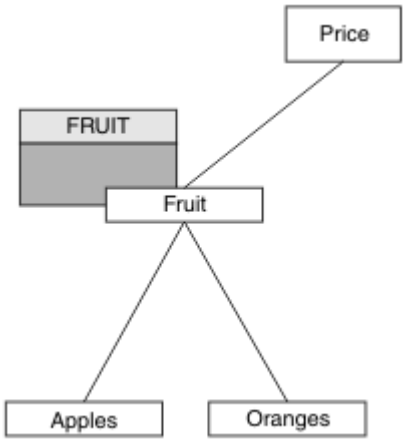


图 *24:* 授予对主题树中主题的访问权的示例

表 *90:* 示例主题和主题对象的访问需求

Topic	需要预订访问权	主题对象
价格	无用户	None
价格/水果	USER1	水果
价格/水果/苹果	USER1	
价格/水果/橙子	USER1	

在 第 431 页的『授予用户预订主题的访问权』中，通过授予 USER1 对主题 “Price/Fruit” 的访问权 访问 z/OS 上的 hlq.SUBSCRIBE.FRUIT 概要文件以及 对 Multiplatforms 版上的 FRUIT 概要文件的预订访问权，授予其预订访问权。 此单个概要文件还授予 USER1 访问权以预订 “Price/Fruit/Apples”，“Price/Fruit/Oranges” 和 “Price/Fruit/#”。

当 USER1 尝试预订主题 “Price/Fruit/Apples” 时，结果为成功。

当 USER2 尝试预订主题 “Price/Fruit/Apples” 时，结果是失败，并显示 MQRC_NOT_AUTHORIZED 消息，以及:

- z/OS 在 z/OS 上，在控制台上看到以下消息，这些消息显示通过已尝试的主题树的完整安全路径:

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- Multi 在 Multiplatforms 版上，以下授权事件:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Apples"
```

请注意下列事项:

- z/OS 您在 z/OS 上接收到的消息与在先前任务中接收到的消息相同，因为相同的主题对象和概要文件正在控制访问权。
- Multi 在 Multiplatforms 版上接收到的事件消息与先前任务中接收到的事件消息类似，但实际主题字符串不同。

授予其他用户访问权以仅预订树中较深的主题

此主题是任务列表中的第三个主题，用于指示如何授予多个用户预订主题的访问权。

开始之前

本主题使用 第 432 页的『授予用户访问权以预订树中更深层的主题』中描述的设置。

关于此任务

在 第 432 页的『授予用户访问权以预订树中更深层的主题』中，拒绝 USER2 访问主题 “Price/Fruit/Apples”。 本主题告诉您如何授予对该主题的访问权，但不授予对任何其他主题的访问权。

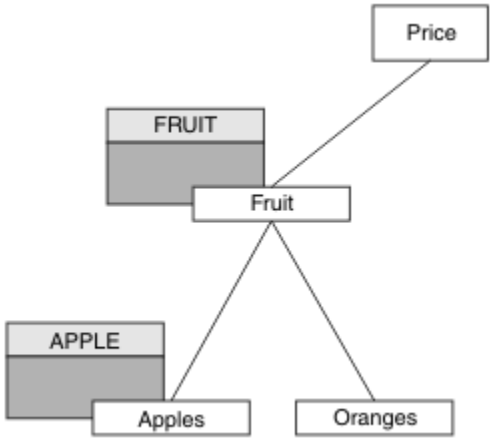


图 *25:* 授予对主题树中特定主题的访问权

表 *91:* 示例主题和主题对象的访问需求

Topic	需要预订访问权	主题对象
价格	无用户	None
价格/水果	USER1	水果
价格/水果/苹果	USER1 和 USER2	Apple
价格/水果/橙子	USER1	

定义新主题对象，如下所示:

过程

1. 发出 MQSC 命令 `DEF TOPIC(APPLE) TOPICSTR('Price/Fruit/Apples')`。
2. 授予访问权，如下所示:

- z/OS **z/OS** :

  在 第 432 页的『授予用户访问权以预订树中更深层的主题』 USER1 中，通过授予用户对 `hlq.SUBSCRIBE.FRUIT` 概要文件的访问权，授予了预订主题 “Price/Fruit/Apples” 的访问权。

  此单个概要文件还授予了 USER1 预订 “Price/Fruit/Oranges” “Price/Fruit/#” 的访问权，即使添加了新主题对象以及与其关联的概要文件，此访问权也会保留。

通过授予用户对 hlq.SUBSCRIBE.APPLE 概要文件的访问权，授予对 USER2 的访问权以预订主题 “Price/Fruit/Apples”。 使用以下 RACF 命令执行此操作:

```
RDEFINE MXTOPIC hlq.SUBSCRIBE.APPLE UACC(NONE)
PERMIT hlq.SUBSCRIBE.FRUIT APPLE(MXTOPIC) ID(USER2) ACCESS(ALTER)
```

- Multi 多平台:

  在 第 432 页的『授予用户访问权以预订树中更深层的主题』 USER1 中，通过授予用户对 FRUIT 概要文件的预订访问权，授予了对主题 “Price/Fruit/Apples” 的预订访问权。

  此单个概要文件还授予了 USER1 访问权以预订 “Price/Fruit/Oranges” 和 “Price/Fruit/#”，即使添加了新主题对象以及与其关联的概要文件，此访问权也会保留。

  通过授予用户对 APPLE 概要文件的预订访问权，授予对 USER2 的访问权以预订主题 “Price/Fruit/Apples”。 执行此操作，对平台使用授权命令:

  ALW **AIX, Linux, and Windows 系统**

  ```
  setmqaut -t topic -n APPLE -p USER2 +sub
  ```

  IBM i **IBM i**

  ```
  GRTMQAUT OBJ(APPLE) OBJTYPE(*TOPIC) USER(USER2) AUT(*SUB)
  ```

结果

z/OS 在 z/OS 上，当 USER1 尝试预订主题 “Price/Fruit/Apples” 时，hlq.SUBSCRIBE.APPLE 概要文件上的第一次安全性检查失败，但在上移树时， hlq.SUBSCRIBE.FRUIT 概要文件允许 USER1 预订，因此预订成功，并且不会向 MQSUB 调用发送返回码。 但是，将为第一次检查生成 RACF ICH 消息:

```
ICH408I USER(USER1   ) ...
  hlq.SUBSCRIBE.APPLE ...
```

当 USER2 尝试预订主题 “Price/Fruit/Apples” 时，结果是成功的，因为安全性检查通过了第一个概要文件。

当 USER2 尝试预订主题 “Price/Fruit/Oranges” 时，结果是失败，并显示 MQRC_NOT_AUTHORIZED 消息，以及:

- z/OS 在 z/OS 上，在控制台上看到以下消息，这些消息显示通过已尝试的主题树的完整安全路径:

  ```
  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.FRUIT ...

  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
  ```

- ALW 在 AIX, Linux, and Windows 平台上，发生以下授权事件:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit/Oranges"
  ```

- IBM i 在 IBMi 上，发生以下授权事件:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

z/OS 此设置的缺点是，在 z/OS 上，您在控制台上接收到其他 ICH 消息。 如果以不同的方式保护主题树，那么可以避免此情况。

更改访问控制以避免其他消息

本主题是任务列表中的第四个主题，用于指示如何授予多个用户访问权以预订主题 并避免在 z/OS 上出现其他 RACF ICH408I 消息。

开始之前

本主题增强了 第 434 页的『授予其他用户访问权以仅预订树中较深的主题』 中描述的设置，以便避免出现其他错误消息。

关于此任务

本主题告诉您如何授予对树中更深层主题的访问权，以及如何在没有用户需要时除去对树下的主题的访问权。

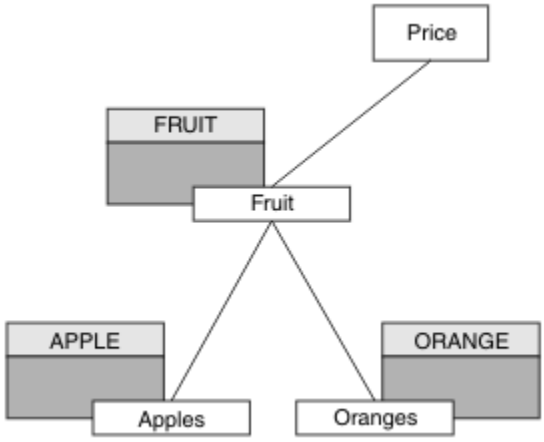


图 *26:* 授予访问控制以避免其他消息的示例。

定义新主题对象，如下所示:

过程

1. 发出 MQSC 命令 `DEF TOPIC(ORANGE) TOPICSTR('Price/Fruit/Oranges')`。
2. 授予访问权，如下所示:
   - z/OS **z/OS** :

     定义新概要文件并添加对该概要文件和现有概要文件的访问权。 使用以下 RACF 命令执行此操作:

     ```
     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 多平台:

使用针对平台的授权命令来设置等效访问权:

AIX, Linux, and Windows 系统

```
setmqaut -t topic -n ORANGE -p USER1 +sub
setmqaut -t topic -n APPLE -p USER1 +sub
```

IBM i

```
GRTMQAUT OBJ(ORANGE) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
GRTMQAUT OBJ(APPLE) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
```

结果

在 z/OS 上，当 USER1 尝试预订主题 “Price/Fruit/Apples” 时，hlq.SUBSCRIBE.APPLE 概要文件上的第一次安全性检查成功。

当 USER2 尝试预订主题 “Price/Fruit/Apples” 时，结果是成功的，因为安全性检查通过了第一个概要文件。

当 USER2 尝试预订主题 “Price/Fruit/Oranges” 时，结果是失败，并显示 MQRC_NOT_AUTHORIZED 消息，以及:

- 在 z/OS 上，在控制台上看到以下消息，这些消息显示通过已尝试的主题树的完整安全路径:

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.ORANGE ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- 在 AIX, Linux, and Windows 上，发生以下授权事件:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      ORANGE, FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

- 在 IBM i 上，发生以下授权事件:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      ORANGE, FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

授予用户访问权以发布到主题

此主题是任务列表中的第一个主题，用于告诉您如何授予多个用户对发布主题的访问权。

关于此任务

此任务假定主题树右侧不存在任何管理主题对象，也没有为发布定义任何概要文件。 使用的假设是发布程序仅使用主题字符串。

应用程序可以通过提供主题对象，主题字符串或两者的组合来发布到主题。 无论应用程序选择哪种方式，效果都是在主题树中的某个点发布。 如果主题树中的此点由管理主题对象表示，那么将根据该主题对象的名称来检查安全概要文件。 例如：

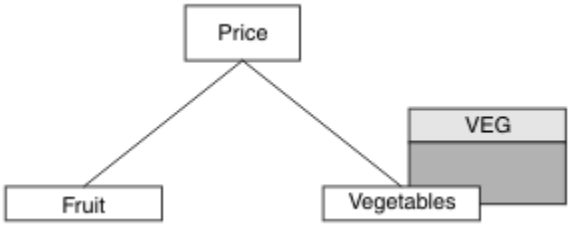


图 *27:* 授予对主题的发布访问权

表 *92:* 发布访问权需求示例		
Topic	**需要发布访问权**	**主题对象**
价格	无用户	无
价格/蔬菜	USER1	VEG

定义新主题对象，如下所示：

过程

1. 发出 MQSC 命令 `DEF TOPIC(VEG) TOPICSTR('Price/Vegetables')`。
2. 授予访问权，如下所示：

- z/OS **z/OS** ：

  通过授予用户对 `hlq.PUBLISH.VEG` 概要文件的访问权，授予对 USER1 的访问权以发布到主题 “Price/Vegetables”。 使用以下 RACF 命令执行此操作：

  ```
  RDEFINE MXTOPIC hlq.PUBLISH.VEG UACC(NONE)
  PERMIT hlq.PUBLISH.VEG CLASS(MXTOPIC) ID(USER1) ACCESS(UPDATE)
  ```

- 其他平台：

  通过授予用户对 VEG 概要文件的访问权，授予对 USER1 的访问权以发布到主题 “Price/Vegetables”。 执行此操作，对平台使用授权命令：

  ALW **AIX, Linux, and Windows 系统**

  ```
  setmqaut -t topic -n VEG -p USER1 +pub
  ```

  IBM i **IBM i**

  ```
  GRTMQAUT OBJ(VEG) OBJTYPE(*TOPIC) USER(USER1) AUT(*PUB)
  ```

结果

当 USER1 尝试发布到主题 “Price/Vegetables” 时，结果是成功; 即 MQOPEN 调用成功。

当 USER2 尝试发布到主题 “Price/Vegetables” 时， MQOPEN 调用将失败，并返回 `MQRC_NOT_AUTHORIZED` 消息，以及：

- z/OS 在 z/OS 上，在控制台上看到以下消息，这些消息显示通过已尝试的主题树的完整安全路径:

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.VEG ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- ALW 在其他平台上，以下授权事件:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables"
```

- IBM i 在 IBMi 上，发生以下授权事件:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables"
```

请注意，这是您所看到的内容的说明; 而不是所有字段。

授予用户访问权以发布到树中更深层的主题

此主题是任务列表中的第二个主题，它告诉您如何授予多个用户对主题的发布访问权。

开始之前

本主题使用 第 437 页的『授予用户访问权以发布到主题』中描述的设置。

关于此任务

如果应用程序发布的主题树中的点未由管理主题对象表示，请向上移动该树，直到找到最接近的父管理主题对象为止。 将根据该主题对象的名称来检查安全概要文件。

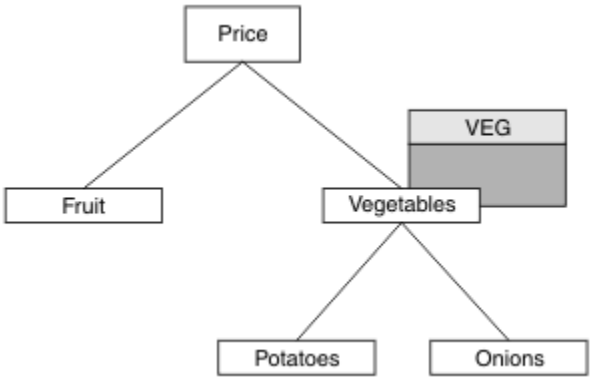


图 *28:* 授予对主题树中主题的发布访问权

表 *93:* 发布访问权需求示例

Topic	需要预订访问权	主题对象
价格	无用户	None
价格/蔬菜	USER1	VEG

表 93: 发布访问权需求示例 (继续)		
Topic	**需要预订访问权**	**主题对象**
价格/蔬菜/土豆	USER1	
价格/蔬菜/洋葱	USER1	

在先前任务 USER1 中，已通过授予其 访问 z/OS 上的 hlq.PUBLISH.VEG 概要文件或 对 Multiplatforms 版上的 VEG 概要文件的发布访问权来授予其发布主题 “Price/Vegetables/Potatoes” 的访问权。 此单个概要文件还授予 USER1 在 “Price/Vegetables/Onions”上发布的访问权。

当 USER1 尝试在主题 “Price/Vegetables/Potatoes” 上发布时，结果是成功; 即 MQOPEN 调用成功。

当 USER2 尝试预订主题 “Price/Vegetables/Potatoes” 时，结果为失败; 即， MQOPEN 调用失败，并带有 MQRC_NOT_AUTHORIZED 消息，以及:

- z/OS 在 z/OS 上，在控制台上看到以下消息，这些消息显示通过已尝试的主题树的完整安全路径:

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.VEG ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- Multi 在 Multiplatforms 版上，以下授权事件:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables/Potatoes"
```

请注意下列事项:

- z/OS 您在 z/OS 上接收到的消息与在先前任务中接收到的消息相同，因为相同的主题对象和概要文件正在控制访问权。
- Multi 在 Multiplatforms 版上接收到的事件消息与先前任务中接收到的事件消息类似，但实际主题字符串不同。

授予对发布和预订的访问权

此主题是任务列表中的最后一个主题，用于告诉您如何授予多个用户发布和预订主题的访问权。

开始之前

本主题使用 第 439 页的『授予用户访问权以发布到树中更深层的主题』中描述的设置。

关于此任务

在先前的任务 USER1 中，授予了预订主题 “Price/Fruit”的访问权。 本主题告诉您如何向该用户授予发布到该主题的访问权。

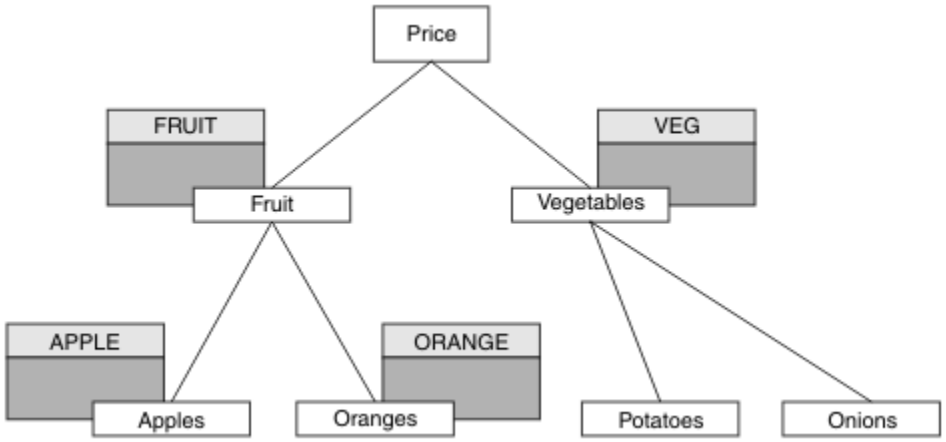


图 *29:* 授予发布和预订访问权

表 *94:* 发布和预订访问权需求示例

Topic	需要预订访问权	需要发布访问权	主题对象
价格	无用户	无用户	None
价格/水果	USER1	USER1	水果
价格/水果/苹果	USER1 和 USER2		Apple
价格/水果/橙子	USER1		橙色

过程

授予访问权，如下所示:

- z/OS **z/OS** :

  在先前的任务 USER1 中，通过授予用户对 `hlq.SUBSCRIBE.FRUIT` 概要文件的访问权，授予了对主题 “Price/Fruit” 的预订访问权。

  要发布到 “Price/Fruit” 主题，请授予对 `hlq.PUBLISH.FRUIT` 概要文件的 USER1 访问权。 使用以下 RACF 命令执行此操作:

  ```
  RDEFINE MXTOPIC hlq.PUBLISH.FRUIT UACC(NONE)
  PERMIT hlq.PUBLISH.FRUIT CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
  ```

- Multi 多平台:

  通过授予用户对 FRUIT 概要文件的发布访问权，授予对 USER1 的访问权以发布到主题 “Price/Fruit”。 执行此操作，对平台使用授权命令:

  ALW **AIX, Linux, and Windows 系统**

  ```
  setmqaut -t topic -n FRUIT -p USER1 +pub
  ```

  IBM i **IBM i**

  ```
  GRTMQAUT OBJ(FRUIT) OBJTYPE(*TOPIC) USER(USER1) AUT(*PUB)
  ```

结果

z/OS 在 z/OS 上，当 USER1 尝试发布到主题 “Price/Fruit” 时，将传递 MQOPEN 调用的安全性检查。

当 USER2 尝试在主题 “Price/Fruit” 上发布时，结果失败并显示 MQRC_NOT_AUTHORIZED 消息，以及:

- z/OS 在 z/OS 上，在控制台上看到以下消息，这些消息显示通过已尝试的主题树的完整安全路径:

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- ALW 在 AIX, Linux, and Windows 平台上，发生以下授权事件:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

- IBM i 在 IBM i 上，发生以下授权事件:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

遵循这些任务的完整集合，为 USER1 和 USER2 提供以下访问权限以发布和预订列出的主题:

表 *95:* 安全示例生成的访问权限的完整列表

Topic	需要预订访问权	需要发布访问权	主题对象
价格	无用户	无用户	None
价格/水果	USER1	USER1	水果
价格/水果/苹果	USER1 和 USER2		Apple
价格/水果/橙子	USER1		橙色
价格/蔬菜		USER1	VEG
价格/蔬菜/土豆			
价格/蔬菜/洋葱			

z/OS 如果您对主题树中不同级别的安全访问有不同的要求，那么仔细规划可确保不会在 z/OS 控制台日志中接收到无关的安全警告。 在树中的正确级别设置安全性可避免误导性安全消息。

预订安全性

MQSO_ALTERNATE_USER_AUTHORITY

AlternateUser 标识字段包含用于验证此 MQSUB 调用的用户标识。 仅当此 AlternateUser 标识有权使用指定的访问选项预订主题时，该调用才能成功，而不管运行应用程序的用户标识是否有权这样做。

MQSO_SET_IDENTITY_CONTEXT

预订将使用 PubAccounting 令牌和 PubApplIdentityData 字段中提供的记帐令牌和应用程序身份数据。

如果指定了此选项，那么将执行相同的授权检查，就像使用带有 MQOO_SET_IDENTITY_CONTEXT 的 MQOPEN 调用访问目标队列一样，但在同样使用 MQSO_MANAGED 选项的情况下，目标队列上没有授权检查。

如果未指定此选项，那么发送到此订户的发布具有与它们相关联的缺省上下文信息，如下所示:

表 *96:* 缺省发布上下文信息

MQMD 中的字段	**使用的值**
UserIdentifier	发布时与预订关联的用户标识 (请参阅 DISPLAY SBSTATUS 上的 SUBUSER 字段)。
AccountingToken	根据环境确定 (如果可能); 否则设置为 MQACT_NONE。
ApplIdentityData	设置为空白。

此选项仅对 MQSO_CREATE 和 MQSO_ALTER 有效。 如果与 MQSO_RESUME 配合使用，那么将忽略 PubAccounting 令牌和 PubApplIdentityData 字段，因此此选项无效。

如果在不使用此选项的情况下更改了预订，而先前该预订提供了身份上下文信息，那么将为已更改的预订生成缺省上下文信息。

如果允许不同用户标识将其与选项 MQSO_ANY_USERID 配合使用的预订由不同用户标识恢复，那么将为现在拥有该预订的新用户标识生成缺省身份上下文，并且将交付包含新身份上下文的任何后续发布。

AlternateSecurityId

这是与 AlternateUser 标识一起传递给授权服务的安全标识，以允许执行相应的授权检查。 仅当指定了 MQSO_ALTERNATE_USER_AUTHORITY 时，才会使用 AlternateSecurityId ，并且 AlternateUserId 字段并非完全空白，直到第一个空字符或字段结束。

MQSO_ANY_USERID 预订选项

指定 MQSO_ANY_USERID 时，订户的身份不会限制为单个用户标识。 这允许任何用户在具有适当权限时更改或恢复预订。 只有单个用户可以在任何时候拥有预订。 尝试恢复使用另一个应用程序当前正在使用的预订将导致调用失败并返回 MQRC_SUBSCRIPTION_IN_USE。

要将此选项添加到现有预订， MQSUB 调用 (使用 MQSO_ALTER) 必须来自与原始预订相同的用户标识。

如果 MQSUB 调用引用了设置了 MQSO_ANY_USERID 的现有预订，并且用户标识与原始预订不同，那么仅当新用户标识有权预订主题时，调用才会成功。 成功完成后，此订户的未来发布将以发布中设置的新用户标识放入订户的队列中。

MQSO_FIXED_USERID

指定 MQSO_FIXED_USERID 时，只能通过单个拥有用户标识来变更或恢复预订。 此用户标识是更改设置此选项的预订的最后一个用户标识，从而除去 MQSO_ANY_USERID 选项，或者如果未发生任何变更，那么是用户标识创建了预订。

如果 MQSUB 动词引用设置了 MQSO_ANY_USERID 的现有预订，并将该预订 (使用 MQSO_ALTER) 更改为使用选项 MQSO_FIXED_USERID ，那么该预订的用户标识现在已固定在此新用户标识上。 仅当新用户标识具有预订主题的权限时，调用才会成功。

如果记录为拥有预订的用户标识以外的用户标识要恢复或变更 MQSO_FIXED_USERID 预订，那么调用将失败，并返回 MQRC_IDENTITY_MATCH。 可以使用 DISPLAY SBSTATUS 命令查看预订的拥有用户标识。

如果未指定 MQSO_ANY_USERID 或 MQSO_FIXED_USERID ，那么缺省值为 MQSO_FIXED_USERID。

队列管理器之间的发布/预订安全性

使用正常通道安全规则将发布/预订内部消息 (例如代理预订和发布) 放入发布/预订系统队列。 本主题中的信息和图突出显示了传递这些消息所涉及的各种流程和用户标识。

本地访问控制

对发布和预订主题的访问权由 发布/预订安全性中描述的本地安全性定义和规则进行管理。 不需要本地主题对象来建立访问控制。 管理员可以选择将访问控制应用于集群主题对象，而不管它们是否存在于集群中。

系统管理员负责对其本地系统进行访问控制。 他们必须信任层次结构或集群集合体的其他成员的管理员对其访问控制策略负责。 由于为每台单独的机器定义了访问控制，因此如果需要精细的级别控制，可能会造成负担。 可能不需要强制实施任何访问控制，或者可以在主题树中的高级对象上定义访问控制。 可以为主题名称空间的每个子部分定义精细级别访问控制。

进行代理预订

组织将其队列管理器连接到队列管理器的信任由正常通道认证方法确认。 如果还允许该可信组织执行分布式发布/预订，那么将执行权限检查。 当通道将消息放入分布式发布/预订队列时，将执行此检查。 例如，如果将消息放入 SYSTEM.INTER.QMGR.CONTROL 队列。 队列权限检查的用户标识取决于接收通道的 PUTAUT 值。 例如，通道的用户标识 MCAUSER，消息上下文，具体取决于值和平台。 有关通道安全性的更多信息，请参阅 通道安全性。

使用远程队列管理器上分布式发布/预订代理程序的用户标识进行代理预订。 例如， 第 444 页的图 30 中的 QM2 。 然后，很容易授予用户对本地主题对象概要文件的访问权，因为该用户标识是在系统中定义的，因此不存在域冲突。

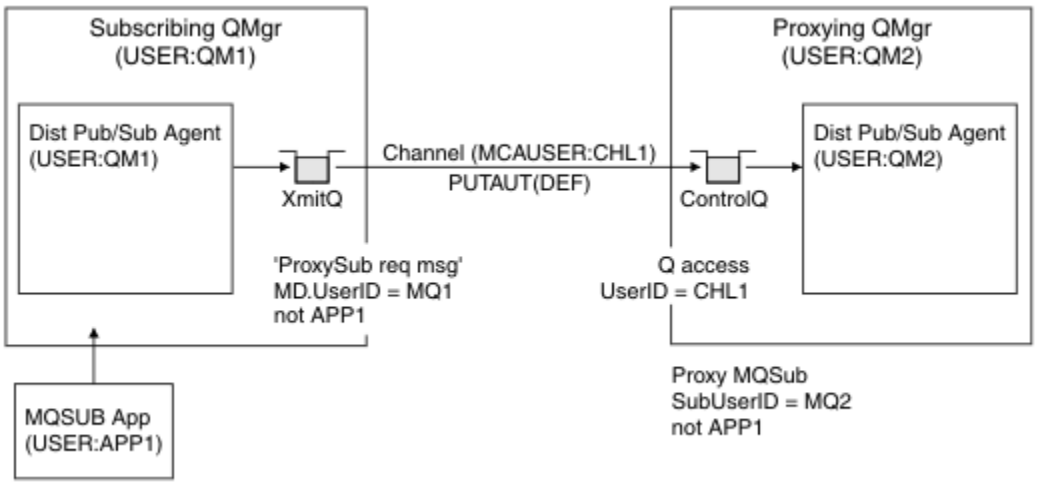


图 *30:* 代理预订安全性，进行预订

发送回远程发布

在发布队列管理器上创建发布时，将为任何代理预订创建发布的副本。 复制的发布的上下文包含进行预订的用户标识的上下文; 第 445 页的图 31 中的 QM2 。 使用作为远程队列的目标队列创建代理预订，因此会将发布消息解析到传输队列。

组织将其队列管理器 QM2 连接到另一个队列管理器 QM1 的信任已通过正常通道认证方法确认。 如果随后允许该可信组织执行分布式发布/预订，那么当通道将发布消息放入分布式发布/预订发布队列 SYSTEM.INTER.QMGR.PUBS 时，将执行权限检查。 队列权限检查的用户标识取决于接收通道的 PUTAUT

值 (例如，通道的用户标识， MCAUSER，消息上下文等，具体取决于值和平台)。 有关通道安全性的更多信息，请参阅 通道安全性。

当发布消息到达预订队列管理器时，将在该队列管理器的权限下完成对主题的另一个 MQPUT ，并且消息的上下文将替换为每个本地订户的上下文，因为每个订户都具有该消息。

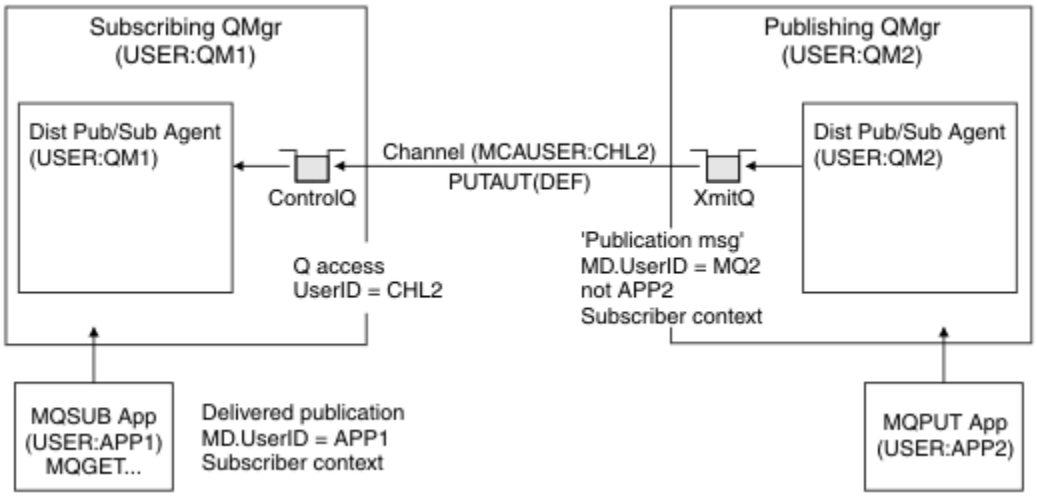


图 *31:* 代理预订安全性，转发发布

在很少考虑安全性的系统上，分布式发布/预订进程可能在 mqm 组中的用户标识下运行，通道上的 MCAUSER 参数为空 (缺省值)，并且根据需要将消息传递到各种系统队列。 无安全保护的系统便于设置概念证明以演示分布式发布/订阅。

在更认真考虑安全性的系统上，这些内部消息受与通过通道的任何消息相同的安全性控制。

如果使用非空白 MCAUSER 和指定必须检查 MCAUSER 的 PUTAUT 值来设置通道，那么必须授予所讨论的 MCAUSER 对 SYSTEM.INTER.QMGR.* 队列的访问权。 如果存在多个不同的远程队列管理器，并且通道在不同的 MCAUSER 标识下运行，那么需要授予所有这些用户标识对 SYSTEM.INTER.QMGR.* 队列的访问权。 例如，在单个队列管理器上配置了多个分层连接时，可能会发生在不同 MCAUSER 标识下运行的通道。

如果使用指定使用消息上下文的 PUTAUT 值设置通道，那么将根据内部消息中的用户标识来检查对 SYSTEM.INTER.QMGR.* 队列的访问。 由于所有这些消息都与来自发送内部消息或发布消息的队列管理器的分布式发布/预订代理程序的用户标识一起放置 (请参阅 第 445 页的图 31)，因此如果要通过此方式设置分布式发布/预订安全性，那么授予对各种系统队列 (每个远程队列管理器一个) 的访问权的用户标识集不会太大。 它仍然具有通道上下文安全性始终存在的所有相同问题; 不同用户标识域的问题以及消息中的用户标识可能未在接收系统上定义的事实。 但是，如果需要，这是完全可以接受的运行方式。

z/OS 系统队列安全性 提供队列列表以及安全设置分布式发布/预订环境所需的访问权。 如果由于安全违例而未能放入任何内部消息或发布内容，那么通道会以正常方式将消息写入日志，并且可以根据正常通道错误处理将消息发送到死信队列。

用于分布式发布/预订的所有队列间管理器消息传递都使用正常通道安全性运行。

有关在主题级别限制发布和代理预订的信息，请参阅 发布/预订安全性。

将缺省用户标识与队列管理器层次结构配合使用

如果您具有在不同平台上运行的队列管理器层次结构，并且正在使用缺省用户标识，请注意，这些缺省用户标识在不同平台之间有所不同，并且在目标平台上可能未知。 因此，在一个平台上运行的队列管理器会拒绝从其他平台上的队列管理器接收到的消息，原因码为 MQRC_NOT_AUTHORIZED。

为了避免消息被拒绝，至少需要将以下权限添加到其他平台上使用的缺省用户标识:

- SYSTEM.BROKER。 队列
- * SYSTEM.BROKER 上的 PUB *SUB 权限。 主题
- SYSTEM.BROKER.CONTROL.QUEUE 队列。

具有队列管理器层次结构的缺省用户标识如下所示:

平台	缺省用户标识
Windows	mqm
AIX and Linux 系统	mqm
IBM i	QMQM
z/OS	通道启动程序地址空间用户标识

如果 IBM i 以外的平台上的队列管理器以分层方式连接到 IBM i 上的队列管理器，请创建并授予对 "qmqm" 用户标识的访问权。

如果 IBM i 或 z/OS 上的队列管理器以分层方式连接到 AIX, Linux, and Windows 上的队列管理器，请创建 "mqm" 用户标识并授予其访问权。

如果 多平台 上的队列管理器以分层方式连接到 z/OS 上的队列管理器，请创建并授予对 z/OS 通道启动程序地址空间用户标识的访问权。

用户标识可以区分大小写。 发端队列管理器 (如果在 多平台上) 强制用户标识为全大写。 接收队列管理器 (如果在 AIX, Linux, and Windows 上) 强制用户标识全部为小写。 因此，必须以小写形式创建 AIX and Linux 系统上创建的所有用户标识。 如果已安装消息出口，那么不会强制将用户标识设置为大写或小写。 必须注意了解消息出口如何处理用户标识。

要避免用户标识转换的潜在问题，请执行以下操作:

- 在 AIX, Linux, and Windows 系统上，确保以小写形式指定用户标识。
- 在 IBM i 和 z/OS 系统上，确保以大写形式指定用户标识。

IBM MQ Console 和 REST API 安全性

通过编辑 mqwebuser.xml 文件中的 mqweb 服务器配置来配置 IBM MQ Console 和 REST API 的安全性。

关于此任务

您可以通过检查 mqweb 服务器的日志文件来跟踪用户操作并审计 IBM MQ Console 和 REST API 的使用情况。

可以使用以下命令对 IBM MQ Console 和 REST API 的用户进行认证:

- 基本注册表
- LDAP 注册表
- 本地操作系统注册表
- z/OS 上的 SAF
- WebSphere Liberty 支持的任何其他注册表类型

可以将角色分配给 IBM MQ Console 用户和 REST API 用户，以确定授予他们对 IBM MQ 对象的访问权级别。 例如，要执行消息传递，必须为用户分配 MQWebUser 角色。 有关可用角色的更多信息，请参阅 第 457 页的『IBM MQ Console 和 REST API 上的角色』。

为用户分配角色后，可以使用许多方法来认证用户。 通过 IBM MQ Console，用户可以使用用户名和密码登录，也可以使用客户机证书认证。 通过 REST API，用户可以使用基本 HTTP 认证，基于令牌的认证或客户机证书认证。

过程

1. 定义用户注册表以认证用户，并为每个用户或组分配角色以授权用户和组使用 IBM MQ Console 或 REST API。 有关更多信息，请参阅 第 447 页的『配置用户和角色』
2. 选择向 mqweb 服务器认证 IBM MQ Console 用户的方式。 您不必对所有用户使用相同的方法:

- 使用令牌认证来认证用户。 在这种情况下，用户在 IBM MQ Console 登录屏幕上输入用户标识和密码。 这会生成一个 LTPA 令牌，可让用户在一段时间内保持登录和授权状态。 无需进一步配置即可使用此认证选项，但您可以选择配置 LTPA 令牌的到期时间。 有关更多信息，请参阅 配置 LTPA 令牌到期时间间隔。
- 使用客户机证书来认证用户。 在这种情况下，用户不会使用用户标识或密码来登录到 IBM MQ Console，而改为使用客户机证书。 有关更多信息，请参阅 第 460 页的『使用 REST API 和 IBM MQ Console 配置客户机证书认证』。

3. 选择向 mqweb 服务器认证 REST API 用户的方式。 您不必对所有用户使用相同的方法:
   - 使用 HTTP 基本认证来认证用户。 在此情况下，将对用户名和密码进行编码，但不会对其进行加密，并随每个 REST API 请求一起发送该用户名和密码，以针对该请求对用户进行认证和授权。 为保证此认证的安全，必须使用安全连接。 即，必须使用 HTTPS。 有关更多信息，请参阅 第 463 页的『将 HTTP 基本认证与 REST API 配合使用』。
   - 使用令牌认证来认证用户。 在这种情况下，用户使用 HTTP POST 方法向 REST API `login` 资源提供用户标识和密码。 这会生成一个 LTPA 令牌，可让用户在一段时间内保持登录和授权状态。 有关更多信息，请参阅第 464 页的『将基于令牌的认证与 REST API 配合使用』。

     为保证此认证的安全，必须使用安全连接。 即，必须使用 HTTPS。 但是，如果已启用 HTTP 连接，那么可以允许将针对 HTTPS 连接发出的 LTPA 令牌用于 HTTP 连接。 有关更多信息，请参阅 配置 LTPA 令牌。
   - 使用客户机证书来认证用户。 在这种情况下，用户不会使用用户标识或密码来登录到 REST API，而改为使用客户机证书。 有关更多信息，请参阅 第 460 页的『使用 REST API 和 IBM MQ Console 配置客户机证书认证』。
4. 可选： 为 REST API 配置跨源资源共享。

   缺省情况下，当脚本与 REST API 不是来自同一源时， Web 浏览器不允许脚本 (例如 JavaScript) 调用 REST API 。 即，未启用跨源请求。 您可以配置跨源资源共享 (CORS) 以允许来自指定 URL 的跨源请求。 有关更多信息，请参阅第 466 页的『为 REST API 配置 CORS』。
5. 可选： 配置 IBM MQ Console 和 REST API 的主机头验证。

   您可以配置主机头验证并创建主机名和端口的允许列表，以确保 IBM MQ Console 和 REST API 仅处理包含特定主机头的请求。 有关更多信息，请参阅第 467 页的『为 IBM MQ Console 和 REST API 配置主机头验证』。

配置用户和角色

要使用 IBM MQ Console 或 REST API，用户需要向定义到 mqweb 服务器的用户注册表进行认证。

关于此任务

已认证的用户需要是其中一个组的成员，这些组授权访问 IBM MQ Console 和 REST API 的功能。 缺省情况下，用户注册表不包含任何用户; 需要通过编辑 `mqwebuser.xml` 文件来添加这些用户。

配置用户和组时，首先配置用户注册表以向其认证用户和组。 此用户注册表在 IBM MQ Console 和 REST API 之间共享。 当您为用户和组配置角色时，可以控制用户和组是否有权访问 IBM MQ Console 和/或 REST API。

配置用户注册表后，为用户和组配置角色以授予其权限。 有多个角色可用，包括特定于使用 REST API for Managed File Transfer 的角色。 每个角色都授予不同级别的访问权。 有关更多信息，请参阅第 457 页的『IBM MQ Console 和 REST API 上的角色』。

在 mqweb 服务器中提供了许多样本 XML 文件，以使用户和组的配置更简单。 熟悉在 WebSphere Liberty (WLP) 中配置安全性的用户可能不希望使用这些样本。 除了此处记录的授权功能外， WLP 还提供其他授权功能。

过程

- 使用 `basic_registry.xml` 文件配置具有基本注册表的用户和组。

  注册表中的用户名和密码用于认证和授权 IBM MQ Console 和 REST API 的用户。

要使用 basic_registry.xml 样本文件配置基本注册表，请参阅 第 448 页的『为 IBM MQ Console 和 REST API 配置基本注册表』。

- 使用 ldap_registry.xml 文件配置具有 LDAP 注册表的用户和组。

  LDAP 注册表中的用户名和密码用于认证和授权使用 IBM MQ Console 和 REST API。

  要使用 ldap_registry.xml 样本文件配置 LDAP 注册表，请参阅 第 452 页的『为 IBM MQ Console 和 REST API 配置 LDAP 注册表』。

- ALW

  使用 local_os_registry.xml 文件配置具有本地操作系统注册表的用户和组。

  操作系统注册表中的用户名和密码用于认证和授权 IBM MQ Console 和 REST API 的用户。

  要使用 local_os_registry.xml 样本文件配置本地操作系统注册表，请参阅 第 451 页的『为 IBM MQ Console 和 REST API 配置本地操作系统注册表』。

- z/OS

  使用 zos_saf_registry.xml 文件在 z/OS 上配置具有系统授权工具 (SAF) 接口的用户和组。

  RACF 或其他安全产品概要文件用于授予用户和组对角色的访问权。 RACF 数据库中的用户名和密码用于认证和授权 IBM MQ Console 和 REST API 的用户。

  要使用 zos_saf_registry.xml 样本文件配置 SAF 接口，请参阅 第 454 页的『Configuring a SAF registry for the IBM MQ Console and REST API』。

- 使用 no_security.xml 文件禁用安全性，包括使用 HTTPS 访问 IBM MQ Console 或 REST API 的能力。

下一步做什么

选择用户认证方式:

IBM MQ Console 认证选项

- 使用令牌认证来认证用户。 在这种情况下，用户在 IBM MQ Console 登录屏幕上输入用户标识和密码。 这会生成一个 LTPA 令牌，可让用户在一段时间内保持登录和授权状态。 无需进一步配置即可使用此认证选项，但您可以选择为 LTPA 令牌配置到期时间间隔。 有关更多信息，请参阅 配置 LTPA 令牌到期时间间隔。
- 使用客户机证书来认证用户。 在这种情况下，用户不会使用用户标识或密码来登录到 IBM MQ Console，而改为使用客户机证书。 有关更多信息，请参阅 第 460 页的『使用 REST API 和 IBM MQ Console 配置客户机证书认证』。

REST API 认证选项

- 使用 HTTP 基本认证来认证用户。 在此情况下，将对用户名和密码进行编码，但不会对其进行加密，并随每个 REST API 请求一起发送该用户名和密码，以针对该请求对用户进行认证和授权。 为保证此认证的安全，必须使用安全连接。 即，必须使用 HTTPS。 有关更多信息，请参阅 第 463 页的『将 HTTP 基本认证与 REST API 配合使用』。
- 使用令牌认证来认证用户。 在这种情况下，用户使用 HTTP POST 方法向 REST API login 资源提供用户标识和密码。 这会生成一个 LTPA 令牌，可让用户在一段时间内保持登录和授权状态。 有关更多信息，请参阅第 464 页的『将基于令牌的认证与 REST API 配合使用』。 您可以配置 LTPA 令牌的到期时间间隔。 有关更多信息，请参阅 配置 LTPA 令牌。
- 使用客户机证书来认证用户。 在这种情况下，用户不会使用用户标识或密码来登录到 REST API，而改为使用客户机证书。 有关更多信息，请参阅 第 460 页的『使用 REST API 和 IBM MQ Console 配置客户机证书认证』。

为 IBM MQ Console 和 REST API 配置基本注册表

您可以在 mqwebuser.xml 文件中配置基本注册表。 XML 文件中的用户名，密码和角色用于认证和授权 IBM MQ Console 和 REST API 的用户。

开始之前

- 在基本注册表中配置用户时，必须为每个用户分配一个角色。 每个角色都提供不同级别的特权来访问 IBM MQ Console 和 REST API，并确定尝试执行允许的操作时使用的安全上下文。 在配置基本注册表之前，您需要了解这些角色。 有关每个角色的更多信息，请参阅 第 457 页的『IBM MQ Console 和 REST API 上的角色』。
- 要完成此任务，您必须是具有足够特权来编辑 `mqwebuser.xml` 文件的用户：
  - z/OS 在 z/OS 上，您必须具有对 `mqwebuser.xml` 文件的写访问权。
  - Multi 在所有其他操作系统上，您必须是特权用户。
  - Linux V 9.4.0 如果 mqweb 服务器是独立 IBM MQ Web Server 安装的一部分，那么您必须对 IBM MQ Web Server 数据目录中的 `mqwebuser.xml` 文件具有写访问权。

过程

1. 从以下某个路径复制样本 XML 文件 `basic_registry.xml` ：
   - 在 IBM MQ 安装中：
     - ALW 在 AIX, Linux, and Windows 上: *MQ_INSTALLATION_PATH* `/web/mq/samp/configuration`
     - z/OS 在 z/OS 上: `PathPrefix /web/mq/samp/configuration`

       其中，`PathPrefix` 是 IBM MQ for z/OS UNIX System Services Components 安装路径。
   - Linux V 9.4.0 在独立 IBM MQ Web Server 安装中: *MQWEB_INSTALLATION_PATH*`/web/mq/samp/configuration`

     其中 *MQWEB_INSTALLATION_PATH* 是将 IBM MQ Web Server 安装文件解压缩到的目录。
2. 将样本文件放在相应的目录中：
   - 在 IBM MQ 安装中：
     - Linux AIX 在 AIX 或 Linux 上: `/var/mqm/web/installations/`*installationName*`/servers/mqweb`
     - Windows 在 Windows: *MQ_DATA_PATH*`\web\installations\`*installationName*`\servers\mqweb` 上，其中 *MQ_DATA_PATH* 是 IBM MQ 数据路径。 此路径是在 IBM MQ 安装期间选择的数据路径。 缺省情况下，此路径为 `C:\ProgramData\IBM\MQ`。
     - z/OS 在 z/OS 上：*WLP_user_directory*`/servers/mqweb`

       其中 *WLP_user_directory* 是运行 **crtmqweb** 脚本以创建 mqweb 服务器定义时指定的目录。
   - Linux V 9.4.0 在独立 IBM MQ Web Server 安装中: *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb`

     其中 *MQ_OVERRIDE_DATA_PATH* 是 **MQ_OVERRIDE_DATA_PATH** 环境变量指向的 IBM MQ Web Server 数据目录。
3. 可选： 如果在 `mqwebuser.xml` 中更改了任何配置设置，请将其复制到样本文件中。
4. 删除现有 `mqwebuser.xml` 文件，并将样本文件重命名为 `mqwebuser.xml`。
5. 编辑新的 `mqwebuser.xml` 文件以在 **basicRegistry** 标记中添加用户和组。

   请注意，具有 `MQWebUser` 角色的任何用户都只能执行授予用户标识在队列管理器上执行的操作。 因此，注册表中定义的用户标识在安装了 IBM MQ 的系统上必须具有相同的用户标识。 这些用户标识必须相同，否则用户标识之间的映射可能会失败。

有关配置基本用户注册表的更多信息，请参阅 WebSphere Liberty 文档中的 为 Liberty 配置基本用户注册表 。

6. 通过编辑 `mqwebuser.xml` 文件将角色分配给用户和组:

   有多个角色可用于授权用户和组使用 IBM MQ Console 和 REST API。 每个角色都授予不同级别的访问权。 有关更多信息，请参阅第 457 页的『IBM MQ Console 和 REST API 上的角色』。

   - 要分配角色并授予对 IBM MQ Console 的访问权，请在 **<enterpriseApplication id="com.ibm.mq.console">** 标记中的相应 **security-role** 标记之间添加用户和组。
   - 要分配角色并授予对 REST API 的访问权，请在 **<enterpriseApplication id="com.ibm.mq.rest">** 标记中的相应 **security-role** 标记之间添加用户和组。

   有关 **security-role** 标记中用户和组信息的格式的帮助，请参阅 示例。

7. 如果在 `mqwebuser.xml` 中为用户提供了密码，那么应使用 WebSphere Liberty 提供的 **securityUtility encoding** 命令对这些密码进行编码，以使其更安全。 有关更多信息，请参阅 WebSphere Liberty 产品文档中的 Liberty:securityUtility 命令 。

示例

在以下示例中，将向组 `MQWebAdminGroup` 授予对具有角色 `MQWebAdmin` 的 IBM MQ Console 的访问权。 用户 `reader` 被授予具有角色 `MQWebAdminRO` 的访问权，用户 `guest` 被授予具有角色 `MQWebUser` 的访问权:

```
<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>
```

在以下示例中，将授予用户 `reader` 和 `guest` 对 IBM MQ Console 的访问权。 向用户 `user` 授予对 REST API 的访问权，并向 `MQAdmin` 组中的任何用户授予对 IBM MQ Console 和 REST API 的访问权。 `mftadmin` 用户被授予对 MFT 的 REST API 的访问权:

```
<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>
```

下一步做什么

选择用户认证方式:

IBM MQ Console 认证选项

- 使用令牌认证来认证用户。 在这种情况下，用户在 IBM MQ Console 登录屏幕上输入用户标识和密码。 这会生成一个 LTPA 令牌，可让用户在一段时间内保持登录和授权状态。 无需进一步配置即可使用此认证选项，但您可以选择为 LTPA 令牌配置到期时间间隔。 有关更多信息，请参阅 配置 LTPA 令牌到期时间间隔。
- 使用客户机证书来认证用户。 在这种情况下，用户不会使用用户标识或密码来登录到 IBM MQ Console，而改为使用客户机证书。 有关更多信息，请参阅 第 460 页的『使用 REST API 和 IBM MQ Console 配置客户机证书认证』。

REST API 认证选项

- 使用 HTTP 基本认证来认证用户。 在此情况下，将对用户名和密码进行编码，但不会对其进行加密，并随每个 REST API 请求一起发送该用户名和密码，以针对该请求对用户进行认证和授权。 为保证此认证的安全，必须使用安全连接。 即，必须使用 HTTPS。 有关更多信息，请参阅 第 463 页的『将 HTTP 基本认证与 REST API 配合使用』。
- 使用令牌认证来认证用户。 在这种情况下，用户使用 HTTP POST 方法向 REST API `login` 资源提供用户标识和密码。 这会生成一个 LTPA 令牌，可让用户在一段时间内保持登录和授权状态。 有关更多信息，请参阅第 464 页的『将基于令牌的认证与 REST API 配合使用』。 您可以配置 LTPA 令牌的到期时间间隔。 有关更多信息，请参阅 配置 LTPA 令牌。
- 使用客户机证书来认证用户。 在这种情况下，用户不会使用用户标识或密码来登录到 REST API，而改为使用客户机证书。 有关更多信息，请参阅 第 460 页的『使用 REST API 和 IBM MQ Console 配置客户机证书认证』。

ALW 为 IBM MQ Console 和 REST API 配置本地操作系统注册表

您可以在 `mqwebuser.xml` 文件中配置本地操作系统注册表。 本地操作系统上的用户名和密码用于认证和授权 IBM MQ Console 和 REST API 的用户。

开始之前

- 对于使用本地操作系统认证功能的客户机证书认证，用户身份是来自客户机证书的专有名称 (DN) 的公共名称 (CN)。 如果用户身份不作为操作系统用户存在，那么客户机证书登录将失败并回退到基于密码的认证。
- 要完成此任务，您必须是具有足够特权来编辑 `mqwebuser.xml` 文件的用户:
  - Linux V 9.4.0 如果 mqweb 服务器是独立 IBM MQ Web Server 安装的一部分，那么您必须对 IBM MQ Web Server 数据目录中的 `mqwebuser.xml` 文件具有写访问权。
  - 如果 mqweb 服务器是 IBM MQ 安装的一部分，那么您必须是 特权用户。

关于此任务

通过本地操作系统注册表，将自动为用户和组分配角色:

- 属于 "mqm" 组或 IBM i 上的 "QMQMADM" 组的任何用户都将被授予 `MQWebAdmin` 和 `MFTWebAdmin` 角色。
- 所有其他用户都被授予 `MQWebUser` 角色。

有关这些角色的更多信息，请参阅 第 457 页的『IBM MQ Console 和 REST API 上的角色』。

只能在 AIX, Linux, and Windows 上使用本地操作系统注册表。 z/OS 通过配置 SAF 注册表在 z/OS 上提供了等效功能。 有关更多信息，请参阅第 454 页的『Configuring a SAF registry for the IBM MQ Console and REST API』。

过程

1. 从以下某个路径复制样本 XML 文件 `local_os_registry.xml`：
   - Linux V 9.4.0 在独立 IBM MQ Web Server 安装中: *MQWEB_INSTALLATION_PATH*`/web/mq/samp/configuration`

     其中 *MQWEB_INSTALLATION_PATH* 是将 IBM MQ Web Server 安装文件解压缩到的目录。
   - 在 IBM MQ 安装中: *MQ_INSTALLATION_PATH*`/web/mq/samp/configuration`
2. 将样本文件放在下列其中一个目录中:
   - Linux V 9.4.0 在独立 IBM MQ Web Server 安装中: *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb`

     其中 *MQ_OVERRIDE_DATA_PATH* 是 **MQ_OVERRIDE_DATA_PATH** 环境变量指向的 IBM MQ Web Server 数据目录。
   - 在 IBM MQ 安装中: *MQ_DATA_PATH*`/web/installations/`*installationName*`/servers/mqweb`
3. 可选： 如果在 `mqwebuser.xml` 中更改了任何配置设置，请将其复制到样本文件中。
4. 删除现有 `mqwebuser.xml` 文件，并将样本文件重命名为 `mqwebuser.xml`。

下一步做什么

选择用户认证方式:

IBM MQ Console 认证选项

- 使用令牌认证来认证用户。 在这种情况下，用户在 IBM MQ Console 登录屏幕上输入用户标识和密码。 这会生成一个 LTPA 令牌，可让用户在一段时间内保持登录和授权状态。 无需进一步配置即可使用此认证选项，但您可以选择为 LTPA 令牌配置到期时间间隔。 有关更多信息，请参阅 配置 LTPA 令牌到期时间间隔。
- 使用客户机证书来认证用户。 在这种情况下，用户不会使用用户标识或密码来登录到 IBM MQ Console，而改为使用客户机证书。 有关更多信息，请参阅 第 460 页的『使用 REST API 和 IBM MQ Console 配置客户机证书认证』。

REST API 认证选项

- 使用 HTTP 基本认证来认证用户。 在此情况下，将对用户名和密码进行编码，但不会对其进行加密，并随每个 REST API 请求一起发送该用户名和密码，以针对该请求对用户进行认证和授权。 为保证此认证的安全，必须使用安全连接。 即，必须使用 HTTPS。 有关更多信息，请参阅 第 463 页的『将 HTTP 基本认证与 REST API 配合使用』。
- 使用令牌认证来认证用户。 在这种情况下，用户使用 HTTP POST 方法向 REST API `login` 资源提供用户标识和密码。 这会生成一个 LTPA 令牌，可让用户在一段时间内保持登录和授权状态。 有关更多信息，请参阅第 464 页的『将基于令牌的认证与 REST API 配合使用』。 您可以配置 LTPA 令牌的到期时间间隔。 有关更多信息，请参阅 配置 LTPA 令牌。
- 使用客户机证书来认证用户。 在这种情况下，用户不会使用用户标识或密码来登录到 REST API，而改为使用客户机证书。 有关更多信息，请参阅 第 460 页的『使用 REST API 和 IBM MQ Console 配置客户机证书认证』。

为 IBM MQ Console 和 REST API 配置 LDAP 注册表

您可以在 `mqwebuser.xml` 文件中配置 LDAP 注册表。 LDAP 注册表中的用户名和密码用于认证和授权 IBM MQ Console 和 REST API 的用户。

开始之前

- 配置 LDAP 注册表时，必须为每个用户分配一个角色。 每个角色都提供不同级别的特权来访问 IBM MQ Console 和 REST API，并确定尝试执行允许的操作时使用的安全上下文。 您需要先了解这些角色，然后

再配置注册表。 有关每个角色的更多信息，请参阅 第 457 页的『IBM MQ Console 和 REST API 上的角色』。

请注意，具有 MQWebUser 角色的任何用户都只能执行授予用户标识在队列管理器上执行的操作。 因此，LDAP 服务器上定义的用户标识在安装了 IBM MQ 的系统上必须具有相同的用户标识。 这些用户标识必须相同，否则用户标识之间的映射可能会失败。

- 要完成此任务，您必须是具有足够特权来编辑 mqwebuser.xml 文件的用户:
  - z/OS 在 z/OS 上，您必须具有对 mqwebuser.xml 文件的写访问权。
  - Multi 在所有其他操作系统上，您必须是特权用户。
  - Linux V 9.4.0 如果 mqweb 服务器是独立 IBM MQ Web Server 安装的一部分，那么您必须对 IBM MQ Web Server 数据目录中的 mqwebuser.xml 文件具有写访问权。

过程

1. 从以下某个路径复制样本 XML 文件 ldap_registry.xml :
   - 在 IBM MQ 安装中:
     - ALW 在 AIX, Linux, and Windows 上: *MQ_INSTALLATION_PATH* /web/mq/samp/configuration
     - z/OS 在 z/OS 上: PathPrefix /web/mq/samp/configuration

       其中，PathPrefix 是 IBM MQ for z/OS UNIX System Services Components 安装路径。
   - Linux V 9.4.0 在独立 IBM MQ Web Server 安装中: *MQWEB_INSTALLATION_PATH*/web/mq/samp/configuration

     其中 *MQWEB_INSTALLATION_PATH* 是将 IBM MQ Web Server 安装文件解压缩到的目录。
2. 将样本文件放在相应的目录中:
   - 在 IBM MQ 安装中:
     - Linux AIX 在 AIX 或 Linux 上: /var/mqm/web/installations/*installationName*/servers/mqweb
     - Windows 在 Windows: *MQ_DATA_PATH*\web\installations*installationName*\servers\mqweb 上，其中 *MQ_DATA_PATH* 是 IBM MQ 数据路径。 此路径是在 IBM MQ 安装期间选择的数据路径。 缺省情况下，此路径为 C:\ProgramData\IBM\MQ。
     - z/OS 在 z/OS 上： *WLP_user_directory*/servers/mqweb

       其中 *WLP_user_directory* 是运行 **crtmqweb** 脚本以创建 mqweb 服务器定义时指定的目录。
   - Linux V 9.4.0 在独立 IBM MQ Web Server 安装中: *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST/servers/mqweb

     其中 *MQ_OVERRIDE_DATA_PATH* 是 **MQ_OVERRIDE_DATA_PATH** 环境变量指向的 IBM MQ Web Server 数据目录。
3. 可选： 如果在 mqwebuser.xml 中更改了任何配置设置，请将其复制到样本文件中。
4. 删除现有 mqwebuser.xml 文件，并将样本文件重命名为 mqwebuser.xml。
5. 编辑新的 mqwebuser.xml 文件以更改 **ldapRegistry** 和 **idsLdapFilterProperties** 标记中的 LDAP 注册表设置。

   有关配置 LDAP 注册表的更多信息，请参阅 WebSphere Liberty 文档中的 在 Liberty 中配置 LDAP 用户注册表 。
6. 通过编辑 mqwebuser.xml 文件将角色分配给用户和组:

有多个角色可用于授权用户和组使用 IBM MQ Console 和 REST API。 每个角色都授予不同级别的访问权。 有关更多信息，请参阅第 457 页的『IBM MQ Console 和 REST API 上的角色』。

- 要分配角色并授予对 IBM MQ Console 的访问权，请在 **<enterpriseApplication id="com.ibm.mq.console">** 标记中的相应 **security-role** 标记之间添加用户和组。
- 要分配角色并授予对 REST API 的访问权，请在 **<enterpriseApplication id="com.ibm.mq.rest">** 标记中的相应 **security-role** 标记之间添加用户和组。

下一步做什么

选择用户认证方式:

IBM MQ Console 认证选项

- 使用令牌认证来认证用户。 在这种情况下，用户在 IBM MQ Console 登录屏幕上输入用户标识和密码。 这会生成一个 LTPA 令牌，可让用户在一段时间内保持登录和授权状态。 无需进一步配置即可使用此认证选项，但您可以选择为 LTPA 令牌配置到期时间间隔。 有关更多信息，请参阅 配置 LTPA 令牌到期时间间隔。
- 使用客户机证书来认证用户。 在这种情况下，用户不会使用用户标识或密码来登录到 IBM MQ Console，而改为使用客户机证书。 有关更多信息，请参阅 第 460 页的『使用 REST API 和 IBM MQ Console 配置客户机证书认证』。

REST API 认证选项

- 使用 HTTP 基本认证来认证用户。 在此情况下，将对用户名和密码进行编码，但不会对其进行加密，并随每个 REST API 请求一起发送该用户名和密码，以针对该请求对用户进行认证和授权。 为保证此认证的安全，必须使用安全连接。 即，必须使用 HTTPS。 有关更多信息，请参阅 第 463 页的『将 HTTP 基本认证与 REST API 配合使用』。
- 使用令牌认证来认证用户。 在这种情况下，用户使用 HTTP POST 方法向 REST API `login` 资源提供用户标识和密码。 这会生成一个 LTPA 令牌，可让用户在一段时间内保持登录和授权状态。 有关更多信息，请参阅第 464 页的『将基于令牌的认证与 REST API 配合使用』。 您可以配置 LTPA 令牌的到期时间间隔。 有关更多信息，请参阅 配置 LTPA 令牌。
- 使用客户机证书来认证用户。 在这种情况下，用户不会使用用户标识或密码来登录到 REST API，而改为使用客户机证书。 有关更多信息，请参阅 第 460 页的『使用 REST API 和 IBM MQ Console 配置客户机证书认证』。

z/OS Configuring a SAF registry for the IBM MQ Console and REST API

The System Authorization Facility (SAF) interface allows the mqweb server to call the external security manager for authentication and authorization checking. A user can then log in to the IBM MQ Console and REST API with a z/OS user ID and password.

Before you begin

- When you configure a SAF registry, you must assign users a role. Each role provides different levels of privilege to access the IBM MQ Console and REST API, and determines the security context that is used when an allowed operation is attempted. You need to understand these roles before you configure the registry. For more information about each of the roles, see “IBM MQ Console 和 REST API 上的角色” on page 457.
- 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 455.
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 455. 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 455. 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 455.

```
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 456. For more information about the roles, see "IBM MQ Console 和 REST API 上的角色" on page 457.

    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 455.

```
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

选择用户认证方式:

IBM MQ Console 认证选项

- 使用令牌认证来认证用户。 在这种情况下，用户在 IBM MQ Console 登录屏幕上输入用户标识和密码。 这会生成一个 LTPA 令牌，可让用户在一段时间内保持登录和授权状态。 无需进一步配置即可使用此认证选项，但您可以选择为 LTPA 令牌配置到期时间间隔。 有关更多信息，请参阅 配置 LTPA 令牌到期时间间隔。
- 使用客户机证书来认证用户。 在这种情况下，用户不会使用用户标识或密码来登录到 IBM MQ Console，而改为使用客户机证书。 有关更多信息，请参阅 "使用 REST API 和 IBM MQ Console 配置客户机证书认证" on page 460。

REST API 认证选项

- 使用 HTTP 基本认证来认证用户。 在此情况下，将对用户名和密码进行编码，但不会对其进行加密，并随每个 REST API 请求一起发送该用户名和密码，以针对该请求对用户进行认证和授权。 为保证此认证的安全，必须使用安全连接。 即，必须使用 HTTPS。 有关更多信息，请参阅 "将 HTTP 基本认证与 REST API 配合使用" on page 463。
- 使用令牌认证来认证用户。 在这种情况下，用户使用 HTTP POST 方法向 REST API `login` 资源提供用户标识和密码。 这会生成一个 LTPA 令牌，可让用户在一段时间内保持登录和授权状态。 有关更多信息，请参阅"将基于令牌的认证与 REST API 配合使用" on page 464。 您可以配置 LTPA 令牌的到期时间间隔。 有关更多信息，请参阅 配置 LTPA 令牌。
- 使用客户机证书来认证用户。 在这种情况下，用户不会使用用户标识或密码来登录到 REST API，而改为使用客户机证书。 有关更多信息，请参阅 "使用 REST API 和 IBM MQ Console 配置客户机证书认证" on page 460。

IBM MQ Console 和 REST API 上的角色

当您授权用户和组使用 IBM MQ Console 或 REST API 时，必须为用户和组分配下列其中一个可用角色: **MQWebAdmin**，**MQWebAdminRO**，**MQWebUser**，**MFTWebAdmin** 和 **MFTWebAdminRO**。 每个角色都提供不同级别的特权来访问 IBM MQ Console 和 REST API，并确定尝试执行允许的操作时使用的安全上下文。

注: 除了 **MQWebUser** 角色之外，用户标识不区分大小写。 请参阅 第 457 页的『MQWebUser』 以了解此角色的特定需求。

MQWebAdmin
分配有此角色的用户或组可以执行所有管理操作，并在用于启动 mqweb 服务器的操作系统用户标识的安全上下文下运行。

具有此角色的用户或组无权访问以下 REST 服务:

- MFT 的 REST API 。 要使用这些服务，还必须为用户或组分配 **MFTWebAdmin** 或 **MFTWebAdminRO** 角色。
- messaging REST API。 要使用 messaging REST API，必须为用户分配 **MQWebUser** 角色。

MQWebAdminRO
此角色授予对 IBM MQ Console 或 REST API 的只读访问权。 分配了此角色的用户或组可以执行以下操作:

- 显示和查询 IBM MQ 对象 (例如，队列和通道) 上的操作。
- 浏览队列上的消息。

分配有此角色的用户或组在用于启动 mqweb 服务器的操作系统用户标识的安全上下文下运行。

具有此角色的用户或组无权访问以下 REST 服务:

- MFT 的 REST API 。 要使用这些服务，还必须为用户或组分配 **MFTWebAdmin** 或 **MFTWebAdminRO** 角色。
- messaging REST API。 要使用 messaging REST API，必须为用户分配 **MQWebUser** 角色。

MQWebUser
分配有此角色的用户或组可以执行授予用户标识以在队列管理器上执行的任何操作。 例如:

- 对 IBM MQ 对象 (例如通道) 启动和停止操作。
- 定义并设置对 IBM MQ 对象 (例如，队列和通道) 的操作。
- 显示和查询 IBM MQ 对象 (例如，队列和通道) 上的操作。
- 使用 messaging REST API 放入和获取消息。

分配了此角色的用户或组在主体的安全上下文下运行，并且只能执行授予用户标识以在队列管理器上执行的操作。

因此，必须先在 IBM MQ 中授予 mqweb 用户注册表中定义的用户或组权限，然后该用户才能执行任何操作。 通过使用此角色，您可以在用户使用 IBM MQ Console 和 REST API 时，精细地控制哪些用户对特定 IBM MQ 资源具有哪种类型的访问权。

注:

- 分配此角色的用户标识的最大长度为 12 个字符。
- 用户标识的大小写在 mqweb 用户注册表和 IBM MQ 系统上必须相同。 如果用户标识的大小写不同，那么用户可能由 IBM MQ Console 和 REST API 进行认证，但无权使用 IBM MQ 资源。

MFTWebAdmin
分配了此角色的用户或组可以执行所有 MFT REST 操作，并在用于启动 mqweb 服务器的操作系统用户标识的安全上下文下运行。

具有此角色的用户或组无权访问任何 IBM MQ REST API 服务。 要使用这些服务，还必须为用户或组分配 **MQWebAdmin**，**MQWebAdminRO** 或 **MQWebUser** 角色。

MFTWebAdminRO
此角色授予对 MFT 的 REST API 的只读访问权。 分配有此角色的用户或组可以执行只读操作 (GET 请求)，例如列表传输和列表代理。

分配有此角色的用户或组在用于启动 mqweb 服务器的操作系统用户标识的安全上下文下运行。

具有此角色的用户或组无权访问任何 IBM MQ REST API 服务。 要使用这些服务，还必须为用户或组分配 **MQWebAdmin**， **MQWebAdminRO** 或 **MQWebUser** 角色。

有关配置用户和组以使用这些角色的更多信息，请参阅 第 447 页的『配置用户和角色』。

重叠角色

可以为用户或组分配多个角色。 当用户在此情况下执行操作时，将使用适用于该操作的最高特权角色。 例如，如果具有角色 **MQWebAdminRO** 和 **MQWebUser** 的用户执行查询队列操作，那么将使用 **MQWebAdminRO** 角色并在启动 Web 服务器的系统用户标识的上下文下尝试该操作。 如果同一用户执行定义操作，那么将使用 **MQWebUser** 角色，并且将在主体的上下文下尝试该操作。

ALW 将 IBM MQ Console 提供的证书更改为浏览器

您可以将 IBM MQ Console 配置为提供 CA 签名的证书以用于认证。 如果将 IBM MQ Console 配置为提供 CA 签名的证书，那么在访问 IBM MQ Console 时，浏览器将不再显示自签名证书警告。

关于此任务

IBM MQ Console 的安全性由运行 IBM MQ Console 的 mqweb 服务器提供。 要更改 mqweb 服务器向浏览器提供的证书，请首先将新证书添加到 mqweb 服务器密钥库。 然后编辑 `mqwebuser.xml` 文件中的安全配置以指定服务器提供的证书。

该过程做出以下假设:

- 您是 特权用户。
- 您正在使用 AIX， Linux 或 Windows 系统。
- `mqwebuser.xml` 文件基于 `basic_registry.xml`， `local_os_registry.xml` 或 `ldap_registry.xml` 样本 XML 文件。

过程

1. 可选： 使用 **runmqktool** 命令更改 mqweb 服务器密钥库 `key.jks` 的缺省密码:

```
runmqktool -storepasswd -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/key.jks -storepass oldPassword
          -new newPassword
```

oldPassword
指定现有 `key.jks` 密码。 缺省密码为 `password`。

newPassword
指定新的 `key.jks` 密码。

2. 创建要发送到认证中心的密钥对和证书请求:

a) 使用 **runmqktool** 命令创建密钥对:

```
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
```

密码
指定 `key.jks` 密钥库密码。

label
指定证书标签。 例如，`MQWebConsole`。

distinguished_name
指定证书的 X.500 专有名称。 将专有名称括在双引号中。

例如， `"cn=MQWebConsole,o=myOrg,c=UK"`

signature_algorithm
指定用于对证书进行签名的算法。 有关更多信息，请参阅 签名算法

b) 使用 **runmqktool** 命令创建证书请求:

```
runmqktool -certreq -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/key.jks -storepass password -alias label
          -file filename
```

密码
指定 key.jks 密钥库密码。

label
指定子步骤 第 458 页的『2.a』中的证书标签。

文件名
指定证书请求的标准文件名。

3. 将证书请求文件发送到认证中心 (CA)。
4. 当您具有来自 CA 的证书时，请使用 **runmqktool** 命令将证书以及证书链中的任何其他证书 (从根 CA 证书开始) 导入到 keys.jks 密钥库中:

```
runmqktool -importcert -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/key.jks -storepass password
          -alias label -file filename
```

密码
指定 key.jks 密钥库密码。

label
指定子步骤 第 458 页的『2.a』中的证书标签。

文件名
指定要导入的证书的标准文件名。

5. 配置 mqweb 服务器以提供 CA 证书:
   a) 打开 mqwebuser.xml 文件。

   可以在以下路径中找到 mqwebuser.xml 文件: *MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb

   b) 通过注释掉以下行来关闭缺省安全配置:

```
<sslDefault sslRef="mqDefaultSSLConfig"/>
```

   如果已将 mqweb 服务器配置为使用客户机证书认证，那么 XML 文件的此行已被注释掉。

   c) 取消注释 mqwebuser.xml 文件中用于启用定制证书配置的部分。 该部分包含以下文本:

```
<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"/>
```

   如果将 mqweb 服务器配置为使用客户机证书认证，那么 XML 文件的此部分已取消注释。

   d) 可选： 如果在步骤 第 458 页的『1』中更改了 key.jks 密钥库的密码，请将 defaultKeyStore 标记中 **password** 的值更改为您设置的密码的编码版本:

      i) 从 *MQ_INSTALLATION_PATH*/web/bin 目录中，输入以下命令:

```
securityUtility encode password
```

      ii) 将此命令的输出放在 defaultKeyStore 的 **password** 字段中。

   e) 如果未使用客户机证书认证，请注释掉以下行:

```
<keyStore id="defaultTrustStore" location="trust.jks" type="JKS" password="password"/>
```

f) 将 **serverKey 别名** 的值从 `default` 更改为 CA 证书标签的值。

6. 使用 **endmqweb** 命令停止 mqweb 服务器。
7. 使用 **strmqweb** 命令启动 mqweb 服务器。

结果

当 Web 服务器启动时，浏览到 IBM MQ Console 并刷新。 将使用 CA 证书，并且会将您直接转至登录页面。

ALW 使用 REST API 和 IBM MQ Console 配置客户机证书认证

您可以将客户机证书映射到主体以认证 IBM MQ Console 和 REST API 用户。

开始之前

- 配置要授权使用 IBM MQ Console 和 REST API 的用户，组和角色。 有关更多信息，请参阅第 447 页的『配置用户和角色』。
- 使用 REST API 时，可以通过在 `login` 资源上使用 HTTP GET 方法来查询当前用户的凭证，提供客户机证书以认证请求。 此请求返回有关用户名以及分配给用户的角色的信息。 有关更多信息，请参阅 GET /login。
- 将客户机证书映射到主体以认证用户时，将使用客户机证书的专有名称与配置的用户注册表中的用户进行匹配:
  - 对于基本注册表，公共名称 (CN) 与用户匹配。 例如， `CN=Fred, O=IBM, C=GB` 与 `Fred` 的用户名匹配。
  - 对于 LDAP 注册表，缺省情况下会将完整专有名称与 LDAP 进行匹配。 您可以设置过滤器和映射以定制匹配。 有关更多信息，请参阅 WebSphere Liberty 文档中的 Liberty :LDAP 证书映射方式 。

关于此任务

当用户使用客户机证书进行认证时，将使用该证书来代替用户名和密码。 对于 REST API，将随每个 REST 请求一起提供客户机证书以认证用户。 对于 IBM MQ Console，当用户使用证书登录时，该用户将无法注销。

ALW 在 AIX， Linux 或 Windows 系统上，此过程采用以下信息:

- `mqwebuser.xml` 文件基于 `basic_registry.xml`， `local_os_registry.xml` 或 `ldap_registry.xml` 样本 XML 文件。
- 您是 特权用户。

z/OS 要在 z/OS 系统上使用 RACF 密钥环配置客户机证书认证，请遵循 第 470 页的『Configuring TLS for the REST API and IBM MQ Console on z/OS』中的过程。

注: 以下过程概述了将客户机证书与 IBM MQ Console 和 REST API 配合使用所需的步骤。 为方便开发者，这些步骤详细说明了如何创建和使用自签名证书。 但是，对于生产，请使用从认证中心获取的证书。

过程

1. 使用 **runmqktool** 命令创建证书:

```
runmqktool -genkeypair -keystore filename -storepass password -storetype PKCS12
           -alias label -dname distinguished_name
           -sigalg signature_algorithm
```

文件名
指定密钥库名称，例如 `user.p12`。 如果密钥库不存在，那么将在命令运行时创建该密钥库。

密码
指定密钥库密码。

label
指定证书标签。 例如，`user1`。

distinguished_name
指定证书的 X.500 专有名称。 将专有名称括在双引号中。

如果您正在使用基本用户注册表，请在 "专有名称" 的 "公共名称 (CN)" 部分中输入用户注册表中用户的名称。 例如，对于用户 `mqadmin`，请使用专有名称 `"CN=mqadmin"`。

如果您正在使用本地操作系统注册表，请在专有名称的公共名称 (CN) 部分中输入本地操作系统用户标识的名称。 例如，对于用户 `mqadmin`，请使用专有名称 `"CN=mqadmin"`。

如果您正在使用 LDAP 用户注册表，请输入与 LDAP 注册表中的专有名称相匹配的专有名称。

signature_algorithm
指定用于对证书进行签名的算法。 有关更多信息，请参阅 签名算法

2. 可选： 从认证中心 (CA) 获取证书。 或者，要使用自签名证书，请继续执行步骤 第 461 页的『3』。

   a) 要从认证中心获取证书，请使用 **runmqktool** 命令创建证书请求:

   ```
   runmqktool -certreq -keystore filename -storepass password -alias label
              -file filename
   ```

   文件名
   指定步骤 第 460 页的『1』中的密钥库名称。

   密码
   指定密钥库密码。

   label
   指定步骤 第 460 页的『1』中的证书标签。

   文件名
   指定证书请求的标准文件名。

   b) 将证书请求文件发送到认证中心 (CA)。

   c) 当您拥有来自 CA 的证书时，请使用 **runmqktool** 命令将该证书导入到密钥库中:

   ```
   runmqktool -importcert -keystore filename -storepass password
              -alias label -file filename
   ```

   文件名
   指定步骤 第 460 页的『1』中的密钥库名称。

   密码
   指定密钥库密码。

   label
   指定步骤 第 460 页的『1』中的证书标签。

   文件名
   指定 CA 证书的标准文件名。

3. 使用 **runmqktool** 命令抽取证书的公用部分:

   ```
   runmqktool -exportcert -keystore filename -storepass password
              -alias label -file filename -rfc
   ```

   文件名
   指定步骤 第 460 页的『1』中的密钥库名称。

   密码
   指定密钥库密码。

   label
   指定步骤 第 460 页的『1』中的证书标签。

文件名
指定抽取的证书的标准文件名。

4. 将证书的公共部分作为签署者证书导入到 mqweb 服务器信任密钥库中，以便服务器可以使用 **runmqktool** 命令来验证客户机证书:

```
runmqktool -importcert -keystore  MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/trust.jks -storepass password
           -alias label -file filename
```

密码
指定 trust.jks 密钥库密码。 您可以为现有 trust.jks 密钥库指定密码，也可以为新的 trust.jks 密钥库指定新密码。

label
指定步骤 第 460 页的『1』 中的证书标签。

文件名
指定抽取的证书的标准文件名。

5. 配置 mqweb 服务器以使用客户机证书认证:
   a) 打开 mqwebuser.xml 文件。

   可以在以下路径中找到 mqwebuser.xml 文件: *MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb

   b) 通过注释掉以下行来关闭缺省安全配置:

```
<sslDefault sslRef="mqDefaultSSLConfig"/>
```

   如果配置了 mqweb 服务器以向浏览器提供 CA 证书，那么该行已被注释掉。

   c) 取消注释 mqwebuser.xml 文件中用于启用客户机证书认证的部分。 该部分包含以下文本:

```
<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"/>
```

   如果配置了 mqweb 服务器以向浏览器提供 CA 证书，那么此部分已取消注释。 但是，您可能需要取消注释 **defaultTrustStore** 行。

   d) 更改 defaultTrustStore 的 **password** 值以与 trust.jks 密钥库的密码匹配:
      i) 从 *MQ_INSTALLATION_PATH*/web/bin 目录中，输入以下命令:

```
securityUtility encode password
```

      ii) 将此命令的输出放在 defaultTrustStore 的 **password** 字段中。

6. 使用 **endmqweb** 命令停止 mqweb 服务器。
7. 使用 **strmqweb** 命令启动 mqweb 服务器。
8. 使用客户机证书进行认证:

   - 要将客户机证书与 IBM MQ Console 配合使用，请将客户机证书安装到用于访问 IBM MQ Console 的 Web 浏览器中。
   - 要将客户机证书与 REST API 配合使用，请为每个 REST 请求提供客户机证书。 使用 HTTP POST , PATCH 或 DELETE 方法时，必须向客户机证书提供额外认证以防止跨站点请求伪造攻击。 即，额外的认证用于确认用于认证请求的凭证正由凭证的所有者使用。

     此额外认证由 ibm-mq-rest-csrf-token HTTP 头提供。 将 ibm-mq-csrf-token 头的值设置为任何值 (包括空白) , 然后提交请求。

示例

要点: 在此示例中，并非所有 cURL 实现都支持自签名证书，因此您必须使用执行此操作的 cURL 实现。

以下 cURL 示例显示如何使用客户机证书认证在队列管理器 QM1 上创建新队列 Q1。 此 cURL 命令的准确配置取决于构建 cURL 所使用的库。 此示例基于具有针对 OpenSSL 构建的 cURL 的 Windows 系统。

- 将 HTTP POST 方法与队列资源配合使用，使用客户机证书进行认证，并包含具有任意值的 ibm-mq-rest-csrf-token HTTP 头。 此值可以是任何值，包括空白。 --cert-type 标志指定证书是 PKCS#12 证书。 --cert 标志指定证书的位置，后跟冒号，然后指定证书的密码:

```
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\"}"
```

将 HTTP 基本认证与 REST API 配合使用

REST API 的用户可以通过在 HTTP 头中提供其用户标识和密码来进行认证。 要将此认证方法与 HTTP 方法 (例如 POST， PATCH 和 DELETE) 配合使用，还必须提供 ibm-mq-rest-csrf-token HTTP 头以及用户标识和密码。

开始之前

- 配置要授权使用 REST API 的用户，组和角色。 有关更多信息，请参阅 第 447 页的『配置用户和角色』。
- 确保已启用 HTTP 基本认证。 检查 mqwebuser.xml 文件中是否存在以下 XML，并且未将其注释掉。 此 XML 必须在 <featureManager> 标记中:

```
<feature>basicAuthenticationMQ-1.0</feature>
```

z/OS 在 z/OS 上，您必须是对 mqwebuser.xml 具有写访问权的用户才能编辑此文件。

Multi 在所有其他操作系统上，您必须是 特权用户 才能编辑 mqwebuser.xml 文件。

- 确保在发送 REST 请求时使用的是安全连接。 由于用户名和密码组合已编码但未加密，因此当您将 HTTP 基本认证与 REST API 配合使用时，必须使用安全连接 (HTTPS)。
- 您可以通过对 login 资源使用 HTTP GET 方法来查询当前用户的凭证，从而提供基本认证信息以认证请求。 此请求将返回有关用户名以及分配给用户的角色的信息。 有关更多信息，请参阅 GET /login。

过程

1. 将用户名与冒号和密码并置。 请注意，用户名区分大小写。
   例如，用户名 admin 和密码 admin 将成为以下字符串:

   ```
   admin:admin
   ```

2. 使用 base64 编码对此用户名和密码字符串进行编码。
3. 将此编码的用户名和密码包含在 HTTP Authorization: Basic 头中。
   例如，使用编码的用户名 admin 和密码 admin，将创建以下头:

   ```
   Authorization: Basic YWRtaW46YWRtaW4=
   ```

4. 使用 HTTP POST， PATCH 或 DELETE 方法时，必须提供额外的认证以及用户名和密码。
   此额外认证由 ibm-mq-rest-csrf-token HTTP 头提供。 请求中必须存在 ibm-mq-rest-csrf-token HTTP 头，但其值可以是任何值，包括空白。
5. 使用相应的头向 IBM MQ 提交 REST 请求。

示例

以下示例显示如何在 Windows 系统上使用基本认证在队列管理器 QM1 上创建新队列 Q1。 此示例使用 cURL:

- 将 HTTP POST 方法与队列资源配合使用，通过基本认证进行认证，并包含具有任意值的 ibm-mq-rest-csrf-token HTTP 头。 此值可以是任何值，包括空白:

```
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\"}"
```

将基于令牌的认证与 REST API 配合使用

REST API 的用户可以通过使用 HTTP POST 方法向 REST API login 资源提供用户标识和密码来进行认证。 将生成 LTPA 令牌，使用户能够对将来的请求进行认证。 此 LTPA 令牌具有前缀 LtpaToken2。 用户可以使用 HTTP DELETE 方法注销，也可以使用 HTTP GET 方法查询当前用户的登录信息。

开始之前

- 配置要授权使用 REST API 的用户，组和角色。 有关更多信息，请参阅 第 447 页的『配置用户和角色』。
- 缺省情况下，包含 LTPA 令牌的 cookie 的名称以 LtpaToken2 开头，并且包含可在 mqweb 服务器重新启动时更改的后缀。 此随机 cookie 名称允许多个 mqweb 服务器在同一系统上运行。 但是，如果您希望 cookie 名称保持一致值，那么可以使用 **setmqweb** 命令指定 cookie 的名称。 有关更多信息，请参阅 配置 LTPA 令牌。
- 缺省情况下， LTPA 令牌 cookie 将在 120 分钟后到期。 您可以使用 **setmqweb** 命令来配置 LTPA 令牌 cookie 的到期时间。 有关更多信息，请参阅 配置 LTPA 令牌。
- 确保在发送 REST 请求时使用的是安全连接。 在 login 资源上使用 HTTP POST 方法时，不会对随请求一起发送的用户名和密码组合进行加密。 因此，当您对 REST API 使用基于令牌的认证时，必须使用安全连接 (HTTPS)。 缺省情况下，不能将 HTTP 与 LTPA 令牌认证配合使用。 您可以通过将 **secureLTPA** 设置为 False，使 LTPA 令牌可供不安全的 HTTP 连接使用。 有关更多信息，请参阅 配置 LTPA 令牌。
- 您可以使用 login 资源上的 HTTP GET 方法来查询当前用户的凭证，并提供 LTPA 令牌以认证请求。 此请求将返回有关用户名以及分配给用户的角色的信息。 有关更多信息，请参阅 GET /login。

过程

1. 登录用户:
   a) 对 login 资源使用 HTTP POST 方法:

   ```
   https://host:port/ibmmq/rest/v1/login
   ```

   在 JSON 请求的主体中包含用户名和密码，格式如下:

   ```
   {
       "username" : name,
       "password" : password
   }
   ```

   b) 将从请求返回的 LTPA 令牌存储在本地 Cookie 存储中。 缺省情况下，此 LTPA 令牌具有前缀 LtpaToken2。
2. 使用存储的 LTPA 令牌将 REST 请求作为 cookie 对每个请求进行认证。
   对于使用 HTTP PUT， PATCH 或 DELETE 方法的请求，请包含 ibm-mq-rest-csrf-token 头。 此头的值可以是任何值，包括空白。
3. 注销用户:
   a) 对 login 资源使用 HTTP DELETE 方法:

   ```
   https://host:9443/ibmmq/rest/v1/login
   ```

您必须提供 LTPA 令牌作为 cookie 以认证请求，并包含 `ibm-mq-rest-csrf-token` 头。 此头的值可以是任何内容，包括空白

b) 处理从本地 Cookie 存储中删除 LTPA 令牌的指令。

注: 如果未处理该指令，并且 LTPA 令牌仍保留在本地 cookie 存储中，那么可以使用 LTPA 令牌来认证将来的 REST 请求。 即，当用户在会话结束后尝试使用 LTPA 令牌进行认证时，将创建一个使用现有令牌的新会话。

示例

以下 cURL 示例显示了如何在 Windows 系统上使用基于令牌的认证在队列管理器 QM1 上创建新队列 Q1:

- 登录并将前缀为 LtpaToken2 的 LTPA 令牌添加到本地 cookie 存储。 用户名和密码信息包含在 JSON 主体中。 -c 标志指定要在其中存储令牌的文件的位置:

```
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
```

- 创建队列。 将 HTTP POST 方法与队列资源配合使用，使用 LTPA 令牌进行认证。 将使用 -b 标志从 `cookiejar.txt` 文件中检索具有前缀 LtpaToken2 的 LTPA 令牌。 CSRF 保护由 `ibm-mq-rest-csrf-token` HTTP 头提供:

```
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\"}"
```

- 从本地 cookie 存储库注销并删除 LTPA 令牌。 LTPA 令牌是使用 -b 标志从 `cookiejar.txt` 文件中检索的。 CSRF 保护由 `ibm-mq-rest-csrf-token` HTTP 头提供。 `cookiejar.txt` 文件的位置由 -c 标志指定，以便从文件中删除 LTPA 令牌:

```
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
```

相关参考

POST /login
GET /login
删除 /login

将 IBM MQ Console 嵌入到 IFrame 中

The HTML <iframe> element can be used to embed one web page into another using an Inline Frame (IFrame). 出于安全原因，缺省情况下无法将 IBM MQ Console 嵌入到 IFrame 中。 但是，您可以使用 mqweb 服务器上的 **mqConsoleFrameAncestors** 配置属性来启用 IFrame。

关于此任务

mqweb 服务器维护可使用 IFrame 嵌入 IBM MQ Console 的 Web 页面的源的允许列表。 源是 URL 方案，域和端口的组合，例如 `https://example.com:1234`。

您可以使用 mqweb 服务器上的 **mqConsoleFrameAncestors** 配置属性来指定列表中的条目。

缺省情况下， **mqConsoleFrameAncestors** 为空白，这意味着无法将 IBM MQ Console 嵌入到 IFrame 中。

过程

通过输入以下命令，指定可以在 IFrame 中嵌入 IBM MQ Console 的 Web 页面的源列表:

```
setmqweb properties -k mqConsoleFrameAncestors -v allowedOrigins
```

其中 *allowedOrigins* 是以逗号分隔的源列表。 每个源应包括:

- 主机名或 IP 地址
- 可选 URL 方案
- 可选端口号

请注意，主机名可以通配符 (*) 开头，端口号也可以使用通配符 (*)。
示例源为:

```
https://example.com:1234
```

这允许从 `https://example.com:1234` 提供的任何 Web 页面将 IBM MQ Console 嵌入到 IFrame 中。

```
https://*.example.com:*
```

这允许具有以 `example.com` 结尾的主机名的任何 HTTPS Web 页面，并使用任何端口将 IBM MQ Console 嵌入到 IFrame 中。

示例

以下示例允许从 `https://site2.example.com:1234` 或 `https://site2.example.com:1235` 提供服务的 Web 页面将 IBM MQ Console 嵌入到 IFrame 中:

```
setmqweb properties -k mqConsoleFrameAncestors -v
https://site2.example.com:1234,https://site2.example.com:1235
```

为 REST API 配置 CORS

缺省情况下，当脚本与 REST API 不是来自同一源时， Web 浏览器不允许脚本 (例如 JavaScript) 调用 REST API 。 即，未启用跨源请求。 您可以配置跨源资源共享 (CORS) 以允许来自指定源的跨源请求。

关于此任务

您可以通过 Web 浏览器 (例如，通过脚本) 访问 REST API 。 由于这些请求来自与 REST API 不同的源，因此 Web 浏览器会拒绝该请求，因为它是跨源请求。 如果域，端口或方案不相同，那么源不同。

例如，如果您具有在 `http://localhost:1999/` 上托管的脚本，那么在 `https://localhost:9443/` 上托管的 Web 站点上发出 HTTP GET 时，将发出跨源请求。 此请求是跨源请求，因为端口号和方案 (HTTP) 不同。

您可以通过配置 CORS 并指定允许访问 REST API 的源来启用跨源请求。

有关 CORS 的更多信息，请参阅 https://www.w3.org/TR/cors/ 和 https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS。

过程

1. 通过输入以下命令来查看当前配置:

   ```
   dspmqweb properties -a
   ```

   `mqRestCorsAllowedOrigins` 条目指定允许的源。 `mqRestCorsMaxAgeInSeconds` 条目指定 Web 浏览器可以高速缓存任何 CORS 预检结果的时间 (以秒计)。
2. 通过输入以下命令指定允许访问 REST API 的源:

   ```
   setmqweb properties -k mqRestCorsAllowedOrigins -v allowedOrigins
   ```

   其中 *allowedOrigins* 指定要允许来自跨源请求的源。 可以使用用双引号 "*" 括起的星号来允许所有跨源请求。 您可以在以逗号分隔的列表中输入多个源，并用双引号括起。 要不允许跨源请求，请输入空引号作为 *allowedOrigins* 的值。
3. 通过输入以下命令，指定要允许 Web 浏览器高速缓存任何 CORS 预检结果的时间 (以秒计):

```
setmqweb properties -k mqRestCorsMaxAgeInSeconds -v time
```

示例

以下示例显示针对 http://localhost:9883， https://localhost:1999 和 https://localhost:9663 启用的跨源请求。 任何 CORS 飞行前检查的高速缓存结果的最长时间设置为 90 秒:

```
setmqweb properties -k mqRestCorsAllowedOrigins -v "http://localhost:9883,https://
localhost:1999,https://localhost:9663"
setmqweb properties -k mqRestCorsMaxAgeInSeconds -v 90
```

为 IBM MQ Console 和 REST API 配置主机头验证

您可以配置 mqweb 服务器以限制对 IBM MQ Console 和 REST API 的访问，以便仅处理使用与指定允许列表匹配的主机头发送的请求。 如果使用不在允许列表上的主机头值，那么将返回错误。

关于此任务

mqweb 服务器使用虚拟主机来定义可接受的主机头的允许列表。 有关虚拟主机的更多信息，请参阅 WebSphere Liberty 文档: https://www.ibm.com/docs/SSEQTP_liberty/com.ibm.websphere.wlp.doc/ae/cwlp_virtual_hosts.html

要完成此任务，您必须是具有足够特权来编辑 mqwebuser.xml 文件的用户:

- z/OS 在 z/OS 上，您必须具有对 mqwebuser.xml 文件的写访问权。
- Multi 在所有其他操作系统上，您必须是特权用户。
- Linux V 9.4.0 如果 mqweb 服务器是独立 IBM MQ Web Server 安装的一部分，那么您必须对 IBM MQ Web Server 数据目录中的 mqwebuser.xml 文件具有写访问权。

过程

1. 打开 mqwebuser.xml 文件。 此文件位于下列其中一个位置:
   - 在 IBM MQ 安装中:
     - Linux AIX 在 AIX 或 Linux 上: /var/mqm/web/installations/*installationName*/servers/mqweb
     - Windows 在 Windows: *MQ_DATA_PATH*\web\installations*installationName*\servers\mqweb 上，其中 *MQ_DATA_PATH* 是 IBM MQ 数据路径。 此路径是在 IBM MQ 安装期间选择的数据路径。 缺省情况下，此路径为 C:\ProgramData\IBM\MQ。
     - z/OS 在 z/OS 上： *WLP_user_directory*/servers/mqweb

       其中 *WLP_user_directory* 是运行 **crtmqweb** 命令以创建 mqweb 服务器定义时指定的目录。
   - Linux V 9.4.0 在独立 IBM MQ Web Server 安装中: *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST/servers/mqweb

     其中 *MQ_OVERRIDE_DATA_PATH* 是 **MQ_OVERRIDE_DATA_PATH** 环境变量指向的 IBM MQ Web Server 数据目录。
2. 在 mqwebuser.xml 文件中添加或取消注释以下代码:

```
<virtualHost allowFromEndpointRef="defaultHttpEndpoint" id="default_host">
        <hostAlias>localhost:9080</hostAlias>
</virtualHost>
```

3. 编辑 **<hostAlias>** 字段，插入要允许的主机名和端口组合。

   此组合可能是您在 mqweb 服务器配置中使用的主机名和端口名。 例如，如果使用 localhost:9443 的缺省配置，那么可能要在 **<hostAlias>** 字段中使用 localhost:9443 。

如有必要，您可以在 **<virtualHost>** 标记中添加多个 **<hostAlias>** 字段，以允许更多主机名和端口组合。 例如，允许使用 HTTP 端口的主机头以及使用 HTTPS 端口的主机头。

审计

可以通过启用队列管理器命令和配置事件来生成 IBM MQ Console 和 REST API 中执行的操作的审计记录，并且在 AIX, Linux, and Windows 时，会将重要状态更改记录在 mqweb 服务器的日志文件中。

重大状态更改

ALW

在 AIX, Linux, and Windows 上， IBM MQ Console 会将重要状态更改作为消息记录在 mqweb 服务器的日志中。 每条消息都指示请求操作的已认证主体名称。

创建，启动，结束或删除队列管理器等重要状态更改将以 [AUDIT] 日志记录级别记录在 mqweb 服务器 `messages.log` 和 `console.log` 文件中。 每个日志条目都指示请求操作的已认证主体名称。

可以在以下位置找到 `messages.log` 和 `console.log` 文件:

- 在 IBM MQ 安装中:
  - Linux AIX 在 AIX 或 Linux 上: `/var/mqm/web/installations/`*`installationName`*`/servers/mqweb/logs`
  - Windows 在 Windows: *MQ_DATA_PATH*`\web\installations\`*`installationName`*`\servers\mqweb\logs` 上，其中 *MQ_DATA_PATH* 是 IBM MQ 数据路径。 此路径是在 IBM MQ 安装期间选择的数据路径。 缺省情况下，此路径为 `C:\ProgramData\IBM\MQ`。
- Linux V 9.4.0 在独立 IBM MQ Web Server 安装中: *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb/logs`

  其中 *MQ_OVERRIDE_DATA_PATH* 是 **MQ_OVERRIDE_DATA_PATH** 环境变量指向的 IBM MQ Web Server 数据目录。

有关配置 mqweb 服务器日志记录级别的更多信息，请参阅 配置日志记录。

命令和配置事件

您可以选择在队列管理器上启用命令和配置事件，以提供有关大多数 IBM MQ Console 和 REST API 活动的信息。 例如，创建通道和查询队列会生成命令和配置事件。 有关启用命令和配置事件的更多信息，请参阅 控制配置，命令和记录器事件。

对于这些命令和配置事件消息， **MQIACF_EVENT_ORIGIN** 字段设置为 `MQEVO_REST` , **MQCACF_EVENT_APPL_IDENTITY** 字段报告已认证主体名称的前 32 个字符。 如果用户具有 `MQWebAdmin` 或 `MQWebAdminRO` 角色，那么 **MQCACF_EVENT_USER_ID** 字段将报告 mqweb 服务器用户标识，而不是发出该命令的主体的用户名。 但是，如果用户具有 `MQWebUser` 角色，那么 **MQCACF_EVENT_USER_ID** 会报告发出该命令的主体的用户名。

相关概念

第 419 页的『审计』

您可以使用事件消息来检查安全性入侵或尝试的入侵。 您还可以使用 IBM MQ Explorer 来检查系统的安全性。

z/OS z/OS 上的 IBM MQ Console 和 REST API 的安全注意事项

IBM MQ Console 和 REST API 具有用于控制用户是否可以发出，显示或变更命令的安全功能。 然后将这些命令传递到队列管理器，然后使用队列管理器安全性来控制是否允许用户向该特定队列管理器发出命令。

过程

1. 请确保 mqweb 服务器启动式任务用户标识具有发出特定 PCF 命令和访问特定队列的相应权限。 有关更多信息，请参阅第 469 页的『Authority required by the mqweb server started task user ID』。
2. 确保授予 MQWebUser 角色的任何用户都具有相应的权限。

   分配给 MQWebUser 角色的 IBM MQ Console 和 REST API 用户在主体的安全上下文下运行。 这些用户标识只能执行授予用户标识在队列管理器上执行的操作，并且需要授予其对与 mqweb 服务器地址空间相同的系统队列的访问权。

   必须向 mqweb 服务器启动式任务用户标识授予分配给 MQWebUser 角色的所有用户的备用用户访问权。

   有关为具有 MQWebUser 角色的用户授予相应权限的更多信息，请参阅 第 469 页的『访问使用 IBM MQ Console 或 REST API 所需的 IBM MQ 资源』。
3. 可选： 为 IBM MQ Console 和 REST API 配置 TLS。 有关更多信息，请参阅第 470 页的『Configuring TLS for the REST API and IBM MQ Console on z/OS』。

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 201, “System queue security” on page 180, and “Profiles for context security” on page 190.
- 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 454 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 171.

z/OS 访问使用 IBM MQ Console 或 REST API 所需的 IBM MQ 资源

在 IBM MQ Console 或 REST API 中由具有 MQWebUser 角色的用户执行的操作是在该用户的安全上下文下执行的。

关于此任务

有关 IBM MQ Console 和 REST API 中的角色的更多信息，请参阅 第 457 页的『IBM MQ Console 和 REST API 上的角色』 。

使用以下过程以 MQWebUser 角色授予用户对使用 IBM MQ Console 或 REST API 所需的队列管理器资源的访问权。

过程

1. 授予 mqweb server started task 用户标识对 MQWebUser 角色中每个用户标识的备用用户访问权。

   在用户将通过 IBM MQ Console 或 REST API 管理的每个队列管理器上执行此操作。

   可以使用以下样本 RACF 命令将 mqweb server started task 用户标识备用用户访问权授予 MQWebUser 角色中的用户:

   ```
   RDEFINE MQADMIN hlq.ALTERNATE.USER.userId UACC(NONE)
   PERMIT hlq.ALTERNATE.USER.userId CLASS(MQADMIN) ACCESS(UPDATE) ID(mqwebUserId)
   SETROPTS RACLIST(MQADMIN) REFRESH
   ```

   其中:

   hlq
   是概要文件前缀，可以是队列管理器名称，也可以是队列共享组名

   userId
   是 MQWebUser 角色中的用户

   mqwebUserId
   是 mqweb server started task 用户标识

   注: 如果使用混合大小写安全性，请使用 MXADMIN 类而不是 MQADMIN 类。
2. 授予 MQWebUser 角色中的每个用户对使用 IBM MQ Console 和 REST API 所必需的系统队列的访问权。

   要执行此操作，请同时对 SYSTEM.ADMIN.COMMAND.QUEUE 和 SYSTEM.REST.REPLY.QUEUE，根据是否正在使用混合大小写安全性，授予每个用户对 MQQUEUE 或 MXQUEUE 类的 UPDATE 访问权。

   您需要在用户将通过 REST API 管理的每个队列管理器 (包括通过 administrative REST API 网关管理的远程队列管理器) 上执行此操作。
3. 要允许具有 MQWebUser 角色的用户管理远程队列管理器，请授予用户对 MQQUEUE 或 MXQUEUE 类中的概要文件的 UPDATE 访问权，以保护用于向远程队列管理器发送命令的传输队列。 请注意，您需要授予用户对网关队列管理器的 UPDATE 访问权。

   在远程队列管理器上，授予同一用户访问权，以放入用于将命令响应消息发送回网关队列管理器的传输队列。
4. 授予 MQWebUser 角色中的用户对执行 IBM MQ Console 和 REST API 支持的操作所需的任何其他资源的访问权。

   访问权需要:

   - 在 REST API 中执行操作，在各个 REST API 资源 的 安全要求 部分中进行了描述
   - 第 201 页的『IBM MQ Console - required command security profiles』 中描述了 IBM MQ Console 的问题命令

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 在 AIX, Linux, and Windows 上管理密钥和证书

在 AIX, Linux, and Windows 上，使用 **runmqakm** 和 V 9.4.0 V 9.4.0 **runmqktool** 命令来管理密钥，证书和证书请求。

关于此任务

runmqakm 命令提供与 **gskitcapicmd** 的功能类似的功能。 V 9.4.0 V 9.4.0 **runmqktool** 命令提供类似于 Java **keytool** 证书管理实用程序的功能。 在使用 **runmqakm** 或 **runmqktool** 命令之前，请确保通过运行 **setmqenv** 命令正确配置了系统环境变量。

runmqktool 命令要求安装 IBM MQ JRE 组件。 如果未安装此组件，那么可以改为使用 **runmqakm** 命令。

如果需要以符合 FIPS 的方式管理 TLS 证书，请使用 **runmqakm** 命令。 这是因为 **runmqakm** 命令支持更强的加密。

过程

- 使用 **runmqakm** 和 **runmqktool** 命令来完成以下操作:
  - 创建 IBM MQ 支持的 CMS 和 PKCS #12 密钥存储库。
  - 创建证书请求。
  - 导出证书。
  - 导入个人证书和 CA 证书。
  - 管理自签名证书。
  - 创建，抽取和添加密钥。

相关信息

密钥工具

ALW AIX, Linux, and Windows 上的 runmqakm 和 runmqktool 命令

在 AIX, Linux, and Windows 系统上，使用 **runmqakm** (GSKCapiCmd) 或 **runmqktool** (keytool) 命令来管理密钥和证书。

注: V 9.4.0 V 9.4.0

从 IBM MQ 9.4.0 中，将除去 **runmqckm** 和 **strmqikm** 命令。 可以使用 **runmqktool** 命令代替 **runmqckm** 命令来管理 PKCS #12 和 JKS 密钥存储库。 没有 **strmqikm** GUI 的替代项。

runmqckm 和 **runmqktool** 命令具有以下重要差异:

- **runmqktool** 命令不支持隐藏文件来存储密钥存储库密码。 在运行 **runmqktool** 命令时，必须始终向该命令提供用于访问密钥存储库的密码 (作为该命令的参数)，或者作为对该命令发出的提示的响应。
- **runmqktool** 命令不支持 CMS 密钥存储库。 因此，要将证书从 JKS 导出到 CMS 密钥存储库，必须完成以下步骤:

  1. 使用 **runmqktool -importkeystore** 命令将证书从 JKS 密钥存储库复制到中间 PKCS #12 密钥存储库。 有关导出证书的更多信息，请参阅 第 482 页的『从 AIX, Linux, and Windows 上的密钥存储库导出个人证书』。
  2. 使用 **runmqakm -cert -import** 命令将证书从中间 PKCS #12 密钥存储库导入到 CMS 密钥存储库。 有关导入证书的更多信息，请参阅 第 484 页的『将个人证书导入到 AIX, Linux, and Windows 上的密钥存储库中』。

以下 IBM MQ 命令可用于管理密钥和证书:

runmqakm

- 提供类似于 **gskitcapicmd** 的函数。
- 支持 CMS 和 PKCS #12 密钥存储库。
- 支持创建隐藏文件以存储加密密钥存储库密码。
- 已认证为符合 FIPS 140-$tag1 标准，并且可以配置为使用 **-fips** 参数以符合 FIPS 的方式运行。

V 9.4.0 V 9.4.0 **runmqktool**

- 提供与 Java **keytool** 命令类似的功能。
- 支持 PKCS #12， JKS 和 JCEKS 密钥存储库。
- 要求安装 IBM MQ Java runtime environment (JRE) 组件。

如果需要以符合 FIPS 的方式管理证书，请使用 **runmqakm** 命令。

有关 **runmqakm** 命令的更多信息，请参阅 runmqakm。

V 9.4.0 V 9.4.0 有关 **runmqktool** 命令的更多信息，请参阅 runmqktool。

本部分中的主题包含如何使用这些命令来完成公共证书管理任务的示例。

ALW 在 AIX, Linux, and Windows 上创建自签名个人证书

遵循此过程在密钥存储库中创建自签名个人证书。

注: IBM MQ 不支持 SHA-3 或 SHA-5 算法。 您可以使用数字签名算法名称 SHA384WithRSA 和 SHA512WithRSA，因为这两种算法都是 SHA-2 系列的成员。

Deprecated 不推荐使用数字签名算法名称 SHA3WithRSA 和 SHA5WithRSA，因为它们分别是 SHA384WithRSA 和 SHA512WithRSA 的缩写形式。

您可以使用 **runmqakm** (GSKCapiCmd) 或 **runmqktool** (keytool) 命令来创建自签名证书。 如果需要以符合 FIPS 的方式管理 SSL 或 TLS 证书，请使用 **runmqakm** 命令。

有关可能要使用自签名证书的原因的更多信息，请参阅 使用自签名证书对两个队列管理器进行相互认证。

并非所有数字证书都可以与所有 CipherSpecs 配合使用。 确保创建与您使用的 CipherSpecs 兼容的证书。 IBM MQ 支持三种不同类型的 CipherSpec。 有关更多信息，请参阅 第 40 页的『椭圆曲线与 RSA CipherSpecs 的互操作性』。

要使用类型 1 CipherSpecs (名称以 ECDHE_ECDSA_开头的)，必须使用 **runmqakm** 命令来创建证书，并且必须指定椭圆曲线 ECDSA 签名算法参数。 例如，通过指定参数 **-sig_alg** EC_ecdsa_with_SHA384。

使用 runmqakm

发出以下命令以使用 **runmqakm** 命令创建自签名个人证书:

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

其中:

-db 文件名
指定密钥存储库的标准文件名。 密钥存储库必须已存在。

-pw *password*
指定密钥存储库的密码。

-label *label*
指定证书标签。 证书标签区分大小写。

IBM MQ 所使用的 TLS 证书的标签是 **CERTLABL** 属性的值 (如果设置了此属性)，或者带有队列管理器名称或附加的 IBM MQ MQI client 用户标识的缺省 ibmwebspheremq (全部为小写)。 有关更多信息，请参阅 第 23 页的『数字证书标签，了解需求』。

-dn 区分名称
指定用双引号括起的 X.500 专有名称。 专有名称中至少需要一个属性。 您可以提供多个 OU 和 DC 属性。

注: runmqakm 命令将邮政编码属性称为 POSTALCODE，而不是 PC。 使用 **runmqakm** 命令请求具有邮政编码的证书时，请始终在 **-dn** 参数中指定 POSTALCODE 。

-size 键大小
指定密钥大小。 该值可以是 512， 1024 或 2048。

-x509version 版本
要创建的 X.509 证书的版本。 值可以是 1， 2 或 3。 缺省值为 3。

-expire 天
证书的到期时间 (以天计)。 证书的缺省值为 365 天。

-无花果
指定以 FIPS 方式运行命令。 仅使用 FIPS IBM Crypto for C (ICC) 组件，此组件必须以 FIPS 方式成功初始化。 在 FIPS 方式下， ICC 组件使用已通过 FIPS 140-2 验证的算法。 如果 ICC 组件未以 FIPS 方式初始化，那么 **runmqakm** 命令将失败。

-sig_alg
指定创建证书时使用的散列算法。 此散列算法用于创建与证书关联的签名。 该值可以是 md5, MD5_WITH_RSA， MD5WithRSA， SHA_WITH_DSA， SHA_WITH_RSA， sha1， SHA1WithDSA, SHA1WithECDSA， SHA1WithRSA， sha224， SHA224_WITH_RSA， SHA224WithDSA, SHA224WithECDSA， SHA224WithRSA， sha256， SHA256_WITH_RSA， SHA256WithDSA, SHA256WithECDSA， SHA256WithRSA， SHA2WithRSA， sha384， SHA384_WITH_RSA, SHA384WithECDSA， SHA384WithRSA， sha512， SHA512_WITH_RSA， SHA512WithECDSA, SHA512WithRSA， SHAWithDSA， SHAWithRSA， EC_ecdsa_with_SHA1, EC_ecdsa_with_SHA224， EC_ecdsa_with_SHA256， EC_ecdsa_with_SHA384 或 EC_ecdsa_with_SHA512。

缺省值为 SHA1WithRSA。

有关这些参数和可指定的值的更多信息，请参阅 runmqakm -cert。

使用 runmqktool

V 9.4.0 V 9.4.0

发出以下命令以使用 **runmqktool** 命令创建自签名个人证书:

```
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
```

其中：

-keystore 文件名
指定密钥存储库的名称。 如果密钥存储库不存在，那么将创建该密钥存储库。

-storepass *password*
指定密钥存储库密码。

-storetype *store_type*
指定密钥存储库类型。

-alias 标签
指定证书标签。 证书标签将转换为小写。

-dname 区分名称
指定以双引号括起的证书的 X.500 专有名称。

-有效期 天
指定证书有效的天数。

-keyalg *key_algorithm*
指定用于创建密钥对的算法。

-keysize 键大小
指定密钥大小。

-sigalg *signature_algorithm*
指定用于对证书进行签名的算法。 有关可指定的签名算法的更多信息，请参阅 签名算法。

有关这些参数以及可指定的值的更多信息，请参阅 genkeypair。

ALW 在 AIX, Linux, and Windows 上请求个人证书

遵循此过程来创建个人证书的请求。

注: IBM MQ 不支持 SHA-3 或 SHA-5 算法。 您可以使用数字签名算法名称 SHA384WithRSA 和 SHA512WithRSA，因为这两种算法都是 SHA-2 系列的成员。

Deprecated 不推荐使用数字签名算法名称 SHA3WithRSA 和 SHA5WithRSA，因为它们分别是 SHA384WithRSA 和 SHA512WithRSA 的缩写形式。

您可以使用 **runmqakm** (GSKCapiCmd) 或 **runmqktool** (keytool) 命令来请求个人证书。 如果需要以符合 FIPS 的方式管理 SSL 或 TLS 证书，请使用 **runmqakm** 命令。

并非所有数字证书都可以与所有 CipherSpecs 配合使用。 确保创建与您使用的 CipherSpecs 兼容的证书。 IBM MQ 支持三种不同类型的 CipherSpec。 有关更多信息，请参阅 第 40 页的『椭圆曲线与 RSA CipherSpecs 的互操作性』。

要使用类型 1 CipherSpecs (名称以 ECDHE_ECDSA_开头的)，必须使用 **runmqakm** 命令来创建证书，并且必须指定椭圆曲线 ECDSA 签名算法参数。 例如，通过指定参数 **-sig_alg** EC_ecdsa_with_SHA384。

如果您正在使用加密硬件，请参阅 第 492 页的『请求 PKCS #11 硬件的个人证书』。

使用 runmqakm

发出以下命令以使用 **runmqakm** 命令创建证书请求:

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

其中：

-db 文件名
指定密钥存储库的标准文件名。 密钥存储库必须已存在。

-pw *password*
指定密钥存储库的密码。

-label *label*
指定证书标签。 证书标签区分大小写。

IBM MQ 所使用的 TLS 证书的标签是 **CERTLABL** 属性的值 (如果设置了此属性)，或者带有队列管理器名称或附加的 IBM MQ MQI client 用户标识的缺省 ibmwebspheremq (全部为小写)。 有关更多信息，请参阅 第 23 页的『数字证书标签，了解需求』。

-dn 区分名称
指定用双引号括起的 X.500 专有名称。 专有名称中至少需要一个属性。 您可以提供多个 OU 和 DC 属性。

注: runmqakm 命令将邮政编码属性称为 POSTALCODE，而不是 PC。 使用 **runmqakm** 命令请求具有邮政编码的证书时，请始终在 **-dn** 参数中指定 POSTALCODE 。

-size 键大小
指定密钥大小。 该值可以是 512， 1024 或 2048。

-file 文件名
指定证书请求的文件名。

-无花果
指定以 FIPS 方式运行命令。 在 FIPS 方式下， IBM Crypto for C (ICC) 组件使用经 FIPS 140-2 验证的算法。 如果 ICC 组件未以 FIPS 方式初始化，那么 **runmqakm** 命令将失败。

-sig_alg
指定创建证书请求时使用的散列算法。 此散列算法用于创建与证书请求关联的签名。 该值可以是 md5, MD5_WITH_RSA， MD5WithRSA， SHA_WITH_DSA， SHA_WITH_RSA， sha1， SHA1WithDSA, SHA1WithECDSA， SHA1WithRSA， sha224， SHA224_WITH_RSA， SHA224WithDSA, SHA224WithECDSA， SHA224WithRSA， sha256， SHA256_WITH_RSA， SHA256WithDSA, SHA256WithECDSA， SHA256WithRSA， SHA2WithRSA， sha384， SHA384_WITH_RSA, SHA384WithECDSA， SHA384WithRSA， sha512， SHA512_WITH_RSA， SHA512WithECDSA, SHA512WithRSA， SHAWithDSA， SHAWithRSA， EC_ecdsa_with_SHA1, EC_ecdsa_with_SHA224， EC_ecdsa_with_SHA256， EC_ecdsa_with_SHA384 或 EC_ecdsa_with_SHA512。

缺省值为 SHA1WithRSA。

有关这些参数以及可指定的值的更多信息，请参阅 runmqakm -certreq。

使用 runmqktool

V 9.4.0 V 9.4.0

必须先使用 **runmqktool -genkeypair** 命令生成密钥对，然后才能使用 **runmqktool** 命令创建证书请求。 有关 **runmqktool -genkeypair** 命令的更多信息，请参阅 第 474 页的『在 AIX, Linux, and Windows 上创建自签名个人证书』。

发出以下命令以使用 **runmqktool** 命令创建证书请求:

```
runmqktool -certreq -keystore filename -storepass password -alias label
           -file filename
```

其中:

-keystore 文件名
指定密钥存储库的名称。

-storepass *password*
指定密钥存储库密码。

-alias 标签
指定证书标签。 这是生成密钥对时指定的证书标签。 证书标签不区分大小写。

-file 文件名
指定证书请求的文件名。

有关这些参数和可指定的值的更多信息，请参阅 certreq。

后续操作

向 CA 提交证书请求。 当您从 CA 接收签名证书时，请将签名证书添加到密钥存储库中。 有关更多信息，请参阅第 478 页的『将个人证书接收到 AIX, Linux, and Windows 上的密钥存储库中』。

ALW 在 AIX, Linux, and Windows 上更新现有个人证书

个人证书具有到期日期，之后无法再使用该证书。 遵循此过程以在个人证书到期之前对其进行更新。

您可以使用 **runmqakm** (GSKCapiCmd) 命令来更新个人证书。

如果需要将更大的密钥大小用于个人证书，那么无法更新现有证书。 您必须通过执行 第 476 页的『在 AIX, Linux, and Windows 上请求个人证书』 中描述的步骤来替换现有密钥，以创建使用所需密钥大小的新证书请求。

使用 runmqakm

使用 **runmqakm** 命令发出以下命令以创建用于更新个人证书的证书请求:

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

其中:

-db 文件名
指定密钥存储库的标准文件名。

-pw *password*
指定密钥存储库的密码。

-label *label*
指定证书标签。 证书标签区分大小写。

-target 文件名
指定证书请求的文件名。

后续操作

向 CA 提交证书请求。 当您从 CA 接收签名证书时，请将签名证书添加到密钥存储库中。 有关更多信息，请参阅第 478 页的『将个人证书接收到 AIX, Linux, and Windows 上的密钥存储库中』。

ALW 将个人证书接收到 AIX, Linux, and Windows 上的密钥存储库中

使用此过程将个人证书接收到密钥存储库中。

认证中心 (CA) 向您发送新的个人证书后，将其添加到从中生成新证书请求的密钥存储库。 如果 CA 将证书作为电子邮件消息的一部分发送，请将证书复制到单独的文件中。

在将 CA 签名的个人证书添加到密钥存储库之前，请完成 第 481 页的『将 CA 证书或可信证书的公用部分添加到 AIX, Linux, and Windows 上的密钥存储库』 中的步骤以将 CA 证书添加到密钥存储库。

您可以使用 **runmqakm** (GSKCapiCmd) 或 **runmqktool** (keytool) 命令将个人证书接收到密钥存储库中。 如果需要以符合 FIPS 的方式管理 SSL 或 TLS 证书，请使用 **runmqakm** 命令。

如果您正在使用加密硬件，请参阅 第 493 页的『将个人证书接收到 PKCS #11 硬件中』。

使用 runmqakm

发出以下命令以使用 **runmqakm** 命令将个人证书添加到密钥存储库:

```
runmqakm -cert -receive -file filename -format format
         -db filename -pw password -fips
```

其中:

-file 文件名
指定个人证书的标准文件名。

-db 文件名
指定密钥存储库的标准文件名。 密钥存储库必须已存在，并且必须与创建证书请求的存储库相同。

-pw *password*
指定密钥存储库的密码。

-format 格式
指定证书的格式。 该值可以是 `ascii`（对于基本 64 位编码 ASCII）或 `binary`（对于二进制 DER 数据）。 缺省值为 `ascii`。

-无花果
指定以 FIPS 方式运行命令。 在 FIPS 方式下， IBM Crypto for C (ICC) 组件使用已通过 FIPS 140-2 验证的算法。 如果 ICC 组件未以 FIPS 方式初始化，那么 **runmqakm** 命令将失败。

有关这些参数和可指定的值的更多信息，请参阅 runmqakm -cert。

使用 runmqktool

V 9.4.0 V 9.4.0

发出以下命令以使用 **runmqktool** 命令将个人证书添加到密钥存储库:

```
runmqktool -importcert -keystore filename -storepass password
           -alias label -file filename
```

其中:

-keystore 文件名
指定密钥存储库的标准文件名。 密钥存储库必须已存在，并且必须与创建证书请求的存储库相同。

-storepass *password*
指定密钥存储库的密码。

-alias 标签
指定用于创建证书请求的证书的标签。 证书标签将转换为小写。

-file 文件名
指定个人证书的标准文件名。

有关这些参数和可指定的值的更多信息，请参阅 importcert。

后续操作

如果将证书添加到队列管理器的 TLS 密钥存储库，请发出 MQSC 命令 **REFRESH SECURITY TYPE(SSL)** 以刷新队列管理器的 TLS 密钥存储库高速缓存。

ALW 从 AIX, Linux, and Windows 上的密钥存储库中抽取 CA 证书

遵循此过程从密钥存储库中抽取认证中心 (CA) 证书。

您可以使用 **runmqakm** (GSKCapiCmd) 或 **runmqktool** (keytool) 命令从密钥存储库中抽取 CA 证书。 如果需要以符合 FIPS 的方式管理 SSL 或 TLS 证书，请使用 **runmqakm** 命令。

使用 runmqakm

发出以下命令以使用 **runmqakm** 命令抽取 CA 证书:

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

其中:

-db 文件名
指定密钥存储库的标准文件名。

-pw *password*
指定密钥存储库的密码。

-label *label*
指定 CA 证书的标签。 证书标签区分大小写。

-target 文件名
指定目标文件的标准文件名。

-format 格式
指定证书的格式。 该值可以是 `ascii`（对于基本 64 位编码 ASCII）或 `binary`（对于二进制 DER 数据）。 缺省值为 `ascii`。

-无花果
指定以 FIPS 方式运行命令。 在 FIPS 方式下， IBM Crypto for C (ICC) 组件使用已通过 FIPS 140-2 验证的算法。 如果 ICC 组件未以 FIPS 方式初始化，那么 **runmqakm** 命令将失败。

有关这些参数和可指定的值的更多信息，请参阅 runmqakm -cert。

使用 runmqktool

V 9.4.0 V 9.4.0

发出以下命令以使用 **runmqktool** 命令抽取 CA 证书:

```
runmqktool -exportcert -keystore filename -storepass filename -alias label
           -file filename -rfc
```

其中:

-keystore 文件名
指定密钥存储库的标准文件名。

-storepass *password*
指定密钥存储库的密码。

-alias 标签
指定 CA 证书的标签。 证书标签不区分大小写。

-file 文件名
指定目标文件的标准文件名。

-rfc
指定输出文件采用 Base64-encoded ASCII 格式，如因特网 RFC 1421 标准所定义。 如果未指定此选项，那么输出文件为二进制格式。

有关这些参数和可指定的值的更多信息，请参阅 exportcert。

ALW 从 AIX, Linux, and Windows 上的密钥存储库中抽取自签名证书的公用部分

遵循此过程从密钥存储库中抽取自签名证书的公共部分。

您可以使用 **runmqakm** (GSKCapiCmd) 或 **runmqktool** (keytool) 命令从密钥存储库中抽取证书的公用部分。 如果需要以符合 FIPS 的方式管理 SSL 或 TLS 证书，请使用 **runmqakm** 命令。

使用 runmqakm

发出以下命令以使用 **runmqakm** 命令抽取自签名证书的公共部分:

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

其中:

-db 文件名
指定密钥存储库的标准文件名。

-pw *password*
指定密钥存储库的密码。

-label *label*
指定 CA 证书的标签。 证书标签区分大小写。

-target 文件名
指定目标文件的标准文件名。

-format 格式
指定证书的格式。 该值可以是 `ascii`（对于基本 64 位编码 ASCII）或 `binary`（对于二进制 DER 数据）。 缺省值是 `ascii`。

-无花果
指定以 FIPS 方式运行命令。 在 FIPS 方式下， IBM Crypto for C (ICC) 组件使用已通过 FIPS 140-2 验证的算法。 如果 ICC 组件未以 FIPS 方式初始化，那么 **runmqakm** 命令将失败。

有关这些参数和可指定的值的更多信息，请参阅 runmqakm -cert。

使用 runmqktool

V 9.4.0 V 9.4.0

发出以下命令以使用 **runmqktool** 命令抽取自签名证书的公共部分:

```
runmqktool -exportcert -keystore filename -storepass filename -alias label
           -file filename -rfc
```

其中:

-keystore 文件名
指定密钥存储库的标准文件名。

-storepass *password*
指定密钥存储库的密码。

-alias 标签
指定 CA 证书的标签。 证书标签不区分大小写。

-file 文件名
指定目标文件的标准文件名。

-rfc
指定输出文件采用 Base64-encoded ASCII 格式，如因特网 RFC 1421 标准所定义。 如果未指定此选项，那么输出文件为二进制格式。

有关这些参数和可指定的值的更多信息，请参阅 exportcert。

ALW 将 CA 证书或可信证书的公用部分添加到 AIX, Linux, and Windows 上的密钥存储库

遵循此过程将 CA 证书或可信证书的公共部分添加到密钥存储库。

您可以使用 **runmqakm** (GSKCapiCmd) 或 **runmqktool** (keytool) 命令将 CA 证书或可信证书的公共部分添加到密钥存储库中。 如果需要以符合 FIPS 的方式管理 SSL 或 TLS 证书，请使用 **runmqakm** 命令。

如果您要添加的证书处于证书链中，那么还必须添加该链中位于该证书上方的所有证书。 必须严格按降序添加证书，从根证书开始，随后是该链中紧跟其下的 CA 证书，以此类推。

注:

- 确保证书采用 ASCII (UTF-8) 或二进制 (DER) 编码。
- 由于 IBM Java 8 **keytool** 命令中的限制，如果文件包含注释，那么 **runmqktool** 无法导入 Internet RFC 1421 定义的可打印编码格式 (也称为 Base64 编码) 的证书。 要导入可打印编码格式的证书，请从文件中除去所有注释。 该文件必须以以 "----- BEGIN" 开头的字符串开头，并以以 "----- END" 开头的字符串结尾。

使用 runmqakm

发出以下命令以使用 **runmqakm** 命令将可信证书添加到密钥存储库:

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

其中:

-db 文件名
指定密钥存储库的标准文件名。 密钥存储库必须已存在。

-pw *password*
指定密钥存储库的密码。

-label *label*
指定证书标签。 证书标签区分大小写。

-file 文件名
指定包含证书的文件的名称。

-format *ascii*
指定证书的格式。 该值可以是 ascii（对于基本 64 位编码 ASCII）或 binary（对于二进制 DER 数据）。 缺省值是 ascii。

-无花果
指定以 FIPS 方式运行命令。 在 FIPS 方式下， IBM Crypto for C (ICC) 组件使用已通过 FIPS 140-2 验证的算法。 如果 ICC 组件未以 FIPS 方式初始化，那么 **runmqakm** 命令将失败。

有关这些参数以及可指定的值的详细信息，请参阅运行 mqakm-cert 。

使用 runmqktool

V 9.4.0 V 9.4.0

发出以下命令以使用 **runmqktool** 命令将可信证书添加到密钥存储库:

```
runmqktool -importcert -keystore filename -storepass password
           -alias label -file filename
```

其中:

-keystore 文件名
指定密钥存储库的标准文件名。 如果密钥存储库不存在，那么将创建该密钥存储库。

-storepass *password*
指定密钥存储库的密码。

-alias 标签
指定证书标签。 证书标签将转换为小写。

-file 文件名
指定个人证书的标准文件名。

有关这些参数和可指定的值的更多信息，请参阅 importcert。

ALW 从 AIX, Linux, and Windows 上的密钥存储库导出个人证书

遵循此过程从密钥存储库导出个人证书。

导出证书会将证书及其关联的公用密钥和专用密钥复制到另一个密钥存储库中。

您可以使用 **runmqakm** (GSKCapiCmd) 或 **runmqktool** (keytool) 命令从密钥存储库导出证书。 如果需要以符合 FIPS 的方式管理 SSL 或 TLS 证书，请使用 **runmqakm** 命令。

使用 runmqakm

发出以下命令以使用 **runmqakm** 命令导出证书:

```
runmqakm -cert -export -db filename -pw password -label label
         -target filename -target_pw password -target_type type
         -encryption strength -fips
```

其中:

-db 文件名
指定包含证书的密钥存储库的标准文件名。

-pw *password*
指定包含证书的密钥存储库的密码。

-label *label*
指定要导出的证书的标签。 证书标签区分大小写。

-target 文件名
指定目标密钥存储库的标准文件名。 如果密钥存储库不存在，那么将创建该密钥存储库。

-target_pw 密码
指定目标密钥存储库的密码。

-target_type 类型
指定目标密钥存储库的类型。 该值可以是 cms 或 pkcs12。 缺省值为 cms。

-encryption 强度
指定证书导出命令中使用的加密强度。 该值可以是 strong 或 弱。 缺省值为 strong。

-无花果
指定以 FIPS 方式运行命令。 在 FIPS 方式下， IBM Crypto for C (ICC) 组件使用已通过 FIPS 140-2 验证的算法。 如果 ICC 组件未以 FIPS 方式初始化，那么 **runmqakm** 命令将失败。

有关这些参数以及可指定的值的详细信息，请参阅运行 mqakm-cert 。

使用 runmqktool

V 9.4.0 V 9.4.0

发出以下命令以使用 **runmqktool** 命令导出证书:

```
runmqktool -importkeystore -srckeystore filename -srcstorepass password
           -destkeystore filename -deststoretype type
           -deststorepass password -destkeypass password
           -srcalias label -destalias label
```

其中:

-srckeystore 文件名
指定包含证书的密钥存储库的标准文件名。

-srcstorepass *password*
指定包含证书的密钥存储库的密码。

-destkeystore 文件名
指定目标密钥存储库的标准文件名。 如果密钥存储库不存在，那么将创建该密钥存储库。

-deststorepass *password*
指定目标密钥存储库的密码。

-destkeypass *password*
指定用于保护目标密钥存储库中的密钥的密码。 如果未指定此参数，那么将使用用于保护源密钥存储库中的密钥的密码来保护密钥。

-deststoretype 类型
指定目标密钥存储库的类型。

-sr 哈里亚斯 标签
指定要导出的证书的标签。 证书标签不区分大小写。

-destalias *label*
指定目标密钥存储库中证书的标签。 如果未指定此参数，那么将向证书分配与源密钥存储库中相同的标签。

证书标签将转换为小写。

-file 文件名
指定目标文件的标准文件名。

有关这些参数和可指定的值的更多信息，请参阅 importkeystore。

ALW 将个人证书导入到 AIX, Linux, and Windows 上的密钥存储库中

遵循此过程以将个人证书导入到密钥存储库中。

导入证书会将证书及其关联的公用密钥和专用密钥从一个密钥存储库复制到另一个密钥存储库。

在将个人证书导入到密钥存储库之前，必须首先将发放 CA 证书的完整有效链添加到密钥存储库。 有关更多信息，请参阅第 481 页的『将 CA 证书或可信证书的公用部分添加到 AIX, Linux, and Windows 上的密钥存储库』。

您可以使用 **runmqakm** (GSKCapiCmd) 或 **runmqktool** (keytool) 命令将证书导入到密钥存储库。 如果需要以符合 FIPS 的方式管理 SSL 或 TLS 证书，请使用 **runmqakm** 命令。

使用 runmqakm

发出以下命令以使用 **runmqakm** 命令导入证书:

```
runmqakm -cert -import -file filename -pw password -type type
         -target filename -target_pw password -target_type type
         -label label -new_label label -fips
```

其中:

-file 文件名
指定包含证书的密钥存储库的标准文件名。

-pw *password*
指定包含证书的密钥存储库的密码。

-type 类型
指定包含证书的密钥存储库的类型。 该值可以是 cms 或 pkcs12。 缺省值为 cms。

-target 文件名
指定目标密钥存储库的标准文件名。 如果密钥存储库不存在，那么将创建该密钥存储库。

-target_pw 密码
指定目标密钥存储库的密码。

-target_type 类型
指定目标密钥存储库的类型。 该值可以是 cms 或 pkcs12。 缺省值为 cms。

-label *label*
指定要从源密钥存储库导入的证书的标签。 证书标签区分大小写。

-new_label 标签
指定分配给目标密钥存储库中证书的标签。 如果未指定此参数，那么将向证书分配与源密钥存储库中相同的标签。

-无花果
指定以 FIPS 方式运行命令。 在 FIPS 方式下， IBM Crypto for C (ICC) 组件使用已通过 FIPS 140-2 验证的算法。 如果 ICC 组件未以 FIPS 方式初始化，那么 **runmqakm** 命令将失败。

有关这些参数和可指定的值的更多信息，请参阅 runmqakm -cert。

使用 runmqktool

V 9.4.0 V 9.4.0

发出以下命令以使用 **runmqktool** 命令导入证书:

```
runmqktool -importkeystore -srckeystore filename -srcstorepass password
           -destkeystore filename -deststoretype type
           -deststorepass password -destkeypass password
           -srcalias label -destalias label
```

其中:

-srckeystore 文件名
指定包含证书的密钥存储库的标准文件名。

-srcstorepass *password*
指定包含证书的密钥存储库的密码。

-destkeystore 文件名
指定目标密钥存储库的标准文件名。 如果密钥存储库不存在，那么将创建该密钥存储库。

-deststorepass *password*
指定目标密钥存储库的密码。

-destkeypass *password*
指定用于保护目标密钥存储库中的密钥的密码。 如果未指定此参数，那么将使用用于保护源密钥存储库中的密钥的密码来保护密钥。

注: 对于 PKCS #12 密钥存储库，必须使用与目标密钥存储库相同的密码来保护密钥。

-deststoretype 类型
指定目标密钥存储库的类型。

-sr 哈里亚斯 标签
指定要在源密钥存储库中使用的证书的标签。 证书标签不区分大小写。

-destalias *label*
指定目标密钥存储库中证书的标签。 如果未指定此参数，那么将向证书分配与源密钥存储库中相同的标签。

证书标签将转换为小写。

-file 文件名
指定目标文件的标准文件名。

有关这些参数和可指定的值的更多信息，请参阅 importkeystore。

ALW 从 Microsoft .pfx 文件导入个人证书

遵循以下过程从 AIX, Linux, and Windows 上的 Microsoft .pfx 文件导入证书。

.pfx 文件可以包含与同一密钥相关的两个证书。 一个是包含公用密钥和专用密钥的个人证书或站点证书。 另一个是仅包含公用密钥的 CA (签署者) 证书。 这些证书不能共存于同一 CMS 密钥存储库中，因此只能导入其中一个证书。

证书标签仅附加到签署者证书。 个人证书由系统生成的唯一用户标识 (UUID) 标识。 遵循此过程从 .pfx 文件导入个人证书，并将个人证书标签设置为分配给 .pfx 文件中 CA 证书的标签。 应该已将发放 CA 证书添加到目标密钥数据库。

使用 runmqakm

发出以下命令以使用 **runmqakm** 命令从 .pfx 文件导入证书:

```
runmqakm -cert -import -file filename -pw password -type pkcs12
         -target filename -target_pw password -target_type type
         -label label -new_label label -fips -pfx
```

其中:

-file 文件名
指定 .pfx 文件的标准名称。

-pw *password*
指定 .pfx 文件的密码。

-type pkcs12
指定密钥存储库的类型。

-target 文件名
指定目标密钥存储库的标准文件名。 如果密钥存储库不存在，那么将创建该密钥存储库。

-target_pw 密码
指定目标密钥存储库的密码。

-target_type 类型
指定目标密钥存储库的类型。 该值可以是 cms 或 pkcs12。 缺省值为 cms。

-label *label*
指定要从源密钥存储库导入的证书的标签。 证书标签区分大小写。

-new_label 标签
指定分配给目标密钥存储库中证书的标签。 如果未指定此参数，那么将向证书分配与源密钥存储库中相同的标签。

-无花果
指定以 FIPS 方式运行命令。 在 FIPS 方式下， IBM Crypto for C (ICC) 组件使用已通过 FIPS 140-2 验证的算法。 如果 ICC 组件未以 FIPS 方式初始化，那么 **runmqakm** 命令将失败。

-pfx
指示源密钥存储库使用 PFX 格式。

有关这些参数和可指定的值的更多信息，请参阅 runmqakm -cert。

ALW 从 PKCS #7 文件导入个人证书

遵循此过程从 AIX, Linux, and Windows 上的 PKCS #7 文件导入证书。

使用 **runmqakm** 命令从 AIX, Linux, and Windows 上的 PKCS #7 文件导入证书。

添加 CA 证书或可信证书的公共部分

发出以下命令以从 PKCS #7 文件添加 CA 证书或可信证书的公共部分:

```
runmqakm -cert -add -db filename -pw password -type type
         -label label -file filename
```

其中:

-db 文件名
指定密钥存储库的标准名称。

-pw *password*
指定密钥存储库的密码。

-type 类型
指定密钥存储库的类型。

-label *label*
指定要添加的证书的标签。 证书标签区分大小写。

标签将分配给添加的第一个证书。 所有其他证书 (如果存在) 都使用其主题名称进行标记。

-file 文件名
指定 PKCS #7 文件的标准名称。

有关这些参数和可指定的值的更多信息，请参阅 runmqakm -cert。

导入个人证书

发出以下命令以从 PKCS #7 文件导入个人证书:

```
runmqakm -cert -import -file filename -pw password -type pkcs7
         -target filename -target_pw password -target_type type
         -label label -new_label label
```

其中：

-file 文件名
指定 PKCS #7 文件的标准名称。

-pw *password*
指定 PKCS #7 文件的密码。

-type pkcs7
指定 PKCS #7 文件的类型。

-target 文件名
指定目标密钥存储库的标准文件名。 如果密钥存储库不存在，那么将创建该密钥存储库。

-target_pw 密码
指定目标密钥存储库的密码。

-target_type 类型
指定目标密钥存储库的类型。 该值可以是 cms 或 pkcs12。 缺省值为 cms。

-label *label*
指定要从 PKCS #7 文件导入的证书的标签。 证书标签区分大小写。

-new_label 标签
指定分配给目标密钥存储库中证书的标签。 如果未指定此参数，那么将向证书分配与源密钥存储库中相同的标签。

有关这些参数和可指定的值的更多信息，请参阅 runmqakm -cert。

ALW 在 AIX, Linux, and Windows 上列出密钥存储库中的证书

使用此过程可列出密钥存储库中的证书。

您可以使用 **runmqakm** (GSKCapiCmd) 或 **runmqktool** (keytool) 命令来显示有关密钥存储库中的证书的信息。

使用 runmqakm

- 发出以下命令以使用 **runmqakm** 命令列出密钥存储库中证书的标签：

```
runmqakm -cert -list -db filename -pw password
```

- 发出以下命令以使用 **runmqakm** 命令列出密钥存储库中证书的详细信息：

```
runmqakm -cert -details -showOID -db filename -pw password
         -label label
```

其中：

-file 文件名
指定密钥存储库的标准文件名。

-pw *password*
指定密钥存储库的密码。

-label *label*
指定要列示的证书的标签。 证书标签区分大小写。

有关这些参数以及可指定的值的详细信息，请参阅运行 mqakm-cert 。

使用 runmqktool

V 9.4.0 V 9.4.0

- 发出以下命令以使用 **runmqktool** 命令列出密钥存储库中证书的标签:

```
runmqktool -list -keystore filename -storepass password
```

- 发出以下命令以使用 **runmqktool** 命令列出密钥存储库中证书的详细信息:

```
runmqktool -list -keystore filename -storepass password -alias label -v
```

其中:

-keystore 文件名
指定密钥存储库的标准文件名。

-storepass *password*
指定密钥存储库的密码。

-alias 标签
指定要列示的证书的标签。 证书标签不区分大小写。

-v
请求包含证书详细信息的详细输出。

有关这些参数和可指定的值的更多信息，请参阅 list。

ALW 在 AIX, Linux, and Windows 上从密钥存储库中删除证书

使用此过程可从密钥存储库中删除个人证书或 CA 证书。

您可以使用 **runmqakm** (GSKCapiCmd) 或 **runmqktool** (keytool) 命令从密钥存储库中删除证书。 如果需要以符合 FIPS 的方式管理 SSL 或 TLS 证书，请使用 **runmqakm** 命令。

使用 runmqakm

发出以下命令以使用 **runmqakm** 命令删除证书:

```
runmqakm -cert -delete -db filename -pw password -label label -fips
```

其中:

-file 文件名
指定密钥存储库的标准文件名。

-pw *password*
指定密钥存储库的密码。

-label *label*
指定要删除的证书的标签。 证书标签区分大小写。

-无花果
指定以 FIPS 方式运行命令。 在 FIPS 方式下， IBM Crypto for C (ICC) 组件使用已通过 FIPS 140-2 验证的算法。 如果 ICC 组件未以 FIPS 方式初始化，那么 **runmqakm** 命令将失败。

有关这些参数以及可指定的值的详细信息，请参阅运行 mqakm-cert 。

使用 runmqktool

V 9.4.0 V 9.4.0

发出以下命令以使用 **runmqktool** 命令删除证书:

```
runmqktool -delete -keystore filename -storepass password -alias label
```

其中:

-keystore 文件名
指定密钥存储库的标准文件名。

-storepass *password*
指定密钥存储库的密码。

-alias 标签
指定要删除的证书的标签。 证书标签不区分大小写。

有关这些参数以及可指定的值的更多信息，请参阅 delete。

ALW 在 AIX, Linux, and Windows 上转换密钥存储库

使用此过程可将密钥存储库转换为其他类型。

您可以使用 **runmqakm** (GSKCapiCmd) 或 **runmqktool** (keytool) 命令将密钥存储库密码转换为其他类型。

使用 runmqakm

发出以下命令以使用 **runmqakm** 命令转换密钥存储库:

```
runmqakm -keydb -convert -db filename -pw password
         -new_db filename -new_pw password
         -old_format type -new_format type
```

其中:

-file 文件名
指定密钥存储库的标准文件名。

-pw *password*
指定密钥存储库的密码。

-new_db 文件名
指定新密钥存储库的标准文件名。

-new_pw *password*
指定新密钥存储库的密码。

-old_format 类型
指定密钥存储库的当前类型。 可以指定以下值:

- pkcs12
- **厘米**

-new_format 类型
指定密钥存储库的新类型。 可以指定以下值:

- pkcs12
- **厘米**

有关这些参数以及可指定的值的详细信息，请参阅运行 mqakm-keydb 。

使用 runmqktool

V 9.4.0 V 9.4.0

发出以下命令以使用 **runmqktool** 命令转换密钥存储库:

```
runmqktool -importkeystore -srckeystore filename -destkeystore filename
           -srcstoretype type -deststoretype type
           -srcstorepass password -deststorepass password
```

其中:

-全部
指定还会更改使用与密钥存储库相同的密码保护的所有条目的密码。

-keystore 文件名
指定密钥存储库的标准文件名。

-destkeystore 文件名
指定新密钥存储库的标准文件名。

-srcstoretype 类型
指定密钥存储库类型。

-deststoretype 类型
指定新的密钥存储库类型。

-srcstorepass *password*
指定密钥存储库的密码。

-deststorepass *password*
指定新密钥存储库的密码。

有关这些参数和可指定的值的更多信息，请参阅 importkeystore。

ALW 在 AIX, Linux, and Windows 上更改密钥存储库密码

使用此过程来更改密钥存储库密码。

您可以使用 **runmqakm** (GSKCapiCmd) 或 **runmqktool** (keytool) 命令来更改密钥存储库密码。

注:

- V 9.4.0 V 9.4.0 **runmqktool** 命令允许独立于保护单个专用密钥和密钥的密码来更改密钥存储库密码。 对于 PKCS #12 密钥存储库，密钥存储库密码和保护密钥存储库中所有密钥的密码必须相同。 如果使用 **runmqktool** 命令来更改密钥存储库密码，请确保指定了 **-all** 参数，以便同时更改密钥密码。
- 如果密钥存储库密码未存储在隐藏文件中，那么还必须更改存储在队列管理器配置或访问密钥存储库的任何 IBM MQ client 应用程序中的密码。 有关更多信息，请参阅第 265 页的『为 AIX, Linux, and Windows 上的队列管理器提供密钥存储库密码』和第 267 页的『为 AIX, Linux, and Windows 上的 IBM MQ MQI client 提供密钥存储库密码』。

使用 runmqakm

发出以下命令以使用 **runmqakm** 命令更改密钥存储库密码:

```
runmqakm -keydb -changepw -db filename -pw password -new_pw password -stash
```

其中:

-file 文件名
指定密钥存储库的标准文件名。

-pw *password*
指定密钥存储库的当前密码。

-new_pw *password*
指定密钥存储库的新密码。

-stash
可选。 指定此选项以将新的密钥存储库密码存储在隐藏文件中。 如果改为使用 IBM MQ 密码保护系统对密码进行加密，那么无需将密码存储在隐藏文件中。

有关这些参数以及可指定的值的详细信息，请参阅运行 mqakm-keydb 。

使用 runmqktool

V 9.4.0 V 9.4.0

发出以下命令以使用 **runmqktool** 命令更改密钥存储库密码:

```
runmqktool -storepasswd -all -keystore filename -storepass password
           -new password
```

其中:

-全部
指定还会更改使用与密钥存储库相同的密码保护的所有条目的密码。

-keystore 文件名
指定密钥存储库的标准文件名。

-storepass *password*
指定密钥存储库的当前密码。

-new 密码
指定密钥存储库的新密码。

有关这些参数以及可指定的值的更多信息，请参阅 storepasswd。

ALW 管理密钥 AIX, Linux, and Windows

按照此过程管理密钥存储库中的密钥。

您可以使用以下方式管理密钥 **runmqakm**（GSKCapiCmd） 命令。 使用生成的密钥 **runmqktool**（keytool）命令不能与 IBM MQ。

创建密钥

发出以下命令来创建随机密钥 **runmqakm** 命令：

```
runmqakm -secretkey -create -db filename -pw password
         -label label -size key_size
```

其中：

-D b 文件名
指定密钥存储库的完全限定文件名。 密钥存储库必须已经存在。

-pw 密码
指定密钥存储库的密码。

-label *label*
指定附加到密钥的标签。

-尺寸密钥大小
指定密钥大小（以字节为单位）。

有关这些参数以及可指定的值的更多信息，请参阅 runmqakm -secretkey。

提取密钥

发出以下命令来提取密钥 **runmqakm** 命令：

```
runmqakm -secretkey -extract -db filename -pw password
         -label label -target filename -format format
```

其中：

-D b 文件名
指定密钥存储库的完全限定文件名。 密钥存储库必须已经存在。

-pw 密码
指定密钥存储库的密码。

-label *label*
指定要提取的密钥的标签。

-目标文件名
指定目标文件的完整限定文件名。

-格式格式
指定目标文件中密钥的格式。 该值可以是 `ascii` 为了 Base64-encodedASCII 或 `binary` 获取密钥的二进制副本。 缺省值是 `ascii`。

有关这些参数以及可指定的值的更多信息，请参阅 runmqakm -secretkey。

添加密钥

发出以下命令来提取密钥 **runmqakm** 命令：

```
runmqakm -secretkey -add -db filename -pw password
         -label label -file filename -format format
```

其中：

-D b 文件名
指定密钥存储库的完全限定文件名。 密钥存储库必须已经存在。

-pw 密码
指定密钥存储库的密码。

-label *label*
指定附加到密钥的标签。

-文件文件名
指定包含密钥的文件的名称。

-格式格式
指定密钥的格式。 该值可以是 ascii 为了 Base64-encodedASCII 或 binary 对于二进制数据。 缺省值是 ascii。

有关这些参数以及可指定的值的更多信息，请参阅 runmqakm -secretkey。

ALW 管理 PKCS #11 硬件上的证书

您可以在支持 PKCS #11 接口的加密硬件上管理数字证书。

您必须创建密钥存储库以准备 IBM MQ 环境，即使您不打算在其中存储任何证书，但会将所有证书存储在加密硬件上。 密钥存储库对于队列管理器在其 **SSLKEYR** 属性中进行引用是必需的，对于客户机应用程序在 MQSSLKEYR 环境变量中进行引用也是必需的。 如果要创建证书请求，那么此密钥存储库也是必需的。

使用 **runmqakm** (GSKCapiCmd) 命令创建密钥存储库。

发出以下命令以使用 **runmqakm** 命令创建密钥存储库：

```
runmqakm -keydb -create -db filename -pw password -type type -stash
```

其中：

-db 文件名
指定密钥存储库的标准文件名。

-pw *password*
指定密钥存储库的密码。

-type 类型
指定数据库的类型。 对于 IBM MQ 所使用的密钥存储库，该值必须为 cms 或 pkcs12 。

-stash
可选。 如果指定了此参数，那么加密的密钥存储库密码将保存到文件中。

ALW 请求 *PKCS #11* 硬件的个人证书

使用此过程为具有加密硬件的队列管理器或 IBM MQ MQI client 请求个人证书。

注: IBM MQ 不支持 SHA-3 或 SHA-5 算法。 您可以使用数字签名算法名称 SHA384WithRSA 和 SHA512WithRSA ，因为这两种算法都是 SHA-2 系列的成员。

Deprecated 不推荐使用数字签名算法名称 SHA3WithRSA 和 SHA5WithRSA ，因为它们分别是 SHA384WithRSA 和 SHA512WithRSA 的缩写形式。

在加密硬件中创建证书请求之前，请完成 第 492 页的『管理 PKCS #11 硬件上的证书』 中描述的步骤以创建密钥存储库。

发出以下命令以使用 **runmqakm** (GSKCapiCmd) 命令创建证书请求:

```
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
```

其中:

-crypto *module_name*
指定随加密硬件一起提供的 PKCS #11 库的标准名称。

-tokenlabel 硬件令牌
指定 PKCS #11 加密设备令牌标签。

-pw *password*
指定用于访问加密硬件的密码。

-label *label*
指定证书标签。

IBM MQ 所使用的 TLS 证书的标签是 **CERTLABL** 属性的值 (如果设置了此属性)，或者带有队列管理器名称或附加的 IBM MQ MQI client 用户标识的缺省 ibmwebspheremq (全部为小写)。 有关更多信息，请参阅 第 23 页的『数字证书标签，了解需求』。

-dn 区分名称
指定用双引号括起的 X.500 专有名称。 专有名称中至少需要一个属性。 您可以提供多个 OU 和 DC 属性。

注: runmqakm 命令将邮政编码属性称为 POSTALCODE，而不是 PC。 使用 **runmqakm** 命令请求具有邮政编码的证书时，请始终在 **-dn** 参数中指定 POSTALCODE 。

-size 键大小
指定密钥大小。 该值可以是 512， 1024 或 2048。

-file 文件名
指定证书请求的文件名。

-无花果
指定以 FIPS 方式运行命令。 在 FIPS 方式下， IBM Crypto for C (ICC) 组件使用经 FIPS 140-2 验证的算法。 如果 ICC 组件未以 FIPS 方式初始化，那么 **runmqakm** 命令将失败。

-sig_alg
指定创建证书请求时使用的散列算法。 此散列算法用于创建与证书请求关联的签名。 该值可以是 md5, MD5_WITH_RSA， MD5WithRSA， SHA_WITH_DSA， SHA_WITH_RSA， sha1， SHA1WithDSA, SHA1WithECDSA， SHA1WithRSA， sha224， SHA224_WITH_RSA， SHA224WithDSA, SHA224WithECDSA， SHA224WithRSA， sha256， SHA256_WITH_RSA， SHA256WithDSA, SHA256WithECDSA， SHA256WithRSA， SHA2WithRSA， sha384， SHA384_WITH_RSA, SHA384WithECDSA， SHA384WithRSA， sha512， SHA512_WITH_RSA， SHA512WithECDSA, SHA512WithRSA， SHAWithDSA， SHAWithRSA， EC_ecdsa_with_SHA1, EC_ecdsa_with_SHA224， EC_ecdsa_with_SHA256， EC_ecdsa_with_SHA384 或 EC_ecdsa_with_SHA512。

缺省值为 SHA1WithRSA。

有关这些参数以及可指定的值的更多信息，请参阅 runmqakm -certreq。

后续操作

向 CA 提交证书请求。 当您从 CA 接收签名证书时，请将签名证书添加到密钥存储库中。 有关更多信息，请参阅第 493 页的『将个人证书接收到 PKCS #11 硬件中』。

ALW 将个人证书接收到 *PKCS #11* 硬件中

使用此过程将队列管理器或 IBM MQ MQI client 的个人证书接收到加密硬件。

将签署个人证书的 CA 的 CA 证书添加到加密硬件或辅助密钥存储库中。 在将签名证书接收到加密硬件之前执行此操作。 要将 CA 证书添加到密钥存储库文件，请遵循 第 481 页的『将 CA 证书或可信证书的公用部分添加到 AIX, Linux, and Windows 上的密钥存储库』中的过程。

发出以下命令以使用 **runmqakm** (GSKCapiCmd) 命令将个人证书添加到密钥存储库:

```
runmqakm -cert -receive -file filename -crypto module_name
         -tokenlabel hardware_token -pw hardware_password
         -format cert_format -fips
         -secondaryDB filename -secondaryDBpw password
```

其中:

-file 文件名
指定包含个人证书的文件的标准文件名。

-crypto *module_name*
指定随加密硬件一起提供的 PKCS #11 库的标准名称。

-tokenlabel 硬件令牌
指定 PKCS #11 加密设备令牌标签。

-pw *hardware_password*
指定用于访问加密硬件的密码。

-format *cert_format*
指定证书的格式。 该值可以是 `ascii`（对于基本 64 位编码 ASCII）或 `binary`（对于二进制 DER 数据）。 缺省值为 ASCII。

-无花果
指定以 FIPS 方式运行命令。 在 FIPS 方式下， IBM Crypto for C (ICC) 组件使用经 FIPS 140-2 验证的算法。 如果 ICC 组件未以 FIPS 方式初始化，那么 **runmqakm** 命令将失败。

-secondaryDB 文件名
指定用于存储 CA 证书的密钥存储库文件的标准文件名。

-secondaryDBpw 密码
指定用于存储 CA 证书的密钥存储库文件的密码。

保护 IBM MQ 组件配置文件中的密码

要使用 IBM MQ 的某些功能部件，您可能需要提供该功能部件所使用的密码。 可以使用密码保护系统来保护提供给 IBM MQ 的 密码。

以下列表说明了用于处理加密密码的每个组件的术语:

初始密钥
用于保护密码的加密密钥。

缺省初始键
在加密密码时未提供初始密钥时使用的缺省加密密钥。

纯文本字符串
加密的字符串，通常是密码。

加密密码字符串
包含格式为 IBM MQ 的加密密码的字符串。

指定初始密钥

对于每个组件，您可以选择指定用于加密密码的初始密钥。

- 如果未指定初始密钥，那么将使用组件的缺省初始密钥。 所有 IBM MQ 安装的缺省初始密钥都相同。 这意味着使用缺省初始密钥加密的密码不受安全保护，因为其他安装可能会对密码进行解密。
- 如果您提供自己唯一的初始密钥，那么只有有权访问您提供的初始密钥的用户才能解密密码。

注意: 要为存储的密码提供最高级别的安全性，请为每个 IBM MQ 组件提供唯一的初始密钥。

如果选择使用您自己的初始密钥，请为列出的每个组件指定唯一的初始密钥。 初始密钥用于保护存储在该组件的配置中的任何密码。 对于要解密的密码，还必须向组件提供相同的初始密钥。

大多数组件都需要在文件中提供初始密钥。 初始密钥文件中包含的初始密钥必须满足以下需求:

- 它必须至少有一个字符长。
- 它必须是单行文本。

初始键的最大长度不受限制，可以指定任何字符。 为了获得足够的安全性，请指定长度至少为 16 个字符的初始密钥。 例如，初始密钥文件可能包含以下字符串:

```
Th1sIs@n3Ncypt|onK$y
```

对初始密钥文件的访问必须仅限于需要使用操作系统文件许可权来访问初始密钥的用户。

有关密码保护的优点和限制的更多信息，请参阅 第 500 页的『通过密码加密进行保护时存在的限制』。

保护每个 IBM MQ 组件中的密码

多个 IBM MQ 组件可以保护存储的密码。 根据组件，可以使用下列其中一种机制来提供这些密码:

- 直接提供给 IBM MQ 队列管理器或 IBM MQ client。
- 在环境变量中指定。
- 存储在配置文件中。

每个组件都提供了用于加密密码的方法。 在大多数组件中，必须先对密码进行加密，然后才能将其提供给 IBM MQ 或存储在配置中。

要点: 无法将为与一个组件配合使用而生成的加密密码复制到另一个组件的配置文件中。 必须使用同一组件提供的实用程序来保护加密以供特定组件使用的密码。

以下部分中列出了有关如何保护支持密码保护的每个 IBM MQ 组件的密码的详细信息:

- Advanced Message Security
- 第 496 页的『Managed File Transfer』
- 第 497 页的『IBM MQ Internet Pass-Thru』
- 第 497 页的『使用加密硬件的 IBM MQ clients』
- 第 498 页的『IBM MQ 队列管理器』
- 第 498 页的『IBM MQ C 客户机应用程序』
- V 9.4.0 第 499 页的『本机 HA 配置』
- V 9.4.0 第 499 页的『IBM MQ 队列管理器 (qm.ini 文件中的 AuthToken 节)』

Advanced Message Security

Advanced Message Security (AMS) Java 客户机需要访问包含用于保护消息的专用密钥的密钥库。

Advanced Message Security (AMS) 配置为执行 MCA 拦截的 MQI 客户机或队列管理器可能需要访问 PKCS#11 加密硬件或包含用于保护消息的专用密钥的 PEM 文件。

要访问这些密钥存储库，必须在名为 `keystore.conf` 的 AMS 配置文件中提供密码。 使用 **runamscred** 命令来保护 `keystore.conf` 文件中包含的敏感信息。 例如

```
runamscred -f <keystore configuration file>
```

runamscred 命令保护使用 **-f** 参数指定的文件中的敏感参数。

IBM MQ 安装中提供了两个 **runamscred** 命令:

- 位于 `<IBM MQ installation root>/bin` 中的 MQI **runamscred** 命令
- 位于 `<IBM MQ installation root>/java/bin` 中的 Java **runamscred** 命令

注意: 为了确保兼容性，

1. 使用 Java **runamscred** 命令来保护与 Java AMS 客户机配合使用的配置文件，使用 MQI **runamscred** 命令来保护使用 AMS 的 IBM MQ MQI clients 的配置文件。
2. 在运行 **runamscred** 命令之后，请验证所有必需的敏感信息是否受保护。
3. 向已启用 AMS 的应用程序提供包含正常受保护密码的文件。

缺省情况下， **runamscred** 命令使用缺省初始密钥对配置文件中的密码进行加密。 要使用特定初始密钥对密码进行加密，请使用下列其中一种机制按优先级顺序指定包含初始密钥的文件的名称:

1. **runamscred** 命令的 **-sf** 参数。
2. **MQS_AMSCRED_KEYFILE** 环境变量。
3. keystore.conf 配置文件中的 **amscred.keyfile** 参数。

警告: 所有 IBM MQ 安装的缺省初始密钥都相同。 要安全地保护密码，请在加密密码时提供对安装唯一的初始密钥。

如果在运行 **runamscred** 命令以加密 AMS 配置中的密码时指定初始密钥文件，那么还必须在 AMS 应用程序运行时指定相同的初始密钥文件。 可以使用以下机制按优先级顺序指定初始密钥文件的名称:

1. **MQS_AMSCRED_KEYFILE** 环境变量。
2. keystore.conf 配置文件中的 **amscred.keyfile** 参数。

缺省情况下， **runamscred** 命令使用与低于 IBM MQ 9.2 的 AMS 版本不兼容的保护系统来保护凭证。 要使用与 IBM MQ 9.2 之前的版本兼容的凭证保护系统来保护配置文件，请在运行 **runamscred** 命令时指定 **-sp 0** 参数。

Managed File Transfer

Managed File Transfer (MFT) 将访问队列管理器和其他资源所需的凭证存储在以下 XML 属性文件中:

MQMFTCredentials.xml
此文件包含以下凭证:

- 用于连接到代理，协调和命令队列管理器的凭证。
- 用于访问用于安全通信的密钥库的密码。

ProtocolBridgeCredentials.xml
此文件包含用于连接到协议服务器 (例如 FTP， SFTP 和 FTPS) 的凭证。

ConnectDirectCredentials.xml
此文件包含 Connect:Direct 代理程序用于连接到 Connect:Direct 节点的凭证。

要保护存储在这些文件中的敏感信息，请使用 fteObfuscate 命令。 使用 **-f** 标志指定要保护的文件的名称。 例如:

```
fteObfuscate -f <File to protect>
```

缺省情况下， **fteObfuscate** 命令使用缺省初始密钥保护凭证。 要保护具有特定初始密钥的凭证，请使用 **-sf** 参数来指定包含初始密钥的文件的路径。 例如:

```
fteObfuscate -f <File to protect> -sf <initial key file>
```

警告: 所有 IBM MQ 安装的缺省初始密钥都相同。 要安全地保护密码，请在加密密码时提供对安装唯一的初始密钥。

注意:

1. 在运行 **fteObfuscate** 后，验证所有敏感信息是否都受保护。
2. 向 MFT 提供正常的受保护文件。

如果在运行 **fteObfuscate** 命令以保护 MFT 配置中的凭证时指定初始密钥文件，那么在 MFT 启动时还必须指定相同的初始密钥文件。 可以使用以下机制按优先级顺序指定初始密钥文件的名称:

1. **com.ibm.wmqfte.cred.keyfile** Java 系统属性。

   注: 在 IBM MQ 9.3.1 和 IBM MQ 9.3.0 Fix Pack 10 之前，此 Java 系统属性的名称拼写错误为 **com.ibm.wqmfte.cred.keyfile**。 从 IBM MQ 9.3.1 和 IBM MQ 9.3.0 Fix Pack 10 开始， Managed File Transfer 使用 Java 系统属性的两个版本来保持与较早版本的兼容性。 如果同时设置了两个 Java 系统属性，那么将使用正确拼写的属性 **com.ibm.wmqfte.cred.keyfile** 的值。
2. 代理程序，记录器，命令和协调属性文件中的属性。
3. `installation.properties` 文件中的 **commonCredentialsKeyFile** 属性。

有关更多信息，请参阅第 501 页的『对 MFT 中存储的凭证进行加密』。

缺省情况下， **fteObfuscate** 命令使用与低于 IBM MQ 9.2 的 MFT 版本不兼容的保护系统来保护凭证。 要使用与 IBM MQ 9.2 之前的版本兼容的凭证保护系统来保护配置文件，请在运行 **fteObfuscate** 命令时指定 **-sp 0** 参数。

IBM MQ Internet Pass-Thru

IBM MQ Internet Pass-Thru (MQIPT) 配置文件可以包含用于访问各种资源的密码。

使用 **mqiptPW** 命令保护 MQIPT 配置文件中的密码。

mqiptPW 命令提示输入要加密的密码，并返回已加密的密码。 将加密密码复制到 MQIPT 配置文件中。

缺省情况下， **mqiptPW** 命令使用缺省初始密钥加密密码。 要使用特定初始密钥对密码进行加密，请使用 **-sf** 参数指定包含初始密钥的文件的路径。

警告: 所有 IBM MQ 安装的缺省初始密钥都相同。 要安全地保护密码，请在加密密码时提供对安装唯一的初始密钥。

有关更多信息，请参阅 指定密码加密密钥。

如果在加密密钥存储库密码时指定初始密钥文件，那么在 MQIPT 启动时还必须指定相同的初始密钥文件。 可以使用以下机制按优先级顺序指定初始密钥文件的名称:

1. 用于启动 MQIPT 的命令上的 **-sf** 参数。
2. **MQS_MQIPTCRED_KEYFILE** 环境变量。
3. **com.ibm.mq.ipt.cred.keyfile** Java 属性。
4. MQIPT 主目录中名为 `mqipt_cred.key` 的文件。 MQIPT 主目录是包含 MQIPT 配置文件的目录。

缺省情况下， **mqiptPW** 命令使用与低于 IBM MQ 9.2 的 MQIPT 版本不兼容的保护系统来保护凭证。 要使用与 IBM MQ 9.2 之前的版本兼容的凭证保护系统来保护密码，请使用在 IBM MQ 9.2 之前的版本中支持的 **mqiptPW** 命令语法。

使用加密硬件的 IBM MQ clients

您可以将 IBM MQ 客户机配置为使用 PKCS #11 加密硬件来存储 TLS 通信中使用的专用密钥和证书。 要访问 PKCS #11 设备，必须在提供给 IBM MQ client 的配置字符串中提供密码。

要点: 无法使用此机制来保护使用 MQSCO 结构中的 **CryptoHardware** 字段提供的密码或队列管理器 **SSLCRYP** 属性。

您可以使用 **runp11cred** 命令来保护此密码，此命令可在 IBM MQ 安装目录的 `bin` 文件夹中找到。

runp11cred 命令提示输入要加密的密码，并返回已加密的密码。 必须将加密密码复制到加密硬件配置字符串中。

例如，如果加密硬件配置字符串如下所示:

```
GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;Passw0rd;SYMMETRIC_CIPHER_ON
```

当 **runp11cred** 命令提示您输入密码时，请输入 `Passw0rd`。 该命令返回类似于以下内容的字符串:

```
<P11>!2!0TyDxrRaS6JUsjON9zfK6S4wEHmSNF0/ZsOdCaTD2dc=!MdpCoxGnFqPtZ1dTLQ58kg==
```

将加密硬件配置字符串中的密码替换为 **runp11cred** 命令返回的字符串，以提供包含加密密码的以下字符串:

```
GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;<P11>!2!0TyDxrRaS6JUsjON9zfK6S4wEHm SNF0/
ZsOdCaTD2dc=!MdpCoxGnFqPtZ1dTLQ58kg==;SYMMETRIC_CIPHER_ON
```

当 IBM MQ client 应用程序运行时，通过下列其中一种方法指定包含加密密码的加密硬件配置字符串:

- 客户机配置文件的 SSL 节中的 **SSLCryptoHardware** 属性。
- **MQSSLCRYP** 环境变量。

缺省情况下， **runp11cred** 命令使用缺省初始密钥对密码进行加密。 要使用您自己的初始密钥保护密码，请使用下列其中一种机制按优先级顺序指定包含初始密钥的文件的名称:

1. **runp11cred** 命令的 **-sf** 参数。
2. **MQS_SSLCRYP_KEYFILE** 环境变量。

警告: 所有 IBM MQ 安装的缺省初始密钥都相同。 要安全地保护密码，请在加密密码时提供对安装唯一的初始密钥。

如果在加密密钥存储库密码时指定初始密钥文件，那么还必须指定在 IBM MQ client 运行时包含初始密钥的文件的名称。 使用下列其中一种机制按优先级顺序指定初始密钥文件名:

1. **MQS_SSLCRYP_KEYFILE** 环境变量。
2. 客户机配置文件的 **SSL** 节中的 **SSLCryptoHardwareKeyFile** 属性。

IBM MQ 队列管理器

IBM MQ 队列管理器在内部将密码存储在多个属性中。 例如，队列管理器 **KEYRPWD** 属性。 队列管理器会在密码存储在磁盘上的文件中之前自动对其进行加密。

可以使用 IBM MQ 密码保护系统或密钥存储库隐藏文件来保护队列管理器 TLS 密钥存储库的密码。 有关这两种方法的更多信息，请参阅 第 263 页的『对 AIX, Linux, and Windows 上的密钥存储库密码进行加密』。

当队列管理器加密密码时，将使用缺省初始密钥，除非您指定自己的初始密钥。 要使用您自己的初始密钥，请先将队列管理器 **INITKEY** 属性设置为唯一的强密钥，然后再设置任何加密队列管理器属性。

警告: 所有 IBM MQ 安装的缺省初始密钥都相同。 要安全地保护密码，请在加密密码时提供对安装唯一的初始密钥。

警告: 如果在设置已加密属性的值之后修改初始密钥，那么不会使用新的初始密钥对已加密属性进行重新加密。 因此，更改初始密钥而不重新提供密钥存储库口令会导致 IBM MQ 无法解密密钥存储库口令，并且无法访问密钥存储库。

有关更多信息，请参阅 INITKEY。

IBM MQ C 客户机应用程序

IBM MQ C 客户机库需要密码才能访问某些安全资源。 例如，用于使用 TLS 连接到队列管理器的应用程序的 TLS 密钥存储库。

可以使用 IBM MQ 密码保护系统或密钥存储库隐藏文件来保护密钥存储库密码。 有关这两种方法的更多信息，请参阅 第 263 页的『对 AIX, Linux, and Windows 上的密钥存储库密码进行加密』。

要使用 IBM MQ 密码保护系统保护密码，请使用 **runmqicred** 命令。 该命令位于 *MQ_INSTALLATION_PATH*/bin 目录中。

runmqicred 命令提示输入要加密的密码，并返回已加密的密码。 客户机应用程序可以使用加密密码来代替纯文本密码。

例如，如果选择使用 *MQKEYRPWD* 环境变量提供 TLS 密钥存储库密码，那么 TLS 密钥库密码为 Passw0rd。 运行 **runmqicred** 时，提示时输入 Passw0rd 。 该命令返回类似于以下内容的字符串:

```
<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w==
```

将此字符串设置为 *MQKEYRPWD* 环境变量的值:

```
export MQKEYRPWD="<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w=="
set MQKEYRPWD="<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w=="
```

缺省情况下， **runmqicred** 命令使用缺省初始密钥加密密码。 要使用您自己的初始密钥保护密码，请使用下列其中一种机制按优先级顺序指定包含该密钥的文件的名称:

1. **runmqicred** 命令的 **-sf** 参数。
2. *MQS_MQI_KEYFILE* 环境变量。

警告: 所有 IBM MQ 安装的缺省初始密钥都相同。 要安全地保护密码，请在加密密码时提供对安装唯一的初始密钥。

如果在加密密码时指定初始密钥文件，那么还必须使初始密钥在运行时可供客户机应用程序使用。

有关更多信息，请参阅第 267 页的『为 AIX, Linux, and Windows 上的 IBM MQ MQI client 提供密钥存储库密码』。

本机 HA 配置

V 9.4.0

可以使用 TLS 对实例之间的本机 HA 日志复制流量进行加密。 用于保护日志复制流量的证书存储在 `qm.ini` 文件的 **NativeHALocalInstance** 节中指定的密钥存储库中。

可以使用 IBM MQ 密码保护系统或密钥存储库隐藏文件来保护密钥存储库密码。 有关这两种方法的更多信息，请参阅 第 263 页的『对 AIX, Linux, and Windows 上的密钥存储库密码进行加密』。

要使用 IBM MQ 密码保护系统来保护本机 HA 密钥存储库密码，请使用 **runmqicred** 命令。

runmqicred 命令提示输入要加密的密码，并返回已加密的密码。 应使用加密密码而不是纯文本密码。 将 `qm.ini` 文件的 **NativeHALocalInstance** 节中 **KeyRepositoryPassword** 属性的值设置为命令返回的加密密码。

缺省情况下， **runmqicred** 命令使用缺省初始密钥加密密码。 要使用您自己的初始密钥保护密码，请使用下列其中一种机制按优先级顺序指定包含该密钥的文件的名称:

1. **runmqicred** 命令的 **-sf** 参数。
2. *MQS_MQI_KEYFILE* 环境变量。

警告: 所有 IBM MQ 安装的缺省初始密钥都相同。 要安全地保护密码，请在加密密码时提供对安装唯一的初始密钥。

如果在加密密钥存储库密码时指定初始密钥文件，那么还必须使用 `qm.ini` 文件的 **NativeHALocalInstance** 节中的 **InitialKeyFile** 属性来指定相同的初始密钥文件。

有关更多信息，请参阅 NativeHALocal qm.ini 文件的实例节。

IBM MQ 队列管理器 (qm.ini 文件中的 AuthToken 节)

Linux V 9.4.0 AIX

从 IBM MQ 9.3.4 开始，连接到在 AIX 或 Linux 系统上运行的 IBM MQ 队列管理器的 IBM MQ MQI clients 可以使用认证令牌向队列管理器进行认证。 队列管理器必须配置为接受认证令牌，并且能够访问令牌发布者的公用密钥证书或用于对令牌进行签名的密钥。 包含可信签发者的公用密钥证书或密钥的密钥存储库使用密码进行保护。

可以使用 IBM MQ 密码保护系统或密钥存储库隐藏文件来保护密钥存储库密码。 有关这两种方法的更多信息，请参阅 第 263 页的『对 AIX, Linux, and Windows 上的密钥存储库密码进行加密』。

要使用 IBM MQ 密码保护系统来保护认证令牌密钥存储库密码，请使用 **runqmcred** 命令对密码进行加密。

runqmcred 命令提示输入要加密的密码，并返回已加密的密码。 必须使用加密密码而不是纯文本密码。 将加密密码复制到文件中，并在 `qm.ini` 文件的 **AuthToken** 节的 **KeyStorePwdFile** 属性中包含该文件的路径。

缺省情况下， **runqmcred** 命令使用缺省初始密钥加密密码。 要使用特定初始密钥对密码进行加密，请使用 **-sf** 参数指定包含初始密钥的文件的路径。

警告: 所有 IBM MQ 安装的缺省初始密钥都相同。 要安全地保护密码，请在加密密码时提供对安装唯一的初始密钥。

要点: 如果在加密密码时提供初始密钥，那么必须在队列管理器 **INITKEY** 属性中指定相同的初始密钥，以便队列管理器可以解密密码。 如果已设置队列管理器 **INITKEY** 属性，请在运行 **runqmcred** 命令时使用相同的初始键。 有关队列管理器 **INITKEY** 属性的更多信息，请参阅 INITKEY。

例如，要使用文件 /home/initial.key 中的初始密钥对认证令牌密钥库密码进行加密，请发出以下命令:

```
runqmcred -sf /home/initial.key
```

有关更多信息，请参阅第 293 页的『配置队列管理器以使用本地密钥库接受认证令牌』。

通过密码加密进行保护时存在的限制

IBM MQ 支持对存储在各种配置文件中的密码进行 AES-128 加密。 使用高级加密标准 (AES) 加密来保护 IBM MQ 配置中的密码时，需要了解其提供的保护限制。

对 IBM MQ 配置文件中的密码进行加密并不意味着该密码是安全的或受保护的。 它只会阻止可以访问加密密码的人轻松恢复密码，但不知道加密密钥。 IBM MQ 进程需要同时访问加密密码和解密密钥以获取要使用的明文密码。 这两个数据项都必须存储在文件系统上可供 IBM MQ 访问的位置中。 加密配置文件中的密码的任何人也需要访问加密密钥。 如果攻击者有权访问与 IBM MQ 相同的文件集，那么对密码应用 AES 加密仅提供最低级别的保护。

尽管如此，对静态密码进行加密很重要，因为它可以防止意外泄露密码，并支持共享配置文件 (如果解密密钥不也是共享的)。

除了确保不共享包含解密密钥的文件外，还必须注意确保该文件不受系统上其他用户的保护。 虽然所有用户都可以访问 IBM MQ 配置文件，但请将包含解密密钥的文件上的许可权限制为最低要求。 必须向 IBM MQ 进程运行的用户标识授予读取包含解密密钥的文件的访问权。 但是，不需要将读取文件的访问权授予组或系统上的所有用户。

保护数据库认证详细信息

如果您正在使用用户名和密码认证来连接到数据库管理器，那么可以将其存储在 MQ XA 凭证库中，以避免将密码以明文形式存储在 qm.ini 文件中。

更新资源管理器的 XAOpenString

要使用凭证库，必须在 qm.ini 文件中修改 XAOpenString 。 该字符串用于连接到数据库管理器。 指定可替换字段以标识在 XAOpenString 字符串中替换用户名和密码的位置。

- +USER+ 字段将替换为存储在 XACre 其内的用户名值。
- +PASSWORD+ 字段将替换为存储在 XACre 其内的密码值。

以下示例显示如何修改 XAOpenString 以使用凭证文件连接到数据库。

连接到 Db2 数据库

```
XAResourceManager:
  Name=mydb2
  SwitchFile=db2swit
  XAOpenString=db=mydbname,uid=+USER+,pwd=+PASSWORD+,toc=t
  ThreadOfControl=THREAD
```

连接到 Oracle 数据库

```
XAResourceManager:
  Name=myoracle
  SwitchFile=oraswit
  XAOpenString=Oracle_XA+Acc=P/+USER+/+PASSWORD++SesTm=35
```

```
            +LogDir=/tmp+threads=true
  ThreadOfControl=THREAD
```

使用数据库到 MQ XA 凭证库的凭证

使用可替换凭证字符串更新 qm.ini 文件后，必须使用 **setmqxacred** 命令将用户名和密码添加到 MQ 凭证库。 您还可以使用 **setmqxacred** 来修改现有凭证，删除凭证或列出凭证。 以下示例提供了一些典型用例:

添加凭证

以下命令安全地保存资源 mqdb2 的队列管理器 QM1 的用户名和密码。

```
setmqxacred -m QM1 -x mydb2 -u user1 -p Password2
```

更新凭证

要更新用于连接到数据库的用户名和密码，请使用新的用户名和密码重新发出 **setmqxacred** 命令:

```
setmqxacred -m QM1 -x mydb2 -u user3 -p Password4
```

必须重新启动队列管理器才能使更改生效。

删除凭证

以下命令将删除凭证:

```
setmqxacred -m QM1 -x mydb2 -d
```

列出凭证

以下命令列出凭证:

```
setmqxacred -m QM1 -l
```

相关参考

setmqxacred

保护 Managed File Transfer

安装后没有进行任何修改的情况下， Managed File Transfer 的安全级别适用于受保护环境中的测试或评估目的。 但是在生产环境中，必须考虑适当控制可以启动文件传输操作的人员、可以读和写正在传输的文件的人员以及如何保护文件完整性。

相关任务

限制特定于 MFT 的资源的组权限

管理特定于 MFT 资源的权限

第 558 页的『将 Advanced Message Security 用于 Managed File Transfer』

此场景说明如何配置 Advanced Message Security 以提供通过 Managed File Transfer 发送的数据的消息隐私。

相关参考

供 MFT 用于访问文件系统的权限

commandPath MFT 属性

用于发布 MFT 代理日志和状态消息的权限

对 MFT 中存储的凭证进行加密

Managed File Transfer (MFT) 需要多个存储在两个 XML 文件中的用户标识和凭证，您可以使用 **fteObfuscate** 命令对这些用户标识和凭证进行模糊处理。

凭证文件

MQMFTCredentials.xml
此文件包含用于连接到代理以及协调和命令队列管理器的用户标识和凭证。 用于访问密钥库以实现与队列管理器的安全连接的凭证也存储在同一文件中。

请参阅 第 504 页的『MFT 和 IBM MQ 连接认证』，以获取定义 `MQMFTCredentials.xml` 文件位置的属性值的详细信息。

ProtocolBridgeCredentials.xml
此文件包含用于连接到协议服务器的用户标识和凭证。

使用 fteObfuscate 命令加密凭证

fteObfuscate 命令接受以下参数:

- **-f** *credentials_file_name* (必需)

  注: Deprecated 此参数将替换 IBM MQ 9.2.0 中不推荐使用的 **-credentialsFile** 参数。
- **-sp** 保护方式
- **-sf** 凭证密钥文件
- **-o** 输出文件名称

有关参数的详细信息，请参阅 **fteObfuscate**。

如果未指定保护方式或凭证密钥文件，那么该命令将使用缺省保护方式，并使用最新算法，但使用固定密钥对凭证进行加密。

如果指定保护方式 `0`，并且未指定凭证密钥文件，那么该命令将与产品的先前发行版中一样工作。 您在控制台上接收到一条警告消息，指示不推荐使用的保护。

如果指定保护方式 `0`，并指定凭证密钥文件，那么将在控制台上接收到错误输出，指示在使用保护方式 `0` 时指定密钥文件无效。

如果指定 1 的保护方式，并且未指定凭证密钥文件，那么该命令将使用最新算法，但使用固定密钥对凭证进行加密。

如果指定 1 的保护方式，并指定凭证密钥文件，那么该命令将使用最新算法对凭证进行加密。

如果指定 1 的保护方式，或者不指定保护方式，并指定不存在的凭证密钥文件，那么将在控制台上输出错误，指示该文件不存在。

如果指定 1 的保护方式，或者不指定保护方式，并指定不可读的凭证密钥文件，那么将在控制台上输出错误，指示该文件不可读。

如果指定 2 的保护方式，并且未指定凭证密钥文件，那么该命令将使用保护方式 2 通过最新算法和要加密的固定密钥来加密凭证。

如果指定 2 的保护方式，并指定凭证密钥文件，那么该命令将使用保护方式 2 来使用最新算法加密凭证，并使用用户指定的密钥进行加密。

如果指定 2 的保护方式，或者不指定保护方式，并指定不存在的凭证密钥文件，那么将在控制台上输出错误，指示该文件不存在。

如果指定 2 的保护方式，或者不指定保护方式，并指定不可读的凭证密钥文件，那么将在控制台上输出错误，指示该文件不可读。

解密凭证

您可以在各种位置指定初始密钥文件的路径。 为了解密使用缺省密钥以外的初始密钥加密的凭证，需要通过以下某种方式向 MFT 提供包含初始密钥的文件的名称 (按此优先顺序):

1. 通过使用 Java 系统属性，例如:

```
-Dcom.ibm.wmqfte.cred.keyfile=/usr/hime/credkeyfile.key
```

注:

- 在 IBM MQ 9.3.1 和 IBM MQ 9.3.0 Fix Pack 10 之前，此 Java 系统属性的名称在产品代码中拼写错误为 `com.ibm.wqmfte.cred.keyfile`。 从 IBM MQ 9.3.1 和 IBM MQ 9.3.0 Fix Pack 10 开始，将属性名称的拼写更正为 `com.ibm.wmqfte.cred.keyfile`。 当检查用户是否指定了包含应用于对凭证进行加密和解密的初始密钥的文件时， Managed File Transfer 将使用这两个版本的 Java 系统属性。 这允许您使用属性名的正确拼写，同时与旧的拼写错误的名称保持向后兼容性。 请注意，如果同时设置了两个 Java 系统属性，那么将使用正确拼写的属性 `com.ibm.wmqfte.cred.keyfile` 的值。
- 在 IBM MQ 9.3.1 和 IBM MQ 9.3.0 Fix Pack 10 之前，请使用属性 `com.ibm.wqmfte.cred.keyfile`。

.

2. 通过在代理，命令，协调或记录器属性文件中设置属性。 下表显示了属性文件的名称以及需要在其中设置的属性:

属性文件	属性名
agent.properties	agentCredentialsKeyFile
command.properties	commandCredentialsKeyFile
coordination.properties	coordinationCredentialsKeyFile
logger.properties	loggerCredentialsKeyFile

3. 在 installation.properties 文件中。

   您可以将 **commonCredentialsKeyFile** 属性添加到现有公共 `installation.properties` 文件中，以便代理程序，记录器和命令可以使用相同的属性，而不是在各个属性文件中添加属性。

如果您在多个位置定义了各种 **CredentialsKeyFile** 属性:

- 用于代理和记录器的凭证密钥文件的路径将记录到该代理或记录器的 `output0.log` 文件中。
- 将在控制台上显示用于命令的凭证密钥文件的路径。

Java 系统属性 **com.ibm.wmqfte.cred.keyfile** 将覆盖所有其他属性。 如果未设置系统属性，那么代理程序将查找 `agent.properties` 文件，后跟初始密钥文件的 `installation.properties` 文件。

如果仍未找到初始密钥文件，并且您已将 **fteObfuscate** 命令上的保护方式设置为 `1`，那么代理程序会在 `output0.log` 文件中记录一条错误消息。

如果在 **fteObfuscate** 命令上将保护方式设置为 `0` ，那么将记录一条警告消息以指示不推荐使用。

记录器和命令遵循查找初始密钥文件的相同步骤。

协议网桥和 Connect:Direct 网桥

Protocol Bridge 使用属性文件 `ProtocolBridgeProperties.xml` 来连接到 FTP， SFTP 和 FTPS 服务器。 此属性文件包含连接到这些服务器所需的连接属性。

如果修改 `ProtocolBridgeProperties.xml` 文件中 **credentialsFile** 或 **credentialsKeyFile** 属性的值，那么需要重新启动网桥代理。

其中一个属性为 **credentialsFile**，该值包含包含连接到这些服务器所需的 UID， PWD 或 Key 的 XML 文件的路径。 该属性的缺省值为 *ProtocolBridgeCredentials.xml* ，该文件与 `MQMFTCredentials.xml` 文件一样位于主目录中。

```
<tns:credentialsFile path="$HOME/ProtocolBridgeCredentials.xml" />
```

就像 *MQMFTCredentails.xml* 一样，您可以使用 **fteObfuscate** 命令对 *ProtocolBridgeCredentials.xml* 进行加密。 出于解密目的，您可以使用其他元素 **credentialsKeyFile** 指定凭证密钥文件的必需路径，如以下文本中所示。 路径可以包含环境变量。

```
<tns:credentialsKeyFile path="$HOME/CredKey.key"/>
```

注: 在 `installation.properties` 中或通过系统属性 **com.ibm.wqmfte.cred.keyfile** 为 **agentCredentialsKeyFile** 代理程序属性 **commonCredentialsKeyFile** 属性指定值不会对为 **credentialsKeyFile** 属性指定的值产生任何影响。

同样， Connect:Direct Bridge 使用 *ConnectDirectNodeProperties.xml* 来连接到 Connect:Direct 服务器。 XML 文件包含必需的连接信息以及定义凭证 XML 文件路径的属性。 此凭证 XML 文件包含 UID 或 PWD 以及连接到 Connect:Direct 服务器所需的其他信息。

```
<tns:credentialsFile path="$HOME/ ConnectDirectCredentials.xml" />
```

就像 *ProtocolBridgeCredentials.xml* 文件一样，您可以使用 **fteObfuscate** 命令对 *ConnectDirectCredentials.xml* 进行加密。 出于解密目的，您可以使用其他元素 **credentialsKeyFile** 指定凭证密钥文件的必需路径，如以下文本中所示。 路径可以包含环境变量。

```
<tns:credentialsKeyFile path="$HOME/CredKey.key"/>
```

注: 在 `installation.properties` 中或通过系统属性 **com.ibm.wqmfte.cred.keyfile** 为 **agentCredentialsKeyFile** 代理程序属性 **commonCredentialsKeyFile** 指定值不会对为 **credentialsKeyFile** 属性指定的值产生任何影响。

您可以指定 **credentialsKeyFile** 元素，而无需在 *ProtocolBridgeProperties.xml* 文件中指定 **credentialsFile** 元素。

如果未指定 **credentialsFile** 元素，那么协议网桥代理将使用缺省凭证文件 *ProtocolBridgeCredentials.xml* ，并且 **credentialsKeyFile** 属性中指定的密钥文件的值将用于解密凭证文件。

同样，您可以指定 **credentialsKeyFile** 元素，而无需在 *ConnectDirectNodeProperties.xml* 文件中指定 **credentialsFile** 元素。

如果未指定 **credentialsFile** 元素，那么缺省凭证文件 *ConnectDirectCredentials.xml* 由 Connect:Direct 网桥使用，并且 **credentialsKeyFile** 属性中指定的密钥文件的值用于解密凭证文件。

在 z/OS 上使用数据集的密钥

在 z/OS 上，可以指定 **MQMFTCredentials** 并使用 PDSE 提供凭证密钥文件。 请参阅第 506 页的『Configuring MQMFTCredentials.xml on z/OS』。

相关参考

MFT 命令与队列管理器的连接关系

MFT 凭证文件格式

fteObfuscate (加密敏感数据)

MFT 和 IBM MQ 连接认证

连接认证允许将队列管理器配置为使用提供的用户标识和密码对应用程序进行认证。 如果关联的队列管理器已启用安全性，并且需要凭证详细信息 (用户标识和密码) ，那么必须先启用连接认证功能，才能成功连接到队列管理器。 可以在兼容性方式或 MQCSP 认证方式下运行连接认证。

提供凭证详细信息的方法

许多 Managed File Transfer 命令支持以下提供凭证详细信息的方法:

由命令行参数提供详细信息。
可以使用 **-mquserid** 和 **-mqpassword** 参数来指定凭证详细信息。 如果未提供 **-mqpassword** ，那么将要求用户提供未显示输入的密码。

由凭证文件 MQMFTCredentials.xml 提供详细信息。
可以在 `MQMFTCredentials.xml` 文件中将凭证详细信息预定义为明文或模糊文本。

Multi 有关在 IBM MQ for Multiplatforms 上设置 MQMFTCredentials.xml 文件的信息，请参阅 第 505 页的『在 Multiplatforms 版上配置 MQMFTCredentials.xml』。

z/OS 有关在 IBM MQ for z/OS 上设置 MQMFTCredentials.xml 文件的信息，请参阅 第 506 页的『Configuring MQMFTCredentials.xml on z/OS』。

优先顺序

确定凭证详细信息的过程为：

1. 命令行参数。
2. 关联的队列管理器和运行命令的用户的 `MQMFTCredentials.xml` 索引。
3. 关联的队列管理器的 `MQMFTCredentials.xml` 索引。
4. 缺省向后兼容性方式，其中未提供凭证详细信息以允许与 IBM MQ 或 IBM WebSphere MQ 的先前发行版兼容

注意：

- **fteStartAgent** 和 **fteStartLogger** 命令不支持命令行参数 **-mquserid** 或 **-mqpassword**，并且只能使用 `MQMFTCredentials.xml` 文件来指定凭证详细信息。
- z/OS

  在 z/OS 上，即使用户的密码包含小写字母，该密码也必须为大写形式。 例如，如果用户的密码是“password”，那么必须输入为“PASSWORD”。

相关参考

MFT 命令与队列管理器的连接关系

MFT 凭证文件格式

在 Multiplatforms 版上配置 MQMFTCredentials.xml

如果 Managed File Transfer (MFT) 配置为启用安全性，那么连接认证需要与队列管理器连接的所有 MFT 命令来提供用户标识和密码凭证。 同样，在连接到数据库时，可能需要 MFT 记录器来指定用户标识和密码。 此凭证信息可以存储在 MFT 凭证文件中。

关于此任务

`MQMFTCredentials.xml` 文件中的元素必须符合 `MQMFTCredentials.xsd` 模式。 有关 `MQMFTCredentials.xml` 格式的信息，请参阅 MFT 凭证文件格式。

您可以在 `MQ_INSTALLATION_PATH/mqft/samples/credentials` 目录中找到样本凭证文件。

您可以为协调队列管理器、命令队列管理器、每个代理以及每个记录器分别提供一个 MFT 凭证文件。 或者，您可以有一个文件供拓扑中的所有内容使用。

MFT 凭证文件的缺省位置如下所示：

Linux AIX **AIX and Linux**

`$HOME`

Windows **Windows**

`%USERPROFILE%` 或 `%HOMEDRIVE%%HOMEPATH%`

如果凭证文件存储在其他位置，那么可以使用以下属性来指定命令应在何处查找该文件：

表 *97: :* 用于定义各种命令的 *MQMFTCredentials.xml* 文件的位置的属性。

命令类型	属性文件	属性名
用于连接到协调队列管理器的命令	coordination.properties	coordinationQMgrAuthenticationCredentialsFile

表 97: : 用于定义各种命令的 *MQMFTCredentials.xml* 文件的位置的属性。 *(继续)*		
命令类型	**属性文件**	**属性名**
连接到命令队列管理器的命令	connection.properties	connectionQMgrAuthenticationCredentialsFile
连接到代理进程的命令	agent.properties	agentQMgrAuthenticationCredentialsFile
连接到记录器进程的命令	logger.properties	loggerQMgrAuthenticationCredentialsFile

表 98: : 用于定义代理程序和记录器进程的 *MQMFTCredentials.xml* 文件的位置的属性。		
命令类型	**属性文件**	**属性名**
MFT 代理程序	agent.properties	agentQMgrAuthenticationCredentialsFile
MFT loggers	logger.properties	loggerQMgrAuthenticationCredentialsFile

有关哪些命令和进程连接到哪个队列管理器的详细信息，请参阅 哪些 MFT 命令和进程连接到哪个队列管理器。

您可以将 **commonCredentialsKeyFile** 属性添加到现有公共 `installation.properties` 文件，以便代理程序，记录器和命令可以使用相同的属性，而不是在各个属性文件中添加属性。

由于凭证文件包含用户标识和密码信息，因此需要特殊许可权以防止对其进行未经授权的访问:

Linux AIX **AIX and Linux**

```
chown <agent owner userid>
chmod 600
```

Windows **Windows**

确保未启用继承，然后除去除运行将使用凭证文件的代理程序或记录器的用户标识以外的所有用户标识。

用于连接到 IBM MQ Explorer Managed File Transfer 插件中的 MFT 协调队列管理器的凭证详细信息取决于配置类型:

全局（本地磁盘上的配置）
全局配置将使用在协调和命令属性中指定的凭证文件。

本地（在 IBM MQ Explorer 中定义）：
局部配置将使用 IBM MQ Explorer 中的关联队列管理器的连接详细信息的属性。

相关任务

第 508 页的『为 MFT 启用连接认证』
可以在兼容性方式或 MQCSP 认证方式下运行与协调队列管理器或命令队列管理器连接的 IBM MQ Explorer MFT 插件的连接认证以及与协调队列管理器或命令队列管理器连接的 Managed File Transfer 代理的连接认证。

创建 IBM MQ 文件传输结构

相关参考

MFT 凭证文件格式
对 MFT 中存储的凭证进行加密
fteObfuscate：对敏感数据进行加密

z/OS Configuring MQMFTCredentials.xml on z/OS

If Managed File Transfer (MFT) is configured with security enabled, connection authentication requires all MFT agents, and commands that connect to a queue manager, to supply user ID and password credentials.

Similarly, MFT loggers might be required to specify a user ID and password when connecting to a database.

This credential information can be stored in the MFT credentials file. Note that the credentials files are optional, however, it is easier to define the file or files that you require before you customize the environment.

In addition to this, if you have credentials files, you receive fewer warning messages. The warning messages inform you that MFT considers that queue manager security is off, and therefore you are not supplying authentication details.

You can find a sample credentials file in the `MQ_INSTALLATION_PATH/mqft/samples/credentials` directory.

Here is an example of an `MQMFTCredentials.xml`file:

```
<?xml version="1.0" encoding="IBM-1047"?>
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MFTCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/MFTCredentials MFTCredentials.xsd">
  <tns:qmgr name="MQPH" user="ADMIN" mqUserId="JOHNDOEH" mqPassword="cXXXX" />
  <tns:qmgr name="MQPI" user="ADMIN" mqUserId="JOHNDOEI" mqPassword="yXXXX" />
  <tns:qmgr name="MQPH" mqUserId="NONEH" mqPassword="yXXXX" />
  <tns:qmgr name="MQPI" mqUserId="NONEI" mqPassword="yXXXX" />
</tns:mqmftCredentials>
```

When a job with userid ADMIN needs to connect to queue manager MQPH, it passes user ID *JOHNDOEH* and uses password *cXXXX*.

If the job is run by any other user ID, and connects MQPH, that job passes user ID *NONEH* and password *yXXXX*.

The default location for the `MQMFTCredentials.xml` file is the user's home directory on z/OS UNIX System Services (USS). It is also possible to store the file in either a different location on USS, or in a member within a partitioned data set.

If the credentials file is stored in a different location, then you can use the following properties to specify where the commands should look for it:

Table 99. : Properties that define the location of the MQMFTCredentials.xml file for various commands.

Type of command	**Property file**	**Property name**
Command which connects to the coordination queue manager	coordination.properties	coordinationQMgrAuthenticationCredentialsFile
Command which connects to the command queue manager	connection.properties	connectionQMgrAuthenticationCredentialsFile
Command that connects to an agent process	agent.properties	agentQMgrAuthenticationCredentialsFile
Command that connects to a logger process	logger.properties	loggerQMgrAuthenticationCredentialsFile

Table 100. : Properties that define the location of the MQMFTCredentials.xml file for agents and logger processes.

Type of command	**Property file**	**Property name**
MFT agents	agent.properties	agentQMgrAuthenticationCredentialsFile
MFT loggers	logger.properties	loggerQMgrAuthenticationCredentialsFile

For details about what commands and processes connect to which queue manager, see Which MFT commands and processes connect to which queue manager.

To create the credentials file within a partitioned data set, carry out the following steps:

- Create a PDSE with format VB and logical record length (Lrecl) 200.
- Create a member within the data set, make a note of the data set and member, and add the following code to the member:

```
<?xml version="1.0" encoding="IBM-1047"?>
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MQMFTCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/MFTCredentials MQMFTCredentials.xsd">
  <!—credentials information goes here-->
</tns:mqmftCredentials>
```

You can protect the credentials file using a security product, for example, RACF, but the user IDs running the Managed File Transfer commands, and administering the agent and logger processes, need read access to this file.

You can obscure information in this file using the JCL in member BFGCROBS. This takes the file and encrypts the IBM MQ user ID and password. For example member BFGCROBS takes the line

```
<tns:qmgr name="MQPI" user="JOHNDOE2" mqUserId="JOHNDOE1" mqPassword="yXXXX" />
```

and creates

```
<tns:qmgr mqPasswordCipher="e977c61e9b9c363c" mqUserIdCipher="c394c5887867157c"
name="MQPI" user="JOHNDOE2"/>
```

If you want to keep the user ID to IBM MQ user ID mapping, you can add comments to the file. For example

```
<!--      name="MQPI" user="ADMIN"    mqUserId="JOHNDOE1 -->
```

These comments are unchanged by the obscuring process.

Note that the content is obscured, not strongly encrypted. You should limit which user IDs have access to the file.

Related tasks

“在 Multiplatforms 版上配置 MQMFTCredentials.xml” on page 505

如果 Managed File Transfer (MFT) 配置为启用安全性，那么连接认证需要与队列管理器连接的所有 MFT 命令来提供用户标识和密码凭证。 同样，在连接到数据库时，可能需要 MFT 记录器来指定用户标识和密码。 此凭证信息可以存储在 MFT 凭证文件中。

为 MFT 启用连接认证

可以在兼容性方式或 MQCSP 认证方式下运行与协调队列管理器或命令队列管理器连接的 IBM MQ Explorer MFT 插件的连接认证以及与协调队列管理器或命令队列管理器连接的 Managed File Transfer 代理的连接认证。

关于此任务

MQCSP 认证方式是缺省方式。

对于 IBM MQ Explorer Managed File Transfer 插件或使用 CLIENT 传输连接到队列管理器的 Managed File Transfer 代理程序的连接认证，仅 MQCSP 认证方式支持长度超过 12 个字符的密码。 如果在使用兼容性方式进行授权时指定长度超过 12 个字符的密码，那么将发生错误，并且代理程序不会向队列管理器进行认证。 请参阅 诊断消息: BFGAG0001 - BFGAG9999 中的 BFGAG0187E 消息。

过程

- 要在 IBM MQ Explorer 中选择协调队列管理器或命令队列管理器的连接认证方式，请完成以下步骤:

a) 选择要连接到的队列管理器。

b) 单击右键，并从弹出菜单中选择**连接详细信息 -> 属性**。

c) 单击**用户标识**选项卡。

d) 确保选中要使用的连接认证方式的复选框:

- 缺省情况下，未选中 **用户标识兼容性方式** 复选框。 这意味着如果选中 **启用用户标识** 复选框，那么 IBM MQ Explorer 将在连接到队列管理器时使用 MQCSP 认证。 如果 IBM MQ Explorer 需要使用兼容性方式而不是 MQCSP 认证来连接到队列管理器，请确保选中 **启用用户标识** 和 **用户标识兼容性方式** 复选框。

- 要使用 `MQMFTCredentials.xml` 文件对 Managed File Transfer 代理程序启用或禁用 MQCSP 认证方式，请将参数 **useMQCSPAuthentication** 添加到相关用户的 `MQMFTCredentials.xml` 文件中。

  useMQCSPAuthentication 参数具有以下值:

  true
  MQCSP 认证方式用于向队列管理器认证用户。

  `true` 是缺省值。 如果未指定 **useMQCSPAuthentication** 参数，那么缺省情况下将其设置为 `true` ，并且 MQCSP 认证方式用于向队列管理器认证用户。

  false
  兼容性方式用于向队列管理器认证用户。

  以下示例显示了如何设置 `MQMFTCredentials.xml` 文件中的 **useMQCSPAuthentication** 参数:

```
<tns:qmgr name="CoordQueueMgr" user="ernest" mqUserId="ernest"
    mqPassword="AveryL0ngPassw0rd2135" useMQCSPAuthentication="true"/>
```

相关概念

第 27 页的『MQCSP 密码保护』
MQCSP 结构中指定的认证凭证可以使用 IBM MQ MQCSP 密码保护功能进行保护，也可以使用 TLS 加密进行加密。

相关参考

第 504 页的『MFT 和 IBM MQ 连接认证』
连接认证允许将队列管理器配置为使用提供的用户标识和密码对应用程序进行认证。 如果关联的队列管理器已启用安全性，并且需要凭证详细信息 (用户标识和密码) ，那么必须先启用连接认证功能，才能成功连接到队列管理器。 可以在兼容性方式或 MQCSP 认证方式下运行连接认证。

MFT 凭证文件格式

MFT 沙箱

您可以限制在传输过程中代理可访问的文件系统区域。 将代理限制到的区域称为沙箱。 您可以对代理或请求传输的用户应用限制。

当代理是协议网桥代理或 Connect:Direct 网桥代理时，不支持沙箱。 不能将代理沙箱用于需要与 IBM MQ 队列传输数据的代理。

相关参考

第 509 页的『使用 MFT 代理沙箱』
要向 Managed File Transfer 添加额外的安全性级别，可以限制代理可以访问的文件系统区域。

第 511 页的『使用 MFT 用户沙箱』
您可以根据请求传输的 MQMD 用户名来限制可在其中传输文件的文件系统的区域。

使用 MFT 代理沙箱

要向 Managed File Transfer 添加额外的安全性级别，可以限制代理可以访问的文件系统区域。

不能将代理沙箱用于需要与 IBM MQ 队列传输数据的代理。 可使用沙箱限制对 IBM MQ 队列的访问，而不使用作为任何沙箱需求推荐解决方案的用户沙箱。 有关用户沙箱的更多信息，请参阅第 511 页的『使用 MFT 用户沙箱』

要启用代理沙箱，请向要限制的代理的 agent.properties 文件添加以下属性：

```
sandboxRoot=[!]restricted_directory_nameseparator...separator[!]restricted_directory_name
```

其中：

- *restricted_directory_name* 是要允许或拒绝的目录路径。
- ！是可选的，并指定拒绝 (排除) *restricted_directory_name* 的以下值。 如果未指定 !，那么表示 *restricted_directory_name* 是一个允许（包含）路径。
- *separator* 是特定于平台的分隔符。

例如，如果要仅限制 AGENT1 对 /tmp 目录的访问权，但不允许访问子目录 private，可在属于 AGENT1 的 agent.properties 文件中按照如下方式设置属性：sandboxRoot=/tmp:!/tmp/private。

高级代理属性中描述了 sandboxRoot 属性。

在协议网桥代理或 Connect:Direct 网桥代理上不支持代理和用户沙箱。

在 AIX, Linux, and Windows 平台上的沙箱中工作

ALW 在 AIX, Linux, and Windows 平台上，沙箱限制 Managed File Transfer Agent 可以对哪些目录进行读写操作。 激活沙箱后，Managed File Transfer Agent 可以读写指定为允许的目录以及指定目录包含的任何子目录，除非在 sandboxRoot 中将这些子目录指定为拒绝。 Managed File Transfer 沙箱不优先于操作系统安全性。 启动 Managed File Transfer Agent 的用户必须对任何目录具有相应的操作系统级别访问权，才能对该目录进行读写。 如果链接到的目录在指定的 sandboxRoot 目录（及子目录）之外，那么不遵循到目录的符号链接。

在 z/OS 系统上的沙箱中工作

z/OS 在 z/OS 上，沙箱限制 Managed File Transfer Agent 可读写的数据集名称限定符。 启动 Managed File Transfer Agent 的用户必须对任何涉及的数据集具有正确的操作系统权限。 如果使用双引号将 sandboxRoot 数据集名称限定符值引起来，那么该值遵循标准的 z/OS 约定并被视为标准值。 如果省略双引号，那么 sandboxRoot 将以当前用户标识为前缀。 例如，如果将 sandboxRoot 属性设置如下：sandboxRoot=//test，那么代理可以访问以下数据集（采用标准 z/OS 表示法）//*username*.test.** 。在运行时，如果完全解析的数据集名称的初始级别与 sandboxRoot 不匹配，那么会拒绝传输请求。

在 IBM i 系统上的沙箱中工作

IBM i 对于 IBM i 系统上集成文件系统中的文件，沙箱限制了 Managed File Transfer Agent 可读写的目录。 激活沙箱后，Managed File Transfer Agent 可以读写指定为允许的目录以及指定目录包含的任何子目录，除非在 sandboxRoot 中将这些子目录指定为拒绝。 Managed File Transfer 沙箱不优先于操作系统安全性。 启动 Managed File Transfer Agent 的用户必须对任何目录具有相应的操作系统级别访问权，才能对该目录进行读写。 如果链接到的目录在指定的 sandboxRoot 目录（及子目录）之外，那么不遵循到目录的符号链接。

相关参考

第 513 页的『针对通配符传输的额外检查』
如果已使用用户或代理沙箱配置代理，以限制代理可将文件传输到的位置，您可以指定要对该代理的通配符传输进行其他检查。

第 509 页的『使用 MFT 代理沙箱』
要向 Managed File Transfer 添加额外的安全性级别，可以限制代理可以访问的文件系统区域。

MFT agent.properties 文件

使用 MFT 用户沙箱

您可以根据请求传输的 MQMD 用户名来限制可在其中传输文件的文件系统的区域。

当代理是协议网桥代理或 Connect:Direct 网桥代理时，不支持用户沙箱。

要启用用户沙箱，请为您想要限制的代理的 agent.properties 文件添加以下属性：

```
userSandboxes=true
```

如果此属性存在并且设置为 true，那么代理会使用 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/agents/*agent_name*/UserSandboxes.xml 文件中的信息来确定请求传输的用户可以访问文件系统的哪些部分。

UserSandboxes.xml XML 由 <agent> 元素组成，该元素包含零个或多个 <sandbox> 元素。 这些元素描述哪些规则适用于哪些用户。 <sandbox> 元素的 user 属性是一种模式，用于匹配请求的 MQMD 用户。

代理会定期重新装入 UserSandboxes.xml 文件，对该文件的任何有效更改都会影响代理的行为。 缺省重新装入时间间隔为 30 秒。 此时间间隔可通过在 agent.properties 文件中指定代理程序属性 xmlConfigReloadInterval 进行更改。

如果指定 userPattern="regex" 属性或值，那么 user 属性会解释为 Java 正则表达式。 有关更多信息，请参阅 MFT 使用的正则表达式。

如果不指定 userPattern="regex" 属性或值，那么 user 属性会被解释为具有以下通配符的模式：

- 星号 (*)，表示零个或更多个字符
- 问号 (?)，表示仅一个字符

将按照 <sandbox> 元素在文件中的列示顺序来执行匹配项。 将仅使用第一个匹配项，将忽略文件中的所有其余潜在匹配项。 如果文件中指定的 <sandbox> 元素与传输请求消息的关联 MQMD 用户均不匹配，那么传输将无法访问文件系统。 如果在 MQMD 用户名与 user 属性之间发现匹配项，那么该匹配项将标识 <sandbox> 元素中应用于传输的一组规则。 这组规则用于确定在传输过程中可以对哪些文件或数据集执行读或写操作。

每组规则都可以指定 <read> 元素（标识可以读取的文件）和一个 <write> 元素（标识可以写入的文件）。 如果在规则集中省略 <read> 或 <write> 元素，那么假定相应地不允许与规则集相关联的用户执行任何读或写操作。

注: 在 UserSandboxes.xml 文件中，<read> 元素必须在 <write> 元素之前，<include> 元素必须在 <exclude> 元素之前。

每个 <read> 或 <write> 元素都包含一个或多个模式，用来确定文件是否在沙箱中以及是否可以传输。 请使用 <include> 和 <exclude> 元素来指定这些模式。 <include> 或 <exclude> 元素的 name 属性指定要匹配的模式。 可选属性 type 指定名称值是文件还是队列模式。 如果未指定 type 属性，那么代理会把该模式视为文件或目录路径模式。 例如：

```
<tns:read>
    <tns:include name="/home/user/**"/>
    <tns:include name="USER.**" type="queue"/>
    <tns:exclude name="/home/user/private/**"/>
</tns:read>
```

代理会使用 <include> 和 <exclude> name 模式来确定可以对哪些文件、数据集或队列执行读或写操作。 如果规范文件路径、数据集或队列名称与至少一个包含的模式匹配，并且不与任何排除的模式匹配，那么允许执行操作。 使用 <include> 和 <exclude> 元素的 name 属性指定的模式会使用适用于平台（在其上运行代理）的路径分隔符和约定。 如果您指定相对文件路径，那么将相对于代理的 transferRoot 属性来解析该路径。

指定队列限制时，支持语法 QUEUE@QUEUEMANAGER，且规则如下：

- 如果条目中缺失 @ 字符，那么该模式将被视为可在任何条目管理器上访问的队列名称。 例如，如果模式为 name，那么该模式将同样被视为 name@**。

- 如果 @ 字符是条目中的第一个字符，那么模式将视为队列管理器名称，可以访问队列管理器上所有队列。例如，如果模式为 @name，那么该模式将同样被视为 **@name。

如果您将以下通配符指定为 <include> 和 <exclude> 元素的 name 属性的一部分，那么这些通配符具有特殊含义：

*****
单个星号与目录名称或 数据集名称或 队列名称的限定符中的零个或多个字符匹配。

?
问号与目录名称或 数据集名称或 队列名称的限定符中的一个字符完全匹配。

******
两个星号字符与零个或多个目录名匹配，或者与 数据集名称或 队列名称中的零个或多个限定符匹配。此外，以路径分隔符结尾的路径在路径末尾添加了隐式 “**” 。 因此， /home/user/ 与 /home/user/** 相同。

例如：

- /**/test/** 匹配路径中含有 test 目录的任何文件
- /test/file? 匹配 /test 目录中以 file 字符串开头并后跟任何单个字符的任何文件
- c:\test*.txt 匹配 c:\test 目录中扩展名为 .txt 的任何文件
- c:\test***.txt 与 'c:\test 目录中的任何文件或其某个具有 .txt 扩展名的子目录相匹配
- z/OS //'TEST.*.DATA' 匹配第一位限定符为 TEST 且第二位和第三位限定符为 DATA 的任何数据集。
- *@QM1 与队列管理器 QM1 上具有单个限定符的任何队列匹配。
- TEST.*.QUEUE@QM1 匹配队列管理器 QM1 上第一位限定符为 TEST 且第二位和第三位限定符为 QUEUE 的任何队列。
- **@QM1 与队列管理器 QM1 上的任何队列匹配。

符号链接

您必须在 <include> 和 <exclude> 元素中指定硬链接，以便完全解析在 UserSandboxes.xml 文件的文件路径中使用的任何符号链接。 例如，如果您有一个符号链接，其中 /var 映射到 /SYSTEM/var，那么必须将此路径指定为 <tns:include name="/SYSTEM/var"/>，否则预期的传输将因用户沙箱安全性错误而失败。

示例

此示例显示如何通过将以下 <sandbox> 元素添加到 AGENT_JUPITER 的配置目录中的文件 UserSandboxes.xml ， 允许具有 MQMD 用户名 guest 的用户从运行代理 AGENT_JUPITER 的系统上的 /home/user/public 目录或其任何子目录传输任何文件：

```
<?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>
```

示例

此示例说明了 MQMD 用户名为 account 且后跟单个数字的任何用户（例如，account4）如何完成以下操作：

- 从 /home/account 目录或其任何子目录中传输文件（运行代理 AGENT_SATURN 的系统上的 /home/account/private 目录除外）
- 在运行代理 AGENT_SATURN 的系统上，将任何文件传输到 /home/account/output 目录或其任何子目录。
- 从本地队列管理器上的队列中读取消息，以前缀 ACCOUNT. 开头，除非它以 ACCOUNT.PRIVATE. 开头 (即具有第二个级别的 PRIVATE)。
- 将数据传输到任何队列管理器上以前缀 ACCOUNT.OUTPUT. 开头的队列。

要允许具有 MQMD 用户名 account 的用户完成这些操作，请将以下 <sandbox> 元素添加到 AGENT_SATURN 配置目录中的文件 UserSandboxes.xml：

```
<?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>
```

相关参考

第 513 页的『针对通配符传输的额外检查』
如果已使用用户或代理沙箱配置代理，以限制代理可将文件传输到的位置，您可以指定要对该代理的通配符传输进行其他检查。

MFT agent.properties 文件

针对通配符传输的额外检查

如果已使用用户或代理沙箱配置代理，以限制代理可将文件传输到的位置，您可以指定要对该代理的通配符传输进行其他检查。

additionalWildcardSandboxChecking 属性

要对通配符传输启用额外检查，请向您想要检查的代理的 agent.properties 文件中添加以下属性：

```
additionalWildcardSandboxChecking=true
```

如果将此属性设置为 true，并且代理发出传输请求以尝试读取已定义沙箱外部的用于进行通配符文件匹配的位置，那么传输将失败。 如果一个传输请求中存在多个传输，并且其中一个请求因尝试读取沙箱外部的位置而失败，那么整个传输也将失败。 如果检查失败，将在错误消息中提供失败原因。

如果在代理的 agent.properties 文件中省略了 additionalWildcardSandboxChecking 属性或者将其设置为 false，那么将不会对该代理的通配符传输进行额外检查。

通配符检查的错误消息

向已配置的沙箱位置之外的位置发出通配符传输请求时报告的消息如下所示。

当传输请求中的通配符文件路径位于受限沙箱外部时，将显示以下消息：

```
BFGSS0077E：读取文件路径 path 的尝试遭到拒绝。
该文件路径在受限传输沙箱外部。
```

当多传输请求内的传输包含路径位于受限沙箱外部的通配符传输请求时，将显示以下消息：

```
BFGSS0078E：尝试读取文件路径 path 已作为另一个传输被忽略
受管传输中的项尝试在受限传输沙箱外部读取。
```

当文件位于受限沙箱外部时，将显示以下消息：

```
BFGSS0079E：读取文件 file path 的尝试遭到拒绝。
因为该文件在受限传输沙箱外部。
```

多传输请求中显示以下消息，其中另一个通配符传输请求导致此请求被忽略：

```
BFGSS0080E：尝试读取文件 文件路径 已作为另一个传输被忽略
受管传输中的项尝试在受限传输沙箱外部读取。
```

对于不包含通配符的单文件传输，传输涉及沙箱外部的文件时报告的消息与较早发行版中一样：

```
失败，BFGIO0056E：读取文件"FILE"的尝试遭到拒绝。
因为该文件在受限传输沙箱外部。
```

相关参考

第 511 页的『使用 MFT 用户沙箱』
您可以根据请求传输的 MQMD 用户名来限制可在其中传输文件的文件系统的区域。

第 509 页的『使用 MFT 代理沙箱』
要向 Managed File Transfer 添加额外的安全性级别，可以限制代理可以访问的文件系统区域。

MFT agent.properties 文件

为 MFT 配置 SSL 或 TLS 加密

您可以将 SSL 或 TLS 与 IBM MQ Managed File Transfer 配合使用，以保护代理及其代理队列管理器之间的通信，命令及其连接到的队列管理器之间的通信，以及拓扑中的各种队列管理器到队列管理器连接之间的通信。

开始之前

您可以使用 SSL 或 TLS 加密对流经 IBM MQ Managed File Transfer 拓扑的消息进行加密。 其中包括：

- 在代理与其代理队列管理器之间传递的消息。
- 命令的消息以及它们要连接到的队列管理器。
- 在拓扑中的代理队列管理器，命令队列管理器和协调队列管理器之间流动的内部消息。

关于此任务

有关将 SSL 与 IBM MQ 配合使用的常规信息，请参阅 第 245 页的『使用 SSL/TLS』。 在 IBM MQ 术语中，Managed File Transfer 是标准 Java 客户机应用程序。

遵循以下步骤将 SSL 与 Managed File Transfer 一起使用：

过程

1. 创建信任密钥库文件和（可选）密钥库文件（这些文件可以是相同的文件）。 如果不需要客户机认证（即，通道上 SSLCAUTH=OPTIONAL），那么不需要提供密钥库。 您仅需要信任库来认证队列管理器的证书。

   用于为信任库和密钥库创建证书的密钥算法必须是 RSA 才能使用 IBM MQ。
2. 设置 IBM MQ 队列管理器以使用 SSL。

   有关设置队列管理器以使用 SSL（例如，通过 IBM MQ Explorer 进行设置）的信息，请参阅在队列管理器上配置 SSL。

3. 将信任密钥库文件和密钥库文件（如果有）保存到适当的位置。 建议位置是 *config_directory*/*coordination_qmgr*/agents/*agent_name* 目录。
4. 在适当的 Managed File Transfer 属性文件中根据需要为每个启用 SSL 的队列管理器设置 SSL 属性。 每个属性集引用单个队列管理器（代理、协调和命令），但一个队列管理器可执行其中的两个或更多角色。

   需要 **CipherSpec** 或 **CipherSuite** 属性之一，否则客户机会在不使用 SSL 的情况下尝试连接。 由于 IBM MQ 和 Java 之间的术语差异，提供了 **CipherSpec** 或 **CipherSuite** 属性。 Managed File Transfer 接受任一属性并执行必要的转换，因此不需要设置这两个属性。 如果指定了 **CipherSpec** 和 **CipherSuite** 属性，那么 **CipherSpec** 将优先。

   PeerName 属性是可选的。 您可以将该属性设置为要连接到的队列管理器的专有名称。 Managed File Transfer 将拒绝到具有不匹配专有名称的错误 SSL 服务器的连接。

   将 **SslTrustStore** 和 **SslKeyStore** 属性设置为指向信任密钥库文件和密钥库文件的文件名。 如果要为已在运行的代理设置这些属性，请停止并重新启动代理以采用 SSL 方式重新连接。

   属性文件包含明文密码，因此请考虑设置适当的文件系统许可权。

   有关 SSL 属性的更多信息，请参阅 第 515 页的『MFT 的 SSL/TLS 属性』。
5. 如果代理队列管理器使用 SSL，您不能在创建代理时提供必需的详细信息。 请使用以下步骤创建代理：
   a) 使用 **fteCreateAgent** 命令创建代理。 您将收到有关无法将代理的存在情况发布到协调队列管理器的警告。
   b) 编辑上一步创建的 `agent.properties` 文件，以添加 SSL 信息。 成功启动代理后，重新尝试发布。
6. 如果在运行 IBM MQ Explorer 代理或实例期间更改了 `agent.properties` 文件或 `coordination.properties` 文件中的 SSL 属性，那么必须重新启动代理或 IBM MQ Explorer。

相关参考

MFT `agent.properties` 文件

MFT 的 SSL/TLS 属性

某些 MFT 属性文件包含 SSL 和 TLS 属性。 您可以将 SSL 或 TLS 与 IBM MQ 和 Managed File Transfer 配合使用，以防止代理程序与队列管理器之间存在未经授权的连接，并对代理程序与队列管理器之间的消息流量进行加密。

以下 MFT 属性文件包括 SSL 属性：

- MFT `agent.properties` 文件的 SSL/TLS 属性
- MFT `coordination.properties` 文件的 SSL/TLS 属性
- MFT `command.properties` 文件的 SSL/TLS 属性
- MFT `logger.properties` 文件的 SSL/TLS 属性

有关将 SSL 或 TLS 与 Managed File Transfer 配合使用的信息，请参阅 第 514 页的『为 MFT 配置 SSL 或 TLS 加密』。

从 IBM WebSphere MQ 7.5 开始，可以在表示文件或目录位置的某些 Managed File Transfer 属性中使用环境变量。 这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化，例如哪个用户在运行进程。 有关更多信息，请参阅 在 MFT 属性中使用环境变量。

相关概念

多平台上的 MFT 配置选项

相关参考

在 MFT 属性中使用环境变量

使用通道认证以客户机方式连接到队列管理器

IBM MQ 使用通道认证记录来更精确地控制通道级别的访问。 这意味着缺省情况下，新创建的队列管理器会拒绝来自 Managed File Transfer 组件的客户机连接。

有关通道认证的更多信息，请参阅 第 43 页的『通道认证记录』。

如果 Managed File Transfer 使用的 SVRCONN 的通道认证配置指定了没有特权的 MCAUSER 标识，那么您必须授予队列管理器、队列和主题的特定权限记录，Managed File Transfer Agent 和命令才能正确工作。 使用 MQSC 命令 SET CHLAUTH 或 PCF 命令 Set Channel Authentication Record 来创建、修改或移除通道认证记录。 对于要连接到 IBM MQ 队列管理器的所有 Managed File Transfer 代理程序，可以设置用于所有代理程序的 MCAUSER 标识，也可以为每个代理程序设置单独的 MCAUSER 标识。

向每个 MCAUSER 标识都授予以下许可权：

- 队列管理器所需的权限记录：
  - connect
  - setid
  - inq
- 队列所需的权限记录。

  对于所有特定于代理程序的队列 (即以下列表中以 *agent_name* 结尾的队列名称)，必须使用客户机连接为要连接到 IBM MQ 队列管理器的每个代理程序创建这些队列权限记录。
  - put, get, dsp (SYSTEM.DEFAULT.MODEL.QUEUE)
  - put, get, setid, browse (SYSTEM.FTE.COMMAND.*agent_name*)
  - put, get (SYSTEM.FTE.DATA.*agent_name*)
  - put, get (SYSTEM.FTE.REPLY.*agent_name*)
  - put, get, inq, browse (SYSTEM.FTE.STATE.*agent_name*)
  - put, get, browse (SYSTEM.FTE.EVENT.*agent_name*)
  - put, get (SYSTEM.FTE)
- 主题所需的权限记录：
  - sub, pub (SYSTEM.FTE)
- 文件传输所需的权限记录。

  如果源代理和目标代理具有单独的 MCAUSER 标识，那么请在源和目标的代理队列上都创建权限记录。

  例如，如果源代理的 MCAUSER 标识为 **user1** 并且目标代理 MCAUSER 标识为 **user2**，那么请为代理用户设置以下权限：

代理程序用户	队列	所需权限
user1	SYSTEM.FTE.DATA.*destination_agent_name*	put
user1	SYSTEM.FTE.COMMAND.*destination_agent_name*	put
user2	SYSTEM.FTE.REPLY.*source_agent_name*	put
user2	SYSTEM.FTE.COMMAND.*source_agent_name*	put

在 Connect:Direct 网桥代理与 Connect:Direct 节点之间配置 SSL 或 TLS

通过创建密钥库和信任库并在 Connect:Direct 网桥代理属性文件中设置属性来配置 Connect:Direct 网桥代理和 Connect:Direct 节点，以通过 SSL 协议彼此建立连接。

关于此任务

这些步骤中包含获取由认证中心签名的密钥的指示信息。 如果您不使用认证中心，那么可以生成自签名证书。 有关生成自签名证书的更多信息，请参阅 第 260 页的『在 AIX, Linux, and Windows 上使用 SSL/TLS』。

这些步骤中包含有关为 Connect:Direct 网桥代理创建新的密钥库和信任库的指示信息。 如果 Connect:Direct 网桥代理已具有用于安全连接到 IBM MQ 队列管理器的密钥库和信任库，那么在安全连接到 Connect:Direct 节点时，可以使用现有的密钥库和信任库。 有关更多信息，请参阅 第 514 页的『为 MFT 配置 SSL 或 TLS 加密』。

过程

对于 Connect:Direct 节点，请完成以下步骤：

1. 针对 Connect:Direct 节点生成密钥和签名证书。

   您可以使用 IBM MQ 随附的 IBM 密钥管理工具来执行此操作。 有关更多信息，请参阅第 245 页的『使用 SSL/TLS』。
2. 向认证中心发送请求以获取签名密钥。 您将收到证书。
3. 创建文本文件；例如 `/test/ssl/certs/CAcert`，以包含认证机构公用密钥。
4. 在 Connect:Direct 节点上安装 Secure+ Option。

   如果该节点已存在，那么可以通过再次运行安装程序、指定现有安装的位置以及选择只安装 Secure+ Option 来安装 Secure+ Option。
5. 创建新的文本文件；例如 `/test/ssl/cd/keyCertFile/`*`node_name`*`.txt`。
6. 将从认证中心收到的证书和位于 `/test/ssl/cd/privateKeys/`*`node_name`*`.key` 中的专用密钥复制到该文本文件中。

   `/test/ssl/cd/keyCertFile/`*`node_name`*`.txt` 的内容必须是以下格式：

```
-----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. 启动 Secure+ Admin Tool。

   - 在 AIX and Linux 系统上，运行命令 **spadmin.sh**。
   - 在 Windows 系统上，单击**开始** > **程序** > **Sterling Commerce Connect:Direct** > **CD Secure+ Admin Tool**

   CD Secure+ Admin Tool 将启动。
8. 在 CD Secure+ Admin Tool 中，双击 **.Local** 行以编辑主 SSL 或 TLS 设置。

   a) 根据使用的协议，选择**启用 SSL 协议**或**启用 TLS 协议**。

   b) 选择**禁用覆盖**。

   c) 至少选择一个密码套件。

   d) 如果需要双向认证，请将 **启用客户机认证** 的值更改为 `Yes`。

e) 在**可信根证书**字段中，输入认证中心的公用证书文件的路径 /test/ssl/certs/CAcert。

f) 在**密钥证书文件**字段中，输入您创建的文件的路径，/test/ssl/cd/keyCertFile/*node_name*.txt。

9. 双击 **.Client** 行以编辑主 SSL 或 TLS 设置。

a) 根据使用的协议，选择**启用 SSL 协议**或**启用 TLS 协议**。

b) 选择**禁用覆盖**。

对于 Connect:Direct 网桥代理，请执行以下步骤：

10. 创建信任库。 您可以通过创建虚密钥然后删除该虚密钥的方法来创建。

可以使用以下命令：

```
keytool -genkey -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

```
keytool -delete -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

11. 将认证中心的公用证书导入该信任库。

可以使用以下命令：

```
keytool -import -trustcacerts -alias myCA
        -file /test/ssl/certs/CAcert
        -keystore /test/ssl/fte/stores/truststore.jks
```

12. 编辑 Connect:Direct 网桥代理属性文件。

在该文件的任何位置中包含以下行：

```
cdNodeProtocol=protocol
cdNodeTruststore=/test/ssl/fte/stores/truststore.jks
cdNodeTruststorePassword=password
```

在此步骤的示例中，*protocol* 是您所使用的协议（即 SSL 或 TLS），*password* 是创建信任密钥库时指定的密码。

13. 如果希望双向认证，请为 Connect:Direct 网桥代理创建密钥和证书。

a) 创建密钥库和密钥。

可以使用以下命令：

```
keytool -genkey -keyalg RSA -alias agent_name
        -keystore /test/ssl/fte/stores/keystore.jks
        -storepass password -validity 365
```

b) 生成签名请求。

可以使用以下命令：

```
keytool -certreq -v -alias agent_name
        -keystore /test/ssl/fte/stores/keystore.jks -storepass password
        -file /test/ssl/fte/requests/agent_name.request
```

c) 将上述步骤中收到的证书导入密钥库。 该证书必须为 x.509 格式。

可以使用以下命令：

```
keytool -import -keystore /test/ssl/fte/stores/keystore.jks
        -storepass password -file certificate_file_path
```

d) 编辑 Connect:Direct 网桥代理属性文件。

在该文件的任何位置中包含以下行：

```
cdNodeKeystore=/test/ssl/fte/stores/keystore.jks
cdNodeKeystorePassword=password
```

在此步骤的示例中，*password* 是创建密钥库时指定的密码。

相关任务

配置 Connect:Direct 网桥

ALW 保护 AMQP 客户机安全

您可以使用一系列安全机制来保护来自 AMQP 客户机的连接，并确保数据在网络上受到适当保护。 您可以将安全性构建到 MQ Light 应用程序中。 您还可以将 IBM MQ 的现有安全功能与 AMQP 客户机配合使用，其方式与将这些功能用于其他应用程序的方式相同。

通道认证规则 (CHLAUTH)

您可以使用通道认证规则来限制与队列管理器的 TCP 连接。 AMQP 通道支持使用您为队列管理器配置的通道认证规则。 如果为通道认证规则定义的概要文件与队列管理器上的任何 AMQP 通道匹配，那么这些规则适用于这些通道。 缺省情况下，将在新的 IBM MQ 队列管理器上启用通道认证，因此您必须先完成至少一些配置，然后才能使用 AMQP 通道。

有关如何配置通道认证规则以允许 AMQP 连接到队列管理器的更多信息，请参阅 创建和使用 AMQP 通道。

连接认证 (CONNAUTH)

您可以使用连接认证来认证与队列管理器的连接。 AMQP 通道支持使用连接认证来控制 AMQP 应用程序对队列管理器的访问。

AMQP 协议使用 SASL（简单认证与安全层）框架来指定认证连接的方式。 存在多种 SASL 机制，而 IBM MQ 支持两种 SASL 机制：ANONYMOUS 和 PLAIN。

如果使用 ANONYMOUS，那么客户机不会向队列管理器传递任何凭证来进行认证。 如果在队列管理器 **CONNAUTH** 属性中指定的 IBM MQ AUTHINFO 对象的 **CHCKCLNT** 值为 REQUIRED 或 REQDADM (如果作为管理用户进行连接)，那么将拒绝连接。 如果 **CHCKCLNT** 的值为 NONE 或 OPTIONAL，那么将接受连接。

如果使用 PLAIN，那么客户机会将用户名和密码传递到队列管理器来进行认证。 如果在队列管理器 **CONNAUTH** 属性中指定的 IBM MQ AUTHINFO 对象的 **CHCKCLNT** 值为 NONE，那么将拒绝连接。 如果 **CHCKCLNT** 的值为 OPTIONAL， REQUIRED 或 REQDADM (如果作为管理用户进行连接)，那么队列管理器将检查用户名和密码。 队列管理器将检查操作系统 (如果 AUTHINFO 对象的类型为 IDPWOS) 或 LDAP 存储库 (如果 AUTHINFO 对象的类型为 IDPWLDAP)。

下表总结了此认证行为：

表 *101: SASL* 机制和连接认证的汇总

SASL 机制	**凭证从客户机传递至队列管理器?**	**CHKCLNT 值**
ANONYMOUS	否	REQUIRED 或 REQDADM - 已拒绝连接 NONE 或 OPTIONAL - 已接受连接
PLAIN	是，用户名和密码	REQUIRED、REQDADM 或 OPTIONAL - 由队列管理器检查用户名和密码 NONE - 已拒绝连接

如果您使用的是 MQ Light 客户机，那么可以将凭证包含在要连接到的 AMQP 地址中来指定凭证，例如：

```
amqp://mwhitehead:mYp4ssw0rd@localhost:5672/sports/football
```

通道上的 MCAUSER 设置

AMQP 通道具有 MCAUSER 属性，此属性可用于设置 IBM MQ 用户标识，将使用此用户标识来授权与该通道的所有连接。 从 AMQP 客户机到该通道的所有连接都将采用您所配置的 MCAUSER 标识。 该用户标识用于授权各个主题上的消息传递。

建议使用通道认证 (CHLAUTH) 来保护与队列管理器的连接安全。 如果要使用通道认证，那么建议将 MCAUSER 的值配置为非特权用户。 这样可确保，如果与通道的某个连接与 CHLAUTH 规则不匹配，那么该连接将无权在队列管理器上执行任何消息传递。

SSL/TLS 支持

AMQP 通道支持使用为队列管理器配置的密钥存储库中的密钥来进行 SSL/TLS 加密。 用于 SSL/TLS 加密的 AMQP 通道配置选项与其他类型的 MQ 通道支持相同的选项；您可以指定密码规范以及队列管理器是否需要来自 AMQP 客户机连接的证书。

通过使用队列管理器的 FIPS 属性，您可以控制用于保护来自 AMQP 客户机的连接安全的 SSL/TLS 密码套件。

有关如何为队列管理器设置密钥存储库的信息，请参阅 第 260 页的『在 AIX, Linux, and Windows 上使用 SSL/TLS』。

有关如何为 AMQP 客户机连接配置 SSL/TLS 支持的信息，请参阅 创建和使用 AMQP 通道。

V 9.4.0 V 9.4.0 从 IBM MQ 9.4.0 开始， AMQP 通道不再支持队列管理器上的 CMS 密钥存储库。 您可以使用 **runmqakm** 命令将 CMS 密钥存储库转换为受支持的 PKCS #12 格式。 例如，可以使用以下命令将名为 `sslTest.kdb` 的密钥存储库从 CMS 格式转换为 PKCS #12 格式。 新密钥存储库名为 `sslTest.p12`，并使用密码 `passw0rd` 进行保护。

```
runmqakm -keydb -convert -type cms -db sslTest.kdb -stashed -new_format pkcs12 -target
sslTest.p12 -new_pw passw0rd
```

Java 认证和授权服务 (JAAS)

您可以选择为 AMQP 通道配置 JAAS 登录模块，该模块会对 AMQP 客户机提供的用户名和密码执行检查。 请参阅 第 521 页的『为 AMQP 通道配置 JAAS』。

相关任务

开发 AMQP 客户机应用程序

创建和使用 AMQP 通道

ALW 限制 AMQP 客户机接管

当新建立的 AMQP 客户机连接与现有 AMQP 客户机连接具有相同的客户机标识时，缺省情况下将断开该现有客户机连接。 但是，您可以配置队列管理器以限制客户机接管行为，以便仅在满足特定条件时才可以接管。

例如，如果由不同团队开发了多个 AMQP 应用程序，而这些应用程序碰巧使用相同的客户机标识，那么此情况下就不应该断开现有客户机连接。 要解决此问题，您可以基于所用 AMQP 通道的名称、客户机 IP 地址和客户机用户标识（如果启用了 SASL 认证）来限制客户机接管。

使用队列管理器属性 **AdoptNewMCA** 和 **AdoptNewMCACheck** 的设置来指定所需级别的客户机接管限制，如下表中所述：

表 102: 用于限制客户机接管的 ***AdoptNewMCA*** 和 ***AdoptNewMCACheck*** 设置		
AdoptNewMCA	**AdoptNewMCACheck**	**在允许客户机接管之前检查条件**
NO 或未定义	不适用	无。 经过了认证并通过了所有 CHLAUTH 规则的所有客户机连接都允许客户机接管。
ALL（或非 NO 值）	QM 或未定义	无。 经过了认证并通过了所有 CHLAUTH 规则的所有客户机连接都允许客户机接管。
ALL（或非 NO 值）	名称	用户标识（如果启用了 SASL） 通道名称
ALL（或非 NO 值）	ADDRESS	用户标识（如果启用了 SASL） IP 地址
ALL（或非 NO 值）	所有	用户标识（如果启用了 SASL） 通道名称 IP 地址

队列管理器属性 **AdoptNewMCA** 和 **AdoptNewMCACheck** 是在 CHANNELS 节中定义的队列管理器配置的一部分。 在 IBM MQ for Windows 和 IBM MQ for Linux x86-64 系统上，使用 IBM MQ Explorer 修改配置信息。 在其他系统上，请通过编辑 `qm.ini` 配置文件来修改这些信息。 有关如何修改队列管理器通道信息的信息，请参阅通道属性。

相关任务

开发 AMQP 客户机应用程序

创建和使用 AMQP 通道

ALW 为 AMQP 通道配置 JAAS

Java 认证和授权服务 (JAAS) 定制模块可用于认证 AMQP 客户机在连接时传递到 AMQP 通道的用户名和密码凭证。

关于此任务

如果已在其他基于 Java 的系统中使用 JAAS 模块进行认证，并且要复用这些模块来认证与 MQ 的 AMQP 连接，那么您可能想要使用定制 JAAS 模块。 或者，如果 MQ 中内置的认证功能不支持您想要使用的认证机制，那么您可能希望编写定制的 JAAS 模块。

将在队列管理器级别为 AMQP 通道配置 JAAS 模块。 这意味着，如果您配置了一个 JAAS 模块以用于认证 AMQP 与队列管理器之间的连接，那么该模块将应用于所有 AMQP 通道。 已调用 JAAS 模块的通道的名称会传输到该模块，这样，您便可以为不同通道编写不同的 JAAS 登录行为。

此外，还会向 JAAS 模块传递以下信息：

- 正在尝试认证的 AMQP 客户机的客户机标识。
- AMQP 客户机的网络地址。
- 调用 JAAS 模块的通道的名称。

过程

请完成以下步骤来为 AMQP 通道配置 JAAS 配置模块：

1. 定义包含一个或多个 JAAS 模块配置节的 `jaas.config` 文件。 该节必须指定实现 JAAS `javax.security.auth.spi.LoginModule` 接口的 Java 类的标准名称。

- 产品随附了一个缺省 jaas.config 文件，此文件位于 *QM_data_directory*/amqp/jaas.config 中。
- 已在缺省 jaas.config 文件中定义了名为 MQXRConfig 的预配置节。

2. 指定 AMQP 通道要使用的节的名称。
   - Linux AIX 向 amqp_unix.properties 文件添加一个属性。
   - Windows 向 amqp_win.properties 文件添加一个属性。

   该属性采用以下格式：

   ```
   com.ibm.mq.MQXR.JAASConfig=JAAS_stanza_name
   ```

   例如：

   ```
   com.ibm.mq.MQXR.JAASConfig=MQXRConfig
   ```

3. 配置队列管理器环境以包含定制模块的类。 AMQP 服务必须有权访问 JAAS 配置节中配置的 Java 类。

   可通过将该 JAAS 类的路径添加到 MQ service.env 文件中来实现这一点。 编辑 MQ 配置目录 (*MQ_config_directory*) 或队列管理器配置目录 (*QM_config_directory*) 中的 service.env 文件，以将 CLASSPATH 变量设置为 JAAS 模块类的位置。

下一步做什么

产品随附了一个样本 JAAS 登录模块，此模块位于 *mq_installation_directory*/amqp/samples 目录下。 不论客户机进行连接时所使用的用户名或密码如何，该样本 JAAS 登录模块将认证所有的客户机连接。

您可以修改该样本的源代码并进行重新编译，以尝试仅认证具有特殊密码的特定用户。 要在 UNIX 系统上将 AMQP 通道配置为使用产品随附的样本 JAAS 登录模块：

1. 编辑 /var/mqm/qmgrs/QMNAME/amqp/amqp_unix.properties 文件并设置 com.ibm.mq.MQXR.JAASConfig=MQXRConfig 属性。
2. 编辑 /var/mqm/service.env 文件并设置 CLASSPATH=*mq_installation_location*/amqp/samples 属性

jaas.config 文件已包含名为 MQXRConfig 的节，该节将样本类 samples.JAASLoginModule 指定为登录模块类。 在您尝试使用该样本模块之前，无需对 jaas.config 进行任何更改。

相关任务

开发 AMQP 客户机应用程序

创建和使用 AMQP 通道

Advanced Message Security

Advanced Message Security (AMS) 是 IBM MQ 的组件，可为流经 IBM MQ 网络的敏感数据提供高级别保护，同时不会影响最终应用程序。

Advanced Message Security 概述

IBM MQ 应用程序可以使用 Advanced Message Security 通过使用公用密钥密码术模型，通过不同级别的保护来发送敏感数据，例如高价值金融交易和个人信息。

相关概念

第 568 页的『消息通道代理程序 (MCA) 拦截和 AMS』

MCA 拦截使在 IBM MQ 下运行的队列管理器能够选择性地将策略应用于服务器连接通道。

相关参考

AMS 消息中使用的 GSKit 返回码

Advanced Message Security 的功能部件和功能

Advanced Message Security 扩展 IBM MQ 安全服务以在消息级别提供数据签名和加密。 扩展服务保证在最初将消息数据放在队列上与检索消息数据之间未进行修改。 此外， AMS 还会验证消息数据的发送方是否有权将已签名的消息放在目标队列上。

AMS 提供了以下函数：

- 保护由 IBM MQ 处理的敏感或高价值事务。
- 在接收应用程序处理流氓或未经授权的消息之前，检测并除去这些消息。
- 验证在从队列到队列的传输过程中是否未修改消息。
- 不仅在数据流经网络时保护数据，而且在数据放入队列时保护数据。
- 保护 IBM MQ 的现有专有应用程序和客户编写的应用程序。
- z/OS 从 IBM MQ 9.1.3 开始， IBM MQ for z/OS 提供了分别从网络中流动的消息中移除和添加 AMS 保护的功能 (可选)。 这称为 服务器到服务器消息通道代理程序 *(MCA)* 拦截。
- ALW 从 IBM MQ 9.1.4 和 IBM MQ 9.1.0 Fix Pack 4 开始，将向在客户应用程序中运行的 IBM MQ 库代码添加检查。 此检查在初始化初期运行，以读取环境变量 *AMQ_AMS_FIPS_OFF* 的值，如果设置为任何值，那么将在该应用程序中以非 FIPS 方式运行 IBM Global Security Kit (GSKit) 代码。

AMS 提供的保护质量

Advanced Message Security， `Integrity`， `Privacy` 和 `Confidentiality` 有三种保护质量。

`Integrity` 保护由数字签名提供，这将提供有关创建消息的人员的保证，并且消息未被更改或篡改。

`Privacy` 保护由数字签名和加密的组合提供。 加密可确保消息数据仅可供预期收件人查看。 即使未经授权的收件人获取加密消息数据的副本，他们也无法查看实际消息数据本身。

`Confidentiality` 保护仅通过加密 (可选密钥复用) 提供。

对性能的影响

AMS 使用对称和非对称加密例程的组合来提供数字签名和加密。 与使用 CPU 密集型的非对称密钥操作相比，对称密钥操作非常快，这反过来会对使用 AMS 保护大量消息的成本产生重大影响。

非对称加密例程

例如，在放置已签名的消息时，将使用非对称密钥操作对消息散列进行签名。

获取已签名的消息时，将使用另一个非对称密钥操作来验证已签名的散列。

因此，每条消息至少需要两个非对称密钥操作来签署和验证消息数据。

非对称和对称加密例程

放置加密消息时，将生成对称密钥，然后使用非对称密钥操作对消息的每个预期接收方进行加密。

然后使用对称密钥对消息数据进行加密。 获取加密消息时，预期收件人需要使用非对称密钥操作来发现用于消息的对称密钥。

因此，所有三种保护质量都包含 CPU 密集型非对称密钥操作的不同元素，这将显着影响应用程序放置和获取消息的最大可实现消息传递速率。

但是， `Confidentiality` 策略允许在消息序列上复用对称密钥。 通过对称密钥复用，可以通过 `Confidentiality` 策略节省大量 CPU 成本。 此操作方式继续使用 PKCS#7 格式来共享对称加密密钥。 但是，没有数字签名，这将消除每个消息的一些非对称密钥操作。 对于每个接收方，仍需要使用非对称密钥操作对对称密钥进行加密，但可以选择在发往相同接收方的多条消息上复用对称密钥。 如果策略允许密钥复用，那么仅第一条消息需要非对称密钥操作。 后续消息只需要使用对称密钥操作。

密钥复用

通过 `Confidentiality` 策略，您可以使用对称密钥复用方法，以显着降低对放入同一队列并面向同一收件人的大量消息进行加密所涉及的成本。

例如，将 10 个已加密的消息放入同一组收件人时，将生成对称密钥，然后使用针对消息的每个预期收件人的非对称密钥操作对第一条消息进行加密。

然后，根据策略控制的限制，加密的对称密钥可以由用于相同收件人的后续消息复用。 要允许后续消息复用对称密钥，应用程序必须在将消息放入队列后保持队列处于打开状态。 对称密钥不能由 MQPUT1 操作复用。 获取加密消息的应用程序可以应用相同的优化，因为应用程序可以检测何时未更改对称密钥，并避免检索对称密钥的费用。

在此示例中，放入和获取应用程序都可以通过复用同一密钥来避免 90% 的非对称密钥操作。

有关如何使用密钥复用的更多信息，请参阅:

- MQSC 命令 SET POLICY
- 控制命令 setmqspl
- IBM i IBM i 命令 SETMQMSPL

AMS 中的关键概念

了解 Advanced Message Security 中的关键概念，以了解工具的工作方式以及如何对其进行有效管理。

公用密钥基础结构和 *Advanced Message Security*

公用密钥基础结构 (PKI) 是一种由设施，策略和服务组成的系统，支持使用公用密钥密码术来获取安全通信。

没有单一标准定义公用密钥基础结构的组件，但 PKI 通常涉及公用密钥证书的使用，并包含提供以下服务的认证中心 (CA) 和其他注册中心 (RA):

- 发放数字证书
- 验证数字证书
- 撤销数字证书
- 分发证书

用户和应用程序的身份由与签名或加密消息关联的证书中的 **专有名称 (DN)** 字段表示。 Advanced Message Security 使用此身份来表示用户或应用程序。 要认证此身份，用户或应用程序必须有权访问存储证书和关联专用密钥的密钥库。 每个证书都由密钥库中的一个标签表示。

相关概念

第 562 页的『将密钥库和证书与 AMS 配合使用』
为向 IBM MQ 应用程序提供透明加密保护， Advanced Message Security 使用密钥库文件，其中存储公用密钥证书和专用密钥。 在 z/OS 上，将使用 SAF 密钥环来代替密钥库文件。

AMS 中的数字证书

Advanced Message Security 将用户和应用程序与 X.509 标准数字证书相关联。 X.509 证书通常由可信认证中心 (CA) 签署，涉及用于加密和解密的专用密钥和公用密钥。

数字证书通过将公用密钥绑定到其所有者 (无论该所有者是个人，队列管理器还是其他某个实体) 来提供防止模拟的保护。 数字证书也称为公用密钥证书，因为它们使您在使用非对称密钥方案时能够保证公用密钥的所有权。 此方案要求为应用程序生成公用密钥和专用密钥。 使用公用密钥加密的数据只能使用相应的专用密钥进行解密，而使用专用密钥加密的数据只能使用相应的公用密钥进行解密。 专用密钥存储在受密码保护的密钥数据库文件中。 只有其所有者有权访问用于解密使用相应公用密钥加密的消息的专用密钥。

如果所有者直接将公用密钥发送给另一个实体，那么会存在消息可能被拦截且公用密钥被另一个密钥替代的风险。 这被称为 "中人" 攻击。 解决方案是通过可信的第三方来交换公用密钥，给用户一个强有力的保证，即公用密钥属于您正在与之通信的实体。 而不是直接发送公用密钥，而是要求可信第三方将其合并到数字证书中。 发放数字证书的可信第三方称为认证中心 (CA)。

有关数字证书的更多信息，请参阅 数字证书中的内容。

数字证书包含实体的公用密钥，并声明公用密钥属于该实体:

- 当证书用于单个实体时，它称为 个人证书 或 用户证书。
- 当证书用于认证中心时，该证书称为 *CA* 证书 或 签署者证书。

注: Advanced Message Security 在 Java 和本机应用程序中都支持自签名证书

相关概念

第 10 页的『密码术』

密码术是在称为 *plaintext* 的可读文本与称为 *ciphertext* 的不可读格式之间进行转换的过程。

Multi 对象权限管理器和 *AMS*

在多平台上，对象权限管理器 (OAM) 是随 IBM MQ 产品提供的授权服务组件。

通过 IBM MQ 用户组和 OAM 控制对 Advanced Message Security 实体的访问。 管理员可以根据需要使用命令行界面来授予或撤销权限。 不同用户组可以对相同对象具有不同种类的访问权限。 例如，一个组可以对特定队列执行 PUT 和 GET 操作，而另一个组可能只允许浏览该队列。 同样，某些组可能对队列具有 GET 和 PUT 权限，但不允许更改或删除该队列。

通过 OAM ，您可以控制:

- 通过消息队列接口 (MQI) 访问 Advanced Message Security 对象。 当应用程序尝试访问对象时， OAM 会检查发出请求的用户概要文件是否具有所请求操作的权限。 这意味着可以保护队列以及队列上的消息免受未经授权的访问。
- 允许使用 PCF 和 MQSC 命令。

相关概念

对象权限管理器

消息队列接口概述

Advanced Message Security 支持的技术

Advanced Message Security 依赖于多个技术组件来提供安全基础结构。

Advanced Message Security 支持以下 IBM MQ 应用程序编程接口 (API):

- 消息队列接口 (MQI)
- IBM MQ Java Message Service (JMS) 1.0.2 和 1.1。
- IBM MQ Java 的基类
- 非受管方式下的 .Net 的 IBM MQ 类

注: Advanced Message Security 支持符合 X.509 的认证中心。

AMS 的已知限制

有许多 IBM MQ 选项不受支持，或者具有 Advanced Message Security 的限制。

- 以下 IBM MQ 选项不受支持或具有限制:

  发布/预订

  点对点的发布/预订消息传递模型的主要优点之一是，发送和接收应用程序不需要相互了解要发送和接收的数据。 使用必须定义预期收件人或授权签署者的 Advanced Message Security 策略将否定此权益。 应用程序可以通过受策略保护的别名队列定义发布到主题，预订应用程序也可以从受策略保护的队列中获取消息。 无法将策略直接分配给主题字符串，只能将策略分配给队列定义。

  通道数据转换

  Advanced Message Security 受保护消息的受保护有效内容使用二进制格式进行传输，这将确保应用程序之间的通道上的数据转换不会使消息摘要失效。 从受策略保护的队列中检索消息的应用程序应该请求数据转换，在成功验证消息并且未受保护之后，将尝试转换受保护的有效内容。

  分发列表

  在保护将消息放入分发列表的应用程序时，可以使用 Advanced Message Security 策略，前提是列表中的每个目标队列都定义了相同的策略。 如果在应用程序打开分发列表时标识了不一致的策略，那么打开操作将失败并返回到应用程序的安全性错误。

  应用程序消息分段

  受策略保护的消息的大小将增加，应用程序无法准确指定消息的段边界。

以受管方式使用 IBM MQ classes for .NET 的应用程序 (客户机连接)
不支持以受管方式 (客户机连接) 使用 IBM MQ classes for .NET 的应用程序。

注: MCA 拦截可用于允许不受支持的客户机使用 AMS。

受管方式下 .NET (XMS) 应用程序的消息服务客户机
不支持受管方式下的 .NET (XMS) 应用程序的消息服务客户机。

注: MCA 拦截可用于允许不受支持的客户机使用 AMS。

IMS 网桥处理的 IBM MQ 个队列
IMS 网桥处理的 IBM MQ 队列不受支持。

注: AMS 在 CICS 网桥队列上受支持。 您应该在 CICS 网桥队列上使用相同的用户标识来 MQPUT (加密) 和 MQGET (解密)。

放入等待 getter
针对为其定义了 AMS 策略的队列， getter 应用程序不支持放置到等待 getter。

z/OS **服务器到服务器 MCA 拦截**
从 IBM MQ for z/OS 9.1.3 开始，只有发送方，服务器，接收方和请求者通道类型支持服务器到服务器 MCA 拦截。

- 用户应避免将多个具有相同专有名称的证书放入单个密钥库文件中，因为未定义保护消息时要使用的证书的选项。
- 如果 **WMQ_PROVIDER_VERSION** 属性设置为 6 ，那么 AMS 在 JMS 中不受支持。
- AMQP 或 MQTT 通道不支持 AMS 拦截器。

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.

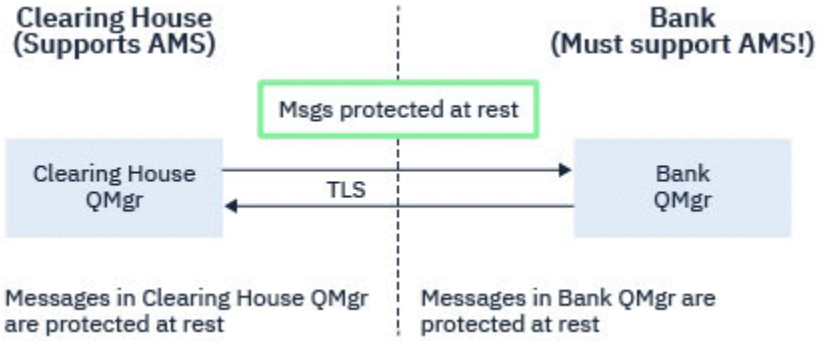


Figure 32. Usage of AMS without AMS interception

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.

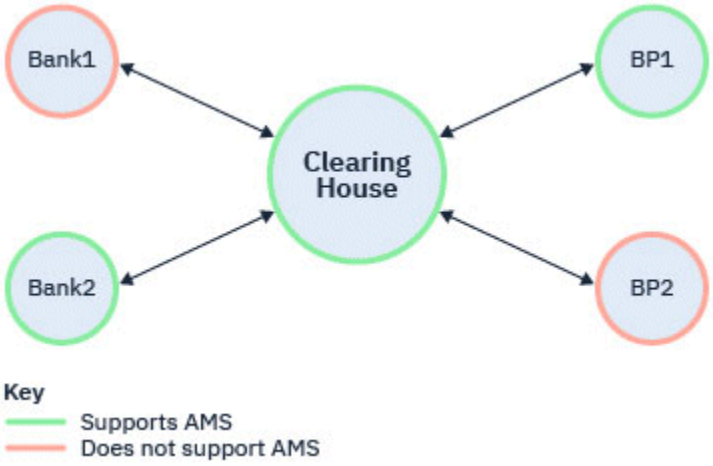


Figure 33. Some partners support AMS and some do not

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

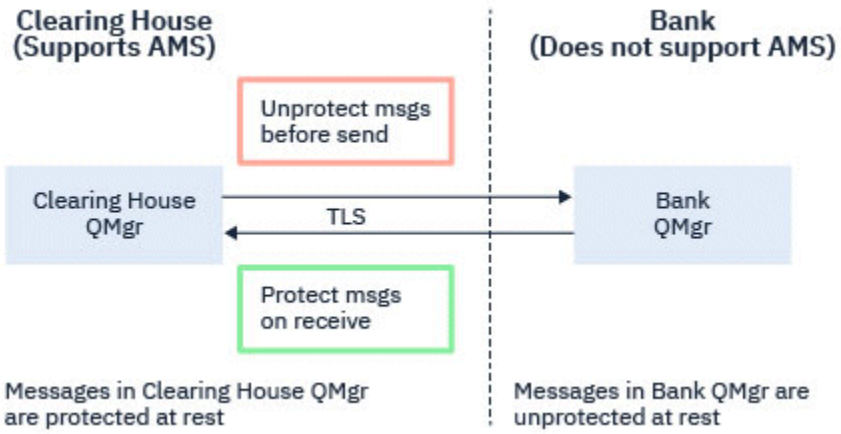


Figure 34. Message flow between business partners

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
按原样传递此通道的消息通道代理发送或接收的任何消息。

对于通道类型 (**CHLTYPE**) 为 SDR、SVR、RCVR 或 RQSTR 的通道，此值有效并且是缺省值。

移除
针对由消息通道代理从传输队列检索到的消息，移除所有 AMS 保护，并将消息发送到合作伙伴。

当消息通道代理从传输队列获取消息时，如果为传输队列定义了 AMS 策略，那么在通道中发送消息之前，将应用该策略以移除对消息的所有 AMS 保护。 如果没有为传输队列定义 AMS 策略，那么按原样发送消息。

该值仅对通道类型为 SDR 或 SVR 的通道有效。

ASPOLICY
根据为目标队列定义的策略，在将入站消息放入目标队列之前，对其应用 AMS 保护。

当消息通道代理接收到入站消息时，如果为目标队列定义了 AMS 策略，那么在将消息放入目标队列之前，将对该消息应用 AMS 保护。 如果没有为目标队列定义 AMS 策略，那么消息按原样放入目标队列。

该值仅对通道类型为 RCVR 或 RQSTR 的通道有效。

User ID for message channel interception

The requirement for user IDs used with server-to-server message channel interception are the same as those for existing AMS enabled applications. For a running channel, the sending message channel agent gets messages from a transmission queue and the receiving message channel agent puts messages to target queues. The message channel agent user ID (MCAUSER) field, set on server to server channels, defines the user ID under which message channel agents perform put and get requests.

With server-to-server message channel interception, AMS functions are performed during get and put requests, as with other AMS enabled applications. Therefore, message channel agent user Ids have the same requirements as those for AMS application user IDs.

The MCAUSER used to perform the put and get is configurable, and dependent on whether it is an outbound or inbound channel. See MCAUSER for details of how the chosen user ID performs actions on the message channel agent. As such, the user ID that the channel initiator is running under is the user ID that is to be used for AMS functions performed during server-to-server message channel interception. Therefore, these user IDs have the same requirements as those for AMS application user IDs.

Authentication is performed using the existing rules for the channel detailed for channels with PUTAUT configuration. See user IDs used by the channel initiator for more information.

Note: Server-to-server message channel interception does not take into account the value of the PUTAUT channel attribute.

Message size and MAXMSGL

Due to AMS protection, the message size of protected messages will be larger than the original message size.

Protected messages are larger than unprotected messages. Therefore, the value of the **MAXMSGL** attribute, on both queues and channels, might need to be altered to take into account the size of protected messages.

Related reference
Server-to-server message channel interception example configurations

AMS 的错误处理

IBM MQ Advanced Message Security 定义了一个错误处理队列，用于管理包含错误或无法不受保护的消息的消息。

有缺陷的消息将作为例外情况处理。 如果接收到的消息不满足其所在队列的安全要求，例如，如果在应加密消息时对该消息进行签名，或者解密或签名验证失败，那么会将该消息发送到错误处理队列。 由于以下原因，可能会将消息发送到错误处理队列:

- 保护质量不匹配-在安全策略中，接收到的消息与 QOP 定义之间存在保护质量不匹配 (QOP)。
- 解密错误-无法解密消息。
- PDMQ 头错误-无法访问 Advanced Message Security (AMS) 消息头。
- 大小不匹配-解密后消息的长度与预期不同。
- 加密算法强度不匹配-消息加密算法弱于所需。
- 未知错误-发生意外错误。

AMS 使用 SYSTEM.PROTECTION.ERROR.QUEUE 作为其错误处理队列。 IBM MQ AMS 放入 SYSTEM.PROTECTION.ERROR.QUEUE 前面有一个 MQDLH 头。

IBM MQ 管理员还可以定义 SYSTEM.PROTECTION.ERROR.QUEUE 作为指向另一个队列的别名队列。

z/OS 在 IBM MQ for z/OS 上，如果正在使用服务器到服务器消息通道代理程序 (MCA) 拦截:

- 如果由于先前声明的其中一个原因， IBM MQ AMS 将消息从传输队列移至错误处理队列，那么发送方 MCA 将继续处理传输队列上的下一条可用消息。
- 通常，现有通道规则适用于:
  - 将消息放入死信队列，
  - 在放入死信队列时执行的操作应失败。

请参阅 第 529 页的『z/OS 上 AMS 的未传递消息』，以获取有关特定方案的更多信息。

z/OS *z/OS 上 AMS 的未传递消息*

与 IBM MQ for z/OS 上的服务器到服务器消息通道代理程序拦截相关的特定方案。

在 IBM MQ for z/OS 上，如果正在使用服务器到服务器消息通道代理程序 (MCA) 拦截:

- 如果在获取和取消保护消息后，发送方 MCA 由于某些原因 (例如，由于消息对于通道过大) 而无法传递消息，那么如果 USEDLQ 发送方通道属性设置为 YES，那么发送方 MCA 会将消息移至本地死信队列 (DLQ)。

  如果是 SYSTEM.DEAD.LETTER.QUEUE 正用作本地 DLQ，消息处于不受保护状态。

  注: IBM MQ AMS 不支持保护放入系统队列的消息。

  如果将指定的 DLQ 用作本地 DLQ，那么如果您定义了与指定的 DLQ 同名的 IBM MQ AMS 策略，那么将对该消息进行保护; 如果您未定义合适的策略，那么将对该消息进行保护。
- 如果由于某种原因无法将消息放入本地 DLQ，那么如果通道的 NPMSPEED 设置为 NORMAL，或者消息是持久消息，那么将回退当前消息批次，并且通道将进入 RETRY 状态。 否则，将废弃该消息，并且发送方 MCA 将继续处理传输队列上的下一条消息。
- 鉴于安全策略对 SYSTEM.DEAD.LETTER.QUEUE，或者 第 594 页的『AMS 中的系统队列保护』中列出的其他 SYSTEM 队列 (如果是 SYSTEM.DEAD.LETTER.QUEUE 正在使用中， MCA 放入此队列的消息将按顺序放置。 即，如果先前对消息进行了保护，那么将对其进行保护; 否则，将对其进行不受保护。

  如果队列管理器 DEADQ 属性已设置为备用 (非系统) 死信队列的名称，并且不存在同名的 AMS 策略，那么 MCA 将按原样放入此队列的消息。 即，如果先前对消息进行了保护，那么将对其进行保护; 否则，将对其进行不受保护。

  如果队列管理器 DEADQ 属性已设置为备用 (非系统) 死信队列的名称，并且存在与 DLQ 同名的 AMS 策略，那么该策略用于保护 MCA 放入此队列的消息。 如果消息先前已受保护，那么不会再次受保护; 这是为了避免双重保护。 如果不存在同名的 AMS 策略，那么将按现有方式放置消息。

- 如果存在将 setmqspl 命令中的容许选项设置为 off 的 DLQ 策略 (即 "**-t** O") ，那么如果消息不受 AMS 保护，因此没有 PDMQ 头，那么放入 DLQ 将失败。 如果消息在没有 PDMQ 头的情况下到达接收方，那么会发生此情况。 这是消息的原始发布程序没有针对目标的策略，并且接收方未设置 SPLPROT (ASPOLICY)。
- 如果为 DLQ 定义的 AMS 策略不允许运行通道启动程序所使用的用户标识来保护消息，那么 MCA 可能无法将消息放入 DLQ。
- 接收方通道通常将未传递的消息放置到本地 DLQ 中，而发送方通道通常将由于某种原因而无法处理的消息放置到本地 DLQ 中，例如，消息对于队列过大，或者 MQXQH 头不正确等。
- DLQ 处理程序通常只查看 DLQ 头 (DLH) ，而不查看消息有效内容本身。 因此，消息有效内容可能受到保护这一事实不会阻止处理程序确定将消息放在 DLQ 上的原因。
- 如果未定义 DLQ ，那么通道:
  - 如果无法传递持久消息，那么异常结束 (并进入重试状态)。
  - 废弃非持久未传递的消息，并继续运行。

相关概念

第 529 页的『AMS 的错误处理』

IBM MQ Advanced Message Security 定义了一个错误处理队列，用于管理包含错误或无法不受保护的消息的消息。

AMS 的用户方案

熟悉可能的场景，以了解您可以使用 Advanced Message Security 实现哪些业务目标。

Windows *Windows 平台上 AMS 的快速入门指南*

使用本指南可快速配置 Advanced Message Security (AMS) 以在 Windows 平台上提供消息安全性。 完成时，您将创建密钥数据库以验证用户身份，并为队列管理器定义签名/加密策略。

开始之前

您应该至少在系统上安装了以下功能部件:

- 服务器
- Development Toolkit (用于样本程序)
- Advanced Message Security (AMS)

有关详细信息，请参阅 IBM MQ Windows 系统功能部件 。

有关使用 **setmqenv** 命令来初始化当前环境以便操作系统可以找到并执行相应的 IBM MQ 命令的信息，请参阅 setmqenv (set IBM MQ environment)。

1. 创建队列管理器和队列

关于此任务

以下所有示例都使用名为 TEST.Q 的队列在应用程序之间传递消息。 Advanced Message Security 使用拦截器在消息通过标准 IBM MQ 接口进入 IBM MQ 基础结构时对消息进行签名和加密。 基本设置在 IBM MQ 中完成，并在以下步骤中进行配置。

您可以使用 IBM MQ Explorer 通过使用所有缺省向导设置来创建队列管理器 QM_VERIFY_AMS 及其名为 TEST.Q 的本地队列，也可以使用在 C:\Program Files\IBM\MQ\bin 中找到的命令。 请记住，您必须是 mqm 用户组的成员才能运行以下管理命令。

过程

1. 创建队列管理器

```
crtmqm QM_VERIFY_AMS
```

2. 启动队列管理器

```
strmqm QM_VERIFY_AMS
```

3. 通过在队列管理器 QM_VERIFY_AMS 的 **runmqsc** 中输入以下命令来创建名为 TEST.Q 的队列

```
DEFINE QLOCAL(TEST.Q)
```

结果

如果过程已完成，那么输入到 **runmqsc** 中的命令将显示有关 TEST.Q 的详细信息:

```
DISPLAY Q(TEST.Q)
```

2. 创建和授权用户

关于此任务

此示例中显示了两个用户: alice(发送方) 和 bob(接收方)。 要使用应用程序队列，需要向这些用户授予使用该队列的权限。 此外，要成功使用我们将定义这些用户的保护策略，必须授予这些用户对某些系统队列的访问权。 有关 **setmqaut** 命令的更多信息，请参阅 **setmqaut**。

过程

1. 创建这两个用户，并确保为这两个用户设置 HOMEPATH 和 HOMEDRIVE 。
2. 授权用户连接到队列管理器并使用队列

```
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. 您还应该允许这两个用户浏览系统策略队列并将消息放入错误队列。

```
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
```

注意: IBM MQ 通过高速缓存策略来优化性能，以便您不必在 SYSTEM.PROTECTION.POLICY.QUEUE 。

IBM MQ 不会高速缓存所有可用的策略。 如果存在大量策略，那么 IBM MQ 会高速缓存有限数量的策略。 因此，如果队列管理器定义的策略数较少，那么无需向 SYSTEM.PROTECTION.POLICY.QUEUE。

但是，如果定义了大量策略，或者如果您使用的是旧客户机，那么应该授予对此队列的浏览权限。 SYSTEM.PROTECTION.ERROR.QUEUE 用于放置由 AMS 代码生成的错误消息。 仅当您尝试将错误消息放入队列时，才会检查对此队列的放置权限。 尝试从受 AMS 保护的队列中放入或获取消息时，不会检查您对该队列的放入权限。

结果

现在将创建用户并向其授予必需的权限。

下一步做什么

要验证是否正确执行了这些步骤，请使用 amqsput 和 amqsget 样本，如 第 534 页的『7. 测试设置』部分中所述。

3. 创建密钥数据库和证书

关于此任务

拦截器需要发送用户的公用密钥来加密消息。 因此，必须创建映射到公用密钥和专用密钥的用户身份的密钥数据库。 在实际系统中，用户和应用程序分散在几台计算机上，每个用户都有自己的专用密钥库。 同样，在本指南中，我们为 alice 和 bob 创建密钥数据库，并在它们之间共享用户证书。

注: 在本指南中，我们使用使用本地绑定连接的 C 语言编写的样本应用程序。 如果计划使用 Java 应用程序 (使用客户机绑定)，那么必须使用 Java **keytool** 命令 V 9.4.0 V 9.4.0 或 IBM MQ **runmqktool** 命令创建 JKS 密钥库和证书。 有关更多信息，请参阅 第 550 页的『AMS with Java 客户机快速入门指南』。 对于所有其他语言以及使用本地绑定的 Java 应用程序，本指南中的步骤是正确的。

过程

1. 为用户 alice 创建新的密钥数据库。
   例如，发出以下命令以创建新的密钥数据库:

   ```
   runmqakm -keydb -create -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -type cms -pw
   passw0rd -stash
   ```

   注:

   - 使用强密码来保护数据库。
   - 包含 **-stash** 参数以将加密的密钥数据库密码隐藏在文件中。
2. 创建新的自签名证书以标识要在加密中使用的用户 alice 。
   例如，发出以下命令以创建新的自签名证书:

   ```
   runmqakm -cert -create -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -stashed
   -label Alice_Cert -dn "CN=alice, O=IBM, C=GB"
   ```

   注:

   - 为了本指南的目的，我们正在使用无需使用认证中心即可创建的自签名证书。 对于生产系统，建议使用由认证中心签署的证书。
   - **-label** 参数指定证书的名称，拦截器将查找该证书以接收必需的信息。
   - **-dn** 参数指定证书的专有名称 (DN) 的详细信息。 专有名称对于每个用户必须唯一。
3. 对用户 bob 重复步骤 第 532 页的『1』 和 第 532 页的『2』 。

结果

现在，这两个用户 alice 和 bob 都具有自签名证书。

4. 创建 *keystore.conf*

关于此任务

必须将 Advanced Message Security 拦截器指向密钥数据库和证书所在的目录。 这是通过 keystore.conf 文件完成的，该文件以纯文本形式保存该信息。 每个用户必须在 .mqs 文件夹中具有单独的 keystore.conf 文件。 必须对 alice 和 bob 执行此步骤。

keystore.conf 的内容必须采用以下格式:

```
cms.keystore = dir/keystore_file
cms.certificate = certificate_label
```

示例

对于此场景， keystore.conf 的内容如下所示:

```
cms.keystore = C:/Documents and Settings/alice/AMS/alicekey
cms.certificate = Alice_Cert
```

注:

- 必须提供没有文件扩展名的密钥库文件的路径。
- 证书标签可以包含空格，因此 "Alice_Cert" 和 "Alice_Cert" 例如， (在末尾有空格) 可识别为两个不同证书的标签。 但是，为了避免混淆，最好不要在标签的名称中使用空格。
- 存在以下密钥库格式: CMS (加密消息语法)， JKS (Java 密钥库) 和 JCEKS (Java 加密扩展密钥库)。 有关更多信息，请参阅第 562 页的『AMS 的密钥库配置文件 (keystore.conf) 的结构』。
- *%HOMEDRIVE%\%HOMEPATH%*\.mqs\keystore.conf (例如， C:\Documents 和 Settings\alice\.mqs\keystore.conf 是 Advanced Message Security 搜索 keystore.conf 文件的缺省位置。 有关如何将非缺省位置用于 keystore.conf 的信息，请参阅 第 562 页的『将密钥库和证书与 AMS 配合使用』。
- 要创建 .mqs 目录，必须使用命令提示符。

5. 共享证书

关于此任务

在两个密钥数据库之间共享证书，以便每个用户都可以成功地识别另一个密钥数据库。 这通过将每个用户的公用证书抽取到一个文件来完成，然后将该文件添加到另一个用户的密钥数据库中。

注: 请注意使用 *extract* 选项，而不是 *export* 选项。 抽取 获取用户的公用密钥，而 导出 获取公用密钥和专用密钥。 错误地使用 *export* 将通过传递其专用密钥来完全损害应用程序。

过程

1. 将标识 alice 的证书解压缩到外部文件:

```
runmqakm -cert -extract -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -pw passw0rd
-label Alice_Cert -target alice_public.arm
```

2. 将证书添加到 bob's 密钥库:

```
runmqakm -cert -add -db "C:\Documents and Settings\bob\AMS\bobkey.kdb" -pw passw0rd -label
Alice_Cert -file alice_public.arm
```

3. 对 bob 重复步骤:

```
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
```

结果

现在，两个用户 alice 和 bob 能够成功地相互标识已创建和共享自签名证书。

下一步做什么

通过使用 GUI 浏览证书或运行以下显示其详细信息的命令，验证证书是否在密钥库中:

```
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. 定义队列策略

关于此任务

通过创建队列管理器和准备拦截消息和访问加密密钥的拦截器，我们可以开始使用 `setmqspl` 命令在 `QM_VERIFY_AMS` 上定义保护策略。 有关此命令的更多信息，请参阅 setmqspl 。 每个策略名称必须与要应用于的队列名称相同。

示例

这是为 `TEST.Q` 队列定义的策略的示例。 在此示例中，使用 Deprecated SHA1 算法对消息进行签名，并使用 AES256 算法进行加密。 `alice` 是唯一有效的发送方， `bob` 是此队列上消息的唯一接收方:

```
setmqspl -m QM_VERIFY_AMS -p TEST.Q -s SHA1 -a "CN=alice,O=IBM,C=GB" -e AES256 -r
"CN=bob,O=IBM,C=GB"
```

注: DN 与密钥数据库中相应用户证书中指定的 DN 完全匹配。

下一步做什么

要验证您定义的策略，请发出以下命令:

```
dspmqspl -m QM_VERIFY_AMS
```

要将策略详细信息打印为一组 `setmqspl` 命令，请使用 `-export` 标志。 这允许存储已定义的策略:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. 测试设置

关于此任务

通过在不同用户下运行不同的程序，您可以验证应用程序是否已正确配置。

过程

1. 切换用户以作为用户 `alice` 运行

   右键单击 `cmd.exe` ，然后选择 **运行方式 ...**。 出现提示时，以用户 `alice` 身份登录。
2. 当用户 `alice` 使用样本应用程序放入消息时:

   ```
   amqsput TEST.Q QM_VERIFY_AMS
   ```
3. 输入消息文本，然后按 Enter 键。
4. 切换用户以作为用户 bob 运行

   通过右键单击 `cmd.exe` 并选择 **运行方式 ...**来打开另一个窗口。 出现提示时，以用户 bob 身份登录。
5. 当用户 bob 使用样本应用程序获取消息时:

   ```
   amqsget TEST.Q QM_VERIFY_AMS
   ```

结果

如果已为这两个用户正确配置应用程序，那么当 bob 运行获取应用程序时，将显示用户 alice 的消息。

8. 测试加密

关于此任务

要验证加密是否按预期进行，请创建引用原始队列 TEST.Q 的别名队列。 此别名队列将没有安全策略，因此没有用户具有解密消息的信息，因此将显示加密数据。

过程

1. 对队列管理器 QM_VERIFY_AMS 使用 **runmqsc** 命令，创建别名队列。

```
DEFINE QALIAS(TEST.ALIAS) TARGET(TEST.Q)
```

2. 授予 bob 从别名队列进行浏览的访问权

```
setmqaut -m QM_VERIFY_AMS -n TEST.ALIAS -t queue -p bob +browse
```

3. 以用户 alice 身份，使用样本应用程序放置另一条消息，就像之前一样:

```
amqsput TEST.Q QM_VERIFY_AMS
```

4. 以用户 bob 身份，这次使用样本应用程序通过别名队列浏览消息:

```
amqsbcg TEST.ALIAS QM_VERIFY_AMS
```

5. 作为用户 bob，使用本地队列中的样本应用程序获取消息:

```
amqsget TEST.Q QM_VERIFY_AMS
```

结果

amqsbcg 应用程序的输出显示队列上的已加密数据，这些数据证明消息已加密。

Linux AIX *AIX and Linux 上 AMS 的快速入门指南*

使用本指南可快速配置 Advanced Message Security 以在 AIX and Linux 上提供消息安全性。 完成时，您将创建密钥数据库以验证用户身份，并为队列管理器定义签名/加密策略。

开始之前

您应该至少在系统上安装以下组件:

- 运行时
- 服务器
- 样本程序
- IBM Global Security Kit (GSKit)
- Advanced Message Security

请参阅以下主题以了解每个特定平台上的组件名称:

- Linux Linux 系统的 IBM MQ 组件
- AIX AIX 系统的 IBM MQ 组件

1. 创建队列管理器和队列

关于此任务

以下所有示例都使用名为 TEST.Q 的队列在应用程序之间传递消息。 Advanced Message Security 使用拦截器在消息通过标准 IBM MQ 接口进入 IBM MQ 基础结构时对消息进行签名和加密。 基本设置在 IBM MQ 中完成，并在以下步骤中进行配置。

您可以使用 IBM MQ Explorer 通过使用所有缺省向导设置来创建队列管理器 QM_VERIFY_AMS 及其名为 TEST.Q 的本地队列，也可以使用在 *MQ_INSTALLATION_PATH*/bin 中找到的命令。 请记住，您必须是 mqm 用户组的成员才能运行以下管理命令。

过程

1. 创建队列管理器

```
crtmqm QM_VERIFY_AMS
```

2. 启动队列管理器

```
strmqm QM_VERIFY_AMS
```

3. 通过在队列管理器 QM_VERIFY_AMS 的 **runmqsc** 中输入以下命令来创建名为 TEST.Q 的队列

```
DEFINE QLOCAL(TEST.Q)
```

结果

如果过程成功完成，那么输入到 **runmqsc** 中的以下命令将显示有关 TEST.Q 的详细信息:

```
DISPLAY Q(TEST.Q)
```

2. 创建和授权用户

关于此任务

此示例中显示了两个用户: alice(发送方) 和 bob(接收方)。 要使用应用程序队列，需要向这些用户授予使用该队列的权限。 此外，要成功使用我们将定义这些用户的保护策略，必须授予这些用户对某些系统队列的访问权。 有关 **setmqaut** 命令的更多信息，请参阅 **setmqaut**。

过程

1. 创建两个用户

```
useradd alice
```

```
useradd bob
```

2. 授权用户连接到队列管理器并使用队列

```
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. 您还应该允许这两个用户浏览系统策略队列并将消息放入错误队列。

```
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
```

注意: IBM MQ 通过高速缓存策略来优化性能，以便您不必在 SYSTEM.PROTECTION.POLICY.QUEUE 。

IBM MQ 不会高速缓存所有可用的策略。 如果存在大量策略，那么 IBM MQ 会高速缓存有限数量的策略。 因此，如果队列管理器定义的策略数较少，那么无需向 SYSTEM.PROTECTION.POLICY.QUEUE。

但是，如果定义了大量策略，或者如果您使用的是旧客户机，那么应该授予对此队列的浏览权限。 SYSTEM.PROTECTION.ERROR.QUEUE 用于放置由 AMS 代码生成的错误消息。 仅当您尝试将错误消息放入队列时，才会检查对此队列的放置权限。 尝试从受 AMS 保护的队列中放入或获取消息时，不会检查您对该队列的放入权限。

结果

现在将创建用户组，并向其授予必需的权限。 这样，分配给这些组的用户也将有权连接到队列管理器并从队列中进行放置和获取。

下一步做什么

要验证是否正确执行了这些步骤，请使用 `amqsput` 和 `amqsget` 样本，如 第 540 页的『8. 测试加密』部分中所述。

3. 创建密钥数据库和证书

关于此任务

要加密消息，拦截器需要发送用户的专用密钥和接收方的公用密钥。 因此，必须创建映射到公用密钥和专用密钥的用户身份的密钥数据库。 在实际系统中，用户和应用程序分散在几台计算机上，每个用户都有自己的专用密钥库。 同样，在本指南中，我们为 `alice` 和 `bob` 创建密钥数据库，并在它们之间共享用户证书。

注: 在本指南中，我们使用使用本地绑定连接的 C 语言编写的样本应用程序。 如果计划使用 Java 应用程序 (使用客户机绑定) ，那么必须使用属于 JRE 的 **keytool** 命令来创建 JKS 密钥库和证书 (请参阅 第 550 页的『AMS with Java 客户机快速入门指南』以获取更多详细信息)。 对于所有其他语言以及使用本地绑定的 Java 应用程序，本指南中的步骤是正确的。

过程

1. 为用户 `alice` 创建新的密钥数据库

```
mkdir /home/alice/.mqs -p
```

```
runmqakm -keydb -create -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -stash
```

注:

- 建议使用强密码来保护数据库。
- **stash** 参数将密码存储到 `key.sth` 文件中，拦截器可用于打开数据库。

2. 确保密钥数据库可读

```
chmod +r /home/alice/.mqs/alicekey.kdb
```

3. 创建用于标识用户 `alice` 以用于加密的证书

```
runmqakm -cert -create -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Alice_Cert -dn
"cn=alice,O=IBM,c=GB" -default_cert yes
```

注:

- 为了本指南的目的，我们正在使用无需使用认证中心即可创建的自签名证书。 对于生产系统，建议不要使用自签名证书，而是依赖于认证中心签署的证书。
- **label** 参数指定证书的名称，拦截器将查找该证书以接收必需的信息。
- **DN** 参数指定 **专有名称** (DN) 的详细信息，对于每个用户必须唯一。

4. 现在我们已经创建了密钥数据库，我们应该设置它的所有权，并确保它不被所有其他用户读取。

```
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. 对用户 bob 重复步骤 1-4

结果

现在，这两个用户 alice 和 bob 都具有自签名证书。

4. 创建 *keystore.conf*

关于此任务

必须将 Advanced Message Security 拦截器指向密钥数据库和证书所在的目录。 这是通过 keystore.conf 文件完成的，该文件以纯文本形式保存该信息。 每个用户必须在 .mqs 文件夹中具有单独的 keystore.conf 文件。 必须对 alice 和 bob 执行此步骤。

keystore.conf 的内容必须采用以下格式:

```
cms.keystore = dir/keystore_file
```

```
cms.certificate = certificate_label
```

示例

对于此场景， keystore.conf 的内容如下所示:

```
cms.keystore = /home/alice/.mqs/alicekey
cms.certificate = Alice_Cert
```

注:

- 必须提供没有文件扩展名的密钥库文件的路径。
- 存在以下密钥库格式: CMS (加密消息语法)， JKS (Java 密钥库) 和 JCEKS (Java 加密扩展密钥库)。 有关更多信息，请参阅第 562 页的『AMS 的密钥库配置文件 (keystore.conf) 的结构』。
- HOME/.mqs/keystore.conf 是 Advanced Message Security 在其中搜索 keystore.conf 文件的缺省位置。 有关如何将非缺省位置用于 keystore.conf 的信息，请参阅 第 562 页的『将密钥库和证书与 AMS 配合使用』。

5. 共享证书

关于此任务

在两个密钥数据库之间共享证书，以便每个用户都可以成功地识别另一个密钥数据库。 这通过将每个用户的公用证书抽取到一个文件来完成，然后将该文件添加到另一个用户的密钥数据库中。

注: 请注意使用 *extract* 选项，而不是 *export* 选项。 抽取 获取用户的公用密钥，而 导出 获取公用密钥和专用密钥。 错误地使用 *export* 将通过传递其专用密钥来完全损害应用程序。

过程

1. 将标识 alice 的证书解压缩到外部文件:

```
runmqakm -cert -extract -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Alice_Cert
-target alice_public.arm
```

2. 将证书添加到 bob's 密钥库:

```
runmqakm -cert -add -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Alice_Cert -file
alice_public.arm
```

3. 对 bob 重复该步骤:

```
runmqakm -cert -extract -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Bob_Cert -target
bob_public.arm
```

4. 将 bob 的证书添加到 alice's 密钥库:

```
runmqakm -cert -add -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Bob_Cert -file
bob_public.arm
```

结果

现在，两个用户 alice 和 bob 能够成功地相互标识已创建和共享自签名证书。

下一步做什么

通过运行以下显示其详细信息的命令，验证证书是否在密钥库中:

```
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. 定义队列策略

关于此任务

通过创建队列管理器和准备拦截消息和访问加密密钥的拦截器，我们可以开始使用 setmqspl 命令在 QM_VERIFY_AMS 上定义保护策略。 有关此命令的更多信息，请参阅 setmqspl 。 每个策略名称必须与要应用于的队列名称相同。

示例

这是为 TEST.Q 队列定义的策略的示例。 在此示例中，消息由用户 alice 使用 Deprecated SHA1 算法进行签名，并使用 256 位 AES 算法进行加密。 alice 是唯一有效的发送方， bob 是此队列上消息的唯一接收方:

```
setmqspl -m QM_VERIFY_AMS -p TEST.Q -s SHA1 -a "CN=alice,O=IBM,C=GB" -e AES256 -r
"CN=bob,O=IBM,C=GB"
```

注: DN 与密钥数据库中相应用户证书中指定的 DN 完全匹配。

下一步做什么

要验证您定义的策略，请发出以下命令:

```
dspmqspl -m QM_VERIFY_AMS
```

要将策略详细信息打印为一组 `setmqspl` 命令，请使用 `-export` 标志。 这允许存储已定义的策略:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. 测试设置

关于此任务

通过在不同用户下运行不同的程序，您可以验证应用程序是否已正确配置。

过程

1. 切换到包含样本的目录。 如果 MQ 安装在非缺省位置，那么这可能位于其他位置。

   ```
   cd /opt/mqm/samp/bin
   ```

2. 切换用户以作为用户 `alice` 运行

   ```
   su alice
   ```

3. 以用户 `alice` 身份，使用样本应用程序放置消息:

   ```
   ./amqsput TEST.Q QM_VERIFY_AMS
   ```

4. 输入消息文本，然后按 Enter 键。
5. 停止以用户 `alice` 身份运行

   ```
   exit
   ```

6. 切换用户以作为用户 `bob` 运行

   ```
   su bob
   ```

7. 作为用户 `bob`，使用样本应用程序获取消息:

   ```
   ./amqsget TEST.Q QM_VERIFY_AMS
   ```

结果

如果已为这两个用户正确配置应用程序，那么当 `bob` 运行获取应用程序时，将显示用户 `alice` 的消息。

8. 测试加密

关于此任务

要验证加密是否按预期进行，请创建引用原始队列 `TEST.Q` 的别名队列。 此别名队列将没有安全策略，因此没有用户具有解密消息的信息，因此将显示加密数据。

过程

1. 对队列管理器 QM_VERIFY_AMS 使用 **runmqsc** 命令，创建别名队列。

```
DEFINE QALIAS(TEST.ALIAS) TARGET(TEST.Q)
```

2. 授予 bob 从别名队列进行浏览的访问权

```
setmqaut -m QM_VERIFY_AMS -n TEST.ALIAS -t queue -p bob +browse
```

3. 以用户 alice 身份，使用样本应用程序放置另一条消息，就像之前一样:

```
./amqsput TEST.Q QM_VERIFY_AMS
```

4. 以用户 bob 身份，这次使用样本应用程序通过别名队列浏览消息:

```
./amqsbcg TEST.ALIAS QM_VERIFY_AMS
```

5. 作为用户 bob，使用本地队列中的样本应用程序获取消息:

```
./amqsget TEST.Q QM_VERIFY_AMS
```

结果

amqsbcg 应用程序的输出将显示队列上的已加密数据，这些数据证明消息已加密。

z/OS *Example AMS configurations on z/OS*

This section provides example configurations of policies and certificates for Advanced Message Security queuing scenarios on z/OS.

See Configuring Advanced Message Security for z/OS for details on how you configure Advanced Message Security.

The examples cover the Advanced Message Security policies required, and the digital certificates that must exist relative to users and key rings. The examples assume that the users involved in the scenarios have been set up by following the instructions provided in Grant users resource permissions for Advanced Message Security.

See also server-to-server message channel interception examples.

z/OS *Local queuing of integrity-protected messages for AMS on z/OS*

This example details the Advanced Message Security policies and certificates needed to send and retrieve integrity-protected messages to and from a queue, local to the putting and getting applications.

The example queue manager and queue are:

```
BNK6        - Queue manager
FIN.XFER.Q7 - Local queue
```

These users are used:

```
WMQBNK6  - AMS task user
TELLER5  - Sending user
FINADM2  - Recipient user
```

Create the user certificates

In this example, only one user certificate is needed. This is the sending user's certificate which is needed to sign integrity-protected messages. The sending user is 'TELLER5'.

The Certificate Authority (CA) certificate is also required. The CA certificate is the certificate of the authority that issued the user's certificate. This can be a chain of certificates. If so, all certificates in the chain are required in the key ring of the Advanced Message Security task user, in this case user WMQBNK6.

A CA certificate can be created using the RACF RACDCERT command. This certificate is used to issue user certificates. For example:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

This RACDCERT command creates a CA certificate which can then be used to issue a user certificate for user 'TELLER5'. For example:

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

Your installation will have procedures for choosing or creating a CA certificate, as well as procedures for issuing certificates and distributing them to relevant systems.

When exporting and importing these certificates, Advanced Message Security requires:

- The CA certificate (chain).
- The user certificate and its private key.

If you are using RACF, the RACDCERT EXPORT command can be used to export certificates to a data set, and the RACDCERT ADD command can be used to import certificates from the data set. For more information about these and other RACDCERT commands, refer to *z/OS: Security Server RACF Command Language Reference*.

The certificates in this case, are required on the z/OS system running queue manager BNK6.

When the certificates have been imported on the z/OS system running BNK6, the user certificate requires the TRUST attribute. The RACDCERT ALTER command can be used to add the TRUST attribute to the certificate. For example:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

In this example, no certificate is required for the recipient user.

Connect certificates to relevant key rings

When the required certificates have been created or imported, and set as trusted, they must be connected to the appropriate user key rings on the z/OS system running BNK6. To create the key rings use the RACDCERT ADDRING commands:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

This creates a key ring for the Advanced Message Security task user, WMQBNK6, and a key ring for the sending user, 'TELLER5'. Note that the key ring name drq.ams.keyring is mandatory, and the name is case-sensitive.

When the key rings have been created, the relevant certificates can be connected:

```
RACDCERT ID(WMQBNK6) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))
```

```
RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

The sending user certificate must be connected as DEFAULT. If the sending user has more than one certificate in its drq.ams.keyring, the default certificate is used for signing purposes.

The creation and modification of certificates is not recognized by Advanced Message Security until the queue manager is stopped and restarted, or the z/OS **MODIFY** command is used to refresh the Advanced Message Security certificate configuration. For example:

```
F BNK6AMSM,REFRESH KEYRING
```

Create the Advanced Message Security policy

In this example, integrity-protected messages are put to queue FIN.XFER.Q7 by an application running as user 'TELLER5', and retrieved from the same queue by an application running as user 'FINADM2', so only one Advanced Message Security policy is required.

Advanced Message Security policies are created using the CSQ0UTIL utility that is documented at The message security policy utility (CSQ0UTIL).

Use the CSQ0UTIL utility to run the following command:

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

In this policy, the queue manager is identified as BNK6. The policy name and associated queue is FIN.XFER.Q7. The algorithm that is used to generate the sender's signature is MD5, and the distinguished name (DN) of the sending user is 'CN=Teller5,O=BCO,C=US'.

After defining the policy, either restart the BNK6 queue manager, or use the z/OS **MODIFY** command to refresh the Advanced Message Security policy configuration. For example:

```
F BNK6AMSM,REFRESH POLICY
```

z/OS *z/OS 上 AMS 的隐私保护消息的本地排队*

此示例详细说明了在放入和获取应用程序的本地队列中发送和检索受隐私保护的消息所需的 Advanced Message Security 策略和证书。 受隐私保护的消息都经过签名和加密。

示例队列管理器和本地队列如下所示:

```
BNK6        - Queue manager
FIN.XFER.Q8 - Local queue
```

使用这些用户:

```
WMQBNK6  - AMS task user
TELLER5  - Sending user
FINADM2  - Recipient user
```

配置此方案的步骤包括:

创建用户证书

在此示例中，需要两个用户证书。 这些是对消息进行签名所需的发送用户证书，以及对消息数据进行加密和解密所需的接收方用户证书。 发送用户为 "TELLER5"，收件人用户为 "FINADM2"。

认证中心 (CA) 证书也是必需的。CA 证书是发放用户证书的权限的证书。 这可以是证书链。 如果是这样，那么在 Advanced Message Security 任务用户的密钥环中需要链中的所有证书，在这种情况下，用户为 WMQBNK6。

可以使用 RACF RACDCERT 命令创建 CA 证书。 此证书用于发放用户证书。 例如:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

此 RACDCERT 命令创建 CA 证书，该证书随后可用于为用户 "TELLER5" 和 "FINADM2" 发放用户证书。 例如：

```
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)
```

您的安装将具有用于选择或创建 CA 证书的过程，以及用于发放证书并将其分发到相关系统的过程。

导出和导入这些证书时， Advanced Message Security 需要：

- CA 证书 (链)。
- 发送用户证书及其专用密钥。
- 接收方用户证书及其专用密钥。

如果您正在使用 RACF，那么可以使用 RACDCERT EXPORT 命令将证书导出到数据集，并且可以使用 RACDCERT ADD 命令从数据集导入证书。 有关这些和其他 RACDCERT 命令的更多信息，请参阅 *z/OS: Security Server RACF Command Language Reference* 中的 RACDCERT (Manage RACF digital 证书) 。

在这种情况下，在运行队列管理器 BNK6 的 z/OS 系统上需要证书。

在运行 BNK6 的 z/OS 系统上导入证书后，用户证书需要 TRUST 属性。 RACDCERT ALTER 命令可用于将 TRUST 属性添加到证书。 例如：

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

将证书连接到相关密钥环

创建或导入所需证书并将其设置为可信证书后，这些证书必须连接到运行 BNK6 的 z/OS 系统上的相应用户密钥环。 要创建密钥环，请使用 RACDCERT ADDRING 命令：

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(FINADM2) ADDRING(drq.ams.keyring)
```

这将为 Advanced Message Security 任务用户创建密钥环，并为发送用户和收件人用户创建密钥环。 请注意，密钥环名称 `drq.ams.keyring` 是必需的，并且名称区分大小写。

创建密钥环后，可以连接相关证书。

```
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))
```

发送用户证书和收件人用户证书必须连接为 DEFAULT。 如果任一用户在其 drq.ams.keyring 中有多个证书，那么缺省证书将用于签名和解密目的。

接收方用户的证书还必须通过 USAGE (SITE) 连接到 Advanced Message Security 任务用户的密钥环。 这是因为 "高级消息安全性" 任务在加密消息数据时需要收件人的公用密钥。 USAGE (SITE) 阻止在密钥环中访问专用密钥。

Advanced Message Security 无法识别证书的创建和修改，直到队列管理器停止并重新启动，或者 z/OS **MODIFY** 命令用于刷新 Advanced Message Security 证书配置。 例如：

```
F BNK6AMSM,REFRESH KEYRING
```

创建 Advanced Message Security 策略

在此示例中，受隐私保护的消息将放入队列 FIN.XFER.Q8 由以用户 "TELLER5" 身份运行的应用程序执行，并由以用户 "FINADM2" 身份运行的应用程序从同一队列中检索，因此仅需要一个 Advanced Message Security 策略。

Advanced Message Security 策略是使用 消息安全策略实用程序 (CSQ0UTIL)中记录的 CSQ0UTIL 实用程序创建的。

使用 CSQ0UTIL 实用程序来运行以下命令：

```
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
```

在此策略中，队列管理器标识为 BNK6。 策略名称和关联队列为 FIN.XFER.Q8。 用于生成发送方签名的算法为 Deprecated SHA1，发送用户的专有名称 (DN) 为 "CN=Teller5,O=BCO,C=US"，接收方用户为 "CN=FinAdm2,O=BCO,C=US"。 用于加密消息数据的算法为 Deprecated 3DES。

定义策略后，请重新启动 BNK6 队列管理器，或者使用 z/OS **MODIFY** 命令来刷新 Advanced Message Security 策略配置。 例如：

```
F BNK6AMSM,REFRESH POLICY
```

z/OS *Remote queuing of integrity-protected messages for AMS on z/OS*

This example details the Advanced Message Security policies and certificates needed to send and retrieve integrity-protected messages to and from queues managed by two different queue managers. The two queue managers can be running on the same z/OS system, or on different z/OS systems, or one queue manager can be on a distributed system running Advanced Message Security.

The example queue managers and queues are:

```
BNK6        - Sending queue manager
BNK7        - Recipient queue manager
FIN.XFER.Q7 - Remote queue on BNK6
FIN.RCPT.Q7 - Local queue on BNK7
```

Note: For this example, BNK6 and BNK7 are queue managers running on different z/OS systems.

These users are used:

```
WMQBNK6  - AMS task user on BNK6
WMQBNK7  - AMStask user on BNK7
TELLER5  - Sending user on BNK6
FINADM2  - Recipient user on BNK7
```

The steps to configure this scenario are as follows:

Create the user certificates

In this example, only one user certificate is needed. This is the sending user's certificate which is needed to sign integrity-protected message. The sending user is 'TELLER5'.

The Certificate Authority (CA) certificate is also required. The CA certificate is the certificate of the authority that issued the user's certificate. This can be a chain of certificates. If so, all certificates in the chain are required in the key ring of the Advanced Message Security task user, in this case user WMQBNK7.

A CA certificate can be created using the RACF RACDCERT command. This certificate is used to issue user certificates. For example:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

This RACDCERT command creates a CA certificate which can then be used to issue user certificate for user 'TELLER5'. For example:

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

Your installation will have procedures for choosing or creating a CA certificate, as well as procedures for issuing certificates and distributing them to relevant systems.

When exporting and importing these certificates, Advanced Message Security require:

- The CA certificate (chain).
- The sending user certificate and its private key.

If you are using RACF, the RACDCERT EXPORT command can be used to export certificates to a data set, and the RACDCERT ADD command can be used to import certificates from the data set. For more information about these and other RACDCERT commands, refer to RACDCERT (Manage RACF digital certificates) in the *z/OS: Security Server RACF Command Language Reference*.

The certificates in this case, are required on the z/OS system running queue manager BNK6 and BNK7.

In this example, the sending certificate must be imported on the z/OS system running BNK6, and the CA certificate must be imported on the z/OS system running BNK7. When the certificates have been imported, the user certificate requires the TRUST attribute. The RACDCERT ALTER command can be used to add the TRUST attribute to the certificate. For example, on BNK6:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

Connect certificates to relevant key rings

When the required certificates have been created or imported, and set as trusted, they must be connected to the appropriate user key rings on the z/OS system running BNK6 and BNK7.

To create the key rings use the RACDCERT ADDRING command, on BNK6:

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

This creates a key ring for the sending user on BNK6. Note that the key ring name drq.ams.keyring is mandatory, and the name is case-sensitive.

On BNK7:

```
RACDCERT ID(WMQBNK7) ADDRING(drq.ams.keyring)
```

This creates a key ring for the Advanced Message Security task user on BNK7. No user key ring is required for 'TELLER5' on BNK7.

When the key rings have been created, the relevant certificates can be connected.

On BNK6:

```
RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

On BNK7:

```
RACDCERT ID(WMQBNK7) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))
```

The sending user certificate must be connected as DEFAULT. If the sending user has more than one certificate in its drq.ams.keyring, the default certificate is used for signing purposes.

The creation and modification of certificates is not recognized by Advanced Message Security until the queue manager is stopped and restarted, or the z/OS **MODIFY** command is used to refresh the Advanced Message Security certificate configuration. For example:

On BNK6:

```
F BNK6AMSM,REFRESH,KEYRING
```

On BNK7:

```
F BNK7AMSM,REFRESH,KEYRING
```

Create the Advanced Message Security policies

In this example, integrity-protected messages are put to remote queue FIN.XFER.Q7 on BNK6 by an application running as user 'TELLER5', and retrieved from local queue FIN.RCPT.Q7 on BNK7 by an application running as user 'FINADM2', so two Advanced Message Security policies are required.

Advanced Message Security policies are created using the CSQ0UTIL utility that is documented at The message security policy utility (CSQ0UTIL).

Use the CSQ0UTIL utility to run the following command to define an integrity policy for the remote queue on BNK6:

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

In this policy, the queue manager is identified as BNK6. The policy name and associated queue is FIN.XFER.Q7. The algorithm that is used to generate the sender's signature is MD5, and the distinguished name (DN) of the sending user is 'CN=Teller5,O=BCO,C=US'.

Also, use the CSQ0UTIL utility to run the following command to define an integrity policy for the local queue on BNK7:

```
setmqspl -m BNK7 -p FIN.RCPT.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

In this policy, the queue manager is identified as BNK7. The policy name and associated queue is FIN.RCPT.Q7. The algorithm expected for the sender's signature is MD5, and the distinguished name (DN) of the sending user is expected to be 'CN=Teller5,O=BCO,C=US'.

After defining the two policies, either restart the BNK6 and BNK7 queue managers, or use the z/OS **MODIFY** command to refresh the Advanced Message Security policy configurations. For example:

On BNK6:

```
F BNK6AMSM,REFRESH,POLICY
```

On BNK7:

```
F BNK7AMSM,REFRESH,POLICY
```

z/OS *z/OS 上 AMS 的隐私保护消息的远程排队*

此示例详细说明了在由两个不同的队列管理器管理的队列之间发送和检索受隐私保护的消息所需的 Advanced Message Security 策略和证书。 这两个队列管理器可以在同一 z/OS 系统上运行，也可以在不同的 z/OS 系统上运行，也可以在运行 Advanced Message Security 的分布式系统上运行一个队列管理器。

示例队列管理器和队列为:

```
BNK6        - Sending queue manager
BNK7        - Recipient queue manager
FIN.XFER.Q7 - Remote queue on BNK6
FIN.RCPT.Q7 - Local queue on BNK7
```

注: 对于此示例， BNK6 和 BNK7 是在同名的不同 z/OS 系统上运行的队列管理器。

使用这些用户:

```
WMQBNK6  - AMS task user on BNK6
WMQBNK7  - AMS task user on BNK7
TELLER5  - Sending user on BNK6
FINADM2  - Recipient user on BNK7
```

配置此方案的步骤如下所示:

创建用户证书

在此示例中，需要两个用户证书。 这些是对消息进行签名所需的发送用户证书，以及对消息数据进行加密和解密所需的接收方用户证书。 发送用户为 "TELLER5"，收件人用户为 "FINADM2"。

认证中心 (CA) 证书也是必需的。CA 证书是发放用户证书的权限的证书。 这可以是证书链。 如果是这样，那么在 Advanced Message Security 任务用户的密钥环中需要链中的所有证书，在本例中为用户 WMQBNK7。

可以使用 RACF RACDCERT 命令创建 CA 证书。 此证书用于发放用户证书。 例如:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

此 RACDCERT 命令创建 CA 证书，该证书随后可用于为用户 "TELLER5" 和 "FINADM2" 发放用户证书。 例如:

```
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)
```

您的安装将具有用于选择或创建 CA 证书的过程，以及用于发放证书并将其分发到相关系统的过程。

导出和导入这些证书时， Advanced Message Security 需要:

- CA 证书 (链)。
- 发送用户证书及其专用密钥。
- 接收方用户证书及其专用密钥。

如果您正在使用 RACF，那么可以使用 RACDCERT EXPORT 命令将证书导出到数据集，并且可以使用 RACDCERT ADD 命令从数据集导入证书。

有关这些和其他 RACDCERT 命令的更多信息，请参阅 *z/OS: Security Server RACF Command Language Reference* 中的 RACDCERT (Manage RACF digital 证书) 。

在这种情况下，在运行队列管理器 BNK6 和 BNK7 的 z/OS 系统上需要证书。

在此示例中，必须在运行 BNK6 的 z/OS 系统上导入发送证书和收件人证书，并且必须在运行 BNK7 的 z/OS 系统上导入 CA 证书和收件人证书。 导入证书后，用户证书需要 TRUST 属性。 RACDCERT ALTER 命令可用于将 TRUST 属性添加到证书。 例如:

在 BNK6:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

在 BNK7:

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

将证书连接到相关密钥环

创建或导入所需证书并将其设置为可信证书后，必须将这些证书连接到运行 BNK6 和 BNK7 的 z/OS 系统上的相应用户密钥环。

要创建密钥环，请使用 RACDCERT ADDRING 命令:

在 BNK6:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

这将为 Advanced Message Security 任务用户创建密钥环，并为 BNK6 上的发送用户创建密钥环。 请注意，密钥环名称 drq.ams.keyring 是必需的，并且名称区分大小写。

在 BNK7:

```
RACDCERT ID(WMQBNK7) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(FINADM2) ADDRING(drq.ams.keyring)
```

这将为 Advanced Message Security 任务用户创建密钥环，并为 BNK7 上的接收方用户创建密钥环。

创建密钥环后，可以连接相关证书。

在 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))
```

在 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))
```

发送用户证书和收件人用户证书必须连接为 DEFAULT。 如果任一用户在其 drq.ams.keyring 中有多个证书，那么缺省证书用于签名和加密/解密目的。

在 BNK6 上，接收方用户的证书还必须通过 USAGE (SITE) 连接到 Advanced Message Security 任务用户的密钥环。 这是因为 "高级消息安全性" 任务在加密消息数据时需要收件人的公用密钥。 USAGE (SITE) 阻止在密钥环中访问专用密钥。

Advanced Message Security 无法识别证书的创建和修改，直到队列管理器停止并重新启动，或者 z/OS **MODIFY** 命令用于刷新 Advanced Message Security 证书配置。 例如：

在 BNK6:

```
F BNK6AMSM,REFRESH,KEYRING
```

在 BNK7:

```
F BNK7AMSM,REFRESH,KEYRING
```

创建 Advanced Message Security 策略

在此示例中，受隐私保护的消息由以用户 "TELLER5" 身份运行的应用程序放入 BNK6 上的远程队列 FIN.XFER.Q7 ，并由以用户 "FINADM2" 身份运行的应用程序从 BNK7 上的本地队列 FIN.RCPT.Q7 中检索，因此需要两个 Advanced Message Security 策略。

Advanced Message Security 策略是使用 消息安全策略实用程序 (CSQ0UTIL)中记录的 CSQ0UTIL 实用程序创建的。

使用 CSQ0UTIL 实用程序运行以下命令为 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
```

在此策略中，队列管理器标识为 BNK6。 策略名称和关联队列为 FIN.XFER.Q7。 用于生成发送方签名的算法为 Deprecated SHA1，发送用户的专有名称 (DN) 为 "CN=Teller5,O=BCO,C=US" ，接收方用户为 "CN=FinAdm2,O=BCO,C=US"。 用于加密消息数据的算法为 Deprecated 3DES。

此外，使用 CSQ0UTIL 实用程序运行以下命令来定义 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
```

在此策略中，队列管理器标识为 BNK7。 策略名称和关联队列为 FIN.RCPT.Q7。 发送方签名所需的算法为 Deprecated SHA1，发送方用户的专有名称 (DN) 预期为 "CN=Teller5,O=BCO,C=US" ，接收方用户为 "CN=FinAdm2,O=BCO,C=US"。 用于解密消息数据的算法为 Deprecated 3DES。

定义这两个策略后，请重新启动 BNK6 和 BNK7 队列管理器，或者使用 z/OS **MODIFY** 命令来刷新 Advanced Message Security 策略配置。 例如：

在 BNK6:

```
F BNK6AMSM,REFRESH,POLICY
```

在 BNK7:

```
F BNK7AMSM,REFRESH,POLICY
```

AMS with Java 客户机快速入门指南

使用本指南来快速配置 Advanced Message Security ，以便为使用客户机绑定进行连接的 Java 应用程序提供消息安全性。 完成时，您将创建密钥库以验证用户身份，并为队列管理器定义签名/加密策略。

开始之前

确保安装了相应的组件，如 第 530 页的『Windows 平台上 AMS 的快速入门指南』 或 第 535 页的『AIX and Linux 上 AMS 的快速入门指南』 中所述。

1. 创建队列管理器和队列

关于此任务

以下所有示例都使用名为 TEST.Q 的队列在应用程序之间传递消息。 Advanced Message Security 使用拦截器在消息通过标准 IBM MQ 接口进入 IBM MQ 基础结构时对消息进行签名和加密。 基本设置在 IBM MQ 中完成，并在以下步骤中进行配置。

过程

1. 创建队列管理器

```
crtmqm QM_VERIFY_AMS
```

2. 启动队列管理器

```
strmqm QM_VERIFY_AMS
```

3. 通过在队列管理器 QM_VERIFY_AMS 的 **runmqsc** 中输入以下命令来创建和启动侦听器

```
DEFINE LISTENER(AMS.LSTR) TRPTYPE(TCP) PORT(1414) CONTROL(QMGR)
```

```
START LISTENER(AMS.LSTR)
```

4. 通过在队列管理器 QM_VERIFY_AMS 的 **runmqsc** 中输入以下命令，为应用程序创建通道以进行连接

```
DEFINE CHANNEL(AMS.SVRCONN) CHLTYPE(SVRCONN)
```

5. 通过在队列管理器 QM_VERIFY_AMS 的 **runmqsc** 中输入以下命令来创建名为 TEST.Q 的队列

```
DEFINE QLOCAL(TEST.Q)
```

结果

如果该过程成功完成，那么输入到 **runmqsc** 中的以下命令将显示有关 TEST.Q 的详细信息:

```
DISPLAY Q(TEST.Q)
```

2. 创建和授权用户

关于此任务

有两个用户出现在此场景中: alice(发送方) 和 bob(接收方)。 要使用应用程序队列，需要向这些用户授予使用该队列的权限。 此外，要成功使用此场景中定义的保护策略，必须向这些用户授予对某些系统队列的访问权。 有关 **setmqaut** 命令的更多信息，请参阅 **setmqaut**。

过程

1. 为您的平台创建两个用户，如 **快速入门指南** (Windows 或 AIX and Linux) 中所述。
2. 授权用户连接到队列管理器并使用队列

```
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. 您还应该允许这两个用户浏览系统策略队列并将消息放入错误队列。

```
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
```

注意: IBM MQ 通过高速缓存策略来优化性能，以便您不必在 SYSTEM.PROTECTION.POLICY.QUEUE 。

IBM MQ 不会高速缓存所有可用的策略。 如果存在大量策略，那么 IBM MQ 会高速缓存有限数量的策略。 因此，如果队列管理器定义的策略数较少，那么无需向 SYSTEM.PROTECTION.POLICY.QUEUE。

但是，如果定义了大量策略，或者如果您使用的是旧客户机，那么应该授予对此队列的浏览权限。 SYSTEM.PROTECTION.ERROR.QUEUE 用于放置由 AMS 代码生成的错误消息。 仅当您尝试将错误消息放入队列时，才会检查对此队列的放置权限。 尝试从受 AMS 保护的队列中放入或获取消息时，不会检查您对该队列的放入权限。

结果

现在将创建用户并向其授予必需的权限。

下一步做什么

要验证是否正确执行了这些步骤，请使用 `JmsProducer` 和 `JmsConsumer` 样本，如 第 554 页的『7. 测试设置』部分中所述。

3. 创建密钥数据库和证书

关于此任务

要加密要拦截器的消息，需要发送用户的公用密钥。 因此，必须创建映射到公用密钥和专用密钥的用户身份的密钥数据库。 在实际系统中，用户和应用程序分散在几台计算机上，每个用户都有自己的专用密钥库。 同样，在本指南中，我们为 `alice` 和 `bob` 创建密钥数据库，并在它们之间共享用户证书。

注: 在本指南中，我们使用使用客户机绑定进行连接的 Java 中编写的样本应用程序。 如果计划使用 Java 应用程序 (使用本地绑定或 C 应用程序)，那么必须使用 **runmqakm** 命令创建 CMS 密钥库和证书。 有关更多信息，请参阅第 530 页的『Windows 平台上 AMS 的快速入门指南』和第 535 页的『AIX and Linux 上 AMS 的快速入门指南』。

过程

1. 创建要在其中创建密钥库的目录，例如 `/home/alice/.mqs`。 您可能希望在 "快速入门指南" 针对您的平台使用的同一目录中创建该目录。 有关更多信息，请参阅第 530 页的『Windows 平台上 AMS 的快速入门指南』和第 535 页的『AIX and Linux 上 AMS 的快速入门指南』。

   注: 在以下步骤中，此目录称为 *keystore-dir*
2. 创建新的密钥库和证书，以标识要在加密中使用的用户 `alice`

   注: **keytool** 命令是 JRE 的一部分。

```
keytool -genkey -alias Alice_Java_Cert -keyalg RSA -keystore keystore-dir/keystore.jks
-storepass passw0rd
-dname "CN=alice, O=IBM, C=GB" -keypass passw0rd
```

   注:

   - 如果 *keystore-dir* 包含空格，那么必须用引号将密钥库的全名括起来
   - 建议使用高强度密码来保护密钥库。
   - 为了本指南的目的，我们正在使用无需使用认证中心即可创建的自签名证书。 对于生产系统，建议不要使用自签名证书，而是依赖于认证中心签署的证书。
   - **alias** 参数指定证书的名称，拦截器将查找该证书以接收必需的信息。
   - **dname** 参数指定 **专有名称** (DN) 的详细信息，对于每个用户必须唯一。

3. 在 AIX and Linux 上，确保密钥库可读

```
chmod +r keystore-dir/keystore.jks
```

4. 对用户 bob 重复 step1-4

结果

现在，这两个用户 `alice` 和 `bob` 都具有自签名证书。

4. 创建 *keystore.conf*

关于此任务

必须将 Advanced Message Security 拦截器指向密钥数据库和证书所在的目录。 这是通过 `keystore.conf` 文件完成的，该文件以纯文本形式保存该信息。 每个用户都必须具有单独的 `keystore.conf` 文件。 应该对 `alice` 和 `bob` 执行此步骤。

示例

对于此场景， `keystore.conf` for `alice` 的内容如下所示:

```
JKS.keystore = keystore-dir/keystore
JKS.certificate = Alice_Java_Cert
JKS.encrypted = no
JKS.keystore_pass = passw0rd
JKS.key_pass = passw0rd
JKS.provider = IBMJCE
```

对于此场景， `keystore.conf` for `bob` 的内容如下所示:

```
JKS.keystore = keystore-dir/keystore
JKS.certificate = Bob_Java_Cert
JKS.encrypted = no
JKS.keystore_pass = passw0rd
JKS.key_pass = passw0rd
JKS.provider = IBMJCE
```

注:

- 必须提供没有文件扩展名的密钥库文件的路径。
- If you already have a `keystore.conf` file because you have followed the instructions in the Quick Start Guide (Windows or AIX and Linux), you can edit the existing file to add these lines.
- 有关更多信息，请参阅第 562 页的『AMS 的密钥库配置文件 (keystore.conf) 的结构』。

5. 共享证书

关于此任务

在两个密钥库之间共享证书，以便每个用户可以成功地识别另一个密钥库。 这是通过抽取每个用户的证书并将其导入其他用户的密钥库来完成的。

要点: 不同的证书管理命令以不同方式使用术语 *extract* 和 *export* 。

- IBM Global Security Kit (GSKit) **runmqakm** 命令使用术语 *extract* 来引用仅从密钥库复制证书公共部分的过程，使用术语 *export* 来引用将证书及其关联的公用密钥和专用密钥从一个密钥库复制到另一个密钥库的过程。
- Java **keytool** 命令 V 9.4.0 V 9.4.0 和 IBM MQ **runmqktool** 命令使用术语 *export* 来引用仅从密钥库复制证书的公共部分的过程。

此区分很重要，因为错误地使用 *export* 可能会通过公开其专用密钥来损害应用程序。 由于区分非常重要，因此 IBM MQ 文档一致地使用这些术语。 由于这些原因，以下过程指示使用 **keytool** 命令中的 `exportcert` 选项 抽取 证书。

过程

1. 抽取标识 alice 的证书。

```
keytool -exportcert -keystore alice-keystore-dir/keystore.jks -storepass passw0rd
-alias Alice_Java_Cert -file alice-keystore-dir/Alice_Java_Cert.cer
```

2. 将标识 alice 的证书导入到 bob 将使用的密钥库中。 出现提示时，指示您将信任此证书。

```
keytool -importcert -file alice-keystore-dir/Alice_Java_Cert.cer -alias Alice_Java_Cert
-keystore bob-keystore-dir/keystore.jks -storepass passw0rd
```

3. 重复 bob 的步骤

结果

现在，两个用户 alice 和 bob 能够成功地相互标识已创建和共享自签名证书。

下一步做什么

通过运行以下显示其详细信息的命令，验证证书是否在密钥库中:

```
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. 定义队列策略

关于此任务

通过创建队列管理器和准备拦截消息和访问加密密钥的拦截器，我们可以开始使用 setmqspl 命令在 QM_VERIFY_AMS 上定义保护策略。 有关此命令的更多信息，请参阅 setmqspl 。 每个策略名称必须与要应用于的队列名称相同。

示例

这是在 TEST.Q 队列上定义的策略的示例，该策略由用户 alice 使用 Deprecated SHA1 算法进行签名，并使用用户 bob 的 256 位 AES 算法进行加密:

```
setmqspl -m QM_VERIFY_AMS -p TEST.Q -s SHA1 -a "CN=alice,O=IBM,C=GB" -e AES256 -r
"CN=bob,O=IBM,C=GB"
```

注: DN 与密钥数据库中相应用户证书中指定的 DN 完全匹配。

下一步做什么

要验证您定义的策略，请发出以下命令:

```
dspmqspl -m QM_VERIFY_AMS
```

要将策略详细信息打印为一组 setmqspl 命令，请使用 -export 标志。 这允许存储已定义的策略:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. 测试设置

开始之前

确保您正在使用的 Java 版本安装了不受限制的 JCE 策略文件。

注: IBM MQ 安装中提供的 Java 版本已具有这些策略文件。 它可以在 *MQ_INSTALLATION_PATH*/java/bin 中找到。

关于此任务

通过在不同用户下运行不同的程序，您可以验证应用程序是否已正确配置。 有关在不同用户下运行程序的更多信息，请参阅 第 530 页的『Windows 平台上 AMS 的快速入门指南』 和 第 535 页的『AIX and Linux 上 AMS 的快速入门指南』。

过程

1. 要运行这些 JMS 样本应用程序，请对您的平台使用 CLASSPATH 设置，如 IBM MQ classes for JMS 使用的环境变量 中所示，以确保包含样本目录。
2. 以用户 alice 身份，使用作为客户机连接的样本应用程序来放置消息:

```
java JmsProducer -m QM_VERIFY_AMS -d TEST.Q -h localhost -p 1414 -l AMS.SVRCONN
```

3. 作为用户 bob，使用作为客户机连接的样本应用程序获取消息:

```
java JmsConsumer -m QM_VERIFY_AMS -d TEST.Q -h localhost -p 1414 -l AMS.SVRCONN
```

结果

如果已为这两个用户正确配置应用程序，那么当 bob 运行获取应用程序时，将显示用户 alice 的消息。

保护 *AMS* 上的远程队列

要完全保护远程队列，必须在要将消息传输到的远程队列和本地队列上设置策略。

将消息放入远程队列时， Advanced Message Security 将拦截操作并根据远程队列的策略集处理消息。 例如，对于加密策略，会先对消息进行加密，然后再将其传递到 IBM MQ 以进行处理。 在 Advanced Message Security 处理了放入远程队列中的消息后， IBM MQ 将其放入关联的传输队列中，并将其转发到目标队列管理器和目标队列。

在本地队列上执行 GET 操作时， Advanced Message Security 会尝试根据本地队列上的策略集对消息进行解码。 要使操作成功，用于对消息进行解密的策略必须与用于对其进行加密的策略相同。 任何差异都将导致消息被拒绝。

如果由于任何原因无法同时设置这两个策略，那么将提供分阶段推出支持。 可以在启用了容错标志的本地队列上设置策略，这指示当尝试从队列中检索消息涉及未设置安全策略的消息时，可以忽略与该队列相关联的策略。 在这种情况下， GET 将尝试解密消息，但将允许传递未加密的消息。 在保护 (并测试) 本地队列之后，可以设置远程队列上的策略。

切记: 完成 Advanced Message Security 转出后，除去容错标志。

相关参考

setmqspl (设置安全策略)

使用 *IBM Integration Bus* 使用 *AMS* 路由受保护消息

Advanced Message Security 可以保护安装了 IBM Integration Bus 或 WebSphere Message Broker 8.0.0.1 (或更高版本) 的基础结构中的消息。 在 IBM Integration Bus 环境中应用安全性之前，您应该了解这两个产品的性质。

关于此任务

Advanced Message Security 提供消息有效内容的端到端安全性。 这意味着只有指定为消息的有效发送方和接收方的参与方才能够生成或接收消息。 这意味着为了保护流经 IBM Integration Bus 的消息，您可以允许 IBM Integration Bus 在不了解其内容的情况下处理消息 (场景 1) 或使其成为能够接收和发送消息的授权用户 (方案 2)。

方案 *1- Integration Bus* 无法查看消息内容

开始之前

您应该已将 IBM Integration Bus 连接到现有队列管理器。 将 *QMgrName* 替换为以下命令中的此现有队列管理器名称。

关于此任务

在此场景中， Alice 将受保护的消息放入输入队列 QIN 中。 根据消息属性 routeTo，消息将路由到 *bob* 的 (QBOB)。[1](QCECIL) 或缺省 (QDEF) 队列。 可以进行路由，因为 Advanced Message Security 仅保护消息有效内容，而不保护其头和属性，这些头和属性仍不受保护，并且可以由 IBM Integration Bus 读取。 Advanced Message Security 仅由 *alice*， *bob* 和 *cecil* 使用。 不必为 IBM Integration Bus 安装或配置该产品。

IBM Integration Bus 从不受保护的别名队列接收受保护的消息，以避免任何解密消息的尝试。 如果它直接使用受保护队列，那么消息将被放入 DEAD LETTER 队列中，因为无法解密。 消息由 IBM Integration Bus 路由，并且未更改地到达目标队列。 因此，它仍由原始作者签名 (*bob* 和 *cecil* 都只接受 *alice* 发送的消息) 并像以前一样受到保护 (只有 *bob* 和 *cecil* 可以读取它)。 IBM Integration Bus 将路由的消息放入不受保护的别名中。 收件人从受保护的输出队列中检索消息，其中 AMS 将透明地解密消息。

过程

1. 配置 爱丽丝， 博布 和 塞西尔 以使用 Advanced Message Security ， 如 **快速入门指南** (Windows 或 AIX) 中所述。

   确保完成以下步骤:

   - 创建和授权用户
   - 创建密钥数据库和证书
   - 创建 keystore.conf
2. 向 *bob* 和 *cecil* 提供 *alice* 的 证书，以便在检查消息上的数字签名时可以由它们识别 *alice* 。

   执行此操作的方法是将标识 *alice* 的证书抽取到外部文件，然后将抽取的证书添加到 *bob* 的 和 *cecil* 的密钥库。 请务必使用 **任务 5 中描述的方法**。 **快速入门指南 (Windows 或 AIX) 中的 共享证书** 。
3. 向 *alice* 提供 *bob* 和 *cecil* 证书，以便 *alice* 可以发送针对 *bob* 和 *cecil* 加密的消息。

   使用上一步中指定的方法执行此操作。
4. 在队列管理器上，定义名为 QIN， QBOB， QCECIL 和 QDEF 的本地队列。

   ```
   DEFINE QLOCAL(QIN)
   ```
5. 将 QIN 队列的安全策略设置为符合条件的配置。 对 QBOB， QCECIL 和 QDEF 队列使用相同的设置。

   ```
   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"
   ```

   此场景假定安全策略，其中 *alice* 是唯一的授权发件人， *bob* 和 *cecil* 是收件人。
6. 分别定义别名队列 AIN， ABOB 和 ACECIL 引用本地队列 QIN， QBOB 和 QCECIL 。

   ```
   DEFINE QALIAS(AIN) TARGET(QIN)
   ```
7. 请验证先前步骤中指定的别名的安全性配置是否不存在; 否则，将其策略设置为 NONE。

   ```
   dspmqspl -m QMgrName -p AIN
   ```
8. 在 IBM Integration Bus 中，创建消息流以根据消息的 routeTo 属性将到达 AIN 别名队列的消息路由到 BOB， CECIL 或 DEF 节点。 要这样做:

[1] 塞西尔

a) 创建名为 IN 的 MQInput 节点，并将 AIN 别名指定为其队列名称。

b) 创建名为 BOB， CECIL 和 DEF 的 MQOutput 节点，并将别名队列 ABOB， ACECIL 和 ADEF 指定为其各自的队列名称。

c) 创建路由节点并将其命名为 TEST。

d) 将 IN 节点连接到 TEST 节点的输入终端。

e) 为 TEST 节点创建 bob 和 cecil 输出终端。

f) 将 bob 输出终端连接到 BOB 节点。

g) 将 cecil 输出终端连接到 CECIL 节点。

h) 将 DEF 节点连接到缺省输出终端。

i) 应用以下规则:

```
$Root/MQRFH2/usr/routeTo/text()="bob"
```

```
$Root/MQRFH2/usr/routeTo/text()="cecil"
```

9. 将消息流部署到 IBM Integration Bus 运行时组件。
10. 以用户 Alice 身份运行会放置一条消息，该消息还包含名为 routeTo 且值为 bob 或 cecil 的消息属性。 运行样本应用程序 **amqsstm** 将允许您执行此操作。

```
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. 以用户 *bob* 身份运行，使用样本应用程序 **amqsget** 从队列 QBOB 检索消息。

结果

当 *alice* 将消息放入 QIN 队列时，该消息受保护。 它由 IBM Integration Bus 从 AIN 别名队列以受保护格式检索。 IBM Integration Bus 决定将读取 routeTo 属性 (作为所有属性) 未加密的消息路由到何处。 IBM Integration Bus 将消息放在相应的不受保护别名上，以避免其进一步保护。 当 *bob* 或 *cecil* 从队列接收到消息时，将解密消息并验证数字签名。

方案 *2- Integration Bus* 可以查看消息内容

关于此任务

在此场景中，允许一组个人将消息发送到 IBM Integration Bus。 另一个组有权接收由 IBM Integration Bus 创建的消息。 无法窃听各方与 IBM Integration Bus 之间的传输。

请记住，仅当打开队列时， IBM Integration Bus 才会读取保护策略和证书，因此您必须在对保护策略进行任何更新后重新装入执行组以使更改生效。

```
mqsireload execution-group-name
```

如果认为 IBM Integration Bus 是允许读取或签署消息有效内容的授权方，那么必须为启动 IBM Integration Bus 服务的用户配置 Advanced Message Security 。 请注意，不一定是同一用户将消息放入/获取到队列中，也不一定是创建和部署 IBM Integration Bus 应用程序的用户。

过程

1. 配置 爱丽丝， 博布， 塞西尔 和 戴夫 以及 IBM Integration Bus 服务用户，以使用 Advanced Message Security ，如 **快速入门指南** (Windows 或 AIX) 中所述。

   确保完成以下步骤:

   - 创建和授权用户
   - 创建密钥数据库和证书
   - 创建 keystore.conf
2. 向 IBM Integration Bus 服务用户提供 *alice*， *bob*， *cecil* 和 *dave* 's 证书。

   要执行此操作，请将标识 *alice*， *bob*， *cecil* 和 *dave* 的每个证书抽取到外部文件，然后将抽取的证书添加到 IBM Integration Bus 密钥库。 请务必使用 **任务 5 中描述的方法**。 **快速入门指南 (Windows 或 AIX) 中的 共享证书** 。
3. 将 IBM Integration Bus 服务用户证书提供给 *alice*， *bob*， *cecil* 和 *dave*。

   使用上一步中指定的方法执行此操作。

   注: *Alice* 和 *bob* 需要 IBM Integration Bus 服务用户证书来正确加密消息。 IBM Integration Bus 服务用户需要 *alice* 的 和 *bob* 的 证书来验证消息的作者。 IBM Integration Bus 服务用户需要 *cecil* 的 和 *dave* 的 证书来加密它们的消息。 *cecil* 和 *dave* 需要 IBM Integration Bus 服务用户证书来验证消息是否来自 IBM Integration Bus。
4. 定义名为 IN 的本地队列，并定义安全策略，将 *alice* 和 *bob* 指定为作者，将 IBM Integration Bus 的服务用户指定为收件人:

   ```
   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. 定义名为 OUT 的本地队列，并使用指定为作者的 IBM Integration Bus 和指定为收件人的 *cecil* 和 *dave* 的服务用户来定义安全策略:

   ```
   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. 在 IBM Integration Bus 中，创建具有 MQInput 和 MQOutput 节点的消息流。 配置 MQInput 节点以使用 IN 队列，配置 MQOutput 节点以使用 OUT 队列。
7. 将消息流部署到 IBM Integration Bus 运行时组件。
8. 以用户 *alice* 或 *bob* 身份运行时，会使用样本应用程序 **amqsput** 将消息放在队列 IN 上。
9. 以用户 *cecil* 或 *dave* 身份运行， OUT 使用样本应用程序 **amqsget** 从队列中检索消息。

结果

alice 或 *bob* 发送到输入队列的消息 IN 已加密，仅允许 IBM Integration Bus 读取该消息。 IBM Integration Bus 仅接受来自 *alice* 和 *bob* 的消息并拒绝任何其他消息。 在将已接受的消息放入输出队列 OUT 之前，将对其进行适当处理，然后使用 *cecil* 的 和 *dave* 的 密钥对其进行签名和加密。 只有 *cecil* 和 *dave* 能够读取它，未由 IBM Integration Bus 签名的消息将被拒绝。

将 *Advanced Message Security* 用于 *Managed File Transfer*

此场景说明如何配置 Advanced Message Security 以提供通过 Managed File Transfer 发送的数据的消息隐私。

开始之前

确保在托管要保护的 Managed File Transfer 所使用的队列的 IBM MQ 安装上安装了 Advanced Message Security 组件。

如果 Managed File Transfer 代理程序以绑定方式进行连接，请确保还在其本地安装上安装了 IBM Global Security Kit (GSKit) 组件。

关于此任务

当两个 Managed File Transfer 代理之间的数据传输中断时，可能机密数据在用于管理传输的底层 IBM MQ 队列上仍不受保护。 此场景说明如何配置和使用 Advanced Message Security 来保护 Managed File Transfer 队列上的此类数据。

在此场景中，我们考虑一个简单的拓扑，该拓扑包含一台具有两个 Managed File Transfer 队列和两个代理程序的机器， AGENT1 和 AGENT2 共享单个队列管理器，如场景 Managed File Transfer 场景中所述。 这两个代理程序在绑定方式或客户机方式下以相同方式进行连接。

1. 创建证书

开始之前

此方案使用简单模型，其中组 FTAGENTS 中的用户 `ftagent` 用于运行 Managed File Transfer Agent 流程。 如果您正在使用自己的用户名和组名，请相应地更改命令。

关于此任务

Advanced Message Security 使用公用密钥密码术对受保护队列上的消息进行签名和/或加密。

注:

- 如果 Managed File Transfer 代理程序以绑定方式运行，那么用于创建 CMS (加密消息语法) 密钥库的命令在适用于您平台的 **快速入门指南** (Windows 或 AIX) 中详细说明。
- 如果 Managed File Transfer 代理程序以客户机方式运行，那么 第 550 页的『AMS with Java 客户机快速入门指南』中详细描述了创建 JKS (Java 密钥库) 所需的命令。

过程

1. Create a self-signed certificate to identify the user `ftagent` as detailed in the appropriate Quick Start Guide.
   使用专有名称 (DN) ，如下所示:

```
CN=ftagent, OU=MFT, O=<organisation>, L=<location>, ST=<state>, C=<country>
```

2. Create a `keystore.conf` file to identify the location of the keystore and the certificate within it as detailed in the appropriate Quick Start Guide.

2. 配置消息保护

关于此任务

您应该使用 **setmqspl** 命令为 AGENT2 使用的数据队列定义安全策略。 在此场景中，将使用同一用户来启动两个代理程序，因此签署者和接收方 DN 相同，并且与我们生成的证书匹配。

过程

1. 关闭 Managed File Transfer 代理程序以准备使用 **fteStopAgent** 命令进行保护。
2. 创建安全策略以保护 SYSTEM.FTE.DATA.AGENT2 队列。

```
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. 确保运行 Managed File Transfer Agent 进程的用户有权浏览系统策略队列并将消息放入错误队列。

```
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. 使用 **fteStartAgent** 命令重新启动 Managed File Transfer 代理程序。
5. 通过使用 **fteListAgents** 命令并验证代理程序是否处于 READY 状态，确认代理程序已成功重新启动。

结果

现在，您可以提交从 AGENT1 到 AGENT2 的传输，并且将在两个代理之间安全地传输文件内容。

Advanced Message Security 安装概述

在各种平台上安装 Advanced Message Security 组件。

过程

- Multi
  在多平台上安装 Advanced Message Security。
- z/OS
  安装 IBM MQ Advanced for z/OS。
- z/OS
  安装 IBM MQ Advanced for z/OS Value Unit Edition。

相关任务

卸载 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 561:

Table 103. CSQ0USMF optional parameters

Parameter	Value	Description
SMFTYPE	nnn	The SMF record type applicable to the audit report. The CSQ0USMF program uses only SMF records that match the SMFTYPE value when generating the report. If you do not specify SMFTYPE, a default value of 180 is used.
M	qmgr	The IBM MQ queue manager name applicable to the audit report. If you do not specify the -M parameter, the audit report will include all audit records for all queue managers represented in the SMFIN data set.

将密钥库和证书与 AMS 配合使用

为向 IBM MQ 应用程序提供透明加密保护， Advanced Message Security 使用密钥库文件，其中存储公用密钥证书和专用密钥。 在 z/OS 上，将使用 SAF 密钥环来代替密钥库文件。

在 Advanced Message Security 中，用户和应用程序由公用密钥基础结构 (PKI) 身份表示。 此类型的身份用于对消息进行签名和加密。 PKI 身份由与已签名和加密消息关联的证书中主题的 **专有名称 (DN)** 字段表示。 要使用户或应用程序对其消息进行加密，他们需要访问存储了证书和关联的专用密钥和公用密钥的密钥库文件。

ALW 在 AIX, Linux, and Windows 上，密钥库的位置在密钥库配置文件中提供，缺省情况下为 `keystore.conf`。 每个 Advanced Message Security 用户都必须具有指向密钥库文件的密钥库配置文件。 Advanced Message Security 接受以下格式的密钥库文件: `.kdb`， `.jceks` 和 `.jks`。

`keystore.conf` 文件的缺省位置为:

- Linux IBM i AIX 在 IBM i 上， AIX and Linux: `$HOME/.mqs/keystore.conf`
- Windows 在 Windows 上： *%HOMEDRIVE%%HOMEPATH%*`\.mqs\keystore.conf`

如果您正在使用指定的密钥库文件名和位置，那么应使用 **MQS_KEYSTORE_CONF** 环境变量指定此名称，如以下示例命令中所示:

- 对于 Java： `java -D`*`MQS_KEYSTORE_CONF=path/filename app_name`*
- 对于 C 客户机和服务器:
  - Linux AIX 在 AIX and Linux 上： `export MQS_KEYSTORE_CONF=`*`path/filename`*
  - Windows 在 Windows 上： `set MQS_KEYSTORE_CONF=`*`path\filename`*

    注: Windows 上的路径可以且应该指定盘符 (如果有多个盘符可用)。

保护 keystore.conf 文件中的敏感信息

为了访问密钥库文件敏感信息 (例如密码) ，必须提供令牌，以便 IBM MQ Advanced Message Security (AMS) 可以访问密钥库并对消息进行签名和加密。

您应该使用 AMS 随附的 **runamscred** 命令来保护密钥库配置文件中包含的敏感信息。 有关如何保护配置文件的详细信息，请参阅 第 578 页的『为配置文件设置 AMS 密码保护』 。

保护密码时，应使用定制的强加密密钥。 为了在运行时访问密码，必须向 AMS 提供此加密密钥。

提供加密密钥文件位置的方法有两种，即:

- `keystore.conf` 文件中的 **amscred.keyfile** 配置属性
- **MQS_AMSCRED_KEYFILE** 环境变量

优先顺序为 **MQS_AMSCRED_KEYFILE**，后跟 **amscred.keyfile**，然后是缺省键。

相关概念

第 587 页的『AMS 中的发送方专有名称』
发送方专有名称 (DN) 标识有权将消息放入队列的用户。 在将消息放入队列之前，发送方使用其证书对消息进行签名。

第 588 页的『AMS 中的收件人专有名称』
接收方专有名称 (DN) 标识有权从队列检索消息的用户。

AMS 的密钥库配置文件 (keystore.conf) 的结构

密钥库配置文件 (`keystore.conf`) 将 Advanced Message Security 指向相应密钥库的位置。

以下每种配置文件类型都具有前缀:

AMSCRED
与密码保护系统相关的参数。

CMS
证书管理系统，配置条目以 `cms.` 为前缀

PKCS#11
公用密钥密码术标准 #11，配置条目以 `pkcs11.` 作为前缀

IBM i **PEM**
隐私增强邮件格式，配置条目以 `pem.` 为前缀

JKS
Java KeyStore，配置条目以以下内容作为前缀: `jks.`

JCEKS
Java 加密 KeyStore，配置条目以以下内容作为前缀: `jceks.`

z/OS MQ Adv. VUE **JCERACFKS**
Java 加密 RACF 密钥环 KeyStore，配置条目以 `jceracfks` 作为前缀。

要点: 从 IBM MQ 9.0 开始，将忽略 `JCEKS.provider` 和 `JKS.provider` 值。 将 Bouncy Castle 提供程序与正在使用的 JRE 提供的任何 JCE/JCE 提供程序结合使用。 有关更多信息，请参阅第 567 页的『对具有 AMS 的非 IBM JRE 的支持』。

密钥库的示例结构:

CMS

```
cms.keystore = /dir/keystore_file
cms.certificate = certificate_label
```

PKCS#11

```
pkcs11.library = dir\cryptoki.dll
pkcs11.certificate = certificatelabel
pkcs11.token = tokenlabel
pkcs11.token_pin = tokenpin
pkcs11.secondary_keystore = dir\signers
pkcs11.encrypted = no
```

IBM i PEM

```
pem.private = /dir/keystore_file_private_key
pem.public = /dir/keystore_file_public_keys
pem.password = password
pem.encrypted = no
```

Java JKS

```
jks.keystore = dir/Keystore
jks.certificate = certificate_label
jks.encrypted = no
jks.keystore_pass = password
jks.key_pass = password
```

Java JCEKS

```
jceks.keystore = dir/Keystore
jceks.certificate = certificate_label
jceks.encrypted = no
jceks.keystore_pass = password
jceks.key_pass = password
```

Java JCERACFKS

```
jceracfks.keystore = safkeyring://user/keyring
jceracfks.certificate = certificate_label
```

Java PKCS#11

```
pkcs11.library = dir\cryptoki.dll
pkcs11.certificate = certificatelabel
pkcs11.token = tokenlabel
pkcs11.token_pin = tokenpin
pkcs11.secondary_keystore = dir\signers
pkcs11.secondary_keystore_pass = password
pkcs11.encrypted = no
```

表 *104:* 每种配置文件类型所需的参数摘要

参数	必需	配置文件类型				
		Java (PKCS#11, JKS , JCEKS 和 JCERACFKS)	**IBM i PEM**	**PKCS#11**	**CMS**	**AMSCRED**
keystore	✓	✓			✓	
IBM i private	✓		IBM i ✓			
IBM i public	✓		IBM i ✓			
IBM i password	✓		IBM i ✓			
library	✓	✓		✓		
certificate	✓	✓		✓	✓	
token	✓	✓		✓		
token_pin	✓	✓		✓		
secondary_keystore	✓	✓		✓		
secondary_keystore_password	✓	✓				
encrypted		✓	IBM i ✓	✓		
keystore_pass	✓	✓				
key_pass		✓				

表 104: 每种配置文件类型所需的参数摘要 (继续)						
		配置文件类型				
参数	必需	Java (PKCS#11, JKS , JCEKS 和 JCERACFKS)	IBM i PEM	PKCS#11	CMS	AMSCRED
provider		✓				
keyfile						✓您

请注意，您可以使用 # 符号添加注释。

配置文件参数定义如下:

keystore
仅 CMS 和 Java 配置。

CMS， JKS 和 JCEKS 配置的密钥库文件的路径。

z/OS MQ Adv. VUE 用于 JCERACFKS 配置的 RACF 密钥环的 URI。

要点:

- 密钥库文件的路径不得包含文件扩展名。
- z/OS MQ Adv. VUE RACF 密钥环的 URI 必须采用以下格式:

```
safkeyring://user/keyring
```

其中:

- *user* 是拥有密钥环的用户标识
- *keyring* 是密钥环名称。

IBM i **private**
仅限 PEM 配置。

包含 PEM 格式的专用密钥和证书的文件的文件名。

IBM i **public**
仅限 PEM 配置。

包含 PEM 格式的可信公用证书的文件的文件名。

IBM i **password**
仅限 PEM 配置。

用于解密加密专用密钥的密码。

您应该使用本机 AMS 密码保护工具来保护此字段; 请参阅 第 566 页的『保护密码』

library
仅 PKCS#11 。

PKCS#11 库的路径名。

certificate
仅 CMS， PKCS#11 和 Java 配置。

证书标签。

token
仅 PKCS#11 。

令牌标签。

token_pin
仅 PKCS#11 。

用于解锁令牌的 PIN。

仅适用于 Java 操作; 您应该使用 Java AMS 密码保护工具来保护此字段; 请参阅 第 566 页的『保护密码』。

仅适用于本机操作; 应使用本机 AMS 密码保护工具来保护此字段; 请参阅 第 566 页的『保护密码』。

secondary_keystore
仅 PKCS#11 。

不带 .kdb 扩展提供的 CMS 密钥库的路径名，其中包含存储在 PKCS #11 令牌上的证书所需的锚点证书(根证书)。 辅助密钥库还可以包含信任链中的中间证书以及隐私安全策略中定义的接收方证书。 此 CMS 密钥库必须随附一个隐藏文件，该文件必须与辅助密钥库位于同一目录中。

对于 Java 环境，需要 JKS 密钥库，并且必须提供 **secondary_keystore_password**。

secondary_keystore_password
仅 Java PKCS#11 。

通过 secondary_keystore 属性提供的 JKS 密钥库的密码。 您应该使用 Java AMS 密码保护工具来保护此字段; 请参阅 第 566 页的『保护密码』。

encrypted

Java ，并且仅从 IBM MQ 9.3.0， PKCS#11 和 IBM i PEM 。

密码的状态。

keystore_pass
仅限 Java 配置。

密钥库文件的密码。

仅适用于 Java 操作。 您应该使用 Java AMS 密码保护工具来保护此字段; 请参阅 第 566 页的『保护密码』。

key_pass
仅限 Java 配置。

用户专用密钥的密码。

仅适用于 Java 操作; 您应该使用 Java AMS 密码保护工具来保护此字段; 请参阅 第 566 页的『保护密码』。

keyfile
提供在保护或解密此配置文件中包含的密码时要使用的初始密钥的位置; 请参阅 第 566 页的『保护密码』

provider
仅限 Java 配置。

实现密钥库证书所需的加密算法的 Java 安全提供程序。

要点: 存储在密钥库中的信息对于使用 IBM MQ 发送的安全数据流至关重要。 安全性管理员在向这些文件分配文件许可权时必须特别注意。

保护密码

您应该保护 keystore.conf 文件中包含的密码和其他敏感信息。 有关更多信息，请参阅 **runamscred**。

keystore.conf 文件的示例:

```
# Native AMS application configuration
cms.keystore = c:\Documents and Settings\Alice\AliceKeystore
cms.certificate = AliceCert

# Java AMS application configuration
jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = no
jceks.keystore_pass = passw0rd
```

```
jceks.key_pass = passw0rd
jceks.provider = IBMJCE
```

相关任务

第 578 页的『为配置文件设置 AMS 密码保护』
将密钥库和专用密钥密码存储为纯文本会带来安全风险，因此 Advanced Message Security 提供了一个可以使用用户密钥对这些密码进行加扰的工具。

对具有 AMS 的非 IBM JRE 的支持

使用非 IBM JRE 运行时， IBM MQ classes for Java 和 IBM MQ classes for JMS 支持 Advanced Message Security 操作。

Advanced Message Security (AMS) 实现 加密消息语法 (CMS)。 CMS 语法用于对任意消息内容进行数字签名，摘要，认证或加密。

从 IBM MQ 9.0 开始， IBM MQ classes for Java 和 IBM MQ classes for JMS 中的 Advanced Message Security 支持使用开放式源代码 Bouncy Castle 软件包来支持 CMS。 这意味着当使用非 IBM JRE 运行时，这些类可以支持 Advanced Message Security 操作。

在 IBM MQ 9.0 之前， Advanced Message Security 在 Java 客户机中的非 IBM JRE 中不受支持。 IBM MQ classes for Java 和 IBM MQ classes for JMS 中的 Advanced Message Security 支持依赖于 Java Cryptography Extensions (JCE) 的 IBM 实现专门提供的 CMS 支持。 由于此限制，仅当使用包含 Java JCE 提供程序的 Java runtime environment (JRE) 时，该功能才可用。

Bouncy Castle JAR 文件的位置和版本编号

支持非 IBM JRE 所需的 Bouncy Castle JAR 文件包含在 IBM MQ classes for Java 和 IBM MQ classes for JMS 安装包中。

使用的 Bouncy Castle JAR 文件是以下文件:

提供程序 JAR 文件，它是 Bouncy Castle 操作的基础。

V 9.4.0 从 IBM MQ 9.4.0 开始，此 JAR 文件称为 `bcprov-jdk18on.jar`。

"PKIX" JAR 文件，其中包含对 Advanced Message Security 所使用的 CMS 操作的支持。

V 9.4.0 从 IBM MQ 9.4.0 开始，此 JAR 文件称为 `bcpkix-jdk18on.jar`。

"util" JAR 文件，其中包含其他 Bouncy Castle JAR 文件所使用的类。

V 9.4.0 从 IBM MQ 9.4.0 开始，此 JAR 文件称为 `bcutil-jdk18on.jar`。

依赖关系

已使用 IBM JRE 和 Oracle JRE 测试 IBM MQ 9.1 和更高版本的类。 它们也可能在任何符合 J2SE-compliant JRE 下成功运行。 但是，您应该注意以下依赖关系:

- 未更改 Advanced Message Security 配置。
- Bouncy Castle 类仅用于 CMS 操作。 所有其他与安全性相关的操作 (例如，密钥库访问，数据实际加密和签名校验和计算) 都使用 JRE 提供的功能。

  要点: 因此，所使用的 JRE 必须包含 JCE 提供程序实现。
- 要使用某些 强 加密算法，您可能需要为 JRE 的 JCE 实现安装 不受限 策略文件。

  请参阅 JRE 文档以获取更多详细信息。
- 如果已启用 Java 安全性:
  - 将 `java.security.SecurityPermissioninsertProvider.BC` 添加到应用程序，以便 Bouncy Castle 类可用作安全提供程序。
  - 将 `java.security.AllPermission` 授予 Bouncy Castle JAR 文件。

    V 9.4.0 从 IBM MQ 9.4.0 开始，这些文件为:

mq_install_dir/java/lib/bcutil-jdk18on.jar
mq_install_dir/java/lib/bcpkix-jdk18on.jar
mq_install_dir/java/lib/bcprov-jdk18on.jar

相关概念

为 IBM MQ classes for JMS 安装哪些内容
针对 Java 的 IBM MQ 类安装的内容

Multi 消息通道代理程序 (MCA) 拦截和 AMS

MCA 拦截使在 IBM MQ 下运行的队列管理器能够选择性地将策略应用于服务器连接通道。

MCA 拦截允许保留在 AMS 外部的客户机仍连接到队列管理器及其要加密和解密的消息。

当无法在客户机上启用 AMS 时， MCA 拦截旨在提供 AMS 功能。 请注意，使用 MCA 拦截和启用了 AMS 的客户机会导致对消息进行双重保护，这可能对接收应用程序造成问题。 有关更多信息，请参阅第 570 页的『在客户机上禁用 Advanced Message Security』。

注: AMQP 或 MQTT 通道不支持 MCA 拦截器。

密钥库配置文件

缺省情况下， MCA 拦截的密钥库配置文件为 `keystore.conf`，并且位于启动队列管理器或侦听器的用户的 HOME 目录路径中的 `.mqs` 目录中。 还可以使用 MQS_KEYSTORE_CONF 环境变量来配置密钥库。 有关配置 AMS 密钥库的更多信息，请参阅第 562 页的『将密钥库和证书与 AMS 配合使用』。

要启用 MCA 拦截，必须提供要在密钥库配置文件中使用的通道的名称。 对于 "MCA 拦截"，只能使用 cms 密钥库类型。

请参阅第 568 页的『AMS 的 MCA 拦截示例』以获取设置 MCA 拦截的示例。

注意: 必须在所选通道上完成客户机认证和加密，例如，通过使用 SSL 和 SSLPEER 或 CHLAUTH TYPE (SSLPEERMAP) 来确保只有授权客户机才能连接和使用此功能。

IBM i

如果您的企业使用 IBM i，并且您选择了商业认证中心 (CA) 来签署证书，那么数字 Certificate Manager 将创建 PEM (隐私增强邮件) 格式的证书请求。 必须将请求转发到所选 CA。

为此，必须使用以下命令为 `channelname` 中指定的通道选择正确的证书:

```
pem.certificate.channel.channelname
```

AMS 的 *MCA* 拦截示例

有关如何设置 AMS MCA 拦截的示例任务。

开始之前

注意: 必须在所选通道上完成客户机认证和加密，例如，通过使用 SSL 和 SSLPEER 或 CHLAUTH TYPE (SSLPEERMAP) 来确保只有授权客户机才能连接和使用此功能。

如果您的企业使用 IBM i，并且您选择了商业认证中心 (CA) 来签署证书，那么数字 Certificate Manager 将创建 PEM (隐私增强邮件) 格式的证书请求。 必须将请求转发到所选 CA。

关于此任务

此任务将指导您完成设置系统以使用 MCA 拦截的过程，然后验证设置。

注: IBM MQ，包含 AMS 拦截器，并在 MQ 客户机和服务器运行时环境中动态启用这些拦截器。

注意:

- 将代码中的 `userID` 替换为您的用户标识。

- 除非在客户机上取消激活 AMS 拦截，否则以下过程在 IBM MQ 中无法正常工作。

过程

1. 通过使用以下命令创建 shell 脚本来创建密钥数据库和证书。

   此外，请更改 **INSTLOC** 和 **KEYSTORELOC** 或运行必需的命令。 请注意，您可能不需要为 bob 创建证书。

```
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. 在两个密钥数据库之间共享证书，以便每个用户都可以成功地识别另一个密钥数据库。

   对于您的企业所使用的平台，请使用 快速入门指南中描述的方法来共享证书，这一点很重要:

   Windows
   任务 5 共享证书

   AIX and Linux
   任务 5 共享证书

   Java 客户机
   任务 5 共享证书

3. 使用以下配置创建 `keystore.conf`:`Keystore.conf location: /home/userID/ssl/ams1/`

```
cms.keystore = /home/userID/ssl/ams1/alicekey
cms.certificate.channel.SYSTEM.DEF.SVRCONN = alice_cert
```

   注意:

   a. 密钥库必须位于队列管理器所在的系统上。

   b. 必须为 `cms.certificate` 指定特定通道以启用 MCA 干预，然后队列管理器对通过该通道连接到具有策略集的队列的应用程序执行 AMS 操作。

4. 创建并启动队列管理器 AMSQMGR1
5. 使用 QMGR 控制下的可用端口号定义 TCP 侦听器。

   例如:

```
DEFINE LISTENER(MY.LISTENER) TRPTYPE(TCP) PORT(14567) CONTROL(QMGR)
```

6. 启动侦听器并验证它是否正确启动。

   例如:

```
START  LISTENER(MY.LISTENER)
DISPLAY LSSTATUS(MY.LISTENER) PORT
```

7. 停止队列管理器。
8. 设置密钥库:

```
export MQS_KEYSTORE_CONF=/home/userID/ssl/ams1/keystore.conf
```

9. 在同一 shell 上启动队列管理器，以便 MQS_KEYSTORE_CONF 环境变量可供队列管理器使用。
10. 设置安全策略并验证:

```
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
```

请参阅 setmqspl 和 dspmqspl 以获取更多信息。

11. 设置 *MQSERVER* 环境变量:

```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/127.0.0.1(14567)'
```

12. 除去安全策略并验证结果:

```
setmqspl -m AMSQMGR1 -p TESTQ -remove
dspmqspl -m AMSQMGR1
```

13. 从 IBM MQ 9.4 安装浏览队列:

```
/opt/mq93/samp/bin/amqsbcg TESTQ AMSQMGR1
```

浏览输出以加密格式显示消息。

14. 设置安全策略并验证结果:

```
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. 从 IBM MQ 9.4 安装运行 **amqsgetc** :

```
/opt/mqm/samp/bin/amqsgetc TESTQ TESTQMGR
```

相关概念

第 562 页的『AMS 的密钥库配置文件 (keystore.conf) 的结构』

密钥库配置文件 (keystore.conf) 将 Advanced Message Security 指向相应密钥库的位置。

相关参考

第 525 页的『AMS 的已知限制』

有许多 IBM MQ 选项不受支持，或者具有 Advanced Message Security 的限制。

在客户机上禁用 Advanced Message Security

如果使用 IBM MQ 客户机从较低版本的产品连接到队列管理器，并且报告了 2085 (MQRC_UNKNOWN_OBJECT_NAME) 错误，那么需要禁用 IBM MQ Advanced Message Security (AMS)。

关于此任务

IBM MQ Advanced Message Security (AMS) 在 IBM MQ 客户机中自动启用，因此缺省情况下，客户机尝试检查队列管理器中对象的安全策略。

如果报告了此错误，那么当您尝试从较低版本的产品连接到队列管理器时，可以禁用 AMS，如下所示:

- 对于 Java 客户机，请通过以下任一方式:
  - 通过设置环境变量 **AMQ_DISABLE_CLIENT_AMS**。
  - 通过设置 Java 系统属性 com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS。
  - 通过使用 **DisableClientAMS** 属性，在 mqclient.ini 文件中的 Security 节下。
- 对于 C 客户机，通过设置环境变量 **MQS_DISABLE_ALL_INTERCEPT**。

注: 不能将 **AMQ_DISABLE_CLIENT_AMS** 环境变量用于 C 客户机。 您需要改为使用 **MQS_DISABLE_ALL_INTERCEPT** 环境变量。

过程

- 要在客户机上禁用 AMS，请使用下列其中一个选项:

AMQ_DISABLE_CLIENT_AMS 环境变量

您需要在以下情况下设置此变量:

- 如果使用的是除 IBM Java runtime environment (JRE) 以外的 Java runtime environment (JRE)
- 如果您正在使用 IBM MQ IBM MQ classes for JMS 或 IBM MQ classes for Java 客户机。

在运行应用程序的环境中创建 **AMQ_DISABLE_CLIENT_AMS** 环境变量并将其设置为 TRUE 。 例如:

```
export AMQ_DISABLE_CLIENT_AMS=TRUE
```

Java 系统属性 com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS

对于 IBM MQ classes for JMS 和 IBM MQ classes for Java 客户机，可以将 Java 系统属性 com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS 设置为 Java 应用程序的值 TRUE 。

例如，可以在调用 Java 命令时将 Java 系统属性设置为 -D 选项:

```
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
```

或者，如果应用程序使用此文件，那么可以在 JMS 配置文件 jms.config 中指定 Java 系统属性。

MQS_DISABLE_ALL_INTERCEPT 环境变量

如果将 IBM MQ 与本机客户机配合使用，并且需要在客户机上禁用 AMS ，那么需要设置此环境变量。

在运行客户机的环境中创建环境变量 **MQS_DISABLE_ALL_INTERCEPT** 并将其设置为 TRUE 。 例如:

```
export MQS_DISABLE_ALL_INTERCEPT =TRUE
```

只能将 **MQS_DISABLE_ALL_INTERCEPT** 环境变量用于 C 客户机。 对于 Java 客户机，需要改为使用 **AMQ_DISABLE_CLIENT_AMS** 环境变量。

mqclient.ini 文件中的 DisableClientAMS 属性

您可以将此选项用于 IBM MQ classes for JMS 和 IBM MQ classes for Java 客户机以及 C 客户机。

在 mqclient.ini 文件的 **Security** 节下添加属性名 DisableClientAMS ，如以下示例中所示:

```
Security:
DisableClientAMS=Yes
```

您还可以启用 AMS ，如以下示例中所示:

```
Security:
DisableClientAMS=No
```

下一步做什么

有关打开 AMS 受保护队列的问题的更多信息，请参阅 将 AMS 与 JMS 配合使用时打开受保护队列的问题。

相关概念

第 568 页的『消息通道代理程序 (MCA) 拦截和 AMS』

MCA 拦截使在 IBM MQ 下运行的队列管理器能够选择性地将策略应用于服务器连接通道。

相关任务

IBM MQ MQI client 配置文件， mqclient.ini

相关参考

IBM MQ classes for JMS 配置文件

AMS 的证书需求

证书必须具有 RSA 公用密钥才能与 Advanced Message Security 配合使用。

有关不同公用密钥类型以及如何创建它们的更多信息，请参阅 第 39 页的『IBM MQ 中的数字证书和 CipherSpec 兼容性』。

密钥用法扩展

密钥使用扩展对证书的使用方式施加了其他限制。

在 Advanced Message Security 中，必须根据 RFC 5280 规范设置 X.509 v3 证书的密钥用法。

对于保护完整性的质量，如果设置了证书密钥使用扩展，那么该集合必须至少包含以下两项中的一项:

- **nonRepudiation**
- **digitalSignature**

为了保护隐私的质量，如果设置了证书密钥使用扩展，那么该集合必须包括:

- **keyEncipherment**

对于保护机密性的质量，如果设置了证书密钥使用扩展，那么该集合必须包括:

- **dataEncipherment**

扩展密钥使用进一步优化密钥使用扩展。 对于所有保护质量，如果设置了证书扩展密钥用法，那么该集合必须包括:

- **emailProtection**

相关概念
第 589 页的『AMS 中的保护质量』
Advanced Message Security 数据保护策略意味着保护质量 (QOP)。

AMS 中的证书验证方法

您可以使用 Advanced Message Security 来检测和拒绝已撤销的证书，以便使用不符合安全标准的证书来保护队列上的消息。

AMS 允许您使用联机证书状态协议 (OCSP) 或证书撤销列表 (CRL) 来验证证书有效性。

可以为 OCSP 和/或 CRL 检查配置 AMS 。 如果同时启用了这两种方法，那么出于性能原因， AMS 会首先将 OCSP 用于撤销状态。 如果在 OCSP 检查后未确定证书的撤销状态，那么 AMS 将使用 CRL 检查。

请注意，缺省情况下已启用 OCSP 和 CRL 检查。

相关概念
第 572 页的『AMS 中的联机证书状态协议 (OCSP)』
联机证书状态协议 (OCSP) 确定是否已撤销证书，因此有助于确定该证书是否可信。 缺省情况下，已启用 OCSP。

第 574 页的『AMS 中的证书撤销列表 (CRL)』
CRL 包含由认证中心 (CA) 标记为由于各种原因不再可信的证书列表，例如，专用密钥已丢失或已损坏。

AMS 中的联机证书状态协议 *(OCSP)*

联机证书状态协议 (OCSP) 确定是否已撤销证书，因此有助于确定该证书是否可信。 缺省情况下，已启用 OCSP。

在 IBM i sytems 上不支持 OCSP。

在 *Advanced Message Security* 的本机拦截器中启用 *OCSP* 检查

缺省情况下，将根据所使用的证书中的信息启用联机证书状态协议 (OCSP) 检入 Advanced Message Security 。

过程

将以下选项添加到密钥库配置文件中：

注: 所有 OCSP 节都是可选的，可以单独指定。

选项	描述
`ocsp.enable=off`	如果要检查的证书具有带有 PKIX_AD_OCSP 访问方法 (其中包含 OCSP 响应程序所在的 URI) 的 "权限信息访问" (AIA) 扩展，请启用 OCSP 检查。 可能的值: `on` 或 `off`。
`ocsp.url=`*`responder_URL`*	OCSP 响应程序的 URL 地址。 如果省略此选项，那么将禁用非 AIA OCSP 检查。
`ocsp.http.proxy.host=`*`OCSP_proxy`*	OCSP 代理服务器的 URL 地址。 如果省略此选项，那么不会将代理用于非 AIA 联机证书检查。
`ocsp.http.proxy.port=`*`port_number`*	OCSP 代理服务器的端口号。 如果省略此选项，那么将使用缺省端口 8080。
`ocsp.nonce.generation=on/off`	查询 OCSP 时生成现时标志。 缺省值为 `off`。
`ocsp.nonce.check=on/off`	从 OCSP 收到响应后检查现时标志。 缺省值为 `off`。
`ocsp.nonce.size=8`	现时标志大小（以字节计）。
`ocsp.http.get=on/off`	指定 HTTP GET 作为请求方法。 如果将该选项设置为 `off`，那么将使用 HTTP POST。 缺省值为 `off`。
`ocsp.max_response_size=20480`	来自所提供 OCSP 响应程序的响应的最大大小（以字节计）。
`ocsp.cache_size=100`	启用内部 OCSP 响应高速缓存并设置高速缓存条目数目限制。
`ocsp.timeout=30`	Advanced Message Security 发生超时前等待服务器响应的时间（以秒计）。
`ocsp.unknown=ACCEPT`	定义在超时时间段内无法访问 OCSP 服务器时的行为。 可能的值为: • `ACCEPT` 允许证书 • `WARN` 允许证书并记录警告 • `REJECT` 阻止使用证书并记录错误

在 *AMS* 中启用 *OCSP* 检入 *Java*
要对 Advanced Message Security 中的 Java 启用 OCSP 检查，请修改 `java.security` 文件或密钥库配置文件。

关于此任务

在 Advanced Message Security 中启用 OCSP 检查有两种方法:

使用 *java.security*
检查您的证书是否包含 "权限信息访问" (AIA) 证书扩展。

过程

1. 如果未设置 AIA，或者您想要覆盖证书，请使用以下属性编辑 $JAVA_HOME/lib/security/java.security 文件:

```
ocsp.responderURL=http://url.to.responder:port
ocsp.responderCertSubjectName=CN=Example CA,O=IBM,C=US
```

   并通过使用以下行编辑 $JAVA_HOME/lib/security/java.security 文件来启用 OCSP 检查:

```
ocsp.enable=true
```

2. 如果设置了 AIA，请通过使用以下行编辑 $JAVA_HOME/lib/security/java.security 文件来启用 OCSP 检查:

```
ocsp.enable=true
```

下一步做什么

如果您正在使用 Java Security Manager，那么过于完成配置，请将以下 Java 许可权添加到 lib/security/java.policy

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
```

使用 *keystore.conf*

过程

将以下属性添加到配置文件:

```
ocsp.enable=true
```

要点: 在配置文件中设置此属性将覆盖 java.security 设置。

下一步做什么

要完成配置，请将以下 Java 许可权添加到 lib/security/java.policy:

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
permission java.security.SecurityPermission "setProperty.ocsp.enable";
```

AMS 中的证书撤销列表 *(CRL)*

CRL 包含由认证中心 (CA) 标记为由于各种原因不再可信的证书列表，例如，专用密钥已丢失或已损坏。

为了验证证书， Advanced Message Security 会构造由签署者证书和认证中心 (CA) 证书链组成的证书链，直至信任锚。 信任锚是包含可信证书或用于断言证书信任的可信根证书的可信密钥库文件。 AMS 使用 PKIX 验证算法验证证书路径。 创建并验证链时， AMS 将完成证书验证，包括根据当前日期验证链中每个证书的发放日期和到期日期，并检查 "结束实体" 证书中是否存在密钥使用情况扩展。 如果将扩展附加到证书，那么 AMS 将验证是否还设置了 **digitalSignature** 或 **nonRepudiation** 。 如果没有，那么将报告并记录 MQRC_SECURITY_ERROR 。 接下来， AMS 将根据配置文件中指定的值从文件或 LDAP 下载 CRL。 AMS 仅支持以 DER 格式编码的 CRL。 如果在密钥库配置文件中找不到与 CRL 相关的配置，那么 AMS 不会执行 CRL 有效性检验。 对于每个 CA 证书， AMS 使用 CA 的专有名称查询 LDAP 以查找其 CRL。 LDAP 查询中包含以下属性:

```
certificateRevocationList,
certificateRevocationList;binary,
authorityRevocationList,
authorityRevocationList;binary
```

```
deltaRevocationList
deltaRevocationList;binary,
```

注: 仅当将 deltaRevocationList 指定为分发点时，才支持。

在本机拦截器中启用证书验证和证书撤销列表支持
您必须修改密钥库配置文件，以便 Advanced Message Security 可以从轻量级目录访问协议 (LDAP) 服务器下载 CLR。

关于此任务

IBM i 对于 IBM i 上的 Advanced Message Security，不支持在本机拦截器中启用证书验证和证书撤销列表支持。

过程

将以下选项添加到配置文件:

注: 所有 CRL 节都是可选的，可以单独指定。

选项	描述
crl.ldap.host=*host_name*	LDAP 服务器主机名。
crl.ldap.port=*port_number*	LDAP 服务器端口号。 最多可以指定 11 个服务器。 多个 LDAP 主机用于确保在发生 LDAP 连接故障时进行透明故障转移。 期望所有 LDAP 服务器都是副本，并且包含相同的数据。 当 AMS Java 拦截器成功连接到 LDAP 服务器时，它不会尝试从提供的其余服务器下载 CRL。
crl.cdp=off	使用此选项可检查或使用证书中的 CRLDistributionPoints 扩展。
crl.ldap.version=3	LDAP 协议版本号。 可能的值: 2 或 3。
crl.ldap.user=cn=*username*	登录到 LDAP 服务器。 如果未指定此值，那么 LDAP 中的 CRL 属性必须是全局可读的
crl.ldap.pass=*password*	LDAP 服务器的密码。
crl.ldap.encrypted=*no/yes*	clr.ldap.pass 是否已加密。 有关更多信息，请参阅 保护 AMS 配置文件中的密码 。
crl.ldap.cache_lifetime=0	LDAP 高速缓存生存期 (以秒为单位)。 可能的值: 0-86400。
crl.ldap.cache_size=50	LDAP 高速缓存大小。 仅当 crl.ldap.cache_lifetime 值大于 0 时，才能指定此选项。
crl.http.proxy.host=some.host.com	用于 CDP CRL 检索的 HTTP 代理服务器端口。
crl.http.proxy.port=8080	HTTP 代理服务器端口号。
crl.http.max_response_size=204800	可以从 IBM Global Security Kit (GSKit)接受的 HTTP 服务器检索的 CRL 的最大大小 (以字节计)。
crl.http.timeout=30	等待服务器响应的时间 (以秒计)，在此时间之后，AMS 将超时。
crl.http.cache_size=0	HTTP 高速缓存大小 (以字节计)。

选项	描述
crl.unknown=ACCEPT	定义在超时时间段内无法访问 CRL 服务器时的行为。 可能的值为: • ACCEPT 允许证书 • WARN 允许证书并记录警告 • REJECT 阻止使用证书并记录错误

在 *AMS* 中的 *Java* 中启用证书撤销列表支持
要在 Advanced Message Security 中启用 CRL 支持，必须修改密钥库配置文件以允许 AMS 从轻量级目录访问协议 (LDAP) 服务器下载 CRL 并配置 java.security 文件。

过程

1. 将以下选项添加到配置文件:

头	描述
crl.ldap.host=*host_name*	LDAP 主机名。
crl.ldap.port=*port_number*	LDAP 服务器端口号。 最多可以指定 11 个服务器。 多个 LDAP 主机用于确保在发生 LDAP 连接故障时进行透明故障转移。 期望所有 LDAP 服务器都是副本，并且包含相同的数据。 当 AMS Java 拦截器成功连接到 LDAP 服务器时，它不会尝试从提供的其余服务器下载 CRL。 Java 不使用 crl.ldap.user 和 crl.ldaworldp.pass 值。 在连接到 LDAP 服务器时，它不会使用用户和密码。 因此， LDAP 中的 CRL 属性必须是世界可读的。
crl.cdp=on/off	使用此选项可检查或使用证书中的 CRLDistributionPoints 扩展。

2. 使用以下属性修改 JRE/lib/security/java.security 文件:

属性名	描述
com.ibm.security.enableCRLDP	此属性采用以下值: true 或 false。 如果设置为 true，那么在执行证书撤销检查时，将使用来自证书的 CRL 分发点扩展的 URL 来查找 CRL。 如果设置为 false 或未设置，那么将禁用使用 CRL 分发点扩展来检查 CRL。
ibm.security.certpath.ldap.cache.lifetime	此属性可用于将 LDAP CertStore 的内存高速缓存中的条目生存期设置为以秒为单位的值。 值 0 表示禁用高速缓存; -1 表示无限生存期。 如果未设置，那么缺省生存期为 30 秒。

属性名	描述
com.ibm.security.enableAIAEXT	此属性采用以下值: true 或 false。 如果设置为 true，那么将检查在所构建证书路径的证书中找到的任何 "权限信息访问" 扩展，以确定它们是否包含 LDAP URI。 对于找到的每个 LDAP URI，将创建一个 LDAPCertStore 对象并将其添加到用于查找构建证书路径所需的其他证书的 CertStores 集合中。 如果设置为 false 或未设置，那么不会创建其他 LDAPCertStore 对象。

z/OS 在 *z/OS* 上启用证书撤销列表 *(CRL)*

Advanced Message Security 支持对用于保护数据消息的数字证书进行证书撤销列表 (CRL) 检查

关于此任务

启用时， Advanced Message Security 将在将消息放入受保护的隐私队列时验证接收方证书，并在从受保护的队列中检索消息时验证发送方证书 (完整性或隐私)。 在这种情况下，验证包括验证相关证书是否未在相关 CRL 中注册。

Advanced Message Security 使用 IBM 系统 SSL 服务来验证发送方和接收方证书。 您可以在以下位置找到有关系统 SSL 证书验证的详细文档 z/OS 密码服务系统安全套接字层编程手动的。

要启用 CRL 检查，请在 AMS 地址空间的启动式任务 JCL 中通过 CRLFILE DD 指定 CRL 配置文件的位置。 *thlqual*.SCSQPROC (CSQ40CRL) 中提供了可定制的样本 CRL 配置文件。 此文件中允许的设置如下所示:

表 *105: Advanced Message Security CRL* 配置变量

变量	有效值	描述
crl.ldap.host[.n]	*hostname-or-hostname: port*	用于托管颁发者证书的 CRL 的 LDAP 服务器的 ipaddr/hostname。 如果未指定 LDAP 服务器的端口号，那么将使用 crl.ldap.port 指定的端口号。
crl.ldap.port	端口	LDAP 服务器的 TCP/IP 端口号。
crl.ldap.user	*ldap_user*	连接到 LDAP 服务器时要使用的 LDAP 用户名。
crl.ldap.pass	*ldap_password*	与 crl.ldap.user 关联的 LDAP 密码。

您可以指定多个 LDAP 服务器主机名和端口，如下所示:

```
crl.ldap.host.1 = hostname -or hostname:port
crl.ldap.host.2 = hostname -or hostname:port
crl.ldap.host.3 = hostname -or hostname:port
```

最多可以指定 10 个主机名。 如果未指定 LDAP 服务器的端口号，那么将使用 crl.ldap.port 指定的端口号。 每个 LDAP 服务器都必须使用相同的 crl.ldap.user/password 组合进行访问。

指定 CRLFILE DD 时，将在 Advanced Message Security 地址空间初始化期间装入配置，并启用 CRL 检查。 如果未指定 CRLFILE DD，或者 CRL 配置文件不可用或无效，那么将禁用 CRL 检查。

AMS 使用 IBM System SSL 证书验证服务执行 CRL 检查，如下所示:

表 *106: Advanced Message Security CRL* 检查		
操作	**保护质量**	**已检查证书**
PUT	隐私	收件人
GET	完整性/隐私	发送方

如果消息操作失败，那么 CRL 检查 Advanced Message Security 将执行以下操作:

表 *107: Advanced Message Security CRL* 检查失败行为	
操作	**CRL 检查失败**
PUT	消息未放入目标队列。 将完成代码 MQCC_FAILED 和原因码 MQRC_SECURITY_ERROR 返回到应用程序。
GET	该消息将从目标队列中除去，并移至系统保护错误队列。 将完成代码 MQCC_FAILED 和原因码 MQRC_SECURITY_ERROR 返回到应用程序。

AMS for z/OS 使用 IBM 系统 SSL 服务来验证证书，包括 CRL 和信任检查。

IBM MQ 使用安全设置，其中证书验证需要 LDAP 服务器可联系，但不需要定义 CRL。

注: 管理员负责确保相关 LDAP 服务可用，并维护相关认证中心的 CRL 条目。

为配置文件设置 AMS 密码保护

将密钥库和专用密钥密码存储为纯文本会带来安全风险，因此 Advanced Message Security 提供了一个可以使用用户密钥对这些密码进行加扰的工具。

开始之前

`keystore.conf` 文件所有者必须确保只有文件所有者才有权读写该文件。 本主题中描述的密码保护只是额外的保护措施。 此外，您应该在安全系统上执行此过程。

确保对将要读取配置文件的 AMS 客户机类型使用正确的 **runamscred** 变体。 如果 AMS 客户机是:

- Java 客户机，您应该使用 Java **runamscred** 命令，该命令位于 `<IBM MQ installation root>/java/bin`
- MQI 客户机，您应该使用位于 `<IBM MQ installation root>/bin` 中的 MQI **runmqascred** 命令

过程

1. 编辑 `keystore.conf` 文件以包含所有必需信息，包括需要保护的密码。

```
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. 将加密密钥放在保护 `keystore.conf` 文件的用户可访问的文件中，以对密码进行加密。
   此密钥必须与稍后将由 AMS 客户机使用的密钥相同:

```
ThisIsAnExampleEncryptionKey
```

3. 运行 **runamscred** 命令以保护提供加密密钥文件的 `keystore.conf` 文件。

```
runamscred -f <location of keystore.conf> -sf <location of encryption keyfile>
```

4. 验证 `keystore.conf` 文件是否已受保护并包含加密密码。

示例

以下示例显示受保护的 keystore.conf 文件的外观:

```
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
```

相关信息

runamscred: 保护 AMS 关键字

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 584 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 584, "AMS" indicates " Advanced Message Security".

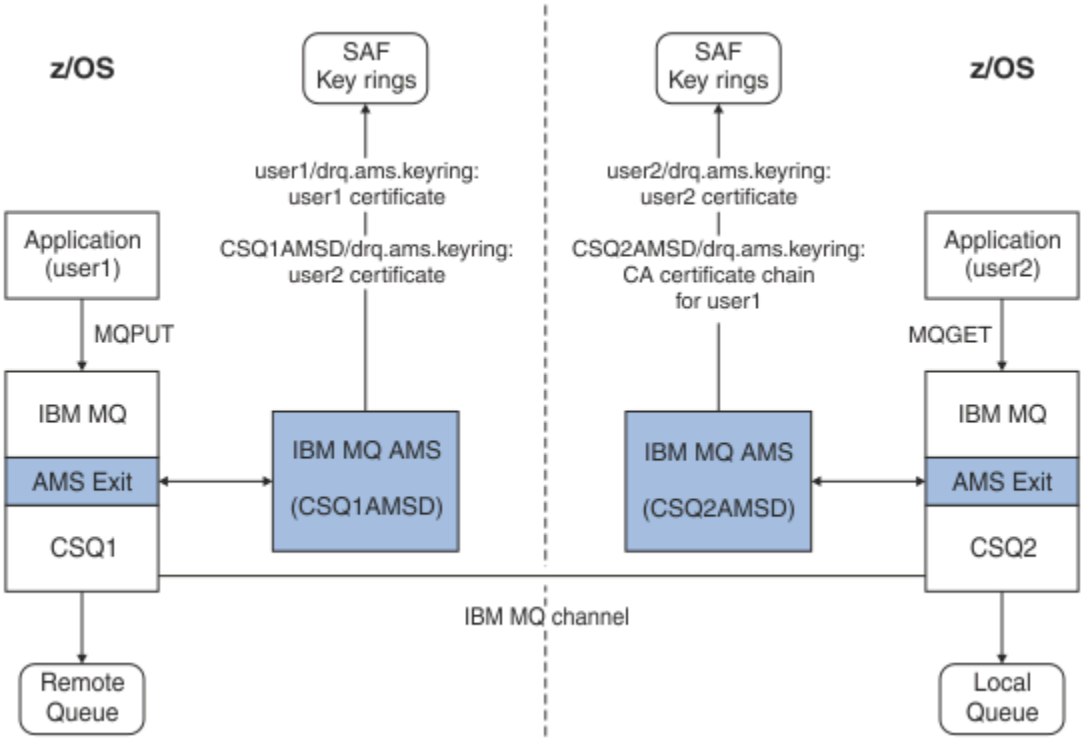


Figure 35. Application and certificate relationships

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.

管理 Advanced Message Security 安全策略

Advanced Message Security 使用安全策略来指定加密和签名算法，以加密和认证流经队列的消息。

AMS 的安全策略概述

Advanced Message Security 安全策略是概念性对象，用于描述以加密方式加密和签名消息的方式。

有关安全策略属性的详细信息，请参阅以下子主题:

相关概念
第 589 页的『AMS 中的保护质量』
Advanced Message Security 数据保护策略意味着保护质量 (QOP)。

第 589 页的『AMS 中的安全策略属性』
您可以使用 Advanced Message Security 来选择特定算法或方法以保护数据。

AMS 中的策略名称

策略名称是用于标识特定 Advanced Message Security 策略及其应用的队列的唯一名称。

策略名称必须与应用策略的队列名称相同。 Advanced Message Security (AMS) 之间存在一对一映射 策略和队列。

通过创建与队列同名的策略，可以激活该队列的策略。 没有匹配策略名称的队列不受 AMS 保护。

策略的作用域与本地队列管理器及其队列相关。 远程队列管理器必须具有其自己的本地定义的策略，以用于其管理的队列。

AMS 中的签名算法

签名算法指示对数据消息进行签名时应使用的算法。

有效值为:

- MD5
- SHA-1
- SHA-2 字型(F):
  - SHA256
  - SHA384 (可接受的最小密钥长度-768 位)
  - SHA512 (可接受的最小密钥长度-768 位)

未指定签名算法或指定算法 NONE 的策略意味着不会对放置在与策略关联的队列上的消息进行签名。

注: 用于消息放置和获取功能的保护质量必须匹配。 如果队列与队列中的消息之间存在保护不匹配的策略质量，那么将不接受该消息并将其发送到错误处理队列。 此规则适用于本地队列和远程队列。

AMS 中的加密算法

加密算法指示对放置在与策略关联的队列上的数据消息进行加密时应使用的算法。

有效值为:

- Deprecated RC2
- Deprecated DES
- Deprecated 3DES
- AES128
- AES256

如果策略未指定加密算法或指定 NONE 算法，那么意味着放置在与策略关联的队列上的消息不会加密。

请注意，指定 NONE 以外的加密算法的策略还必须至少指定一个接收方 DN 和签名算法，因为 Advanced Message Security 加密消息也已签名。

要点: 用于消息放置和获取功能的保护质量必须匹配。 如果队列与队列中的消息之间存在保护不匹配的策略质量，那么将不接受该消息并将其发送到错误处理队列。 此规则适用于本地队列和远程队列。

AMS 中的容错

容错属性指示 Advanced Message Security 是否可以接受未指定安全策略的消息。

从具有用于加密消息的策略的队列中检索消息时，如果消息未加密，那么会将其返回到调用应用程序。 有效值为:

0
否 (**缺省值**)。

1
是。

未指定容错值或指定 0 的策略暗示与策略关联的队列上的消息必须与策略规则匹配。

容错是可选的，存在以促进配置转出，其中策略已应用于队列，但这些队列已包含未指定安全策略的消息。

AMS 中的发送方专有名称

发送方专有名称 (DN) 标识有权将消息放入队列的用户。 在将消息放入队列之前，发送方使用其证书对消息进行签名。

Advanced Message Security (AMS) 在检索消息之前，不检查有效用户是否已将消息放在数据保护的队列上。 此时，如果策略规定了一个或多个有效发件人，并且将消息放入队列的用户不在有效发件人列表中，那么 AMS 会向接收应用程序返回错误，并将消息放入 AMS 错误队列中。

可以为一个策略指定 0 个或 0 个以上的发送方 DN。 如果没有为策略指定发送方 DN，那么任何发送方都可以将受数据保护的消息放入队列中，前提是发送方的证书是可信的。 通过将公用证书添加到可供接收应用程序使用的密钥库，可信任发送方的证书。

发送方专有名称的格式如下:

```
CN=Common Name,O=Organization,C=Country
```

要点:

- 所有 DN 组件名称都必须为大写。 必须按下表中显示的顺序指定 DN 中的所有组件名称标识:

组件名称	值
CN	此 DN 的对象的公共名称，例如全名或设备的预期用途。
OU	DN 对象所属组织中的单位，例如公司部门或产品名称。
O	DN 对象所属的组织，例如公司。
L	DN 对象所在的位置 (城市或市政府)。
ST	DN 对象所在的省/直辖市/自治区名称。

组件名称	值
C	专有名称 (DN) 对象所在的国家或地区。

- 如果为策略指定了一个或多个发送方 DN，那么只有那些用户可以将消息放入与该策略关联的队列。
- 指定发送方 DN 时，必须完全匹配与放入消息的用户关联的数字证书中包含的 DN。
- AMS 仅支持具有来自 Latin-1 字符集的值的 DN。 要创建具有该集的字符的 DN，必须首先创建具有 DN 的证书，该 DN 是使用开启了 UTF-8 编码的 AIX and Linux 以 UTF-8 编码创建的。 然后，必须在开启 UTF-8 编码的情况下从 Linux 或 AIX 平台创建策略，或者使用 AMS 插件来 IBM MQ。
- AMS 用于将发送方的名称从 x.509 格式转换为 DN 格式的方法始终使用 ST = 作为 "省/直辖市/自治区" 值。
- 以下特殊字符需要转义字符:

```
, (comma)
+ (plus)
" (double quote)
\ (backslash)
< (less than)
> (greater than)
; (semicolon)
```

- 如果专有名称包含嵌入空白，那么应将 DN 括在双引号中。

相关概念

第 588 页的『AMS 中的收件人专有名称』
接收方专有名称 (DN) 标识有权从队列检索消息的用户。

AMS 中的收件人专有名称

接收方专有名称 (DN) 标识有权从队列检索消息的用户。

可以为一个策略指定 0 个或 0 个以上的接收方 DN。 收件人专有名称具有以下格式:

```
CN=Common Name,O=Organization,C=Country
```

要点:

- 所有 DN 组件名称都必须为大写。 必须按下表中显示的顺序指定 DN 中的所有组件名称标识:

组件名称	值
CN	此 DN 的对象的公共名称，例如全名或设备的预期用途。
OU	DN 对象所属组织中的单位，例如公司部门或产品名称。
O	DN 对象所属的组织，例如公司。
L	DN 对象所在的位置 (城市或市政府)。
ST	DN 对象所在的省/直辖市/自治区名称。
C	专有名称 (DN) 对象所在的国家或地区。

- 如果没有为策略指定接收方 DN，那么任何用户都可以从与该策略关联的队列获取消息。
- 如果为策略指定了一个或多个接收方 DN，那么只有那些用户可以从与该策略关联的队列获取消息。
- 指定接收方 DN 时，必须完全匹配与获取消息的用户关联的数字证书中包含的 DN。
- Advanced Message Security 仅支持具有来自 Latin-1 字符集的值的 DN。 要创建具有该集的字符的 DN，必须首先创建具有使用 UTF-8 编码创建的 DN 的证书，使用 AIX 或开启 UTF-8 编码的 Linux 。 然后，必须在开启 UTF-8 编码的情况下从 Linux 或 AIX 平台创建策略，或者使用 Advanced Message Security 插件来 IBM MQ。

相关概念

第 587 页的『AMS 中的发送方专有名称』

发送方专有名称 (DN) 标识有权将消息放入队列的用户。 在将消息放入队列之前，发送方使用其证书对消息进行签名。

AMS 中的安全策略属性

您可以使用 Advanced Message Security 来选择特定算法或方法以保护数据。

安全策略是一个概念对象，用于描述以加密方式加密和签名消息的方式。

表 *109: AMS* 中的安全策略属性

属性	描述
策略名称	队列管理器的策略的唯一名称。
签名算法	用于在发送前对消息进行签名的密码算法。
加密算法	用于在发送前对消息进行加密的密码算法。
收件人列表	消息的潜在接收方的证书专有名称 (DN) 列表。
签名 DN 核对表	要在消息检索期间验证的签名 DN 的列表。

在 Advanced Message Security 中，使用对称密钥对消息进行加密，并使用收件人的公用密钥对对称密钥进行加密。 公用密钥使用 RSA 算法进行加密，其有效长度最多为 2048 位的密钥。 实际非对称密钥加密取决于证书密钥长度。

受支持的对称密钥算法如下所示:

- Deprecated RC2
- Deprecated DES
- Deprecated 3DES
- AES128
- AES256

Advanced Message Security 还支持以下加密散列函数:

- Deprecated MD5
- Deprecated SHA-1
- SHA-2 字型(F):
  - SHA256
  - SHA384 (可接受的最小密钥长度-768 位)
  - SHA512 (可接受的最小密钥长度-768 位)

注: 用于消息放置和获取功能的保护质量必须匹配。 如果队列与队列中的消息之间存在保护不匹配的策略质量，那么将不接受该消息并将其发送到错误处理队列。 此规则适用于本地队列和远程队列。

AMS 中的保护质量

Advanced Message Security 数据保护策略意味着保护质量 (QOP)。

Advanced Message Security 中的三个质量保护级别由 IBM MQ 9.0 和更高版本中的第四个级别补充，所有这些级别都取决于用于对消息进行签名和加密的密码算法:

- 必须对放置在队列上的隐私消息进行签名和加密。
- 完整性-放置在队列上的消息必须由发送方签名。
- 机密性-必须对放入队列中的消息进行加密。 有关更多信息，请参阅 第 523 页的『AMS 提供的保护质量』
- 无-没有适用的数据保护。

一种策略，规定在将消息放入队列时必须对其进行签名，该策略的 QOP 为 INTEGRITY。 INTEGRITY 的 QOP 表示策略规定了签名算法，但没有规定加密算法。 受完整性保护的消息也称为 "SIGNED"。

这是一个策略，用于规定在将消息放入队列时必须对其进行签名和加密，该策略具有 PRIVACY 的 QOP。 PRIVACY 的 QOP 表示当策略规定了签名算法和加密算法时。 受隐私保护的消息也称为 "密封"。

这是一个策略，用于规定在将消息放入队列时必须对其进行加密，该策略的 QOP 为保密。 保密的 QOP 表示策略规定了加密算法。

未规定签名算法或加密算法的策略具有 QOP NONE。 对于具有 QOP 为 NONE 的策略的队列， Advanced Message Security 不提供数据保护。

在 AMS 中管理安全策略

安全策略是一个概念对象，用于描述以加密方式加密和签名消息的方式。

运行与安全策略相关的所有管理任务的位置取决于您正在使用的平台。

- ALW 在 AIX, Linux, and Windows 上，使用 删除策略， 显示策略和 设置策略 (或等效 PCF) 命令来管理安全策略。
  - Linux AIX 在 AIX and Linux 上，可以从 *MQ_INSTALLATION_PATH*/bin 运行管理任务。
  - Windows 在 Windows 平台上，可以在安装时更新 PATH 环境变量时从任何位置运行管理任务。
- IBM i 在 IBM i 上，安装 IBM MQ 时，会将 DSPMQMSPL， SETMQMSPL 和 WRKMQMSPL 命令安装到系统主语言的 QSYS 系统库中。

  根据语言功能部件装入，将其他本地语言版本安装到 QSYS29xx 库中。 例如，以美国英语作为主语言，以韩国语作为辅助语言的机器将美国英语命令安装到 QSYS 中，而 QSYS2962 中的韩国语辅助语言装入是韩国语的语言装入。
- z/OS 在 z/OS 上，使用消息安全策略实用程序 (CSQ0UTIL) 来运行管理命令。 在 z/OS 上创建，修改或删除策略时， Advanced Message Security 不会识别这些更改，直到停止并重新启动队列管理器，或者使用 z/OS MODIFY 命令来刷新 Advanced Message Security 策略配置。 例如：

```
F <qmgr ssid>AMSM,REFRESH POLICY
```

相关任务

第 590 页的『在 AMS 中创建安全策略』
安全策略定义在放置消息时保护消息的方式，或者在接收消息时必须如何保护消息。

第 591 页的『在 AMS 中更改安全策略』
您可以使用 Advanced Message Security 来变更已定义的安全策略的详细信息。

第 592 页的『在 AMS 中显示和转储安全策略』
根据您提供的命令行参数，使用 **dspmqspl** 命令可显示所有安全策略的列表或指定策略的详细信息。

第 593 页的『在 AMS 中除去安全策略』
要在 Advanced Message Security 中除去安全策略，必须使用 `setmqspl` 命令。

操作 Advanced Message Security

相关参考

消息安全策略实用程序 (CSQ0UTIL)

在 *AMS* 中创建安全策略

安全策略定义在放置消息时保护消息的方式，或者在接收消息时必须如何保护消息。

开始之前

创建安全策略时必须满足一些入口条件：

- 该队列管理器必须正在运行。

- 安全策略的名称必须遵循 用于命名 IBM MQ 对象的规则。
- 您必须具有必要的权限才能连接到队列管理器并创建安全策略:
  - z/OS 在 z/OS 上，授予 消息安全策略实用程序 (CSQ0UTIL)中记录的权限。
  - Multi 在多平台上，必须使用 **setmqaut** 命令授予必要的 +connect， +inq 和 +chg 权限。

  有关配置安全性的更多信息，请参阅 第 110 页的『设置安全性』。
- z/OS 在 z/OS 上，确保已根据 CSQ4INSM 中的定义定义了必需的系统对象。

示例

以下是在队列管理器 QMGR 上创建策略的示例。 策略指定使用 SHA256 算法对消息进行签名，并使用 AES256 算法对具有 DN 的证书进行加密 :CN=joe， O=IBM， C=US 和 DN :CN=jane， O=IBM， C = US。 此策略附加到 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
```

以下是在队列管理器 QMGR 上创建策略的示例。 该策略指定使用 3DES 算法对具有 DN 的证书加密消息 :CN=john， O=IBM， C=US 和 CN=jeff， O=IBM， C=US，并使用 SHA256 算法对具有 DN 的证书进行签名 :CN=phil， O=IBM， C=US

```
setmqspl -m QMGR -p MY.OTHER.QUEUE -s SHA256 -e 3DES -r CN=john,O=IBM,C=US -r
CN=jeff,O=IBM,C=US -a CN=phil,O=IBM,C=US
```

注:

- 用于消息放置和获取的保护的质量必须匹配。 如果为消息定义的策略保护质量弱于为队列定义的策略保护质量，那么会将消息发送到错误处理队列。 此策略对本地队列和远程队列都有效。

相关参考

setmqspl 命令属性的完整列表

在 *AMS* 中更改安全策略

您可以使用 Advanced Message Security 来变更已定义的安全策略的详细信息。

开始之前

- 要操作的队列管理器必须正在运行。
- 您必须具有必要的权限才能连接到队列管理器并创建安全策略。
  - z/OS 在 z/OS 上，授予 消息安全策略实用程序 (CSQ0UTIL)中记录的权限。
  - Multi 在多平台上，必须使用 **setmqaut** 命令授予必要的 +connect， +inq 和 +chg 权限。

  有关配置安全性的更多信息，请参阅 第 110 页的『设置安全性』。

关于此任务

要更改安全策略，请将 setmqspl 命令应用于提供新属性的现有策略。

示例

以下是在名为 QMGR 的队列管理器上创建名为 MYQUEUE 的策略的示例，指定将使用 3DES 算法对具有专有名称 (DN) 为 CN=alice， O=IBM， C=US 的证书的作者 (**-a**) 加密消息，并使用 SHA256 算法对具有 DN 为 CN=jeff， O=IBM， C = US 的证书的接收方 (**-r**) 进行签名。

```
setmqspl -m QMGR -p MYQUEUE -e 3DES -s SHA256 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

要变更此策略，请发出 setmqspl 命令，其中包含示例中的所有属性，仅更改要修改的值。 在此示例中，先前创建的策略将连接到新队列，并且其加密算法将更改为 AES256:

```
setmqspl -m QMGR -p MYQUEUE -e AES256 -s SHA256 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

相关参考

setmqspl (设置安全策略)

在 *AMS* 中显示和转储安全策略

根据您提供的命令行参数，使用 **dspmqspl** 命令可显示所有安全策略的列表或指定策略的详细信息。

开始之前

- 要显示安全策略详细信息，队列管理器必须存在并且正在运行。
- 您必须具有必要的权限才能连接到队列管理器并创建安全策略。
  - z/OS 在 z/OS 上，授予 消息安全策略实用程序 (CSQ0UTIL)中记录的权限。
  - Multi 在多平台上，必须使用 **setmqaut** 命令授予必要的 +connect， +inq 和 +chg 权限。

有关配置安全性的更多信息，请参阅 第 110 页的『设置安全性』。

关于此任务

以下是 **dspmqspl** 命令标志的列表:

表 *110:* ***dspmqspl*** 命令标志。

命令标志	**说明**
-m	队列管理器名称 (必需)。
-p	策略名称。
-export	添加此标志将生成可轻松应用于其他队列管理器的输出。

示例

以下示例显示如何为 venus.queue.manager 创建两个安全策略:

```
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
```

此示例显示了一个命令，该命令显示为 venus.queue.manager 定义的所有策略的详细信息及其生成的输出:

```
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
```

此示例显示了一个命令，该命令显示为 venus.queue.manager 定义的所选安全策略的详细信息及其生成的输出:

```
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
```

在下一个示例中，首先创建安全策略，然后使用 **-export** 标志导出策略:

```
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 在 z/OS 上，导出的策略信息由 CSQ0UTIL 写入 EXPORT DD。

Multi 在 Multiplatforms 版上，将输出重定向到文件，例如:

```
dspmqspl -m venus.queue.manager -export > policies.[bat|sh]
```

要导入安全策略:

- Linux AIX 在 AIX and Linux 上:
  1. 以属于 mqm IBM MQ 管理组的用户身份登录。
  2. 问题 `. policies.sh`。
- Windows 在 Windows 上，运行 `policies.bat`。
- z/OS 在 z/OS 上，使用 CSQ0UTIL 实用程序，向 SYSIN 指定包含导出的策略信息的数据集。

相关参考

dspmqspl 命令属性的完整列表

在 *AMS* 中除去安全策略

要在 Advanced Message Security 中除去安全策略，必须使用 `setmqspl` 命令。

开始之前

管理安全策略时必须满足一些入口条件:

- 该队列管理器必须正在运行。
- 您必须具有必要的权限才能连接到队列管理器并创建安全策略。
  - z/OS 在 z/OS 上，授予 消息安全策略实用程序 (CSQ0UTIL)中记录的权限。
  - Multi 在多平台上，必须使用 **setmqaut** 命令授予必要的 +connect， +inq 和 +chg 权限。

  有关配置安全性的更多信息，请参阅 第 110 页的『设置安全性』。

关于此任务

使用带有 **-remove** 选项的 **setmqspl** 命令。

示例

以下是除去策略的示例:

```
setmqspl -m QMGR -remove -p MY.OTHER.QUEUE
```

相关参考

setmqspl 命令属性的完整列表

AMS 中的系统队列保护

系统队列支持 IBM MQ 与其辅助应用程序之间的通信。 每当创建队列管理器时，还会创建系统队列以存储 IBM MQ 内部消息和数据。 您可以使用 Advanced Message Security 保护系统队列，以便只有授权用户才能访问或解密这些队列。

系统队列保护遵循与常规队列保护相同的模式。 请参阅第 590 页的『在 AMS 中创建安全策略』。

Windows 要在 Windows 上使用系统队列保护，请将 keystore.conf 文件复制到以下目录:

```
c:\Documents and Settings\Default User\.mqs\keystore.conf
```

z/OS 在 z/OS 上，要为 SYSTEM.ADMIN.COMMAND.QUEUE 提供保护，命令服务器必须有权访问包含密钥和配置的 keystore 和 keystore.conf，以便命令服务器可以访问密钥和证书。 对 SYSTEM.ADMIN.COMMAND.QUEUE 的安全策略所作的所有更改都需要重新启动命令服务器。

根据策略设置，将对从命令队列发送和接收的所有消息进行签名或签名和加密。 如果管理员定义授权签署者，那么未通过签署者专有名称 (DN) 检查的命令消息不会由命令服务器执行，并且不会路由到 Advanced Message Security 错误处理队列。 作为回复发送到 IBM MQ Explorer 临时动态队列的消息不受 AMS 保护。

安全策略不会影响以下 SYSTEM 队列:

- 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 流式队列和 AMS

可以流式采集重复的 Advanced Message Security (AMS) 受保护消息。

如果队列定义了 AMS 策略，导致对放入该队列的消息进行签名和/或加密，那么您还可以配置队列的 **STREAMQ** 属性，以将每个受保护消息的副本放入另一个队列。 使用为原始队列配置的同一策略对重复的流式消息进行签名和/或加密。

在以下示例中，您将配置两个队列: QUEUE1 和 QUEUE2。 QUEUE1 的 **STREAMQ** 属性已配置为将流式消息放入 QUEUE2:

```
DEFINE QLOCAL(QUEUE2)
DEFINE QLOCAL(QUEUE1) STREAMQ(QUEUE2)
```

AMS 受保护消息正由具有证书 CN=bob,O=IBM,C=GB 的用户放入 QUEUE1 。

具有证书 CN=alice,O=IBM,C=GB 的应用程序将使用来自 QUEUE1 的消息。 具有证书 CN=fred,O=IBM,C=GB 的单独应用程序将使用来自 QUEUE2 的消息。

QUEUE1 应用了以下 AMS 隐私策略:

```
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)
```

如果在策略中为 QUEUE1 配置了加密算法，那么策略中列出的收件人必须同时包含来自 QUEUE1 的原始消息的收件人以及将使用来自 QUEUE2 的重复消息的收件人。

当应用程序尝试使用来自 QUEUE2 的消息时，它将根据 QUEUE2 上设置的策略执行完整性检查和/或解密消息。 如果应用程序想要使用来自 QUEUE2 的流式消息，那么必须在 QUEUE2 上设置合适的策略，以允许检查消息的完整性并正确解密。

尤其是，签名算法，签署者和加密算法必须与应用于 QUEUE1 的策略相同。 QUEUE2 的策略收件人必须包含使用来自 QUEUE2 的消息的收件人的身份。

注: 应用于 QUEUE2 的策略不需要列出在 QUEUE1 上的策略集中指定的所有收件人。

例如，可以在 QUEUE2 上设置以下策略，以允许具有证书专有名称 CN=fred,O=IBM,C=GB 的应用程序从中读取 AMS 受保护的消息:

```
SET POLICY(QUEUE2) SIGNALG(SHA256) SIGNER('CN=bob,O=IBM,C=GB') ENCALG(AES256)
RECIP('CN=fred,O=IBM,C=GB') ACTION(ADD)
```

相关概念

流式队列

在 AMS 中授予 OAM 许可权

文件许可权授权所有用户执行 setmqspl 和 dspmqspl 命令。 但是， Advanced Message Security 依赖于对象权限管理器 (OAM) ，并且不属于 mqm 组 (即 IBM MQ 管理组) 或无权读取已授予的安全策略设置的用户每次尝试执行这些命令都会导致错误。

过程

要向用户授予必需的许可权，请运行:

```
setmqaut -m SOME.QUEUE.MANAGER -t qmgr -p SOME.USER +connect +inq
setmqaut -m SOME.QUEUE.MANAGER -t queue -n SYSTEM.PROTECTION.POLICY.QUEUE -p SOME.USER +browse
+put
setmqaut -m SOME.QUEUE.MANAGER -t queue -n SYSTEM.PROTECTION.ERROR.QUEUE -p SOME.USER +put
```

注: 仅当您打算使用 Advanced Message Security 7.0.1 将客户机连接到队列管理器时，才需要设置这些 OAM 权限。

注意: 对 SYSTEM.PROTECTION.POLICY.QUEUE 并非在所有情况下都是必需的。 IBM MQ 通过高速缓存策略来优化性能，以便您不必在 SYSTEM.PROTECTION.POLICY.QUEUE 。

IBM MQ 不会高速缓存所有可用的策略。 如果存在大量策略，那么 IBM MQ 会高速缓存有限数量的策略。 因此，如果队列管理器定义的策略数较少，那么无需向 SYSTEM.PROTECTION.POLICY.QUEUE。

但是，如果定义了大量策略，或者如果您使用的是旧客户机，那么应该授予对此队列的浏览权限。 SYSTEM.PROTECTION.ERROR.QUEUE 用于放置由 AMS 代码生成的错误消息。 仅当您尝试将错误消息放入队列时，才会检查对此队列的放置权限。 尝试从受 AMS 保护的队列中放入或获取消息时，不会检查您对该队列的放入权限。

在 AMS 中授予安全许可权

使用命令资源安全性时，必须设置许可权以允许 Advanced Message Security 运行。 本主题在示例中使用 RACF 命令。 如果您的企业使用不同的外部安全管理器 (ESM) ，那么必须对该 ESM 使用等效命令。

授予安全许可权有三个方面:

- 第 597 页的『AMSM 地址空间』
- 第 597 页的『CSQ0UTIL』
- 第 597 页的『使用定义了 Advanced Message Security 策略的队列』

注意: 示例命令使用以下变量。

1. *QMgrName* -队列管理器的名称。

   z/OS 在 z/OS 上，此值也可以是队列共享组的名称。
2. *username* -这可以是组名。
3. 这些示例显示 MQQUEUE 类。 z/OS 这也可以是 MXQUEUE， GMQQUEUE 或 GMXQUEUE。 请参阅第 174 页的『Profiles for queue security』，以了解更多信息。

此外，如果概要文件已存在，那么您不需要 RDEFINE 命令。

AMSM 地址空间

您需要对运行 Advanced Message Security 地址空间的用户名发出一些 IBM MQ 安全性。

- 对于与队列管理器的批处理连接，请发出

```
RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
            PERMIT QMgrName.BATCH CLASS(MQCONN) ID(username) ACCESS(READ)
```

- 用于访问 SYSTEM.PROTECTION.POLICY.QUEUE，问题:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
            PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

CSQ0UTIL

允许用户运行 **setmqspl** 和 **dspmqspl** 命令的实用程序需要以下许可权，其中用户名是作业用户标识:

- 对于与队列管理器的批处理连接，请发出:

```
  RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
              PERMIT QMgrName.BATCH CLASS(MQCONN) ID(username) ACCESS(READ)
```

- 用于访问 SYSTEM.PROTECTION.POLICY.QUEUE，对于 **setmqpol** 命令是必需的，请发出:

```
  RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
               PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(ALTER)
```

- 用于访问 SYSTEM.PROTECTION.POLICY.QUEUE，对于 **dspmqpol** 命令是必需的，请发出:

```
  RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
                PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

使用定义了 Advanced Message Security 策略的队列

当应用程序对定义了策略的队列执行任何操作时，该应用程序需要其他许可权以允许 Advanced Message Security 保护消息。

应用程序需要:

- 对 SYSTEM.PROTECTION.POLICY.QUEUE。 通过发出以下命令来执行此操作:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
         PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

- 放置对 SYSTEM.PROTECTION.ERROR.QUEUE。 通过发出以下命令来执行此操作:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.ERROR.QUEUE UACC(NONE)
           PERMIT QMgrName.SYSTEM.PROTECTION.ERROR.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

IBM i 在 IBM i 上为 AMS 设置证书和密钥库配置文件

设置 Advanced Message Security 保护时的第一个任务是创建证书，并将其与环境相关联。 通过保存在集成文件系统 (IFS) 中的文件来配置关联。

过程

1. 要使用 IBM i 随附的 OpenSSL 工具创建自签名证书，请从 QShell 发出以下命令:

   ```
   /QOpenSys/usr/bin/openssl req -x509 -newkey rsa:2048 -keyout
   $HOME/private.pem -out $HOME/mycert.pem -nodes -days 365
   ```

   此命令提示输入新自签名证书的各种专有名称属性，包括:

   - 公共名称 (CN =)
   - 组织 (O =)
   - 国家 (C =)

   这将以 PEM (隐私增强邮件) 格式创建未加密的专用密钥和匹配的证书。

   为了简单起见，只需输入公共名称，组织和国家或地区的值。 创建策略时，这些属性和值很重要。

   可以通过在命令行上使用 **-config** 参数指定定制 openssl 配置文件来定制其他提示和属性。 有关配置文件语法的更多详细信息，请参阅 OpenSSL 文档。

   例如，以下命令添加其他 X.509 v3 证书扩展:

   ```
   /QOpenSys/usr/bin/openssl req -x509 -newkey rsa:2048
   -keyout $HOME/private.pem -out $HOME/mycert.pem -nodes -days 365 -config myconfig.cnf
   ```

   其中 myconfig.cnf 是包含以下内容的 ASCII 流文件:

   ```
   [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 要求证书和专用密钥都保存在同一文件中。 发出以下命令以实现此目的:

```
cat $HOME/mycert.pem >> $HOME/private.pem
```

$HOME 中的 private.pem 文件现在包含匹配的专用密钥和证书，而 mycert.pem 文件包含您可以对其加密消息和验证签名的所有公用证书。

这两个文件需要通过在缺省位置中创建密钥库配置文件 keystore.conf 来与您的环境相关联。

缺省情况下， AMS 会在主目录的 .mqs 子目录中查找密钥库配置。

3. 在 QShell 中，创建 keystore.conf 文件:

```
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 在 *IBM i* 上为 *AMS* 创建策略

在创建策略之前，您需要创建一个队列以保存受保护的消息。

过程

1. 在命令行提示符处输入;

```
CRTMQMQ QNAME(PROTECTED) QTYPE(*LCL) MQMNAME (mqmname)
```

其中 mqmname 是队列管理器的名称。

使用 DSPMQM 命令来检查队列管理器是否能够使用安全策略。 确保 **Security Policy Capability** 显示 **YES*。

您可以定义的最简单策略是完整性策略，这是通过创建具有数字签名算法但没有加密算法的策略来实现的。

消息已签名但未加密。 如果要对消息进行加密，那么必须指定加密算法以及一个或多个预期的消息收件人。

通过专有名称来标识预期消息收件人的公用密钥库中的证书。

2. 通过在 QShell 中使用以下命令，显示公用密钥库中证书的专有名称 mycert.pem (在 $HOME 中):

```
/QOpenSys/usr/bin/openssl x509 -in $HOME/mycert.pem -noout -subject -nameopt RFC2253
```

您需要输入专有名称作为预期收件人，并且策略名称必须与要保护的队列名称匹配。

3. 在 CL 命令提示符处输入，例如:

```
SETMQMSPL POLICY(PROTECTED) MQMNAME (mqmname)SIGNALG(*SHA256) ENCALG(*AES256) RECIP('CN=..,
O=.., C=..')
```

其中 mqmname 是队列管理器的名称。

创建策略后，通过该队列名称放入，浏览或破坏性除去的任何消息都将受 AMS 策略约束。

相关参考

显示消息队列管理器 (DSPMQM)

设置 MQM 安全策略 (SETMQMSPL)

IBM i 在 *IBM i* 上测试 *AMS* 的策略

使用产品随附的样本应用程序来测试安全策略。

关于此任务

您可以使用 IBM MQ 随附的样本应用程序 (例如， AMQSPUT4， AMQSGET4， AMQSGBR4) 和工具 (例如，WRKMQMMSG) 使用 PROTECTED 队列名称来放置，浏览和获取消息。

如果已正确配置所有内容，那么应用程序行为应该与此用户的不受保护队列的行为没有差别。

但是，没有为 Advanced Message Security 设置用户，或者没有解密消息所需的专用密钥的用户将无法查看消息。 用户接收到完成代码 RCFAIL，相当于 MQCC_FAILED (2) 和原因码 RC2063 (MQRC_SECURITY_ERROR)。

要查看 AMS 保护是否生效，请将一些测试消息放入 PROTECTED 队列，例如使用 AMQSPUT0。 然后，您可以创建别名队列以在静态时浏览原始受保护数据。

过程

要向用户授予必需的许可权，请运行:

```
CRTMQMQ QNAME(ALIAS) QTYPE(*ALS) TGTQNAME(PROTECTED) MQMNAME(yourqm)
```

使用 ALIAS 队列名称 (例如，使用 AMQSBCG4 或 WRKMQMMSG) 进行浏览时，应该会显示较大的 `scrambled` 消息，其中 PROTECTED 队列的浏览会显示 CLEARTEXT 消息。

已加扰的消息可视，但原始 cleartext 无法使用 ALIAS 队列进行解密，因为 AMS 没有策略来强制匹配此名称。 因此，将返回原始受保护数据。

相关参考

设置 MQM 安全策略 (SETMQMSPL)

使用 MQ 消息 (WRKMQMMSG)

AMS 的命令和配置事件

通过 Advanced Message Security，您可以生成命令和配置事件消息，这些消息可以记录并用作用于审计的策略更改的记录。

IBM MQ 生成的命令和配置事件是发送到发生该事件的队列管理器上的专用队列的 PCF 格式的消息。

配置事件消息将发送到 SYSTEM.ADMIN.CONFIG.EVENT 队列。

命令事件消息将发送到 SYSTEM.ADMIN.COMMAND.EVENT 队列。

无论您使用何种工具来管理 Advanced Message Security 安全策略，都会生成事件。

在 Advanced Message Security 中，有四种类型的事件由安全策略上的不同操作生成:

- 第 590 页的『在 AMS 中创建安全策略』，生成两条 IBM MQ 事件消息:
  - 配置事件
  - 命令事件
- 第 591 页的『在 AMS 中更改安全策略』，这将生成三条 IBM MQ 事件消息:
  - 包含旧安全策略值的配置事件
  - 包含新安全策略值的配置事件
  - 命令事件
- 第 592 页的『在 AMS 中显示和转储安全策略』，生成一条 IBM MQ 事件消息:
  - 命令事件
- 第 593 页的『在 AMS 中除去安全策略』，这将生成两条 IBM MQ 事件消息:
  - 配置事件
  - 命令事件

为 *AMS* 启用和禁用事件日志记录

您可以使用队列管理器属性 **CONFIGEV** 和 **CMDEV** 来控制命令和配置事件。 要启用这些事件，请将相应的队列管理器属性设置为 ENABLED。 要禁用这些事件，请将相应的队列管理器属性设置为 DISABLED。

过程

配置事件

要启用配置事件，请将 **CONFIGEV** 设置为 ENABLED。 要禁用配置事件，请将 **CONFIGEV** 设置为 DISABLED。 例如，可以使用以下 MQSC 命令来启用配置事件:

```
ALTER QMGR CONFIGEV (ENABLED)
```

命令事件

要启用命令事件，请将 **CMDEV** 设置为 ENABLED。 要对除 **DISPLAY MQSC** 命令和 Inquire PCF 命令以外的命令启用命令事件，请将 **CMDEV** 设置为 NODISPLAY。 要禁用命令事件，请将 **CMDEV** 设置为 DISABLED。 例如，可以使用以下 MQSC 命令来启用命令事件:

```
ALTER QMGR CMDEV (ENABLED)
```

相关任务

在 IBM MQ 中控制配置，命令和记录器事件

AMS 的命令事件消息格式

命令事件消息由 MQCFH 结构和后面的 PCF 参数组成。

以下是选定的 MQCFH 值:

```
Type = MQCFT_EVENT;
Command = MQCMD_COMMAND_EVENT;
MsgSeqNumber = 1;
Control = MQCFC_LAST;
ParameterCount = 2;
CompCode = MQCC_WARNING;
Reason = MQRC_COMMAND_PCF;
```

注: ParameterCount 值为两个，因为始终有两个 MQCFGR 类型 (组) 的参数。 每个组都包含相应的参数。 事件数据由两个组组成: CommandContext 和 CommandData。

CommandContext 包含:

EventUserId

描述:	发出生成该事件的命令或调用的用户标识。 (这是用于检查发出命令或调用的权限的用户标识; 对于从队列接收的命令，这也是来自命令消息的 MD 的用户标识 (UserIdentifier)。
标识:	MQCACF_EVENT_USER_ID。
数据类型:	MQCFST。
最大长度:	MQ_USER_ID_LENGTH。
已返回:	一直等在那里。

EventOrigin

描述:	导致事件的操作的来源。
标识:	MQIACF_EVENT_ORIGIN。
数据类型:	MQCFIN。

值:

MQEVO_CONSOLE
控制台命令-命令行。

MQEVO_MSG
来自 IBM MQ Explorer 插件的命令消息。

已返回: 一直等在那里。

EventQMgr

描述: 输入命令或调用的队列管理器。 (执行命令并生成事件的队列管理器位于事件消息的 MD 中)。

标识: MQCACF_EVENT_Q_MGR。

数据类型: MQCFST。

最大长度: MQ_Q_MGR_NAME_LENGTH。

已返回: 一直等在那里。

EventAccountingToken

描述: 对于作为消息 (MQEVO_MSG) 接收的命令，这是来自命令消息的 MD 的记帐令牌 (AccountingToken)。

标识: MQBACF_EVENT_ACCOUNTING_TOKEN。

数据类型: MQCFBS。

最大长度: MQ_ACCOUNTING_TOKEN_LENGTH。

已返回: 仅当 EventOrigin 是 MQEVO_MSG 时。

EventIdentity 数据

描述: 对于作为消息 (MQEVO_MSG) 接收的命令，来自命令消息的 MD 的应用程序身份数据 (ApplIdentity 数据)。

标识: MQCACF_EVENT_APPL_IDENTITY。

数据类型: MQCFST。

最大长度: MQ_APPL_IDENTITY_DATA_LENGTH。

已返回: 仅当 EventOrigin 是 MQEVO_MSG 时。

EventApplType

描述: 对于作为消息 (MQEVO_MSG) 接收的命令，这是来自命令消息的 MD 的应用程序类型 (PutAppl 类型)。

标识: MQIACF_EVENT_APPL_TYPE。

数据类型: MQCFIN。

已返回: 仅当 EventOrigin 是 MQEVO_MSG 时。

EventApplName

描述: 对于作为消息 (MQEVO_MSG) 接收的命令，这是来自命令消息 MD 的应用程序的名称 (PutApplName)。

标识: MQCACF_EVENT_APPL_NAME。

数据类型: MQCFST。

最大长度: MQ_APPL_NAME_LENGTH。

已返回: 仅当 EventOrigin 是 MQEVO_MSG 时。

EventApplOrigin

描述: 对于作为消息 (MQEVO_MSG) 接收的命令，这是来自命令消息的 MD 的应用程序源数据 (ApplOrigin 数据)。

标识: MQCACF_EVENT_APPL_ORIGIN。

数据类型: MQCFST。

最大长度: MQ_APPL_ORIGIN_DATA_LENGTH。

已返回: 仅当 EventOrigin 是 MQEVO_MSG 时。

命令

描述: 命令代码。

标识: MQIACF_COMMAND。

数据类型: MQCFIN。

值: **MQCMD_INQUIRE_PROT_POLICY 数字值 205**
MQCMD_CREATE_PROT_POLICY 数字值 206
MQCMD_DELETE_PROT_POLICY 数字值 207
MQCMD_CHANGE_PROT_POLICY 数字值 208
这些在 IBM MQ 8.0 `cmqcfc.h` 中定义

已返回: 一直等在那里。

CommandData 包含组成 PCF 命令的 PCF 元素。

AMS 的配置事件消息格式

配置事件是标准 Advanced Message Security 格式的 PCF 消息。

可以在 事件消息 MQMD (消息描述符)中找到 MQMD 消息描述符的可能值。

以下是所选的 MQMD 值:

```
Format = MQFMT_EVENT
Peristence = MQPER_PERSISTENCE_AS_Q_DEF
PutApplType = MQAT_QMGR           //for both CLI and command server
```

消息缓冲区由 MQCFH 结构及其后的参数结构组成。 可以在 事件消息 MQCFH (PCF 头)中找到可能的 MQCFH 值。

以下是选定的 MQCFH 值:

```
Type = MQCFT_EVENT
Command = MQCMD_CONFIG_EVENT
MsgSeqNumber = 1 or 2          // 2 will be in case of Change Object event
Control = MQCFC_LAST or MQCFC_NOT_LAST     //MQCFC_NOT_LAST will be in case of 1 Change Object
event
ParameterCount = reflects number of PCF parameters following MQCFH
CompCode = MQCC_WARNING
Reason = one of {MQRC_CONFIG_CREATE_OBJECT, MQRC_CONFIG_CHANGE_OBJECT,
MQRC_CONFIG_DELETE_OBJECT}
```

MQCFH 之后的参数为:

EventUserID

描述: 发出生成该事件的命令或调用的用户标识。 (这是用于检查发出命令或调用的权限的用户标识; 对于从队列接收的命令，这也是来自命令消息的 MD 的用户标识 (UserIdentifier)。

标识: **MQCACF_EVENT_USER_ID**

数据类型: MQCFST。

最大长度: MQ_USER_ID_LENGTH。

已返回: 一直等在那里。

SecurityId

描述: MQMD.AccountingToken 或本地命令的 Windows SID。

标识: **MQBACF_EVENT_SECURITY_ID**

数据类型: MQCBS。

最大长度: MQ_SECURITY_ID_LENGTH。

已返回: 一直等在那里。

EventOrigin

描述: 导致事件的操作的来源。

标识: **MQIACF_EVENT_ORIGIN**

数据类型: MQCFIN。

值:

MQEVO_CONSOLE
控制台命令-命令行。

MQEVO_MSG
来自 IBM MQ Explorer 插件的命令消息。

已返回: 一直等在那里。

EventQMgr

描述: 输入命令或调用的队列管理器。 (执行命令并生成事件的队列管理器位于事件消息的 MD 中)。

标识: **MQCACF_EVENT_Q_MGR**

数据类型: MQCFST

最大长度: MQ_Q_MGR_NAME_LENGTH

已返回: 一直等在那里。

ObjectType

描述: 对象类型。

标识: **MQIACF_OBJECT_TYPE**

数据类型: MQCFIN

值:

MQOT_PROT_POLICY
Advanced Message Security 保护策略。 **1019** -在 IBM MQ 8.0 或 `cmqc.h` 文件中定义的数字值。

已返回: 一直等在那里。

PolicyName

描述: Advanced Message Security 策略名称。

标识: **MQCA_POLICY_NAME**。

数据类型: MQCFST。

值: **2112** -在 IBM MQ 8.0 或 `cmqc.h` 文件中定义的数字值。

最大长度: MQ_OBJECT_NAME_LENGTH。

已返回: 一直等在那里。

PolicyVersion

描述: Advanced Message Security 策略版本。

标识: **MQIA_POLICY_VERSION**

数据类型: MQCFIN

值 **238** -在 IBM MQ 8.0 或 `cmqc.h` 文件中定义的数字值。

已返回: 始终

TolerateFlag

描述: Advanced Message Security 策略容错标志。

标识: **MQIA_ALLOWED 不受保护**

数据类型: MQCFIN

值 **235** -在 IBM MQ 8.0 或 `cmqc.h` 文件中定义的数字值。

已返回: 一直等在那里。

SignatureAlgorithm

描述: Advanced Message Security 策略签名算法。

标识: **MQIA_SIGNATURE_ALGORITHM**

数据类型: MQCFIN

值: **236** -在 IBM MQ 8.0 或 `cmqc.h` 文件中定义的数字值。

已返回: 只要 Advanced Message Security 策略中定义了签名算法

EncryptionAlgorithm

描述: Advanced Message Security 策略加密算法。

标识: **MQIA_ENCRYPTION_ALGORITHM**

数据类型: MQCFIN

值: **237** -在 IBM MQ 8.0 或 `cmqc.h` 文件中定义的数字值。

已返回: 只要 IBM MQ 策略中定义了加密算法

SignerDNs

描述: 允许签署者的主题 DistinguishedName 。

标识: **MQCA_SIGNER_DN**

数据类型: MQCFSL

值: **2113** -在 IBM MQ 8.0 或 `cmqc.h` 文件中定义的数字值。

最大长度: 策略中的最长签署者 DN ，但不再是 MQ_专有名称长度

已返回: 无论何时在 IBM MQ 策略中定义。

RecipientDNs

描述: 允许签署者的主题 DistinguishedName 。

标识: **MQCA_RECIPIENT_DN**

数据类型: MQCFSL

值: **2114** -在 IBM MQ 8.0 或 `cmqc.h` 文件中定义的数字值。

最大长度: 策略中的最长接收方 DN ，但不再是 MQ_区别 ished_name_length。

已返回: 无论何时在 IBM MQ 策略中定义。

声明

本信息是为在美国国内供应的产品和服务而编写的。

IBM 可能在其他国家或地区不提供本文档中讨论的产品、服务或功能。 有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。 任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。 只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。 但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 可能已拥有或正在申请与本文档内容有关的各项专利。 提供本文档并未授予用户使用这些专利的任何许可。 您可以用书面方式将许可查询寄往：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区: International Business Machines Corporation “按现状” 提供本出版物，不附有任何种类的 (无论是明示的还是暗含的) 保证，包括但不限于暗含的有关非侵权，适销和适用于某种特定用途的保证。 某些国家或地区在某些交易中不允许免除明示或暗含的保证。 因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。 此处的信息将定期更改；这些更改将编入本资料的新版本中。 IBM 可以随时对本资料中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。 那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Corporation
软件互操作性协调员，部门 49XA
北纬 3605 号公路
罗切斯特，明尼苏达州 55901
U.S.A.

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。 因此，在其他操作环境中获得的数据可能会有明显的不同。 有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。 此外，有些测量是通过推算而估计的， 实际结果可能会有差异。 本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。 IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。 有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息包含日常商业运作所使用的数据和报表的示例。 为了尽可能全面地说明这些数据和报表，这些示例包括个人、公司、品牌和产品的名称。 所有这些名字都是虚构的，若现实生活中实际业务企业使用的名字和地址与此相似，纯属巧合。

版权许可：

本信息包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。 如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。 这些示例并未在所有条件下作全面测试。 因此，IBM 不能担保或默示这些程序的可靠性、可维护性或功能。

如果您正在查看本信息的软拷贝，图片和彩色图例可能无法显示。

编程接口信息

编程接口信息 (如果提供) 旨在帮助您创建用于此程序的应用软件。

本书包含有关允许客户编写程序以获取 IBM MQ 服务的预期编程接口的信息。

但是，该信息还可能包含诊断、修改和调优信息。 提供诊断、修改和调优信息是为了帮助您调试您的应用程序软件。

要点: 请勿将此诊断，修改和调整信息用作编程接口，因为它可能会发生更改。

商标

IBM IBM 徽标 ibm.com 是 IBM Corporation 在全球许多管辖区域的商标。 当前的 IBM 商标列表可从 Web 上的 “Copyright and trademark information”www.ibm.com/legal/copytrade.shtml 获取。 其他产品和服务名称可能是 IBM 或其他公司的商标。

Microsoft 和 Windows 是 Microsoft Corporation 在美国和/或其他国家或地区的商标。

UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。

Linux 是 Linus Torvalds 在美国和/或其他国家或地区的商标。

此产品包含由 Eclipse 项目 (https://www.eclipse.org/) 开发的软件。

Java 和所有基于 Java 的商标和徽标是 Oracle 和/或其附属公司的商标或注册商标。

部件号:

(1P) P/N: