7.5

保护 *IBM WebSphere MQ*

注

在使用本资料及其支持的产品之前，请阅读第 275 页的『声明』中的信息。

此版本适用于 IBM® WebSphere MQ V 7 发行版 5 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。

当您向 IBM 发送信息时，授予 IBM 以它认为适当的任何方式使用或分发信息的非独占权利，而无需对您承担任何责任。

内容

安全性

对于 IBM WebSphere MQ 应用程序开发者和配置 IBM WebSphere MQ 权限的系统管理员而言，安全性都是一个重要的考虑因素。

安全性概述

此主题集合介绍了 IBM WebSphere 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 WebSphere MQ 中实现的这些安全机制。

安全概念和机制

此主题集合描述了要在 IBM WebSphere MQ 安装中考虑的安全性方面。

常见的安全方面如下:

- 第 5 页的『标识和认证』
- 第 6 页的『Authorization』
- 第 6 页的『审计』
- 第 6 页的『保密性』
- 第 6 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。 一个机制可以单独运作，也可以与其他机制一起运作，以提供特定服务。 常见安全机制的示例如下:

- 第 7 页的『密码术』
- 第 8 页的『消息摘要和数字签名』
- 第 9 页的『数字证书』
- 第 12 页的『公共密钥基础结构 (PKI)』

在规划 IBM WebSphere 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 38 页的『规划安全需求』。

相关概念

第 171 页的『使用 SSL 或 TLS 连接两个队列管理器』
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。

第 92 页的『使用 SSL 或 TLS』
这些主题提供了有关执行与将 SSL 或 TLS 与 IBM WebSphere MQ 配合使用相关的单个任务的指示信息。

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。

不可抵赖性

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。

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消息，那么具有 交付证明 的不可抵赖性服务可以向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艰难的目标。 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 WebSphere MQ 环境中的相关安全服务，因为 IBM WebSphere MQ 是一种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 WebSphere MQ with IBM WebSphere MQ 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 WebSphere MQ 环境中提供您自己的不可抵赖性服务的建议。

相关概念

第 18 页的『IBM WebSphere MQ 中的标识和认证』
在 IBM WebSphere MQ 中，可以使用消息上下文信息和相互认证来实现标识和认证。

Authorization

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。

相关概念

第 18 页的『IBM WebSphere MQ 中的授权』
您可以使用授权来限制特定个人或应用程序可以在 IBM WebSphere MQ 环境中执行的操作。

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。

有关如何设置授权的更多信息，请参阅 第 41 页的『规划授权』 和关联的子主题。

相关概念

第 18 页的『在 IBM WebSphere MQ 中进行审计』
IBM WebSphere MQ 可以发出事件消息以记录已发生异常活动。

保密性

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。

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。

加密概念

此主题集合描述适用于 WebSphere MQ 的密码术概念。

术语 实体 用于指队列管理器， WebSphere MQ MQI 客户机，单个用户或任何其他能够交换消息的系统。

相关概念

第 19 页的『IBM WebSphere MQ 中的密码术』

IBM WebSphere MQ 通过使用安全套接字层 (SSL) 和传输安全层 (TLS) 协议提供密码术。

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)。

请参阅 词汇表 以获取密码术的定义。

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 16 页的『SSL 和 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 7 页的图 1 说明对称密钥密码术。
- 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 8 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。

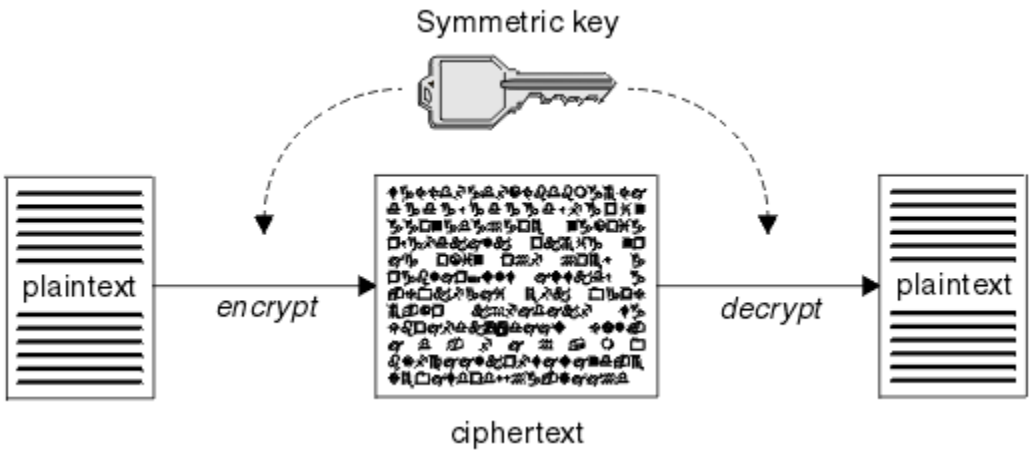


图 *1:* 对称密钥密码术 *(symmetric key cryptography)*

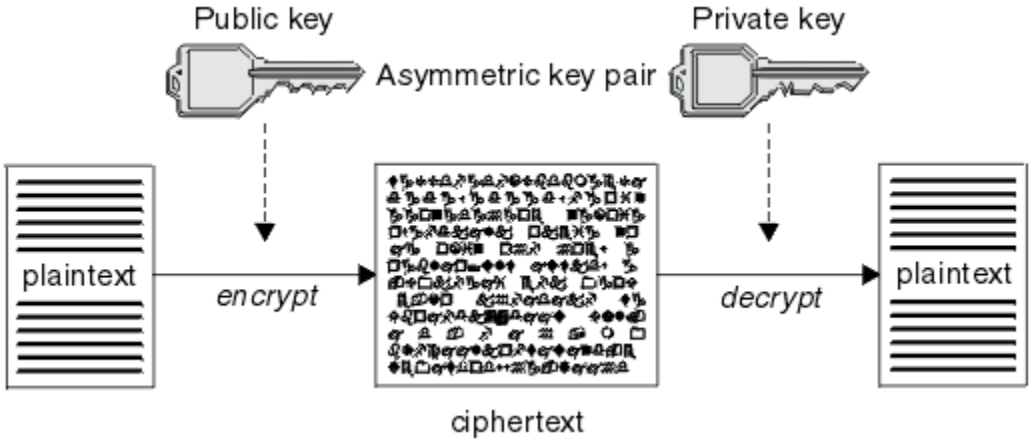


图 *2:* 非对称密钥密码术 *(asymmetric key cryptography)*

第 8 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

消息摘要是由散列函数计算的消息内容的固定大小数字表示。 可以对消息摘要进行加密，形成数字签名。

消息的大小本来是可变的。 消息摘要是消息内容的固定大小数字表示。 消息摘要由散列函数计算，该函数是满足两个条件的变换:

- 散列函数必须是单向的。 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 16 页的『SSL 和 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 10 页的『认证中心』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。

WebSphere MQ 所使用的数字证书符合 X.509 标准，该标准指定所需信息以及发送该信息的格式。 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。 如果扩展不是关键扩展，那么当无法识别该字段时，系统可以忽略该字段。

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

WebSphere 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 WebSphere MQ 是对等系统，因此它在 SSL 术语中被视为客户机认证。 因此，用于 SSL 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 WebSphere MQ 中的证书验证策略的更多信息，请参阅 第 28 页的『IBM WebSphere MQ 中的证书验证策略』。

认证中心

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 124 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。

通常可在 DN 中找到以下属性类型：

SERIALNUMBER	证书序列号
MAIL	电子邮件地址
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 WebSphere MQ 允许某些格式不正确的 DN。 有关更多信息，请参阅 SSLPEER 值的 WebSphere MQ 规则。

相关概念

第 9 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 (例如， UNIX 上的 iKeyman 工具， Linux® 以及 z/OS 上的 Windows 系统和 RACF) 生成。 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 11 页的图 3 解释了从 CA 获取数字证书的过程。

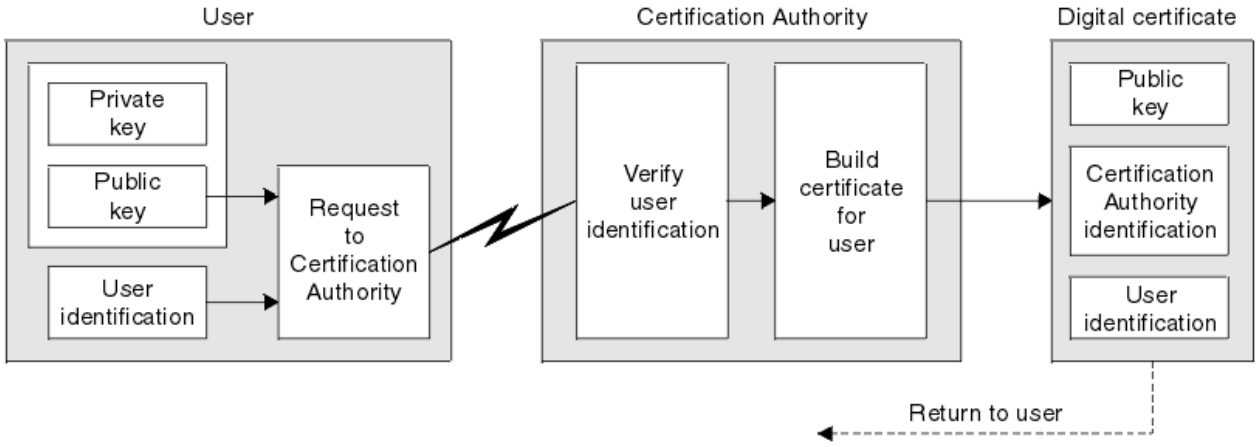


图 *3:* 获取数字证书

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 9 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 12 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。

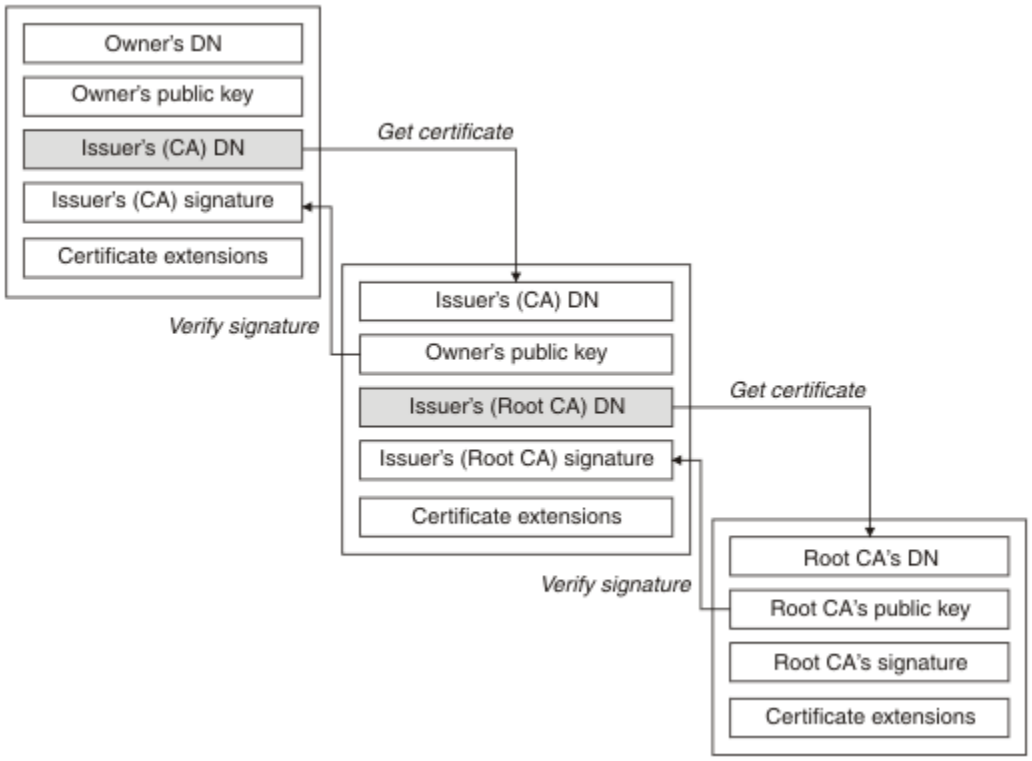


图 *4:* 信任链

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。

请参阅 词汇表 以获取证书到期的定义。

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。

WebSphere MQ 可以通过向联机证书状态协议 (OCSP) 响应程序 (仅在 UNIX， Linux 和 Windows 系统上) 发送请求来检查是否撤销了证书。 或者，他们可以访问 LDAP 服务器上的 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 124 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 9 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：SSL 和 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 WebSphere MQ 同时支持 SSL 和 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 19 页的『IBM WebSphere MQ 中的安全协议』

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

安全套接字层 *(SSL)* 和传输层安全性 *(TLS)* 概念

SSL 和 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。

SSL 和 TLS 协议可提供因特网通信安全性，并允许客户机/服务器应用程序以保密且可靠的方式进行通信。 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。

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

第 182 页的『指定 CipherSpec』提供了 IBM WebSphere MQ 支持的 CipherSpecs 的完整列表。

有关 SSL 协议的更多信息，请参阅 https://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt 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 站点上提供的信息，网址为 https://www.ietf.org

SSL 或 *TLS* 握手概述

SSL 或 TLS 握手使 SSL 或 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 SSL 或 TLS 客户机与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， SSL 或 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， SSL 握手所涉及的步骤如下所示:

1. SSL 或 TLS 客户机发送 “client hello” 消息，其中列出了诸如 SSL 或 TLS 版本之类的加密信息，并且按客户机的首选顺序列出了客户机支持的 CipherSuites 。 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. SSL 或 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 ，会话标识和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. SSL 或 TLS 客户机验证服务器的数字证书。 有关更多信息，请参阅第 14 页的『SSL 和 TLS 如何提供标识，认证，机密性和完整性』。

4. SSL 或 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 SSL 或 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. SSL 或 TLS 服务器验证客户机的证书。 有关更多信息，请参阅第 14 页的『SSL 和 TLS 如何提供标识，认证，机密性和完整性』。
7. SSL 或 TLS 客户机向服务器发送 “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. SSL 或 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 SSL 或 TLS 会话的持续时间内，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 14 页的图 5 说明了 SSL 或 TLS 握手。

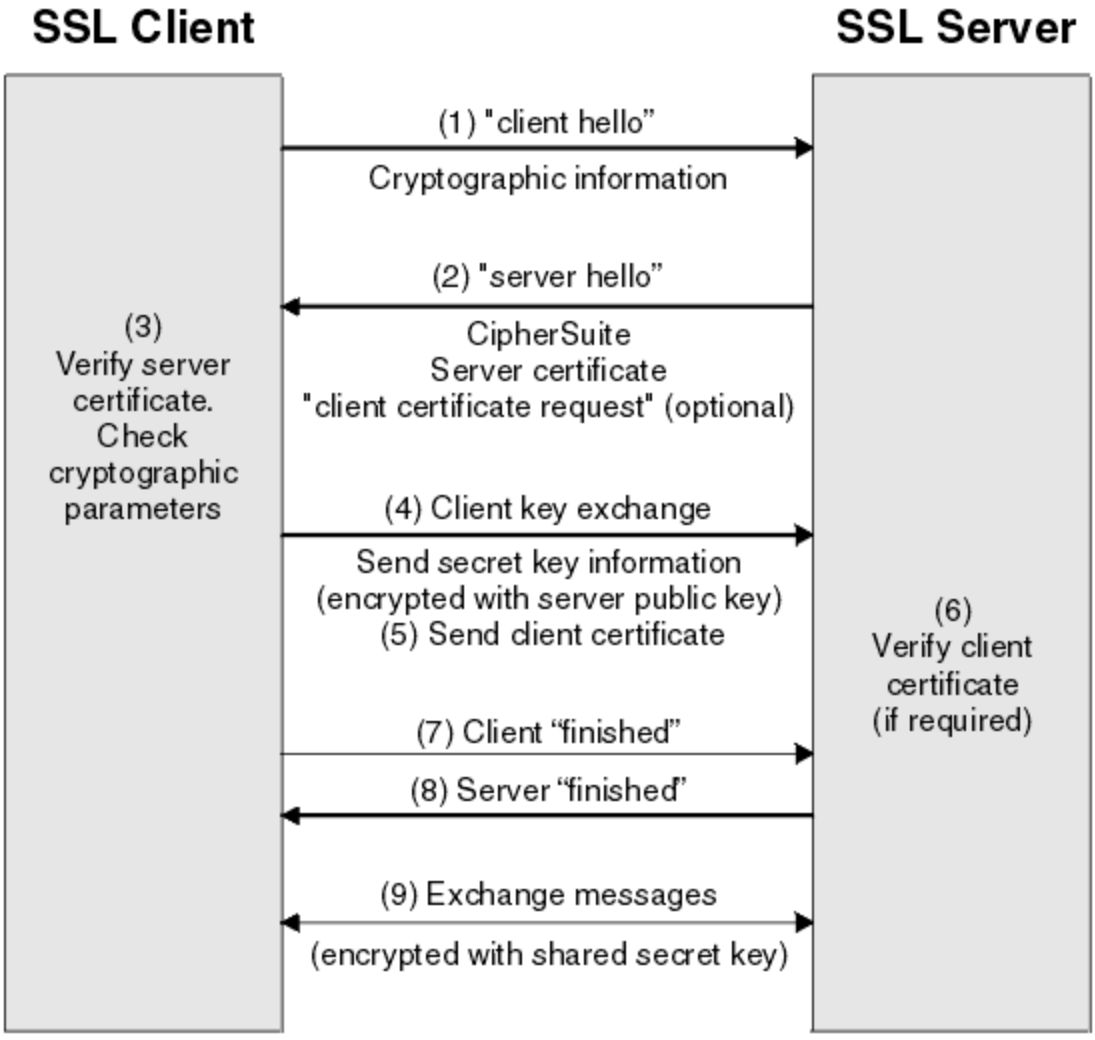


图 *5: SSL* 或 *TLS* 握手概述

SSL 和 *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 13 页的『SSL 或 TLS 握手概述』。

SSL 和 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。

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 14 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 14 页的『7』 和 第 14 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 SSLor 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 SSL 或 TLS 客户机， CA Y 将证书发放到 SSL 或 TLS 服务器:

仅对于服务器认证， SSL 或 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

以及 SSL 或 TLS 客户机需要:

- CA Y 的 CA 证书

如果 SSL 或 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

SSL 或 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 13 页的『3』 和 第 14 页的『6』 中所述， SSL 或 TLS 客户机将验证服务器的证书，而 SSL 或 TLS 服务器将验证客户机的证书。 此验证有四个方面:

1. 检查数字签名 (请参阅 第 16 页的『SSL 和 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 11 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 124 页的『使用已撤销证书』)。

密钥重置

在 SSL 或 TLS 握手期间，将生成 密钥 以对 SSL 或 TLS 客户机与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 8 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。

SSL 和 TLS 如何提供机密性

SSL 和 TLS 使用对称和非对称加密的组合来确保消息隐私。 在 SSL 或 TLS 握手期间， SSL 或 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。 SSL 或 TLS 客户机与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，从而确保即使消息被拦截也保持私有。 SSL 支持广泛的密码算法。 由于 SSL 和 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 7 页的『密码术』。

SSL 和 TLS 如何提供完整性

SSL 和 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 189 页的『消息的数据完整性』。

使用 SSL 或 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 182 页的『指定 CipherSpec』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 或 SSL 连接的两端必须同意相同的 CipherSpec 才能进行通信。

要点: 处理 IBM WebSphere 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 182 页的『指定 CipherSpec』。

CipherSuite 是由 SSL 或 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 或 SSL 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 SSL_RSA_WITH_RC4_128_MD5 指定:

- RSA 密钥交换和认证算法
- RC4 加密算法，使用 128 位密钥
- MD5 MAC 算法

有几种算法可用于密钥交换和认证，但 RSA 算法是目前应用最广泛的算法。 在所使用的加密算法和 MAC 算法中有更多种。

SSL 和 *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 17 页的图 6 演示了此过程。

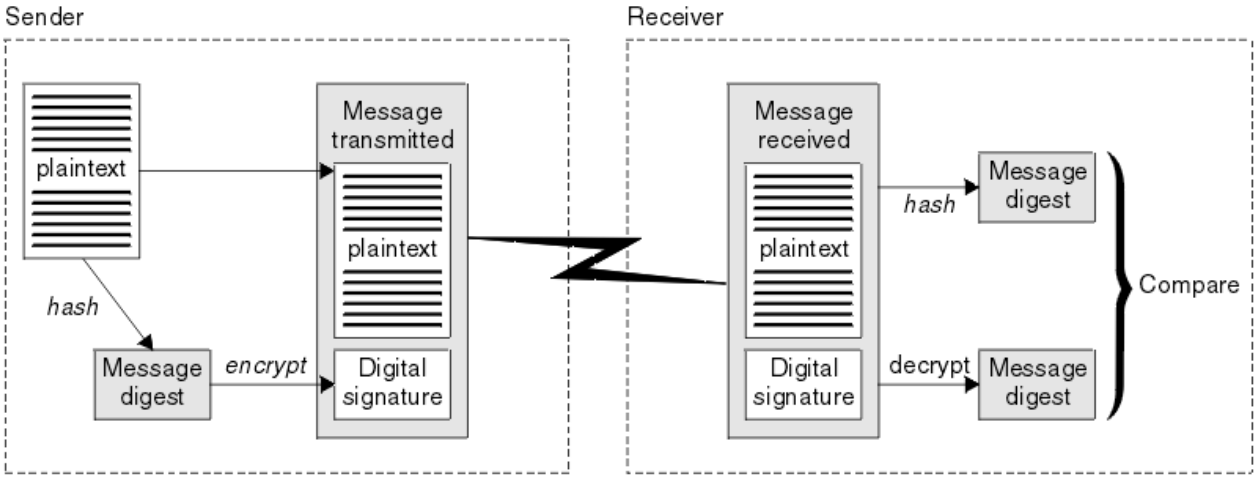


图 *6:* 数字签名过程

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。

IBM WebSphere 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。

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 TLS V 1.2 配合使用的符合 Suite B 的概要文件。 当针对符合 Suite B 的操作进行配置时，将仅使用上面列出的一组受限密码算法。
2. 用于 TLS 版本 1.0 或 TLS 版本 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 Windows 上，可以将 UNIX 和 Linux 系统 WebSphere MQ 配置为符合符合符合 Suite B 的 TLS 1.2 概要文件，但不支持 Suite B 过渡概要文件。 有关更多信息，请参阅第 25 页的『IBM WebSphere MQ 中的 NSA Suite B 密码术』。

相关信息

第 17 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 WebSphere MQ 安全性机制

此主题集合说明如何在 IBM WebSphere MQ 中实现各种安全概念。

IBM WebSphere MQ 提供了实现 第 5 页的『安全概念和机制』中引入的所有安全概念的机制。 以下部分更详细地讨论了这些问题。

IBM WebSphere MQ 中的标识和认证

在 IBM WebSphere MQ 中，可以使用消息上下文信息和相互认证来实现标识和认证。

以下是 IBM WebSphere MQ 环境中的标识和认证的一些示例:

- 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中。 当应用程序将消息放入队列时，队列管理器可以生成此消息。 或者，如果与应用程序关联的用户标识已获得授权，那么应用程序可以提供该信息。

  消息中的上下文信息允许接收应用程序查找有关消息发起方的信息。 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。
- 当消息通道启动时，通道两端的消息通道代理程序 (MCA) 可以对其合作伙伴进行认证。 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。 对于接收 MCA，有类似的保证，即将收到来自真正合作伙伴的消息。

相关概念

第 5 页的『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。

IBM WebSphere MQ 中的授权

您可以使用授权来限制特定个人或应用程序可以在 IBM WebSphere MQ 环境中执行的操作。

以下是 IBM WebSphere 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 WebSphere 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 41 页的『规划授权』和关联的子主题。

相关概念

第 6 页的『Authorization』
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。

在 IBM WebSphere MQ 中进行审计

IBM WebSphere MQ 可以发出事件消息以记录已发生异常活动。

以下是 IBM WebSphere 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 SSL 不允许连接。 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，并可以确定需要执行的操作。

相关概念

第 6 页的『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。

IBM WebSphere MQ 中的机密性

您可以通过加密消息在 IBM WebSphere MQ 中实现机密性。

以下是如何在 IBM WebSphere MQ 环境中确保机密性的一些示例:

- 发送 MCA 从传输队列获取消息后， IBM WebSphere MQ 会使用 SSL 或 TLS 对消息进行加密，然后再通过网络将其发送到接收 MCA。 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 WebSphere 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 IBM WebSphere MQ Advanced Message Security 对存储在队列中的消息进行加密。

相关概念

第 6 页的『保密性』
机密性 服务可防止敏感信息未经授权的泄露。

IBM WebSphere MQ 中的数据完整性

您可以使用数据完整性服务来检测是否已修改消息。

以下是如何在 IBM WebSphere MQ 环境中确保数据完整性的一些示例:

- 您可以使用 SSL 或 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 SSL 和 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 WebSphere MQ CipherSpecs 都提供消息摘要算法，但不提供消息数据完整性的 TLS_RSA_WITH_NULL_NULL 除外。
- 当消息存储在本地队列上时， IBM WebSphere MQ 提供的访问控制机制可能被视为足以防止有意修改消息内容。 但是，为了获得更高级别的安全性，您可以使用 IBM WebSphere MQ 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。

相关概念

第 6 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。

IBM WebSphere MQ 中的密码术

IBM WebSphere MQ 通过使用安全套接字层 (SSL) 和传输安全层 (TLS) 协议提供密码术。

有关更多信息，请参阅第 19 页的『IBM WebSphere MQ 中的安全协议』。

相关概念

第 6 页的『加密概念』
此主题集合描述适用于 WebSphere MQ 的密码术概念。

IBM WebSphere MQ 中的安全协议

IBM WebSphere MQ 支持传输层安全性 (TLS) 和安全套接字层 (SSL) 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 SSL 或 TLS 协议来提供链路级别安全性。调用者 MCA 是 SSL 或 TLS 客户机，响应者 MCA 是 SSL 或 TLS 服务器。 WebSphere MQ 支持 SSL 协议的 V 3.0 和传输层安全性 (TLS) 协议的 V 1.0 和 V 1.2 。 通过提供 CipherSpec 作为通道定义的一部分，指定 SSL 或协议所使用的密码算法。

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 SSL 或 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 通道客户机端的 WebSphere MQ 代码代表 WebSphere MQ 客户机应用程序的用户执行操作。 在 SSL 或 TLS 握手期间，WebSphere MQ 代码会将用户的数字证书发送到 MQI 通道的服务器端的 MCA。

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

数字证书存储在 密钥存储库 中。 队列管理器属性 *SSLKeyRepository* 指定保存队列 管理器的 数字证书的密钥存储库的位置。 在 WebSphere MQ 客户机系统上， MQSSLKEYR 环境变量指定保存 用户的 数字证书的密钥存储库的位置。 或者， WebSphere MQ 客户机应用程序可以在 MQCONNX 调用上的 SSL 和 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。

相关概念

第 13 页的『加密安全性协议：SSL 和 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 WebSphere MQ 同时支持 SSL 和 TLS。

IBM WebSphere MQ 支持 *SSL* 和 *TLS*

IBM WebSphere MQ 支持安全套接字层 (SSL) 协议和传输层安全性 (TLS) 协议。

有关 SSL 和 TLS 协议的更多信息，请参阅相关信息。

IBM WebSphere MQ 为 SSL V 3.0 和 TLS 1.0 以及 TLS 1.2:

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

UNIX, Linux, and Windows 和 HP Integrity NonStop Server 系统
对于 UNIX, Linux, and Windows 和 HP Integrity NonStop Server 系统， SSL 和 TLS 支持随 IBM WebSphere MQ 一起安装。

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

SSL 或 *TLS* 密钥存储库
相互认证的 SSL 或 TLS 连接要求在连接的每一端都有一个密钥存储库 (可以通过不同平台上的不同名称来识别)。 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。 在支持 SSL 和 TLS 的平台和环境上使用的特定商店名称包括:

Java 和 JMS	密钥库和信任库
Windows UNIX Linux Windows UNIX Linux	密钥数据库文件
Windows， UNIX and Linux 系统	

有关更多信息，请参阅 第 9 页的『数字证书』 和 第 13 页的『安全套接字层 (SSL) 和传输层安全性 (TLS) 概念』。

相互认证的 SSL 或 TLS 连接在连接的每一端都需要密钥存储库。 密钥存储库可能包含:

- 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 WebSphere MQ MQI 客户机相关联。 如果需要相互认证，那么个人证书在 SSL 或 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 21 页的『保护 IBM WebSphere 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:

Windows， UNIX and Linux 系统

在 Windows 上， UNIX and Linux 系统上的密钥存储库是密钥数据库文件。 密钥数据库文件的名称必须具有文件扩展名 `.kdb`。 例如，在 UNIX 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 WebSphere MQ 安装在缺省位置，那么 Windows 上的等效路径为 `C:\Program Files\IBM\WebSphere MQ\Qmgrs\QM1\ssl\key.kdb`。

在 Windows UNIX and Linux 系统上，每个密钥数据库文件都具有关联的密码隐藏文件。 此文件包含允许程序访问密钥数据库的编码密码。 密码隐藏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干，并且必须以后缀 `.sth` 结尾，例如 `/var/mqm/qmgrs/QM1/ssl/key.sth`

注: 在 Windows UNIX and Linux 系统上， PKCS #11 加密硬件卡可以包含在密钥数据库文件中以其他方式保存的证书和密钥。 在 PKCS #11 卡上保存证书和密钥时， WebSphere MQ 仍需要访问密钥数据库文件和密码存储文件。

在 Windows 和 UNIX 系统上，密钥数据库还包含与队列管理器或 WebSphere MQ MQI 客户机相关联的个人证书的专用密钥。

保护 *IBM WebSphere MQ* 密钥存储库

IBM WebSphere 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 WebSphere 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。

刷新队列管理器的密钥存储库

更改密钥存储库的内容时，队列管理器不会立即获取新内容。 要使队列管理器使用新的密钥存储库内容，必须发出 REFRESH SECURITY TYPE (SSL) 命令。

此过程是有意的，可防止出现多个正在运行的通道可能使用不同版本的密钥存储库的情况。 作为安全控件，队列管理器随时只能装入密钥存储库的一个版本。

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

您还可以使用 PCF 命令或 WebSphere 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 WebSphere MQ Explorer 部分中的 MQCMD_REFRESH_SECURITY 命令 和主题 刷新 *SSL* 或 *TLS* 安全性 。

相关概念

第 21 页的『刷新客户机的 SSL 密钥存储库内容和 SSL 设置视图』
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* 密钥存储库内容和 *SSL* 设置视图

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 WebSphere 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 刷新安全性) 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 21 页的『刷新队列管理器的密钥存储库』
更改密钥存储库的内容时，队列管理器不会立即获取新内容。 要使队列管理器使用新的密钥存储库内容，必须发出 REFRESH SECURITY TYPE (SSL) 命令。

联邦信息处理标准 *(FIPS)*

本主题介绍了美国国家标准技术学会的联邦信息处理标准 (FIPS) 加密验证程序以及可用于 Windows， UNIX and Linux 和 z/OS 系统的 SSL 或 TLS 通道上的加密功能。

此处提供了 UNIX， Linux 和 Windows 系统上 IBM WebSphere MQ SSL 或 TLS 连接的 FIPS 140-$tag1 合规性 第 22 页的『适用于 UNIX， Linux 和 Windows 的联邦信息处理标准 (FIPS)』。

如果存在加密硬件，那么可以将 IBM WebSphere 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 WebSphere 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 90 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 95 页的『使用 iKeyman， iKeycmd， runmqakm 和 runmqckm』
在 UNIX， Linux 和 Windows 系统上，使用 iKeyman GUI 或从命令行使用 iKeycmd 或 runmqakm 来管理密钥和数字证书。

相关任务

在 WebSphere MQ Java 类中启用 SSL

将安全套接字层 (SSL) 与 WebSphere MQ JMS 类配合使用

相关参考

JMS 对象的 SSL 属性

相关信息

第 17 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)。

适用于 *UNIX*， *Linux* 和 *Windows* 的联邦信息处理标准 *(FIPS)*

在 Windows UNIX and Linux 系统上的 SSL 或 TLS 通道上需要密码术时， WebSphere MQ 使用名为 IBM Crypto for C (ICC) 的密码术包。 在 Windows UNIX and Linux 平台上， ICC 软件已通过美国国家标准与技术研究所的联邦信息处理标准 (FIPS) Cryptomodule Validation Program ，级别为 140-2。

在 Windows UNIX and Linux 系统上， WebSphere MQ SSL 或 TLS 连接的 FIPS 140-$tag1 合规性如下所示:

- 对于所有 IBM WebSphere 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:
  - 已安装的 GSKit ICC 版本在已安装的操作系统版本和硬件体系结构上已通过 FIPS 140-2 认证。
  - 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。
- 对于所有 IBM WebSphere MQ MQI 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:
  - 已安装的 GSKit ICC 版本在已安装的操作系统版本和硬件体系结构上已通过 FIPS 140-2 认证。
  - 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。
- 对于使用客户机方式的 Java 应用程序的 IBM WebSphere MQ 类，如果满足以下条件，那么连接将使用 JRE 的 SSL 和 TLS 实现，并且符合 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 JMS 应用程序的 IBM WebSphere MQ 类，如果满足以下条件，那么连接将使用 JRE 的 SSL 和 TLS 实现，并且符合 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:
  - 已安装的 GSKit ICC 版本在已安装的操作系统版本和硬件体系结构上已通过 FIPS 140-2 认证。
  - 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。
- 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:
  - 已安装的 GSKit ICC 版本在已安装的操作系统版本和硬件体系结构上已通过 FIPS 140-2 认证。
  - 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。

所有受支持的 AIX， Linux， HP-UX， Solaris， Windows 和 z/OS 平台都经过 FIPS 140-2 认证，但每个修订包或更新包随附的自述文件中都有说明。

对于使用 GSKit 的 SSL 和 TLS 连接， FIPS 140-$tag1 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** 输出的示例摘录:

```
  Icc
 ============
@ (#)CcompanyName: 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 (包括在 GSKit 8 中) 的 NIST 认证语句: Cryptographic Module Validation Program。

如果存在加密硬件，那么可以将 IBM WebSphere 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。

注: 在 Intel 系统上运行时，为符合 FIPS 140-2 标准的操作配置的 32 位 Solaris x86 SSL 和 TLS 客户机将失败。 由于未在 Intel 芯片上装入符合 FIPS 140-2 的 GSKit-Crypto Solaris x86 32 位库文件，因此会发生此故障。 在受影响的系统上，将在客户机错误日志中报告错误 AMQ9655。 要解决此问题，请禁用 FIPS 140-2 合规性或重新编译客户机应用程序 64 位，因为 64 位代码不受影响。

在符合 FIPS 140-2 的情况下运行时实施三重 DES 限制

当 WebSphere MQ 配置为符合 FIPS 140-2 运行时，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， WebSphere 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 187 页的『重置 SSL 和 TLS 密钥』。

WebSphere 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 90 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 95 页的『使用 iKeyman， iKeycmd， runmqakm 和 runmqckm』
在 UNIX， Linux 和 Windows 系统上，使用 iKeyman GUI 或从命令行使用 iKeycmd 或 runmqakm 来管理密钥和数字证书。

相关任务

在 WebSphere MQ Java 类中启用 SSL

将安全套接字层 (SSL) 与 WebSphere MQ JMS 类配合使用

相关参考

JMS 对象的 SSL 属性

相关信息

第 17 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 WebSphere MQ MQI 客户机上的 *SSL* 和 *TLS*

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

IBM WebSphere MQ 在 Windows UNIX and Linux 系统上为 IBM WebSphere MQ MQI 客户机提供 SSL 和 TLS 支持。 如果使用 IBM WebSphere MQ classes for Java ，请参阅 使用 WebSphere MQ classes for Java ，如果使用 IBM WebSphere MQ classes for JMS ，请参阅 使用 WebSphere MQ classes for JMS。 本节的其余部分不适用于 Java 或 JMS 环境。

您可以使用 IBM WebSphere 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 WebSphere MQ MQI 客户机指定密钥存储库。 您有三个选项可用于指定通道使用 SSL:

- 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 SSL。

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

如果在客户机上对 SSL 密钥存储库的内容， SSL 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 SSL 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 SSL 通道。 将使用所有新的 SSL 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 WebSphere MQ MQI 客户机在具有加密硬件的 Windows UNIX and Linux 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。

在 IBM WebSphere MQ MQI 客户机上支持 SSL 密钥重置和 FIPS。 有关更多信息，请参阅第 187 页的『重置 SSL 和 TLS 密钥』和第 22 页的『适用于 UNIX， Linux 和 Windows 的联邦信息处理标准 (FIPS)』。

请参阅 第 89 页的『设置 IBM WebSphere MQ MQI 客户机安全性』， 以获取有关 IBM WebSphere MQ MQI 客户机的 SSL 支持的更多信息。

相关任务

使用配置文件配置客户机

指定 *MQI* 通道使用 *SSL*

要使 MQI 通道使用 SSL，客户机连接通道的 *SSLCipherSpec* 属性的值必须是客户机平台上 IBM WebSphere 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 WebSphere MQ MQI 客户机应用程序发出 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 调用上提供 MQCD 结构中的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 WebSphere MQ 客户机访问的任何条目。

不能使用 MQSERVER 环境变量在使用 SSL 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 187 页的『为 IBM WebSphere MQ MQI 客户机指定 CipherSpec』

您有三个选项可用于为 IBM WebSphere MQ MQI 客户机指定 CipherSpec 。

IBM WebSphere MQ 中的 *CipherSpecs* 和 *CipherSuites*

IBM WebSphere MQ 支持 SSL 和 TLS CipherSpecs 以及 RSA 和 Diffie-Hellman 算法。

WebSphere MQ 支持 SSL V3 和 TLS V1.0 以及 V1.2 CipherSpecs。

WebSphere MQ 支持 RSA 和 Diffie-Hellman 密钥交换和认证算法。 SSL 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 16 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 WebSphere MQ 中的 *NSA Suite B* 密码术

本主题提供有关如何在 Windows， Linux 和 UNIX 系统上配置 IBM WebSphere MQ 以符合套件 B 兼容 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 WebSphere 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 WebSphere MQ Version 7.5 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 WebSphere MQ 以实施 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 READMEs)。

在 Windows， UNIX 和 Linux 系统上，可以将 IBM WebSphere MQ 配置为符合表 1 中显示的安全级别的符合套件 B 的 TLS 1.2 概要文件。

表 1: 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
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 WebSphere MQ 的缺省行为是不符合 Suite B 标准，但可以将 IBM WebSphere MQ 配置为符合 Windows， UNIX 和 Linux 系统上的一个或两个安全级别。 在成功配置 IBM WebSphere 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 WebSphere MQ CipherSpecs 的更多信息，请参阅 第 182 页的『指定 CipherSpec』

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 WebSphere 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 runmqckm 和 runmqakm 选项。

注: **iKeycmd** 和 **iKeyman** 工具不支持为符合 Suite B 的操作创建数字证书。

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 101 页的『在 UNIX, Linux, and Windows 系统上创建自签名个人证书』

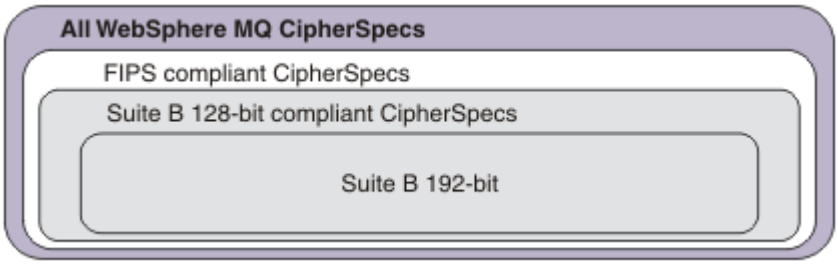
要请求 CA 签署的数字证书以供 Suite B 生产使用，请参阅 第 103 页的『在 UNIX, Linux, and Windows 系统上请求个人证书』。

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

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 WebSphere MQ 时，可以使用当前支持的 Suite B

CipherSpecs 。 因此，可以同时配置 WebSphere MQ 以实现 FIPS 和 Suite B 合规性，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 WebSphere MQ

有关如何在 Windows， UNIX 和 Linux 上为符合 Suite B 的操作配置 IBM WebSphere MQ 的信息，请参阅第 27 页的『为 Suite B 配置 IBM WebSphere MQ』。

IBM WebSphere MQ 不支持在 IBM i 和 z/OS 平台上执行符合 Suite B 的操作。 WebSphere MQ Java 和 JMS 客户机也不支持符合 Suite B 的操作。

相关概念

第 90 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。

为 *Suite B* 配置 *IBM WebSphere MQ*
IBM WebSphere MQ 可以配置为在 UNIX, Linux, and Windows 系统上按照 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。 IBM WebSphere MQ 可配置为与 Suite B 合规运行，以提供增强的安全级别。 有关 Suite B 的更多信息，请参阅第 17 页的『国家安全局 (NSA) 套件 B 密码术』。 有关 Suite B 配置及其对 SSL 和 TLS 通道的影响的更多信息，请参阅第 25 页的『IBM WebSphere 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。

您还可以使用带有 **MQIA_SUITE_B_STRENGTH** 参数的 PCF **MQCMD_CHANGE_Q_MGR** 命令为符合 Suite B 的操作配置队列管理器

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 调用的 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。
2. 通过将 MQSUITEB 环境变量设置为以下一个或多个值:
   - 无
   - 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 MQI 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:
   - 无
   - 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。

有关 MQSCO 结构的完整详细信息，请参阅 MQSCO-SSL 配置选项。

有关在客户机配置文件中使用 Suite B 的更多信息，请参阅 客户机配置文件的 SSL 节 。

有关使用 MQSUITEB 环境变量的更多信息，请参阅 环境变量。

.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 .NET 的 IBM WebSphere MQ 类中使用 Suite B 的信息，请参阅 MQEnvironment .NET 类。

IBM WebSphere 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。
- 对于 UNIX, Linux, and Windows 系统，证书验证策略由 GSKit 提供，并且可以进行配置。 支持两种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。

有关如何在 UNIX, Linux, and Windows 系统上配置证书验证策略的信息，请参阅 第 28 页的『在 IBM WebSphere MQ 中配置证书验证策略』。 有关 "基本" 和 "标准" 证书验证策略之间的差异的更多信息，请参阅 UNIX， Linux 和 Windows 系统上的证书验证和信任策略设计 。

在 *IBM WebSphere MQ* 中配置证书验证策略

您可以通过四种方式指定使用哪个 SSL/TLS 证书验证策略来验证从远程伙伴系统接收的数字证书。

在队列管理器上，可以通过以下方式设置证书验证策略:

- 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 。

在客户机上，有几种方法可用于设置证书验证策略。 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 *CertificateVal* 策略 字段。 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*。 有关使用此变量的更多信息，请参阅 MQCERTVPOL 。
3. 使用客户机 SSL 节调整参数设置 *CertificateVal* 策略。 有关使用此设置的更多信息，请参阅 客户机配置文件的 SSL 节 。

有关证书验证策略的更多信息，请参阅 第 28 页的『IBM WebSphere MQ 中的证书验证策略』。

IBM WebSphere MQ 中的数字证书和 *CipherSpec* 兼容性

本主题通过概述 CipherSpecs 与 IBM WebSphere 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。

在 IBM WebSphere MQ 的前发行版中，所有受支持的 SSL 和 TLS CipherSpecs 都将 RSA 算法用于数字签名和密钥协议。 所有受支持的数字证书类型都与所有受支持的 CipherSpecs 兼容，因此可以更改任何通道的 CipherSpec ，而无需更改数字证书。

在 IBM WebSphere MQ v7.5 中，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，需要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。 如果您希望使用基于 TLS 1.2 协议的较新 CipherSpecs ，请确保数字证书在其数字签名中不使用 MD5 算法。 使用 SSL 3.0 和 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` 是需要显示的数字签名算法的证书标签。

注: 虽然可以使用 **iKeycmd (runmqckm)** 工具和 **iKeyman (strmqikm)** GUI 来查看所选择的数字签名算法，但 **runmqakm** 工具提供了更广泛的范围。

执行 **runmqakm** 命令将生成显示使用指定的签名算法的输出:

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

`Signature Algorithm` 行显示使用了 `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。 有三种类型的 CipherSpec，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 WebSphere MQ SSL 和 TLS 连接，但与椭圆曲线密码术的用户特别相关。

下表概述了 CipherSpecs 与数字证书之间的关系:

表 *2: 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，椭圆曲线密钥和椭圆曲线数字签名算法的 CipherSpecs 。	RSA	RSA	(ECDHE)
3	(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: 类型 1 和 2 CipherSpecs 仅受 UNIX, Linux, and Windows 平台上的 WebSphere MQ 队列管理器和 MQI 客户机支持。

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。

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 "RSA with MD5" 或 "ECDSA with SHA-256"。 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 WebSphere MQ 配置为符合 FIPS 140-$tag3 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。

WebSphere MQ 队列管理器只能使用单个个人证书来标识自身。 这意味着队列管理器上的所有通道都将使用相同的数字证书，因此每个队列管理器一次只能使用一种类型的 CipherSpec 。 同样， WebSphere MQ 客户机应用程序只能使用单个个人证书来标识自身。 这意味着单个应用程序进程中的所有 SSL 和 TLS 连接都将使用相同的数字证书，因此每个客户机应用程序进程一次只能使用一种类型的 CipherSpec 。

三种类型的 CipherSpec 不会直接互操作: 这是当前 SSL 和 TLS 标准的限制。 例如，假定您已选择将 ECDHE_ECDSA_AES_128_CBC_SHA256 CipherSpec 用于队列管理器 QM1 上名为 TO.QM1 的接收方通道。 ECDHE_ECDSA_AES_128_CBC_SHA256 是类型 1 CipherSpec，因此 QM1 必须具有具有椭圆曲线密钥和基于 ECDSA 的数字签名的个人证书。 因此，直接与 QM1 通信的所有客户机和其他队列管理器都必须具

有满足类型 1 CipherSpec 要求的数字证书。 连接到队列管理器 QM1 的其他通道可能使用其他 CipherSpecs (例如 ECDHE_ECDSA_3DES_EDE_CBC_SHA256) ，但只能使用类型 1 CipherSpecs 与 QM1 进行通信。

规划 WebSphere MQ 网络时，请仔细考虑哪些通道需要 SSL 或 TLS ，并确保需要互操作的所有客户机和队列管理器使用相同类型的 CipherSpecs 和相应的数字证书。 IETF 标准 RFC 4492 ， RFC 5246 和 RFC 6460 描述了在 TLS 1.2 中的椭圆曲线 CipherSpecs 的详细用法。

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。

执行 **runmqakm** 命令将生成显示公用密钥类型的输出:

```
Label : ibmwebspheremqexample
Key Size : 384
Version : X509 V3
Serial : 9ad5eeef5d756f41
Issuer : CN=Example Certificate Authority,OU=Test,O=Example,C=US
Subject : CN=Example Queue Manager,OU=Test,O=Example,C=US
Not Before : 21 August 2011 13:10:24 GMT+01:00
Not After : 21 August 2012 13:10:24 GMT+01:00
Public Key
    30 76 30 10 06 07 2A 86 48 CE 3D 02 01 06 05 2B
    81 04 00 22 03 62 00 04 3E 6F A9 06 B6 C3 A0 11
    F8 D6 22 78 FE EF 0A FE 34 52 C0 8E AB 5E 81 73
    D0 97 3B AB D6 80 08 E7 31 E9 18 3F 6B DE 06 A7
    15 D6 9D 5B 6F 56 3B 7F 72 BB 6F 1E C9 45 1C 46
    60 BE F2 DC 1B AD AC EC 64 4C 0E 06 65 6E ED 93
    B8 F5 95 E0 F9 2A 05 D6 21 02 BD FB 06 63 A1 CC
    66 C6 8A 0A 5C 3F F7 D3
Public Key Type : EC_ecPublicKey (1.2.840.10045.2.1)
Fingerprint : SHA1 :
    3C 34 58 04 5B 63 5F 5C C9 7A E7 67 08 2B 84 43
    3D 43 7A 79
Fingerprint : MD5 :
    49 13 13 E1 B2 AC 18 9A 31 41 DC 8C B4 D6 06 68
Fingerprint : SHA256 :
    6F 76 78 68 F3 70 F1 53 CE 39 31 D9 05 C5 C5 9F
    F2 B8 EE 21 49 16 1D 90 64 6D AC EB 0C A7 74 17
Signature Algorithm : EC_ecdsa_with_SHA384 (1.2.840.10045.4.3.3)
Value
    30 65 02 30 0A B0 2F 72 39 9E 24 5A 22 FE AC 95
    0D 0C 6D 6C 2F B3 E7 81 F6 C1 36 1B 9A B0 6F 07
    59 2A A1 4C 02 13 7E DD 06 D6 FE 4B E4 03 BC B1
    AC 49 54 1E 02 31 00 90 0E 46 2B 04 37 EE 2C 5F
    1B 9C 69 E5 99 60 84 84 10 71 1A DA 63 88 33 E2
    22 CC E6 1A 4E F4 61 CC 51 F9 EE A0 8E F4 DC B5
    0B B9 72 58 C3 C7 A4
Trust Status : Enabled
```

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。

您还可以使用具有相同参数的 **iKeycmd (runmqckm)** 工具。 如果打开密钥存储库并双击证书的标签，那么还可以使用 **iKeyman (strmqikm)** GUI 来查看数字签名算法。 但是，建议您使用 **runmqakm** 工具来查看数字证书，因为它支持更广泛的算法。

椭圆曲线 CipherSpecs 和 NSA Suite B

当 WebSphere MQ 配置为符合符合符合 Suite 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 25 页的『IBM WebSphere 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 *3:* 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 182 页的『指定 CipherSpec』
通过在 **DEFINE CHANNEL** MQSC 命令或 **ALTER CHANNEL** MQSC 命令中使用 **SSLCIPH** 参数来指定 CipherSpec 。

第 90 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 25 页的『IBM WebSphere MQ 中的 NSA Suite B 密码术』
本主题提供有关如何在 Windows， Linux 和 UNIX 系统上配置 IBM WebSphere MQ 以符合套件 B 兼容 TLS 1.2 概要文件的信息。

第 17 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。

IBM WebSphere MQ 中支持的 *CipherSpec* 值

缺省 CipherSpecs 集合仅允许以下值:

TLS 1.0

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA

TLS 1.2

- ECDHE_ECDSA_AES_128_CBC_SHA256
- ECDHE_ECDSA_AES_256_CBC_SHA384
- ECDHE_ECDSA_AES_128_GCM_SHA256
- ECDHE_ECDSA_AES_256_GCM_SHA384
- ECDHE_RSA_AES_128_CBC_SHA256
- ECDHE_RSA_AES_256_CBC_SHA384
- ECDHE_RSA_AES_128_GCM_SHA256
- ECDHE_RSA_AES_256_GCM_SHA384
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_GCM_SHA384

启用不推荐的 CipherSpecs

缺省情况下，不允许您在通道定义上指定不推荐的 CipherSpec 。 如果尝试指定不推荐的 CipherSpec，那么将在队列管理器的错误日志中接收到消息 AMQ9788 。

您可以通过编辑 `qm.ini` 文件来重新启用不推荐的 CipherSpecs 。 在 `qm.ini` 文件的 SSL 节中，添加以下行:

```
SSL:
AllowWeakCipherSpec=Yes
```

您还可以在服务器上的运行时通过将环境变量 *AMQ_SSL_衰减-cipher_enable* 设置为任何值来重新启用一个或多个不推荐的 CipherSpecs 。 此环境变量将启用 CipherSpecs ，而不考虑 `qm.ini` 文件中指定的值。

通道认证记录

要在通道级别演练对授权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 WebSphere 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 *4:* 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，您可能会遇到一些情况：尝试从不应有权访问 WebSphere MQ 系统的 IP 地址进行连接，必须在可以更新防火墙之前暂时先阻止该地址。 这些连接尝试甚至可能不是来自 WebSphere MQ 通道，而是来自错误配置为将 WebSphere MQ 侦听器作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 151 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 153 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 155 页的『阻止对客户机声明的用户标识进行访问』，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 155 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 156 页的『阻止访问 SSL 专有名称』，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 156 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 153 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 WebSphere MQ MQI 客户机的连接使用时，使用所指定的一个不同的 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 154 页的『将客户机断言的用户标识映射到 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 154 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 153 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某个通道名称、IP 地址、队列管理器名称或 SSL/TLS DN 匹配，那么将使用最具体的匹配项。 按照如下所示来确定视为最具体的匹配。

- 对于通道名称：
  - 最具体的匹配是不带通配符的名称，例如，A.B.C。
  - 最通用的匹配是单个星号 (*)，这与所有通道名称都匹配。
  - 在最左侧位置具有星号的模式比在最左侧位置具有定义值的模式更通用。 因此，*.B.C 比 A.* 更通用。
  - 在第二个位置具有星号的模式比在第二个位置具有定义值的模式更通用，对于每个后续位置，亦是如此。 因此，A.*.C 比 A.B.* 更通用。
  - 如果两个或更多个模式在同一位置具有星号，那么星号后面节点数更少的模式更通用。 因此， A. * 比 A.*.C 更通用
- 对于 IP 地址：
  - 最具体的匹配是不带通配符的名称，例如，192.0.2.6。
  - 最通用的匹配是单个星号 (*)，这与所有通道名称都匹配。
  - 在最左侧位置具有星号的模式比在最左侧位置具有定义值的模式更通用。 因此，*.0.2.6 比 192.* 更通用。
  - 在第二个位置具有星号的模式比在第二个位置具有定义值的模式更通用，对于每个后续位置，亦是如此。 因此，192.*.2.6 比 192.0.* 更通用。
  - 如果两个或更多个模式在同一位置具有星号，那么星号后面节点数更少的模式更通用。 因此 192 .* 比 192.*.2.* 更通用。
  - 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，只要没有星号与尾部星号相邻。 例如，192.*.2.* 有效，但 192.0.*.* 无效。
  - IPv6 地址模式不能包含双冒号和尾部星号，因为生成的地址会不明确。 例如，2001::* 可以展开为 2001:0000:*、2001:0000:0000:*，等等。
- 对于队列管理器名称：
  - 最具体的匹配是不带通配符的名称，例如，192.0.2.6。
  - 最通用的匹配是单个星号 (*)，这与所有通道名称都匹配。
  - 在最左侧位置具有星号的模式比在最左侧位置具有定义值的模式更通用。 因此，*QUEUEMANAGER 比 QUEUEMANAGER* 更通用。
  - 在第二个位置具有星号的模式比在第二个位置具有定义值的模式更通用，对于每个后续位置，亦是如此。 因此，Q*MANAGER 比 QUEUE* 更通用。

– 如果两个或更多个模式在同一位置具有星号，那么星号后面字符数更少的模式更通用。 因此，Q* 比 Q*MGR 更通用。

- 对于 SSL 或 TLS 专有名称 (DN)，子串的优先顺序如下所示：

表 *5:* 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	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 WebSphere MQ 使用 O=IBM 的通道认证记录，优先于 C=UK 的通道认证记录（如果两者均存在）。

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 46 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。

IBM WebSphere MQ 中的消息安全性

IBM WebSphere MQ 基础架构中的消息安全性由单独许可的组件 IBM WebSphere MQ Advanced Message Security 提供。

IBM WebSphere MQ Advanced Message Security (AMS) 扩展 IBM WebSphere 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 226 页的『IBM WebSphere MQ Advanced Message Security』

IBM WebSphere MQ Advanced Message Security (AMS) 是单独许可的 IBM WebSphere MQ Advanced Message Security 组件，可为流经 IBM WebSphere MQ Advanced Message Security 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 WebSphere MQ 环境中规划安全性时需要考虑的事项。

您可以将 IBM WebSphere 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。

WebSphere MQ 提供了一系列链路级别安全服务，包括对安全套接字层 (SSL) 和传输层安全性 (TLS) 的支持。

实现 WebSphere 时，必须考虑安全性的某些方面。 在 UNIX， Linux 和 Windows 系统上，如果忽略这些方面而不执行任何操作，那么无法使用 WebSphere MQ。

下面描述了安全注意事项。

管理 WebSphere MQ 的权限

WebSphere MQ 管理员需要以下权限:

- 发出命令以管理 WebSphere MQ
- 使用 IBM WebSphere MQ Explorer

有关更多信息，请参阅:

- 第 165 页的『在 UNIX, Linux, and Windows 系统上管理 IBM WebSphere MQ 的权限』

使用 WebSphere MQ 对象的权限

应用程序可以通过发出 MQI 调用来访问以下 WebSphere 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 WebSphere MQ 对象，以及访问通道和认证信息对象。 这些对象可由 WebSphere MQ 保护，以便与应用程序关联的用户标识需要权限才能访问这些对象。

有关更多信息，请参阅第 42 页的『应用程序使用 IBM WebSphere 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 WebSphere 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 56 页的『通道授权』。

其他考虑事项

仅当使用某些 WebSphere 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 63 页的『队列管理器集群的安全性』
- 第 64 页的『IBM WebSphere MQ 发布/预订的安全性』
- 第 65 页的『IBM WebSphere MQ 因特网传递的安全性』

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 WebSphere 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 SSL 或 TLS 的 IBM WebSphere MQ 通道使用数字证书作为标识和认证机制。 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 WebSphere MQ 进行认证。 有关更多信息，请参阅:

- 第 153 页的『将远程队列管理器映射到 MCAUSER 用户标识』
- 第 154 页的『将客户机断言的用户标识映射到 MCAUSER 用户标识』
- 第 154 页的『将 SSL 或 TLS 专有名称映射到 MCAUSER 用户标识』
- 第 156 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 WebSphere MQ MQI 客户机。 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。

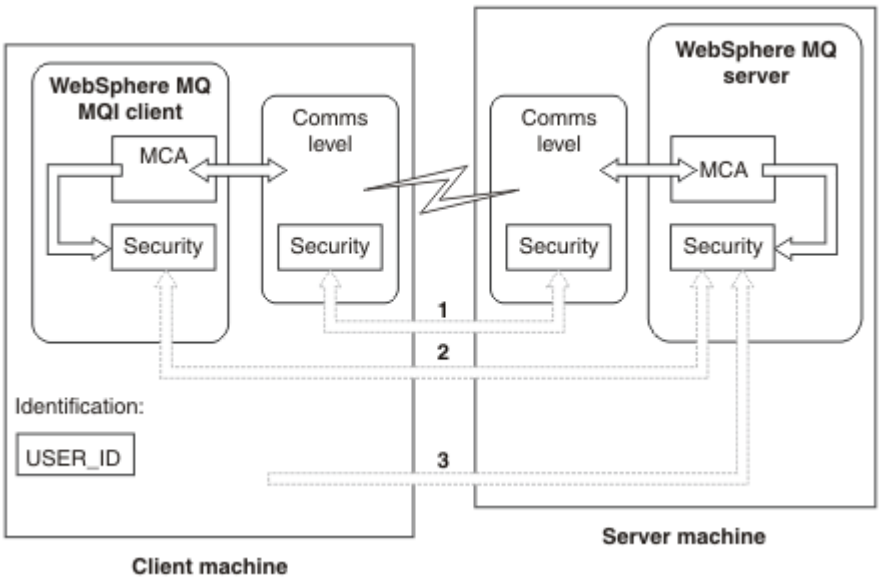


图 7: 客户机/服务器连接中的安全性

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 SSL 或 TLS。 有关更多信息，请参阅 第 13 页的『加密安全性协议：SSL 和 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 SSL/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 33 页的『通道认证记录』中提供了完整的描述。

3. 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。

4. 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 WebSphere 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。

   Windows 客户机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。

   为了在 IBM WebSphere MQ Client for HP Integrity NonStop Server 上帮助标识，客户机传递运行客户机应用程序的 OSS Safeguard 别名。 此标识的格式通常为 <PRIMARYGROUP>.<ALIAS>。 如果需要，您可以使用通道认证记录或安全出口将此用户标识映射到队列管理器上的备用用户标识。 有关消息出口的更多信息，请参阅 第 123 页的『消息出口中的身份映射』。 有关定义通道认证记录的更多信息，请参阅 第 154 页的『将客户机断言的用户标识映射到 MCAUSER 用户标识』。

   服务器安全出口可以使用用户标识和 (如果可用) 安全标识的值来确定 IBM WebSphere MQ MQI 客户机的身份。

用户标识

如果 IBM WebSphere MQ MQI 客户机在 Windows 上，并且 IBM WebSphere MQ 服务器也在 Windows 上，并且有权访问定义了客户机用户标识的域，那么 IBM WebSphere MQ 支持最多 20 个字符的用户标识。 在 UNIX and Linux 平台和配置上，最大长度为 12 个字符。

如果 WebSphere MQ for Windows 服务器以包含 @ 字符的用户标识 (例如， abc@d) 运行，那么该服务器不支持 Windows 客户机的连接。 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。

请注意， UNKNOWN 是保留用户标识， NOBODY 用户标识对于 WebSphere MQ 也具有特殊含义。 在名为 UNKNOWN 或 NOBODY 的操作系统中创建用户标识可能会产生意外结果。

虽然用户标识用于认证，但组用于授权 (Windows 除外)。

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。

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 WebSphere MQ 对象，包括从 IBM WebSphere MQ MQI 客户机连接的对象。

必须授予个人或应用程序访问权才能使用 IBM WebSphere 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 WebSphere 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 WebSphere 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 WebSphere MQ 的权限

IBM WebSphere 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 WebSphere MQ 管理员需要以下权限:

- 发出命令以管理 IBM WebSphere MQ
- 使用 IBM WebSphere MQ Explorer

有关更多信息，请参阅适用于您的操作系统的主题。

在 *UNIX* 和 *Windows* 系统上管理 *IBM WebSphere MQ* 的权限

IBM WebSphere MQ 管理员是 mqm 组的成员。 此组有权访问所有 IBM WebSphere MQ 资源，并且可以发出 IBM WebSphere MQ 控制命令。 管理员可针对其他用户授予特定权限。

要在 UNIX 和 Windows 系统上成为 IBM WebSphere MQ 管理员，用户必须是 *mqm* 组的成员。 此组是在您安装 WebSphere 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 UNIX 系统上的 root 用户。

可以向非 mqm 组成员的用户授予管理特权，但他们无法发出 IBM WebSphere MQ 控制命令，并且他们有权仅执行已授予其访问权的命令。

此外，在 Windows 系统上， SYSTEM 和 Administrator 帐户具有对 IBM WebSphere 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 WebSphere 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 WebSphere MQ 资源。

管理员可以使用控制命令 **runmqsc** 来发出 WebSphere 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。

WebSphere 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 WebSphere MQ Explorer 来管理本地系统上的队列管理器。 当 WebSphere 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:

- 有关对队列管理器，队列，通道，进程，名称列表和认证信息对象运行的命令，请参阅 第 42 页的『应用程序使用 IBM WebSphere 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。

有关在 UNIX 和 Windows 系统上管理 WebSphere MQ 所需的权限的更多信息，请参阅相关信息。

应用程序使用 IBM WebSphere 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 WebSphere 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 WebSphere 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。

使用 IBM WebSphere MQ classes for Java ， IBM WebSphere MQ classes for JMS ， IBM WebSphere MQ classes for .NET 或 Message Service Clients for C/C++ and .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 WebSphere 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 56 页的『通道授权』。

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 WebSphere MQ。 IBM WebSphere 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 WebSphere 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。 在 UNIX, Linux, and Windows 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。

当命令服务器处理在 WebSphere MQ 对象上运行的 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 43 页的『消息上下文』。

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。

在 *UNIX*， *Linux* 和 *Windows* 系统上使用 *IBM WebSphere MQ* 对象的权限

IBM WebSphere 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。

1. 认证。

   IBM WebSphere 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 ，那么检查由 OAM 在其 MQZID_AUTHENTICATE_USER 函数中执行。 这是检查: 将 MQCSP 结构中的用户标识与 *IdentityContext* (MQZIC) 中的用户标识进行比较，以确定它们是否匹配。 如果它们不匹配，那么检查将失败。

此基本检查并非旨在作为用户的完整认证。 例如，不会通过检查 MQCSP 结构中提供的密码来检查用户的真实性。 此外，如果应用程序省略了 MQCSP 结构，那么不会执行任何检查。

如果需要通过授权服务组件在队列管理器中提供更完整的认证服务，那么随 IBM WebSphere MQ 提供的 OAM 不会提供此服务。 您必须编写新的授权服务组件，或者从供应商处获取一个授权服务组件。

2. 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 UNIX， Linux 和 Windows 系统上，当应用程序发出 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 WebSphere MQ 对象时， 授权服务 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。

在 Windows 上，即使启用了 UAC， OAM 也会授予 Administrators 组的成员访问所有 IBM WebSphere MQ 对象的权限。

此外，在 Windows 系统上， SYSTEM 帐户具有对 IBM WebSphere MQ 资源的完全访问权。

当 PCF 命令对其中一个 IBM WebSphere 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。

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 WebSphere MQ 产品提供的组件。

IBM WebSphere 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 WebSphere MQ 对象维护一个访问控制表 (ACL)。 在 UNIX 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 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 并且 Linux 已提高此限制，但 IBM WebSphere MQ 继续在所有 UNIX 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 WebSphere 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 WebSphere 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。

在 UNIX， Linux 和 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。

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。

使用 *PCF* 访问 *OAM* 命令
在 UNIX， Linux 和 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 *6: 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 。

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 WebSphere 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，或根据以下内容设置 MCAUSER 值: 远程 IP 地址，远程用户标识，提供的 SSL 或 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。

UNIX and Linux 系统上对象的安全性

如果此标识将使用 IBM WebSphere 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。

UNIX and Linux 系统上的用户标识

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 系统上对象的安全性

如果此标识将使用 IBM WebSphere 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 Windows 以外的平台， UNIX and Linux 系统将大写字符用于消息中的用户标识。

为了允许 Windows， UNIX and Linux 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 UNIX， Linux 和 Windows 系统上提供消息出口，那么不会执行自动转换。

使用定制授权服务

IBM WebSphere MQ 提供可安装的授权服务。 您可以选择安装备用服务。

IBM WebSphere MQ 随附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 WebSphere 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 7: 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 48 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已使用的标识时使用。
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 48 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 WebSphere MQ MQI 客户机，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。

请确保 MCAUSER 字段在 Windows 平台上限制为 12 个字符，因为任何额外的字符都将被截断，这可能会导致授权失败。

当正在使用客户机端安全性出口时， IBM WebSphere 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:

- 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) ，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 WebSphere 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 WebSphere 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。

对于 UNIX， Linux 和 Windows 系统上的服务器，从客户机接收的 `MCAUserIdentifier` 字段的内容将更改为小写。

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。

对于 UNIX 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 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 SSL 或 TLS，在这种情况下，必须规划数字证书的使用情况。 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 49 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 53 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 57 页的『使用 SSL 保护通道』
IBM WebSphere MQ 中的 SSL 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 49 页的图 8 中说明了链接级别和应用程序级别安全性。

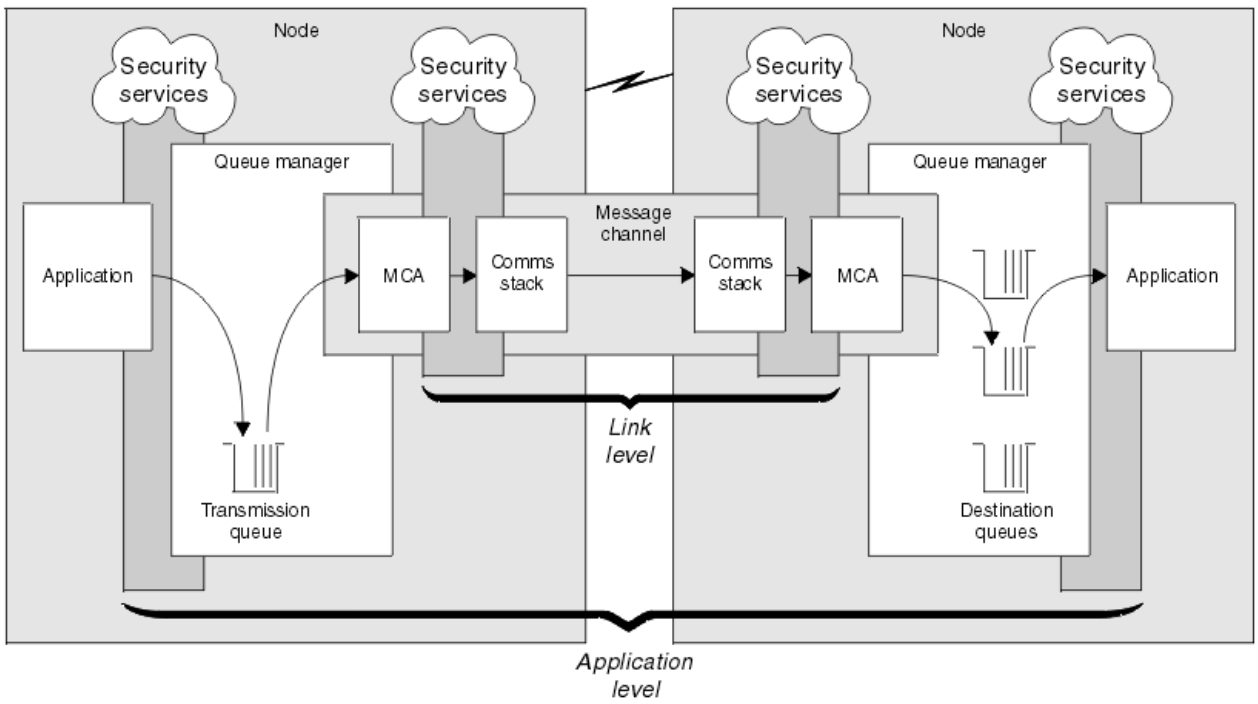


图 *8:* 链接级别安全性和应用程序级别安全性

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。

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 49 页的图 8 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

如果队列管理器在受控可信环境中运行，那么 WebSphere 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 WebSphere MQ 通道协议流中的数据 (消息数据除外)。 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 49 页的图 8 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 WebSphere MQ 提供的链路级别安全性

在 IBM WebSphere MQ 中提供机密性和数据完整性的主要方法是使用 SSL 或 TLS。 有关在 IBM WebSphere MQ 中使用 SSL 和 TLS 的更多信息，请参阅 第 20 页的『IBM WebSphere MQ 支持 SSL 和 TLS』。 对于认证， IBM WebSphere 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 33 页的『通道认证记录』。

提供您自己的链接级别安全性

此主题集合描述如何提供您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 53 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 IBM WebSphere MQ for Windows (SSPI 通道出口程序) 随附的通道出口程序。 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 WebSphere 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 49 页的图 8 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*

IBM WebSphere MQ Advanced Message Security (AMS) 是单独许可的 IBM WebSphere MQ 组件，可为流经 IBM WebSphere 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 WebSphere 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， IBM WebSphere MQ Advanced Message Security 可帮助应对这些挑战。

IBM WebSphere MQ Advanced Message Security 通过以下方式扩展了 IBM WebSphere 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 WebSphere MQ 服务器和客户机。
- 它与 IBM WebSphere MQ 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 226 页的『IBM WebSphere MQ Advanced Message Security』。

提供您自己的应用程序级别安全性

此主题集合描述了如何提供您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 WebSphere MQ 提供了两个出口，即 API 出口和 API 交叉出口。

这些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 WebSphere 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 WebSphere 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 53 页的图 9 中进行了说明，并在以下主题中进行了描述。

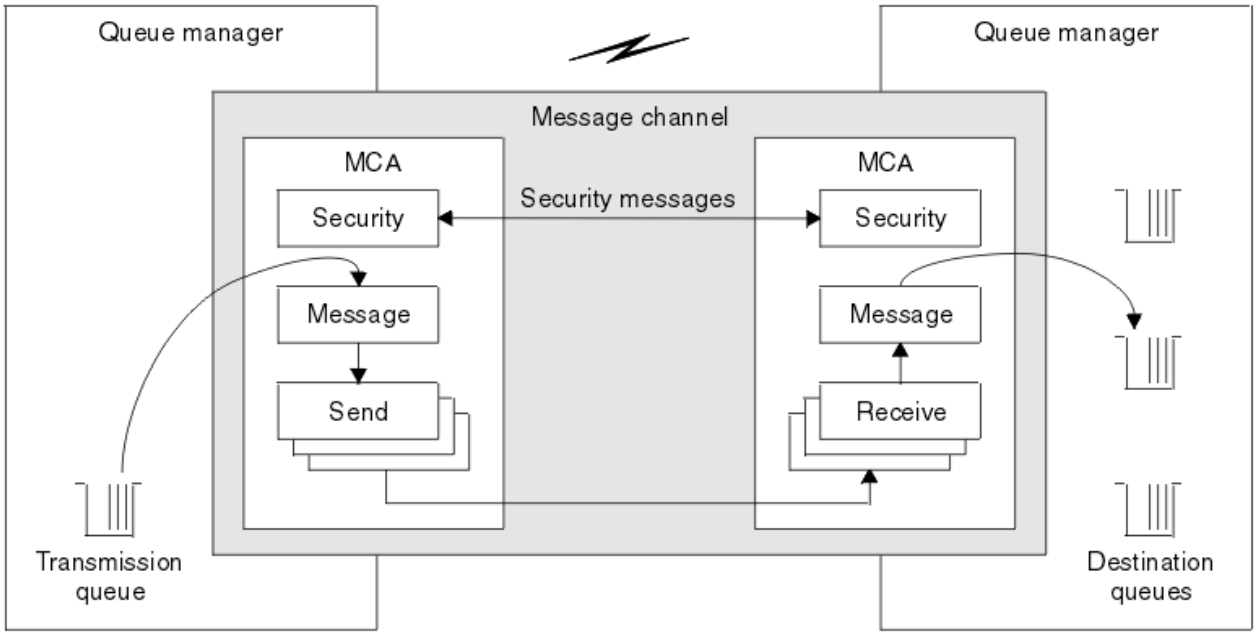


图 *9:* 消息通道上的安全性，消息，发送和接收出口

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 51 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 WebSphere MQ MQI 客户机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 51 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 WebSphere 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 WebSphere 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 51 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，您可以选择使用 IBM WebSphere MQ Advanced Message Security 对消息进行数字签名，以防止未经授权的修改。 如果标准设施不满足您的要求，您还可以使用 API 出口程序。

在链接级别，您可以选择使用 SSL 或 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 57 页的『使用 SSL 保护通道』
IBM WebSphere MQ 中的 SSL 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 19 页的『IBM WebSphere MQ 中的数据完整性』
您可以使用数据完整性服务来检测是否已修改消息。

第 52 页的『规划 Advanced Message Security』
IBM WebSphere MQ Advanced Message Security (AMS) 是单独许可的 IBM WebSphere MQ 组件，可为流经 IBM WebSphere 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 WebSphere 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 WebSphere MQ 接口对 IBM WebSphere MQ 进行的更改
配置 IBM WebSphere MQ 以发出检测事件，特别是命令事件和配置事件。

在 IBM WebSphere MQ 的控制范围外对其进行的更改
某些更改可能会影响 IBM WebSphere MQ 的行为方式，但无法由 IBM WebSphere 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 WebSphere 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 WebSphere MQ 配置为发出检测事件。

IBM WebSphere 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 WebSphere 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 WebSphere 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 WebSphere 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，需要有权访问各种 IBM WebSphere MQ 资源的用户标识。

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 WebSphere MQ 中，可以通过 SSL 或 TLS 支持来保护通道。

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。 请参阅 上下文的授权 以获取更多信息。
- 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。
- 使用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 WebSphere 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，该帐户将具有完全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。
  – 如果您使用缺少 WebSphere 管理特权的用户标识，那么必须将通道的 dsp 和 ctrlx 权限授予该用户标识以使通道工作。 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。 例如，发出下列命令:

```
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 33 页的『通道认证记录』
要在通道级别演练对授权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。

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 WebSphere MQ 通道启动器不是 IBM WebSphere MQ 对象; 对它们的访问不受 OAM 控制。 IBM WebSphere 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入传输队列，那么这需要特殊授权; 请参阅 第 74 页的表 10。

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 53 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* 保护通道
IBM WebSphere MQ 中的 SSL 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。

SSL 支持的命令和属性

安全套接字层 (SSL) 协议提供通道安全性，具有防止窃听，篡改和冒充的保护。 IBM WebSphere MQ 对 SSL 的支持使您能够在通道定义上指定特定通道使用 SSL 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 SSL:

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 SSL:

SSLCRLNL
SSLCRLNL 属性指定用于提供证书撤销位置以允许增强 TLS/SSL 证书检查的认证信息对象的名称列表。

SSLCRYP
在 Windows 上， UNIX and Linux 系统设置 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 SSL 的通道无法建立 SSL 连接，是否报告 SSL 事件消息。

SSLFIPS
指定如果在 IBM WebSphere 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 Windows 上，UNIX and Linux 系统将密钥存储库与队列管理器相关联。 密钥数据库保存在 *GSKit* 密钥数据库中。 (IBM Global Security Kit (GSKit) 使您能够在 WindowsUNIX and Linux 系统上使用 SSL 安全性。)

SSLRKEYC
在重新协商密钥之前要在 SSL 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 SSL:

SSLCAUTH
定义 IBM WebSphere MQ 是否需要并验证来自 SSL 客户机的证书。

SSLCIPH
指定加密强度和功能 (CipherSpec) ，例如 NULL_MD5 或 RC4_MD5_US。 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)。

本部分描述了用于支持认证信息对象的 `setmqaut`， `dspmqaut`， `dmpmqaut`， `rcrmqobj`， `rcdmqimg` 和 `dspmqfls` 命令。 它还描述了用于在 UNIX and Linux 系统上管理证书的 iKeycmd 命令，以及用于在 UNIX， Linux 和 Windows 系统上管理证书的 runmqakm 工具。 请参阅下列各部分：

- 设置为
- 长石 (dspmqaut)
- dmpmqaut
- rcrmqobj
- rcdmqimg
- dspmqfls

- 管理密钥和证书

有关使用 SSL 的通道安全性的概述，请参阅

- 第 20 页的『IBM WebSphere MQ 支持 SSL 和 TLS』

有关与 SSL 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 SSL 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

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:

在 Windows， UNIX 和 Linux 系统上，可以使用自签名证书。 有关指示信息，请参阅第 101 页的『在 UNIX, Linux, and Windows 系统上创建自签名个人证书』。

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，需要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使用特定的 CipherSpec ，那么必须获取合适的数字证书。

有关 CipherSpecs 与数字证书之间的关系的更多信息，请参阅 第 28 页的『IBM WebSphere 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 WebSphere MQ 中的证书验证策略的更多信息，请参阅 第 28 页的『IBM WebSphere MQ 中的证书验证策略』。

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， WebSphere 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，以获取有关可在您的平台上使用的更强加密形式 (例如， z/OS 上的三重 DES 24 字节加密) 的信息。

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。

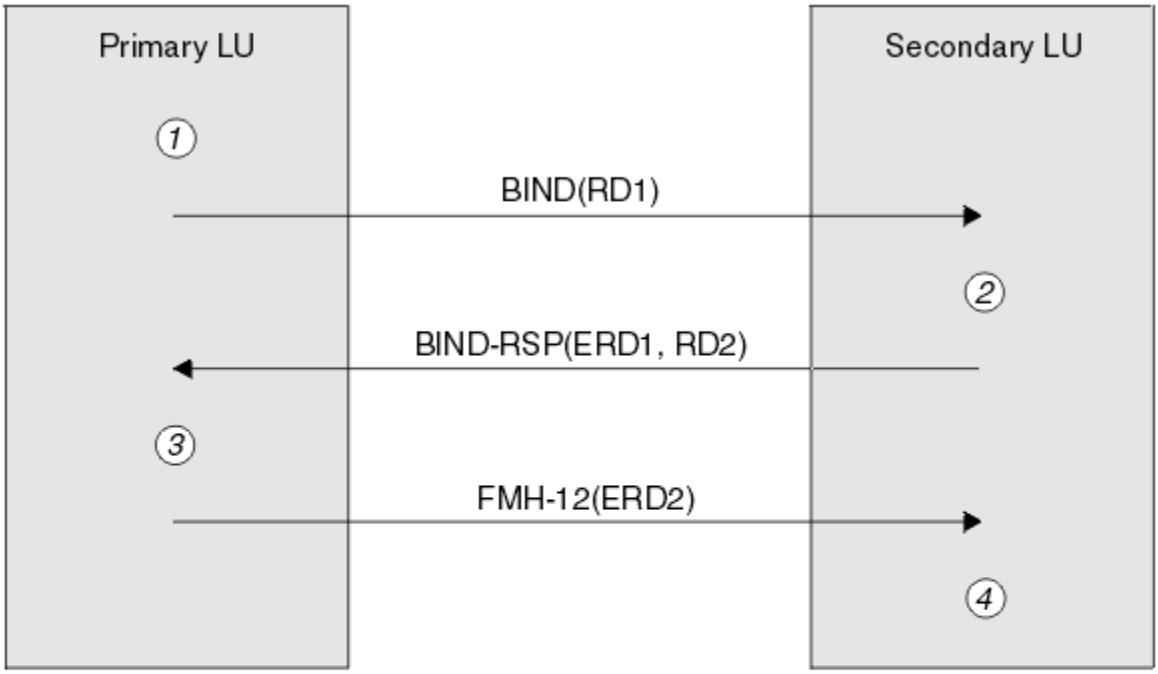
会话级别认证

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 61 页的图 10 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 *10:* 用于会话级别认证的流

会话级别认证的协议如下所示。 过程中的数字对应于 第 61 页的图 10 中的数字。

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。

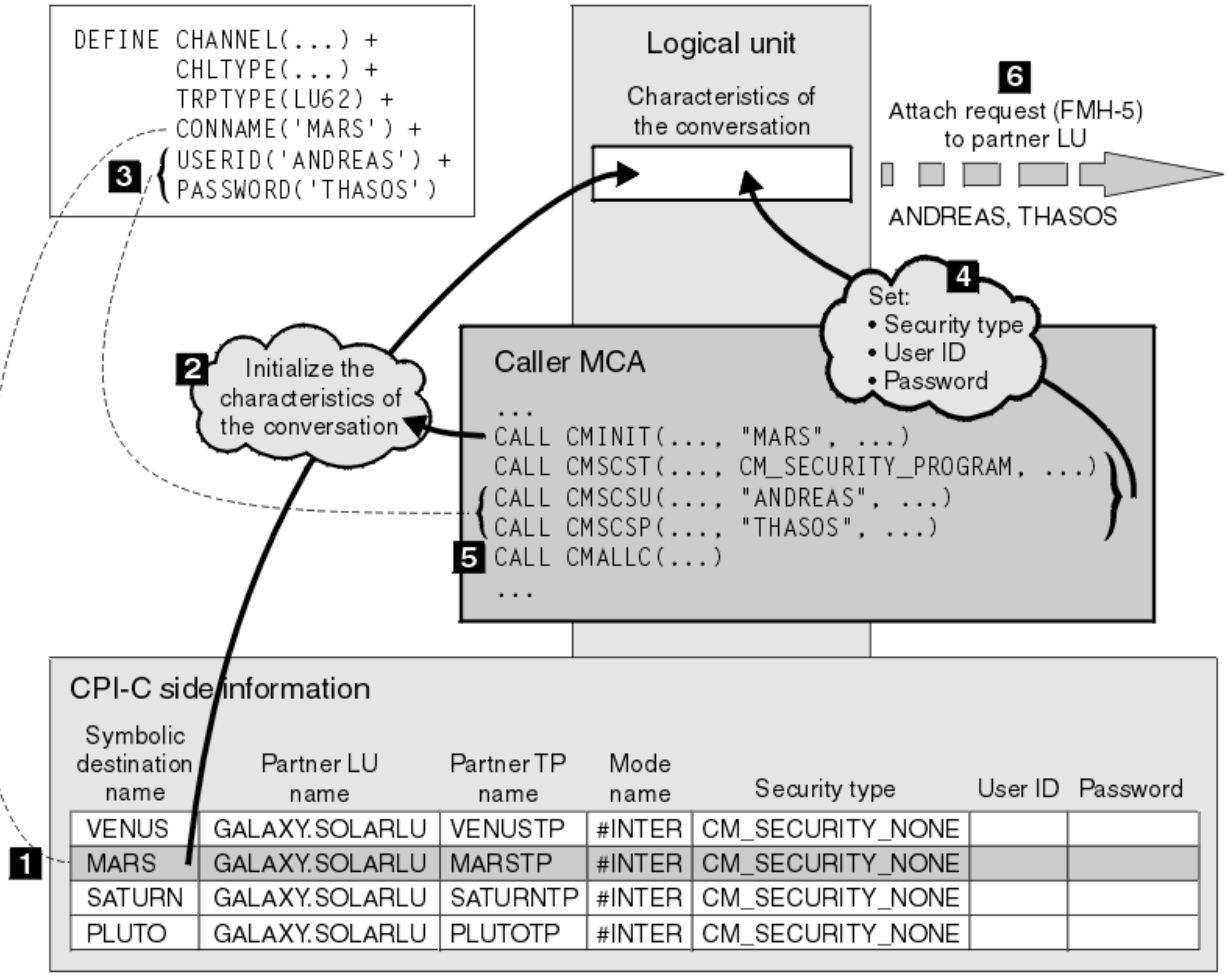
以下主题描述了 IBM WebSphere MQ 如何为对话级别认证提供支持。

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。 有关特定于 z/OS 的信息，请参阅 *z/OS MVS Planning: APPC/MVS Management*， SA22-7599。

有关 CPI-C 的更多信息，请参阅 公共编程接口通信 *CPI-C* 规范 SC31-6180。 有关 APPC/MVS TP 对话可调用服务的更多信息，请参阅 *z/OS MVS Programming: Writing Transaction Program for APPC/MVS*，SA22-7621。

在 *UNIX* 系统和 *Windows* 系统上的 *IBM WebSphere MQ* 中支持对话级别认证
使用本主题来获取 UNIX, Linux, and Windows 上的对话级别认证工作方式的概述。

第 62 页的图 11 中说明了 IBM WebSphere MQ for WebSphere MQ on UNIX 系统和 WebSphere MQ for Windows 中对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。



Symbolic destination name	Partner LU name	Partner TP name	Mode name	Security type	User ID	Password
VENUS	GALAXY.SOLARLU	VENUSTP	#INTER	CM_SECURITY_NONE		
MARS	GALAXY.SOLARLU	MARSTP	#INTER	CM_SECURITY_NONE		
SATURN	GALAXY.SOLARLU	SATURNTP	#INTER	CM_SECURITY_NONE		
PLUTO	GALAXY.SOLARLU	PLUTOTP	#INTER	CM_SECURITY_NONE		

图 *11: WebSphere MQ* 支持对话级别认证

在 IBM i， UNIX 系统和 Windows 系统上， 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。

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 WebSphere MQ 决定要将其发送到哪个队列管理器。 当许多应用程序将消息发送到集群队列时，WebSphere 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。 如果托管克隆队列管理器的某个系统发生故障，那么 WebSphere 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。

相关任务

第 208 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 WebSphere MQ 发布/预订的安全性

如果您正在使用 IBM WebSphere 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 WebSphere 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 WebSphere 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 WebSphere MQ 多点广播可能需要安全进程的原因。

IBM WebSphere 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 WebSphere MQ 应用程序 (例如， LLM 应用程序，请参阅 多点广播互操作性与 WebSphere MQ Low Latency Messaging 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 WebSphere 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 47 页的『客户机的访问控制』。

网络安全性

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 6 页的『加密概念』。

防火墙和因特网通道

您通常会使用防火墙来阻止来自敌对 IP 地址的访问，例如在 "拒绝服务" 攻击中。 但是，您可能需要临时阻止 IBM WebSphere MQ 中的 IP 地址，可能是在等待安全管理员更新防火墙规则时。

要阻止一个或多个 IP 地址，请创建类型为 BLOCKADDR 或 ADDRESSMAP 的通道认证记录。 有关更多信息，请参阅第 151 页的『阻止特定 IP 地址』。

IBM WebSphere MQ 因特网传递的安全性

通过因特网传递可以简化通过防火墙的通信，但这会产生安全影响。

IBM WebSphere MQ Internet pass-thru 是 SupportPac MS81 中提供的 IBM WebSphere MQ 基本产品扩展。

WebSphere MQ 因特网传递使两个队列管理器能够通过因特网交换消息，或者使 WebSphere MQ 客户机应用程序能够通过因特网连接到队列管理器，而无需直接 TCP/IP 连接。 如果防火墙禁止两个系统之间的直接 TCP/IP 连接，那么这很有用。 它使 WebSphere MQ 通道协议流入和流出防火墙的通道变得更简单，更易于管理，方法是通过隧道传送 HTTP 中的流或充当代理。 通过使用安全套接字层 (SSL) ，它还可用于对通过因特网发送的消息进行加密和解密。

当 WebSphere MQ 系统与 IPT 通信时，除非您在 IPT 中使用 SSLProxyMode ，否则请确保 WebSphere MQ 使用的 CipherSpec 与 IPT 使用的 CipherSuite 匹配:

- 当 IPT 充当 SSL 或 TLS 服务器并且 WebSphere MQ 作为 SSL 或 TLS 客户机进行连接时， WebSphere MQ 所使用的 CipherSpec 必须对应于在相关 IPT 密钥环中启用的 CipherSuite 。
- 当 IPT 充当 SSL 或 TLS 客户机并连接到 WebSphere MQ SSL 或 TLS 服务器时， IPT CipherSuite 必须与接收 WebSphere MQ 通道上定义的 CipherSpec 相匹配。

如果从 IPT 迁移到集成的 WebSphere MQ SSL 和 TLS 支持，请使用 iKeyman 从 IPT 传输数字证书。

有关更多信息，请参阅 WebSphere MQ Internet Pass-Thru (SupportPac MS81)。

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。

在 UNIX, Linux, and Windows 系统上设置安全性

特定于 UNIX, Linux, and Windows 系统的安全注意事项。

IBM WebSphere 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 WebSphere MQ
您可以定义一组可以发出命令来管理 IBM WebSphere MQ 的用户。

谁可以使用 IBM WebSphere 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 WebSphere 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 WebSphere 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。

使用终端服务连接到 IBM WebSphere MQ

如果您正在使用终端服务，那么 **Create global objects** 用户权限可能会导致问题。

如果您是使用终端服务连接到 Windows 系统，并且在创建或启动队列管理器时迁到问题，那么这可能是由于 Windows 的最新版本中的用户权限 **Create global objects** 所致。

Create global objects 用户权限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 **Create global objects** 用户权限。

缺省情况下，管理员应用了 **Create global objects** 用户权限，因此管理员可以在使用终端服务进行连接时创建和启动队列管理器，而无需更改用户权限。

如果使用终端服务时管理 WebSphere MQ 的各种方法不起作用，请尝试设置 **Create global objects** 用户权限:

1. 打开 "管理工具" 面板:

   Windows 2003 和 Windows XP
   使用 **控制面板** > **管理工具**访问此面板。

   Windows Vista 和 Windows Server 2008
   使用 **控制面板** > **系统和维护** > **管理工具**来访问此面板。
2. 双击**本地安全策略**。
3. 展开 `Local Policies`。
4. 单击 `User Rights Assignment`。
5. 将新用户或组添加到 **Create global objects** 策略。

在 Windows 上创建和管理组

这些指示信息将指导您完成在工作站或成员服务器上管理组的过程。

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，无法识别您对主体组成员资格所作的任何更改。

使用 "计算机管理" 面板来处理用户和组。 在用户重新登录之前，对当前登录用户所作的任何更改都可能不会生效。

Windows 2003 和 Windows XP
使用 **控制面板** > **管理工具** > **计算机管理**访问此面板。

Windows Vista 和 Windows Server 2008
使用 **控制面板** > **系统和维护** > **管理工具** > **计算机管理**访问此面板。

Windows 7
使用 **管理工具** > **计算机管理** 访问此面板

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**。

在 *Windows* 上将用户添加到组中

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。

在 *Windows* 上显示组中的人员

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。

在 *Windows* 上从组中除去用户

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。

在 HP-UX 上创建和管理组

在 HP-UX 上，如果您未使用 NIS 或 NIS + ，请使用系统管理管理器 (SAM) 来处理组。

在 *HP-UX* 上创建组

使用系统管理管理器将用户添加到组

过程

1. 在系统管理管理器 (SAM) 中，双击 "用户和组的帐户"。

2. 双击 "组"。
3. 从 "操作" 下拉列表中选择 "添加" 以显示 "添加新组" 面板。
4. 输入组的名称，然后选择要添加到组的用户。
5. 单击 "应用" 以创建组。

结果
您现在已创建组。

在 *HP-UX* 上向组添加用户
使用系统管理管理器将用户添加到组。

过程
1. 在系统管理管理器 (SAM) 中，双击 "用户和组的帐户"。
2. 双击 "组"。
3. 突出显示组的名称，并从 "操作" 下拉列表中选择 "修改" 以显示 "修改现有组" 面板。
4. 选择要添加到组的用户，然后单击 "添加"。
5. 如果要将其他用户添加到组，请对每个用户重复步骤 4。
6. 完成向列表添加名称后，单击 "确定"。

结果
您现在已将用户添加到组。

在 *HP-UX* 上显示组中的人员
使用系统管理管理器显示组中的人员

过程
1. 在系统管理管理器 (SAM) 中，双击 "用户和组的帐户"。
2. 双击 "组"。
3. 突出显示该组的名称，并从 "操作" 下拉列表中选择 "修改" 以显示 "修改现有组" 面板，显示该组中的用户列表。

结果
这样会显示组成员。

在 *HP-UX* 上从组中除去用户
使用系统管理管理器从组中除去用户。

过程
1. 在系统管理管理器 (SAM) 中，双击 "用户和组的帐户"。
2. 双击 "组"。
3. 突出显示组的名称，并从 "操作" 下拉列表中选择 "修改" 以显示 "修改现有组" 面板。
4. 选择要从组中除去的用户，然后单击 "除去"。
5. 如果要从组中除去其他用户，请对每个用户重复步骤 4。
6. 完成从列表中除去名称后，单击 "确定"。

结果
您现在已从组中除去用户

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。

创建组

使用 SMITTY 创建组。

过程

1. 从 SMITTY 中，选择 "安全性和用户" ，然后按 Enter 键。
2. 选择组并按 Enter 键。
3. 选择 "添加组" ，然后按 Enter 键。
4. 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)。
5. 按 Enter 键以创建组。

结果

您现在已创建组。

将用户添加到组

使用 SMITTY 将用户添加到组。

过程

1. 从 SMITTY 中，选择 "安全性和用户" ，然后按 Enter 键。
2. 选择组并按 Enter 键。
3. 选择 "更改/显示组的特征" ，然后按 Enter 键。
4. 输入组的名称以显示组的成员列表。
5. 将要添加的用户的名称添加到组中，以逗号分隔。
6. 按 Enter 键以将名称添加到组。

显示组中的人员

显示使用 SMITTY 的组中的人员。

过程

1. 从 SMITTY 中，选择 "安全性和用户" ，然后按 Enter 键。
2. 选择组并按 Enter 键。
3. 选择 "更改/显示组的特征" ，然后按 Enter 键。
4. 输入组的名称以显示组的成员列表。

结果

这样会显示组成员。

从组中除去用户

使用 SMITTY 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. 选择组并按 Enter 键。
3. 选择 "更改/显示组的特征" ，然后按 Enter 键。
4. 输入组的名称以显示组的成员列表。
5. 删除要从组中除去的用户的名称。

6. 按 Enter 键以从组中除去名称。

结果
您现在已从组中除去用户。

在 Solaris 上创建和管理组

在 Solaris 上，如果您未使用 NIS 或 NIS + ，请使用 /etc/group 文件来处理组。

在 *Solaris* 上创建组
使用 **groupadd** 命令创建组。

过程

输入以下命令： groupadd *group-name*
其中 *group-name* 是组的名称。

结果
文件 /etc/group 包含组信息。

在 *Solaris* 上向组添加用户
使用 **usermod** 命令将用户添加到组。

过程

要将成员添加到补充组，请执行 usermod 命令并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，如果用户是组 groupa 的成员，并且要同时成为 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。

在 *Solaris* 上显示组中的人员
要发现谁是组的成员，请在 /etc/group 文件中查看该组的条目。

在 *Solaris* 上从组中除去用户
使用 **usermod** 命令从组中除去用户。

过程

要从补充组中除去成员，请执行 **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，那么将使用以下命令从 mqm 组中除去该用户: usermod -G groupa,groupb *user-name*，其中 *user-name* 是用户名。

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。

在 *Linux* 上创建组
使用 **groupadd** 命令创建组。

过程

要创建新组，请输入以下命令: groupadd -g *group-ID group-name*
，其中 *group-ID* 是组的数字标识， *group-name* 是组的名称。

结果
文件 /etc/group 包含组信息。

将用户添加到 *Linux* 上的组

使用 **usermod** 命令将用户添加到组。

过程

要将成员添加到补充组，请执行 usermod 命令并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，如果用户是组 groupa 的成员，并且也要成为 groupb 的成员，那么将使用以下命令: `usermod -G groupa,groupb user-name`
，其中 *user-name* 是用户名。

显示 *Linux* 上的组中的人员

使用 **getent** 命令显示组中的人员。

过程

要显示作为组成员的人员，请输入以下命令: `getent group group-name`
，其中 *group-name* 是组的名称。

从组中除去用户

使用 **usermod** 命令从组中除去用户。

过程

要从补充组中除去成员，请执行 **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，那么将使用以下命令从 mqm 组中除去该用户: `usermod -G groupa,groupb user-name`
，其中 *user-name* 是用户名。

授权的工作方式

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。

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迁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 73 页的表 8， 第 73 页的表 9， 第 74 页的表 10 和 第 74 页的表 11 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 *8: MQCONN* 调用所需的安全性授权

需要授权:	**队列对象 (第 74 页的『1』)**	**进程对象**	**队列管理器对象**
MQCONN	不适用	不适用	MQZAO_CONNECT

表 *9: MQOPEN* 调用所需的安全性授权

需要授权:	**队列对象 (第 74 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
MQOO_SAVE_ ALL_CONTEXT **(第 75 页的『2』)**	MQZAO_INPUT	不适用	不适用
MQOO_OUTPUT (正常队列) **(第 75 页的『3』)**	MQZAO_OUTPUT	不适用	不适用
MQOO_PASS_ IDENTITY_CONTEXT **(第 75 页的『4』)**	MQZAO_PASS_ IDENTITY_CONTEXT	不适用	不检查
MQOO_PASS_ALL_ 上下文 **(第 75 页的『4』， 第 75 页的『5』)**	MQZAO_PASS _ALL_CONTEXT	不适用	不检查
MQOO_SET_ IDENTITY_CONTEXT **(第 75 页的『4』， 第 75 页的『5』)**	MQZAO_SET_ IDENTITY_CONTEXT	不适用	MQZAO_SET_ IDENTITY_CONTEXT **(第 75 页的『6』)**
MQOO_SET_ ALL_CONTEXT **(第 75 页的『4』， 第 75 页的『7』)**	MQZAO_SET_ ALL_CONTEXT	不适用	MQZAO_SET_ ALL_CONTEXT **(第 75 页的『6』)**

表 *9: MQOPEN* 调用所需的安全性授权 *(继续)*			
需要授权:	**队列对象 (第 74 页的『1』)**	**进程对象**	**队列管理器对象**
MQOO_OUTPUT (传输队列) **(第 75 页的『8』)**	MQZAO_SET_ ALL_CONTEXT	不适用	MQZAO_SET_ ALL_CONTEXT **(第 75 页的『6』)**
MQOO_SET	MQZAO_SET	不适用	不检查
MQOO_ALTERNATE_ USER_AUTHORITY	**(第 75 页的『9』)**	**(第 75 页的『9』)**	MQZAO_ALTERNATE_ USER_AUTHORITY **(第 75 页的『9』, 第 75 页的『10』)**

表 *10: MQPUT1* 调用所需的安全性授权			
需要授权:	**队列对象 (第 74 页的『1』)**	**进程对象**	**队列管理器对象**
MQPMO_PASS_ IDENTITY_CONTEXT	MQZAO_PASS_ IDENTITY_CONTEXT **(第 75 页的『11』)**	不适用	不检查
MQPMO_PASS_ALL _CONTEXT	MQZAO_PASS_ ALL_CONTEXT **(第 75 页的『11』)**	不适用	不检查
MQPMO_SET_ IDENTITY_CONTEXT	MQZAO_SET_ IDENTITY_CONTEXT **(第 75 页的『11』)**	不适用	MQZAO_SET_ IDENTITY_CONTEXT **(第 75 页的『6』)**
MQPMO_SET_ ALL_CONTEXT	MQZAO_SET_ ALL_CONTEXT **(第 75 页的『11』)**	不适用	MQZAO_SET_ ALL_CONTEXT **(第 75 页的『6』)**
(传输队列) **(第 75 页的『8』)**	MQZAO_SET_ ALL_CONTEXT	不适用	MQZAO_SET_ ALL_CONTEXT **(第 75 页的『6』)**
MQPMO_ALTERNATE_ USER_AUTHORITY	**(第 75 页的『12』)**	不适用	MQZAO_ALTERNATE_ USER_AUTHORITY **(第 75 页的『10』)**

表 *11: MQCLOSE* 调用所需的安全性授权			
需要授权:	**队列对象 (第 74 页的『1』)**	**进程对象**	**队列管理器对象**
MQCO_DELETE	MQZAO_DELETE **(第 75 页的『13』)**	不适用	不适用
MQCO_DELETE _PURGE	MQZAO_DELETE **(第 75 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 75 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. 仅当满足以下两个条件时，才会执行此检查:
    - 正在关闭并删除永久动态队列。
    - 返回所使用对象句柄的 MQOPEN 调用未创建队列。

    否则，将不进行检查。

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

Object	需要授权
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 80 页的『1』)

Object	需要授权
队列	MQZAO_CREATE (第 80 页的『2』)
Topic	MQZAO_CREATE (第 80 页的『2』)
进程	MQZAO_CREATE (第 80 页的『2』)
队列管理器	不适用
名称列表	MQZAO_CREATE (第 80 页的『2』)
认证信息	MQZAO_CREATE (第 80 页的『2』)
通道	MQZAO_CREATE (第 80 页的『2』)
客户机连接通道	MQZAO_CREATE (第 80 页的『2』)
侦听器	MQZAO_CREATE (第 80 页的『2』)
服务	MQZAO_CREATE (第 80 页的『2』)
通信信息	MQZAO_CREATE (第 80 页的『2』)

DEFINE 对象 REPLACE (第 80 页的『1』， 第 80 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

Object	需要授权
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
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
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

命令	Object	需要授权
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

命令	Object	需要授权
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

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

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
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 (未替换) (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
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 86 页的『2』)**

复制 对象 (含替换) (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

Object	需要授权
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

Object	需要授权
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
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 对象 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
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

命令	Object	需要授权
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	
Reset Cluster	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	
恢复队列管理器集群	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

命令	Object	需要授权
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。

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 WebSphere 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 WebSphere MQ 时这些差异可能如何影响 IBM WebSphere MQ 安全性的描述。

SPI 通道出口程序

WebSphere MQ for Windows 提供了一个安全出口程序，可用于消息和 MQI 通道。 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 WebSphere MQ V 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 WebSphere MQ MQI 客户机应用程序。 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 WebSphere MQ Version 7.0 支持的所有 Windows 平台上受支持。

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与 WebSphere MQ MQI 客户机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。

在 *Windows* 上收到 "找不到组" 错误时

发生此问题的原因可能是，当 Windows 服务器升级到域控制器或从域控制器降级时， WebSphere MQ 将失去对本地 mqm 组的访问权。 要解决此问题，请重新创建本地 mqm 组。

症状是指示缺少本地 mqm 组的错误，例如:

```
>crtmqm qm0
AMQ8066:Local mqm group not found.
```

更改服务器与域控制器之间的机器状态可能会影响 WebSphere MQ 的操作，因为 WebSphere MQ 使用本地定义的 mqm 组。 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。

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 WebSphere 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 WebSphere MQ 用户标识所需的权限级别。

当您在 *Windows* 上迁到 *IBM WebSphere MQ* 和域控制器问题时

将 Windows 服务器提升到域控制器时，安全设置可能会出现某些问题。

将 Windows 2000， Windows 2003 或 Windows Server 2008 服务器提升到域控制器时，将向您提供选择与用户和组许可权相关的缺省或非缺省安全设置的选项。 此选项控制任意用户是否能够从 Active Directory 中检索组成员资格。 由于 WebSphere MQ 依赖于组成员资格信息来实现其安全策略，因此执行 WebSphere 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。

在 Windows 2000 上，使用缺省安全选项创建域时， WebSphere MQ 在安装过程中创建的缺省用户标识可以根据需要获取其他用户的组成员资格。 然后产品正常安装，创建缺省对象，如果需要，队列管理器可以确定本地用户和域用户的访问权限。

在 Windows 2000 上，使用非缺省安全选项创建域时，或者在使用缺省安全选项创建域时，在 Windows 2003 和 Windows Server 2008 上， WebSphere MQ 在安装期间创建的用户标识无法始终确定所需的组成员资格。 在这种情况下，您需要知道:

- 具有非缺省值的 Windows 2000 或具有缺省安全许可权的 Windows 2003 和 Windows Server 2008 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 WebSphere MQ Windows 服务配置为在域用户下运行

具有非缺省值的 *Windows 2000* 域，或具有缺省安全许可权的 *Windows 2003* 和 *Windows Server 2008* 域在这些操作系统上安装 WebSphere MQ 的行为会有所不同，取决于是本地用户还是域用户执行安装。

如果 **本地** 用户安装 WebSphere MQ，那么 "准备 WebSphere MQ 向导" 会检测到为 IBM WebSphere 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。 "准备 WebSphere MQ 向导" 会询问用户有关网络配置的问题，以确定是否在 Windows 2000 或更高版本上运行的域控制器上定义了其他用户帐户。 如果是这样，那么 IBM WebSphere 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。 "准备 WebSphere MQ 向导" 会提示用户输入此用户的帐户详细信息。 其联机帮助提供可发送给域管理员的所需域用户帐户的详细信息。

如果 **域** 用户安装 WebSphere MQ，那么 "准备 WebSphere MQ 向导" 将检测到为 IBM WebSphere 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， "准备 WebSphere MQ 向导" 始终提示用户输入域用户帐户的帐户详细信息，以供 IBM WebSphere MQ Windows 服务使用。

当 IBM WebSphere MQ Windows 服务需要使用域用户帐户时，在使用 "准备 WebSphere MQ 向导" 进行配置之前， WebSphere MQ 无法正常运行。 "准备 WebSphere MQ 向导" 不允许用户继续执行其他任务，直到使用合适的帐户配置了 Windows 服务为止。

如果 Windows 2000 域已配置为具有非缺省安全许可权，那么使 WebSphere MQ 能够正常工作的通常解决方案是使用适当的域用户帐户对其进行配置，如上文所述。

请参阅 为 WebSphere MQ 创建和设置域帐户 以获取更多信息。

将 *IBM WebSphere MQ* 服务配置为在 *Windows* 上的域用户下运行
使用 "准备 IBM WebSphere MQ " 向导来输入域用户帐户的帐户详细信息。 或者，您可以使用 "计算机管理" 面板来变更特定于安装的 IBM WebSphere MQ 服务的 **登录** 详细信息。

有关更多信息，请参阅 更改 IBM WebSphere MQ Windows 服务用户帐户的密码

将安全模板文件应用于 *Windows*

应用模板可能会影响应用于 WebSphere 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 WebSphere MQ 之前应用该模板。

Windows 支持基于文本的安全模板文件，您可以使用这些文件将统一安全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 WebSphere 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 WebSphere MQ 之前配置机器。

如果将高度安全的模板应用于已安装 WebSphere MQ 的机器，那么将除去您对 WebSphere 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)。

嵌套组

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 WebSphere 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 域处于 *Windows 2000* 混合 功能级别。 (Windows Server 2003 ， Windows XP ， Windows Vista 和 Windows Server 2008 都遵循 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。

除了 Active Directory 需求外，还对 WebSphere MQ 所使用的标识施加了进一步的限制。 WebSphere MQ 使用的网络 API 不支持域功能级别支持的所有配置。 因此， WebSphere 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。

为连接到 *IBM WebSphere MQ* 的 *Windows* 应用程序配置其他权限

在可以授予对应用程序进程的同步访问权之前，运行 IBM WebSphere MQ 进程的帐户可能需要其他授权。

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 WebSphere MQ ，那么可能会迁到问题。

IBM WebSphere MQ 需要对应用程序进程的同步访问权，以便协调某些操作。 APAR IC35116 已更改 IBM WebSphere MQ ，因此指定了相应的特权。 但是，运行 IBM WebSphere 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。

当服务器应用程序首次尝试连接到队列管理器 IBM WebSphere MQ 时，将修改该进程以授予 IBM WebSphere MQ 管理员同步权限。 要配置对运行 IBM WebSphere MQ 进程的用户标识的其他权限，请完成以下步骤:

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" ，单击 "调试程序"。
2. 双击 "调试程序" ，然后将您的 IBM WebSphere 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。

在 HP Integrity NonStop Server 上设置安全性

特定于 HP Integrity NonStop Server 系统的安全注意事项。

IBM WebSphere MQ Client for HP Integrity NonStop Server 支持传输层安全性 (TLS) 和安全套接字层 (SSL) 协议，以在连接到队列管理器时提供链路级别安全性。 通过使用 OpenSSL 的实现来支持这些协议。 OpenSSL 需要随机数据源以提供强大的加密操作。

OpenSSL

IBM WebSphere MQ Client for HP Integrity NonStop Server 的 OpenSSL 安全性概述。

OpenSSL 工具箱是用于通过网络进行安全通信的安全套接字层 (SSL) 和传输层安全性 (TLS) 协议的开放式源代码实现。

工具箱由 OpenSSL 项目开发。 有关 OpenSSL 项目的更多信息，请参阅 https://www.openssl.org。 HP Integrity NonStop Server 的 IBM WebSphere MQ 客户机包含已修改版本的 OpenSSL 库和 **openssl** 命令。 库和 **openssl** 命令从 OpenSSL 工具箱 1.0.1c 移植，并且仅作为对象代码提供。 未提供源代码。

OpenSSL 库由 IBM WebSphere MQ 客户机应用程序根据需要动态装入。 仅支持 IBM WebSphere MQ 提供的 OpenSSL 库用于 IBM WebSphere MQ 客户机应用程序。

openssl 命令 (可用于证书管理目的) 安装在 OSS 目录 `opt_installation_path/opt/mqm/bin` 中。

通过使用 **openssl** 命令，您可以创建和管理具有各种公共数据格式的密钥和数字证书，并执行简单认证中心 (CA) 任务。

OpenSSL 处理的密钥和证书数据的缺省格式是隐私增强邮件 (PEM) 格式。 PEM 格式的数据是 base64 编码的 ASCII 数据。 因此，可以使用基于文本的系统 (如电子邮件) 来传输数据，也可以使用文本编辑器和 Web 浏览器来剪切和粘贴数据。 PEM 是基于文本的加密交换的因特网标准，在因特网 RFC 1421,1422,1423 和 1424 中指定。 IBM WebSphere MQ 假定扩展名为 `.pem` 的文件包含 PEM 格式的数据。 PEM 格式的文件可以包含多个证书和其他编码对象，并且可以包含注释。

其他操作系统上的 IBM WebSphere MQ SSL 支持可能要求使用 "专有编码规则" (DER) 对文件中的密钥和证书数据进行编码。 DER 是一组编码规则，用于在安全通信中使用 ASN.1 表示法。 使用 DER 编码的数据是二进制数据，使用 DER 编码的密钥和证书数据的格式也称为 PKCS#12 或 PFX。 包含此数据的文件通常具有扩展名 `.p12` 或 `.pfx`。 **openssl** 命令可以在 PEM 和 PKCS#12 格式之间进行转换。

熵守护程序

OpenSSL 需要随机数据源以提供强大的加密操作。 随机数生成是通常由操作系统或系统范围的守护进程提供的功能。 HP Integrity NonStop Server 操作系统在操作系统中不提供此功能。

当您使用 IBM WebSphere MQ client for HP Integrity NonStop Server 随附的 SSL 和 TLS 支持时，需要一个称为熵守护程序的进程来提供随机数据的源。 当您启动需要 SSL 或 TLS 的客户机通道时， OpenSSL 期望熵守护程序正在 OSS 文件系统 (`/etc/egd-pool`) 中的套接字上运行并提供其服务。

IBM WebSphere MQ Client for HP Integrity NonStop Server 未提供熵守护程序。 使用以下熵守护程序测试 HP Integrity NonStop Server 的 IBM WebSphere MQ 客户机:

- amqjkdm0 (由 IBM WebSphere MQ 5.3 服务器提供)
- `/usr/local/bin/prngd` (V 0.9.27，由 HP Integrity NonStop Server 开放式源代码技术库提供)

设置 IBM WebSphere MQ MQI 客户机安全性

您必须考虑 IBM WebSphere MQ MQI 客户机安全性，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。

  您应该在 SSL 或 TLS 中使用相互认证。 有关更多信息，请参阅 第 92 页的『使用 SSL 或 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 131 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 154 页的『将客户机断言的用户标识映射到 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 33 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。

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 SSL 或 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)。 如果客户机进程当前没有活动的 SSL 或 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 SSL 连接都必须仅使用与此值关联的 CipherSpecs 。 这在此连接以及所有其他 SSL 或 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 WebSphere MQ 所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。 WebSphere MQ 不保证拒绝所有此类连接，您负责确定 WebSphere MQ 配置是否符合 FIPS。

相关概念

第 22 页的『适用于 UNIX， Linux 和 Windows 的联邦信息处理标准 (FIPS)』
在 Windows UNIX and Linux 系统上的 SSL 或 TLS 通道上需要密码术时， WebSphere MQ 使用名为 IBM Crypto for C (ICC) 的密码术包。 在 Windows UNIX and Linux 平台上， ICC 软件已通过美国国家标准与技术研究所的联邦信息处理标准 (FIPS) Cryptomodule Validation Program ，级别为 140-2。

客户机配置文件的 SSL 节

相关参考

FipsRequired (MQLONG)

MQSSLFIPS

在 AIX 上运行具有 GSKit V8.0 的多个安装的 SSL 或 TLS 客户机应用程序

在具有多个 GSKit V8.0 安装的 AIX 系统上运行时， AIX 上的 SSL 或 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 V8.0 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 SSL 或 TLS 时返回 MQRC_CHANNEL_CONFIG_ERROR 。 对于失败的客户机应用程序， /var/mqm/errors 记录记录错误 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 WebSphere MQ 客户机从两个不同的 GSKit V8.0 安装装入一组混合库。 在 Db2 环境中执行 IBM WebSphere 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 WebSphere MQ 库目录，以便 IBM WebSphere 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 WebSphere MQ environment)

在 UNIX, Linux, and Windows 系统上为 SSL 或 TLS 设置通信

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 UNIX, Linux 和 Windows 系统上，可以使用自签名证书执行测试。

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 95 页的『在 UNIX, Linux, and Windows 系统上使用 SSL 或 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 WebSphere MQ 实现， SSL 或 TLS 服务器始终从客户机请求证书。

在 UNIX， Linux 和 Windows 系统上，仅当 SSL 或 TLS 客户机具有以正确 IBM WebSphere MQ 格式标注的证书时，才会发送证书:

- 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 WebSphere 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 WebSphere 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，请参阅第 171 页的『使用 SSL 或 TLS 连接两个队列管理器』。

使用 SSL 或 TLS

这些主题提供了有关执行与将 SSL 或 TLS 与 IBM WebSphere 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 120 页的『识别和认证用户』
- 第 131 页的『授予对对象的访问权』
- 第 171 页的『消息的机密性』
- 第 189 页的『消息的数据完整性』
- 第 205 页的『确保集群安全』

在 HP Integrity NonStop Server 上使用 SSL 或 TLS

描述 HP Integrity NonStop Server OpenSSL 安全性实现的 IBM WebSphere MQ 客户机，包括安全服务，组件，受支持的协议版本，受支持的 CipherSpecs 和不受支持的安全性功能。

IBM WebSphere MQ SSL 和 TLS 支持为客户机通道提供以下安全服务:

- 服务器认证和 (可选) 客户机认证。
- 对流经通道的数据进行加密和解密。
- 对流经通道的数据进行完整性检查。

IBM WebSphere MQ Client for HP Integrity NonStop Server 随附的 SSL 和 TLS 支持包含以下组件:

- OpenSSL 库和 **openssl** 命令。
- IBM WebSphere MQ password stash 命令， **amqrsslc**。

IBM WebSphere MQ Client for HP Integrity NonStop Server 未随附用于 SSL 或 TLS 客户机通道操作的以下必需组件:

- 用于为 OpenSSL 密码术提供随机数据源的熵守护程序。

支持的协议版本

HP Integrity NonStop Server 的 IBM WebSphere MQ 客户机支持以下协议版本:

- SSL 3.0
- TLS 1.0
- TLS 1.2

受支持的 CipherSpecs

HP Integrity NonStop Server 的 IBM WebSphere MQ 客户机支持以下 CipherSpecs 版本:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- RC4_SHA_US
- RC4_MD5_US
- TRIPLE_DES_SHA_US
- TLS_RSA_WITH_3DES_EDE_CBC_SHA（不推荐）
- DES_SHA_EXPORT1024
- RC4_56_SHA_EXPORT1024
- RC4_MD5_EXPORT
- RC2_MD5_EXPORT
- DES_SHA_EXPORT
- TLS_RSA_WITH_DES_CBC_SHA
- NULL_SHA
- NULL_MD5
- FIPS_WITH_DES_CBC_SHA
- FIPS_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_RSA_WITH_NULL_SHA256
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_GCM_SHA384
- ECDHE_ECDSA_AES_128_CBC_SHA256
- ECDHE_ECDSA_AES_256_CBC_SHA384
- ECDHE_RSA_AES_128_CBC_SHA256
- ECDHE_RSA_AES_256_CBC_SHA384
- ECDHE_ECDSA_AES_128_GCM_SHA256
- ECDHE_ECDSA_AES_256_GCM_SHA384
- ECDHE_RSA_AES_128_GCM_SHA256
- ECDHE_RSA_AES_256_GCM_SHA384

不受支持的安全性功能

HP Integrity NonStop Server 的 IBM WebSphere MQ 客户机当前不支持:

- PKCS#11 加密硬件支持
- LDAP 证书撤销列表检查
- OCSP 联机证书状态协议检查
- FIPS 140-2， NSA SUITE B 密码套件控件

证书管理

使用一组文件来存储数字证书和证书撤销信息。

IBM WebSphere MQ SSL 和 TLS 支持使用一组文件来存储数字证书和证书撤销信息。 这些文件位于通过在 MQCONNX 调用上通过 *MQSSLKEYR* 环境变量传递的 MQSCO 结构中的 KeyRepository 字段以编程方式指定的目录中，或者位于使用 SSLKeyRepository 属性的 `mqclient.ini` 的 SSL 节中。

MQSCO 结构优先于 MQSSLKEYR 环境变量，该环境变量优先于 ini 文件节值。

要点: 密钥存储库位置指定 HP Integrity NonStop Server 平台上的目录位置而不是文件名。

IBM WebSphere MQ Client for HP Integrity NonStop Server 在密钥存储库位置中使用以下区分大小写的指定文件:

- 第 94 页的『个人证书库』
- 第 94 页的『证书信任库』
- 第 94 页的『口令隐藏文件』
- 第 95 页的『证书撤销列表文件』

个人证书库
个人证书库文件 `cert.pem`。

此文件包含个人证书和加密专用密钥，供客户机使用，采用 PEM 格式。 当您使用不需要客户机认证的 SSL 或 TLS 通道时，此文件的存在是可选的。 如果通道需要客户机认证，并且在通道定义上指定了 SSLCAUTH (REQUIRED)，那么此文件必须存在并且同时包含证书和加密专用密钥。

必须对此文件设置文件许可权，以允许对证书库的所有者进行读访问。

正确格式化的 `cert.pem` 文件必须正好包含两个具有以下页眉和页脚的部分:

```
-----BEGIN PRIVATE KEY-----
Base 64 ASCII encoded private key data here
-----END PRIVATE KEY-----
```

```
-----BEGIN CERTIFICATE-----
Base 64 ASCII encoded certificate data here
-----END CERTIFICATE-----
```

加密专用密钥的口令存储在口令隐藏文件 `Stash.sth` 中。

证书信任库
证书信任库文件 `trust.pem`。

此文件包含验证客户机所连接的队列管理器所使用的个人证书所需的证书 (PEM 格式)。 证书信任库对于所有 SSL 或 TLS 客户机通道都是必需的。

必须设置文件许可权以限制对此文件的写访问权。

格式正确的 `trust.pem` 文件必须包含一个或多个具有以下页眉和页脚的部分:

```
-----BEGIN CERTIFICATE-----
Base 64 ASCII encoded certificate data here
-----END CERTIFICATE-----
```

口令隐藏文件
口令隐藏文件 `Stash.sth`。

此文件是 IBM WebSphere MQ 专用的二进制格式，包含加密口令，供您访问 `cert.pem` 文件中保存的专用密钥时使用。 专用密钥本身存储在 `cert.pem` 证书库中。

此文件是通过将 IBM WebSphere MQ **amqrsslc** 命令行工具与 **-s** 参数配合使用来创建或更改的。 例如，其中目录 `/home/alice` 包含 `cert.pem` 文件:

```
amqrsslc -s /home/alice/cert

  Enter password for Keystore /home/alice/cert.pem :
    password

  Stashed the password in file /home/alice/Stash.sth
```

必须对此文件设置文件许可权，以允许对关联个人证书库的所有者进行读访问。

证书撤销列表文件
证书撤销列表文件 `crl.pem`。

此文件包含客户机用于验证 PEM 格式的数字证书的证书撤销列表 (CRL)。 此文件的存在是可选的。 如果此文件不存在，那么在验证证书时不会执行证书撤销检查。

必须设置文件许可权以限制对此文件的写访问权。

格式正确的 `crl.pem` 文件必须包含一个或多个具有以下页眉和页脚的部分:

```
-----BEGIN X509 CRL-----
Base 64 ASCII encoded CRL data here
-----END X509 CRL-----
```

在 UNIX, Linux, and Windows 系统上使用 SSL 或 TLS

在 UNIX， Linux 和 Windows 系统上，安全套接字层 (SSL) 支持随 IBM WebSphere MQ 一起安装。

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。

使用 *iKeyman*， *iKeycmd*， *runmqakm* 和 *runmqckm*

在 UNIX， Linux 和 Windows 系统上，使用 iKeyman GUI 或从命令行使用 iKeycmd 或 runmqakm 来管理密钥和数字证书。

- 对于 **UNIX and Linux** 系统:
  - 使用 **strmqikm** 命令来启动 iKeyman GUI。
  - 使用 **runmqckm** 命令通过 iKeycmd 命令行界面执行任务。
  - 使用 **runmqakm** 命令通过 runmqakm 命令行界面执行任务。 **runmqakm** 的命令语法与 **runmqckm** 的语法相同。

    如果需要以符合 FIPS 的方式管理 SSL 证书，请使用 **runmqakm** 命令而不是 **runmqckm** 或 **strmqikm** 命令。

  请参阅 管理密钥和证书 ，以获取 **runmqckm** 和 **runmqakm** 命令的命令行界面的完整描述。

  如果您要使用存储在 PKCS #11 加密硬件上的证书或密钥，请注意 iKeycmd 和 iKeyman 是 64 位程序。 PKCS #11 支持所需要的外部模块将装入 64 位进程中，因此您必须安装 64 位 PKCS #11 库以管理加密硬件。 只有 Windows 和 Linux x86 32 位平台例外，因为 iKeyman 和 iKeycmd 程序在这些平台上是 32 位的。

  在以下平台上，其中 JRE 在产品的较低版本中为 32 位，但仅在 IBM WebSphere MQ Version 7.5 中为 64 位，您可能需要安装适用于 **iKeyman** 和 **iKeycmd** JRE 的寻址方式的其他 PKCS#11 驱动程序。 这是因为 PKCS#11 驱动程序必须使用与 JRE 相同的寻址方式。 下表显示了 IBM WebSphere MQ Version 7.5 JRE 寻址方式。

表 *12: IBM WebSphere MQ Version 7.5 JRE* 寻址方式

平台	**JRE 寻址方式**
Windows (32 位或 64 位)	32
Linux for System x 32 位	32
Linux for System x 64 位	64
Linux for System p	64
针对 System z 的 Linux	64
HP-UX	64
Solaris SPARC	64
Solaris x86-64	64

表 12: IBM WebSphere MQ Version 7.5 JRE 寻址方式 (继续)	
平台	**JRE 寻址方式**
AIX	64

在运行 **strmqikm** 命令以启动 iKeyman GUI 之前，请确保您正在能够运行 X Window System 的机器上工作，并执行以下操作:

- 设置 DISPLAY 环境变量，例如:

```
export DISPLAY=mypc:0
```

- 确保 PATH 环境变量包含 **/usr/bin** 和 **/bin**。 这也是 **runmqckm** 和 **runmqakm** 命令所必需的。 例如:

```
export PATH=$PATH:/usr/bin:/bin
```

- 对于 **Windows** 系统:
  - 使用 **strmqikm** 命令来启动 iKeyman GUI。
  - 使用 **runmqckm** 命令通过 iKeycmd 命令行界面执行任务。

    如果需要以符合 FIPS 的方式管理 SSL 证书，请使用 **runmqakm** 命令而不是 **runmqckm** 或 **strmqikm** 命令。

要在 UNIX， Linux 或 Windows 系统上请求 SSL 跟踪，请参阅 strmqtrc。

相关参考

runmqckm 和 runmqakm 命令

在 *UNIX, Linux, and Windows* 系统上设置密钥存储库

您可以使用 iKeyman 用户界面或使用 **iKeycmd** 或 **runmqakm** 命令来设置密钥存储库。

关于此任务

SSL 或 TLS 连接在连接的每一端都需要 密钥存储库 。 每个 IBM WebSphere MQ 队列管理器和 IBM WebSphere 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 20 页的『SSL 或 TLS 密钥存储库』。

在 UNIX, Linux, and Windows 系统上，数字证书存储在使用 **iKeyman** 用户界面或使用 **iKeycmd** 或 **runmqakm** 命令管理的密钥数据库文件中。 这些数字证书具有标签。 特定标签将个人证书与队列管理器或 IBM WebSphere MQ MQI client 相关联。 SSL 和 TLS 将该证书用于认证目的。 在 UNIX, Linux, and Windows 系统上， IBM WebSphere MQ 使用 ibmwebspheremq 作为标签前缀，以避免与其他产品的证书混淆。 前缀后跟队列管理器或 IBM WebSphere MQ MQI client 用户登录标识的名称 (已更改为小写)。 确保以小写形式指定整个证书标签。

密钥数据库文件名包含路径和词干名称:

- 在 UNIX 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_INSTALLATION_PATH*\Qmgrs*queue_manager_name*\ssl，其中 *MQ_INSTALLATION_PATH* 是 IBM WebSphere MQ 的安装目录。 例如，C:\program files\IBM\WebSphere MQ\Qmgrs\QM1\ssl。

  缺省主干名称为 key。 (可选) 您可以选择自己的路径和词干名称，但扩展名必须为 .kdb。

  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 WebSphere MQ 客户机，没有缺省路径或主干名称。 严格控制对此文件的访问。 扩展必须为 .kdb。

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 版本 2。

有关检查和指定密钥数据库文件名的信息，请参阅 第 100 页的『在 UNIX， Linux 或 Windows 系统上更改队列管理器的密钥存储库位置』 。 可以在创建密钥数据库文件之前或之后指定密钥数据库文件名。

从中运行 **iKeyman** 或 **iKeycmd** 命令的用户标识必须具有创建或更新密钥数据库文件的目录的写许可权。 对于使用缺省 `ssl` 目录的队列管理器，从中运行 **iKeyman** 或 **iKeycmd** 的用户标识必须是 mqm 组的成员。 对于 IBM WebSphere MQ MQI client，如果从不同于客户机运行时的用户标识运行 **iKeyman** 或 **iKeycmd** ，那么必须变更文件许可权以允许 IBM WebSphere MQ MQI client 在运行时访问密钥数据库文件。 有关更多信息，请参阅第 98 页的『在 Windows 上访问和保护密钥数据库文件』或第 98 页的『访问和保护 UNIX and Linux 系统上的密钥数据库文件』。

在 **iKeyman** 或 **iKeycmd** V 7.0 中，将使用一组预定义的认证中心 (CA) 证书自动填充新的密钥数据库。 在 **iKeyman** 或 **iKeycmd** 版本 8.0 中，不会自动填充密钥数据库，这将使初始设置更加安全，因为您只需要在密钥数据库文件中包含所需的 CA 证书。

注: 由于 GSKit 版本 8.0 的行为发生了此更改，导致不再将 CA 证书自动添加到存储库中，因此必须手动添加首选 CA 证书。 此行为更改使您能够更精细地控制所使用的 CA 证书。 请参阅第 99 页的『将缺省 CA 证书添加到 GSKit 版本为 8.0 的 UNIX, Linux, and Windows 系统上的空密钥存储库中』。

您可以使用命令行或 **strmqikm** (iKeyman) 用户界面来创建密钥数据库。

注: 如果必须以符合 FIPS 的方式管理 TLS 证书，请使用 **runmqakm** 命令。 **strmqikm** 用户界面未提供符合 FIPS 的选项。

过程

使用命令行创建密钥数据库。

1. 运行下列任一命令:
   - 在 UNIX, Linux, and Windows 系统上:

     ```
     runmqckm -keydb -create -db filename -pw password -type cms -stash
     ```

   - 使用 runmqakm:

     ```
     runmqakm -keydb -create -db filename -pw password -type cms
         -stash -fips -strong
     ```

   其中:

   -db 文件名
   指定 CMS 密钥数据库的标准文件名，并且必须具有文件扩展名 `.kdb`。

   -pw *password*
   指定 CMS 密钥数据库的密码。

   -type *cms*
   指定数据库的类型。 (对于 IBM WebSphere MQ，必须为 `cms`。)

   -stash
   将密钥数据库密码保存到文件。

   -无花果
   禁止使用 BSafe 加密库。 仅使用 ICC 组件，并且必须以 FIPS 方式成功初始化此组件。 在 FIPS 方式下， 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 0x20 - 0x7E 范围内的标准 ASCII 可打印字符集中。

或者，使用 **strmqikm** (iKeyman) 用户界面创建密钥数据库。

2. 在 UNIX and Linux 系统上，以 root 用户身份登录。 在 Windows 系统上，以管理员身份或 MQM 组的成员身份登录。
3. 通过运行 **strmqikm** 命令来启动 iKeyman 用户界面。
4. 从 **密钥数据库文件** 菜单中，单击 **新建**。

   此时将打开 "新建" 窗口。
5. 单击**密钥数据库类型**，并选择 **CMS**（证书管理系统）。
6. 在 **文件名** 字段中，输入文件名。

   此字段已包含文本 key.kdb。 如果您的主干名称为 key，请保持此字段不变。 如果指定了其他主干名称，请将 key 替换为您的主干名称。 但是，不得更改 .kdb 扩展。
7. 在 **位置** 字段中，输入路径。

   例如：

   - 对于队列管理器: /var/mqm/qmgrs/QM1/ssl (在 UNIX and Linux 系统上) 或 C:\Program Files\IBM\WebSphere MQ\qmgrs\QM1\ssl (在 Windows 系统上)。

     路径必须与队列管理器的 **SSLKeyRepository** 属性值匹配。
   - 对于 IBM WebSphere MQ 客户机: /var/mqm/ssl (在 UNIX and Linux 系统上) 或 C:\mqm\ssl (在 Windows 系统上)。
8. 单击**打开**。

   这样会打开“密码提示”窗口。
9. 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。
10. 选中 **将密码隐藏到文件中** 复选框。

    注: 如果不隐藏密码，那么尝试启动 SSL 或 TLS 通道将失败，因为它们无法获取访问密钥数据库文件所需的密码。
11. 单击**确定**。

    将打开 "个人证书" 窗口。
12. 设置访问许可权，如 第 98 页的『在 Windows 上访问和保护密钥数据库文件』 或 第 98 页的『访问和保护 UNIX and Linux 系统上的密钥数据库文件』中所述。

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*.kdb， *key*.sth， *key*.crl 和 *key*.rdb(其中 *key* 是密钥数据库的主干名称) 的访问控制，以向一组受限用户授予权限。

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。

访问和保护 *UNIX and Linux* 系统上的密钥数据库文件

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*.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 -----)。

将缺省 *CA* 证书添加到 *GSKit* 版本为 *8.0* 的 *UNIX, Linux, and Windows* 系统上的空密钥存储库中
遵循此过程将一个或多个缺省 CA 证书添加到 GSKit 版本为 8 的空密钥存储库。

在 GSKit 版本 7.0 中，创建新密钥存储库时的行为是自动为常用认证中心添加一组缺省 CA 证书。 对于 GSKit V 8 ，此行为已更改，因此不再自动将 CA 证书添加到存储库中。 现在需要用户手动将 CA 证书添加到密钥存储库中。

使用 iKeyman

在要添加 CA 证书的机器上执行以下步骤:

1. 使用 **strmqikm** 命令启动 iKeyman GUI (在 UNIX， Linux 和 Windows 系统上)。
2. 从**密钥数据库文件**菜单中，单击**打开**。 这样会显示“打开”窗口。
3. 单击**密钥数据库类型**，并选择 **CMS**（证书管理系统）。
4. 单击**浏览**以浏览至包含密钥数据库文件的目录。
5. 选择要向其添加该证书的密钥数据库文件，例如，key.kdb。
6. 单击**打开**。 这样会打开“密码提示”窗口。
7. 输入在创建密钥数据库时设置的密码，然后单击**确定**。 密钥数据库文件的名称显示在**文件名**字段中。
8. 在**密钥数据库内容**字段中，选择**签署者证书**。
9. 单击 **填充**。 将打开 "添加 CA 的证书" 窗口。
10. 可以添加到存储库的 CA 证书将以分层树结构显示。 选择您希望信任其 CA 证书的组织的顶级条目，以查看有效 CA 证书的完整列表。
11. 从列表中选择要信任的 CA 证书，然后单击 **确定**。 这些证书将添加到密钥存储库。

使用命令行

使用以下命令列出，然后使用 iKeycmd 添加 CA 证书:

- 发出以下命令以列出缺省 CA 证书以及发放这些证书的组织:

```
runmqckm -cert -listsigners
```

- 发出以下命令以添加 *label* 字段中指定的组织的所有 CA 证书:

```
runmqckm -cert -populate -db filename -pw password -label label
```

其中:

-db *filename*	是密钥数据库的标准路径名。
-pw *password*	是密钥数据库的密码。
-label *label*	是证书附加的标签。

注: 将 CA 证书添加到密钥存储库会导致 WebSphere MQ 信任该 CA 证书签署的所有个人证书。 请仔细考虑您希望信任哪些认证中心，并仅添加认证客户机和管理器所需的 CA 证书集。 建议不要添加完整的缺省 CA 证书集，除非这是安全策略的明确要求。

在 *UNIX, Linux, and Windows* 系统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 WebSphere 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 UNIX and Linux 系统上: `/var/mqm/qmgrs/QM1/ssl/key`，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` 是主干名称。
   MQ_INSTALLATION_PATH 表示安装了 WebSphere MQ 的高级目录。

在 *UNIX*， *Linux* 或 *Windows* 系统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 UNIX and Linux 系统上:

```
ALTER QMGR SSLKEYR('/var/mqm/qmgrs/QM1/ssl/MyKey')
```

密钥数据库文件具有标准文件名: `/var/mqm/qmgrs/QM1/ssl/MyKey.kdb`

在 Windows 上:

```
ALTER QMGR SSLKEYR('C:\Program Files\IBM\WebSphere MQ\Qmgrs\QM1\ssl\Mykey')
```

密钥数据库文件具有标准文件名: `C:\Program Files\IBM\WebSphere MQ\Qmgrs\QM1\ssl\Mykey.kdb`

注意: 请确保在 SSLKEYR 关键字上的文件名中不包含 `.kdb` 扩展名，因为队列管理器会自动追加此扩展名。

您还可以使用 WebSphere MQ Explorer 或 PCF 命令来变更队列管理器的属性。

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 111 页的『将个人证书导入到 UNIX, Linux, and Windows 系统上的密钥存储库中』中所述。

在 *UNIX, Linux, and Windows* 系统上查找 *IBM WebSphere MQ MQI* 客户机的密钥存储库

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 环境变量以获取 IBM WebSphere MQ MQI 客户机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 100 页的『在 UNIX, Linux, and Windows 系统上指定 IBM WebSphere MQ MQI 客户机的密钥存储库位置』中所述。 MQCONNX 调用中设置的值将覆盖 MQSSLKEYR 的值。

在 *UNIX, Linux, and Windows* 系统上指定 *IBM WebSphere MQ MQI* 客户机的密钥存储库位置

IBM WebSphere MQ MQI 客户机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。

您可以通过以下两种方法之一来指定 IBM WebSphere MQ MQI 客户机的密钥数据库文件的位置:

- 设置 MQSSLKEYR 环境变量。 例如，在 UNIX and Linux 系统上:

```
export MQSSLKEYR=/var/mqm/ssl/key
```

密钥数据库文件具有标准文件名:

```
/var/mqm/ssl/key.kdb
```

在 Windows 上:

```
set MQSSLKEYR=C:\Program Files\IBM\WebSphere MQ\ssl\key
```

密钥数据库文件具有标准文件名:

```
C:\Program Files\IBM\WebSphere MQ\ssl\key.kdb
```

注: .kdb 扩展名是文件名的必需部分，但不包含在环境变量的值中。

- 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 。

当对证书或证书库的更改在 *UNIX*， *Linux* 或 *Windows* 系统上生效时。

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，对密钥数据库文件中的证书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先运行 SSL 通道时。
- 当新的入站 TCP/IP 单通道进程首次接收到启动 SSL 通道的请求时。
- 发出 MQSC 命令 REFRESH SECURITY TYPE (SSL) 以刷新 Websphere MQ SSL 环境时。
- 对于客户机应用程序进程，当进程中的最后一个 SSL 连接关闭时。 下一个 SSL 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时，首先运行 SSL 通道。 如果进程池进程已运行 SSL 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的线程运行的通道，当通道启动程序启动或重新启动时，首先运行 SSL 通道。 如果通道启动程序进程已运行 SSL 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 SSL 通道的请求。 如果侦听器已运行 SSL 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 WebSphere MQ Explorer 或 PCF 命令刷新 WebSphere MQ SSL 环境。

在 *UNIX, Linux, and Windows* 系统上创建自签名个人证书

您可以使用 iKeyman， iKeycmd 或 runmqakm 来创建自签名证书。

注: IBM WebSphere 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。

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。

有关可能要使用自签名证书的原因的更多信息，请参阅 第 172 页的『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。 确保创建与需要使用的 CipherSpecs 兼容的证书。 WebSphere MQ 支持三种不同类型的 CipherSpec。 有关详细信息，请参阅 第 28 页的『IBM WebSphere MQ 中的数字证书和 CipherSpec 兼容性』 主题中的 第 30 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_开头的规范)，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， **-sig_alg** EC_ecdsa_with_SHA384 。

使用 iKeyman

iKeyman 未提供符合 FIPS 的选项。 如果需要以符合 FIPS 的方式管理 SSL 或 TLS 证书，请使用 **runmqakm** 命令。

使用以下过程来获取队列管理器或 WebSphere MQ MQI 客户机的自签名证书:

1. 使用 **strmqikm** 命令启动 iKeyman GUI。
2. 从 **密钥数据库文件** 菜单中，单击 **打开**。 将显示 "打开" 窗口。
3. 单击**密钥数据库类型**，并选择 **CMS**（证书管理系统）。
4. 单击**浏览**以浏览至包含密钥数据库文件的目录。
5. 选择要在其中保存证书的密钥数据库文件，例如 key.kdb。
6. 单击**打开**。 将显示 "密码提示" 窗口。
7. 输入在创建密钥数据库时设置的密码，然后单击**确定**。 密钥数据库文件的名称将显示在 **文件名** 字段中。
8. 从 **创建** 菜单中，单击 **新建自签名证书**。 此时将显示 "新建自签名证书" 窗口。
9. 在 **密钥标签** 字段中，输入:
   - 对于队列管理器， ibmwebspheremq 后跟转换为小写的队列管理器的名称。 例如，对于 QM1，ibmwebspheremqqm1 或
   - 对于 WebSphere MQ 客户机， ibmwebspheremq 后跟转换为小写的登录用户标识，例如 ibmwebspheremqmyuserid 。
10. 在 **Distinguished name** 或任何 **Subject alternative name** 字段中输入或选择任何字段的值。
11. 对于剩余的字段，可以接受缺省值，也可以输入或选择新值。 有关专有名称的更多信息，请参阅 第 10 页的『专有名称』。
12. 单击**确定**。 **个人证书** 列表显示您创建的自签名个人证书的标签。

使用命令行

使用以下命令通过 iKeycmd 或 runmqakm 创建自签名个人证书:

- 在 UNIX， Linux 和 Windows 系统上使用 iKeycmd :

```
runmqckm -cert -create -db filename -pw
password -label label
        -dn distinguished_name -size key_size
 -x509version version -expire days
 -sig_alg algorithm
```

  您可以使用 -san_dsname *DNS_names* ， -san_emailaddr *email_addresses* 或 -san_ipaddr *IP_addresses* 来代替 -dn *distinguished_name* 。

- 使用 runmqakm:

```
runmqakm -cert -create -db filename -pw
password -label label
        -dn distinguished_name -size key_size
 -x509version version -expire days

        -fips -sig_alg algorithm
```

-db *filename*	CMS 密钥数据库的标准文件名。
-pw *password*	CMS 密钥数据库的密码。
-label *label*	附加到证书的密钥标签。

-dn *distinguished_name*	以双引号括起的 X.500 专有名称。 至少需要一个属性。 您可以提供多个 OU 或 DC 属性。
-size *key_size*	密钥大小。 对于 iKeycmd，该值可以是 512 或 1024。 对于 runmqakm ，该值可以是 512,1024,2048 或 4096。
-x509version *version*	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 *days*	证书的到期时间 (以天计)。 证书的缺省值为 365 天。
-fips	指定以 FIPS 方式运行命令。 此方式禁用 BSafe 密码库。 仅使用 ICC 组件，并且必须以 FIPS 方式成功初始化此组件。 当处于 FIPS 方式下时，ICC 组件使用已进行 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	对于 runmqakm ，这是创建自签名证书期间使用的散列算法。 此散列算法用于创建与新创建的自签名证书相关联的签名。 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。
-sig_alg	对于 iKeycmd，这是用于创建条目的密钥对的非对称签名算法。 值可以是 MD2_WITH_RSA， MD2WithRSA， MD5_WITH_RSA ， MD5WithRSA, SHA1WithDSA， SHA1WithRSA， SHA256_WITH_RSA, SHA256WithRSA ， SHA2WithRSA， SHA384_WITH_RSA, SHA384WithRSA， SHA512_WITH_RSA， SHA512WithRSA, SHA_WITH_DSA， SHA_WITH_RSA ， SHAWithDSA 或 SHAWithRSA。 缺省值为 SHA1WithRSA。
-san_dnsname *DNS_names*	要创建的条目的 DNS 名称的逗号或空格分隔列表。
-san_emailaddr *email_addresses*	要创建的条目的电子邮件地址的逗号或空格分隔列表。
-san_ipaddr *IP_addresses*	要创建的条目的 IP 地址的逗号或空格分隔列表。

distributed 在 *UNIX, Linux, and Windows* 系统上请求个人证书

您可以使用 **strmqikm** (iKeyman) 来请求个人证书 GUI ，或者从命令行使用 **runmqckm** 或 **runmqakm** 命令。 如果需要以符合 FIPS 的方式管理 SSL 或 TLS 证书，请使用 **runmqakm** 命令。

关于此任务

您可以使用 iKeyman GUI 或从命令行请求个人证书，但需遵循以下注意事项:

- WebSphere 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。
- 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。
- 并非所有数字证书都可以与所有 CipherSpecs 配合使用。 确保您请求与需要使用的 CipherSpecs 兼容的证书。 WebSphere MQ 支持三种不同类型的 CipherSpec。 有关详细信息，请参阅 第 28 页的『IBM

WebSphere MQ 中的数字证书和 CipherSpec 兼容性』 主题中的 第 30 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) ，必须使用 **runmqakm** 命令来请求证书，并且必须指定椭圆曲线 ECDSA 签名算法参数; 例如 **-sig_alg** EC_ecdsa_with_SHA384。
- 仅 runmqakm 命令提供符合 FIPS 的选项。
- 如果您正在使用加密硬件，请参阅 第 117 页的『请求 PKCS #11 硬件的个人证书』。

使用 *iKeyman* 用户界面

关于此任务

iKeyman 未提供符合 FIPS 的选项。 如果需要以符合 FIPS 的方式管理 SSL 或 TLS 证书，请使用 **runmqakm** 命令。

过程

通过使用 iKeyman 用户界面，完成以下步骤以应用个人证书:

1. 使用 **strmqikm** 命令启动 iKeyman 用户界面。
2. 从 **密钥数据库文件** 菜单中，单击 **打开**。
   " **打开** " 窗口将打开。
3. 单击**密钥数据库类型**，并选择 **CMS**（证书管理系统）。
4. 单击**浏览**以浏览至包含密钥数据库文件的目录。
5. 选择要从中生成请求的密钥数据库文件; 例如， key.kdb。
6. 单击**打开**。
   此时将打开 " **密码提示** " 窗口。
7. 输入在创建密钥数据库时设置的密码，然后单击**确定**。
   密钥数据库文件的名称显示在 **文件名** 字段中。
8. 从 **创建** 菜单中，单击 **新建证书请求**。 此时将打开 " **新建密钥和证书请求** " 窗口。
9. 在 **密钥标签** 字段中，输入以下标签:
   - 对于队列管理器，请输入 ibmwebspheremq ，后跟已更改为小写的队列管理器的名称。 例如，对于名为 QM1 的队列管理器，请输入 ibmwebspheremqqm1。
   - 对于 IBM WebSphere MQ MQI client，请输入 ibmwebspheremq ，后跟您的登录用户标识 (全部为小写); 例如， ibmwebspheremqmyuserid 。
10. 在 **专有名称** 字段或任何 **主题备用名称** 字段中输入或选择任何字段的值。 对于剩余的字段，可以接受缺省值，也可以输入或选择新值。
    有关专有名称的更多信息，请参阅 第 10 页的『专有名称』。
11. 在 **输入要在其中存储证书请求的文件的名称** 字段中，接受缺省值 certreq.arm，或者输入具有完整路径的新值。
12. 单击**确定**。
    这将显示确认窗口。
13. 单击**确定**。
    个人证书请求 列表显示您创建的新个人证书请求的标签。 证书请求存储在您在步骤 第 104 页的『11』 中选择的文件中。
14. 通过将文件发送到认证中心 (CA) 或通过将文件复制到 CA 的 Web 站点上的请求表单来请求新的个人证书。

使用命令行

过程

使用以下命令通过使用 **runmqckm** 或 **runmqakm** 命令来请求个人证书:

- 使用 **runmqckm**:

```
runmqckm -certreq -create -db filename -pw
password -label label
        -dn distinguished_name -size key_size
 -file filename -sig_alg algorithm
```

  您可以使用 -san_dsname *DNS_names* ， -san_emailaddr *email_addresses* 或 -san_ipaddr *IP_addresses* 来代替 -dn *distinguished_name* 。

- 使用 runmqakm:

```
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 CMS 密钥数据库的标准文件名。

-pw *password*
指定 CMS 密钥数据库的密码。

-label *label*
指定附加到证书的密钥标签。

-dn 区分名称
指定用双引号括起的 X.500 专有名称。 至少需要一个属性。 您可以提供多个 OU 和 DC 属性。

-size 键大小
指定密钥大小。 如果您正在使用 **runmqckm**，那么该值可以是 512 或 1024。 如果您正在使用 **runmqakm**，那么该值可以是 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。 此方式禁用 BSafe 密码库。 仅使用 ICC 组件，并且必须以 FIPS 方式成功初始化此组件。 在 FIPS 方式下， 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
对于 **runmqckm**，指定用于创建条目的密钥对的非对称签名算法。 该值可以是 MD2_WITH_RSA， MD2WithRSA， MD5_WITH_RSA， MD5WithRSA， SHA1WithDSA， SHA1WithRSA， SHA256_WITH_RSA， SHA256WithRSA， SHA2WithRSA， SHA384_WITH_RSA， SHA384WithRSA， SHA512_WITH_RSA， SHA512WithRSA， SHA_WITH_DSA， SHA_WITH_RSA， SHAWithDSA 或 SHAWithRSA 。 缺省值为 SHA1WithRSA

-sig_alg
对于 **runmqakm**，指定在创建证书请求期间使用的散列算法。 此散列算法用于创建与新创建的证书请求相关联的签名。 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。

-san_dnsname *DNS_names*
指定要创建的条目的 DNS 名称的逗号分隔列表或空格分隔列表。

-san_emailaddr *email_地址*
指定要创建的条目的电子邮件地址的逗号分隔列表或空格分隔列表。

-san_ipaddr *IP_address*
指定要创建的条目的 IP 地址的逗号分隔列表或空格分隔列表。

在 *UNIX, Linux, and Windows* 系统上更新现有个人证书

您可以使用 iKeyman 用户界面或使用 **iKeycmd** 或 **runmqakm** 命令来更新个人证书。

开始之前

如果您需要对个人证书使用更大的密钥大小，那么以下描述的更新步骤不起作用，因为重新创建的证书请求是从现有密钥生成的。

遵循 第 103 页的『在 UNIX, Linux, and Windows 系统上请求个人证书』 中描述的步骤，使用所需的密钥大小来创建新的证书请求。 此过程将替换现有密钥。

关于此任务

个人证书具有到期日期，之后无法再使用该证书。 此任务说明如何在现有个人证书到期之前对其进行更新。

使用 *iKeyman* 用户界面

关于此任务

iKeyman 未提供符合 FIPS 的选项。 如果需要以符合 FIPS 的方式管理 SSL 或 TLS 证书，请使用 **runmqakm** 命令。

过程

通过使用 iKeyman 用户界面，完成以下步骤以应用个人证书:

1. 在 UNIX, Linux, and Windows 系统上使用 **strmqikm** 命令启动 iKeyman 用户界面。
2. 从 **密钥数据库文件** 菜单中，单击 **打开**。
   " **打开** " 窗口将打开。
3. 单击**密钥数据库类型**，并选择 **CMS**（证书管理系统）。
4. 单击**浏览**以浏览至包含密钥数据库文件的目录。
5. 选择要从中生成请求的密钥数据库文件; 例如， `key.kdb`。
6. 单击**打开**。
   此时将打开 " **密码提示** " 窗口。
7. 输入在创建密钥数据库时设置的密码，然后单击**确定**。
   密钥数据库文件的名称显示在 **文件名** 字段中。
8. 从下拉选择菜单中选择 **个人证书** ，然后从要更新的列表中选择证书。
9. 单击 **重新创建请求 ...** 按钮。
   将打开一个窗口，供您输入文件名和文件位置信息。
10. 在 **文件名**字段中，接受缺省值 `certreq.arm`，或者输入新值 (包括完整文件路径)。
11. 单击**确定**。 证书请求存储在您在步骤 第 106 页的『9』 中选择的文件中。
12. 通过将文件发送到认证中心 (CA) 或通过将文件复制到 CA 的 Web 站点上的请求表单来请求新的个人证书。

使用命令行

过程

使用以下命令通过使用 **iKeycmd** 或 **runmqakm** 命令来请求个人证书:

- 在 UNIX, Linux, and Windows 系统上使用 **iKeycmd** :

```
runmqckm -certreq -recreate -db filename -pw
password -label label
 -target filename
```

- 使用 runmqakm:

```
runmqakm -certreq -recreate -db filename -pw
password -label label
  -target filename
```

其中:

-db 文件名
指定 CMS 密钥数据库的标准文件名。

-pw *password*
指定 CMS 密钥数据库的密码。

-target 文件名
指定证书请求的文件名。

下一步做什么

从认证中心收到已签名的个人证书后，可以使用 第 107 页的『在 UNIX， Linux 和 Windows 系统上的密钥存储库中接收个人证书』中描述的步骤将其添加到密钥数据库。

在 *UNIX*， *Linux* 和 *Windows* 系统上的密钥存储库中接收个人证书

使用此过程将个人证书接收到密钥数据库文件中。 密钥存储库必须与创建证书请求的存储库相同。

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数据库文件。 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。

使用 iKeyman

如果需要以符合 FIPS 的方式管理 SSL 证书，请使用 runmqakm 命令。 iKeyman 未提供符合 FIPS 的选项。

确保要导入的证书文件对当前用户具有写许可权，然后对队列管理器或 WebSphere MQ MQI 客户机使用以下过程以将个人证书接收到密钥数据库文件中:

1. 使用 **strmqikm** 命令启动 iKeyman GUI (在 Windows UNIX and Linux 上)。
2. 从**密钥数据库文件**菜单中，单击**打开**。 这样会显示“打开”窗口。
3. 单击**密钥数据库类型**，并选择 **CMS**（证书管理系统）。
4. 单击**浏览**以浏览至包含密钥数据库文件的目录。
5. 选择要向其添加该证书的密钥数据库文件，例如，key.kdb。
6. 单击 **打开**，然后单击 **确定**。 这样会打开“密码提示”窗口。
7. 输入在创建密钥数据库时设置的密码，然后单击**确定**。 密钥数据库文件的名称将显示在 **文件名** 字段中。 选择 **个人证书** 视图。
8. 单击**接收**。 "从文件接收证书" 窗口将打开。
9. 输入新个人证书的证书文件名和位置，或者单击 **浏览** 以选择名称和位置。
10. 单击**确定**。 如果您的密钥数据库中已有个人证书，那么将打开一个窗口，询问您是否要将要添加的密钥设置为数据库中的缺省密钥。
11. 单击 **是** 或 **否**。 这样会打开“输入标签”窗口。
12. 单击**确定**。 **个人证书** 字段显示您添加的新个人证书的标签。

使用命令行

使用以下命令通过 iKeycmd 将个人证书添加到密钥数据库文件:

- 在 UNIX， Linux 和 Windows 上，发出以下命令:

```
runmqckm -cert -receive -file filename -db filename -pw
password
        -format ascii
```

其中:

-file *filename*	是包含个人证书的文件的标准文件名。
-db *filename*	是 CMS 密钥数据库的标准文件名。
-pw *password*	是 CMS 密钥数据库的密码。
-format *ascii*	是证书的格式。 对于 Base64-encoded ASCII， 该值可以是 ascii， 对于二进制 DER 数据，该值可以是 binary。 缺省值为 ascii。

如果您正在使用加密硬件，请参阅 第 119 页的『将个人证书导入到 PKCS #11 硬件』。

从密钥存储库中抽取 *CA* 证书

遵循此过程以抽取 CA 证书。

使用 iKeyman

如果需要以符合 FIPS 的方式管理 SSL 证书，请使用 runmqakm 命令。 iKeyman 未提供符合 FIPS 的选项。

在要从中抽取 CA 证书的机器上执行以下步骤:

1. 使用 **strmqikm** 命令启动 iKeyman GUI。
2. 从**密钥数据库文件**菜单中，单击**打开**。 这样会显示“打开”窗口。
3. 单击**密钥数据库类型**，并选择 **CMS**（证书管理系统）。
4. 单击**浏览**以浏览至包含密钥数据库文件的目录。
5. 选择要从中抽取的密钥数据库文件，例如 key.kdb。
6. 单击**打开**。 这样会打开“密码提示”窗口。
7. 输入在创建密钥数据库时设置的密码，然后单击**确定**。 密钥数据库文件的名称将显示在 **文件名** 字段中。
8. 在 **密钥数据库内容** 字段中，选择 **签署者证书**，然后选择要抽取的证书。
9. 单击 **抽取**。 此时将打开 "将证书抽取到文件" 窗口。
10. 选择证书的 **数据类型**，例如，对于扩展名为 .arm 的文件，选择 **Base64-encoded ASCII 数据**。
11. 输入要用于存储证书的证书文件名和位置，或者单击 **浏览** 以选择名称和位置。
12. 单击**确定**。 证书将写入您指定的文件。

使用命令行

使用以下命令可使用 iKeycmd 来抽取 CA 证书:

- 在 UNIX， Linux 和 Windows 上:

```
runmqckm -cert -extract -db filename -pw password -label label -target filename
         -format ascii
```

其中:

-db *filename*	是 CMS 密钥数据库的标准路径名。

-pw *password*	是 CMS 密钥数据库的密码。
-label *label*	是证书附加的标签。
-target *filename*	是目标文件的名称。
-format *ascii*	是证书的格式。 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。

从 *UNIX*，*Linux* 和 *Windows* 系统上的密钥存储库中抽取自签名证书的公用部分

遵循此过程以抽取自签名证书的公共部分。

使用 iKeyman

如果需要以符合 FIPS 的方式管理 SSL 证书，请使用 runmqakm 命令。 iKeyman 未提供符合 FIPS 的选项。

在要从中抽取自签名证书的公共部分的机器上执行以下步骤:

1. 使用 **strmqikm** 命令启动 iKeyman GUI (在 UNIX， Linux 和 Windows 上)。
2. 从**密钥数据库文件**菜单中，单击**打开**。 这样会显示“打开”窗口。
3. 单击**密钥数据库类型**，并选择 **CMS**（证书管理系统）。
4. 单击**浏览**以浏览至包含密钥数据库文件的目录。
5. 选择要从中抽取证书的密钥数据库文件，例如 key.kdb。
6. 单击**打开**。 这样会打开“密码提示”窗口。
7. 输入在创建密钥数据库时设置的密码，然后单击**确定**。 密钥数据库文件的名称将显示在 **文件名** 字段中。
8. 在 **密钥数据库内容** 字段中，选择 **个人证书** ，然后选择证书。
9. 单击**提取证书**。 此时将打开 "将证书抽取到文件" 窗口。
10. 选择证书的 **数据类型** ，例如，对于扩展名为 .arm 的文件，选择 **Base64-encoded ASCII 数据** 。
11. 输入要用于存储证书的证书文件名和位置，或者单击 **浏览** 以选择名称和位置。
12. 单击**确定**。 证书将写入您指定的文件。 请注意，当您抽取 (而不是导出) 证书时，仅包含证书的公用部分，因此不需要密码。

使用命令行

使用以下命令可使用 iKeycmd 或 runmqakm 来抽取自签名证书的公共部分:

- 在 UNIX， Linux 和 Windows 上:

```
runmqckm -cert -extract -db filename -pw password -label label -target filename
         -format ascii
```

- 使用 runmqakm:

```
runmqakm -cert -extract -db filename -pw password -label label
            -target filename -format ascii -fips
```

其中:

-db *filename*	是 CMS 密钥数据库的标准路径名。
-pw *password*	是 CMS 密钥数据库的密码。
-label *label*	是证书附加的标签。
-target *filename*	是目标文件的名称。
-format *ascii*	是证书的格式。 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。

将 *CA* 证书 (或自签名证书的公用部分) 添加到 *UNIX, Linux, and Windows* 系统上的密钥存储库

请执行以下过程以将 CA 证书（或自签名证书的公共部分）添加到密钥存储库中。

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。

虽然以下指示信息引用的是 CA 证书，但是它们也适用于自签名证书的公共部分。

注: 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。 必须确保证书采用 ASCII (UTF-8) 或二进制 (DER) 编码，因为 IBM 全局安全工具箱 (GSKit) 不支持采用其他类型编码的证书。 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。

使用 iKeyman

如果需要以符合 FIPS 的方式管理 SSL 证书，请使用 runmqakm 命令。 iKeyman 未提供符合 FIPS 的选项。

在要添加 CA 证书的机器上执行以下步骤:

1. 使用 **strmqikm** 命令启动 iKeyman GUI (在 UNIX， Linux 和 Windows 系统上)。
2. 从**密钥数据库文件**菜单中，单击**打开**。 这样会显示“打开”窗口。
3. 单击**密钥数据库类型**，并选择 **CMS**（证书管理系统）。
4. 单击**浏览**以浏览至包含密钥数据库文件的目录。
5. 选择要向其添加该证书的密钥数据库文件，例如，key.kdb。
6. 单击**打开**。 这样会打开“密码提示”窗口。
7. 输入在创建密钥数据库时设置的密码，然后单击**确定**。 密钥数据库文件的名称显示在**文件名**字段中。
8. 在**密钥数据库内容**字段中，选择**签署者证书**。
9. 单击**添加**。 这样会打开“从文件添加 CA 证书”窗口。
10. 输入证书文件名和用于存储证书的位置，或单击**浏览**以选择名称和位置。
11. 单击**确定**。 这样会打开“输入标签”窗口。
12. 在“输入标签”窗口中，输入证书名称。
13. 单击**确定**。 这样会将该证书添加至密钥数据库中。

使用命令行

使用以下命令以通过 iKeycmd 添加 CA 证书:

- 在 UNIX， Linux 和 Windows 上，发出以下命令:

```
runmqckm -cert -add -db filename -pw password -label label -file filename
        -format ascii
```

其中:

-db *filename*	是 CMS 密钥数据库的标准路径名。
-pw *password*	是 CMS 密钥数据库的密码。
-label *label*	是证书附加的标签。
-file *filename*	是包含证书的文件的名称。
-format *ascii*	是证书的格式。 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。

从密钥存储库导出个人证书

遵循此过程以导出个人证书。

使用 iKeyman

如果需要以符合 FIPS 的方式管理 SSL 证书，请使用 runmqakm 命令。 iKeyman 未提供符合 FIPS 的选项。

在要从中导出个人证书的机器上执行以下步骤:

1. 使用 **strmqikm** 命令启动 iKeyman GUI (在 Windows UNIX and Linux 上)。
2. 从**密钥数据库文件**菜单中，单击**打开**。 这样会显示“打开”窗口。
3. 单击**密钥数据库类型**，并选择 **CMS**（证书管理系统）。
4. 单击**浏览**以浏览至包含密钥数据库文件的目录。
5. 选择要从中导出证书的密钥数据库文件，例如 key.kdb。
6. 单击**打开**。 这样会打开“密码提示”窗口。
7. 输入在创建密钥数据库时设置的密码，然后单击**确定**。 密钥数据库文件的名称将显示在 **文件名** 字段中。
8. 在 **密钥数据库内容** 字段中，选择 **个人证书** ，然后选择要导出的证书。
9. 单击 **导出/导入**。 此时将打开 "导出/导入密钥" 窗口。
10. 选择 **导出密钥**。
11. 选择要导出的证书的 **密钥文件类型** ，例如 **PKCS12**。
12. 输入要将证书导出到的文件名和位置，或者单击 **浏览** 以选择名称和位置。
13. 单击**确定**。 这样会打开“密码提示”窗口。 请注意，当您导出 (而不是抽取) 证书时，将同时包含该证书的公用部分和专用部分。 这就是导出的文件受密码保护的原因。 抽取证书时，仅包含证书的公共部分，因此不需要密码。
14. 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。
15. 单击**确定**。 证书将导出到您指定的文件。

使用命令行

使用以下命令通过 iKeycmd 导出个人证书:

- 在 UNIX， Linux 和 Windows 上:

```
runmqckm -cert -export -db filename -pw password -label label -type cms
        -target filename -target_pw password -target_type pkcs12
```

其中:

-db *filename*	是 CMS 密钥数据库的标准路径名。
-pw *password*	是 CMS 密钥数据库的密码。
-label *label*	是证书附加的标签。
-type *cms*	是数据库的类型。
-target *filename*	是目标文件的标准路径名。
-target_pw *password*	是用于加密证书的密码。
-target_type *pkcs12*	是证书的类型。

将个人证书导入到 *UNIX, Linux, and Windows* 系统上的密钥存储库中

遵循此过程以导入个人证书

在将 PKCS #12 格式的个人证书导入密钥数据库文件之前，必须首先将发放 CA 证书的完整有效链添加到密钥数据库文件中 (请参阅 第 110 页的『将 CA 证书 (或自签名证书的公用部分) 添加到 UNIX, Linux, and Windows 系统上的密钥存储库』)。

PKCS #12 文件应视为临时文件并在使用后删除。

使用 iKeyman

如果需要以符合 FIPS 的方式管理 SSL 证书，请使用 runmqakm 命令。 iKeyman 未提供符合 FIPS 的选项。

在要将个人证书导入到的机器上执行以下步骤:

1. 使用 **strmqikm** 命令启动 iKeyman GUI。
2. 从 **密钥数据库文件** 菜单中，单击 **打开**。 将显示 "打开" 窗口。
3. 单击**密钥数据库类型**，并选择 **CMS**（证书管理系统）。
4. 单击**浏览**以浏览至包含密钥数据库文件的目录。
5. 选择要向其添加该证书的密钥数据库文件，例如，`key.kdb`。
6. 单击**打开**。 将显示 "密码提示" 窗口。
7. 输入在创建密钥数据库时设置的密码，然后单击**确定**。 密钥数据库文件的名称显示在**文件名**字段中。
8. 在 **密钥数据库内容** 字段中，选择 **个人证书**。
9. 如果 "个人证书" 视图中有证书，请执行以下步骤:
   a. 单击 **导出/导入**。 这样会显示 "导出/导入密钥" 窗口。
   b. 选择 **导入密钥**。
10. 如果 "个人证书" 视图中没有证书，请单击 **导入**。
11. 选择要导入的证书的 **密钥文件类型** ，例如 PKCS12。
12. 输入证书文件名和用于存储证书的位置，或单击**浏览**以选择名称和位置。
13. 单击**确定**。 将显示 "密码提示" 窗口。
14. 在 **密码** 字段中，输入导出证书时使用的密码。
15. 单击**确定**。 此时将显示 "更改标签" 窗口。 例如，如果目标密钥数据库中已存在具有相同标签的证书，那么此窗口允许更改正在导入的证书的标签。 更改证书标签不会影响证书链验证。 这可用于将个人证书标签更改为 WebSphere MQ 所需的标签，以便将证书与特定队列管理器或客户机 (例如 `ibmwebspheremqqm1`) 相关联。
16. 要更改标签，请从 **选择要更改的标签** 列表中选择所需标签。 标签将复制到 **输入新标签** 输入字段中。 将标签文本替换为新标签的文本，然后单击 **应用**。
17. **输入新标签** 输入字段中的文本将复制回 **选择要更改的标签** 字段中，以替换最初选择的标签，从而重新标记相应的证书。
18. 更改所有需要更改的标签后，单击 **确定**。 "更改标签" 窗口将关闭，原始 IBM 密钥管理窗口将重新显示 **个人证书** 和使用正确标记的证书更新的 **签署者证书** 字段。
19. 证书将导入到目标密钥数据库。

使用命令行

要使用 iKeycmd 导入个人证书，请使用以下命令:

- 在 UNIX， Linux 和 Windows 上:

```
runmqckm -cert -import -file filename -pw password -type pkcs12 -target filename
-target_pw password -target_type cms -label label
```

其中:

`-file` *filename*	是包含 PKCS #12 证书的文件的标准文件名。
`-pw` *password*	是 PKCS #12 证书的密码。
`-type` *pkcs12*	是文件的类型。
`-target` *filename*	是目标 CMS 密钥数据库的名称。
`-target_pw` *password*	是 CMS 密钥数据库的密码。

`-target_type cms`	是由 `-target` 指定的数据库类型
`-label label`	是要从源密钥数据库导入的证书的标签。
`-new_label label`	是将在目标数据库中分配证书的标签。 如果省略 `-new_label` 选项，那么缺省值是使用与 `-label` 选项相同的选项。

iKeycmd 未提供用于直接更改证书标签的命令。 使用以下步骤来更改证书标签:

1. 使用 **-cert -export** 命令将证书导出到 PKCS #12 文件。 为 `-label` 选项指定现有证书标签。
2. 使用 **-cert -delete** 命令从原始密钥数据库中除去证书的现有副本。
3. 使用 **-cert -import** 命令从 PKCS #12 文件导入证书。 为 `-label` 选项指定旧标签，并为 `-new_label` 选项指定所需的新标签。 该证书将以所需标签重新导入到密钥数据库中。

从 *Microsoft .pfx* 文件导入

使用 iKeyman 将此过程从 Microsoft .pfx 文件中获取 mport。 不能使用 runmqakm 来导入 .pfx 文件。

.pfx 文件可以包含与同一密钥相关的两个证书。 一个是个人或站点证书 (包含公用密钥和专用密钥)。 另一个是 CA (签署者) 证书 (仅包含公用密钥)。 这些证书不能共存于同一个 CMS 密钥数据库文件中，因此只能导入其中一个证书。 此外， "友好名称" 或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。 此部分显示如何从 pfx 文件导入个人证书，同时使用先前分配给 CA (签署者) 证书的友好名称对其进行标记。 应该已将发放 CA (签署者) 证书添加到目标密钥数据库。 请注意， PKCS#12 文件应视为临时文件并在使用后删除。

执行以下步骤以从源 pfx 密钥数据库导入个人证书:

1. 使用 **strmqikm** 命令启动 iKeyman GUI (在 Linux， UNIX 或 Windows 上)。 此时将显示“IBM 密钥管理”窗口。
2. 从**密钥数据库文件**菜单中，单击**打开**。 此时将显示 "打开" 窗口。
3. 选择密钥数据库类型 **PKCS12**。
4. **建议您先备份 pfx 数据库，然后再执行此步骤**。 选择要导入的 pfx 密钥数据库。 单击**打开**。 此时将显示“密码提示”窗口。
5. 输入密钥数据库密码，然后单击 **确定**。 此时将显示“IBM 密钥管理”窗口。 标题栏显示所选 pfx 密钥数据库文件的名称，指示该文件已打开且就绪。
6. 从列表中选择 **签署者证书** 。 所需证书的 "友好名称" 在 "签署者证书" 面板中显示为标签。
7. 选择标签条目，然后单击 **删除** 以除去签署者证书。 此时将显示 "确认" 窗口。
8. 单击**是**。 所选标签不再显示在 "签署者证书" 面板中。
9. 对所有签署者证书重复步骤 6 ， 7 和 8。
10. 从**密钥数据库文件**菜单中，单击**打开**。 此时将显示 "打开" 窗口。
11. 选择要将 pfx 文件导入到的目标密钥 CMS 数据库。 单击**打开**。 此时将显示“密码提示”窗口。
12. 输入密钥数据库密码，然后单击 **确定**。 此时将显示“IBM 密钥管理”窗口。 标题栏显示所选密钥数据库文件的名称，指示该文件已打开并就绪。
13. 从列表中选择 **个人证书** 。
14. 如果 "个人证书" 视图中有证书，请执行以下步骤:
    a. 单击 **导出/导入密钥**。 这样会显示 "导出/导入密钥" 窗口。
    b. 从 "选择操作类型" 中选择 **导入** 。
15. 如果 "个人证书" 视图中没有证书，请单击 **导入**。
16. 选择 PKCS12 文件。
17. 输入步骤 4 中使用的 pfx 文件的名称。 单击**确定**。 此时将显示“密码提示”窗口。
18. 指定在删除签署者证书时指定的相同密码。 单击**确定**。
19. 将显示 "更改标签" 窗口 (因为应该只有单个证书可供导入)。 证书的标签应该是格式为 xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx 的 UUID。

20. 要更改标签，请从 **选择要更改的标签:** 面板中选择 UUID。 该标签将复制到 **输入新标签:** 字段中。 将标签文本替换为在步骤 7 中删除的友好名称文本，然后单击 **应用**。 友好名称的格式必须为 ibmwebspheremq，后跟队列管理器名称或小写的 WebSphere MQ MQI 客户机用户登录标识。
21. 单击**确定**。 现在已除去 "更改标签" 窗口，并且原始 IBM 密钥管理窗口将重新出现，并且 "个人证书" 和 "签署者证书" 面板将使用正确标记的个人证书进行更新。
22. pfx 个人证书现在已导入到 (目标) 数据库中。

无法使用 iKeycmd 来更改证书标签。

从 *PKCS #7* 文件导入

iKeyman 和 iKeycmd 工具不支持 PKCS #7 (.p7b) 文件。 使用 runmqckm 工具从 PKCS #7 文件导入证书。

使用以下命令从 PKCS #7 文件添加 CA 证书:

```
runmqckm -cert -add -db filename -pw password -type cms -file filename
-label label
```

-db *filename*	是 CMS 密钥数据库的标准文件名。
-pw *password*	是密钥数据库的密码。
-type *cms*	是密钥数据库的类型。
-file *filename*	是 PKCS #7 文件的名称。
-label *label*	是在目标数据库中分配证书的标签。 第一个证书采用给定的标签。 所有其他证书 (如果存在) 都使用其主题名称进行标记。

使用以下命令从 PKCS #7 文件导入个人证书:

```
runmqckm -cert -import -db filename -pw password -type pkcs7 -target filename
-target_pw password -target_type cms -label label -new_label label
```

-db *filename*	是包含 PKCS #7 证书的文件的标准文件名。
-pw *password*	是 PKCS #7 证书的密码。
-type *pkcs7*	是文件的类型。
-target *filename*	是目标密钥数据库的名称。
-target_pw *password*	是目标密钥数据库的密码。
-target_type *cms*	是由 -target 指定的数据库类型
-label *label*	是要导入的证书的标签。
-new_label *label*	是将在目标数据库中分配证书的标签。 如果省略 -new_label 选项，那么缺省值是使用与 -label 选项相同的选项。

从 *UNIX, Linux, and Windows* 系统上的密钥存储库中删除证书

使用此过程可除去个人证书或 CA 证书。

使用 iKeyman

如果需要以符合 FIPS 的方式管理 SSL 证书，请使用 runmqakm 命令。 iKeyman 未提供符合 FIPS 的选项。

1. 使用 **strmqikm** 命令启动 iKeyman GUI (在 UNIX， Linux 和 Windows 系统上)。
2. 从**密钥数据库文件**菜单中，单击**打开**。 这样会显示“打开”窗口。
3. 单击**密钥数据库类型**，并选择 **CMS**（证书管理系统）。
4. 单击**浏览**以浏览至包含密钥数据库文件的目录。
5. 选择要从中删除证书的密钥数据库文件，例如 key.kdb。

6. 单击**打开**。 这样会打开“密码提示”窗口。
7. 输入在创建密钥数据库时设置的密码，然后单击**确定**。 密钥数据库文件的名称将显示在 **文件名** 字段中。
8. 从下拉列表中，选择 **个人证书** 或 **签署者证书**
9. 选择要删除的证书。
10. 如果您还没有该证书的副本并且想要保存该证书，请单击 **导出/导入** 并将其导出 (请参阅 第 110 页的『从密钥存储库导出个人证书』)。
11. 选择证书后，单击 **删除**。 此时将打开 "确认" 窗口。
12. 单击**是**。 **个人证书** 字段不再显示您删除的证书的标签。

使用命令行

使用以下命令可使用 iKeycmd 或 runmqakm 删除证书:

- 在 UNIX， Linux 和 Windows 上:

```
runmqckm -cert -delete -db filename -pw password -label label
```

其中:

-db *filename*	是 CMS 密钥数据库的标准文件名。
-pw *password*	是 CMS 密钥数据库的密码。
-label *label*	是附加到个人证书的标签。
-fips	指定以 FIPS 方式运行命令。 此方式禁用 BSafe 密码库。 仅使用 ICC 组件，并且必须以 FIPS 方式成功初始化此组件。 当处于 FIPS 方式下时，ICC 组件使用已进行 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。

生成用于密钥存储库保护的强密码

您可以使用 **runmqakm** 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 14 -strong -fips
```

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 UNIX and Linux 系统上，如果以下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:

```
!   \   "    '
```

输入密码以响应来自 **runmqckm**， **runmqakm** 或 iKeyman GUI 的提示时，无需引用或转义密码。 由于操作系统 shell 在这些情况下不影响数据输入，因此不需要执行此操作。

在 *UNIX, Linux, and Windows* 系统上配置加密硬件

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 UNIX， Linux 或 Windows 系统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 WebSphere MQ Explorer 在 UNIX， Linux 或 Windows 系统上配置加密硬件。 有关更多信息，请参阅联机帮助。

您可以使用以下任一方法在 UNIX， Linux 或 Windows 系统上为 WebSphere 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。 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。

- 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。

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 116 页的『管理 PKCS #11 硬件上的证书』中对此进行了描述。

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。

关于此任务

您必须创建密钥数据库以准备 IBM WebSphere MQ 环境，即使您不打算在其中存储认证中心 (CA) 证书，但会将所有证书存储在加密硬件上。 队列管理器需要在其 SSLKEYR 字段中引用密钥数据库，或者客户机应用程序需要在 MQSSLKEYR 环境变量中引用密钥数据库。 如果要创建证书请求，那么此密钥数据库也是必需的。

您可以使用命令行或 **strmqikm** (iKeyman) 用户界面来创建密钥数据库。

过程

使用命令行创建密钥数据库。

1. 运行下列任一命令：
   - 在 UNIX, Linux, and Windows 系统上：

     ```
     runmqckm -keydb -create -db filename -pw password -type cms -stash
     ```

   - 使用 runmqakm：

     ```
     runmqakm -keydb -create -db filename -pw password -type cms
         -stash -fips -strong
     ```

   其中：

   -db 文件名
   指定 CMS 密钥数据库的标准文件名，并且必须具有文件扩展名 .kdb。

   -pw *password*
   指定 CMS 密钥数据库的密码。

   -type *cms*
   指定数据库的类型。 (对于 IBM WebSphere MQ，必须为 cms。)

   -stash
   将密钥数据库密码保存到文件。

   -无花果
   禁止使用 BSafe 加密库。 仅使用 ICC 组件，并且必须以 FIPS 方式成功初始化此组件。 在 FIPS 方式下， 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 0x20 - 0x7E 范围内的标准 ASCII 可打印字符集中。

或者，使用 **strmqikm** (iKeyman) 用户界面创建密钥数据库。

2. 在 UNIX and Linux 系统上，以 root 用户身份登录。 在 Windows 系统上，以管理员身份或 MQM 组的成员身份登录。
3. 通过运行 **strmqikm** 命令来启动 iKeyman 用户界面。
4. 单击**密钥数据库文件** > **打开**。
5. 单击 **密钥数据库类型** ，然后选择 **PKCS11Direct**。
6. 在 **文件名** 字段中，输入用于管理加密硬件的模块的名称; 例如， `PKCS11_API.so`。

   如果您要使用存储在 PKCS #11 加密硬件上的证书或密钥，请注意 iKeycmd 和 iKeyman 是 64 位程序。 PKCS #11 支持所需要的外部模块将装入 64 位进程中，因此您必须安装 64 位 PKCS #11 库以管理加密硬件。 只有 Windows 和 Linux x86 32 位平台例外，因为 iKeyman 和 iKeycmd 程序在这些平台上是 32 位的。
7. 在 **位置** 字段中，输入路径:
   - 例如，在 UNIX and Linux 系统上，这可能是 `/usr/lib/pksc11`。
   - 在 Windows 系统上，可以输入库名; 例如， `cryptoki`。

   单击**确定**。 "打开密码令牌" 窗口将打开。
8. 在 **加密令牌密码** 字段中，输入在配置加密硬件时设置的密码。
9. 如果您的加密硬件有能力保存接收或导入个人证书所需的签署者证书，请清除两个辅助密钥数据库复选框，然后继续执行步骤 第 117 页的『13』。

   如果需要辅助 CMS 密钥数据库来保存签署者证书，请选择 **打开现有辅助密钥数据库文件** 或 **创建新的辅助密钥数据库文件**。
10. 在 **文件名** 字段中，输入文件名。 此字段已包含文本 `key.kdb`。 如果您的主干名称为 `key`，请保持此字段不变。 如果指定了其他主干名称，请将 `key` 替换为您的主干名称。 不得更改 `.kdb` 后缀。
11. 在 **位置** 字段中，输入路径，例如:
    - 对于队列管理器: `/var/mqm/qmgrs/QM1/ssl`
    - 对于 IBM WebSphere MQ MQI 客户机: `/var/mqm/ssl`

    单击**确定**。 这样会打开“密码提示”窗口。
12. 请输入密码。

    如果在步骤 第 117 页的『9』中选择了 **打开现有辅助密钥数据库文件** ，请在 **密码** 字段中输入密码。

    如果在步骤 第 117 页的『9』中选择了 **创建新的辅助密钥数据库文件** ，请完成以下子步骤:

    a) 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。

    b) 选择 **将密码隐藏到文件中**。 请注意，如果不隐藏密码，那么尝试启动 SSL 通道将失败，因为这些通道无法获取访问密钥数据库文件所需的密码。

    c) 单击**确定**。 此时将打开一个窗口，确认密码在文件 `key.sth` 中 (除非您指定了另一个词干名称)。
13. 单击**确定**。 将显示 "密钥数据库内容" 框架。

请求 *PKCS #11* 硬件的个人证书
对于队列管理器或 IBM WebSphere MQ MQI 客户机，请使用此过程来请求加密硬件的个人证书。

使用 *iKeyman* 用户界面

关于此任务

注: WebSphere 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。

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。

过程

要从 iKeyman 用户界面请求个人证书，请完成以下步骤:

1. 完成用于处理加密硬件的步骤。 请参阅第 116 页的『管理 PKCS #11 硬件上的证书』。
2. 从 **创建** 菜单中，单击 **新建证书请求**。
   此时将打开 "新建密钥和证书请求" 窗口。
3. 在 **密钥标签** 字段中，输入以下标签:
   - 对于队列管理器，请输入 `ibmwebspheremq`，后跟已更改为小写的队列管理器的名称。 例如，对于名为 `QM1` 的队列管理器，请输入 `ibmwebspheremqqm1`。
   - 对于 IBM WebSphere MQ MQI client，输入 `ibmwebspheremq`，后跟您的登录用户标识 (全部为小写); 例如， `ibmwebspheremqmyuserid`。
4. 输入 **公共名称** 和 **组织**的值，然后选择 **国家或地区**。 对于其余可选字段，请接受缺省值，或者输入或选择新值。
   请注意，只能在 **组织单元** 字段中提供一个名称。 有关这些字段的更多信息，请参阅第 10 页的『专有名称』。
5. 在 **输入要在其中存储证书请求的文件的名称** 字段中，接受缺省值 `certreq.arm`，或者输入具有完整路径的新值。
6. 单击**确定**。
   确认窗口将打开。
7. 单击**确定**。
   个人证书请求 列表显示您创建的新个人证书请求的标签。 证书请求存储在您在步骤第 118 页的『5』中选择的文件中。
8. 通过将文件发送到认证中心 (CA) 或通过将文件复制到 CA 的 Web 站点上的请求表单来请求新的个人证书。

使用命令行

过程

使用以下命令通过使用 **runmqckm** 或 **runmqakm** 命令来请求个人证书:

- 使用 **runmqckm**:

```
runmqckm -certreq -create -db filename -pw
password -label label
        -dn distinguished_name -size key_size
 -file filename -sig_alg algorithm
```

  您可以使用 `-san_dsname DNS_names`， `-san_emailaddr email_addresses` 或 `-san_ipaddr IP_addresses` 来代替 `-dn distinguished_name`。

- 使用 runmqakm:

```
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 CMS 密钥数据库的标准文件名。

-pw *password*
指定 CMS 密钥数据库的密码。

-label *label*
指定附加到证书的密钥标签。

-dn 区分名称
指定用双引号括起的 X.500 专有名称。 至少需要一个属性。 您可以提供多个 OU 和 DC 属性。

-size 键大小
指定密钥大小。 如果您正在使用 **runmqckm**，那么该值可以是 512 或 1024。 如果您正在使用 **runmqakm**，那么该值可以是 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。 此方式禁用 BSafe 密码库。 仅使用 ICC 组件，并且必须以 FIPS 方式成功初始化此组件。 在 FIPS 方式下， 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
对于 **runmqckm**，指定用于创建条目的密钥对的非对称签名算法。 该值可以是 MD2_WITH_RSA， MD2WithRSA， MD5_WITH_RSA， MD5WithRSA， SHA1WithDSA， SHA1WithRSA， SHA256_WITH_RSA， SHA256WithRSA， SHA2WithRSA， SHA384_WITH_RSA， SHA384WithRSA， SHA512_WITH_RSA， SHA512WithRSA， SHA_WITH_DSA， SHA_WITH_RSA， SHAWithDSA 或 SHAWithRSA 。 缺省值为 SHA1WithRSA

-sig_alg
对于 **runmqakm**，指定在创建证书请求期间使用的散列算法。 此散列算法用于创建与新创建的证书请求相关联的签名。 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。

-san_dnsname *DNS_names*
指定要创建的条目的 DNS 名称的逗号分隔列表或空格分隔列表。

-san_emailaddr *email_*地址
指定要创建的条目的电子邮件地址的逗号分隔列表或空格分隔列表。

-san_ipaddr *IP_address*
指定要创建的条目的 IP 地址的逗号分隔列表或空格分隔列表。

将个人证书导入到 *PKCS #11* 硬件
将此过程用于队列管理器或 IBM WebSphere MQ MQI 客户机，以将个人证书导入到加密硬件。

使用 *iKeyman*

过程

要从 iKeyman 用户界面请求个人证书，请完成以下步骤:

1. 完成用于处理加密硬件的步骤。 请参阅第 116 页的『管理 PKCS #11 硬件上的证书』。
2. 单击 **接收**。 "从文件接收证书" 窗口将打开。
3. 选择新个人证书的 **数据类型** ; 例如，对于扩展名为 .arm 的文件，选择 Base64-encoded ASCII data 。
4. 输入新个人证书的证书文件名和位置，或者单击 **浏览** 以选择名称和位置。
5. 单击**确定**。 如果您的密钥数据库中已有个人证书，那么将打开一个窗口，询问您是否要将要添加的密钥设置为数据库中的缺省密钥。
6. 单击**是**或**否**。 这样会打开“输入标签”窗口。
7. 输入标签。
   例如，您可以使用与请求个人证书时相同的标签。 请注意，标签必须采用正确的 IBM WebSphere MQ 格式:

- 对于队列管理器， ibmwebspheremq 后跟小写的队列管理器名称。 例如，对于名为 QM1 的队列管理器，标签将为: ibmwebspheremqqm1。
- 对于 IBM WebSphere MQ MQI 客户机， ibmwebspheremq 后跟小写的登录用户标识。 例如，对于用户标识 MyUserID，标签将为: ibmwebspheremqmyuserid。

8. 单击**确定**。 **个人证书** 列表显示您添加的新个人证书的标签。 此标签通过在您提供的标签之前添加密码令牌标签来构成。

使用命令行

过程

要从命令行请求个人证书，请完成以下步骤:

1. 打开针对您的环境所配置的命令窗口。
2. 输入适用于您的操作系统和配置的相应命令:

- 在 Windows UNIX and Linux 系统上，使用下列其中一个命令:

```
runmqckm -cert -receive -file filename -crypto path
-tokenlabel hardware_token -pw hardware_password -format cert_format
```

```
runmqakm -cert -receive -file filename -crypto path
-tokenlabel hardware_token -pw hardware_password -format cert_format -fips
```

其中:

-file 文件名
指定包含个人证书的文件的标准文件名。

-crypto *path*
指定硬件随附的 PKCS #11 库的标准路径。

-tokenlabel 硬件令牌
指定在安装期间提供给加密硬件的存储部分的标签。

-pw *hardware_password*
指定用于访问硬件的密码。

-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
指定该命令以 FIPS 方式运行。此方式禁止使用 BSafe 加密库。 仅使用 ICC 组件，并且必须以 FIPS 方式成功初始化此组件。 在 FIPS 方式下， 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。

识别和认证用户

您可以使用 MQCSP 结构或多种类型的用户出口程序来识别和认证用户。

使用 MQCSP 结构

在 MQCONNX 调用上指定 MQCSP 连接安全性参数结构; 此结构包含用户标识和密码。 如果需要，可以在安全出口中变更 MQCSP。

注: 对象权限管理器 (OAM) 不使用密码。 但是， OAM 对用户标识执行一些有限的工作，这可能被视为一种微不足道的认证形式。 如果在应用程序中使用这些参数，那么这些检查会阻止您采用另一个用户标识。

在安全出口中实现标识和认证

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 WebSphere

MQ 客户机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 WebSphere MQ MQI 客户机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 86 页的『SPI 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。

安全套接字层 (SSL) 和传输层安全性 (TLS) 使用 PKI 技术，如刚才所述。 有关 SSL 和 TLS 如何执行认证的更多信息，请参阅 第 13 页的『安全套接字层 (SSL) 和传输层安全性 (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 188 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 60 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。

在消息出口中实现标识和认证

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 123 页的『API 出口和 API 交叉出口中的身份映射』。

在 API 出口和 API 交叉出口中实现标识和认证

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。

特权用户

特权用户是对 WebSphere MQ 具有完全管理权限的用户。

除了下表中列出的用户外，对队列具有 `+crt` 权限的任何组的成员都是间接管理员。 同样，在队列管理器上具有 `+set` 权限的任何用户以及在命令队列上具有 `+put` 权限的任何用户都是管理员。

您不应将这些特权授予普通用户和应用程序。

表 *13:* 特权用户 *(按平台排列).*

特权用户的表。 在 Windows 上， SYSTEM ， mqm 组的所有成员以及 Administrators 组的所有成员都是特权用户。 在 UNIX and Linux 系统上， mqm 组的所有成员都是特权用户。 在 IBM i 上，概要文件 (用户) qmqm 和 qmqmadm ， qmqmadm 组的所有成员以及使用 *ALLOBJ 设置定义的任何用户都是特权用户。

平台	特权用户
Windows 系统	• SYSTEM • mqm 组的成员 • Administrators 组的成员
UNIX and Linux 系统	• mqm 组的成员

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 连接安全性参数结构包含用户标识和密码，授权服务可以使用此用户标识和密码来标识和认证用户。

随 IBM WebSphere MQ 提供的授权服务组件称为对象权限管理器 (OAM)。 OAM 根据 MQCSP 中包含的标识授权用户，但不验证密码。 可以通过将链式出口与 OAM 配合使用，或者通过将 OAM 替换为备用授权服务，在授权服务中实现密码验证。

您可以在安全出口中变更 MQCSP。

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 WebSphere MQ MQI 客户机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 WebSphere MQ MQI 客户机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 86 页的『SPI 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。

安全套接字层 (SSL) 和传输层安全性 (TLS) 使用 PKI 技术，如刚才所述。 有关安全套接字层如何执行认证的更多信息，请参阅 第 13 页的『安全套接字层 (SSL) 和传输层安全性 (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 188 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 60 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 123 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。

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 SSL 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。

在 UNIX， Linux 和 Windows 系统上， WebSphere 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 WebSphere MQ classes for Java 和 IBM WebSphere MQ classes for JMS 无法在客户机通道定义表文件中使用 OCSP 信息。 但是，您可以按照使用联机证书协议部分中所述来配置 OCSP。

在 z/O 和 IBM i WebSphere MQ SSL 支持上，仅使用 LDAP 服务器上的 CRL 和 ARL 来检查是否存在已撤销的证书。

有关证书的更多信息

权限，请参阅 第 9 页的『数字证书』。

已撤销证书和 OCSP

IBM WebSphere 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 Windows UNIX and Linux 系统上的 WebSphere MQ 。

要使用 OCSP 检查数字证书的撤销状态，WebSphere MQ 可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 或 SSL 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：

正常
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 WebSphere MQ 不能访问 OCSP 响应程序。
- OCSP 响应程序已发送响应，但 WebSphere 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 WebSphere 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 UNIX and Linux 系统或 Windows 注册表的 `qm.ini` 文件的 SSL 节中。 可以使用 IBM WebSphere MQ Explorer 进行设置。 对于客户机，该属性保留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 WebSphere 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 WebSphere MQ 将允许启动 SSL 通道并且不会生成警告或 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 WebSphere 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，您无需设置任何其他证书；建立 SSL 连接时所执行的步骤足以验证 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 响应与对其进行签名的证书一起发送（并且该证书是由已对 SSL 连接可信的 CA 签署），因此无需任何其他证书设置。
- 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。 您必须向执行 OCSP 检查的客户机或队列管理器的密钥数据库中添加 OCSP 响应程序证书的副本；请参阅第 110 页的『将 CA 证书 (或自签名证书的公用部分) 添加到 UNIX, Linux, and Windows 系统上的密钥存储库』。 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。 如果未添加该证书，那么 WebSphere MQ 将无法验证 OCSP 响应上的数字签名，并且 OCSP 检查将产生“未知”结果，这可能会导致 IBM WebSphere MQ 关闭通道（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 存在限制，仅当针对整个 Java 虚拟机 (JVM) 进程启用 OCSP 时，WebSphere MQ 才可针对 SSL 和 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。
- 根据现行的任何 Java 安全管理器策略，使用 java.security.Security.setProperty() API。

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

属性名	描述
	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，在与使用 SSL 或 TLS 安全套接字的其他应用程序代码共享 JVM 时，该配置可能会带来意外的副作用。 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。 在应用 Java 临时修订和产品维护时应谨慎操作，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 检查中所述的 AMS Java 拦截器，请注意应避免使用与密钥库配置文件中的 AMS OCSP 配置相冲突的 java.security OCSP 配置。

使用证书撤销列表和权限撤销列表

WebSphere 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：

- 在 z/OS 上，系统 SSL 支持 Tivoli Public Key Infrastructure 产品在 LDAP 服务器中存储的 CRL 和 ARL。
- 在其他平台上， CRL 和 ARL 支持符合 PKIX X.509 V2 CRL 概要文件建议。

WebSphere 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 WebSphere MQ MQI 客户机接收到证书时，它会检查 CRL 以确认该证书仍然有效。 WebSphere 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 WebSphere MQ 将按照 LDAP CRL 服务器位置在 *SSLCRLNamelist* 属性指定的认证信息对象的名称列表中出现的顺序来查询这些位置，直到 WebSphere 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。

有关 LDAP 的更多信息，请参阅 将轻量级目录访问协议服务用于 WebSphere MQ for Windows。

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 WebSphere MQ SSL 配合使用，这些属性的二进制数据必须符合 DER (确定编码规则) 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 128 页的图 12 显示了一个样本 LDIF 文件，您可以将该文件创建为 LDAP 服务器的输入，以装入 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 *12:* 认证中心的样本 *LDIF* 文件。 这可能因实施而异。

第 128 页的图 13 显示了当您装入 第 128 页的图 12 中显示的样本 LDIF 文件以及 CA2 的类似文件时，LDAP 服务器创建的 DIT 结构，此文件是由 PKI 组织 (也在 IBM 中) 设置的假想认证中心。

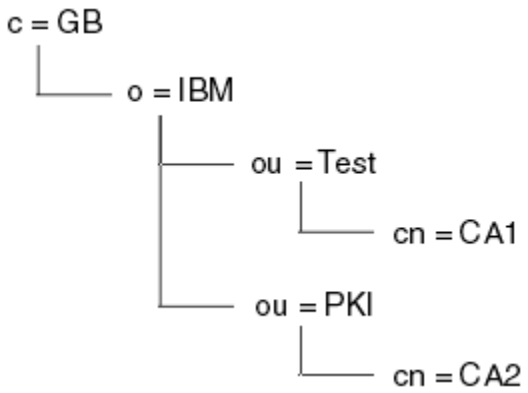


图 *13: LDAP* 目录信息树结构的示例

WebSphere 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。 WebSphere 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. 从您在步骤 第 128 页的『2』中创建的一个或多个 LDIF 文件添加条目。

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。

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 *SSLCRLNamelist* 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。

   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。

   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。
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 *SSLCRLNamelist* 的队列管理器属性。 此属性的队列管理器初始值为空。

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。

使用 *IBM WebSphere MQ Explorer* 访问 *CRL* 和 *ARL*
您可以使用 IBM WebSphere MQ Explorer 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. 将 CRL (LDAP) 对象的名称 (从步骤 第 129 页的『2.a』开始) 添加到列表中。
   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. 在 **CRL 名称列表** 字段中输入名称列表的名称 (紧接步骤 第 129 页的『3.a』)。

使用 *IBM WebSphere MQ MQI* 客户机访问 *CRL* 和 *ARL*

您有三个选项用于指定 LDAP 服务器，这些服务器保存要由 IBM WebSphere MQ MQI 客户机检查的 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)。

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 WebSphere MQ MQI 客户机通道访问 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 WebSphere MQ MQI 客户机系统上，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，按优先顺序在此处列出。

当 WebSphere MQ MQI 客户机应用程序发出 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 WebSphere MQ MQI 客户机访问 OCSP 响应程序或包含 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

因此， WebSphere MQ MQI 客户机可以访问包含 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。

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) 以供客户机使用的那些平台上指定该对象。

要使 WebSphere MQ MQI 客户机能够访问包含 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:

在服务器平台上 AIX， HP-UX， Linux， Solaris 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 **SSLCRLNameList** 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 WebSphere MQ MQI 客户机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 **SSLCRLNameList** 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 WebSphere MQ MQI 客户机可以使用 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 **SSLCRLNameList** 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。

在 Windows 上使用 Active Directory

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 WebSphere MQ classes for Java* 和 *IBM WebSphere MQ classes for JMS* 访问 *CRL* 和 *ARL*

用于 Java 的 IBM WebSphere MQ 类和用于 JMS 访问 CRL 的 IBM WebSphere MQ 类与其他平台不同。

有关使用 IBM WebSphere 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 WebSphere 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 WebSphere 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 脚本 (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 WebSphere MQ Explorer。

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。

在 UNIX, 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。 它还包含可用于确定要执行哪些任务以将安全性应用于系统的核对表，以及授予用户管理 IBM WebSphere MQ 和使用 IBM WebSphere 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。

在 UNIX， Linux 和 Windows 系统上使用 OAM 控制对对象的访问

对象权限管理器 (OAM) 提供了一个命令接口，用于授予和撤销对 WebSphere MQ 对象的权限。

您必须获得适当的授权才能使用这些命令，如 第 165 页的『在 UNIX, Linux, and Windows 系统上管理 IBM WebSphere MQ 的权限』中所述。 有权管理 WebSphere MQ 的用户标识具有对队列管理器的 超级用户 权限，这意味着您不必授予他们发出任何 MQI 请求或命令的进一步许可权。

授予对 UNIX, Linux, and Windows 系统上的 IBM WebSphere MQ 对象的访问权

使用 **setmqaut** 控制命令或 **MQCMD_SET_AUTH_REC** PCF 命令来授予用户和用户组对 IBM WebSphere MQ 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 **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 133 页的『在 UNIX, Linux, and Windows 系统上使用 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，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
```

在本示例中:

- 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 UNIX 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
```

将该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。

在 UNIX, Linux, and Windows 系统上使用 OAM 通用概要文件

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 **setmqaut** 命令。

在 **setmqaut** 命令中使用通用概要文件使您能够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，限定符为 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 UNIX 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，在上面的示例中，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. **

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 **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 UNIX and Linux 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 WebSphere 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
```

在 *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 UNIX 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

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，在上面的示例中，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. **

转储概要文件设置

使用 **dmpmqaut** 控制命令或 **MQCMD_INQUIRE_AUTH_RECS** PCF 命令转储与指定概要文件关联的当前权限。

有关 **dmpmqaut** 控制命令及其语法的完整定义，请参阅 dmpmqaut，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: UNIX 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 WebSphere 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
```

显示访问权设置

使用 **dspmqaut** 控制命令或 **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。

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 **MQCMD_INQUIRE_ENTITY_AUTH** PCF 命令及其语法的完整定义，请参阅 查询实体权限。

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
```

更改和撤销对 IBM WebSphere 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** 命令。 要撤销作为具有权限的组的成员的特定用户的访问权，请从该组中除去该用户。

以下描述了从组中除去用户的过程:

- 第 66 页的『在 Windows 上创建和管理组』
- 第 68 页的『在 HP-UX 上创建和管理组』
- 第 70 页的『在 AIX 上创建和管理组』
- 第 71 页的『在 Solaris 上创建和管理组』
- 第 71 页的『在 Linux 上创建和管理组』

.

创建 IBM WebSphere 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 **MQCMD_INQUIRE_ENTITY_AUTH** PCF 命令及其语法的完整定义，请参阅 Inquire Entity Authority。

在 Windows 上，先删除与特定 Windows 用户帐户对应的 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。

阻止对 UNIX, Linux, and Windows 系统进行安全访问检查

要关闭所有安全性检查，可以禁用 OAM。 这可能适用于测试环境。 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 (如果执行此操作，那么以后无法添加 OAM):

  请参阅 环境变量，以获取有关设置 MQSNOAUT 变量的 含义 的更多信息。
- 编辑队列管理器配置文件以除去服务。 (如果执行此操作，那么以后无法添加 OAM。)

如果在禁用 OAM 时使用 setmqaut 或 dspmqaut，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。

警告: 除去 OAM 时，不能将其放回现有队列管理器。 此 是因为 OAM 需要在对象创建时就位。 要在除去 WebSphere MQ OAM 后再次使用该 OAM ， 需要重建队列管理器。

相关概念

可安装服务

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 WebSphere MQ 系统。

关于此任务

在此任务期间，您决定需要执行哪些操作才能将相应级别的安全性应用于 WebSphere 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 139 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 144 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 148 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 149 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 150 页的『除去与队列管理器的连接』 中所述
   b) 是: 请参阅 第 150 页的『允许用户应用程序连接到队列管理器』。

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 *14:* 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 140 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 140 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 141 页的『授予对某些通道的有限管理访问权』 中所述

表 *14:* 授予对队列管理器资源子集的部分管理访问权 *(继续)*	
用户需要管理此类型的对象	**执行此操作**
队列管理器	授予对队列管理器的部分管理访问权，如 第 141 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 142 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 143 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 143 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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName ReqdAction
```

- 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。

  ReqdAction
  允许组执行的操作:

  – 在 UNIX， Linux 和 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。

  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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName ReqdAction
```

- 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:

  – 在 UNIX， Linux 和 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。

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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName ReqdAction
```

- 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。

  ReqdAction
  允许组执行的操作:

  – 在 UNIX， Linux 和 Windows 系统上，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。

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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName ReqdAction
```

- 对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*MQM) USER(GroupName) AUT(ReqdAction)
MQMNAME('QMgrName')
```

结果

要确定用户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:

- 在 UNIX， Linux 和 Windows 系统上，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。

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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName ReqdAction
  ```

- 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:

  – 在 UNIX， Linux 和 Windows 系统上，以下权限的任意组合: + chg， + clr， + crt， + dlt 和 + dsp。 authorization + alladm 相当于 + chg + clr + dlt + dsp。

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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName ReqdAction
```

- 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:

  – 在 UNIX， Linux 和 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。

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。

注: 服务对象在 z/OS 上不存在。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName ReqdAction
```

- 对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*SVC) USER(GroupName) AUT(ReqdAction)
MQMNAME('QMgrName')
```

结果

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:

- 在 UNIX， Linux 和 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。

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 *15:* 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 144 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 145 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 145 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 146 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 146 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 147 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 147 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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +alladm
```

- 对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*ALLADM) MQMNAME('QMgrName')
```

- 对于 z/OS，请发出以下命令:

```
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。 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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +alladm
```

- 对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*TOPIC) USER(GroupName) AUT(ALLADM)
MQMNAME('QMgrName')
```

- 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)
```

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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName +alladm
```

- 对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*CHL) USER(GroupName) AUT(ALLADM) MQMNAME('QMgrName')
```

- 对于 z/OS，请发出以下命令:

```
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。 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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -t qmgr -g GroupName +alladm
```

- 对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*MQM) USER(GroupName) AUT(*ALLADM) MQMNAME('QMgrName')
```

- 对于 z/OS，请发出以下命令:

```
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。 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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +alladm
```

- 对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*PRC) USER(GroupName) AUT(*ALLADM) MQMNAME('QMgrName')
```

- 对于 z/OS，请发出以下命令:

```
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。 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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName +alladm
```

- 对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALLADM)
MQMNAME('QMgrName')
```

- 对于 z/OS，请发出以下命令:

```
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。 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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName +alladm
```

- 对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*SVC) USER(GroupName) AUT(*ALLADM) MQMNAME('QMgrName')
```

- 对于 z/OS，请发出以下命令:

```
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。 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。

过程

- 使用向导:
  a) 在 WebSphere 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。
- 对于 UNIX 和 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
setmqaut -m QMgrName -n ** -t topic -g GroupName +dsp
setmqaut -m QMgrName -n ** -t channel -g GroupName +dsp +inq
setmqaut -m QMgrName -n ** -t clntconn -g GroupName +dsp
setmqaut -m QMgrName -n ** -t authinfo -g GroupName +dsp
setmqaut -m QMgrName -n ** -t listener -g GroupName +dsp
setmqaut -m QMgrName -n ** -t namelist -g GroupName +dsp
setmqaut -m QMgrName -n ** -t process -g GroupName +dsp
setmqaut -m QMgrName -n ** -t service -g GroupName +dsp
setmqaut -m QMgrName -t qmgr -g GroupName +dsp +inq +connect
```

仅当要使用 MQ Explorer 时，才需要 SYSTEM.ADMIN.COMMAND.QUEUE 和 SYSTEM.MQEXPLORER.REPLY.MODEL 的特定权限。

- 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')
```

- 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 MQTOPIC QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MQTOPIC) ID(GroupName) ACCESS(READ)
RDEFINE MQPROC QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MQPROC) ID(GroupName) ACCESS(READ)
```

```
RDEFINE MQNLIST QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MQNLIST) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
PERMIT QMgrName.BATCH CLASS(MQCONN) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.CICS UACC(NONE)
PERMIT QMgrName.CICS CLASS(MQCONN) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.IMS UACC(NONE)
PERMIT QMgrName.IMS CLASS(MQCONN) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.CHIN UACC(NONE)
PERMIT QMgrName.CHIN CLASS(MQCONN) ID(GroupName) ACCESS(READ)
```

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。

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

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。

过程

- 使用向导:
  a) 在 WebSphere 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。
- 对于 UNIX 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.QUEUE -t queue -g GroupName +dsp +inq +get
setmqaut -m QMgrName -n '**' -t topic -g GroupName +alladm
setmqaut -m QMgrName -n @class -t topic -g GroupName +crt
setmqaut -m QMgrName -n '**' -t channel -g GroupName +alladm
setmqaut -m QMgrName -n @class -t channel -g GroupName +crt
setmqaut -m QMgrName -n '**' -t clntconn -g GroupName +alladm
setmqaut -m QMgrName -n @class -t clntconn -g GroupName +crt
setmqaut -m QMgrName -n '**' -t authinfo -g GroupName +alladm
setmqaut -m QMgrName -n @class -t authinfo -g GroupName +crt
setmqaut -m QMgrName -n '**' -t listener -g GroupName +alladm
setmqaut -m QMgrName -n @class -t listener -g GroupName +crt
setmqaut -m QMgrName -n '**' -t namelist -g GroupName +alladm
setmqaut -m QMgrName -n @class -t namelist -g GroupName +crt
setmqaut -m QMgrName -n '**' -t process -g GroupName +alladm
setmqaut -m QMgrName -n @class -t process -g GroupName +crt
setmqaut -m QMgrName -n '**' -t service -g GroupName +alladm
setmqaut -m QMgrName -n @class -t service -g GroupName +crt
setmqaut -m QMgrName -t qmgr -g GroupName +alladm +conn
```

- 对于 Windows 系统，请发出与 UNIX and Linux 系统相同的命令，但使用概要文件名称 @CLASS 而不是 @class。
- 对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*ALL) USER('GroupName') AUT(*ALLADM) MQMNAME('QMgrName')
```

- 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。 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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -t qmgr -g GroupName -connect
```

- 对于 IBM i，请发出以下命令:

```
RVKMQMAUT OBJ ('QMgrName') OBJTYPE(*MQM) USER(*ALL) AUT(*CONNECT)
```

- 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。 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 157 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 151 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 157 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 157 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 162 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 163 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 164 页的『授予访问进程的权限』
您有使用名称列表的应用程序	请参阅 第 164 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 SSL 或 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 SSL 或 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 SSL 或 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) 根据在传递到 MQCD 结构中的出口的 SSLPeerNamePtr 和 SSLPeerName 长度字段中接收到的 SSL DN 的值，编写安全出口以分配 MCAUSER 值。
3. 将 SSL 或 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 SSL 或 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 151 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 122 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。 如果在下面显示的命令中指定服务器连接通道的名称，那么它没有任何作用。

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

将客户机断言的用户标识映射到 *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原始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 是客户机声明的用户标识。

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

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
) 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 WebSphere MQ 规则的字符串。 请参阅 WebSphere MQ 规则以获取 SSLPEER 值。

user 是要用于使用指定 DN 的所有连接的用户标识。

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。 如果在下面显示的命令中指定服务器连接通道的名称，那么它没有任何作用。

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

阻止对客户机声明的用户标识进行访问

您可以使用通道认证记录来阻止客户机断言的用户标识启动通道。

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 是客户机声明的用户标识。

相关参考

SET CHLAUTH

阻止访问 *SSL* 专有名称

您可以使用通道认证记录来阻止 SSL 专有名称启动通道。

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')
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 WebSphere MQ 规则的字符串。 请参阅 WebSphere MQ 规则以获取 SSLPEER 值。

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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

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

- 对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ('QMgrName') OBJTYPE(*MQM)  USER('GroupName') AUT(*CONNECT)
```

- 对于 z/OS，请发出以下命令:

```
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。

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 158 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 158 页的『授予设置上下文的权限』

语句	操作
应用程序传递上下文	请参阅 第 159 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 206 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 160 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 160 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 161 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。

过程
- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +get
  ```

- 对于 IBM i，请发出以下命令:

  ```
  GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*GET) MQMNAME('QMgrName')
  ```

- 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。

过程
- 对于 UNIX， Linux 和 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
```

- 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')
```

- 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。

过程

- 对于 UNIX， Linux 和 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
```

- 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')
```

- 对于 z/OS，请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。

过程
- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +put
```

- 对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME('QMgrName')
```

- 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。

过程
- 对于 UNIX， Linux 和 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
```

- 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')
```

- 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 160 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。

在 UNIX， Linux 和 Windows 系统上，还可以使用 SET AUTHREC 命令。

过程

- 对于 UNIX， Linux 和 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。

- 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。

- 对于 z/OS，请发出以下命令:

```
RDEFINE MQQUEUE  QMgrNameObjectProfile UACC(NONE)
PERMIT QMgrNameObjectProfile CLASS(MQADMIN)
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 *16:* 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 162 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 162 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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +pub
```

- 对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*TOPIC) USER(GroupName) AUT(*PUB) MQMNAME('QMgrName')
```

- 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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +sub
```

- 对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*TOPIC) USER(GroupName) AUT(*SUB) MQMNAME('QMgrName')
```

- 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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName +inq
```

- 对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*MQM) USER(GroupName) AUT(*INQ) MQMNAME('QMgrName')
```

- 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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +all
```

- 对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*PRC) USER(GroupName) AUT(*ALL) MQMNAME('QMgrName')
```

- 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。

过程

- 对于 UNIX， Linux 和 Windows 系统，请发出以下命令:

```
setmqaut -m QMgrName -n
ObjectProfile -t namelist -g GroupName
 +all
```

- 对于 IBM i，请发出以下命令:

```
GRTMQMAUT OBJ('ObjectProfile
') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALL) MQMNAME('
QMgrName')
```

- 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。

在 UNIX, Linux, and Windows 系统上管理 IBM WebSphere MQ 的权限

IBM WebSphere MQ 管理员可以使用所有 IBM WebSphere 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。 其他注意事项适用于 Windows 系统。

IBM WebSphere MQ 管理员有权使用所有 WebSphere MQ 命令 (包括用于向其他用户授予 WebSphere MQ 权限的命令)

要成为 IBM WebSphere MQ 管理员，您必须是名为 *mqm* 组的特殊组的成员 (或者 Windows 系统上的 Administrators 组的成员)。 安装 WebSphere MQ 时，将自动创建 mqm 组; 将更多用户添加到该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 WebSphere 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 WebSphere 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。

管理员可以使用控制命令 **runmqsc** 来发出 IBM WebSphere MQ 脚本 (MQSC) 命令。 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。 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。 WebSphere 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 WebSphere MQ Explorer 来管理本地系统上的队列管理器。 当 IBM WebSphere 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 WebSphere 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

此外，在 Windows 系统上， SYSTEM 帐户具有对 WebSphere MQ 资源的完全访问权。

在 UNIX 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 WebSphere MQ 对象都由用户标识 mqm 拥有。

在 Windows 系统上， Administrators 组的成员还可以像 SYSTEM 帐户一样管理任何队列管理器。 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 WebSphere 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 组的成员具有使用所有对象的权限，但是在 Windows 系统上，如果您具有本地用户和具有相同名称的域认证用户，那么可能存在权限被拒绝的情况。 在第 168 页的『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设施上执行的操作，即使他们是 Administrators 组的成员也是如此。 如果您的用户标识在 Administrators 组中，但不在 mqm 组中，那么必须使用提升的命令提示符来发出 WebSphere 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 57 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 (除非要使用 MQCONNX 调用上的快速路径绑定)。
- 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 命令来显示队列管理器。
- 使用 dspmqtrc 命令可显示 WebSphere 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 V 5.3 已提高此限制，但 WebSphere 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 WebSphere MQ 会将其替换为值 UNKNOWN。 请勿定义值为 UNKNOWN 的用户标识。

管理 mqm 组

mqm 组中的用户被授予基于 WebSphere 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 WebSphere MQ 管理员的帐户。

以下内容中描述了这些任务:

- 在 Windows 上创建和管理组
- 在 HP-UX 上创建和管理组
- 在 AIX 上创建和管理组
- 在 Solaris 上创建和管理组
- 在 Linux 上创建和管理组

如果域控制器在 Windows 2000 或 Windows 2003 上运行，那么域管理员可能必须设置一个特殊帐户以供 WebSphere MQ 使用。 这在 配置 WebSphere MQ 帐户中进行了描述。

在 UNIX, Linux, and Windows 系统上使用 IBM WebSphere MQ 对象的权限

所有对象都受 IBM WebSphere 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 WebSphere 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，可能允许一个组执行放置和获取操作; 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 165 页的『在 UNIX, Linux, and Windows 系统上管理 IBM WebSphere MQ 的权限』
IBM WebSphere MQ 管理员可以使用所有 IBM WebSphere 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。 其他注意事项适用于 Windows 系统。

在 UNIX, Linux, and Windows 系统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， WebSphere MQ 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 WebSphere MQ; WebSphere 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)
通过打开对象并对其发出命令来访问 WebSphere 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)。

WebSphere 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。 注释 第 74 页的『1』对此进行了描述。

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。

IBM WebSphere MQ 在 UNIX, Linux, and Windows 系统上如何实施访问控制

IBM WebSphere 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 WebSphere 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 WebSphere MQ 的一部分。 WebSphere MQ 提供了称为 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 138 页的『阻止对 UNIX, 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 WebSphere MQ 对象。 第 131 页的『在 UNIX， Linux 和 Windows 系统上使用 OAM 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。此队列的访问权限制为 mqm 组中的用户，此外在 Windows 上限制为 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。

WebSphere 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 131 页的『在 UNIX， Linux 和 Windows 系统上使用 OAM 控制对对象的访问』。

WebSphere 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。 WebSphere MQ 遵循 OAM 的决策; 如果 OAM 无法做出决策，那么 WebSphere MQ 不允许访问。

在 UNIX, Linux, and Windows 系统上标识用户标识

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 WebSphere 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 UNIX 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 WebSphere 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 169 页的『UNIX， Linux 和 Windows 系统上的上下文权限』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。

主体和组

主体可以属于组。 您可以将特定资源的访问权授予组而不是个人，以减少所需的管理量。 在 UNIX and Linux 系统上，所有访问控制表 (ACL) 都基于组，但在 Windows 系统上， ACLS 都基于用户标识和组。

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。 此过程在以下内容中描述:

- 在 Windows 上创建和管理组
- 在 HP-UX 上创建和管理组
- 在 AIX 上创建和管理组
- 在 Solaris 上创建和管理组
- 在 Linux 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 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。

UNIX and Linux 系统

所有 ACL 都基于组。 当授予用户对特定资源的访问权时，用户标识的主组将包含在 ACL 中。 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 组中的授权，以向没有特定授权的用户授予对 WebSphere MQ 资源的访问权。

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。

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 系统

ACL 基于用户标识和组。 检查与针对 UNIX 系统的检查相同，不同用户标识也可以显示在 ACL 中。 您可以在具有相同用户标识的不同域上具有不同的用户。 WebSphere MQ 允许由域名限定用户标识，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\GroupName
```

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 安全性。
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 WebSphere 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，如果从通过全局组具有本地 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。

Windows 安全标识 *(SID)*

Windows 上的 WebSphere MQ 使用可用的 SID。 如果未随授权请求提供 Windows SID ，那么 WebSphere 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 WebSphere MQ for Windows 上创建消息时，WebSphere MQ 会将 SID 存储在消息描述符中。 当 Windows 上的 WebSphere MQ 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 WebSphere MQ 仅根据用户名来标识用户。 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 WebSphere MQ 权限。 要防止此问题，请在每个授权请求中包含 SID; SID 由 WebSphere 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。

UNIX， Linux 和 Windows 系统上的备用用户权限

您可以指定用户标识在访问 WebSphere 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 WebSphere MQ 对象上使用该权限。

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。

UNIX， Linux 和 Windows 系统上的上下文权限

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 概述 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 WebSphere 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。 *MCAUserIdentifier* 包含 MCA 用户标识的前 12 个字节。 如果 MCA 用户标识不为空，那么它指定消息通道代理程序用于授权访问 MQ 资源的用户标识。 确保 MCAUSER 在 Windows 平台上少于 12 个字符。

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。

在 IBM i， UNIX， Linux 和 Windows 系统上，除非 *MCAUserIdentifier* 的值为空，否则当 MCA 在连接到队列管理器之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。

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 WebSphere 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。 同样，在 MQI 通道上，通道客户机端的安全出口可以发送与 WebSphere MQ MQI 客户机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 SSL，那么合作伙伴的专有名称 (DN) 将传递到 MQCD 的 SSLPeerNamePtr 字段中的出口，并且该证书的签发者的 DN 将传递到 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 WebSphere MQ v7.1 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。

WebSphere MQ 对象权限管理器用户认证

在 WebSphere MQ MQI 客户机连接上，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 WebSphere 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 (或在 z/OS 上设置为 ALTMCA)，那么当 MCA 打开目标队列时，每条入局消息的 *UserIdentifier* 字段中的用户标识将用于权限检查。

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 123 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 WebSphere 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。

消息的机密性

要保持机密性，请对消息进行加密。 根据您的需要， WebSphere MQ 中有多种加密消息的方法。

您选择的 CipherSpec 将确定您具有的机密性级别。

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 WebSphere MQ Advanced Message Security 来加密消息，或者编写您自己的 API 出口或 API 交叉出口。

如果仅在通过通道传输消息时需要对消息进行加密，因为您在队列管理器上具有足够的安全性，那么可以使用 SSL 或 TLS，也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。

有关 WebSphere MQ Advanced Message Security 的更多信息，请参阅 第 52 页的『规划 Advanced Message Security』。在 第 20 页的『IBM WebSphere MQ 支持 SSL 和 TLS』中描述了将 SSL 和 TLS 与 WebSphere MQ 配合使用。 在 第 188 页的『在用户出口程序中实现机密性』中描述了在消息加密中使用出口程序。

使用 SSL 或 TLS 连接两个队列管理器

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。 您还必须获取和管理数字证书。 在测试系统上，可以使用自签名证书或本地认证中心 (CA) 颁发的证书。 在生产系统上，请勿使用自签名证书。 有关更多信息，请参阅 ..//zs14140_.dita。

有关创建和管理证书的完整信息，请参阅 第 95 页的『在 UNIX, Linux, and Windows 系统上使用 SSL 或 TLS』。

此主题集合介绍了设置 SSL 通信所涉及的任务，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 WebSphere MQ 实现， SSL 或 TLS 服务器始终从客户机请求证书。

注意:

1. 在此上下文中， SSL 客户机是指启动握手的连接。
2. 请参阅 词汇表 以获取更多详细信息。

在 UNIX， Linux 和 Windows 系统上，仅当 SSL 或 TLS 客户机具有以正确的 WebSphere MQ 格式标注的证书时，才会发送证书，后跟 `ibmwebspheremq` ，然后将队列管理器的名称更改为小写。 例如，对于 `QM1`，`ibmwebspheremqqm1`。

WebSphere 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，那么仅当充当 SSL 或 TLS 服务器的通道的末尾定义了 SSLCAUTH 参数设置为 REQUIRED 或 SSLPEER 参数值时，认证才会失败。 有关匿名连接队列管理器的更多信息，即，当 SSL 或 TLS 客户机未发送证书时，请参阅 第 176 页的『使用单向认证连接两个队列管理器』。

使用自签名证书对两个队列管理器进行相互认证

遵循以下样本指示信息，以使用自签名 SSL 或 TLS 证书在两个队列管理器之间实现相互认证。

关于此任务

方案:

- 您有两个需要安全通信的队列管理器 QM1 和 QM2。 您需要在 QM1 和 QM2 之间执行相互认证。
- 您已决定使用自签名证书测试安全通信。

生成的配置如下所示:

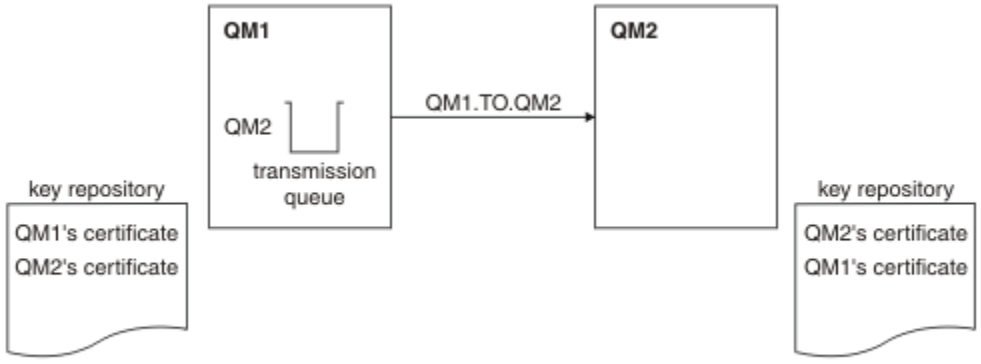


图 *14:* 由此任务生成的配置

在 第 172 页的图 14 中， QM1 的密钥存储库包含 QM1 的证书和来自 QM2 的公用证书。 QM2 的密钥存储库包含 QM2 的证书以及来自 QM1 的公用证书。

过程

1. 根据操作系统，在每个队列管理器上准备密钥存储库:
   - 在 UNIX， Linux 和 Windows 系统上。
2. 为每个队列管理器创建自签名证书:
   - 在 UNIX， Linux 和 Windows 系统上。
3. 抽取每个证书的副本:
   - 在 UNIX， Linux 和 Windows 系统上。
4. 使用诸如 FTP 中所述。
5. 将合作伙伴证书添加到每个队列管理器的密钥存储库:
   - 在 UNIX， Linux 和 Windows 系统上。
6. 在 QM1 上，通过发出类似以下示例的命令来定义发送方通道和关联的传输队列:

   ```
   DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QM1.MACH.COM) XMITQ(QM2)
   SSLCIPH(RC4_MD5_US) DESCR('Sender channel using SSL from QM1 to QM2')

   DEFINE QLOCAL(QM2) USAGE(XMITQ)
   ```

   此示例使用 CipherSpec RC4_MD5。 通道两端的 CipherSpecs 必须相同。
7. 在 QM2 上，通过发出类似于以下示例的命令来定义接收方通道:

   ```
   DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(RC4_MD5_US)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from QM1 to QM2')
   ```

   通道必须与您在步骤 6 中定义的发送方通道同名，并使用相同的 CipherSpec。
8. 启动通道中所述。

结果

创建密钥存储库和通道，如 第 172 页的图 14 中所述

下一步做什么

使用 DISPLAY 命令检查任务是否已成功完成。 如果任务成功，那么生成的输出类似于以下示例中显示的输出。

从队列管理器 QM1，输入以下命令:

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

生成的输出类似于以下示例:

```
DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(QM1.TO.QM2)                     CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QM2)
   SSLCERTI("CN=QM2,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5E:02,CN=QM2,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QM2)
```

从队列管理器 QM2 中，输入以下命令:

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

生成的输出类似于以下示例:

```
DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
```

```
   CHANNEL(QM2.TO.QM1)                     CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QM1)
   SSLCERTI("CN=QM1,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QM1,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
   XMITQ( )
```

在每种情况下， SSLPEER 的值必须与步骤 2 中创建的伙伴证书中的 DN 的值匹配。 颁发者名称与同级名称匹配，因为证书是自签名的。

SSLPEER 是可选的。 如果指定了此值，那么必须设置其值，以便允许使用伙伴证书 (在步骤 2 中创建) 中的 DN。 有关使用 SSLPEER 的更多信息，请参阅 WebSphere MQ 规则以获取 SSLPEER 值。

使用 CA 签名的证书对两个队列管理器进行相互认证

遵循以下样本指示信息以使用 CA 签署的 SSL 或 TLS 证书在两个队列管理器之间实现相互认证。

关于此任务

方案:

- 您有两个名为 QMA 和 QMB 的队列管理器，它们需要安全地进行通信。 您需要在 QMA 和 QMB 之间执行相互认证。
- 将来您计划在生产环境中使用此网络，因此您决定从一开始就使用 CA 签署的证书。

生成的配置如下所示:

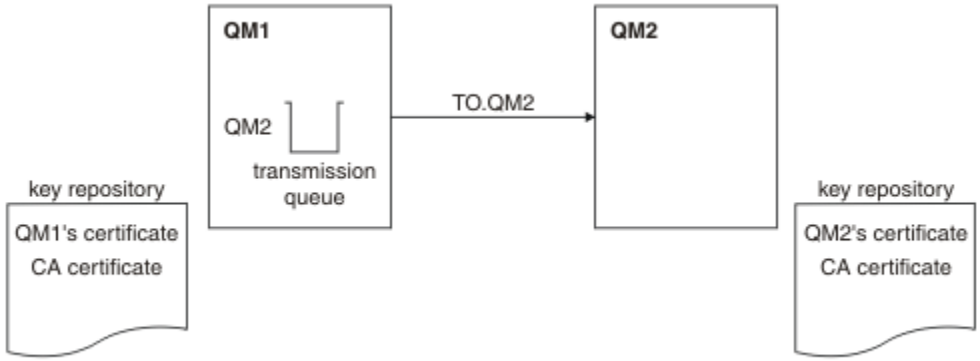


图 *15:* 由此任务生成的配置

在 第 174 页的图 15 中， QMA 的密钥存储库包含 QMA 的证书和 CA 证书。 QMB 的密钥存储库包含 QMB 的证书和 CA 证书。 在此示例中， QMA 的证书和 QMB 的证书都由同一 CA 发放。 如果 QMA 的证书和 QMB 的证书是由不同的 CA 发放的，那么 QMA 和 QMB 的密钥存储库必须同时包含两个 CA 证书。

过程

1. 根据操作系统，在每个队列管理器上准备密钥存储库:
   - 在 UNIX， Linux 和 Windows 系统上。
2. 请求每个队列管理器的 CA 签名证书。
   您可以对两个队列管理器使用不同的 CA。
   - 在 UNIX， Linux 和 Windows 系统上。
3. 将认证中心证书添加到每个队列管理器的密钥存储库:
   如果队列管理器使用不同的认证中心，那么必须将每个认证中心的 CA 证书添加到两个密钥存储库。
   - 在 UNIX， Linux 和 Windows 系统上。

4. 将 CA 签名的证书添加到每个队列管理器的密钥存储库:
   - 在 UNIX， Linux 和 Windows 系统上。
5. 在 QMA 上，通过发出类似于以下示例的命令来定义发送方通道和关联的传输队列:

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QMB.MACH.COM) XMITQ(QMB)
SSLCIPH(RC2_MD5_EXPORT) DESCR('Sender channel using SSL from QMA to QMB')

DEFINE QLOCAL(QMB) USAGE(XMITQ)
```

   此示例使用 CipherSpec RC4_MD5。 通道两端的 CipherSpecs 必须相同。
6. 在 QMB 上，通过发出类似于以下示例的命令来定义接收方通道:

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(RC2_MD5_EXPORT)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL to QMB')
```

   通道必须与您在步骤 6 中定义的发送方通道同名，并使用相同的 CipherSpec。
7. 启动通道:

结果

将创建密钥存储库和通道，如 第 174 页的图 15 中所述。

下一步做什么

使用 DISPLAY 命令检查任务是否已成功完成。 如果任务成功，那么生成的输出类似于以下示例中显示的输出。

从队列管理器 QMA ，输入以下命令:

```
DISPLAY CHS(TO.QMB) SSLPEER SSLCERTI
```

生成的输出类似于以下示例:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QMB)                         CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QMB)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMB,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QMB)
```

从队列管理器 QMB 中，输入以下命令:

```
DISPLAY CHS(TO.QMB) SSLPEER SSLCERTI
```

生成的输出类似于以下示例:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QMB)                         CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QMA)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMA,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
   XMITQ( )
```

在每种情况下， SSLPEER 的值必须与步骤 2 中创建的伙伴证书中的专有名称 (DN) 的值匹配。 签发者名称与对步骤 4 中添加的个人证书进行签名的 CA 证书的主体集 DN 相匹配。

使用单向认证连接两个队列管理器

遵循以下样本指示信息来修改具有相互认证的系统，以允许队列管理器使用单向认证连接到另一个系统; 即，当 SSL 或 TLS 客户机未发送证书时。

关于此任务

方案:

- 您的两个队列管理器 (QM1 和 QM2) 已按 第 174 页的『使用 CA 签名的证书对两个队列管理器进行相互认证』中的方式进行设置。
- 您希望更改 QM1 ，以便它使用单向认证连接到 QM2。

生成的配置如下所示:

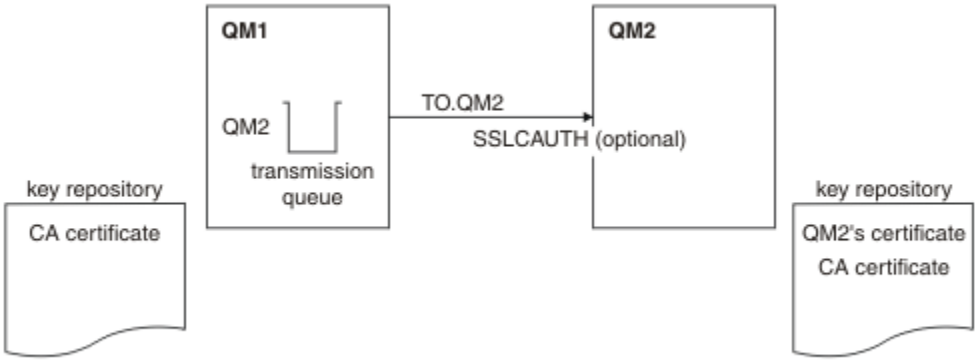


图 *16:* 允许单向认证的队列管理器

过程

1. 根据操作系统，从其密钥存储库中除去 QM1 的个人证书:
   - 在 UNIX， Linux 和 Windows 系统上。 该证书标注如下:
     - `ibmwebspheremq` 后跟已折叠为小写的队列管理器的名称。 例如，对于 `QM1` ，`ibmwebspheremqqm1`。
2. 可选： 在 QM1 上，如果先前已运行任何 SSL 或 TLS 通道，请刷新 SSL 或 TLS 环境中所述。
3. 在接收方中所述。

结果

密钥存储库和通道已更改，如 第 176 页的图 16 中所示

下一步做什么

如果发送方通道正在运行，并且您在步骤 2 中发出了 REFRESH SECURITY TYPE (SSL) 命令，那么通道将自动重新启动。 如果发送方通道未运行，请将其启动。

在通道的服务器端，通道状态屏幕上存在对等名称参数值指示客户机证书已流动。

通过发出一些 DISPLAY 命令验证任务是否已成功完成。 如果任务成功，那么生成的输出类似于以下示例中显示的输出:

从 QM1 队列管理器中，输入以下命令:

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

生成的输出将类似于以下示例:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(TO.QMB) SSLPEER
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QM2)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMB,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QM2)
```

从 QM2 队列管理器中，输入以下命令:

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

生成的输出将类似于以下示例:

```
DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QMA)                            SSLCERTI( )
   SSLPEER( )                              STATUS(RUNNING)
   SUBSTATE(RECEIVE)                       XMITQ( )
```

在 QM2 上， SSLPEER 字段为空，显示 QM1 未发送证书。 在 QM1 上， SSLPEER 的值与 QM2 个人证书中的 DN 的值相匹配。

安全地将客户机连接到队列管理器

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。 您还必须获取和管理数字证书。 在测试系统上，可以使用自签名证书或本地认证中心 (CA) 颁发的证书。 在生产系统上，请勿使用自签名证书。 有关更多信息，请参阅 ..//zs14140_.dita。

有关创建和管理证书的完整信息，请参阅 第 95 页的『在 UNIX, Linux, and Windows 系统上使用 SSL 或 TLS』。

此主题集合介绍了设置 SSL 通信所涉及的任务，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 WebSphere MQ 实现， SSL 或 TLS 服务器始终从客户机请求证书。

在 UNIX, Linux, and Windows 系统上，仅当 SSL 或 TLS 客户机具有以正确的 WebSphere MQ 格式标注的证书时， SSL 或 TLS 客户机才会发送证书，该格式为 `ibmwebspheremq` 后跟更改为小写的登录用户标识，例如 `ibmwebspheremqmyuserid`。

WebSphere 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，那么仅当充当 SSL 或 TLS 服务器的通道的末尾定义了 SSLCAUTH 参数设置为 REQUIRED 或 SSLPEER 参数值时，认证才会失败。 有关以匿名方式连接队列管理器的更多信息，请参阅 第 180 页的『以匿名方式将客户机连接到队列管理器』。

使用自签名证书对客户机和队列管理器进行相互认证

遵循以下样本指示信息，通过使用自签名 SSL 或 TLS 证书来实现客户机与队列管理器之间的相互认证。

关于此任务

方案:

- 您具有需要安全通信的客户机 C1 和队列管理器 QM1。 您需要在 C1 和 QM1 之间执行相互认证。
- 您已决定使用自签名证书来测试安全通信。

IBM i 上的 DCM 不支持自签名证书，因此此任务不适用于 IBM i 系统。

生成的配置如下所示:

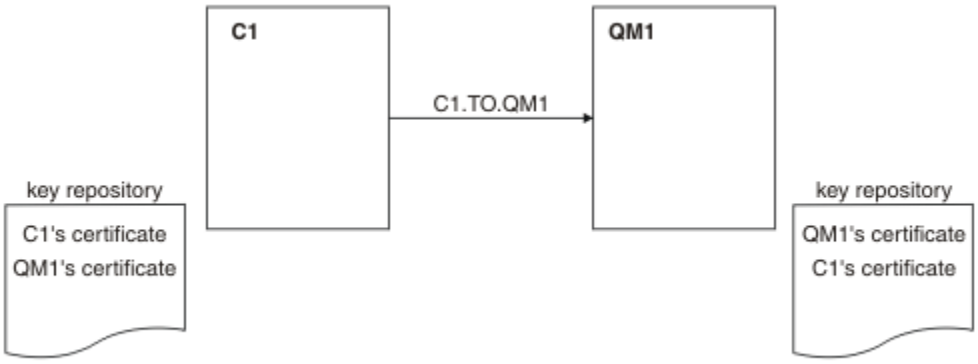


图 *17:* 由此任务生成的配置

在 第 178 页的图 17 中， QM1 的密钥存储库包含 QM1 的证书和来自 C1 的公用证书。 C1 的密钥存储库包含 C1 的证书以及来自 QM1 的公用证书。

过程

1. 根据操作系统，在客户机和队列管理器上准备密钥存储库:
   - 在 UNIX， Linux 和 Windows 系统上。
2. 为客户机和队列管理器创建自签名证书:
   - 在 UNIX， Linux 和 Windows 系统上。
3. 抽取每个证书的副本:
   - 在 UNIX， Linux 和 Windows 系统上。
4. 使用诸如 FTP 中所述。
5. 将伙伴证书添加到客户机和队列管理器的密钥存储库:
   - 在 UNIX， Linux 和 Windows 系统上。
6. 在队列管理器上发出命令 REFRESH SECURITY TYPE (SSL)。
7. 通过以下任一方式定义客户机连接通道:
   - 将 MQCONNX 调用与 C1 上的 MQSCO 结构配合使用，如 在 WebSphere MQ MQI 客户机上创建客户机连接通道中所述。
   - 使用客户机通道定义表，如 在服务器上创建服务器连接和客户机连接定义 中所述。
8. 在 QM1 上，通过发出类似于以下示例的命令来定义服务器连接通道:

   ```
   DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC4_MD5_US)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
   ```

   通道必须与您在步骤 6 中定义的客户机连接通道同名，并使用相同的 CipherSpec。

结果

创建密钥存储库和通道，如 第 178 页的图 17 中所述

下一步做什么

使用 DISPLAY 命令检查任务是否已成功完成。 如果任务成功，那么生成的输出与以下示例中显示的输出相似。

从队列管理器 QM1，输入以下命令:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

生成的输出类似于以下示例:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                       CURRENT
   SSLCERTI("CN=QM1,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5E:02,CN=QM2,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                           SUBSTATE(RECEIVE)
```

可选择设置通道定义的 SSLPEER 过滤器属性。 如果设置了通道定义 SSLPEER ，那么其值必须与步骤 2 中创建的伙伴证书中的主体集 DN 相匹配。 成功连接后， DISPLAY CHSTATUS 输出中的 SSLPEER 字段将显示远程客户机证书的主题 DN。

使用 CA 签名的证书对客户机和队列管理器进行相互认证

遵循以下样本指示信息，通过使用 CA 签署的 SSL 或 TLS 证书来实现客户机与队列管理器之间的相互认证。

关于此任务

方案:

- 您具有需要安全通信的客户机 C1 和队列管理器 QM1。 您需要在 C1 和 QM1 之间执行相互认证。
- 将来您计划在生产环境中使用此网络，因此您决定从一开始就使用 CA 签署的证书。

生成的配置如下所示:

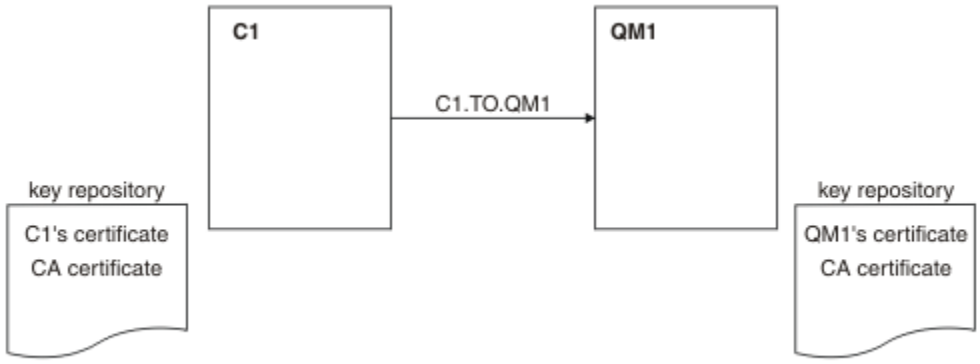


图 *18:* 由此任务生成的配置

在 第 179 页的图 18 中， C1 的密钥存储库包含 C1 的证书和 CA 证书。 QM1 的密钥存储库包含 QM1 的证书和 CA 证书。 在此示例中， C1 的证书和 QM1 的证书都由同一 CA 发放。 如果 C1 的证书和 QM1 的证书是由不同的 CA 发放的，那么 C1 和 QM1 的密钥存储库必须同时包含两个 CA 证书。

过程

1. 根据操作系统，在客户机和队列管理器上准备密钥存储库:
   - 在 UNIX， Linux 和 Windows 系统上。

2. 请求客户机和队列管理器的 CA 签名证书。
   您可以对客户机和队列管理器使用不同的 CA。
   - 在 UNIX， Linux 和 Windows 系统上。
3. 将认证中心证书添加到客户机和队列管理器的密钥存储库。
   如果客户机和队列管理器使用不同的认证中心，那么必须将每个认证中心的 CA 证书添加到两个密钥存储库。
   - 在 UNIX， Linux 和 Windows 系统上。
4. 将 CA 签名的证书添加到客户机和队列管理器的密钥存储库:
   - 在 UNIX， Linux 和 Windows 系统上。
5. 通过以下任一方式定义客户机连接通道:
   - 将 MQCONNX 调用与 C1 上的 MQSCO 结构配合使用，如 在 WebSphere MQ MQI 客户机上创建客户机连接通道中所述。
   - 使用客户机通道定义表，如 在服务器上创建服务器连接和客户机连接定义 中所述。
6. 在 QM1 上，通过发出类似于以下示例的命令来定义服务器连接通道:

```
DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC2_MD5_EXPORT)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
```

   通道必须与您在步骤 6 中定义的客户机连接通道同名，并使用相同的 CipherSpec。

结果

将创建密钥存储库和通道，如 第 179 页的图 18 中所述。

下一步做什么

使用 DISPLAY 命令检查任务是否已成功完成。 如果任务成功，那么生成的输出类似于以下示例中显示的输出。

从队列管理器 QM1 中，输入以下命令:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
```

生成的输出类似于以下示例:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                     CURRENT
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMA,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

DISPLAY CHSTATUS 输出中的 SSLPEER 字段显示在步骤 2 中创建的远程客户机证书的主题 DN。 签发者名称与对步骤 4 中添加的个人证书进行签名的 CA 证书的主体集 DN 相匹配。

以匿名方式将客户机连接到队列管理器

遵循以下样本指示信息来修改具有相互认证的系统，以允许队列管理器以匿名方式连接到另一个队列管理器。

关于此任务

方案:

- 您的队列管理器和客户机 (QM1 和 C1) 已设置为与 第 179 页的『使用 CA 签名的证书对客户机和队列管理器进行相互认证』 中一样。

- 您希望更改 C1 ，以便它以匿名方式连接到 QM1。

生成的配置如下所示:

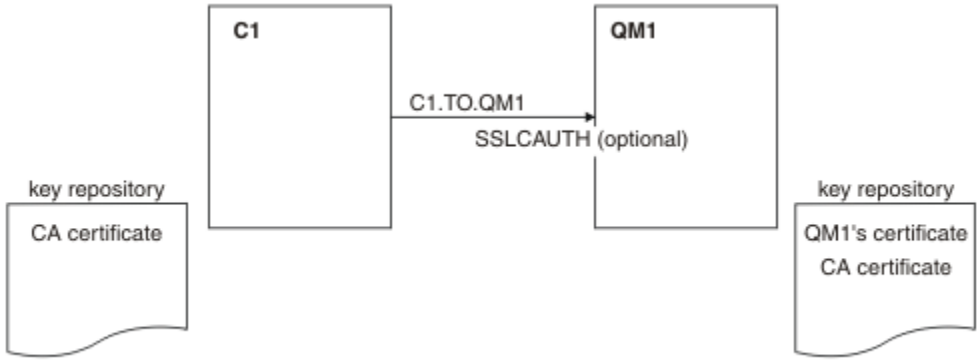


图 *19:* 允许匿名连接的客户机和队列管理器

过程

1. 根据操作系统，从 C1 的密钥存储库中除去个人证书:
   - 在 UNIX， Linux 和 Windows 系统上。 该证书标注如下:
     - ibmwebspheremq 后跟转换为小写的登录用户标识，例如 ibmwebspheremqmyuserid。
2. 重新启动客户机应用程序，或者使客户机应用程序关闭并重新打开所有 SSL 或 TLS 连接。
3. 通过发出以下命令，允许队列管理器上的匿名连接:

```
ALTER CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) SSLCAUTH(OPTIONAL)
```

结果

密钥存储库和通道已更改，如 第 181 页的图 19 中所示

下一步做什么

在通道的服务器端，通道状态屏幕上存在对等名称参数值指示客户机证书已流动。

通过发出一些 DISPLAY 命令验证任务是否已成功完成。 如果任务成功，那么生成的输出类似于以下示例中显示的输出:

从队列管理器 QM1，输入以下命令:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

生成的输出将类似于以下示例:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                     CURRENT
   SSLCERTI( )                             SSLPEER( )
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

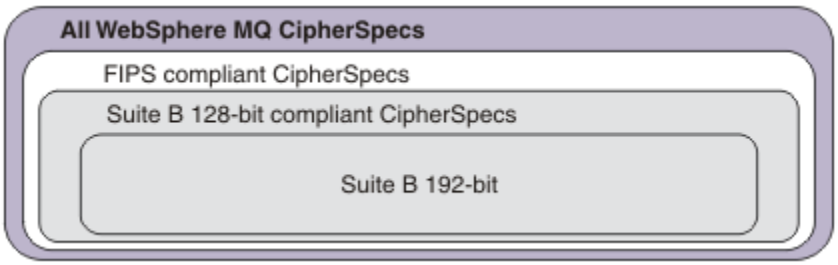
SSLCERTI 和 SSLPEER 字段为空，显示 C1 未发送证书。

指定 CipherSpec

通过在 **DEFINE CHANNEL** MQSC 命令或 **ALTER CHANNEL** MQSC 命令中使用 **SSLCIPH** 参数来指定 CipherSpec 。

可以与 IBM WebSphere MQ 配合使用的某些 CipherSpecs 符合 FIPS。 其他 (例如 NULL_MD5) 则不存在。 同样，某些符合 FIPS 的 CipherSpecs 也符合 Suite B ，但其他不符合。 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:



下表列出了可与 IBM WebSphere MQ SSL 和 TLS 支持一起使用的密码规范。 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。 除非由 CipherSpec 确定，否则 SSL 握手期间使用的密钥大小即是证书中存储的大小，如下表中所述。

CipherSpec 名称	使用的协议	MAC 算法	加密算法	加密位	FIPS[1]	套件 B 128 位	套件 B 192 位
NULL_MD5 [a]	SSL 3.0	MD5	无	0	否	否	否
NULL_SHA [a]	SSL 3.0	SHA-1	无	0	否	否	否
RC4_MD5_EXPORT [2] [a]	SSL 3.0	MD5	RC4	40	否	否	否
RC4_MD5_US [a]	SSL 3.0	MD5	RC4	128	否	否	否
RC4_SHA_US [a]	SSL 3.0	SHA-1	RC4	128	否	否	否
RC2_MD5_EXPORT [2] [a]	SSL 3.0	MD5	RC2	40	否	否	否
DES_SHA_EXPORT [2] [a]	SSL 3.0	SHA-1	DES	56	否	否	否
RC4_56_SHA_EXPORT1024 [3] [b]	SSL 3.0	SHA-1	RC4	56	否	否	否
DES_SHA_EXPORT1024 [3] [b]	SSL 3.0	SHA-1	DES	56	否	否	否
TLS_RSA_WITH_AES_128_CBC_SHA [a]	TLS 1.0	SHA-1	AES	128	Yes	否	否
TLS_RSA_WITH_AES_256_CBC_SHA [4] [a]	TLS 1.0	SHA-1	AES	256	Yes	否	否
TLS_RSA_WITH_DES_CBC_SHA [a]	TLS 1.0	SHA-1	DES	56	否 [5]	否	否
FIPS_WITH_DES_CBC_SHA [b]	SSL 3.0	SHA-1	DES	56	否 [6]	否	否
TLS_RSA_WITH_AES_128_GCM_SHA256 [b]	TLS 1.2	AEAD AES-128 GCM	AES	128	Yes	否	否
TLS_RSA_WITH_AES_256_GCM_SHA384 [b]	TLS 1.2	AEAD AES-256 GCM	AES	256	Yes	否	否
TLS_RSA_WITH_AES_128_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	128	Yes	否	否
TLS_RSA_WITH_AES_256_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	256	Yes	否	否

CipherSpec 名称	使用的协议	MAC 算法	加密算法	加密位	FIPS[1]	套件 B 128 位	套件 B 192 位
ECDHE_ECDSA_RC4_128_SHA256 [b]	TLS 1.2	SHA-1	RC4	128	否	否	否
ECDHE_RSA_RC4_128_SHA256 [b]	TLS 1.2	SHA_1	RC4	128	否	否	否
ECDHE_ECDSA_AES_128_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	128	Yes	否	否
ECDHE_ECDSA_AES_256_CBC_SHA384 [b]	TLS 1.2	SHA-384	AES	256	Yes	否	否
ECDHE_RSA_AES_128_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	128	Yes	否	否
ECDHE_RSA_AES_256_CBC_SHA384 [b]	TLS 1.2	SHA-384	AES	256	Yes	否	否
ECDHE_ECDSA_AES_128_GCM_SHA256 [b]	TLS 1.2	AEAD AES-128 GCM	AES	128	Yes	Yes	否
ECDHE_ECDSA_AES_256_GCM_SHA384 [b]	TLS 1.2	AEAD AES-256 GCM	AES	256	Yes	否	Yes
ECDHE_RSA_AES_128_GCM_SHA256 [b]	TLS 1.2	AEAD AES-128 GCM	AES	128	Yes	否	否
ECDHE_RSA_AES_256_GCM_SHA384 [b]	TLS 1.2	AEAD AES-256 GCM	AES	256	Yes	否	否
TLS_RSA_WITH_NULL_SHA256 [b]	TLS 1.2	SHA-256	无	0	否	否	否
ECDHE_RSA_NULL_SHA256 [b]	TLS 1.2	SHA-1	无	0	否	否	否
ECDHE_ECDSA_NULL_SHA256 [b]	TLS 1.2	SHA-1	无	0	否	否	否
TLS_RSA_WITH_NULL_NULL [b]	TLS 1.2	无	无	0	否	否	否
TLS_RSA_WITH_RC4_128_SHA256 [b]	TLS 1.2	SHA-1	RC4	128	否	否	否

注意:

1. 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。
2. 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。
3. 握手密钥大小是 1024 位。
4. 此 CipherSpec 不能用于保护从 WebSphere MQ 资源管理器到队列管理器的连接，除非将相应的不受限制策略文件应用于资源管理器所使用的 JRE。
5. 在 2007 年 5 月 19 日之前，该 CipherSpec 经 FIPS 140-2 认证。
6. 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。
7. 此 CipherSpec 可用于传输最多 32 GB 数据，超过此数据量之后，连接将因错误 AMQ9288 而终止。 要避免此错误，请避免使用三重 DES，或在使用此 CipherSpec 时启用密钥重置。

平台支持:

- a 在所有受支持的平台上可用。
- b 仅在 UNIX, Linux, and Windows 平台上可用。

相关概念

第 28 页的『IBM WebSphere MQ 中的数字证书和 CipherSpec 兼容性』

本主题通过概述 CipherSpecs 与 IBM WebSphere 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

不推荐使用的 CipherSpecs

您可以根据需要与 WebSphere MQ 配合使用的不推荐使用的 CipherSpecs 的列表。

请参阅 第 32 页的『IBM WebSphere MQ 中支持的 CipherSpec 值』，以获取有关如何启用不推荐的 CipherSpecs 的更多信息。

下表中列出了可用于 WebSphere MQ TLS 支持的不推荐的 CipherSpecs：

平台支持 第 186 页的『1』	CipherSpec 名称	使用的协议	数据完整性	加密算法	加密位	FIPS 第 186 页的『2』	套件 B	不推荐使用时更新
全部	DES_SHA_EXPORT 第 186 页的『3』	SSL 3.0	SHA-1	DES	56	否	否	7.5.0.6
Windows UNIX Linux	DES_SHA_EXPORT1024 第 186 页的『4』	SSL 3.0	SHA-1	DES	56	否	否	7.5.0.6
Windows UNIX Linux	FIPS_WITH_DES_CBC_SHA	SSL 3.0	SHA-1	DES	56	否 第 186 页的『6』	否	7.5.0.6
Windows UNIX Linux	FIPS_WITH_3DES_EDE_CBC_SHA	SSL 3.0	SHA-1	3DES	168	否 第 186 页的『7』	否	7.5.0.8
全部	NULL_MD5	SSL 3.0	MD5	None	0	否	否	7.5.0.6
全部	NULL_SHA	SSL 3.0	SHA-1	None	0	否	否	7.5.0.6
全部	RC2_MD5_EXPORT 第 186 页的『3』	SSL 3.0	MD5	RC2	40	否	否	7.5.0.7
全部	RC4_MD5_EXPORT 第 186 页的『3』	SSL 3.0	MD5	RC4	40	否	否	7.5.0.7
全部	RC4_MD5_US	SSL 3.0	MD5	RC4	128	否	否	7.5.0.7
全部	RC4_SHA_US	SSL 3.0	SHA-1	RC4	128	否	否	7.5.0.7
Windows UNIX Linux	RC4_56_SHA_EXPORT1024 第 186 页的『4』	SSL 3.0	SHA-1	RC4	56	否	否	7.5.0.7
全部	TRIPLE_DES_SHA_US	SSL 3.0	SHA-1	3DES	168	否	否	7.5.0.8
全部	TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	SHA-1	DES	56	否 第 186 页的『5』	否	7.5.0.6

平台支持 第 186 页的『1』	CipherSpec 名称	使用的协议	数据完整性	加密算法	加密位	FIPS 第 186 页的『2』	套件 B	不推荐使用时更新
Windows UNIX Linux	ECDHE_ECDSA_NULL_SHA256	TLS 1.2	SHA-1	None	0	否	否	7.5.0.6
Windows UNIX Linux	ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	SHA-1	RC4	128	否	否	7.5.0.7
Windows UNIX Linux	ECDHE_RSA_NULL_SHA256	TLS 1.2	SHA-1	None	0	否	否	7.5.0.6
Windows UNIX Linux	ECDHE_RSA_RC4_128_SHA256	TLS 1.2	SHA-1	RC4	128	否	否	7.5.0.7
Windows UNIX Linux	TLS_RSA_WITH_NULL_NULL	TLS 1.2	None	None	0	否	否	7.5.0.6
全部	TLS_RSA_WITH_NULL_SHA256	TLS 1.2	SHA-256	None	0	否	否	7.5.0.6
Windows UNIX Linux	TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	SHA-1	RC4	128	否	否	7.5.0.7
全部	TLS_RSA_WITH_3DES_EDE_CBC_SHA 第 186 页的『8』	TLS 1.0	SHA-1	3DES	168	Yes	否	7.5.0.8
Windows UNIX Linux	ECDHE_ECDSA_3DES_EDE_CBC_SHA256 第 186 页的『8』	TLS 1.2	SHA-1	3DES	168	Yes	否	7.5.0.8
Windows UNIX Linux	ECDHE_RSA_3DES_EDE_CBC_SHA256 第 186 页的『8』	TLS 1.2	SHA-1	3DES	168	Yes	否	7.5.0.8

平台支持 第 186 页的『1』	CipherSpec 名称	使用的协议	数据完整性	加密算法	加密位	FIPS 第 186 页的『2』	套件 B	不推荐使用时更新

注意:

1. 如果未指出特定平台，CipherSpec 将在所有平台上可用。
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. 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。
4. 握手密钥大小是 1024 位。
5. 在 2007 年 5 月 19 日之前，该 CipherSpec 经 FIPS 140-2 认证。
6. 在 2007 年 5 月 19 日之前，该 CipherSpec 经 FIPS 140-2 认证。 名称 FIPS_WITH_DES_CBC_SHA 是历史名称，表明此 CipherSpec 先前（但已不再）符合 FIPS 标准。 不推荐使用此 CipherSpec。
7. 名称 FIPS_WITH_3DES_EDE_CBC_SHA 是历史名称，表明此 CipherSpec 先前（但已不再）符合 FIPS 标准。 不推荐使用此 CipherSpec。
8. 此 CipherSpec 可用于传输最多 32 GB 数据，超过此数据量之后，连接将因错误 AMQ9288 而终止。 要避免此错误，请避免使用三重 DES，或在使用此 CipherSpec 时启用密钥重置。

使用 IBM WebSphere MQ Explorer 获取有关 CipherSpecs 的信息

您可以使用 IBM WebSphere MQ Explorer 来显示 CipherSpecs 的描述。

使用以下过程来获取有关 第 182 页的『指定 CipherSpec』中的 CipherSpecs 的信息:

1. 打开 **IBM WebSphere 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。

指定 CipherSpecs 的替代方法

对于操作系统提供 SSL 支持的平台，您的系统可能支持新的 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。

注: 本部分不适用于 UNIX， Linux 或 Windows 系统，因为 CipherSpecs 随 WebSphere MQ 产品一起提供，因此新的 CipherSpecs 在装运后不可用。

对于操作系统提供 SSL 支持的平台，您的系统可能支持 第 182 页的『指定 CipherSpec』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，规范 必须 对应于 SSL CipherSpec ，该规范有效且受系统正在运行的 SSL 版本支持。

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，请参阅相应的产品文档 (在 IBM i 产品文档中搜索 *cipher_spec*)。

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
表示十六进制值的双字符字符串。 十六进制代码对应于 SSL 协议中定义的值。

有关更多信息，请参阅 *z/OS Cryptographic Services System SSL Programming* SC24-5901 的 API 参考章节中对 gsk_environment_open () 的描述，其中列出了所有受支持的 SSL V3.0 和 TLS V1.0 密码规范，格式为 2 位十六进制代码。

WebSphere MQ 集群的注意事项

对于 WebSphere MQ 集群，在 第 182 页的『指定 CipherSpec』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 208 页的『SSL 和集群』。

为 IBM WebSphere MQ MQI 客户机指定 CipherSpec

您有三个选项可用于为 IBM WebSphere MQ MQI 客户机指定 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 WebSphere MQ classes for Java 和 IBM WebSphere MQ classes for JMS 指定 CipherSuite

IBM WebSphere MQ classes for Java 和 IBM WebSphere MQ classes for JMS 指定的 CipherSuites 与其他平台不同。

有关使用 IBM WebSphere MQ classes for Java 指定 CipherSuite 的信息，请参阅 安全套接字层 (SSL) 支持。

有关将 CipherSuite 与 IBM WebSphere MQ JMS 类配合使用的信息，请参阅 将安全套接字层 (SSL) 与 WebSphere MQ JMS 类配合使用。

重置 SSL 和 TLS 密钥

IBM WebSphere 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，或者在通道处于空闲状态一段时间后，将重置密钥。

密钥重置值始终由 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。

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 KeyReset 计数字段
2. 通过使用环境变量 MQSSLRESET
3. 通过在 MQI 客户机配置文件中设置 SSLKeyResetCount 属性

这些变量可以设置为 0 到 999 999 999 999 范围内的整数，表示在重新协商 SSL 或 TLS 密钥之前在 SSL 或 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 SSL 或 TLS 密钥。 如果指定 1 字节到 32 KB 范围内的 SSL 或 TLS 密钥重置计数，那么 SSL 或 TLS 通道将使用 32 KB 的密钥重置计数。 这是为了避免对小型 SSL 或 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。

有关 MQSCO 结构的完整详细信息，请参阅 KeyResetCount (MQLONG)。 有关 MQSSLRESET 的完整详细信息，请参阅 MQSSLRESET。 有关在客户机配置文件中使用 SSL 或 TLS 的更多信息，请参阅 客户机配置文件的 SSL 节。

Java

对于 IBM WebSphere 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 字段或环境属性 MQC.SSL_RESET_COUNT_PROPERTY 的值表示 WebSphere MQ classes for Java 客户机代码在重新协商密钥之前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。 字节数还包括 WebSphere 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 WebSphere 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 WebSphere 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 SSL 或 TLS 的 MQI 通道创建连接，该通道具有在 4 MB 数据流动后重新协商的密钥，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 SSL 或 TLS 对话中发送和接收的未加密字节数。

有关在 IBM WebSphere MQ classes for .NET 中使用对象属性的信息，请参阅 获取和设置属性值。

XMS .NET

对于 XMS .NET 非受管客户机，请参阅 与 IBM WebSphere MQ 队列管理器的安全连接。

相关参考

ALTER QMGR

显示队列管理器

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 188 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。

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 SSL 或 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 WebSphere MQ Advanced 消息服务 (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。

使用 SSL 或 TLS 连接两个队列管理器

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。 您还必须获取和管理数字证书。 在测试系统上，可以使用自签名证书或本地认证中心 (CA) 颁发的证书。 在生产系统上，请勿使用自签名证书。 有关更多信息，请参阅 ../zs14140_.dita。

有关创建和管理证书的完整信息，请参阅 第 95 页的『在 UNIX, Linux, and Windows 系统上使用 SSL 或 TLS』。

此主题集合介绍了设置 SSL 通信所涉及的任务，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 WebSphere MQ 实现， SSL 或 TLS 服务器始终从客户机请求证书。

注意:

1. 在此上下文中， SSL 客户机是指启动握手的连接。
2. 请参阅 词汇表 以获取更多详细信息。

在 UNIX， Linux 和 Windows 系统上，仅当 SSL 或 TLS 客户机具有以正确的 WebSphere MQ 格式标注的证书时，才会发送证书，后跟 `ibmwebspheremq`，然后将队列管理器的名称更改为小写。 例如，对于 `QM1`，`ibmwebspheremqqm1`。

WebSphere 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，那么仅当充当 SSL 或 TLS 服务器的通道的末尾定义了 SSLCAUTH 参数设置为 REQUIRED 或 SSLPEER 参数值时，认证才会失败。 有关匿名连接队列管理器的更多信息，即，当 SSL 或 TLS 客户机未发送证书时，请参阅 第 176 页的『使用单向认证连接两个队列管理器』。

数字证书标签，了解需求

设置 SSL 和 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。

关于此任务

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 主题专有名称 或 主题公共名称 字段分开。 请注意， 主题专有名称 和 主题公共名称 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关联的标签。

如何使用证书标签?

IBM WebSphere MQ 使用证书标签来查找在 SSL 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。

证书标签遵循命名约定; 您需要确保使用与所使用平台对应的正确标签命名约定。

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

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 WebSphere MQ 实现， SSL 或 TLS 服务器始终从客户机请求证书，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。

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，认证将失败。

有关使用单向认证 (即，当 SSL 或 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 第 176 页的『使用单向认证连接两个队列管理器』。

和 *UNIX, Linux, and Windows* 系统

关于此任务

在， UNIX, Linux, and Windows 系统上， SSL 或 TLS 服务器向客户机发送证书，仅当服务器找到以正确 IBM WebSphere MQ 格式标注的证书时。 在这些系统上，正确的格式为 `ibmwebspheremq`，后跟队列管理器的名称更改为小写。

例如，对于名为 `QM1` 的队列管理器，证书标签要求为:

```
ibmwebspheremqqm1
```

如果在队列管理器的密钥存储库中找不到与正确大小写和格式的所需标签匹配的证书，那么将发生错误，并且 SSL 或 TLS 握手将失败。

IBM WebSphere MQ 客户机

关于此任务

从 IBM WebSphere MQ 客户机应用程序连接时，仅当 SSL 或 TLS 客户机具有一个具有 `ibmwebspheremq` 格式标签的证书，后跟运行客户机应用程序进程的用户的用户名时， SSL 或 TLS 客户机才会发送证书。

例如，对于用户名 `wasadmin`，证书标签需求如下所示，折叠为小写:

```
   ibmwebspheremqwasadmin
```

以上标签要求适用于 Message Service Client for C ，或 C++ 和 .NET。

IBM WebSphere MQ Java 或 IBM WebSphere MQ JMS 客户机

关于此任务

IBM WebSphere MQ Java 或 IBM WebSphere 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 SSL 或 TLS 握手期间选择个人证书，因此不受证书标签要求的约束。

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 WebSphere 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，可以通过设置来启用调试

```
javax.net.debug=ssl
```

在 JVM 环境中。

您可以在命令行上使用 `-Djavax.net.debug=ssl` ，也可以通过配置在应用程序中设置变量。

相关概念

第 111 页的『将个人证书导入到 UNIX, Linux, and Windows 系统上的密钥存储库中』
遵循此过程以导入个人证书

使用自签名证书对两个队列管理器进行相互认证

遵循以下样本指示信息，以使用自签名 SSL 或 TLS 证书在两个队列管理器之间实现相互认证。

关于此任务

方案:

- 您有两个需要安全通信的队列管理器 QM1 和 QM2。 您需要在 QM1 和 QM2 之间执行相互认证。
- 您已决定使用自签名证书测试安全通信。

生成的配置如下所示:

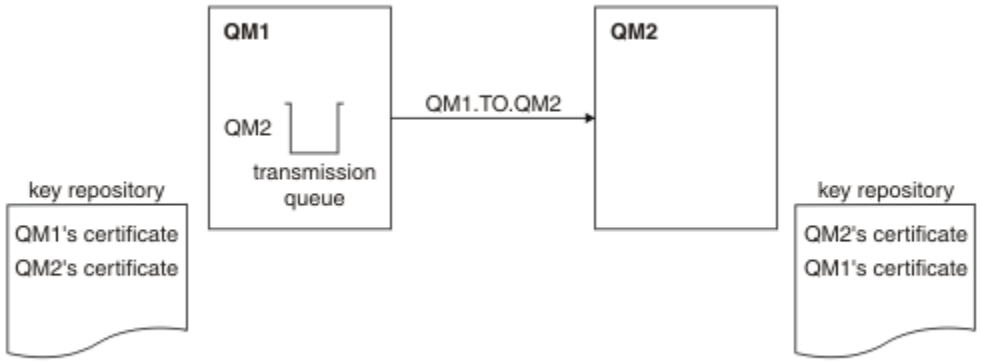


图 *20:* 由此任务生成的配置

在 第 172 页的图 14 中， QM1 的密钥存储库包含 QM1 的证书和来自 QM2 的公用证书。 QM2 的密钥存储库包含 QM2 的证书以及来自 QM1 的公用证书。

过程

1. 根据操作系统，在每个队列管理器上准备密钥存储库:

- 在 UNIX， Linux 和 Windows 系统上。

2. 为每个队列管理器创建自签名证书:
   - 在 UNIX， Linux 和 Windows 系统上。
3. 抽取每个证书的副本:
   - 在 UNIX， Linux 和 Windows 系统上。
4. 使用诸如 FTP 中所述。
5. 将合作伙伴证书添加到每个队列管理器的密钥存储库:
   - 在 UNIX， Linux 和 Windows 系统上。
6. 在 QM1 上，通过发出类似以下示例的命令来定义发送方通道和关联的传输队列:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QM1.MACH.COM) XMITQ(QM2)
SSLCIPH(RC4_MD5_US) DESCR('Sender channel using SSL from QM1 to QM2')

DEFINE QLOCAL(QM2) USAGE(XMITQ)
```

   此示例使用 CipherSpec RC4_MD5。 通道两端的 CipherSpecs 必须相同。
7. 在 QM2 上，通过发出类似于以下示例的命令来定义接收方通道:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(RC4_MD5_US)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from QM1 to QM2')
```

   通道必须与您在步骤 6 中定义的发送方通道同名，并使用相同的 CipherSpec。
8. 启动通道中所述。

结果

创建密钥存储库和通道，如 第 172 页的图 14 中所述

下一步做什么

使用 DISPLAY 命令检查任务是否已成功完成。 如果任务成功，那么生成的输出类似于以下示例中显示的输出。

从队列管理器 QM1，输入以下命令:

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

生成的输出类似于以下示例:

```
DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(QM1.TO.QM2)                     CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QM2)
   SSLCERTI("CN=QM2,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5E:02,CN=QM2,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QM2)
```

从队列管理器 QM2 中，输入以下命令:

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

生成的输出类似于以下示例:

```
DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(QM2.TO.QM1)                     CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QM1)
   SSLCERTI("CN=QM1,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
```

```
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QM1,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                          SUBSTATE(RECEIVE)
   XMITQ( )
```

在每种情况下， SSLPEER 的值必须与步骤 2 中创建的伙伴证书中的 DN 的值匹配。 颁发者名称与同级名称匹配，因为证书是自签名的。

SSLPEER 是可选的。 如果指定了此值，那么必须设置其值，以便允许使用伙伴证书 (在步骤 2 中创建) 中的 DN。 有关使用 SSLPEER 的更多信息，请参阅 WebSphere MQ 规则以获取 SSLPEER 值 。

使用 CA 签名的证书对两个队列管理器进行相互认证

遵循以下样本指示信息以使用 CA 签署的 SSL 或 TLS 证书在两个队列管理器之间实现相互认证。

关于此任务

方案:

- 您有两个名为 QMA 和 QMB 的队列管理器，它们需要安全地进行通信。 您需要在 QMA 和 QMB 之间执行相互认证。
- 将来您计划在生产环境中使用此网络，因此您决定从一开始就使用 CA 签署的证书。

生成的配置如下所示:

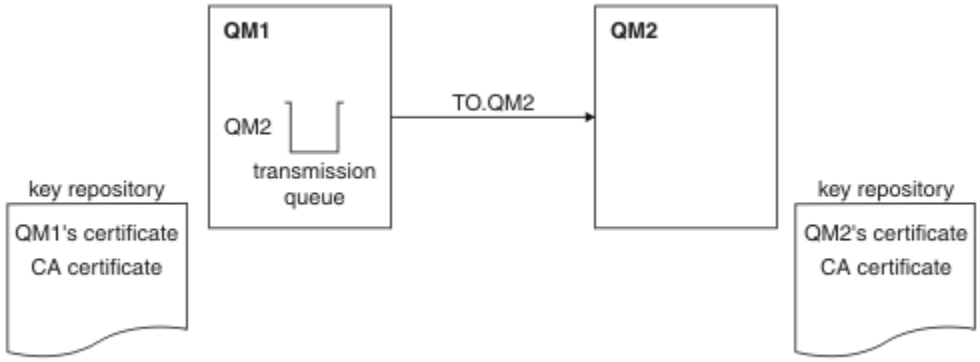


图 *21:* 由此任务生成的配置

在 第 174 页的图 15 中， QMA 的密钥存储库包含 QMA 的证书和 CA 证书。 QMB 的密钥存储库包含 QMB 的证书和 CA 证书。 在此示例中， QMA 的证书和 QMB 的证书都由同一 CA 发放。 如果 QMA 的证书和 QMB 的证书是由不同的 CA 发放的，那么 QMA 和 QMB 的密钥存储库必须同时包含两个 CA 证书。

过程

1. 根据操作系统，在每个队列管理器上准备密钥存储库:
   - 在 UNIX， Linux 和 Windows 系统上。
2. 请求每个队列管理器的 CA 签名证书。
   您可以对两个队列管理器使用不同的 CA。
   - 在 UNIX， Linux 和 Windows 系统上。
3. 将认证中心证书添加到每个队列管理器的密钥存储库:
   如果队列管理器使用不同的认证中心，那么必须将每个认证中心的 CA 证书添加到两个密钥存储库。
   - 在 UNIX， Linux 和 Windows 系统上。
4. 将 CA 签名的证书添加到每个队列管理器的密钥存储库:
   - 在 UNIX， Linux 和 Windows 系统上。

5. 在 QMA 上，通过发出类似于以下示例的命令来定义发送方通道和关联的传输队列:

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QMB.MACH.COM) XMITQ(QMB)
SSLCIPH(RC2_MD5_EXPORT) DESCR('Sender channel using SSL from QMA to QMB')

DEFINE QLOCAL(QMB) USAGE(XMITQ)
```

   此示例使用 CipherSpec RC4_MD5。 通道两端的 CipherSpecs 必须相同。
6. 在 QMB 上，通过发出类似于以下示例的命令来定义接收方通道:

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(RC2_MD5_EXPORT)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL to QMB')
```

   通道必须与您在步骤 6 中定义的发送方通道同名，并使用相同的 CipherSpec。
7. 启动通道:

结果

将创建密钥存储库和通道，如 第 174 页的图 15 中所述。

下一步做什么

使用 DISPLAY 命令检查任务是否已成功完成。 如果任务成功，那么生成的输出类似于以下示例中显示的输出。

从队列管理器 QMA ，输入以下命令:

```
DISPLAY CHS(TO.QMB) SSLPEER SSLCERTI
```

生成的输出类似于以下示例:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QMB)                         CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QMB)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMB,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QMB)
```

从队列管理器 QMB 中，输入以下命令:

```
DISPLAY CHS(TO.QMB) SSLPEER SSLCERTI
```

生成的输出类似于以下示例:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QMB)                         CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QMA)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMA,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
   XMITQ( )
```

在每种情况下， SSLPEER 的值必须与步骤 2 中创建的伙伴证书中的专有名称 (DN) 的值匹配。 签发者名称与对步骤 4 中添加的个人证书进行签名的 CA 证书的主体集 DN 相匹配。

使用单向认证连接两个队列管理器

遵循以下样本指示信息来修改具有相互认证的系统，以允许队列管理器使用单向认证连接到另一个系统; 即，当 SSL 或 TLS 客户机未发送证书时。

关于此任务

方案:

- 您的两个队列管理器 (QM1 和 QM2) 已按 第 174 页的『使用 CA 签名的证书对两个队列管理器进行相互认证』中的方式进行设置。
- 您希望更改 QM1 ，以便它使用单向认证连接到 QM2。

生成的配置如下所示:

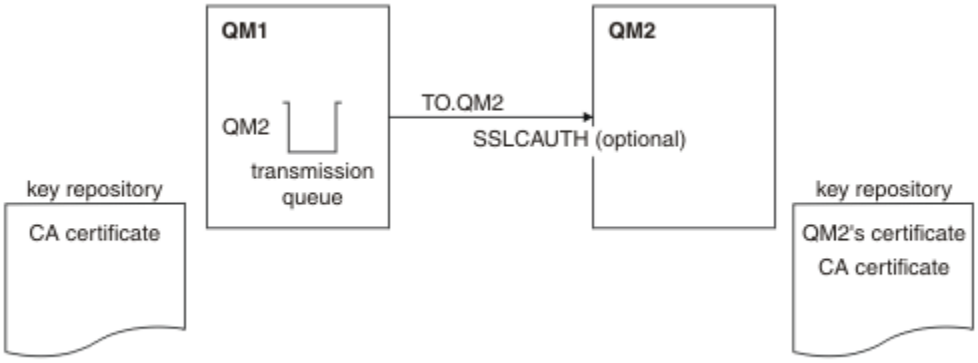


图 *22:* 允许单向认证的队列管理器

过程

1. 根据操作系统，从其密钥存储库中除去 QM1 的个人证书:
   - 在 UNIX， Linux 和 Windows 系统上。 该证书标注如下:
     - `ibmwebspheremq` 后跟已折叠为小写的队列管理器的名称。 例如，对于 `QM1` ，`ibmwebspheremqqm1`。
2. 可选： 在 QM1 上，如果先前已运行任何 SSL 或 TLS 通道，请刷新 SSL 或 TLS 环境中所述。
3. 在接收方中所述。

结果

密钥存储库和通道已更改，如 第 176 页的图 16 中所示

下一步做什么

如果发送方通道正在运行，并且您在步骤 2 中发出了 REFRESH SECURITY TYPE (SSL) 命令，那么通道将自动重新启动。 如果发送方通道未运行，请将其启动。

在通道的服务器端，通道状态屏幕上存在对等名称参数值指示客户机证书已流动。

通过发出一些 DISPLAY 命令验证任务是否已成功完成。 如果任务成功，那么生成的输出类似于以下示例中显示的输出:

从 QM1 队列管理器中，输入以下命令:

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

生成的输出将类似于以下示例:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(TO.QMB) SSLPEER
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QM2)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMB,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QM2)
```

从 QM2 队列管理器中，输入以下命令:

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

生成的输出将类似于以下示例:

```
DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QMA)                            SSLCERTI( )
   SSLPEER( )                              STATUS(RUNNING)
   SUBSTATE(RECEIVE)                       XMITQ( )
```

在 QM2 上， SSLPEER 字段为空，显示 QM1 未发送证书。 在 QM1 上， SSLPEER 的值与 QM2 个人证书中的 DN 的值相匹配。

安全地将客户机连接到队列管理器

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。 您还必须获取和管理数字证书。 在测试系统上，可以使用自签名证书或本地认证中心 (CA) 颁发的证书。 在生产系统上，请勿使用自签名证书。 有关更多信息，请参阅 ../zs14140_.dita。

有关创建和管理证书的完整信息，请参阅 第 95 页的『在 UNIX, Linux, and Windows 系统上使用 SSL 或 TLS』。

此主题集合介绍了设置 SSL 通信所涉及的任务，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 WebSphere MQ 实现， SSL 或 TLS 服务器始终从客户机请求证书。

在 UNIX, Linux, and Windows 系统上，仅当 SSL 或 TLS 客户机具有以正确的 WebSphere MQ 格式标注的证书时， SSL 或 TLS 客户机才会发送证书，该格式为 `ibmwebspheremq` 后跟更改为小写的登录用户标识，例如 `ibmwebspheremqmyuserid`。

WebSphere 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，那么仅当充当 SSL 或 TLS 服务器的通道的末尾定义了 SSLCAUTH 参数设置为 REQUIRED 或 SSLPEER 参数值时，认证才会失败。 有关以匿名方式连接队列管理器的更多信息，请参阅 第 180 页的『以匿名方式将客户机连接到队列管理器』。

使用自签名证书对客户机和队列管理器进行相互认证

遵循以下样本指示信息，通过使用自签名 SSL 或 TLS 证书来实现客户机与队列管理器之间的相互认证。

关于此任务

方案:

- 您具有需要安全通信的客户机 C1 和队列管理器 QM1。 您需要在 C1 和 QM1 之间执行相互认证。
- 您已决定使用自签名证书来测试安全通信。

IBM i 上的 DCM 不支持自签名证书，因此此任务不适用于 IBM i 系统。

生成的配置如下所示:

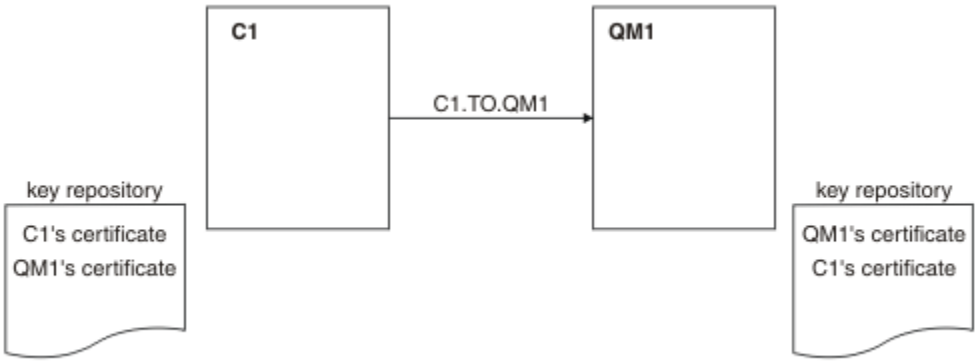


图 *23:* 由此任务生成的配置

在 第 178 页的图 17 中， QM1 的密钥存储库包含 QM1 的证书和来自 C1 的公用证书。 C1 的密钥存储库包含 C1 的证书以及来自 QM1 的公用证书。

过程

1. 根据操作系统，在客户机和队列管理器上准备密钥存储库:
   - 在 UNIX， Linux 和 Windows 系统上。
2. 为客户机和队列管理器创建自签名证书:
   - 在 UNIX， Linux 和 Windows 系统上。
3. 抽取每个证书的副本:
   - 在 UNIX， Linux 和 Windows 系统上。
4. 使用诸如 FTP 中所述。
5. 将伙伴证书添加到客户机和队列管理器的密钥存储库:
   - 在 UNIX， Linux 和 Windows 系统上。
6. 在队列管理器上发出命令 REFRESH SECURITY TYPE (SSL)。
7. 通过以下任一方式定义客户机连接通道:
   - 将 MQCONNX 调用与 C1 上的 MQSCO 结构配合使用，如 在 WebSphere MQ MQI 客户机上创建客户机连接通道中所述。
   - 使用客户机通道定义表，如 在服务器上创建服务器连接和客户机连接定义 中所述。
8. 在 QM1 上，通过发出类似于以下示例的命令来定义服务器连接通道:

   ```
   DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC4_MD5_US)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
   ```

   通道必须与您在步骤 6 中定义的客户机连接通道同名，并使用相同的 CipherSpec。

结果

创建密钥存储库和通道，如 第 178 页的图 17 中所述

下一步做什么

使用 DISPLAY 命令检查任务是否已成功完成。 如果任务成功，那么生成的输出与以下示例中显示的输出相似。

从队列管理器 QM1，输入以下命令:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

生成的输出类似于以下示例:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                       CURRENT
   SSLCERTI("CN=QM1,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5E:02,CN=QM2,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                           SUBSTATE(RECEIVE)
```

可选择设置通道定义的 SSLPEER 过滤器属性。 如果设置了通道定义 SSLPEER ，那么其值必须与步骤 2 中创建的伙伴证书中的主体集 DN 相匹配。 成功连接后， DISPLAY CHSTATUS 输出中的 SSLPEER 字段将显示远程客户机证书的主题 DN。

使用 CA 签名的证书对客户机和队列管理器进行相互认证

遵循以下样本指示信息，通过使用 CA 签署的 SSL 或 TLS 证书来实现客户机与队列管理器之间的相互认证。

关于此任务

方案:

- 您具有需要安全通信的客户机 C1 和队列管理器 QM1。 您需要在 C1 和 QM1 之间执行相互认证。
- 将来您计划在生产环境中使用此网络，因此您决定从一开始就使用 CA 签署的证书。

生成的配置如下所示:

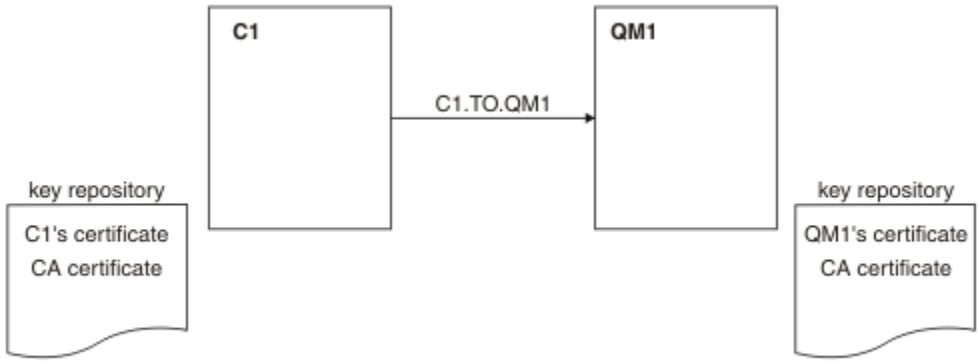


图 *24:* 由此任务生成的配置

在 第 179 页的图 18 中， C1 的密钥存储库包含 C1 的证书和 CA 证书。 QM1 的密钥存储库包含 QM1 的证书和 CA 证书。 在此示例中， C1 的证书和 QM1 的证书都由同一 CA 发放。 如果 C1 的证书和 QM1 的证书是由不同的 CA 发放的，那么 C1 和 QM1 的密钥存储库必须同时包含两个 CA 证书。

过程

1. 根据操作系统，在客户机和队列管理器上准备密钥存储库:
   - 在 UNIX， Linux 和 Windows 系统上。

2. 请求客户机和队列管理器的 CA 签名证书。
   您可以对客户机和队列管理器使用不同的 CA。
   - 在 UNIX， Linux 和 Windows 系统上。
3. 将认证中心证书添加到客户机和队列管理器的密钥存储库。
   如果客户机和队列管理器使用不同的认证中心，那么必须将每个认证中心的 CA 证书添加到两个密钥存储库。
   - 在 UNIX， Linux 和 Windows 系统上。
4. 将 CA 签名的证书添加到客户机和队列管理器的密钥存储库:
   - 在 UNIX， Linux 和 Windows 系统上。
5. 通过以下任一方式定义客户机连接通道:
   - 将 MQCONNX 调用与 C1 上的 MQSCO 结构配合使用，如 在 WebSphere MQ MQI 客户机上创建客户机连接通道中所述。
   - 使用客户机通道定义表，如 在服务器上创建服务器连接和客户机连接定义 中所述。
6. 在 QM1 上，通过发出类似于以下示例的命令来定义服务器连接通道:

   ```
   DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC2_MD5_EXPORT)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
   ```

   通道必须与您在步骤 6 中定义的客户机连接通道同名，并使用相同的 CipherSpec。

结果

将创建密钥存储库和通道，如 第 179 页的图 18 中所述。

下一步做什么

使用 DISPLAY 命令检查任务是否已成功完成。 如果任务成功，那么生成的输出类似于以下示例中显示的输出。

从队列管理器 QM1 中，输入以下命令:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
```

生成的输出类似于以下示例:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                     CURRENT
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMA,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

DISPLAY CHSTATUS 输出中的 SSLPEER 字段显示在步骤 2 中创建的远程客户机证书的主题 DN。 签发者名称与对步骤 4 中添加的个人证书进行签名的 CA 证书的主体集 DN 相匹配。

以匿名方式将客户机连接到队列管理器

遵循以下样本指示信息来修改具有相互认证的系统，以允许队列管理器以匿名方式连接到另一个队列管理器。

关于此任务

方案:

- 您的队列管理器和客户机 (QM1 和 C1) 已设置为与 第 179 页的『使用 CA 签名的证书对客户机和队列管理器进行相互认证』中一样。

- 您希望更改 C1 ，以便它以匿名方式连接到 QM1。

生成的配置如下所示:

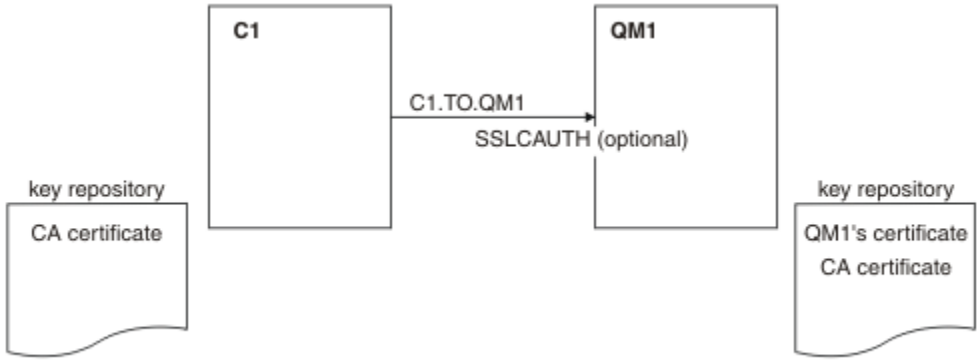


图 *25:* 允许匿名连接的客户机和队列管理器

过程

1. 根据操作系统，从 C1 的密钥存储库中除去个人证书:
   - 在 UNIX， Linux 和 Windows 系统上。 该证书标注如下:
     - ibmwebspheremq 后跟转换为小写的登录用户标识，例如 ibmwebspheremqmyuserid。
2. 重新启动客户机应用程序，或者使客户机应用程序关闭并重新打开所有 SSL 或 TLS 连接。
3. 通过发出以下命令，允许队列管理器上的匿名连接:

```
ALTER CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) SSLCAUTH(OPTIONAL)
```

结果

密钥存储库和通道已更改，如 第 181 页的图 19 中所示

下一步做什么

在通道的服务器端，通道状态屏幕上存在对等名称参数值指示客户机证书已流动。

通过发出一些 DISPLAY 命令验证任务是否已成功完成。 如果任务成功，那么生成的输出类似于以下示例中显示的输出:

从队列管理器 QM1，输入以下命令:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

生成的输出将类似于以下示例:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                     CURRENT
   SSLCERTI( )                             SSLPEER( )
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

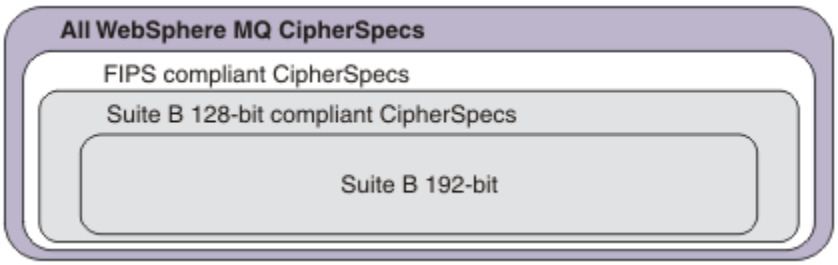
SSLCERTI 和 SSLPEER 字段为空，显示 C1 未发送证书。

指定 CipherSpec

通过在 **DEFINE CHANNEL** MQSC 命令或 **ALTER CHANNEL** MQSC 命令中使用 **SSLCIPH** 参数来指定 CipherSpec 。

可以与 IBM WebSphere MQ 配合使用的某些 CipherSpecs 符合 FIPS。 其他 (例如 NULL_MD5) 则不存在。 同样，某些符合 FIPS 的 CipherSpecs 也符合 Suite B ，但其他不符合。 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:



下表列出了可与 IBM WebSphere MQ SSL 和 TLS 支持一起使用的密码规范。 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。 除非由 CipherSpec 确定，否则 SSL 握手期间使用的密钥大小即是证书中存储的大小，如下表中所述。

CipherSpec 名称	使用的协议	MAC 算法	加密算法	加密位	FIPS[1]	套件 B 128 位	套件 B 192 位
NULL_MD5 [a]	SSL 3.0	MD5	无	0	否	否	否
NULL_SHA [a]	SSL 3.0	SHA-1	无	0	否	否	否
RC4_MD5_EXPORT [2] [a]	SSL 3.0	MD5	RC4	40	否	否	否
RC4_MD5_US [a]	SSL 3.0	MD5	RC4	128	否	否	否
RC4_SHA_US [a]	SSL 3.0	SHA-1	RC4	128	否	否	否
RC2_MD5_EXPORT [2] [a]	SSL 3.0	MD5	RC2	40	否	否	否
DES_SHA_EXPORT [2] [a]	SSL 3.0	SHA-1	DES	56	否	否	否
RC4_56_SHA_EXPORT1024 [3] [b]	SSL 3.0	SHA-1	RC4	56	否	否	否
DES_SHA_EXPORT1024 [3] [b]	SSL 3.0	SHA-1	DES	56	否	否	否
TLS_RSA_WITH_AES_128_CBC_SHA [a]	TLS 1.0	SHA-1	AES	128	Yes	否	否
TLS_RSA_WITH_AES_256_CBC_SHA [4] [a]	TLS 1.0	SHA-1	AES	256	Yes	否	否
TLS_RSA_WITH_DES_CBC_SHA [a]	TLS 1.0	SHA-1	DES	56	否 [5]	否	否
FIPS_WITH_DES_CBC_SHA [b]	SSL 3.0	SHA-1	DES	56	否 [6]	否	否
TLS_RSA_WITH_AES_128_GCM_SHA256 [b]	TLS 1.2	AEAD AES-128 GCM	AES	128	Yes	否	否
TLS_RSA_WITH_AES_256_GCM_SHA384 [b]	TLS 1.2	AEAD AES-256 GCM	AES	256	Yes	否	否
TLS_RSA_WITH_AES_128_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	128	Yes	否	否
TLS_RSA_WITH_AES_256_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	256	Yes	否	否

CipherSpec 名称	使用的协议	MAC 算法	加密算法	加密位	FIPS[1]	套件 B 128 位	套件 B 192 位
ECDHE_ECDSA_RC4_128_SHA256 [b]	TLS 1.2	SHA-1	RC4	128	否	否	否
ECDHE_RSA_RC4_128_SHA256 [b]	TLS 1.2	SHA_1	RC4	128	否	否	否
ECDHE_ECDSA_AES_128_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	128	Yes	否	否
ECDHE_ECDSA_AES_256_CBC_SHA384 [b]	TLS 1.2	SHA-384	AES	256	Yes	否	否
ECDHE_RSA_AES_128_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	128	Yes	否	否
ECDHE_RSA_AES_256_CBC_SHA384 [b]	TLS 1.2	SHA-384	AES	256	Yes	否	否
ECDHE_ECDSA_AES_128_GCM_SHA256 [b]	TLS 1.2	AEAD AES-128 GCM	AES	128	Yes	Yes	否
ECDHE_ECDSA_AES_256_GCM_SHA384 [b]	TLS 1.2	AEAD AES-256 GCM	AES	256	Yes	否	Yes
ECDHE_RSA_AES_128_GCM_SHA256 [b]	TLS 1.2	AEAD AES-128 GCM	AES	128	Yes	否	否
ECDHE_RSA_AES_256_GCM_SHA384 [b]	TLS 1.2	AEAD AES-256 GCM	AES	256	Yes	否	否
TLS_RSA_WITH_NULL_SHA256 [b]	TLS 1.2	SHA-256	无	0	否	否	否
ECDHE_RSA_NULL_SHA256 [b]	TLS 1.2	SHA-1	无	0	否	否	否
ECDHE_ECDSA_NULL_SHA256 [b]	TLS 1.2	SHA-1	无	0	否	否	否
TLS_RSA_WITH_NULL_NULL [b]	TLS 1.2	无	无	0	否	否	否
TLS_RSA_WITH_RC4_128_SHA256 [b]	TLS 1.2	SHA-1	RC4	128	否	否	否

注意:

1. 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。
2. 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。
3. 握手密钥大小是 1024 位。
4. 此 CipherSpec 不能用于保护从 WebSphere MQ 资源管理器到队列管理器的连接，除非将相应的不受限制策略文件应用于资源管理器所使用的 JRE。
5. 在 2007 年 5 月 19 日之前，该 CipherSpec 经 FIPS 140-2 认证。
6. 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。
7. 此 CipherSpec 可用于传输最多 32 GB 数据，超过此数据量之后，连接将因错误 AMQ9288 而终止。 要避免此错误，请避免使用三重 DES，或在使用此 CipherSpec 时启用密钥重置。

平台支持:

- a 在所有受支持的平台上可用。
- b 仅在 UNIX, Linux, and Windows 平台上可用。

相关概念

第 28 页的『IBM WebSphere MQ 中的数字证书和 CipherSpec 兼容性』

本主题通过概述 CipherSpecs 与 IBM WebSphere 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

使用 IBM WebSphere MQ Explorer 获取有关 CipherSpecs 的信息

您可以使用 IBM WebSphere MQ Explorer 来显示 CipherSpecs 的描述。

使用以下过程来获取有关 第 182 页的『指定 CipherSpec』中的 CipherSpecs 的信息:

1. 打开 **IBM WebSphere 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。

指定 CipherSpecs 的替代方法

对于操作系统提供 SSL 支持的平台，您的系统可能支持新的 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。

注: 本部分不适用于 UNIX， Linux 或 Windows 系统，因为 CipherSpecs 随 WebSphere MQ 产品一起提供，因此新的 CipherSpecs 在装运后不可用。

对于操作系统提供 SSL 支持的平台，您的系统可能支持 第 182 页的『指定 CipherSpec』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，规范 必须 对应于 SSL CipherSpec ，该规范有效且受系统正在运行的 SSL 版本支持。

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，请参阅相应的产品文档 (在 IBM i 产品文档中搜索 *cipher_spec*)。

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

表示十六进制值的双字符字符串。 十六进制代码对应于 SSL 协议中定义的值。

有关更多信息，请参阅 *z/OS Cryptographic Services System SSL Programming* SC24-5901 的 API 参考章节中对 gsk_environment_open () 的描述，其中列出了所有受支持的 SSL V3.0 和 TLS V1.0 密码规范，格式为 2 位十六进制代码。

WebSphere MQ 集群的注意事项

对于 WebSphere MQ 集群，在 第 182 页的『指定 CipherSpec』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 208 页的『SSL 和集群』。

为 IBM WebSphere MQ MQI 客户机指定 CipherSpec

您有三个选项可用于为 IBM WebSphere MQ MQI 客户机指定 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 WebSphere MQ classes for Java 和 IBM WebSphere MQ classes for JMS 指定 CipherSuite

IBM WebSphere MQ classes for Java 和 IBM WebSphere MQ classes for JMS 指定的 CipherSuites 与其他平台不同。

有关使用 IBM WebSphere MQ classes for Java 指定 CipherSuite 的信息，请参阅 安全套接字层 (SSL) 支持。

有关将 CipherSuite 与 IBM WebSphere MQ JMS 类配合使用的信息，请参阅 将安全套接字层 (SSL) 与 WebSphere MQ JMS 类配合使用。

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 WebSphere 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 SSL 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。 通过在 z/OS 上使用 RACF 或在其他平台上使用 OAM 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 WebSphere 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：

   - WebSphere MQ for z/OS 上的 RACF 或其他外部安全性管理器
   - 其他平台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。 在 z/OS 上，检查可能涉及使用从网络接收的用户标识以及从 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。 在 z/OS 上，检查可能涉及使用从网络接收的用户标识和/或派生自 MCAUSER 的用户标识。 如果链接可信且已认证，请使用此选项。

   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。

   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

在您的平台上，授权访问以连接到队列管理器并将其放入该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

- 对于 UNIX， Linux 和 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
```

  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。

  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 161 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。

相关信息

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，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 SSL/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。

Windows UNIX Linux **Windows， UNIX and Linux 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。

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)
```

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它在集群中。 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 和集群

为集群配置 SSL 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 SSL ，那么必须在所有使用该通道进行通信的队列管理器上配置 SSL。

有关 SSL 的更多信息，请参阅 WebSphere MQ 支持 SSL 和 TLS。 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 WebSphere 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 SSL 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 SSL 支持。 此 SSL 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 WebSphere 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=WebSphere 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 字符串 RC4_SHA_US 在 IBM i 上具有值 05 ，但在 UNIX， Linux 或 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 SSL 支持实现此 CipherSpec 的平台上，并且可以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 RC4_MD5_US)。

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

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 WebSphere 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。

将集群中的所有队列管理器升级到 WebSphere MQ V6 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 SSL 从每个证书和密钥中工作。

关于此任务

一次更改一个 CLUSRCVR ，并允许更改流经集群，然后再更改下一个。 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. 按您喜欢的任何顺序将 CLUSRCVR 通道切换到 SSL。

   这些更改在未更改为 SSL 的通道上以相反方向流动。
2. 将所有手动 CLUSSDR 通道切换到 SSL。

   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，当集群正在运行中时，使用 **REFRESH CLUSTER** 命令可能会破坏该集群，并且将在 27 天的时间间隔之后，集群对象才会再次自动向所有相关队列管理器发送状态更新。 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。

相关概念

第 182 页的『指定 CipherSpec』

通过在 **DEFINE CHANNEL** MQSC 命令或 **ALTER CHANNEL** MQSC 命令中使用 **SSLCIPH** 参数来指定 CipherSpec 。

第 28 页的『IBM WebSphere MQ 中的数字证书和 CipherSpec 兼容性』
本主题通过概述 CipherSpecs 与 IBM WebSphere 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 或 TLS

要关闭 SSL 或 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 ' '，单引号 。

   可以按您喜欢的任何顺序关闭集群接收方通道上的 SSL 或 TLS。

   请注意，这些更改在使 SSL 或 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. 在所有手动集群发送方通道上关闭 SSL 或 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 211 页的图 26 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。

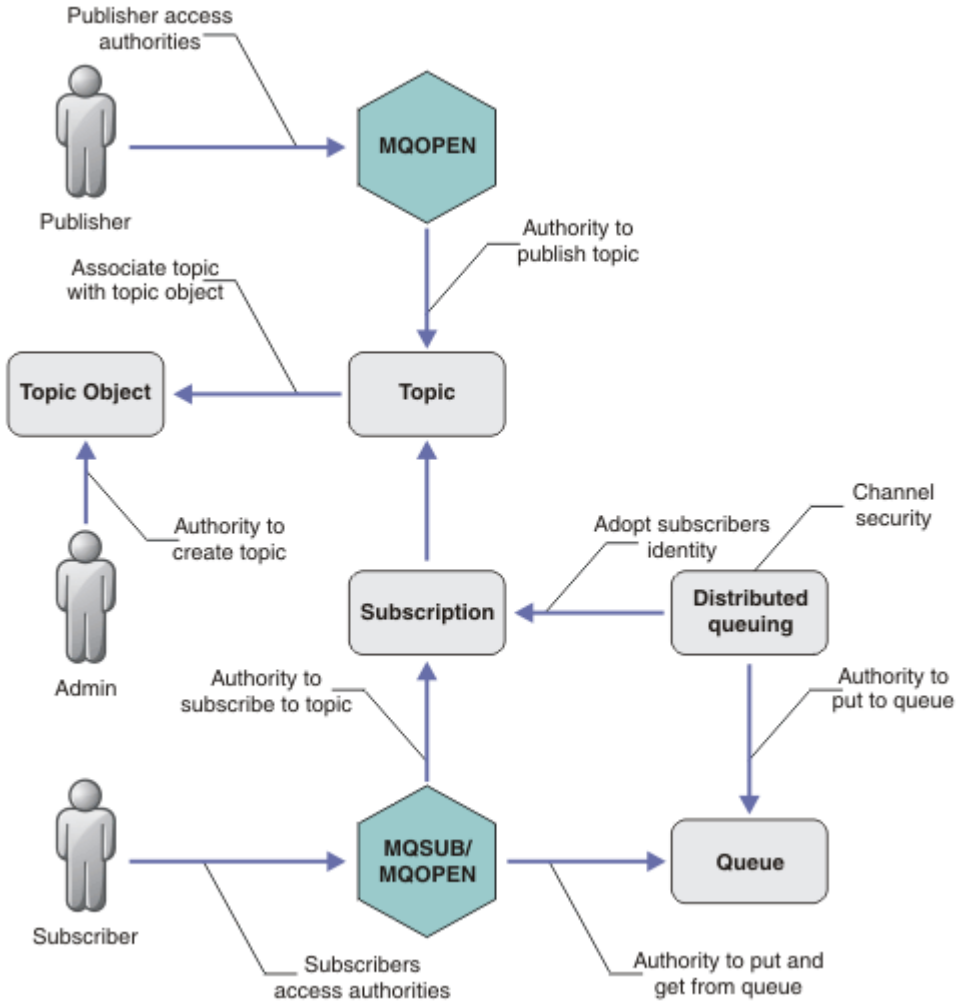


图 *26:* 发布/预订安全关系

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 213 页的『主题安全模型』 说明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 216 页的『授予用户预订主题的访问权』 和 第 221 页的『授予用户访问权以发布到主题』。

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 214 页的『使用主题对象名称进行预订』

使用主题进行预订

第 214 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 215 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 225 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 215 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。

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 WebSphere 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置到目标队列上。 您可以修改通道配置以使用 PUTAUT=CTX 来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。

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。

Authorization

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. WebSphere MQ 管理员-组成员 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。

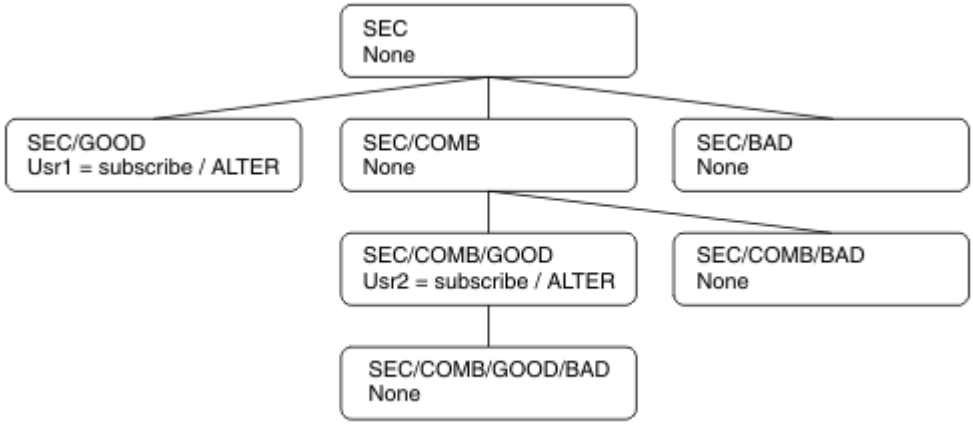
安全性属性是访问控制表，指示特定操作系统用户标识或安全组对主题对象具有的权限。

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 *17:* 示例主题对象权限

主题名称	主题字符串	权限-非 z/OS	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。

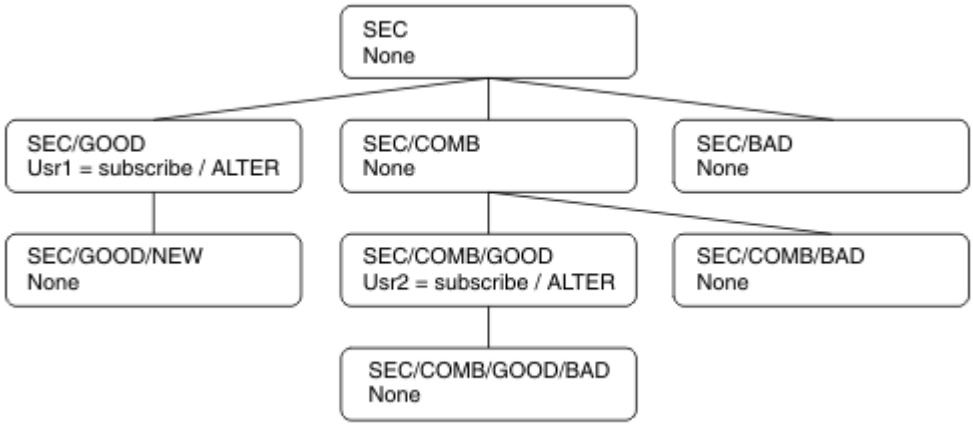
使用存在主题节点的主题字符串进行预订

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 在底层队列上进行检查。 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 216 页的表 18 中所示:

表 *18:* 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		使用管理员标识
	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对主题进行访问控制设置的方案。

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。

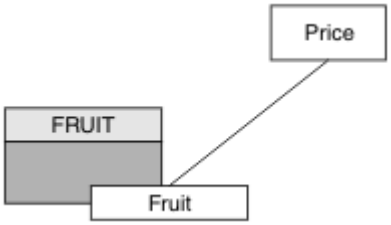


图 *27:* 主题对象访问示例

表 19: 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:
   - 其他平台:

     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:

     
     Windows， UNIX and Linux 系统

     ```
     setmqaut -t topic -n FRUIT -p USER1 +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:

- 
  在其他平台上，以下授权事件:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

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 216 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。

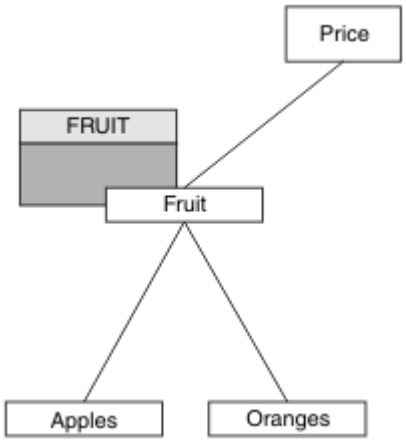


图 *28:* 授予对主题树中主题的访问权的示例

表 *20:* 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	

在先前任务中，通过授予 USER1 对 z/OS 上的 hlq.SUBSCRIBE.FRUIT 概要文件的访问权以及对其他平台上的 FRUIT 概要文件的预订访问权，授予了对主题 "Price/Fruit" 的预订访问权。 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:

- 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 ...
```

- 在其他平台上，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:

- 您在 z/OS 上接收到的消息与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。
- 您在其他平台上接收到的事件消息与先前任务中接收到的事件消息相似，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 217 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

在先前的任务 USER2 中，拒绝了对主题 "Price/Fruit/Apples"的访问。 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。

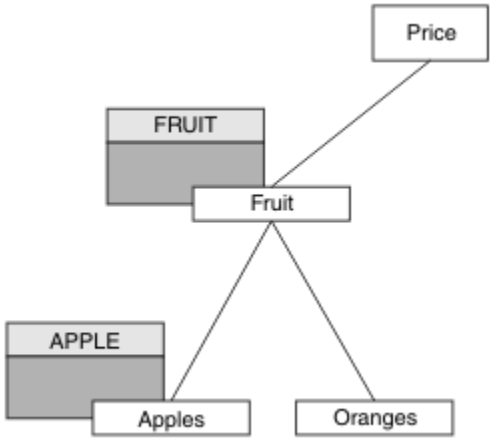


图 *29:* 授予对主题树中特定主题的访问权

表 *21:* 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:
   - 其他平台:

     在先前的任务 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:

     Windows UNIX Linux **Windows， UNIX and Linux 系统**

     ```
     setmqaut -t topic -n APPLE -p USER2 +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:

- Windows UNIX Linux 在 Windows， UNIX 和 Linux 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"
```

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，它告诉您如何授予多个用户预订主题的访问权，以及如何在 z/OS 上避免其他 RACF ICH408I 消息。

开始之前

本主题增强了 第 218 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。

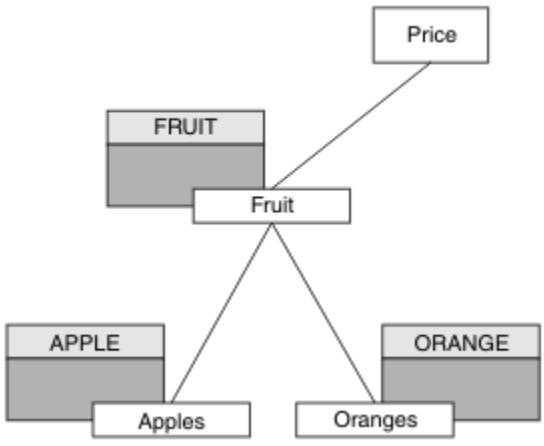


图 *30:* 授予访问控制以避免其他消息的示例。

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:

- 其他平台:

  通过对平台使用授权命令来设置等效访问权:

  Windows UNIX Linux **Windows， UNIX and Linux 系统**

```
setmqaut -t topic -n ORANGE -p USER1 +sub
setmqaut -t topic -n APPLE -p USER1 +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。

同样，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:

- Windows UNIX Linux 在其他平台上，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:

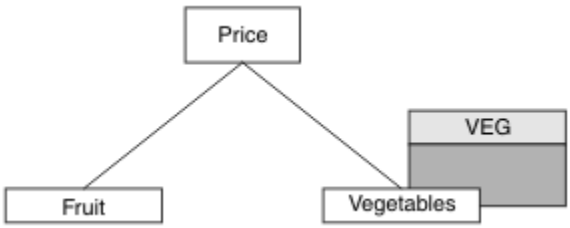


图 *31:* 授予对主题的发布访问权

表 *22:* 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	None
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:
   - 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:

     Windows UNIX Linux **Windows， UNIX and Linux 系统**

     ```
     setmqaut -t topic -n VEG -p USER1 +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:

- Windows UNIX Linux 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"
```

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 221 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。

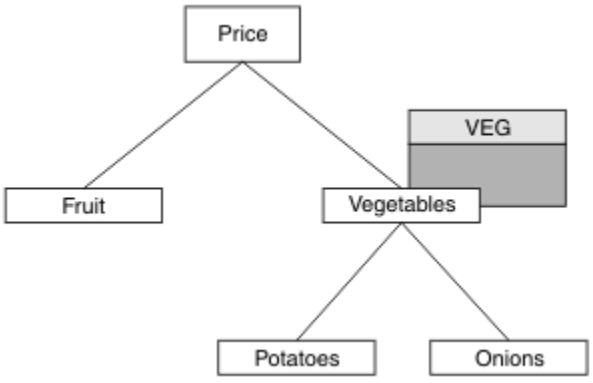


图 *32:* 授予对主题树中主题的发布访问权

表 *23:* 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
价格/蔬菜/土豆	USER1	
价格/蔬菜/洋葱	USER1	

在先前任务中， USER1 通过授予其对 z/OS 上的 hlq.PUBLISH.VEG 概要文件的访问权或对其他平台上的 VEG 概要文件的发布访问权，获得了对发布主题 "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:

- 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 ...
```

- 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/Potatoes"
```

请注意下列事项:

- 您在 z/OS 上接收到的消息与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。
- 您在其他平台上接收到的事件消息与先前任务中接收到的事件消息相似，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 222 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。

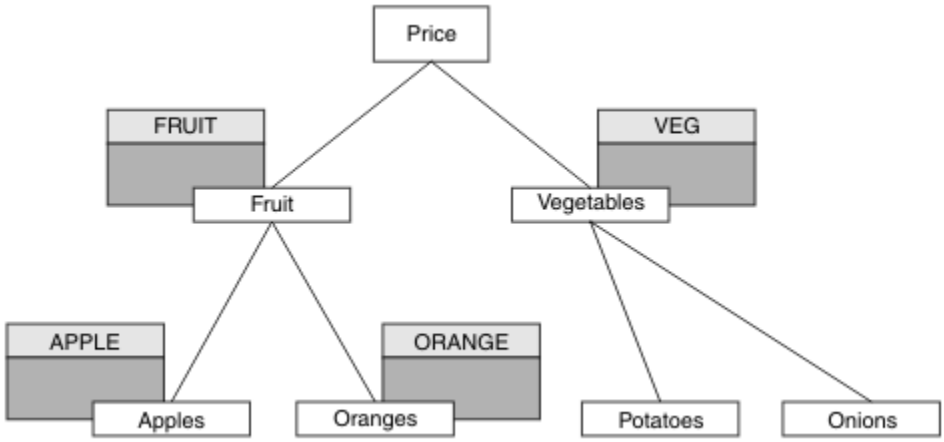


图 *33:* 授予发布和预订访问权

表 *24:* 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:

- 其他平台:

  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:

  Windows UNIX Linux **Windows， UNIX and Linux 系统**

```
setmqaut -t topic -n FRUIT -p USER1 +pub
```

结果

在 z/OS 上，当 USER1 尝试发布到主题 "Price/Fruit" 时，将传递对 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:

- Windows UNIX Linux 在 Windows， UNIX 和 Linux 平台上，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 *25:* 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			

如果您对主题树中的不同级别的安全访问有不同的要求，请仔细规划以确保不会在 z/OS 控制台日志上收到额外的安全警告。 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 *26:* 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。

IBM WebSphere MQ Advanced Message Security

IBM WebSphere MQ Advanced Message Security (AMS) 是单独许可的 IBM WebSphere MQ Advanced Message Security 组件，可为流经 IBM WebSphere MQ Advanced Message Security 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。

IBM WebSphere MQ Advanced Message Security 概述

IBM WebSphere MQ 应用程序可以使用 IBM WebSphere MQ 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。

相关参考

IBM WebSphere MQ AMS 消息中使用的 GSKit 返回码

在 V 7.0.1 与 V 7.5 之间更改的行为

随着 IBM Advanced Message Security 成为 WebSphere MQ 7.5 中的组件， IBM WebSphere MQ AMS 功能的某些方面已更改，可能影响现有应用程序，管理脚本或管理过程的内容。

在将队列管理器升级到 V 7.5 之前，请仔细查看以下更改列表。 决定在开始将系统迁移到 IBM WebSphere MQ V 7.5:

- IBM WebSphere MQ AMS 安装是 WebSphere MQ 安装过程的一部分。
- IBM WebSphere MQ AMS 安全功能通过其安装启用，并通过安全策略进行控制。 您不需要启用拦截器以允许 IBM WebSphere MQ AMS 开始拦截数据。
- WebSphere MQ V 7.5 中的 IBM WebSphere MQ AMS 不需要像在 IBM WebSphere MQ AMS 的独立版本中一样使用 **cfgmqs** 命令。

IBM WebSphere MQ Advanced Message Security 的功能部件和功能

Advanced Message Security 扩展 WebSphere 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， IBM WebSphere MQ 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。

以下是 IBM WebSphere MQ AMS 函数的完整列表:

- 保护由 WebSphere 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 WebSphere MQ 的现有专有应用程序和客户编写的应用程序。

错误处理

Advanced Message Security 定义错误处理队列以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 WebSphere MQ 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。

WebSphere MQ AMS 使用 SYSTEM.PROTECTION.ERROR.QUEUE 作为其错误处理队列。 IBM WebSphere MQ AMS 放入 SYSTEM.PROTECTION.ERROR.QUEUE 前面有 MQDLH 头。

WebSphere 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。

关键概念

了解 Advanced Message Security 中的关键概念，以了解该工具的工作方式以及如何有效管理该工具。

公用密钥基础结构

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 247 页的『使用密钥库和证书』
为了向 WebSphere MQ 应用程序提供透明加密保护， Advanced Message Security 使用存储公用密钥证书和专用密钥的密钥库文件。

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 7 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。

对象权限管理器
对象权限管理器 (OAM) 是随 WebSphere MQ 产品提供的授权服务组件。

通过 WebSphere 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 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

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 WebSphere MQ 应用程序编程接口 (API):

- 消息队列接口 (MQI)
- WebSphere MQ Java 消息服务 (JMS) 1.0.2 和 1.1。
- WebSphere MQ Java 基类
- 非受管方式下的 .Net 的 WebSphere MQ 类

注: Advanced Message Security 支持符合 X.509 的认证中心。

已知限制
了解 IBM WebSphere MQ Advanced Message Security 的限制。

- 不支持以下 IBM WebSphere MQ 选项:

  - 发布/预订。
  - 通道数据转换。
  - 分发列表。
  - 应用程序消息分段
  - 在 HP-UX 平台上使用使用 API 出口的非线程应用程序。
  - 受管方式下的 .NET 的 IBM WebSphere MQ 类 (客户机或绑定连接)。
  - Message Service Client for .NET (XMS) 应用程序。
  - 用于 C/C++ (XMS supportPac IA94) 应用程序的消息服务客户机。
- 所有 Java 应用程序都依赖于 IBM Java 运行时。

  IBM WebSphere MQ Advanced Message Security 不支持其他供应商提供的 JRE。
- 在客户机方式下使用 IBM WebSphere MQ Advanced Message Security 的 JMS 和 Java 客户机应用程序。

  任何 JMS 或 Java 客户机应用程序 (包括 IBM WebSphere MQ Explorer 和 IBM WebSphere MQ Managed File Transfer 代理程序) 在客户机方式下不能将 IBM WebSphere MQ Advanced Message Security 与 Version 7.5 之前的 WebSphere MQ 队列管理器配合使用。

  为了使用消息保护策略，这些应用程序需要与 IBM WebSphere MQ Version 7.5 队列管理器进行交互，或者以本地绑定方式连接到应用程序所在机器上的队列管理器。
- 您应该避免将两个或更多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 IBM WebSphere MQ Advanced Message Security 利益方对此类证书的功能。
- IBM WebSphere MQ Version 7.5 资源适配器不支持 IBM WebSphere MQ Advanced Message Security。如果需要将消息保护与在应用程序服务器环境中运行的 IBM WebSphere MQ classes for JMS 或 IBM WebSphere MQ classes for Java 应用程序配合使用，那么:
  - 必须将应用程序服务器配置为使用 Version 8.0 或更高版本的资源适配器。
  - 或者必须使用消息通道代理程序 (MCA) 拦截。

用户方案

熟悉可能的方案，以了解您可以使用 Advanced Message Security 实现的业务目标。

Windows 平台快速入门指南

使用本指南可快速配置 IBM Advanced Message Security 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

有关详细信息，请参阅 IBM WebSphere 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和执行相应的 WebSphere MQ 命令的信息，请参阅 setmqenv。

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 WebSphere MQ 接口进入 WebSphere MQ 基础结构时对消息进行签名和加密。 基本设置在 WebSphere MQ 中完成，并在以下步骤中进行配置。

您可以使用 WebSphere 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 \WebSphere 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. 您还必须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
```

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 233 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。 如果您计划使用使用客户机绑定的 Java 应用程序，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 239 页的『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. 使用 IBM Key Management GUI (`strmqikm.exe`) 为用户 `alice` 创建新的密钥数据库。

   ```
   Type:       CMS
   Filename:   alicekey.kdb
   Location:   C:/Documents and Settings/alice/AMS
   ```

   注:

   - 建议使用强密码来保护数据库。
   - 确保选中 **隐藏文件密码** 复选框。
2. 将密钥数据库内容视图更改为 **个人证书**。
3. 选择 **新建自签名证书**; 在此方案中使用自签名证书。
4. 使用以下字段创建用于标识用户 `alice` 的证书以用于加密:

   ```
   Key label: Alice_Cert
   Common Name: alice
   Organisation: IBM
   Country: 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，建议不要使用自签名证书，而是依赖于认证中心签署的证书。
   - **Key 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。
   - **Common Name** 和可选参数指定 **专有名称** (DN) 的详细信息，对于每个用户必须唯一。
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 located.This 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。 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，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 248 页的『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 247 页的『使用密钥库和证书』。
- 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/alice/AMS/bobkey.kdb" -pw passw0rd
-label Bob_Cert -target bob_public.arm

runmqakm -cert -add -db "C:/Documents and Settings/bob/AMS/alicekey.kdb" -pw passw0rd -label
Bob_Cert -file bob_public.arm
```

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，将使用 SHA1 算法对消息进行签名，并使用 AES256 算法对消息进行加密。 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。

UNIX 平台快速入门指南

使用本指南可快速配置 IBM Advanced Message Security 以在 UNIX 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以下组件:

- 运行时
- 服务器
- 样本程序
- IBM Global Security Kit
- MQ 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 系统的 IBM WebSphere MQ 组件
- HP-UX 系统的 IBM WebSphere MQ 组件
- AIX 系统的 IBM WebSphere MQ 组件
- Solaris 系统的 IBM WebSphere 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 WebSphere MQ 接口进入 WebSphere MQ 基础结构时对消息进行签名和加密。 基本设置在 WebSphere MQ 中完成，并在以下步骤中进行配置。

您可以使用 WebSphere 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_INSTALL_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. 您还必须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
```

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 238 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。 如果您计划使用使用客户机绑定的 Java 应用程序，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 239 页的『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 248 页的『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 247 页的『使用密钥库和证书』。

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 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。

Java 客户机快速入门指南

使用本指南来快速配置 IBM 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以及为队列管理器定义的签名/加密策略。

开始之前

确保安装了相应的组件，如 **快速入门指南** (窗口 或 UNIX) 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 WebSphere MQ 接口进入 WebSphere MQ 基础结构时对消息进行签名和加密。 基本设置在 WebSphere 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

有两个用户出现在我们的场景中: 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. 为您的平台创建两个用户，如 **快速入门指南** (窗口 或 UNIX) 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
```

3. 您还必须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
```

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 242 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 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。 这显示在 **快速入门指南** (窗口 或 UNIX) 中。

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** (Windows 或 UNIX) 针对您的平台使用的同一目录中创建该目录。

   注: 在以下步骤中，此目录将称为 *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 UNIX 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 的内容将如下所示:

```
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 的内容将如下所示:

```
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。
- 如果您已具有 keystore.conf，因为您已遵循 **快速入门指南** (Windows 或 UNIX)，那么可以编辑现有指南以添加到上述行中。
- 有关更多信息，请参阅第 248 页的『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。

注: 术语 *extract* 和 *export* 由不同的证书工具以不同方式使用。 例如， IBM GSKit Keyman (ikeyman) 工具区分 抽取 证书 (公用密钥) 和 导出 专用密钥。 对于提供这两个选项的工具而言，此区分非常重要，因为错误地使用 *export* 将通过传递其专用密钥来完全损害应用程序。 由于区分非常重要，因此 WebSphere MQ 文档力求一致地使用这些术语。 但是， Java keytool 提供了一个名为 *exportcert* 的命令行选项，该选项仅抽取公用密钥。 由于这些原因，以下过程指的是使用 *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 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
确保正在使用的 Java 版本安装了不受限制的 JCE 策略文件。

注: WebSphere 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。 请参阅适用于您平台的 **快速入门指南** (Windows 或 UNIX) ，以获取有关在不同用户下运行程序的详细信息。

过程
1. 要运行这些 JMS 样本应用程序，请对平台使用 CLASSPATH 设置，如 用于 JMS 的 IBM WebSphere MQ 类所使用的环境变量 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。

保护远程队列
要完全保护远程队列连接，必须在要将消息传输到的远程队列和本地队列上设置相同的策略。

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来处理消息。 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 WebSphere MQ 以进行处理。 在 Advanced Message Security 处理放入远程队列中的消息后， WebSphere 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。

对本地队列执行 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，请除去该容错标志。

相关参考
setmqspl (设置安全策略)

使用 *WebSphere Message Broker* 路由受保护的消息
IBM Advanced Message Security 可以保护安装了 WebSphere Message Broker V 8.0.0.1 (或更高版本) 的基础结构中的消息。 在 WebSphere Message Broker 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 WebSphere Message Broker 的消息，您可以允许 WebSphere Message Broker 在不知道其内容 (方案 1) 的情况下处理消息，或者使其成为能够接收和发送消息的授权用户 (方案 2)。

方案 *1-Message Broker* 无法查看消息内容

开始之前
您应该已将 WebSphere Message Broker 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 WebSphere Message Broker 读取。 Advanced Message Security 仅由 *alice*， *bob* 和 *cecil* 使用。 不必为 WebSphere Message Broker 安装或配置它。

WebSphere Message Broker 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 WebSphere Message Broker 路由，并且未更改地到达目标队列。 因此，它仍然由原始作者签名 (*bob* 和 *cecil* 都只接受 *alice* 发送的消息) ，并且像以前一样受到保护 (只有 *bob* 和 *cecil* 可以读取它)。 WebSphere Message Broker 将路由的消息放置到不受保护的别名中。 收件人从受保护的输出队列中检索消息，其中 IBM WebSphere MQ AMS 将透明地解密消息。

[1] 塞西尔

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 UNIX) 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 UNIX) 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 WebSphere Message Broker 中，根据消息的 routeTo 属性，创建消息流以将到达 AIN 别名队列的消息路由到 BOB ， CECIL 或 DEF 节点。 要这样做:

   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 WebSphere Message Broker 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 WebSphere Message Broker 从 AIN 别名队列以受保护格式检索。 WebSphere Message Broker 决定将读取 routeTo 属性 (作为所有属性) 的消息路由到何处，该属性未加密。 WebSphere Message Broker 将消息放置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-Message Broker* 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 WebSphere Message Broker。 另一个组有权接收由 WebSphere Message Broker 创建的消息。 无法窃用参与方与 WebSphere Message Broker 之间的传输。

请记住，仅当打开队列时， WebSphere Message Broker 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 WebSphere Message Broker 被视为允许读取或签署消息有效内容的授权方，那么必须为启动 WebSphere Message Broker 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 WebSphere Message Broker 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 WebSphere Message Broker 服务用户，以使用 Advanced Message Security ，如 **快速入门指南** (Windows 或 UNIX) 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 WebSphere Message Broker 服务用户提供 *alice*， *bob*， *cecil* 和 *dave 's* 证书。

   执行此操作的方法是将标识 *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 WebSphere Message Broker 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 UNIX) 中的 共享证书** 。

3. 将 WebSphere Message Broker 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 WebSphere Message Broker 服务用户证书来正确加密消息。 WebSphere Message Broker 服务用户需要 *alice* 的 和 *bob* 的 证书来验证消息的作者。 WebSphere Message Broker 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 WebSphere Message Broker 服务用户证书，以验证消息是否来自 WebSphere Message Broker。

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 WebSphere Message Broker 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，并定义将 WebSphere Message Broker 的服务用户指定为作者，将 *cecil* 和 *dave* 指定为收件人的安全策略:

```
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 WebSphere Message Broker 中，使用 MQInput 和 MQOutput 节点创建消息流。 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 WebSphere Message Broker 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 IN 的消息已加密，仅允许 WebSphere Message Broker 读取该消息。 WebSphere Message Broker 将仅接受来自 *alice* 和 *bob* 的消息，并将拒绝任何其他消息。 将适当处理已接受的消息，然后使用 *cecil* 的 和 *dave* 的 密钥对其进行签名和加密，然后再将其放入输出队列 OUT。 只有 *cecil* 和 *dave* 能够读取它，未由 WebSphere Message Broker 签名的消息将被拒绝。

将 *IBM WebSphere MQ Advanced Message Security* 用于 *IBM WebSphere MQ Managed File Transfer*

此场景说明如何配置 Advanced Message Security 以提供通过 IBM WebSphere MQ Managed File Transfer 发送的数据的消息隐私。

开始之前

确保在托管 IBM WebSphere MQ Managed File Transfer 要保护的队列的 WebSphere MQ 安装上安装了 Advanced Message Security 组件。

如果 IBM WebSphere MQ Managed File Transfer 代理程序以绑定方式进行连接，请确保在其本地安装上也安装了 GSKit 组件。

关于此任务

当两个 IBM WebSphere MQ 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 WebSphere 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 IBM WebSphere MQ 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 IBM WebSphere MQ Managed File Transfer 队列和两个代理程序 AGENT1 和 AGENT2 的机器，共享单个队列管理器 hubQM，如场景 使用脚本进行基本文件传输中所述。 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

此场景使用简单模型，其中组 FTAGENTS 中的用户 ftagent 用于运行 IBM WebSphere MQ Managed File Transfer 代理程序进程。 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 IBM WebSphere MQ 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的平台的 **快速入门指南** (Windows 或 UNIX) 中进行了详细描述。
- 如果 IBM WebSphere MQ Managed File Transfer 代理程序以客户机方式运行，那么 第 239 页的『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=IBM, L=Hursley, ST=Hampshire, C=GB
```

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 IBM WebSphere MQ 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=IBM,
L=Hursley, ST=Hampshire, C=GB"
-e AES128 -r "CN=ftagent, OU=MFT, O=IBM, L=Hursley, ST=Hampshire, C=GB"
```

3. 确保运行 IBM WebSphere MQ Managed File Transfer 代理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 IBM WebSphere MQ 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。

安装 IBM WebSphere MQ Advanced Message Security

在各种平台上安装 IBM WebSphere MQ Advanced Message Security 组件。

关于此任务

有关完整的安装过程，请参阅 安装 IBM WebSphere MQ Advanced Message Security。

相关任务

卸载 IBM WebSphere MQ Advanced Message Security

使用密钥库和证书

为了向 WebSphere MQ 应用程序提供透明加密保护， Advanced Message Security 使用存储公用密钥证书和专用密钥的密钥库文件。

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。

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`， `.jks`。

`keystore.conf` 文件的缺省位置为:

- 在 UNIX 平台上: `$HOME/.mqs/keystore.conf`
- 在 Windows 平台上：*%HOMEDRIVE%%HOMEPATH%*`\.mqs\keystore.conf`

如果您正在使用指定的密钥库文件名和位置，那么应使用以下命令

- 对于 Java：`java -D MQS_KEYSTORE_CONF=path/filename app_name`
- 对于 C 客户机和服务器:
  - 在 UNIX and Linux 上：`export MQS_KEYSTORE_CONF=path/filename`
  - 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)。

相关概念

第 259 页的『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。

第 260 页的『接收方专有名称』
接收方专有名称 (DN) 标识有权从队列检索消息的用户。

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。

有两种类型的配置 CMS 和 Java (JKS 和 JCEKS)。 CMS 配置条目以 `cms.` 为前缀， Java 以 `jks.` 或 `jceks.` 为前缀，具体取决于密钥库的类型。

根据配置文件的类型，该配置文件可以具有下列其中一种结构:

```
cms.keystore = /<dir>/<keystore_file>
cms.certificate = certificate_label

jceks.keystore = <dir>/Keystore
jceks.certificate = <certificate_label>
jceks.encrypted = no
jceks.keystore_pass = <password>
jceks.key_pass = <password>
jceks.provider = IBMJCE

jks.keystore = <dir>/Keystore
jks.certificate = <certificate_label>
jks.encrypted = no
jks.keystore_pass = <password>
jks.key_pass = <password>
jks.provider = IBMJCE
```

配置文件参数定义如下:

keystore
密钥库文件的路径。

要点:

- 密钥库文件的路径不得包含文件扩展名。
- 对于 Java 密钥库文件， IBM WebSphere MQ AMS 支持以下文件格式: `.jks`， `.jceks` 和 `.jck`。

certificate
证书标签。

encrypted
密码的状态。

keystore_pass
密钥库文件的密码。

注:

- 对于 CMS 密钥库， IBM WebSphere MQ AMS 依赖于隐藏文件（`.sth`），而 JKS 和 JCEKS 可能需要证书和用户专用密钥的密码。
- 以纯文本存储密码存在安全风险。

key_pass
用户专用密钥的密码。

要点: 以纯文本形式存储密码可能会带来安全风险。

provider
实现密钥库证书所需的加密算法的 Java 安全提供程序。

注: 目前， IBMJCE 是 Advanced Message Security 支持的唯一提供程序。

要点: 存储在密钥库中的信息对于使用 WebSphere MQ 发送的安全数据流至关重要，这就是为什么安全管理员在将文件许可权分配给这些文件时必须特别注意的原因。

以下是 `keystore.conf` 文件的示例:

```
cms.keystore = c:\Documents and Settings\Alice\AliceKeystore
cms.certificate = AliceCert

jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = no
jceks.keystore_pass = <password>
jceks.key_pass = <password>
jceks.provider = IBMJCE
```

相关任务
第 257 页的『在 Java 中保护密码』
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 提供了一个工具，可以使用密钥库文件中提供的用户密钥对这些密码进行加密。

消息通道代理程序 (MCA) 拦截

MCA 拦截使在 IBM WebSphere MQ 下运行的队列管理器能够选择性地对服务器连接通道应用策略。

MCA 拦截允许保留在 IBM WebSphere MQ AMS 外部的客户机仍然连接到队列管理器，并允许对其消息进行加密和解密。

当无法在客户机上启用 IBM WebSphere MQ AMS 时， MCA 拦截旨在提供 IBM WebSphere MQ AMS 功能。请注意，使用 MCA 拦截和启用了 IBM WebSphere MQ AMS 的客户机会导致对消息进行双重保护，这可能会对接收应用程序造成问题。

如果报告了 2085 （MQRC_UNKNOWN_OBJECT_NAME）错误 (如果使用 Version 7.5 或更高版本的客户机从较低版本的产品连接到队列管理器)，那么需要在该客户机上禁用 IBM WebSphere MQ Advanced Message Security 。 有关更多信息，请参阅第 251 页的『在客户机上禁用 IBM WebSphere MQ Advanced Message Security』。

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 IBM WebSphere MQ AMS 密钥库的更多信息，请参阅 第 247 页的『使用密钥库和证书』。

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。

有关设置 MCA 拦截的示例，请参阅 第 249 页的『IBM WebSphere MQ AMS MCA 拦截示例』。

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 WebSphere MQ AMS MCA 拦截示例

有关如何设置 IBM WebSphere MQ 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。

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: 在 IBM WebSphere MQ Version 7.5 之前， IBM WebSphere MQ AMS 是需要单独安装并配置拦截器以保护应用程序的附加产品。 从 Version 7.5 开始，在 MQ 客户机和服务器运行时环境中自动包含并动态启用拦截器。 在此 MCA 拦截示例中，在通道的服务器端提供拦截器，并使用较旧的客户机运行时 (在步骤 12 中) 在通道中放置不受保护的消息，以便可以看到该消息受 MCA 拦截器保护。 如果此示例使用了 Version 7.5 或更高版本的客户机，那么将导致消息受到两次保护，因为 MQ 客户机运行时拦截器和 MCA 拦截器都将在消息进入 MQ 时对其进行保护。

注意: 将代码中的 `userID` 替换为您的用户标识。

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/mq75
KEYSTORELOC=/home/testusr/ssl/ams1
mkdir -p $KEYSTORELOC
chmod -R 777 $KEYSTORELOC
chown -R mqm:mqm $KEYSTORELOC
export PATH=$PATH:$INSTLOC/gskit8/bin
echo "PATH = $PATH"
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$INSTLOC/gskit8/lib64

gsk8capicmd_64 -keydb -create -db $KEYSTORELOC/alicekey.kdb -pw passw0rd -stash
gsk8capicmd_64 -keydb -create -db $KEYSTORELOC/bobkey.kdb -pw passw0rd -stash
gsk8capicmd_64 -cert -create -db $KEYSTORELOC/alicekey.kdb -pw passw0rd
-label alice_cert -dn "cn=alice,O=IBM,c=IN" -default_cert yes
gsk8capicmd_64 -cert -create -db $KEYSTORELOC/bobkey.kdb -pw passw0rd
-label bob_cert  -dn "cn=bob,O=IBM,c=IN"  -default_cert yes
```

2. 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。

   请务必使用 **任务 5 中描述的方法。 快速入门指南 (Windows 或 UNIX) 中的 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
```

4. 创建并启动队列管理器 `AMSQMGR1`
5. 使用 *port* 14567 和 *control* `QMGR` 定义侦听器
6. 禁用通道权限或设置通道权限的规则。

   请参阅 SET CHLAUTH 以获取更多信息。

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。
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. 设置通道配置:

```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/127.0.0.1(14567)'
```

12. 从不会自动启用 MCA 拦截器的 MQ 客户机 (例如， IBM WebSphere MQ Version 7.1 或更低版本的客户机) 运行 **amqsputc** 。 放入以下两条消息:

```
/opt/mqm/samp/bin/amqsputc TESTQ TESTQMGR
```

13. 除去安全策略并验证结果:

```
setmqspl -m AMSQMGR1 -p TESTQ -remove
dspmqspl -m AMSQMGR1
```

14. 从 IBM WebSphere MQ Version 7.5 安装浏览队列:

```
/opt/mq75/samp/bin/amqsbcg TESTQ AMSQMGR1
```

浏览输出以加密格式显示消息。

15. 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
```

16. 从 IBM WebSphere MQ Version 7.5 安装运行 **amqsgetc** :

```
/opt/mqm/samp/bin/amqsgetc TESTQ TESTQMGR
```

相关任务

第 239 页的『Java 客户机快速入门指南』
使用本指南来快速配置 IBM 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以及为队列管理器定义的签名/加密策略。

相关参考

第 228 页的『已知限制』
了解 IBM WebSphere MQ Advanced Message Security 的限制。

在客户机上禁用 IBM WebSphere MQ Advanced Message Security

如果使用 Version 7.5 或更高版本的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，那么需要在客户机上禁用 IBM WebSphere MQ Advanced Message Security (AMS)。

关于此任务

从 Version 7.5 开始，将在 IBM WebSphere MQ 客户机中自动启用 IBM WebSphere MQ Advanced Message Security (AMS) ， 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。 但是，较低版本的产品 (例如 Version 7.1) 上的服务器未启用 AMS ， 这将导致报告 2085 (MQRC_UNKNOWN_OBJECT_NAME) 错误。

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，可以在客户机上禁用 AMS ， 如下所示:

- 对于 Java 客户机，请通过以下任一方式:
  - V7.5.0.4 通过设置环境变量 AMQ_DISABLE_CLIENT_AMS。
  - V7.5.0.4 通过设置 Java 系统属性 com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS。
  - V7.5.0.5 通过使用 DisableClientAMS 属性，在 `mqclient.ini` 文件中的 **Security** 节下。
- 对于 C 客户机，请采用以下任一方式:
  - V7.5.0.4 通过设置环境变量 AMQ_DISABLE_CLIENT_AMS。
  - V7.5.0.5 通过使用 DisableClientAMS 属性，在 `mqclient.ini` 文件中的 **Security** 节下。

过程

- 要在客户机上禁用 AMS，请使用下列其中一个选项:

  V 7.5.0.4 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 运行时环境 (JRE) 以外的 Java 运行时环境 (JRE)

  – 如果您正在使用 Version 7.5 或更高版本 IBM WebSphere MQ classes for Java 或 IBM WebSphere MQ classes for JMS 客户机。

  您还可以使用 AMQ_DISABLE_CLIENT_AMS 来禁用 C 客户机的 AMS 功能。

  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
  ```

  V 7.5.0.4 com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS 系统属性

  对于 IBM WebSphere MQ classes for JMS 和 IBM WebSphere MQ classes for Java 客户机，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:

  ```
  java -Dcom.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS=TRUE -cp <MQ_INSTALLATION_PATH>/java/lib/
  com.ibm.mqjms.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。

  V 7.5.0.5 mqclient.ini 文件中的 DisableClientAMS 属性

  对于 IBM WebSphere MQ classes for JMS 和 IBM WebSphere MQ classes for Java 客户机以及 C 客户机，请在 `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 第 272 页的『使用 JMS 时打开受保护队列时发生问题』。

相关概念

第 249 页的『消息通道代理程序 (MCA) 拦截』

MCA 拦截使在 IBM WebSphere MQ 下运行的队列管理器能够选择性地对服务器连接通道应用策略。

相关任务

使用配置文件配置客户机

相关参考

IBM WebSphere MQ classes for JMS 配置文件

AMS 的证书要求

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 28 页的『IBM WebSphere 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，必须将密钥用法设置为如下所示: 对于用于保护完整性质量的 X.509 V3 或更高版本标准中的证书，如果设置了密钥用法扩展，那么它们必须至少包含以下两者之一:

- **nonRepudiation**
- **digitalSignature**

对于保护隐私的质量，如果设置了密钥使用扩展，那么这些扩展还必须包含 **keyEncipherment** 扩展。

相关概念

第 261 页的『保护质量』

Advanced Message Security 数据保护策略意味着保护质量 (QOP)。

IBM WebSphere MQ Advanced Message Security 中的证书验证方法

您可以使用 IBM WebSphere MQ 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。

IBM WebSphere MQ 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 IBM WebSphere MQ AMS 。 如果同时启用了这两种方法，那么出于性能原因， IBM WebSphere MQ 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 IBM WebSphere MQ AMS 将使用 CRL 检查。

相关概念

第 253 页的『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，因此有助于确定是否可以信任该证书。

第 255 页的『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。

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，因此有助于确定是否可以信任该证书。

在本机拦截器中启用 *OCSP* 检查

要在 Advanced Message Security 中启用联机证书状态协议 (OCSP) 检查，必须修改密钥库配置文件。

过程

将以下选项添加到密钥库配置文件中:

注: 该表中提供了各个选项的缺省值。

必须指定以下值之一:

- `ocsp.enable=on`
- `ocsp.url=<`*`reposnder_URL`*`>`
- `ocsp.http.proxy.host=<`*`OCSP_proxy`*`>`

选项	描述
`ocsp.enable=off`	如果要检查的证书具有使用 PKIX_AD_OCSP 访问方法（包含 OCSP 响应程序所在位置的 URI）的权限信息访问扩展，请启用 OCSP 检查。 可能的值为：`on/off`。
`ocsp.url=<`*`reposnder_URL`*`>`	OCSP 响应程序的 URL 地址。
`ocsp.http.proxy.host=<`*`OCSP_proxy`*`>`	OCSP 代理服务器的 URL 地址。
`ocsp.http.proxy.port=<`*`port_number`*`>`	OCSP 代理服务器的端口号。

选项	描述
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。
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）。

在 *Java* 中启用 *OCSP* 检查
要在 Advanced Message Security 中对 Java 启用 OCSP 检入，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*
检查您的证书是否设置了 "权限信息访问" (AIA)。

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 安全管理器，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";
```

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。IBM WebSphere MQ AMS 使用 PKIX 验证算法验证证书路径。 创建并验证链时， IBM WebSphere MQ 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 IBM WebSphere MQ 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， IBM WebSphere MQ AMS 根据配置文件中指定的值从文件或 LDAP 下载 CRL。 IBM WebSphere MQ 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 IBM WebSphere MQ AMS 不会执行 CRL 有效性检验。 对于每个 CA 证书， IBM WebSphere MQ 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。

过程

将以下选项添加到配置文件:

注: 该表中提供了各个选项的缺省值。

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 IBM WebSphere MQ 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。

选项	描述
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.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 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，IBM WebSphere MQ AMS 将超时。
crl.http.cache_size=0	HTTP 高速缓存大小 (以字节计)。

在 *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 IBM WebSphere MQ 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 IBM WebSphere MQ 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。
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。

在 Java 中保护密码

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 提供了一个工具，可以使用密钥库文件中提供的用户密钥对这些密码进行加密。

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取该文件。 本章中描述的密码保护只是额外的保护措施。

过程

1. 编辑 keystore.conf 文件以包含密钥库和用户标签的路径。

```
jceks.keystore = c:/Documents and Setting/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.provider = IBMJCE
```

2. 要运行该工具，请发出:

```
java -cp com.ibm.mq.jmqi.jar com.ibm.mq.ese.config.KeyStoreConfigProtector keystore_password
private_key_password
```

将生成包含加密密码的输出，并可将其复制到 keystore.conf 文件。

要自动将输出复制到 keystore.conf 文件，请运行:

```
java -cp com.ibm.mq.jmqi.jar com.ibm.mq.ese.config.KeyStoreConfigProtector keystore_password
private_key_password >> ~/<path_to_keystore>/keystore.conf
```

注:

有关 keystore.conf 在各种平台上的缺省位置的列表，请参阅 第 247 页的『使用密钥库和证书』。

示例

以下是此类输出的示例:

```
#Fri Jul 30 15:20:29 CEST 2010
jceks.key_pass=MMXh997n5ZOr8uRlJmc5qity9MN2CggGBMKCDxdbn1AyPklvdgTsOLG6X3C1YT7oDzwaqZFlOR4t\r\nm
Zsc7JGAx8nqqxLnAucdGn0NWo6xnjZB1n501YGol2k/
PhaQHhFXKMAU9dKg0f8djOtCAOlX4ETe\r\nfYl9LBUt2wk87uM7dSs\=
jceks.keystore_pass=OIdeayBnSCfLG4cFuxEVrk6SYyAsdSPpDqgPf16s9s1M04cqZjNbhgjoA2EXonudHZHH+4s2drvQ
\r\nCUvQgu9GuaBMJK2F2OjtHJJ1Y4BVeLW2c2okgawo/
W2J1AdUYKkJ0raYTkDouLaTYTQeulyG0xIl\r\niD2si1xUCxhYvvyhbbY\=
jceks.encrypted=yes
```

管理 IBM WebSphere MQ Advanced Message Security 安全策略

IBM WebSphere MQ 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。

安全策略概述

IBM 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 261 页的『保护质量』
Advanced Message Security 数据保护策略意味着保护质量 (QOP)。

第 260 页的『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。

策略名称

策略名称是唯一名称，用于标识特定 Advanced Message Security 策略及其应用的队列。

策略名称必须与应用策略的队列名称相同。 Advanced Message Security (IBM WebSphere MQ AMS) 策略与队列之间存在一对一映射。

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 IBM WebSphere MQ 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。

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。

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:

- RC2
- DES

- 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 加密的消息也已签名。

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。

容差

容错属性指示 IBM 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。

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。

在检索消息之前， IBM Advanced Message Security (IBM WebSphere MQ AMS) 不会检查消息是否已由有效用户放入数据保护的队列中。 此时，如果策略规定了一个或多个有效发送方，而在队列中放入该消息的用户不在有效发送方列表中，那么 IBM WebSphere MQ AMS 会向获取应用程序返回一个错误，并将消息放入其错误队列中。

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，那么只要用户的证书可信，他们都可以将数据保护型消息放入队列。

发送方专有名称的格式如下:

```
CN=Common Name,O=Organization,C=Country
```

要点:

- 所有 DN 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。

- IBM WebSphere MQ AMS 仅支持具有来自 Latin-1 字符集的值的 DN。 要创建具有集合字符的 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 UNIX 平台或使用 iKeyman 实用程序以 UTF-8 编码创建的。 然后，必须从开启了 UTF-8 编码的 UNIX 平台创建策略，或者使用 IBM WebSphere MQ AMS 插件来 WebSphere MQ。

相关概念

第 260 页的『接收方专有名称』
接收方专有名称 (DN) 标识有权从队列检索消息的用户。

接收方专有名称

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 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。 要创建具有集合字符的 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 UNIX 平台或使用 iKeyman 实用程序以 UTF-8 编码创建的。 然后，必须从开启了 UTF-8 编码的 UNIX 平台创建策略，或者使用 Advanced Message Security 插件来 WebSphere MQ。

相关概念

第 259 页的『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。

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。 下表显示了 Advanced Message Security 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。

属性	描述
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:

- RC2
- DES
- 3DES
- AES128
- AES256

Advanced Message Security 还支持以下加密散列函数:

- 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)

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。

保护质量

Advanced Message Security 数据保护策略意味着保护质量 (QOP)。

Advanced Message Security 中的三个保护级别质量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。
- 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"。

未规定签名算法或加密算法的策略具有 QOP NONE。 Advanced Message Security 为具有 QOP 为 NONE 的策略的队列提供无数据保护。

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。

与安全策略相关的所有管理任务都从以下位置运行:

- 在 UNIX 平台上: <MQInstallRoot>/bin
- 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。

相关任务

第 262 页的『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 262 页的『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 263 页的『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 264 页的『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。

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 WebSphere MQ 对象的规则。
- 您必须具有必要的 +connect +inq +chg 权限才能创建安全策略。 有关授权更改命令的完整语法，请参阅 **setmqaut** 。
- 确保您具有对 WebSphere MQ 队列和队列管理器进行操作所需的许可权。 有关更多信息，请参阅 第 266 页的『授予 OAM 许可权』

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 SHA1 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 SHA1 -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 DES 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 MD5 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 MD5 -e DES -r CN=john,O=IBM,C=US -r CN=jeff,O=IBM,C=US
-a CN=phil,O=IBM,C=US
```

注:

- 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

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 +connect +inq +chg 权限才能创建安全策略。 有关授权更改命令的完整语法，请参阅 **setmqaut** 。

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，该策略指定将使用 RC2 算法对具有 DN:CN=bob， O=IBM， C=US 的证书进行加密，并使用 SHA1 算法对具有 DN:CN=jeff， O=IBM， C = US 的证书进行签名。

```
setmqspl -m QMGR -p MYQUEUE -e RC2 -s SHA1 -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 SHA1 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

相关参考

设置 MQspl

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。
- 您必须具有应用于队列管理器的必需 +connect +inq +dsp 许可权，才能显示和转储安全策略。 有关授权更改命令的完整语法，请参阅 **setmqaut** 。

关于此任务

以下是 dspmqspl 命令标志的列表:

表 *27: 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

在此示例中，我们将为 venus.queue.manager 创建两个安全策略:

```
setmqspl -m venus.queue.manager -p AMS_POL_04_ONE -s MD5 -a "CN=signer1,O=IBM,C=US" -e NONE
setmqspl -m venus.queue.manager -p AMS_POL_06_THREE -s MD5 -a "CN=another signer,O=IBM,C=US" -e
NONE
```

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: MD5
Encryption algorithm: NONE
Signer DNs:
  CN=signer1,O=IBM,C=US
Recipient DNs: -
Toleration: 0
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Policy Details:
Policy name: AMS_POL_06_THREE
Quality of protection: INTEGRITY
```

```
Signature algorithm: MD5
Encryption algorithm: NONE
Signer DNs:
  CN=another signer,O=IBM,C=US
Recipient DNs: -
Toleration: 0
```

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: MD5
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 MD5 -a "CN=signer1,O=IBM,C=US" -e NONE

dspmqspl -m venus.queue.manager -export > policies.[bat|sh]
```

要导入安全策略:

- 在 Windows 平台上，运行 policies.bat
- 在 UNIX 平台上:
  1. 以属于 mqm WebSphere MQ 管理组的用户身份登录。
  2. 问题 . policies.sh。

相关参考

dspmqspl 命令属性的完整列表

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 +connect +inq +chg 权限才能创建安全策略。 有关授权更改命令的完整语法，请参阅 **setmqaut** 。

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

系统队列保护

系统队列支持 WebSphere 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 WebSphere 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 262 页的『创建安全策略』。

要在 Windows 平台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
```

要为 `SYSTEM.ADMIN.COMMAND.QUEUE` 提供保护，命令服务器必须有权访问 `keystore` 和 `keystore.conf`，其中包含密钥和配置，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要求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。 如果管理员定义了授权签署者，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 WebSphere MQ Explorer 临时动态队列的消息不受 WebSphere MQ AMS 保护。

对 Advanced Message Security 安全策略的更改要求您重新启动 WebSphere MQ 命令服务器

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
- SYSTEM.ADMIN.CONFIG.EVENT
- SYSTEM.ADMIN.LOGGER.EVENT
- SYSTEM.ADMIN.PERFM.EVENT
- SYSTEM.ADMIN.PUBSUB.EVENT
- SYSTEM.ADMIN.QMGR.EVENT
- SYSTEM.ADMIN.STATISTICS.QUEUE
- SYSTEM.ADMIN.TRACE.ROUTE.QUEUE
- SYSTEM.AUTH.DATA.QUEUE
- SYSTEM.BROKER.ADMIN.STREAM
- SYSTEM.BROKER.CONTROL.QUEUE
- SYSTEM.BROKER.DEFAULT.STREAM
- SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS
- SYSTEM.CHANNEL.INITQ
- SYSTEM.CHANNEL.SYNCQ
- SYSTEM.CICS.INITIATION.QUEUE
- SYSTEM.CLUSTER.COMMAND.QUEUE
- SYSTEM.CLUSTER.HISTORY.QUEUE
- SYSTEM.CLUSTER.REPOSITORY.QUEUE
- SYSTEM.CLUSTER.TRANSMIT.QUEUE
- SYSTEM.DEAD.LETTER.QUEUE
- SYSTEM.DURABLE.SUBSCRIBER.QUEUE
- SYSTEM.HIERARCHY.STATE
- SYSTEM.INTER.QMGR.CONTROL

- SYSTEM.INTER.QMGR.FANREQ
- SYSTEM.INTER.QMGR.PUBS
- SYSTEM.INTERNAL.REPLY.QUEUE
- SYSTEM.PENDING.DATA.QUEUE
- SYSTEM.PROTECTION.ERROR.QUEUE
- SYSTEM.PROTECTION.POLICY.QUEUE
- SYSTEM.RETAINED.PUB.QUEUE
- SYSTEM.SELECTION.EVALUATION.QUEUE
- SYSTEM.SELECTION.VALIDATION.QUEUE

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， IBM Advanced Message Security 依赖于对象权限管理器 (OAM)，并且不属于 mqm 组 (即 WebSphere 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
```

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。

WebSphere MQ 生成的命令和配置事件是发送到专用队列的 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 262 页的『创建安全策略』，生成两条 WebSphere MQ 事件消息:
  - 配置事件
  - 命令事件
- 第 262 页的『更改安全策略』，这将生成三条 WebSphere 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 263 页的『显示和转储安全策略』，这将生成一条 WebSphere MQ 事件消息:
  - 命令事件
- 第 264 页的『除去安全策略』，这将生成两条 WebSphere MQ 事件消息:
  - 配置事件
  - 命令事件

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 Websphere MQ 中控制配置，命令和记录器事件

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:	始终。

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 WebSphere MQ Explorer 插件的命令消息。

已返回: 始终。

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: 始终。

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。

EventIdentityData

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。

Command

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
这些是在 WebSphere MQ 7.5 `cmqcfc.h` 中定义的

已返回: 始终。

CommandData 包含组成 PCF 命令的 PCF 元素。

配置事件消息格式

配置事件是标准 Advanced Message Security 格式的 PCF 消息。

有关 MQMD 消息描述符的可能值，请参阅 事件消息 MQMD (消息描述符)。

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。 有关可能的 MQCFH 值，请参阅 事件消息 MQCFH (PCF 头)。

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 或适用于本地命令的 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 WebSphere 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 WebSphere MQ 7.5 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 WebSphere MQ 7.5 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 WebSphere MQ 7.5 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 WebSphere MQ 7.5 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 WebSphere MQ 7.5 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 WebSphere MQ 7.5 或 cmqc.h 文件中定义的数字值。

已返回: 只要 WebSphere 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 WebSphere MQ 7.5 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 WebSphere 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 WebSphere MQ 7.5 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 WebSphere MQ 策略中定义。

问题和解决方法

本部分描述了如何解决 IBM 的任何安装可能出现的问题，请使用此信息来确定和解决与 Advanced Message Security 相关的任何问题。

com.ibm.security.pkcsutil.PKCSException: 对内容进行加密时出错

错误 `com.ibm.security.pkcsutil.PKCSException: Error encrypting contents` 表明 IBM Advanced Message Security 在访问密码算法方面存在问题。

如果 Advanced Message Security 返回以下错误:

```
DRQJP0103E   The IBM WebSphere MQ Advanced Message Security Java interceptor failed to protect
message.
  com.ibm.security.pkcsutil.PKCSException: Error encrypting contents
(java.security.InvalidKeyException: Illegal key size or default parameters)
```

请验证 `JAVA_HOME/lib/security/local_policy.jar/*.policy` 中的 JCE 安全策略是否授予对 MQ AMS 策略中使用的签名算法的访问权限。

如果要使用的签名算法未在当前安全策略中指定，请从以下位置下载正确的 Java 策略文件:

- IBM SDK Policy files for Java 1.4.2。
- IBM SDK Policy files for Java 5.0。
- IBM SDK Policy files for Java 6.0。
- IBM SDK Policy files for Java 7.0。

OSGi 支持

要将 OSGi 捆绑软件与 IBM Advanced Message Security 一起使用，需要其他参数。

在 OSGi 捆绑软件启动期间运行以下参数:

```
-Dorg.osgi.framework.system.packages.extra=com.ibm.security.pkcs7
```

在 keystore.conf 中使用加密密码时，必须在 OSGi 捆绑软件正在运行时添加以下语句:

```
-Dorg.osgi.framework.system.packages.extra=com.ibm.security.pkcs7,com.ibm.misc
```

限制: IBM WebSphere MQ AMS 仅支持对受 OSGi 捆绑软件保护的队列使用 MQ 基本 Java 类进行通信。

使用 JMS 时打开受保护队列时发生问题

使用 IBM WebSphere MQ Advanced Message Security 时，打开受保护队列可能会出现各种问题。

您正在运行 JMS ，并且收到错误 2085 (MQRC_UNKNOWN_OBJECT_NAME) 以及错误 JMSMQ2008。

您已验证是否已按 第 239 页的『Java 客户机快速入门指南』中所述设置 IBM WebSphere MQ Advanced Message Security 。

可能的原因是您正在使用非 IBM Java 运行时环境。 这是 第 228 页的『已知限制』中描述的已知限制。

您未设置 AMQ_DISABLE_CLIENT_AMS 环境变量。

解决问题

解决此问题有四种选择：

1. 在受支持的 IBM Java 运行时环境 (JRE) 下启动 JMS 应用程序。
2. 将应用程序移动到正在运行队列管理器的同一台计算机上，并使用绑定方式连接进行连接。

   绑定方式连接将使用平台本机库来执行 IBM WebSphere MQ API 调用。 因此，将使用本机 AMS 拦截器来执行 AMS 操作，而不依赖于 JRE 的功能。
3. 请使用 MCA 拦截器，因为这样可以在消息到达队列管理器时立即对消息进行签名和加密，而无需客户机执行任何 AMS 处理。

   假定对队列管理器应用了保护，必须使用备用机制来保护从客户机传输到队列管理器的消息。 最常见的情况是通过在应用程序所使用的服务器连接通道上配置 SSL/TLS 加密来实现。
4. 如果您不希望使用 IBM WebSphere MQ Advanced Message Security，请设置 AMQ_DISABLE_CLIENT_AMS 环境变量。

请参阅第 249 页的『消息通道代理程序 (MCA) 拦截』，以了解更多信息。

注: 必须为 MCA 拦截器将消息传递到的每个队列制订安全策略。 换句话说，目标队列需要制订 AMS 安全策略，其中应包含签署者和接收方的专有名称 (DN)，并且该专有名称与分配给 MCA 拦截器的证书的专有名称匹配。 即，由队列管理器使用的 `keystore.conf` 中的 `cms.certificate.channel.SYSTEM.DEF.SVRCONN` 属性指定的证书的 DN。

声明

本信息是为在美国提供的产品和服务编写的。

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默示只能使用 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、程序或服务的操作，由用户自行负责。

IBM 公司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。 您可以以书面形式将许可查询寄往：

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：

知识产权许可
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 063-8506 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 可以随时对本出版物中描述的产品和/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。 所有这些名称都是虚构的，如与实际商业企业所使用的名称和地址有任何雷同，纯属巧合。

版权许可：

本信息包含源语言形式的样本应用程序，用以阐明在不同操作平台上的编程技术。 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 WebSphere 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 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 (http://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: