9.2

保護 *IBM MQ* 安全

附註

使用本資訊及其支援的產品之前，請先閱讀第 579 頁的『注意事項』中的資訊。

目錄

保護 IBM MQ

安全是 IBM MQ 應用程式開發人員及 IBM MQ 系統管理者的重要考量。

安全更新

確保安全區域內及操作員工作站上的所有軟硬體都在其支援生命週期內，已使用必要的軟體更新進行升級，並已立即套用安全更新。

您可以找到下列安全更新項目的進一步相關資訊:

- IBM Security Bulletins 中的所有平台
- IBM Z System Integrity 入口網站上 z/OS 的安全和系統完整性 APAR。

安全概觀

此主題集合介紹 IBM MQ 安全概念。

安全概念和機制 (適用於任何電腦系統) 會先呈現，然後在 IBM MQ 中實作這些安全機制時，再討論這些安全機制。

安全概念和機制

此主題集合說明在 IBM MQ 安裝中要考量的安全層面。

一般接受的安全層面如下:

- 第 5 頁的『識別及鑑別』
- 第 6 頁的『授權』
- 第 6 頁的『審核』
- 第 6 頁的『機密性』
- 第 6 頁的『資料完整性』

安全機制 是用來實作安全服務的技術工具和技術。 機制本身或與其他機制一起運作，以提供特定服務。 一般安全機制的範例如下:

- 第 7 頁的『加密法』
- 第 8 頁的『訊息摘要和數位簽章』
- 第 9 頁的『數位憑證』
- 第 12 頁的『公開金鑰基礎架構 (PKI)』

當您規劃 IBM MQ 實作時，請考量實作對您重要的安全層面所需的安全機制。 如需在閱讀這些主題之後考量事項的相關資訊，請參閱 第 69 頁的『規劃安全需求』。

相關概念

第 230 頁的『使用 SSL/TLS』
這些主題提供如何執行與搭配使用 TLS 與 IBM MQ 相關的單一作業的指示。

相關工作

使用 TLS 連接兩個佇列管理程式

識別及鑑別

識別 是指能夠唯一識別系統中執行之系統或應用程式的使用者。 鑑別 是指能夠證明使用者或應用程式真正是該人員或該應用程式所要求的人員。

例如，考量透過輸入使用者 ID 和密碼來登入系統的使用者。 系統會使用使用者 ID 來識別使用者。 系統會檢查所提供的密碼是否正確，以在登入時鑑別使用者。

不可否認性

不可否認性 服務可以視為識別及鑑別服務的延伸。 一般而言，當以電子方式傳送資料時，即適用不可否認性; 例如，向股票經紀人發出買賣股票的訂單，或向銀行發出將資金從一個帳戶轉移到另一個帳戶的訂單。

不可否認性服務的整體目標是能夠證明特定訊息與特定個人相關聯。

不可否認性服務可以包含多個元件，其中每一個元件提供不同的功能。 如果訊息的傳送者拒絕傳送訊息，則具有 起源證明 的不可否認性服務可以向接收者提供該特定個人傳送訊息的不可否認證據。 如果訊息的接收端拒絕接收訊息，則具有 遞送證明 的不可否認性服務可以向傳送端提供該特定個人接收訊息的不可否認證據。

在實踐中，幾乎百分之百肯定的證據或不可否認的證據是一個困難的目標。 在現實世界中，沒有任何東西是完全安全的。 管理安全更關心將風險管理到企業可接受的層次。 在這種環境下，對不可否認性服務的更現實的期望是能夠在法院提供可以受理的證據，並支援你的案件。

不可否認性是 IBM MQ 環境中的相關安全服務，因為 IBM MQ 是透過電子方式傳輸資料的方法。 例如，您可能需要即時證明與特定個人相關聯的應用程式已傳送或接收特定訊息。

IBM MQ with Advanced Message Security 不提供不可否認性服務作為其基本功能的一部分。 不過，本產品說明文件確實包含如何透過撰寫您自己的結束程式，在 IBM MQ 環境內提供您自己的不可否認性服務的建議。

相關概念

第 18 頁的『IBM MQ 中的識別及鑑別』
在 IBM MQ 中，您可以使用訊息環境定義資訊及交互鑑別來實作識別及鑑別。

授權

授權 會限制只存取授權使用者及其應用程式，以保護系統中的重要資源。 它可防止未獲授權使用資源或以未獲授權的方式使用資源。

相關概念

第 18 頁的『IBM MQ 中的授權』
您可以使用授權來限制特定個人或應用程式在 IBM MQ 環境中可以執行的動作。

審核

審核 是記錄及檢查事件的處理程序，以偵測是否發生任何非預期或未獲授權的活動，或是否嘗試執行此類活動。

如需如何設定授權的相關資訊，請參閱 第 72 頁的『規劃授權』 及相關聯的子主題。

相關概念

第 19 頁的『IBM MQ 中的審核』
IBM MQ 可以發出事件訊息，以記錄發生異常活動。

機密性

機密性 服務可保護機密性資訊免遭未獲授權的揭露。

當機密資料儲存在本端時，在假設無法存取資料時無法讀取資料時，存取控制機制可能足以保護它。 如果需要更高層次的安全，則可以加密資料。

當機密資料透過通訊網路傳輸時，特別是透過不安全的網路 (例如網際網路) 傳輸時，會加密機密資料。 在網路環境中，存取控制機制對截取資料的嘗試無效，例如竊聽。

資料完整性

資料完整性 服務會偵測是否有未獲授權的資料修改。

有兩種方式可以變更資料: 不小心、透過硬體及傳輸錯誤，或因為蓄意攻擊。 許多硬體產品及傳輸通訊協定都有偵測及更正硬體及傳輸錯誤的機制。 資料完整性服務的目的是偵測蓄意攻擊。

資料完整性服務僅旨在偵測資料是否已修改。 如果資料已修改，則不會將資料還原至其原始狀態。

存取控制機制可以增進資料完整性，因為如果拒絕存取，就無法修改資料。 但是，與機密性一樣，存取控制機制在網路環境中並不有效。

加密概念

此主題集合說明適用於 IBM MQ 的加密法概念。

術語 實體 用來指能夠交換訊息的佇列管理程式、 IBM MQ MQI client、個別使用者或任何其他系統。

相關概念

第 20 頁的『IBM MQ 中的加密法』
IBM MQ 使用「傳輸安全層 (TLS)」通訊協定來提供加密法。

加密法

加密法是在可讀取文字 (稱為 純文字) 與無法讀取格式 (稱為 密文) 之間進行轉換的程序。

這會發生如下:

1. 寄件人將純文字訊息轉換為密文。 這部分程序稱為 加密 (有時稱為 加密)。
2. 密文會傳送至接收端。
3. 接收端會將密文訊息轉換回其純文字格式。 這部分程序稱為 解密 (有時稱為 解密)。

轉換涉及一連串數學運算，這些運算會在傳輸期間變更訊息的外觀，但不會影響內容。 加密技術可以確保機密性，並防止訊息遭到未獲授權的檢視 (竊聽) ，因為加密訊息是無法理解的。 數位簽章提供訊息完整性的保證，使用加密技術。 如需相關資訊，請參閱第 16 頁的『SSL/TLS 中的數位簽章』。

加密技術涉及一般演算法，由使用金鑰所產生的特定演算法。 演算法有兩種類別:

- 需要雙方使用相同秘密金鑰的那些項目。 使用共用金鑰的演算法稱為 對稱 演算法。 第 7 頁的圖 1 說明對稱金鑰加密法。
- 使用一個金鑰進行加密，使用另一個金鑰進行解密的那些金鑰。 其中一項必須保密，而另一項則可以公開。 使用公開和私密金鑰組的演算法稱為 非對稱 演算法。 第 8 頁的圖 2 說明非對稱金鑰加密法，也稱為 公開金鑰加密法。

使用的加密和解密演算法可以公開，但共用秘密金鑰和私密金鑰必須保密。

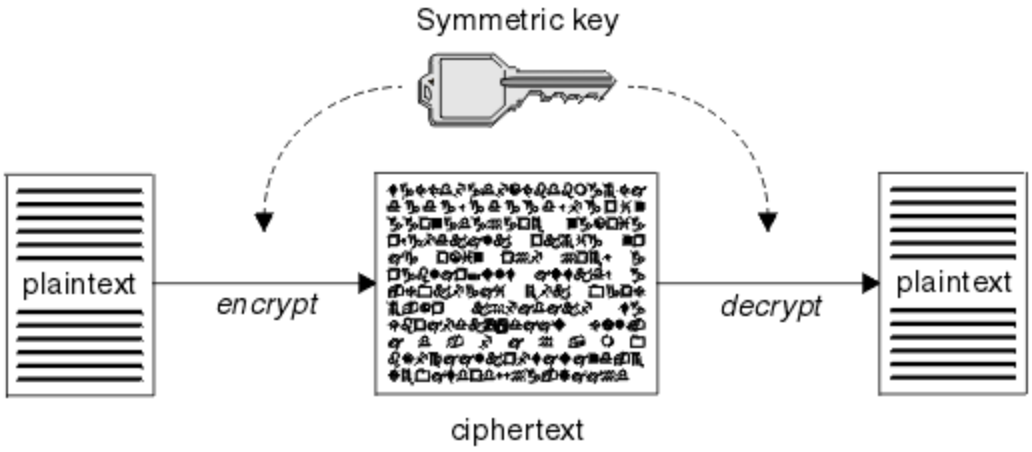


圖 *1:* 對稱金鑰加密法 *(symmetric key cryptography)*

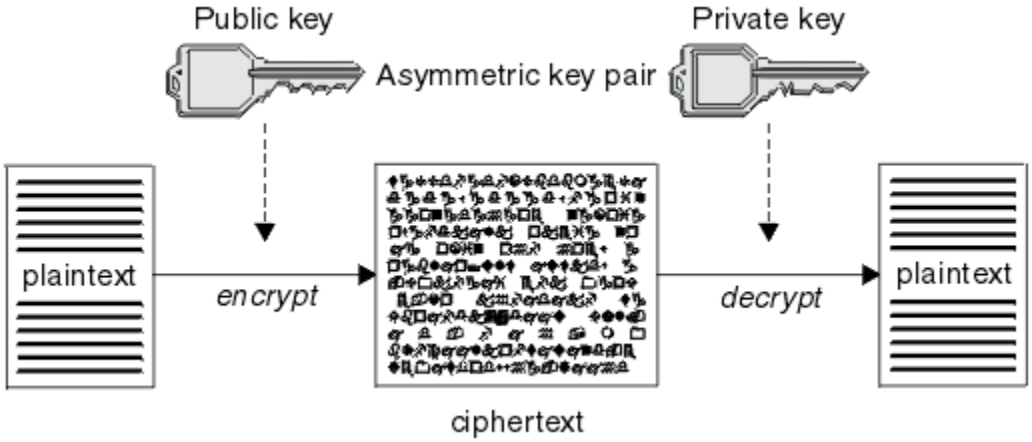


圖 *2:* 非對稱金鑰加密法 *(asymmetric key cryptography)*

第 8 頁的圖 2 顯示以接收端公開金鑰加密並以接收端私密金鑰解密的純文字。 只有預期的接收端會保留用來解密密文的私密金鑰。 請注意，傳送端也可以使用私密金鑰來加密訊息，這可讓任何保留傳送端公開金鑰的人解密訊息，並確保訊息必須來自傳送端。

使用非對稱演算法時，訊息會使用公開或私密金鑰來加密，但只能使用其他金鑰來解密。 只有私密金鑰是秘密，任何人都可以知道公開金鑰。 使用對稱演算法時，只有雙方才能知道共用金鑰。 這稱為 金鑰配送問題。 非對稱演算法較慢，但具有沒有金鑰配送問題的優點。

與加密法相關聯的其他術語如下:

強度
加密強度取決於金鑰大小。 非對稱演算法需要大型金鑰，例如:

1024 位元	低強度非對稱金鑰
2048 位元	中強度非對稱金鑰
4096 位元	高強度非對稱金鑰

對稱金鑰較小: 256 位元金鑰為您提供高度加密。

區塊密碼演算法
這些演算法會依區塊來加密資料。 例如，來自 RSA Data Security Inc. 的 RC2 演算法使用區塊長度為 8 個位元組。 區塊演算法通常比串流演算法慢。

串流密碼演算法
這些演算法會對資料的每一個位元組進行操作。 串流演算法通常比區塊演算法更快。

訊息摘要和數位簽章

訊息摘要是訊息內容的固定大小數值表示法。 訊息摘要由雜湊函數計算並可加密，形成數位簽章。

用來計算訊息摘要的雜湊函數必須符合兩個準則:

- 一定是單向的 除了測試所有可能的訊息之外，不可能反轉函數來尋找對應於特定訊息摘要的訊息。
- 它必須在計算上不可行，才能找到雜湊至相同摘要的兩個訊息。

訊息摘要會隨訊息本身一起傳送。 接收端可以產生訊息的摘要，並將它與傳送端的摘要進行比較。 當兩個訊息摘要相同時，會驗證訊息的完整性。 在傳輸期間對訊息的任何竄改幾乎肯定會導致不同的訊息摘要。

使用秘密對稱金鑰建立的訊息摘要稱為「訊息鑑別碼 (MAC)」，因為它可以提供訊息未修改的保證。

傳送端也可以產生訊息摘要，然後使用非對稱金鑰配對的私密金鑰來加密摘要，形成數位簽章。 然後，簽章必須由接收端解密，然後再與本端產生的摘要進行比較。

相關概念
第 16 頁的『SSL/TLS 中的數位簽章』

通過加密訊息的表示來形成數字簽名。 加密會使用簽署人的私密金鑰，為了效率，通常會對訊息摘要而非訊息本身進行操作。

數位憑證

數位憑證可防止假冒，認證公開金鑰屬於指定的實體。 它們由「憑證管理中心」發出。

數位憑證提供防止假冒的保護，因為數位憑證會將公開金鑰連結至其擁有者，不論該擁有者是個人、佇列管理程式或某個其他實體。 數位憑證也稱為公開金鑰憑證，因為當您使用非對稱金鑰架構時，它們會向您保證公開金鑰的所有權。 數位憑證包含實體的公開金鑰，並且是公開金鑰屬於該實體的陳述式:

- 當憑證適用於個別實體時，該憑證稱為 個人憑證 或 使用者憑證。
- 當憑證是用於「憑證管理中心」時，該憑證稱為 *CA* 憑證 或 簽章者憑證。

如果公開金鑰由其擁有者直接傳送至另一個實體，則可能會截取訊息，而公開金鑰會被另一個實體替代。 這稱為 中間人攻擊。 解決此問題的方法是透過具公信力第三者交換公開金鑰，讓您強力保證公開金鑰確實屬於與您通訊的實體。 您要求具公信力的協力廠商將公開金鑰納入數位憑證中，而不是直接傳送公開金鑰。 發出數位憑證的具公信力第三者稱為「憑證管理中心 (CA)」，如 第 10 頁的『憑證管理中心』中所述。

數位憑證中的內容

數位憑證包含 X.509 標準所決定的特定資訊片段。

IBM MQ 所使用的數位憑證符合 X.509 標準，該標準指定所需的資訊以及傳送它的格式。 X.509 是 X.500 系列標準的鑑別架構部分。

數位憑證至少包含下列所認證實體的相關資訊:

- 擁有者的公開金鑰
- 擁有者的識別名稱
- 發出憑證之 CA 的識別名稱
- 憑證開始有效的日期
- 憑證的到期日
- 憑證資料格式的版本號碼，如 X.509 中所定義。 X.509 標準的現行版本是第 3 版，且大部分憑證都符合該版本。
- 序號。 這是由發出憑證的 CA 指派的唯一 ID。 在發出憑證的 CA 內，序號是唯一的: 相同 CA 憑證所簽署的兩個憑證都沒有相同的序號。

X.509 第 2 版憑證也包含「發證者 ID」和「主旨 ID」， X.509 第 3 版憑證可以包含一些延伸。 部分憑證延伸 (例如「基本限制」延伸) 是 標準，但其他憑證延伸是實作特有的。 延伸可以是 重要，在此情況下，系統必須能夠辨識欄位; 如果無法辨識欄位，則必須拒絕憑證。 如果延伸不重要，則系統可以忽略它 (如果它無法辨識的話)。

個人憑證中的數位簽章是使用簽署該憑證之 CA 的私密金鑰所產生。 任何需要驗證個人憑證的人都可以使用 CA 的公開金鑰來執行此動作。 CA 的憑證包含其公開金鑰。

數位憑證不包含您的私密金鑰。 您必須保持私密金鑰的秘密。

個人憑證的需求

IBM MQ 支援符合 X.509 標準的數位憑證。 它需要用戶端鑑別選項。

因為 IBM MQ 是對等式系統，所以在 SSL/TLS 術語中被視為用戶端鑑別。 因此，用於 SSL/TLS 鑑別的任何個人憑證都需要容許使用用戶端鑑別的金鑰。 並非所有伺服器憑證都已啟用此選項，因此憑證提供者可能需要針對安全憑證在主要 CA 上啟用用戶端鑑別。

除了指定數位憑證資料格式的標準之外，還有用來判斷憑證是否有效的標準。 這些標準已經過一段時間的更新，以防止某些類型的安全侵害。 例如，較舊的 X.509 第 1 版及第 2 版憑證未指出憑證是否可合法用來簽署其他憑證。 因此，惡意使用者可能從合法來源取得個人憑證，並建立設計用來假冒其他使用者的新憑證。

使用 X.509 第 3 版憑證時，會使用 BasicConstraints 和 KeyUsage 憑證延伸來指定哪些憑證可以合法簽署其他憑證。 IETF RFC 5280 標準指定一系列憑證驗證規則，符合標準的應用軟體必須實作這些規則，才能防止假冒攻擊。 一組憑證規則稱為憑證驗證原則。

如需 IBM MQ 中憑證驗證原則的相關資訊，請參閱 第 36 頁的『IBM MQ 中的憑證驗證原則』。

憑證管理中心

「憑證管理中心 (CA)」是具公信力的協力廠商，可發出數位憑證，以確保實體的公開金鑰真正屬於該實體。

CA 的角色如下:

- 在接收數位憑證的要求時，在建置、簽署及傳回個人憑證之前驗證要求者的身分
- 在 CA 憑證中提供 CA 自己的公開金鑰
- 發佈在「憑證撤銷清冊 (CRL)」中不再受信任的憑證清單。 如需相關資訊，請參閱第 282 頁的『使用已撤銷的憑證』。
- 透過操作 OCSP 回應者伺服器來提供憑證撤銷狀態的存取權

識別名稱

識別名稱 (DN) 可唯一識別 X.509 憑證中的實體。

小心: 在 SSLPEER 過濾器中只能使用下表中的屬性。 憑證 DN 可以包含其他屬性，但這些屬性不容許過濾。

表 *1:* 在 *DN* 中找到可在 *SSLPEER* 過濾器中使用的屬性類型

屬性類型	**說明**
SERIALNUMBER	憑證序號
MAIL	電子郵件位址
E	電子郵件位址（已淘汰，最好使用 MAIL）
UID 或 USERID	使用者 ID
CN	通用名稱
T	標題
OU	組織單位名稱
DC	網域元件
O	組織名稱
STREET	街道/地址的第一行
L	地區名稱
ST（或 SP、S）	州/省（縣/市）名稱
PC	郵遞區號
C	國家/地區
UNSTRUCTUREDNAME	主機名稱
UNSTRUCTUREDADDRESS	IP 位址
DNQ	識別名稱限定元

X.509 標準定義的其他屬性通常不是 DN 的一部分，但可以提供數位憑證的選用延伸。

X.509 標準提供以字串格式指定的 DN。 例如:

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

「通用名稱 (CN)」可以說明個別使用者或任何其他實體，例如 Web 伺服器。

DN 可以包含多個 OU 及 DC 屬性。 每一個其他屬性只允許一個實例。 組織單位項目的順序很重要: 訂單指定組織單位名稱的階層，並優先使用最高層次的單位。 DC 項目的順序也很重要。

IBM MQ 容許某些形態異常的 DN。 如需相關資訊，請參閱 IBM MQ SSLPEER 值的規則。

相關概念
第 9 頁的『數位憑證中的內容』
數位憑證包含 X.509 標準所決定的特定資訊片段。

從憑證管理中心取得個人憑證
您可以從授信外部憑證管理中心 (CA) 取得憑證。

您可以透過將資訊以憑證申請形式傳送至 CA 來取得數位憑證。 X.509 標準定義此資訊的格式，但部分 CA 具有自己的格式。 憑證申請通常由系統使用的憑證管理工具產生; 例如:

- Multi 多平台上的 **strmqikm** 指令 (iKeyman 工具) ， 以及 AIX, Linux®, and Windows 上的 **runmqckm** 和 **runmqakm** 指令。
- z/OS z/OS 上的 RACF 。

此資訊包含您的「識別名稱」及公開金鑰。 當您的憑證管理工具產生憑證申請時，它也會產生您必須保持安全的私密金鑰。 永不配送您的私密金鑰。

當 CA 收到您的要求時，憑證管理中心會先驗證您的身分，然後再建置憑證並將它作為個人憑證傳回給您。

第 11 頁的圖 3 說明從 CA 取得數位憑證的程序。

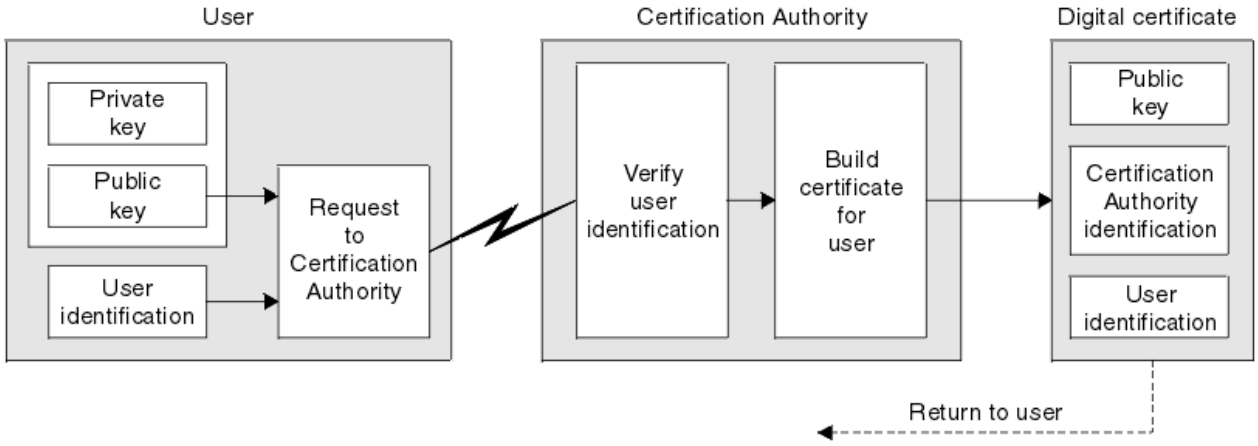


圖 *3:* 取得數位憑證

在圖表中:

- 使用者識別包括您的「主旨識別名稱」。
- 「憑證管理中心」識別包括發出憑證之 CA 的「識別名稱」。

數位憑證包含圖表中所顯示欄位以外的其他欄位。 如需數位憑證中其他欄位的相關資訊，請參閱 第 9 頁的『數位憑證中的內容』。

憑證鏈如何運作
當您收到另一個實體的憑證時，可能需要使用 憑證鏈 來取得 主要 *CA* 憑證。

憑證鏈 (也稱為 憑證路徑) 是用來鑑別實體的憑證清單。 鏈或路徑以該實體的憑證開始，且鏈中的每一個憑證都由鏈中下一個憑證所識別的實體簽署。 鏈結會以主要 CA 憑證終止。 主要 CA 憑證一律由憑證管理中心 (CA) 本身簽署。 在達到主要 CA 憑證之前，必須驗證鏈中所有憑證的簽章。

第 12 頁的圖 4 說明從憑證擁有者到主要 CA 的憑證路徑，其中信任鏈開始。

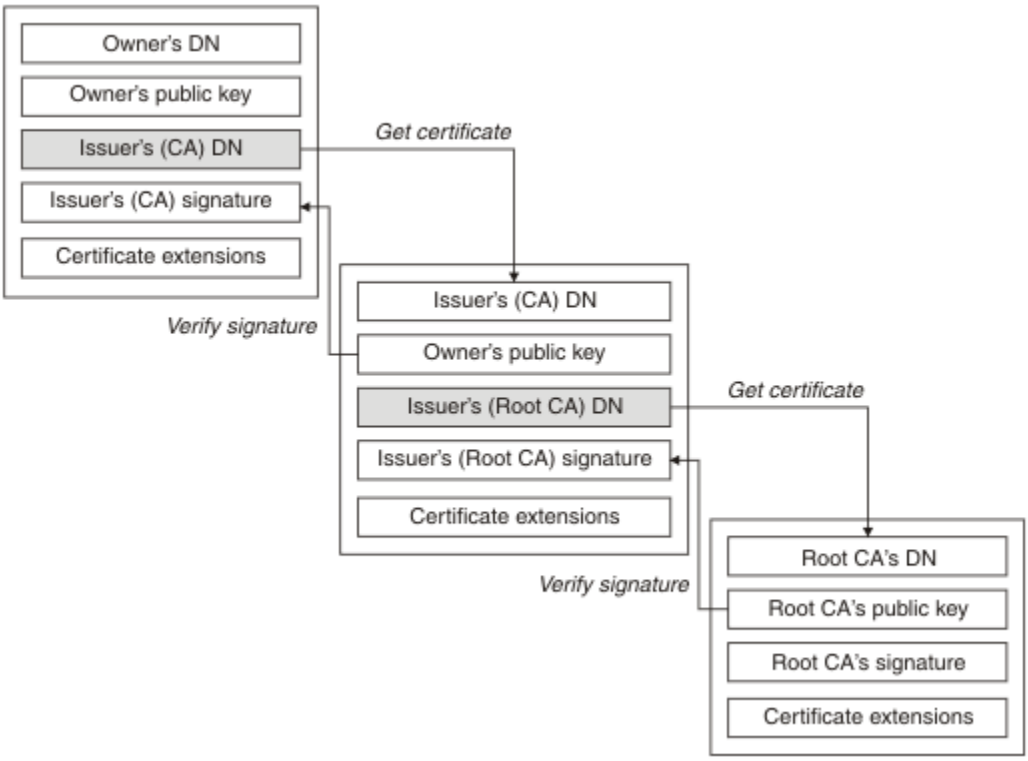


圖 *4:* 信任鏈

每一個憑證可以包含一或多個延伸。 屬於 CA 的憑證通常包含 BasicConstraints 延伸，並設定 isCA 旗標以指出容許它簽署其他憑證。

當憑證不再有效時
數位憑證可以到期或撤銷。

數位憑證會在固定期間內發出，且在到期日期之後無效。

憑證可以因各種原因而撤銷，包括:

- 擁有者已移至不同的組織。
- 私密金鑰不再是秘密金鑰。

IBM MQ 可以將要求傳送至「線上憑證狀態通訊協定 (OCSP)」回應端 (僅限 AIX, Linux, and Windows) ， 以檢查憑證是否已撤銷。 或者，他們可以存取 LDAP 伺服器上的「憑證撤銷清冊 (CRL)」。 OCSP 撤銷及 CRL 資訊由「憑證管理中心」發佈。 如需相關資訊，請參閱 第 282 頁的『使用已撤銷的憑證』。

公開金鑰基礎架構 *(PKI)*

「公開金鑰基礎架構 (PKI)」是一種機能、原則及服務的系統，支援使用公開金鑰加密法來鑑別交易中涉及的各方。

沒有定義「公開金鑰基礎架構」元件的單一標準，但 PKI 通常包含憑證管理中心 (CA) 及註冊管理中心 (RAs)。 CA 提供下列服務:

- 發出數位憑證
- 驗證數位憑證
- 撤銷數位憑證
- 配送公開金鑰

X.509 標準提供業界標準「公開金鑰基礎架構」的基礎。

如需數位憑證及憑證管理中心 (CA) 的相關資訊，請參閱 第 9 頁的『數位憑證』 。 當要求數位憑證時，請驗證所提供的資訊。 如果 RA 驗證該資訊，則 CA 可以向要求者發出數位憑證。

PKI 也可以提供工具來管理數位憑證和公開金鑰。 PKI 有時稱為用來管理數位憑證的 信任階層 ，但大部分定義都包含其他服務。 有些定義包括加密和數位簽章服務，但這些服務並不是 PKI 作業的必要項目。

加密安全通訊協定: TLS

加密通訊協定提供安全連線，可讓雙方以隱私權及資料完整性進行通訊。 「傳輸層安全 (TLS)」通訊協定是從 Secure Sockets Layer (SSL) 通訊協定發展而來。 IBM MQ 支援 TLS。

這兩種通訊協定的主要目標是提供機密性 (有時稱為 隱私權)、資料完整性、識別及使用數位憑證進行鑑別。

雖然這兩個通訊協定類似，但差異足以讓 SSL 3.0 和各種 TLS 版本無法交互作業。

相關概念

第 20 頁的『IBM MQ 中的 TLS 安全通訊協定』
IBM MQ 支援傳輸層安全 (TLS) 通訊協定，以提供訊息通道及 MQI 通道的鏈結層次安全。

傳輸層安全 *(TLS)* 概念

TLS 通訊協定可讓雙方識別及鑑別彼此，並以機密性和資料完整性進行通訊。 TLS 通訊協定是從 Netscape SSL 3.0 通訊協定發展而來，但 TLS 與 SSL 無法交互作業。

TLS 通訊協定提供透過網際網路的通訊安全，並容許主從式應用程式以機密且可靠的方式進行通訊。 這些通訊協定有兩層:「記錄通訊協定」和「信號交換通訊協定」，它們在傳輸通訊協定 (例如 TCP/IP) 之上分層。 它們都使用非對稱和對稱加密法技術。

TLS 連線由應用程式起始，這會變成 TLS 用戶端。 接收連線的應用程式會變成 TLS 伺服器。 每個新的階段作業都以 TLS 通訊協定所定義的信號交換開始。

IBM MQ 支援的 CipherSpecs 完整清單提供於 第 348 頁的『啟用 CipherSpecs』。

如需 SSL 通訊協定的相關資訊，請參閱 https://developer.mozilla.org/docs/Mozilla/Projects/NSS 中提供的資訊。 如需 TLS 通訊協定的相關資訊，請參閱「TLS 工作群組」在「網際網路工程工作小組」網站上提供的資訊，網址為: https://www.ietf.org

SSL/TLS 信號交換的概觀

SSL/TLS 信號交換可讓 TLS 用戶端和伺服器建立它們用來通訊的秘密金鑰。

本節提供可讓 TLS 用戶端和伺服器彼此通訊的步驟摘要。

- 同意要使用的通訊協定版本。
- 選取加密演算法。
- 透過交換及驗證數位憑證來彼此鑑別。
- 使用非對稱加密技術來產生共用秘密金鑰，以避免金鑰配送問題。 然後， TLS 會使用共用金鑰來對訊息進行對稱加密，這比非對稱加密更快。

如需加密演算法及數位憑證的相關資訊，請參閱相關資訊。

在概觀中， TLS 信號交換中涉及的步驟如下:

1. TLS 用戶端會傳送 "client hello" 訊息，其中列出加密資訊 (例如 TLS 版本) ，並依用戶端喜好設定的順序列出用戶端支援的 CipherSuites 。 訊息也包含在後續計算中使用的隨機位元組字串。 通訊協定可讓 "client hello" 併入用戶端支援的資料壓縮方法。
2. TLS 伺服器會回應 "server hello" 訊息，其中包含伺服器從用戶端提供的清單中選擇的 CipherSuite 、階段作業 ID 及另一個隨機位元組字串。 伺服器也會傳送其數位憑證。 如果伺服器需要數位憑證來進行用戶端鑑別，伺服器會傳送 "用戶端憑證申請" ，其中包含支援的憑證類型及可接受憑證管理中心 (CA) 的識別名稱。
3. TLS 用戶端會驗證伺服器的數位憑證。 如需相關資訊，請參閱第 14 頁的『TLS 如何提供識別、鑑別、機密性及完整性』。
4. TLS 用戶端會傳送隨機位元組字串，可讓用戶端及伺服器計算用於加密後續訊息資料的秘密金鑰。 隨機位元組字串本身會以伺服器的公開金鑰加密。

5. 如果 TLS 伺服器傳送 "用戶端憑證申請"，則用戶端會傳送隨機位元組字串 (以用戶端的私密金鑰加密)，以及用戶端的數位憑證，或 "無數位憑證警示"。 此警示只是警告，但在某些實作中，如果用戶端鑑別是必要的，信號交換會失敗。
6. TLS 伺服器會驗證用戶端的憑證。 如需相關資訊，請參閱第 14 頁的『TLS 如何提供識別、鑑別、機密性及完整性』。
7. TLS 用戶端會傳送 "已完成" 訊息給伺服器，以秘密金鑰加密，指出信號交換的用戶端部分已完成。
8. TLS 伺服器會向用戶端傳送 "已完成" 訊息，該訊息已使用秘密金鑰加密，指出信號交換的伺服器部分已完成。
9. 在 TLS 階段作業期間，伺服器和用戶端現在可以交換使用共用秘密金鑰對稱加密的訊息。

第 14 頁的圖 5 說明 TLS 信號交換。

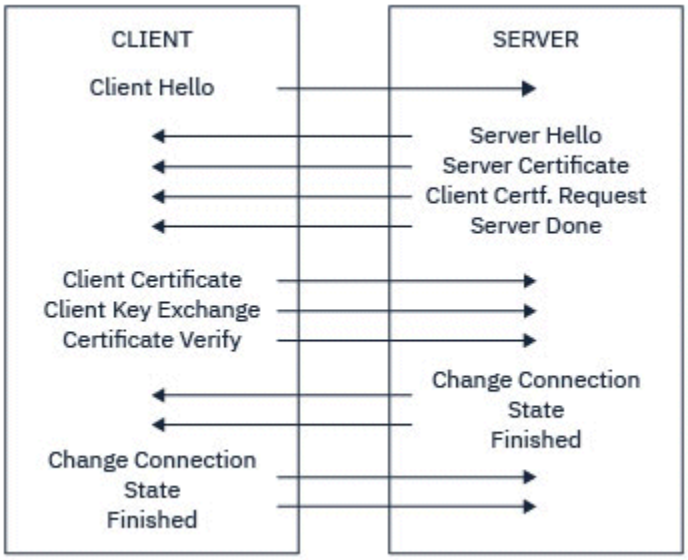


圖 *5: TLS* 信號交換概觀

TLS 如何提供識別、鑑別、機密性及完整性

在用戶端和伺服器鑑別期間，有一個步驟需要使用非對稱金鑰配對中的其中一個金鑰來加密資料，並使用該配對中的另一個金鑰來解密資料。 訊息摘要是用來提供完整性。

如需 TLS 信號交換中所涉及步驟的概觀，請參閱 第 13 頁的『SSL/TLS 信號交換的概觀』。

TLS 如何提供鑑別

對於伺服器鑑別，用戶端會使用伺服器的公開金鑰來加密用來計算秘密金鑰的資料。 只有在伺服器可以使用正確的私密金鑰來解密資料時，才能產生秘密金鑰。 隨機位元組字串本身會以伺服器的公開金鑰來加密 (概觀中的步驟 第 13 頁的『4』)。

對於用戶端鑑別，伺服器會使用用戶端憑證中的公開金鑰來解密用戶端在信號交換步驟 第 14 頁的『5』 期間傳送的資料。 使用秘密金鑰加密的已完成訊息交換 (概觀中的步驟 第 14 頁的『7』 和 第 14 頁的『8』) 確認鑑別已完成。

如果任何鑑別步驟失敗，則信號交換會失敗，且階段作業會終止。

在 TLS 信號交換期間交換數位憑證是鑑別程序的一部分。 如需憑證如何針對模擬提供保護的相關資訊，請參閱相關資訊。 所需的憑證如下所示，其中 CA X 會將憑證發出至 TLS 用戶端，而 CA Y 會將憑證發出至 TLS 伺服器:

僅針對伺服器鑑別， TLS 伺服器需要:

- CA Y 發給伺服器的個人憑證
- 伺服器的私密金鑰

及 TLS 用戶端需要:

- CA Y 的 CA 憑證

如果 TLS 伺服器需要用戶端鑑別，則伺服器會使用向用戶端發出個人憑證之 CA (在本例中為 CA X) 的公開金鑰來驗證用戶端的數位憑證，以驗證用戶端的身分。 對於伺服器和用戶端鑑別，伺服器都需要:

- CA Y 發給伺服器的個人憑證
- 伺服器的私密金鑰
- CA X 的 CA 憑證

及用戶端需要:

- CA 發給用戶端的個人憑證 X
- 用戶端的私密金鑰
- CA Y 的 CA 憑證

TLS 伺服器和用戶端可能都需要其他 CA 憑證，才能形成主要 CA 憑證的憑證鏈。 如需憑證鏈的相關資訊，請參閱相關資訊。

憑證驗證期間發生的情況

如概觀的步驟 第 13 頁的『3』 及 第 14 頁的『6』 中所述， TLS 用戶端會驗證伺服器的憑證，而 TLS 伺服器會驗證用戶端的憑證。 此驗證有四個層面:

1. 會檢查數位簽章 (請參閱 第 16 頁的『SSL/TLS 中的數位簽章』)。
2. 已檢查憑證鏈; 您應該具有中繼 CA 憑證 (請參閱 第 11 頁的『憑證鏈如何運作』)。
3. 會檢查到期和啟動日期以及有效期間。
4. 會檢查憑證的撤銷狀態 (請參閱 第 282 頁的『使用已撤銷的憑證』)。

重設秘密金鑰

在 TLS 信號交換期間，會產生 秘密金鑰 ，以加密 TLS 用戶端與伺服器之間的資料。 秘密金鑰用於套用至資料的數學公式中，以將純文字轉換為無法讀取的密文字，並將密文字轉換為純文字。

秘密金鑰是從隨信號交換一起傳送的隨機文字產生，用來將純文字加密成密文字。 在 MAC (訊息鑑別碼) 演算法中也會使用秘密金鑰，用來判斷訊息是否已變更。 如需相關資訊，請參閱第 8 頁的『訊息摘要和數位簽章』。

如果探索到秘密金鑰，則可以從密文中解密訊息的純文字，或者可以計算訊息摘要，容許在不偵測的情況下變更訊息。 即使是複雜的演算法，也可以將所有可能的數學轉換套用至密文，以最終發現純文字。 若要將秘密金鑰毀損時可解密或變更的資料量減至最少，可以定期重新協議秘密金鑰。 當已重新協議秘密金鑰時，無法再使用先前的秘密金鑰來解密使用新秘密金鑰加密的資料。

TLS 如何提供機密性

TLS 使用對稱和非對稱加密的組合來確保訊息隱私。 在 TLS 信號交換期間， TLS 用戶端和伺服器同意將加密演算法和共用秘密金鑰僅用於一個階段作業。 在 TLS 用戶端與伺服器之間傳輸的所有訊息都會使用該演算法及金鑰進行加密，確保訊息即使被截取仍保持私密。 因為 TLS 在傳輸共用秘密金鑰時使用非對稱加密，所以沒有金鑰配送問題。 如需加密技術的相關資訊，請參閱 第 7 頁的『加密法』。

TLS 如何提供完整性

TLS 透過計算訊息摘要來提供資料完整性。 如需相關資訊，請參閱第 397 頁的『訊息的資料完整性』。

如果通道定義中的 CipherSpec 使用 第 348 頁的『啟用 CipherSpecs』中的雜湊演算法，則使用 TLS 可確保資料完整性。

尤其是如果擔心資料完整性，您應該避免選擇雜湊演算法列為「無」的 CipherSpec 。 也強烈建議不要使用 MD5 ，因為這現在已非常舊，而且在大部分實際用途上不再安全。

CipherSpecs 和 *CipherSuites*

加密安全通訊協定必須同意安全連線所使用的演算法。 CipherSpecs 和 CipherSuites 定義演算法的特定組合。

CipherSpec 可識別加密演算法與「訊息鑑別碼 (MAC)」演算法的組合。 TLS 連線的兩端必須同意相同的 CipherSpec ，才能進行通訊。

IBM MQ 支援 TLS1.3 和 TLS1.2 通訊協定及 CipherSpecs。 不過，如果您需要啟用已淘汰的 CipherSpecs，則可以這麼做。

如需下列相關資訊，請參閱 第 348 頁的『啟用 CipherSpecs』：

- IBM MQ 支援的 CipherSpecs
- 如何啟用已淘汰的 SSL 3.0 及 TLS 1.0 CipherSpecs

重要: 在處理 IBM MQ 通道時，您可以使用 CipherSpec。 在處理 Java 通道、 JMS 通道或 MQTT 通道時，您可以指定 CipherSuite。

如需 CipherSpecs 的相關資訊，請參閱 第 348 頁的『啟用 CipherSpecs』。

CipherSuite 是 TLS 連線所使用的加密演算法套組。 套組包含三個不同的演算法:

- 在信號交換期間使用的金鑰交換和鑑別演算法
- 用來加密資料的加密演算法
- MAC (訊息鑑別碼) 演算法，用來產生訊息摘要

套組的每一個元件都有數個選項，但只有在指定 TLS 連線時，某些組合才有效。 有效 CipherSuite 的名稱可定義所使用演算法的組合。 例如， CipherSuite TLS_RSA_WITH_AES_128_CBC_SHA 指定:

- RSA 金鑰交換和鑑別演算法
- AES 加密演算法，使用 128 位元金鑰及密碼區塊鏈結 (CBC) 模式
- SHA-1 訊息鑑別碼 (MAC)

SSL/TLS 中的數位簽章

通過加密訊息的表示來形成數字簽名。 加密會使用簽署人的私密金鑰，為了效率，通常會對訊息摘要而非訊息本身進行操作。

數位簽章會隨著所簽署的資料而不同，與手寫簽章不同，手寫簽章並不取決於所簽署文件的內容。 如果相同實體以數位方式簽署兩個不同的訊息，則兩個簽章會不同，但可以使用相同的公開金鑰 (即簽署訊息之實體的公開金鑰) 來驗證這兩個簽章。

數位簽章處理程序的步驟如下:

1. 傳送端計算訊息摘要，然後使用傳送端的私密金鑰來加密摘要，形成數位簽章。
2. 傳送端以訊息傳送數位簽章。
3. 接收端會使用傳送端的公開金鑰來解密數位簽章，並重新產生傳送端的訊息摘要。
4. 接收端會從接收到的訊息資料計算訊息摘要，並驗證這兩個摘要是否相同。

第 17 頁的圖 6 說明此程序。

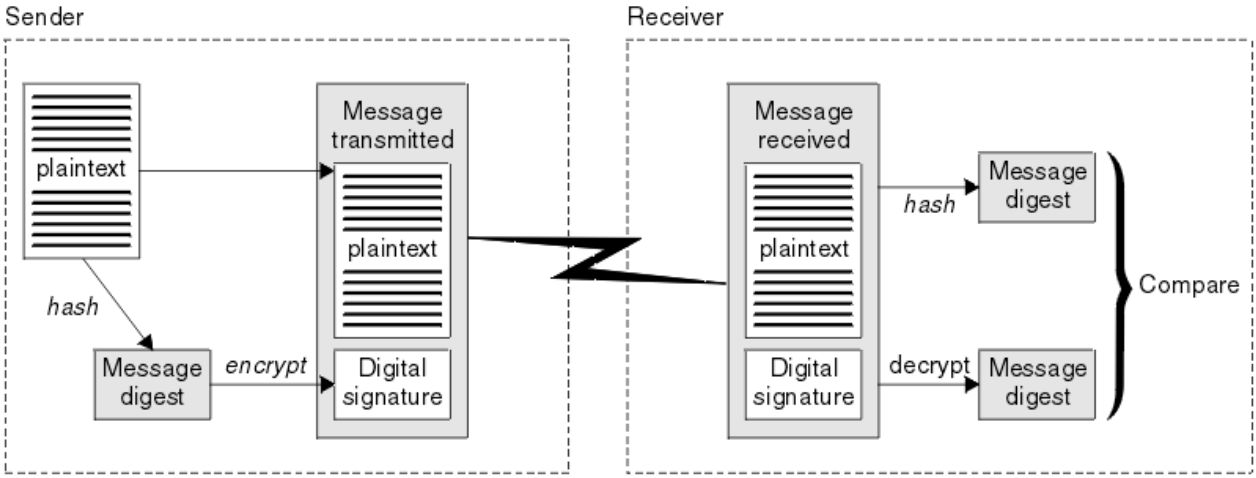


圖 *6:* 數位簽章處理程序

如果已驗證數位簽章，接收端會知道:

- 在傳輸期間未修改訊息。
- 訊息是由要求傳送它的實體所傳送。

數位簽章是完整性和鑑別服務的一部分。 數位簽章也提供來源證明。 只有傳送者知道私密金鑰，這提供有力的證據證明傳送者是訊息的創始者。

註: 您也可以加密訊息本身，以保護訊息中資訊的機密性。

聯邦資訊處理標準

美國政府針對 IT 系統和安全 (包括資料加密) 提供技術建議。 國家標準與技術機構 (NIST) 是涉及 IT 系統和安全的重要機構。 NIST 會產生建議和標準，包括「聯邦資訊存取安全標準 (FIPS)」。

其中一個重要標準是 FIPS 140-2 ，它需要使用強式加密演算法。 FIPS 140-2 也指定雜湊演算法的需求，用來保護封包在傳輸中不受修改。

註: 在 AIX, Linux, and Windows 上， IBM MQ 透過 IBM Crypto for C (ICC) 加密模組提供 FIPS 140-2 相符性。 此模組的憑證已移至「歷程」狀態。 客戶應該檢視 IBM Crypto for C (ICC) 憑證 ， 並注意 NIST 提供的任何建議。 目前正在進行取代 FIPS 140-3 模組，您可以在 處理程序清單中的 NIST CMVP 模組中搜尋它，以檢視其狀態。

IBM MQ 提供 FIPS 140-2 支援 (已配置為這樣做)。

一段時間後，分析師會針對現有加密和雜湊演算法開發攻擊。 採用新的演算法來對抗這些攻擊。 FIPS 140-2 會定期更新，以考量這些變更。

相關概念

第 17 頁的『國家安全域性 (NSA) Suite B 加密法』
美利堅合眾國政府就包括資料加密在內的 IT 系統和安全問題提供技術諮詢。 美國國家安全域性 (NSA) 在其 Suite B 標準中建議一組可交互作業的加密演算法。

國家安全域性 *(NSA) Suite B* 加密法

美利堅合眾國政府就包括資料加密在內的 IT 系統和安全問題提供技術諮詢。 美國國家安全域性 (NSA) 在其 Suite B 標準中建議一組可交互作業的加密演算法。

Suite B 標準指定僅使用一組特定安全加密演算法的作業模式。 「套組 B」標準指定:

- 加密演算法 (AES)
- 金鑰交換演算法 (橢圓曲線 Diffie-Hellman ， 也稱為 ECDH)
- 數位簽章演算法 (橢圓曲線數位簽章演算法，也稱為 ECDSA)
- 雜湊演算法 (SHA-256 或 SHA-384)

此外， IETF RFC 6460 標準還指定 Suite B 相容設定檔，這些設定檔定義符合 Suite B 標準所需的詳細應用程式配置和行為。 它定義兩個設定檔:

1. 與 TLS 1.2 搭配使用的套組 B 相容設定檔。 針對「套組 B」相容作業進行配置時，只會使用列出的受限加密演算法集。
2. 與 TLS 1.0 或 TLS 1.1 搭配使用的過渡設定檔。 此設定檔啟用與非 Suite B 相容伺服器的交互作業能力。 針對套組 B 轉移作業配置時，可能會使用其他加密及雜湊演算法。

「套組 B」標準在概念上類似於 FIPS 140-2 ，因為它會限制已啟用的加密演算法集，以提供安全的保證層次。

在 AIX, Linux, and Windows 系統上， IBM MQ 可以配置為符合 Suite B 相容 TLS 1.2 設定檔，但不支援 Suite B 過渡設定檔。 如需進一步資訊，請參閱第 34 頁的『IBM MQ 中的 NSA Suite B 加密法』。

相關參考

第 17 頁的『聯邦資訊處理標準』
美國政府針對 IT 系統和安全 (包括資料加密) 提供技術建議。 國家標準與技術機構 (NIST) 是涉及 IT 系統和安全的重要機構。 NIST 會產生建議和標準，包括「聯邦資訊存取安全標準 (FIPS)」。

IBM MQ 安全機制

此主題集合說明如何在 IBM MQ 中實作各種安全概念。

IBM MQ 提供機制來實作 第 5 頁的『安全概念和機制』中引進的所有安全概念。 下列各節將更詳細地討論這些問題。

IBM MQ 中的識別及鑑別

在 IBM MQ 中，您可以使用訊息環境定義資訊及交互鑑別來實作識別及鑑別。

以下是 IBM MQ 環境中的一些識別及鑑別範例:

- 每一則訊息都可以包含 訊息環境定義 資訊。 此資訊保留在訊息描述子中。 當應用程式將訊息放入佇列時，佇列管理程式可以產生訊息。 或者，如果授權與應用程式相關聯的使用者 ID 來提供資訊，則應用程式可以提供此資訊。

  訊息中的環境定義資訊可讓接收端應用程式找出訊息的發送端。 例如，它包含放置訊息的應用程式名稱，以及與應用程式相關聯的使用者 ID。
- 當訊息通道啟動時，通道每一端的訊息通道代理程式 (MCA) 可以鑑別其友機。 此技術稱為 交互鑑別。 對於傳送端 MCA ，它提供保證其即將向其傳送訊息的夥伴是真實的。 對於接收 MCA ，也有類似的保證，它即將接收來自真正夥伴的訊息。

相關概念

第 5 頁的『識別及鑑別』
識別 是指能夠唯一識別系統中執行之系統或應用程式的使用者。 鑑別 是指能夠證明使用者或應用程式真正是該人員或該應用程式所要求的人員。

IBM MQ 中的授權

您可以使用授權來限制特定個人或應用程式在 IBM MQ 環境中可以執行的動作。

以下是 IBM MQ 環境中的一些授權範例:

- 僅容許授權管理者發出指令來管理 IBM MQ 資源。
- 只有在與應用程式相關聯的使用者 ID 已獲授權可連接至佇列管理程式時，才容許應用程式連接至佇列管理程式。
- 容許應用程式只開啟其功能所需的那些佇列。
- 容許應用程式僅訂閱其功能所需的那些主題。
- 容許應用程式只在佇列上執行其功能所需的那些作業。 例如，應用程式可能只需要瀏覽特定佇列上的訊息，而不需要放置或取得訊息。

如需如何設定授權的相關資訊，請參閱 第 72 頁的『規劃授權』及相關聯的子主題。

相關概念

第 6 頁的『授權』

授權 會限制只存取授權使用者及其應用程式，以保護系統中的重要資源。 它可防止未獲授權使用資源或以未獲授權的方式使用資源。

IBM MQ 中的審核

IBM MQ 可以發出事件訊息，以記錄發生異常活動。

以下是 IBM MQ 環境中的一些審核範例:

- 應用程式嘗試開啟未獲授權開啟的佇列。 發出檢測事件訊息。 透過檢查事件訊息，您可以探索發生此嘗試，並可以決定需要採取的動作。
- 應用程式嘗試開啟通道，但嘗試失敗，因為 SSL 不容許連線。 發出檢測事件訊息。 透過檢查事件訊息，您可以探索發生此嘗試，並可以決定需要採取的動作。

相關概念

第 6 頁的『審核』

審核 是記錄及檢查事件的處理程序，以偵測是否發生任何非預期或未獲授權的活動，或是否嘗試執行此類活動。

IBM MQ 中的機密性

您可以透過加密訊息，在 IBM MQ 中實作機密性。

在 IBM MQ 環境中可以確保機密性，如下所示:

- 在傳送端 MCA 從傳輸佇列取得訊息之後， IBM MQ 會使用 TLS 來加密訊息，然後再透過網路將訊息傳送至接收端 MCA。 在通道的另一端，在接收 MCA 將訊息放入其目的地佇列之前，訊息會先解密。
- 當訊息儲存在本端佇列時， IBM MQ 所提供的存取控制機制可能被視為足以保護其內容免於未獲授權的揭露。 不過，為了提高安全層次，您可以使用 Advanced Message Security 來加密儲存在佇列中的訊息。
- z/OS V 9.2.0 儲存在本端佇列上的訊息可以使用 z/OS 資料集加密進行靜態加密。

  請參閱 IBM MQ for z/OS 上具有資料集加密之靜態資料的機密性 一節。 的文件以取得相關資訊。

相關概念

第 6 頁的『機密性』

機密性 服務可保護機密性資訊免遭未獲授權的揭露。

IBM MQ 中的資料完整性

您可以使用資料完整性服務來偵測訊息是否已修改。

在 IBM MQ 環境中可以確保資料完整性，如下所示:

- 您可以使用 TLS 來偵測在透過網路傳輸訊息時是否故意修改訊息內容。 在 TLS 中，訊息摘要演算法提供對傳輸中已修改訊息的偵測。

  所有 IBM MQ CipherSpecs 都提供訊息摘要演算法，但 TLS_RSA_WITH_NULL_NULL 除外，它不提供訊息資料完整性。

  IBM MQ 會在接收已修改的訊息時偵測這些訊息; 在接收已修改的訊息時， IBM MQ 會擲出 AMQ9661 錯誤訊息，且通道會停止。
- 當訊息儲存在本端佇列時， IBM MQ 所提供的存取控制機制可能被視為足以防止故意修改訊息內容。

  不過，為了更安全的層次，您可以使用 Advanced Message Security 來偵測在將訊息放入佇列與從佇列擷取訊息之間，是否刻意修改訊息內容。

  偵測到已修改的訊息時，嘗試接收訊息的應用程式會收到 2063 回覆碼，如果使用 MQGET 呼叫，則會將訊息移至 SYSTEM.PROTECTION.ERROR.QUEUE

相關概念

第 6 頁的『資料完整性』

資料完整性 服務會偵測是否有未獲授權的資料修改。

IBM MQ 中的加密法

IBM MQ 使用「傳輸安全層 (TLS)」通訊協定來提供加密法。

如需相關資訊，請參閱 第 20 頁的『IBM MQ 中的 TLS 安全通訊協定』。

相關概念

第 7 頁的『加密概念』
此主題集合說明適用於 IBM MQ 的加密法概念。

IBM MQ 中的 TLS 安全通訊協定

IBM MQ 支援傳輸層安全 (TLS) 通訊協定，以提供訊息通道及 MQI 通道的鏈結層次安全。

訊息通道及 MQI 通道可以使用 TLS 通訊協定來提供鏈結層次安全。 呼叫端 MCA 是 TLS 用戶端，而回應端 MCA 是 TLS 伺服器。

V 9.2.0 IBM MQ 支援 TLS 通訊協定的 1.2 和 1.3 版。 依預設，不會啟用舊版 TLS 及 SSL，但可以在必要時啟用。 您可以提供 CipherSpec 作為通道定義的一部分，以指定 TLS 通訊協定所使用的加密演算法。

V 9.2.0 請參閱 第 348 頁的『啟用 CipherSpecs』，以取得 IBM MQ 所支援的 CipherSpecs 清單，以及參閱 第 361 頁的『已淘汰 CipherSpecs』，以取得已淘汰的 CipherSpec 清單。

您可以使用 SECPROT 及 SSLCIPH 參數來顯示通道上使用的安全通訊協定及 CipherSpec 。

在訊息通道的每一端，以及在 MQI 通道的伺服器端， MCA 會代表它所連接的佇列管理程式執行動作。 在 TLS 信號交換期間， MCA 會將佇列管理程式的數位憑證傳送至通道另一端的友機 MCA。 MQI 通道用戶端端的 IBM MQ 程式碼代表 IBM MQ 用戶端應用程式的使用者運作。 在 TLS 信號交換期間， IBM MQ 程式碼會將使用者的數位憑證傳送至 MQI 通道伺服器端的 MCA。

當佇列管理程式和 IBM MQ 用戶端使用者作為 TLS 用戶端時，不需要它們有相關聯的個人數位憑證，除非在通道的伺服器端指定 SSLCAUTH (REQUIRED)。

數位憑證儲存在 金鑰儲存庫中。 佇列管理程式屬性 **SSLKeyRepository** 指定存放佇列管理程式數位憑證之金鑰儲存庫的位置。 在 IBM MQ 用戶端系統上， MQSSLKEYR 環境變數指定保留使用者數位憑證之金鑰儲存庫的位置。 或者， IBM MQ 用戶端應用程式可以在 MQCONNX 呼叫 TLS 配置選項結構 MQSCO 的 **KeyRepository** 欄位中指定其位置。 如需金鑰儲存庫及如何指定其位置的相關資訊，請參閱相關主題。

支援 TLS

V 9.2.0 IBM MQ 提供所有平台上的 TLS 1.2 及 TLS 1.3 支援。 如需 TLS 通訊協定的相關資訊，請參閱子主題中的資訊。

Java 和 JMS 用戶端
這些用戶端使用 JVM 來提供 TLS 支援。

AIX, Linux, and Windows
TLS 支援隨 IBM MQ 一起安裝。

IBM i
TLS 支援是 IBM i 作業系統不可或缺的。

z/OS
TLS 支援是 z/OS 作業系統不可或缺的。 z/OS 上的 TLS 支援稱為 系統 *SSL*。

如需 IBM MQ TLS 支援之任何必要條件的相關資訊，請參閱 系統需求 IBM MQ。

相關概念

第 13 頁的『加密安全通訊協定: TLS』
加密通訊協定提供安全連線，可讓雙方以隱私權及資料完整性進行通訊。 「傳輸層安全 (TLS)」通訊協定是從 Secure Sockets Layer (SSL) 通訊協定發展而來。 IBM MQ 支援 TLS。

SSL/TLS 金鑰儲存庫

相互鑑別的 TLS 連線在連線的每一端都需要一個金鑰儲存庫。 金鑰儲存庫包括數位憑證及私密金鑰。

此資訊使用一般術語 金鑰儲存庫 來說明數位憑證及其相關聯私密金鑰的儲存庫。 在支援 TLS 的不同平台和環境上，會以不同名稱來參照金鑰儲存庫:

- IBM i 在 IBM i 上: *certificate store*
- 在 Java 和 JMS 上: *keystore* 和 信任儲存庫
- ALW 在 AIX, Linux, and Windows 上: *key database file*
- z/OS 在 z/OS 上: *keyring*

如需相關資訊，請參閱第 9 頁的『數位憑證』和第 13 頁的『傳輸層安全 (TLS) 概念』。

相互鑑別的 TLS 連線在連線的每一端都需要一個金鑰儲存庫。 金鑰儲存庫可以包含下列憑證和要求:

- 來自各種憑證管理中心的許多 CA 憑證，可讓佇列管理程式或用戶端驗證它在連線遠端從其友機接收的憑證。 個別憑證可能在憑證鏈中。
- 從「憑證管理中心」收到一或多個個人憑證。 您可以將個別個人憑證與每一個佇列管理程式或 IBM MQ MQI client 相關聯。 如果需要交互鑑別，則個人憑證在 TLS 用戶端上是必要的。 如果不需要交互鑑別，則用戶端上不需要個人憑證。 金鑰儲存庫也可能包含對應於每個個人憑證的私密金鑰。
- 等待授信 CA 憑證簽署的憑證申請。

如需保護金鑰儲存庫的相關資訊，請參閱第 22 頁的『保護 IBM MQ 金鑰儲存庫』。

金鑰儲存庫的位置視您使用的平台而定:

IBM i

金鑰儲存庫是憑證儲存庫。 預設系統憑證儲存庫位於整合檔案系統 (IFS) 中的 `/QIBM/UserData/ICSS/Cert/Server/Default`。 IBM MQ 會將憑證儲存庫的密碼儲存在 密碼隱藏檔中。 例如，佇列管理程式 `QM1` 的隱藏檔是 `/QIBM/UserData/mqm/qmgrs/QM1/ssl/Stash.sth`。

或者，您可以指定改用 IBM i 系統憑證儲存庫。 若要執行此動作，請將佇列管理程式 **SSLKEYR** 屬性的值變更為 `*SYSTEM`。 此值指出佇列管理程式必須使用系統憑證儲存庫，且佇列管理程式已登錄為使用「數位 Certificate Manager (DCM)」的應用程式。

憑證儲存庫也包含佇列管理程式的私密金鑰。

AIX, Linux, and Windows 系統

金鑰儲存庫是金鑰資料庫檔。 金鑰資料庫檔的名稱必須具有副檔名 `.kdb`。 例如，在 AIX and Linux 上，佇列管理程式 `QM1` 的預設金鑰資料庫檔為 `/var/mqm/qmgrs/QM1/ssl/key.kdb`。 如果 IBM MQ 安裝在預設位置，則 Windows 上的對等路徑是 `C:\ProgramData\ IBM \ MQ\Qmgrs\QM1\ssl\key.kdb`。

每一個金鑰資料庫檔都有相關聯的密碼隱藏檔。 此檔案保留容許程式存取金鑰資料庫的已編碼密碼。 密碼隱藏檔必須位於相同的目錄中，且與金鑰資料庫具有相同的檔案系統，且必須以字尾 `.sth` 結尾，例如 `/var/mqm/qmgrs/QM1/ssl/key.sth`

註: PKCS #11 加密硬體卡可以包含金鑰資料庫檔中所保留的憑證及金鑰。 在 PKCS #11 卡上保留憑證及金鑰時， IBM MQ 仍需要同時存取金鑰資料庫檔及密碼隱藏檔。

在 AIX, Linux, and Windows 系統上，金鑰資料庫也包含與佇列管理程式或 IBM MQ MQI client 相關聯之個人憑證的私密金鑰。

z/OS

憑證保留在 z/OS 的金鑰環中。

其他外部安全管理程式 (ESM) 也會使用金鑰環來儲存憑證。

私密金鑰由 RACF 管理。

保護 *IBM MQ* 金鑰儲存庫
IBM MQ 的金鑰儲存庫是一個檔案。 請確定只有預期的使用者可以存取金鑰儲存庫檔案。 這可防止侵入者或其他未獲授權的使用者將金鑰儲存庫檔案複製到另一個系統，然後在該系統上設定相同的使用者 ID 來假冒預期的使用者。

檔案的許可權視使用者的 umask 及使用的工具而定。 在 Windows 上， IBM MQ 帳戶需要許可權 `BypassTraverseChecking`，這表示檔案路徑中資料夾的許可權沒有作用。

請檢查金鑰儲存庫檔案的檔案許可權，並確定檔案及包含的資料夾不是全球可讀取的，最好是群組無法讀取。

在您使用的任何系統上，將金鑰儲存庫設為唯讀是良好的作法，只允許管理者啟用寫入作業以執行維護。

實際上，您必須保護所有金鑰儲存庫，無論位置為何，以及它們是否受密碼保護; 保護金鑰儲存庫。

數位憑證標籤，瞭解需求
設定 TLS 以使用數位憑證時，您可能必須遵循特定的標籤需求，視所使用的平台及您用來連接的方法而定。

憑證標籤是什麼?

憑證標籤是代表儲存在金鑰儲存庫中的數位憑證的唯一 ID，並提供在執行金鑰管理功能時用來參照特定憑證的方便人類可讀名稱。 第一次將憑證新增至金鑰儲存庫時，您可以指派憑證標籤。

憑證標籤與憑證的 **Subject Distinguished Name** 或 **Subject Common Name** 欄位分開。 請注意，**Subject Distinguished Name** 和 **Subject Common Name** 是憑證本身內的欄位。 這些是在建立憑證時定義，且無法變更。 不過，必要的話，您可以變更與數位憑證相關聯的標籤。

憑證標籤語法

憑證標籤可以包含具有下列條件的字母、數字及標點符號:

- Multi 憑證標籤最多可以包含 64 個字元。
- z/OS 憑證標籤最多可以包含 32 個字元。
- 憑證標籤可以包含空格。
- 標籤區分大小寫。
- 在使用 EBCDIC katakana 的系統上，您無法使用小寫字元。

下列各節指定憑證標籤值的其他需求。

如何使用憑證標籤?

IBM MQ 使用憑證標籤來尋找在 TLS 信號交換期間傳送的個人憑證。 當金鑰儲存庫中存在多個個人憑證時，這可消除語義不明確。

您可以將憑證標籤設為您選擇的值。 如果您未設定值，則會使用遵循命名慣例的預設標籤，視您使用的平台而定。 如需詳細資料，請參閱下列各節: 特定平台。

附註:

1. 您無法在 Java 或 JMS 系統上自行設定憑證標籤。
2. 通道自動定義 (CHAD) 結束程式所建立的自動定義通道無法設定憑證標籤，因為在建立通道時已發生 TLS 信號交換。 在入埠通道的 CHAD 結束程式中設定憑證標籤沒有作用。

在此環境定義中， TLS 用戶端是指起始信號交換的連線夥伴，可能是 IBM MQ 用戶端或另一個佇列管理程式。

在 TLS 信號交換期間， TLS 用戶端一律會從伺服器取得並驗證數位憑證。 使用 IBM MQ 實作， TLS 伺服器一律從用戶端要求憑證，且用戶端一律提供憑證給伺服器 (如果找到憑證的話)。 如果用戶端找不到個人憑證，用戶端會將 `no certificate` 回應傳送至伺服器。

如果傳送用戶端憑證，則 TLS 伺服器一律會驗證用戶端憑證。 如果用戶端未傳送憑證，則在 **SSLCAUTH** 參數設為 *REQUIRED* 或 **SSLPEER** 參數值集的情況下定義作為 TLS 伺服器的通道結尾，鑑別會失敗。

請注意，只有在遠端對等節點的 IBM MQ 版本完全支援憑證標籤配置，且通道使用 TLS CipherSpec 時，入埠通道 (包括接收端、要求端、叢集接收端、不完整伺服器及伺服器連線通道) 才會傳送已配置的憑證。

不完整的伺服器通道是未設定 CONNAME 欄位的伺服器通道。

在所有其他情況下，佇列管理程式 **CERTLABL** 參數會決定傳送的憑證。 尤其是，不論通道特定標籤設定為何，下列項目只會接收佇列管理程式的 **CERTLABL** 參數所配置的憑證:

- Java 及 JMS 用戶端支援「伺服器名稱指示 (SNI)」，亦即，以通道為基礎的憑證。
- IBM MQ 8.0 之前的 IBM MQ 版本。
- 受管理 .NET 用戶端

此外，通道使用的憑證必須適用於通道 CipherSpec -如需進一步資訊，請參閱 第 37 頁的『IBM MQ 中的數位憑證及 CipherSpec 相容性』。

IBM MQ 8.0 以及更新版本支援在相同佇列管理程式上使用多個憑證，使用通道定義中使用 **CERTLABL** 屬性指定的每個通道憑證標籤。 佇列管理程式 (例如，伺服器連線或接收端) 的入埠通道依賴使用「TLS 伺服器名稱指示 (SNI)」來偵測通道名稱，以便從佇列管理程式提供正確的憑證。 如需在佇列管理程式上使用多個憑證的相關資訊，請參閱 第 24 頁的『IBM MQ 如何提供多個憑證功能』。

如果通道透過 IBM MQ Internet Pass-Thru (MQIPT) 連接至目的地佇列管理程式，且 MQIPT 路徑同時設定 **SSLServer** 及 **SSLClient**，則端點之間會有兩個個別的 TLS 階段作業。 在早於 IBM MQ 9.2.5 的版本中， SNI 資料不會在階段作業岔斷之間流動。 這會防止在目的地佇列管理程式上使用個別通道憑證，以進行 MQIPT 與佇列管理程式之間的 TLS 連線。 V 9.2.5 從 IBM MQ 9.2.5 開始， MQIPT 可以配置為容許目的地佇列管理程式使用多個憑證，方法是將 SNI 設定為通道名稱，或將在入埠連線上接收到的 SNI 傳遞至路徑。 如需多個憑證支援及 MQIPT 的相關資訊，請參閱 IBM MQ 使用 MQIPT 的多個憑證支援。

如需使用單向鑑別來連接佇列管理程式的相關資訊，即 TLS 用戶端未傳送憑證時，請參閱 使用單向鑑別來連接兩個佇列管理程式。

多平台系統

Multi

在 多平台上， TLS 伺服器會將憑證傳送至用戶端。

對於佇列管理程式及用戶端，會依序搜尋下列來源，以找出非空白值。 第一個非空白值決定憑證標籤。 憑證標籤必須存在於金鑰儲存庫中。 如果找不到符合標籤的正確大小寫及格式相符憑證，則會發生錯誤，且 TLS 信號交換失敗。

佇列管理程式

1. 通道憑證標籤屬性 **CERTLABL**。
2. 佇列管理程式憑證標籤屬性 **CERTLABL**。
3. 預設值，格式如下: `ibmwebspheremq`，並附加佇列管理程式名稱，全部都是小寫。 例如，對於名為 `QM1` 的佇列管理程式，預設憑證標籤為 `ibmwebspheremqqm1`。

IBM MQ 用戶端

1. CLNTCONN 通道定義中的憑證標籤屬性 **CERTLABL**。
2. MQSCO 結構 **CertificateLabel** 屬性。
3. 環境變數 **MQCERTLABL**。
4. 用戶端 `.ini` 檔 (在其 SSL 區段中) **CertificateLabel** 屬性
5. 預設值，格式如下: `ibmwebspheremq`，附加用戶端應用程式執行的使用者 ID (全部為小寫)。 例如，對於使用者 ID `USER1`，預設憑證標籤為 `ibmwebspheremquser1`。

z/OS 系統

z/OS

IBM MQ 在 z/OS 上不支援用戶端。 不過， z/OS 佇列管理程式可以在起始連線時扮演 TLS 用戶端的角色，或在接受連線要求時扮演 TLS 伺服器的角色。 z/OS 佇列管理程式的憑證標籤需求同時適用於這兩個角色，且不同於 多平台上的需求。

對於佇列管理程式及用戶端，會依序搜尋下列來源，以找出非空白值。 第一個非空白值決定憑證標籤。 憑證標籤必須存在於金鑰儲存庫中。 如果找不到符合標籤的正確大小寫及格式相符憑證，則會發生錯誤，且 TLS 信號交換失敗。

1. 通道憑證標籤屬性 **CERTLABL**。
2. 如果共用，則為佇列共用群組憑證標籤屬性 **CERTQSGL**。

   如果未共用，則為佇列管理程式憑證標籤屬性 **CERTLABL**。
3. 預設值，格式如下: `ibmWebSphereMQ` ，並附加佇列管理程式或佇列共用群組的名稱。 請注意，此字串區分大小寫，且必須如下所示撰寫。 例如，對於名為 `QM1` 的佇列管理程式，預設憑證標籤為 `ibmWebSphereMQQM1`。
4. 如果在選項 第 24 頁的『3』中找不到格式為的憑證， IBM MQ 會嘗試使用在金鑰環中標示為預設值的憑證。

如需如何顯示金鑰儲存庫的相關資訊，請參閱 第 268 頁的『在 z/OS 上尋找佇列管理程式的金鑰儲存庫』。

IBM MQ Java 和 IBM MQ JMS 用戶端

IBM MQ Java 和 IBM MQ JMS 用戶端在 TLS 信號交換期間使用其 Java Secure Socket Extension (JSSE) 提供者的機能來選取個人憑證，因此不受憑證標籤需求的約束。

預設行為是 JSSE 用戶端反覆運算金鑰儲存庫中的憑證，並選取第一個找到可接受的個人憑證。 不過，這個行為只是預設值，取決於 JSSE 提供者的實作。

此外， JSSE 介面可透過應用程式在執行時期進行配置及直接存取，高度自訂。 如需特定詳細資料，請參閱 JSSE 提供者提供的文件。

為了進行疑難排解，或更充分地瞭解 IBM MQ Java 用戶端應用程式與您的特定 JSSE 提供者一起執行的信號交換，您可以在 JVM 環境中設定 `javax.net.debug=ssl` 來啟用除錯。

您可以透過配置或在指令行上輸入 `-Djavax.net.debug=ssl` ，在應用程式內設定變數。

Linux *IBM MQ 如何提供多個憑證功能*

「伺服器名稱指示 (SNI)」是 TLS 通訊協定的延伸，可讓用戶端指出它需要的服務。 在 IBM MQ 術語中，這等同於通道。

IBM MQ 使用 SNI 延伸，容許在通道定義上使用 CERTLABL 參數跨不同通道指定多個憑證。

IBM MQ 使用的 SNI 位址是根據所要求的通道名稱，後面接著字尾 `.chl.mq.ibm.com`。

IBM MQ 通道名稱會對映至有效的 SNI 名稱，如下所示:

- A 至 Z 的大寫字母會摺疊成小寫
- 9 的數字 `0` 保持不變
- 所有其他字元 (包括小寫字母 a 至 z) 都會轉換成兩位數十六進位 ASCII 字元碼 (小寫) ，後面接著連字號。
  - a 至 z 的小寫字母分別對映至十六進位 `61-` 至 `7a-`
  - 百分比 (%) 對映至十六進位 `25-`
  - 連字號 (-) 對映至十六進位 `2d-`
  - 點 (.) 對映至十六進位 `2e-`
  - 正斜線 (/) 對映至十六進位 `2f-`
  - 底線 (_) 對映至十六進位 `5f-`

在 EBCDIC 平台上，在套用此對映之前，通道名稱會轉換成 ASCII。

例如，通道名稱 `TO.QMGR1` 對映至 `to2e-qmgr1.chl.mq.ibm.com` 的 SNI 位址。

相反地，小寫通道名稱 `to.qmgr1` 對映至 `74-6f-2e-71-6d-67-72-1.chl.mq.ibm.com` 的 SNI 位址。

註: 在產生的 SNI URL 必須符合 URL 格式化規格的環境中，例如當用戶端透過 Red Hat® OpenShift® Route 連接至在 Red Hat OpenShift 中執行的佇列管理程式時，通道名稱不得以小寫字母結尾。

SSL 段落的其他 **OutboundSNI** 內容可讓您選取在起始 TLS 連線時，是應該將 SNI 設為遠端系統的目標 IBM MQ 通道名稱，還是設為主機名稱。 如需 **OutboundSNI** 內容的相關資訊，請參閱 qm.ini 檔的 SSL 段落和用戶端配置檔的 SSL 段落，以取得詳細資料。

多個憑證需要將 SNI 設為 IBM MQ 通道名稱。 如果使用主機名稱、自訂或無 SNI 來連接至已配置憑證標籤的 IBM MQ 通道，則會拒絕連接應用程式，並在遠端佇列管理程式錯誤日誌中列印 AMQ9673 訊息。

V 9.2.5 如果通道透過 IBM MQ Internet Pass-Thru (MQIPT) 連接至目的地佇列管理程式，則必須將 MQIPT 配置為將 SNI 設為通道名稱，或透過在入埠連線上接收到的 SNI 傳遞至路徑，以容許目的地佇列管理程式使用多個憑證。 如需多個憑證支援及 MQIPT 的相關資訊，請參閱 IBM MQ 使用 MQIPT 的多個憑證支援。

如需如何使用此內容的相關資訊，請參閱 連接至部署在 Red Hat OpenShift 叢集中的佇列管理程式。

重新整理佇列管理程式的金鑰儲存庫
當您變更金鑰儲存庫的內容時，佇列管理程式不會立即挑選新的內容。 若要讓佇列管理程式使用新的金鑰儲存庫內容，您必須發出 REFRESH SECURITY TYPE (SSL) 指令。

此處理程序是故意的，可防止多個執行中通道可能使用不同版本的金鑰儲存庫。 作為安全控制項，佇列管理程式隨時只能載入金鑰儲存庫的一個版本。

如需 REFRESH SECURITY TYPE (SSL) 指令的相關資訊，請參閱 重新整理安全。

您也可以使用 PCF 指令或 IBM MQ Explorer 來重新整理金鑰儲存庫。 如需相關資訊，請參閱本產品說明文件 IBM MQ Explorer 一節中的 MQCMD_REFRESH_SECURITY 指令 及 重新整理 *TLS* 安全 主題。

相關概念
第 25 頁的『重新整理用戶端的 SSL/TLS 金鑰儲存庫內容及 SSl/TLS 設定視圖』
若要使用金鑰儲存庫的重新整理內容來更新用戶端應用程式，您必須停止並重新啟動用戶端應用程式。

重新整理用戶端的 *SSL/TLS* 金鑰儲存庫內容及 *SSl/TLS* 設定視圖
若要使用金鑰儲存庫的重新整理內容來更新用戶端應用程式，您必須停止並重新啟動用戶端應用程式。

您無法重新整理 IBM MQ 用戶端上的安全; 用戶端沒有同等的 REFRESH SECURITY TYPE (SSL) 指令 (請參閱 REFRESH SECURITY) 以取得相關資訊。

每當您變更安全憑證時，必須停止並重新啟動應用程式，以使用金鑰儲存庫的重新整理內容來更新用戶端應用程式。

如果重新啟動通道會重新整理配置，且您的應用程式有重新連線邏輯，您可以發出 STOP CHL STATUS (INACTIVE) 指令來重新整理用戶端的安全。

相關概念
第 25 頁的『重新整理佇列管理程式的金鑰儲存庫』
當您變更金鑰儲存庫的內容時，佇列管理程式不會立即挑選新的內容。 若要讓佇列管理程式使用新的金鑰儲存庫內容，您必須發出 REFRESH SECURITY TYPE (SSL) 指令。

MQCSP 密碼保護
從 IBM MQ 8.0 開始，您可以使用 IBM MQ 功能來傳送受保護的 MQCSP 結構中包含的密碼，或使用 TLS 加密來加密的密碼。

重要: MQCSP 密碼保護適用於測試及開發目的，因為使用 MQCSP 密碼保護比設定 TLS 加密更簡單，但沒有那麼安全。 基於正式作業目的，您應該優先使用 TLS 加密，而不是 IBM MQ 密碼保護，特別是在用戶端與佇列管理程式之間的網路不受信任時，因為 TLS 加密更安全。

如果您確切關心正在使用的加密，以及它提供的保護程度，則需要使用完整 TLS 加密。 在此狀況下，演算法是公開已知的，您可以使用 **SSLCIPH** 通道屬性為企業選取適當的演算法。

如需 MQCSP 結構的相關資訊，請參閱 MQCSP 結構。

當符合下列所有條件時，會使用密碼保護:

- 連線的兩端都使用 IBM MQ 8.0 或更新版本。

- 通道未使用 TLS 加密。 如果通道具有空白 **SSLCIPH** 屬性，或 **SSLCIPH** 屬性設為不提供加密的 CipherSpec，則通道不會使用 TLS 加密。 空值密碼 (例如 NULL_SHA) 不提供加密。
- 您設定 **MQCSP**。**AuthenticationType** 至 MQCSP_AUTH_USER_ID_AND_PWD。 設定此值可讓您評估更多檢查，以決定是否執行密碼保護。 預設值 **MQCSP**。**AuthenticationType** 是 MQCSP_AUTH_NONE。 使用預設值，不會執行密碼保護。 如需相關資訊，請參閱 **AuthenticationType**。
- 如果用戶端是 IBM MQ Explorer，且未啟用使用者識別相容模式，這不是預設值。 此條件僅適用於 IBM MQ Explorer。

如果不符合這些條件，除非 **PasswordProtection** 配置設定禁止，否則會以純文字傳送密碼。

PasswordProtection 配置設定

用戶端及佇列管理程式 .ini 配置檔的「通道」區段中的 **PasswordProtection** 屬性可以防止以純文字傳送密碼。 屬性可以採用下列其中一個值。 預設值為 compatible:

相容
如果佇列管理程式或用戶端執行早於 IBM MQ 8.0 的 IBM MQ 版本，則可以純文字傳送密碼。 也就是說，為了相容性，容許使用純文字密碼。

因此:

- 如果使用 TLS 加密且 CipherSpec 不是空值，則由 TLS CipherSpec 加密傳送密碼。
- 如果佇列管理程式或用戶端正在執行早於 IBM MQ 8.0 的 IBM MQ 版本，且未使用 TLS 加密，則會以純文字傳送密碼。 密碼以純文字傳送，因為 IBM MQ 8.0 之前的 IBM MQ 版本只能以純文字傳送密碼。
- 如果佇列管理程式及用戶端都執行 IBM MQ 8.0 或更新版本的 IBM MQ，且使用空值 CipherSpec 或未使用 TLS 加密，則密碼會受到傳送保護。 **MQCSP**。**AuthenticationType** 必須設為 MQCSP_AUTH_USER_ID_AND_PWD。
- 如果佇列管理程式和用戶端都在執行 IBM MQ 8.0 或更新版本的 IBM MQ 版本，以及 **MQCSP**，則在傳送密碼之前連線會失敗。**AuthenticationType** 未設為 MQCSP_AUTH_USER_ID_AND_PWD。

一律
密碼必須使用非空值 CipherSpec 或 **MQCSP** 的 CipherSpec 進行加密。**AuthenticationType** 必須設為 MQCSP_AUTH_USER_ID_AND_PWD。 否則，連線會失敗。 亦即，不容許純文字密碼。

因此:

- 如果使用 TLS 加密且 CipherSpec 不是空值，則由 TLS CipherSpec 加密傳送密碼。
- 如果佇列管理程式和用戶端都執行 IBM MQ 8.0 或更新版本的 IBM MQ，且未使用 TLS 加密，或使用空值 CipherSpec，則密碼會受到保護。 **MQCSP**。**AuthenticationType** 必須設為 MQCSP_AUTH_USER_ID_AND_PWD。
- 如果佇列管理程式或用戶端正在執行早於 IBM MQ 8.0 的 IBM MQ 版本，且未使用 TLS 加密，則在傳送密碼之前連線會失敗。 由於 IBM MQ 8.0 之前的 IBM MQ 版本只能以純文字傳送密碼，且 always 需要密碼加密或受保護，因此連線會失敗。

選用
密碼可以選擇性地受傳送保護，但如果 **MQCSP** 則以純文字傳送。**AuthenticationType** 未設為 MQCSP_AUTH_USER_ID_AND_PWD。 也就是說，任何用戶端都可以傳送純文字密碼。

因此:

- 如果使用 TLS 加密且 CipherSpec 不是空值，則由 TLS CipherSpec 加密傳送密碼。
- 如果使用空值 CipherSpec 及 **MQCSP**，則會以純文字傳送密碼。**AuthenticationType** 未設為 MQCSP_AUTH_USER_ID_AND_PWD。
- 如果佇列管理程式或用戶端正在執行早於 IBM MQ 8.0 的 IBM MQ 版本，且未使用 TLS 加密，則會以純文字傳送密碼。 密碼以純文字傳送，因為 IBM MQ 8.0 之前的 IBM MQ 版本只能以純文字傳送密碼。

- 如果佇列管理程式及用戶端都在 IBM MQ 8.0 或更新版本執行 IBM MQ 版本、未使用 TLS 加密或使用空值 CipherSpec，以及 **MQCSP**，則密碼會受到傳送保護。**AuthenticationType** 設為 MQCSP_AUTH_USER_ID_AND_PWD。

WARN
任何用戶端都可以傳送純文字密碼。 如果收到純文字密碼，則會將警告訊息 (AMQ9297) 寫入佇列管理程式錯誤日誌中。

對於 Java 和 JMS 用戶端， **PasswordProtection** 屬性的行為會根據使用相容模式或 MQCSP 模式的選擇而變更:

- 如果 Java 和 JMS 用戶端以相容模式運作，則在連線處理期間不會傳送 MQCSP 結構。 因此，**PasswordProtection** 屬性的行為與針對執行早於 IBM MQ 8.0 之 IBM MQ 版本的用戶端所說明的行為相同。
- 如果 Java 及 JMS 用戶端以 MQCSP 模式運作，則 **PasswordProtection** 屬性的行為是所說明的行為。

如需使用 Java 及 JMS 用戶端進行連線鑑別的相關資訊，請參閱 第 67 頁的『與 Java 用戶端的連線鑑別』。

數位憑證管理程式 *(digital certificate manager, DCM)*

使用 DCM 來管理 IBM i 上的數位憑證及私密金鑰。

「數位 Certificate Manager (DCM)」可讓您管理數位憑證，並在 IBM i 伺服器上的安全應用程式中使用它們。 使用「數位 Certificate Manager」，您可以向「憑證管理中心 (CA)」或其他協力廠商要求並處理數位憑證。 您也可以充當本端「憑證管理中心」，為您的使用者建立及管理數位憑證。

DCM 也支援使用「憑證撤銷清冊 (CRL)」來提供更強大的憑證及應用程式驗證程序。 您可以使用 DCM 來定義特定憑證管理中心 CRL 在 LDAP 伺服器上的位置，以便 IBM MQ 可以驗證尚未撤銷特定憑證。

DCM 支援並可以自動偵測各種格式的憑證。 當 DCM 偵測到 PKCS #12 編碼憑證或包含已加密資料的 PKCS #7 憑證時，它會自動提示使用者輸入用來加密憑證的密碼。 DCM 不會提示輸入未包含已加密資料的 PKCS #7 憑證。

DCM 提供瀏覽器型使用者介面，您可以用來管理應用程式及使用者的數位憑證。 使用者介面分為兩個主要框架: 導覽框架和作業框架。

您可以使用導覽頁框來選取作業，以管理憑證或使用它們的應用程式。 部分個別作業會直接顯示在主要導覽頁框中，但導覽頁框中的大部分作業會組織成種類。 例如，「管理憑證」是作業種類，包含各種個別引導式作業，例如「檢視憑證」、「更新憑證」及「匯入憑證」。 如果導覽頁框中的項目是包含多個作業的種類，則會在其左側顯示箭頭。 箭頭指出當您選取種類鏈結時，會顯示展開的作業清單，可讓您選擇要執行的作業。

如需 DCM 的相關重要資訊，請參閱下列 IBM Redbooks 出版品:

- *IBM i Wired Network Security: OS/400 V5R1 DCM* 及加密加強功能(SG24-6168)。 具體而言，請參閱附錄，以取得將 IBM i 系統設定為本端 CA 的相關重要資訊。
- *AS/400 Internet Security: Developing a Digital Certificate Infrastructure*(SG24-5659)。 具體而言，請參閱第 5 章。 數位 *Certificate Manager for AS/400* ，說明 AS/400 DCM。

聯邦資訊存取安全標準 *(FIPS)*

本主題介紹 US National Institute of Standards and Technology 的 Federal Information Processing Standards (FIPS) Cryptomodule Validation Program ，以及可在 TLS 通道上使用的加密函數。

註: 在 AIX, Linux, and Windows 上， IBM MQ 透過 IBM Crypto for C (ICC) 加密模組提供 FIPS 140-2 相符性。 此模組的憑證已移至「歷程」狀態。 客戶應該檢視 IBM Crypto for C (ICC) 憑證 ，並注意 NIST 提供的任何建議。 目前正在進行取代 FIPS 140-3 模組，您可以在 處理程序清單中的 NIST CMVP 模組中搜尋它，以檢視其狀態。

此資訊適用於下列平台:

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

ALW 如需 AIX, Linux, and Windows 上 IBM MQ TLS 連線的 FIPS 140-2 相符性的相關資訊，請參閱 第 28 頁的『AIX, Linux, and Windows 的聯邦資訊存取安全標準 (FIPS)』。

z/OS 如需 z/OS 上 IBM MQ TLS 連線的 FIPS 140-2 相符性的相關資訊，請參閱 第 30 頁的『z/OS 的聯邦資訊存取安全標準 (FIPS)』。

如果存在加密硬體，則 IBM MQ 所使用的加密模組可以配置為硬體製造商所提供的那些加密模組。 如果這樣做，則只有在那些加密模組經過 FIPS 認證時，配置才符合 FIPS 標準。

一段時間後，「聯邦資訊存取安全標準」會更新，以反映針對加密演算法及通訊協定的新攻擊。 例如，部分 CipherSpecs 可能停止使用 FIPS 認證。 當發生這類變更時，也會更新 IBM MQ 來實作最新標準。 因此， 您可能會在套用維護項目之後看到行為的變更。

相關概念

第 227 頁的『指定在執行時期於 MQI 用戶端上僅使用 FIPS 認證的 CipherSpecs』
使用符合 FIPS 標準的軟體來建立金鑰儲存庫，然後指定通道必須使用 FIPS 認證的 CipherSpecs。

第 240 頁的『使用 runmqckm、 runmqakm 和 strmqikm 來管理數位憑證』
在 AIX, Linux, and Windows 系統上，使用 **strmqikm** (iKeyman) 來管理金鑰和數位憑證 GUI ，或從指令行使用 **runmqckm** (iKeycmd) 或 **runmqakm** (GSKCapiCmd)。

相關工作

在 IBM MQ classes for Java 中啟用 TLS

將傳輸層安全 (TLS) 與 IBM MQ classes for JMS 搭配使用

相關參考

JMS 物件的 TLS 內容

第 17 頁的『聯邦資訊處理標準』
美國政府針對 IT 系統和安全 (包括資料加密) 提供技術建議。 國家標準與技術機構 (NIST) 是涉及 IT 系統和安全的重要機構。 NIST 會產生建議和標準，包括「聯邦資訊存取安全標準 (FIPS)」。

ALW *AIX, Linux, and Windows* 的聯邦資訊存取安全標準 *(FIPS)*

當 AIX, Linux, and Windows 系統上的 SSL/TLS 通道需要加密法時， IBM MQ 會使用稱為 IBM Crypto for C (ICC)的加密法套件。 在 AIX, Linux, and Windows 平台上， ICC 軟體已通過美國國家標準與技術機構 (US National Institute of Standards and Technology) 的 Federal Information Processing Standards (FIPS) Cryptomodule Validation Program ， 層次 140-2。

註: 在 AIX, Linux, and Windows 上， IBM MQ 透過 IBM Crypto for C (ICC) 加密模組提供 FIPS 140-2 相符性。 此模組的憑證已移至「歷程」狀態。 客戶應該檢視 IBM Crypto for C (ICC) 憑證 ， 並注意 NIST 提供的任何建議。 目前正在進行取代 FIPS 140-3 模組，您可以在 處理程序清單中的 NIST CMVP 模組中搜尋它，以檢視其狀態。

AIX, Linux, and Windows 系統上 IBM MQ TLS 連線的 FIPS 140-2 相符性如下:

- 對於所有 IBM MQ 訊息通道 (CLNTCONN 通道類型除外) ， 如果符合下列條件，則連線符合 FIPS 標準:
  - 已安裝的 IBM Global Security Kit (GSKit) ICC 版本已在已安裝的作業系統版本及硬體架構上認證符合 FIPS 140-2 標準。
  - 佇列管理程式的 SSLFIPS 屬性已設為 YES。
  - 已使用僅符合 FIPS 標準的軟體 (例如 **runmqakm** 搭配 `-fips` 選項) 來建立及操作所有金鑰儲存庫。
- 對於所有 IBM MQ MQI client 應用程式，如果符合下列條件，則連線會使用 GSKit 且符合 FIPS 標準:
  - 已安裝的 GSKit ICC 版本已在已安裝的作業系統版本及硬體架構上認證符合 FIPS 140-2 標準。
  - 您已指定只使用 FIPS 認證的加密法，如 MQI 用戶端的相關主題中所述。
  - 已使用僅符合 FIPS 標準的軟體 (例如 **runmqakm** 搭配 `-fips` 選項) 來建立及操作所有金鑰儲存庫。
- 對於使用用戶端模式的 IBM MQ classes for Java 應用程式，如果符合下列條件，連線會使用 JRE 的 TLS 實作，且符合 FIPS 標準:
  - 在已安裝的作業系統版本及硬體架構上，用來執行應用程式的「 Java 執行時期環境」符合 FIPS 標準。
  - 您已指定僅使用 FIPS 認證的加密法，如 Java 用戶端的相關主題中所述。

  - 已使用僅符合 FIPS 標準的軟體 (例如 **runmqakm** 搭配 `-fips` 選項) 來建立及操作所有金鑰儲存庫。
- 對於使用用戶端模式的 IBM MQ classes for JMS 應用程式，如果符合下列條件，連線會使用 JRE 的 TLS 實作，且符合 FIPS 標準:
  - 在已安裝的作業系統版本及硬體架構上，用來執行應用程式的「Java 執行時期環境」符合 FIPS 標準。
  - 您已指定僅使用 FIPS 認證的加密法，如 JMS 用戶端的相關主題中所述。
  - 已使用僅符合 FIPS 標準的軟體 (例如 **runmqakm** 搭配 `-fips` 選項) 來建立及操作所有金鑰儲存庫。
- 對於未受管理的 .NET 用戶端應用程式，如果符合下列條件，則連線會使用 GSKit 且符合 FIPS 標準:
  - 已安裝的 GSKit ICC 版本已在已安裝的作業系統版本及硬體架構上認證符合 FIPS 140-2 標準。
  - 您已指定僅使用 FIPS 認證的加密法，如 .NET 用戶端的相關主題中所述。
  - 已使用僅符合 FIPS 標準的軟體 (例如 **runmqakm** 搭配 `-fips` 選項) 來建立及操作所有金鑰儲存庫。
- 對於未受管理的 XMS .NET 用戶端應用程式，如果符合下列條件，連線會使用 GSKit 且符合 FIPS 標準:
  - 已安裝的 GSKit ICC 版本已在已安裝的作業系統版本及硬體架構上認證符合 FIPS 140-2 標準。
  - 您已指定僅使用 FIPS 認證的加密法，如 XMS .NET 文件中所述。
  - 已使用僅符合 FIPS 標準的軟體 (例如 **runmqakm** 搭配 `-fips` 選項) 來建立及操作所有金鑰儲存庫。

所有支援的平台都經過 FIPS 140-2 認證，但每一個修正套件或產品更新套件隨附的 Readme 檔中所註明的除外。

對於使用 GSKit 的 TLS 連線，經 FIPS 140-2 認證的元件稱為 *ICC*。 此元件的版本可判定任何給定平台上的 GSKit FIPS 相符性。 若要判定目前已安裝的 ICC 版本，請執行 **dspmqver -p 64 -v** 指令。

以下是與 ICC 相關之 **dspmqver -p 64 -v** 輸出的範例擷取:

```
icc
============
@( #)CompanyName: IBM Corporation
@(#)LegalTrademarks: IBM
@( #)FileDescription: IBM Crypto for C-language
@(#)FileVersion:   8.0.0.0
@( #)LegalCopyright: 授權材料 - 屬於 IBM
@ (#) ICC
@(#) (C) 版權所有 IBM Corp. 2002, 2025。
@ (#) All Rights Reserved. US Government 使用者
@ (#) Restricted Rights-Use , duplication or disclosure
@ (#) restricted by GSA ADP Schedule Contract with IBM Corp.
@( #)ProductName: icc_8.0 ( GoldCoast Build) 100415
@(#)ProductVersion:  8.0.0.0
@(#)ProductInfo:   10/04/15.03:32:19.10/04/15.18:41:51
@ (#) CMVCInfo:
```

GSKit ICC 8 (包括在 GSKit 8 中) 的 NIST 憑證陳述式位於下列位址: Cryptographic Module Validation Program。

如果存在加密硬體，則 IBM MQ 所使用的加密模組可以配置為硬體製造商所提供的那些加密模組。 如果這樣做，則只有在那些加密模組經過 FIPS 認證時，配置才符合 FIPS 標準。

符合 FIPS 140-2 標準運作時施行三重 DES 演算法限制

當 IBM MQ 配置為符合 FIPS 140-2 標準運作時，會施行與三重 DES 演算法 (3DES) CipherSpecs 相關的其他限制。 這些限制可讓您符合美國 NIST SP800-67 建議。

1. 三重 DES 演算法金鑰的所有部分都必須是唯一的。
2. 根據 NIST SP800-67 中的定義，三重 DES 金鑰的任何部分都不能是「弱」、「半弱」或「可能弱」索引鍵。
3. 在必須重設秘密金鑰之前，無法透過連線傳輸超過 32 GB 的資料。 依預設， IBM MQ 不會重設秘密階段作業金鑰，因此必須配置此重設。 當使用三重 DES 演算法 CipherSpec 及 FIPS 140-2 相符性時，如果無法啟用秘密金鑰重設，則會導致在超出位元組計數上限之後關閉連線，並發生錯誤 AMQ9288 。 如需如何配置秘密金鑰重設的相關資訊，請參閱 第 389 頁的『重設 SSL 和 TLS 秘密金鑰』。

IBM MQ 會產生已符合規則 1 和 2 的三重 DES 演算法階段作業金鑰。 不過，若要滿足第三個限制，您必須在 FIPS 140-2 配置中使用三重 DES 演算法 CipherSpecs 時啟用秘密金鑰重設。 或者，您可以避免使用三重 DES 演算法。

相關概念

第 227 頁的『指定在執行時期於 MQI 用戶端上僅使用 FIPS 認證的 CipherSpecs』
使用符合 FIPS 標準的軟體來建立金鑰儲存庫，然後指定通道必須使用 FIPS 認證的 CipherSpecs。

第 240 頁的『使用 runmqckm、 runmqakm 和 strmqikm 來管理數位憑證』
在 AIX, Linux, and Windows 系統上，使用 **strmqikm** (iKeyman) 來管理金鑰和數位憑證 GUI，或從指令行使用 **runmqckm** (iKeycmd) 或 **runmqakm** (GSKCapiCmd)。

相關工作

在 IBM MQ classes for Java 中啟用 TLS

將傳輸層安全 (TLS) 與 IBM MQ classes for JMS 搭配使用

相關參考

JMS 物件的 TLS 內容

第 17 頁的『聯邦資訊處理標準』
美國政府針對 IT 系統和安全 (包括資料加密) 提供技術建議。 國家標準與技術機構 (NIST) 是涉及 IT 系統和安全的重要機構。 NIST 會產生建議和標準，包括「聯邦資訊存取安全標準 (FIPS)」。

z/OS *z/OS 的聯邦資訊存取安全標準 (FIPS)*

當 z/OS 上的 SSL/TLS 通道需要加密法時， IBM MQ 會使用稱為「系統 SSL」的服務。 System SSL 的目標是提供在設計為遵循美國國家標準與技術機構 (US National Institute of Standards and Technology，層次 140-2) 的「聯邦資訊存取安全標準 (FIPS) Cryptomodule 驗證程式」的模式下安全執行的功能。

使用 IBM MQ TLS 連線實作符合 FIPS 140-2 標準的連線時，需要考量一些點:

- 若要啟用 IBM MQ 訊息通道以符合 FIPS 標準，請確保符合下列條件:
  - 已安裝並配置系統 SSL 安全層次 3 FMID (請參閱 規劃安裝 IBM MQ)。
  - 已驗證系統 SSL 模組。
  - 佇列管理程式的 SSLFIPS 屬性已設為 **YES**。

在 FIPS 模式中執行時，如果可用，系統 SSL 會利用 CP Assist for Cryptographic Function (CPACF)。 以非 FIPS 模式執行時由 ICSF 支援的硬體所執行的加密功能，在以 FIPS 模式執行時仍會繼續遭到不當運用，但必須在軟體中執行的 RSA 簽章產生作業除外。

表 *2: FIPS* 模式與非 *FIPS* 模式演算法支援之間的差異。

	非 FIPS		**FIPS**	
演算法	**金鑰大小**	**硬體**	**金鑰大小**	**硬體**
RC2	40 和 128			
RC4	40 和 128			
DES	56	x		
TDES	168	x	168	x
AES	128 和 256	x	128 和 256	x
MD5	48			
SHA-1	160	x	160	x
SHA-2	224、256、384 及 512	x	224、256、384 及 512	x
RSA	512-4096	x	1024-4096	x
DSA	512-1024		1024	

表 2: *FIPS* 模式與非 *FIPS* 模式演算法支援之間的差異。 *(繼續)*				
	非 FIPS		**FIPS**	
演算法	**金鑰大小**	**硬體**	**金鑰大小**	**硬體**
DH	512-2048		2048	

在 FIPS 模式中，系統 SSL 只能使用使用表格 1 所示演算法及金鑰大小的憑證。 在 X.509 憑證驗證期間，如果發現與 FIPS 模式不相容的演算法，則無法使用憑證，且會被視為無效。

對於在 WebSphere Application Server 內使用用戶端模式的 IBM MQ 類別應用程式，請參閱 美國聯邦資訊處理標準支援。

如需系統 SSL 模組配置的相關資訊，請參閱 系統 SSL 模組驗證設定 。

相關參考

第 17 頁的『聯邦資訊處理標準』

美國政府針對 IT 系統和安全 (包括資料加密) 提供技術建議。 國家標準與技術機構 (NIST) 是涉及 IT 系統和安全的重要機構。 NIST 會產生建議和標準，包括「聯邦資訊存取安全標準 (FIPS)」。

Multi 使用 *mqcertck* 驗證佇列管理程式的 *TLS* 配置

MQCERTCK 指令是用來尋找佇列管理程式的 TLS 配置中常見錯誤的工具，並提供一些解決問題的建議。

簡介

mqcertck 指令會檢查:

- 佇列管理程式 **SSLKEYR** 屬性中所參照佇列管理程式之金鑰儲存庫的存在及許可權。
- 佇列管理程式 **CERTLABL** 屬性中所參照之佇列管理程式憑證的憑證存在及有效性。
- 已啟用 TLS 之通道的 **CERTLABL** 屬性中所參照之任何憑證的存在及有效性。
- 金鑰儲存庫及用戶端應用程式的憑證，包括檢查憑證是否已由佇列管理程式授權。

註: mqcertck 指令在 z/OS 或 IBM i 上無法使用。

用法

若要使用 **mqcertck** 指令，請從指令行執行指令 `mqcertck` 及其必要參數，以及您需要的任何選用參數。

如需指令及指令所採用參數的說明，請參閱 mqcertck 。

範例

您剛剛完成佇列管理程式 QM1 的設定，以容許從用戶端連接至佇列管理程式的 SVRCONN 通道的 TLS 連線。

您正在使用多個憑證特性，因此您的佇列管理程式及通道都在其 **CERTLABL** 屬性中指定了憑證標籤。 建立通道時，您在通道的 **CERTLABL** 屬性中發生錯誤，因此當用戶端嘗試連接時，佇列管理程式會傳回 2393 回覆碼 MQRC_SSL_INITIALIZATION_ERROR。

在啟動佇列管理程式之前，您可以使用 **mqcertck** 指令來驗證佇列管理程式的 TLS 配置。

您執行指令 `mqcertck QM1` 並接收下列輸出:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2025.
+------------------------------------------------------------
| IBM MQ TLS Configuration Test tool
+------------------------------------------------------------
| Problem identified:
|  No certificate could be found for the channel
|  MQCERTCK.CHANNEL
|  This tool looked in the Queue Manager's key repository
|  located at: 'C:\MQ Data\qmgrs\QM1\ssl\key.kdb'
|  for a certificate with label 'chacert',
|  which is the certificate specified in the channel's
```

```
|  CERTLABL attribute, but was unable to find one.
|
| Possible resolution:
|  A valid certificate with the label chacert
|  needs to be added to the key repository.
|
|  Alternatively, alter the channel definition to remove
|  the CERTLABL value. This can be done by executing the
|  following command in runmqsc:
|      ALTER CHANNEL(<Name>) CHLTYPE(<TYPE>) CERTLABL(' ')
+-----------------------------------------------------------
| mqcertck has ended. See above for any problems found.
| If there are problems then resolve these and run this
| tool again.
|
+-----------------------------------------------------------
```

此輸出會提示您檢查伺服器連線通道 MQCERTCK.CHANNEL。 在這裡，您會看到您所做的錯誤，並且可以在重新執行 mqcertck 指令之前更正錯誤，以驗證您已解決問題。

驗證用戶端連線

mqcertck 指令能夠驗證用戶端金鑰儲存庫，以及佇列管理程式的 TLS 配置。 若要這樣做， **mqcertck** 必須能夠從執行佇列管理程式的機器存取用戶端的金鑰儲存庫。

執行 **mqcertck** 指令時，如果您提供 **-clientkeyr** 參數與用戶端金鑰儲存庫的位置 (不包括延伸規格) **mqcertck** ，則會根據佇列管理程式來檢查此金鑰儲存庫。

如果您知道用戶端將使用哪個通道來連接佇列管理程式，則可以使用 **-clientchannel** 旗標來指定此通道。

如果用戶端使用交互鑑別來連接佇列管理程式，您可以使用 **-clientusername** 或 **-clientlabel** 參數，以告知 **mqcertck** 指令要在用戶端金鑰儲存庫中使用哪一個憑證。

如果您使用預設憑證，且未提供憑證標籤給用戶端應用程式，則可以使用 **-clientusername** 及執行此應用程式的 **username** 參數。

在 **mqcertck** 指令的作業期間，該指令會產生憑證標籤 ibmwebspheremqXXXX ，其中 XXXX 是在 **-clientusername** 參數中傳遞的值。

為了完整驗證用戶端金鑰儲存庫， **mqcertck** 指令會使用 IBM Global Security Kit (GSKit)來建立虛擬連線。 若要這樣做，指令需要有可在其用戶端測試期間連結的埠。 使用的預設埠是 5857，不過，如果已在使用中，您可以指定在用戶端測試期間使用不同的埠。

註: 雖然 **mqcertck** 指令連結至埠，但 **mqcertck** 不會使用任何外部通訊，且會在本端執行所有測試。

IBM MQ MQI client 上的 *SSL/TLS*

IBM MQ 在用戶端上支援 TLS。 您可以透過各種方式自訂 TLS 的使用。

IBM MQ 為 AIX, Linux, and Windows 系統上的 IBM MQ MQI clients 提供 TLS 支援。 如果您使用 IBM MQ classes for Java，請參閱 使用 IBM MQ classes for Java ，如果您使用 IBM MQ classes for JMS，請參閱 使用 IBM MQ classes for JMS。 本節的其餘部分不適用於 Java 或 JMS 環境。

您可以在 IBM MQ 用戶端配置檔中使用 MQSSLKEYR 值，或在應用程式發出 MQCONNX 呼叫時，指定 IBM MQ MQI client 的金鑰儲存庫。 您有三個選項可指定通道使用 TLS:

- 使用通道定義表
- 在 MQCONNX 呼叫中使用 SSL 配置選項結構 MQSCO
- 使用 Active Directory (在 Windows 系統上)

您無法使用 MQSERVER 環境變數來指定通道使用 TLS。

只要通道的另一端未指定 TLS ，您可以繼續在沒有 TLS 的情況下執行現有 IBM MQ MQI client 應用程式。

如果在用戶端機器上對「TLS 金鑰儲存庫」的內容、「TLS 金鑰儲存庫」的位置、「鑑別資訊」或「加密硬體」參數進行變更，則您需要結束所有 TLS 連線，以便在應用程式用來連接至佇列管理程式的用戶端連線通道中反映這些變更。 一旦所有連線都已結束，請重新啟動 TLS 通道。 會使用所有新的 TLS 設定。 這些設定類似於佇列管理程式系統上由 REFRESH SECURITY TYPE (SSL) 指令重新整理的設定。

當 IBM MQ MQI client 在具有加密硬體的 AIX, Linux, and Windows 系統上執行時，您可以使用 MQSSLCRYP 環境變數來配置該硬體。 此變數相當於 ALTER QMGR MQSC 指令上的 SSLCRYP 參數。 如需 ALTER QMGR MQSC 指令上 SSLCRYP 參數的說明，請參閱 ALTER QMGR 。 如果您使用 SSLCRYP 參數的 GSK_PCS11 版本，則必須完全以小寫形式指定 PKCS #11 記號標籤。

IBM MQ MQI clients 支援 TLS 秘密金鑰重設及 FIPS。 如需相關資訊，請參閱第 389 頁的『重設 SSL 和 TLS 秘密金鑰』和第 28 頁的『AIX, Linux, and Windows 的聯邦資訊存取安全標準 (FIPS)』。

如需 IBM MQ MQI clients 的 TLS 支援的相關資訊，請參閱 第 226 頁的『設定 IBM MQ MQI client 安全』 。

相關工作

使用配置檔來配置用戶端

指定 *MQI* 通道使用 *SSL/TLS*

若要讓 MQI 通道使用 TLS，用戶端連線通道的 *SSLCipherSpec* 屬性值必須是用戶端平台上 IBM MQ 支援的 CipherSpec 名稱。

您可以透過下列方式，使用此屬性的值來定義用戶端連線通道。 它們按優先順序遞減順序列出。

1. 當 PreConnect 結束程式提供要使用的通道定義結構時。

   PreConnect 結束程式可以在通道定義結構 MQCD 的 *SSLCipherSpec* 欄位中提供 CipherSpec 的名稱。 此結構在 PreConnect 結束程式所使用 MQNXP 結束程式參數結構的 **ppMQCDArrayPtr** 欄位中傳回。
2. 當 IBM MQ MQI client 應用程式發出 MQCONNX 呼叫時。

   應用程式可以在通道定義結構 MQCD 的 *SSLCipherSpec* 欄位中指定 CipherSpec 的名稱。此結構由連接選項結構 MQCNO 參照，該結構是 MQCONNX 呼叫中的參數。
3. 使用用戶端通道定義表 (CCDT)。

   用戶端通道定義表中的一個以上項目可以指定 CipherSpec 的名稱。例如，如果您使用 DEFINE CHANNEL MQSC 指令建立項目，則可以在指令上使用 SSLCIPH 參數來指定 CipherSpec 的名稱。
4. 在 Windows 上使用 Active Directory 。

   在 Windows 系統上，您可以使用 **setmqscp** 控制指令，在 Active Directory 中發佈用戶端連線通道定義。 其中一個以上定義可以指定 CipherSpec 的名稱。

例如，如果用戶端應用程式在 MQCONNX 呼叫的 MQCD 結構中提供用戶端連線通道定義，則此定義優先於用戶端通道定義表中可由 IBM MQ 用戶端存取的任何項目。

您無法使用 MQSERVER 環境變數，在使用 TLS 之 MQI 通道的用戶端提供通道定義。

若要檢查用戶端憑證是否已傳送，請在通道的伺服器端顯示通道狀態，以顯示對等節點名稱參數值。

相關概念

第 368 頁的『指定 IBM MQ MQI client 的 CipherSpec』

您有三個選項可指定 IBM MQ MQI client 的 CipherSpec 。

IBM MQ 中的 *CipherSpecs* 和 *CipherSuites*

IBM MQ 支援 TLS1.3 和 TLS 1.2 CipherSpecs，以及 RSA 和 Diffie-Hellman 演算法。 不過，如果您需要啟用已淘汰的 CipherSpecs，則可以這麼做。

如需下列相關資訊，請參閱 第 348 頁的『啟用 CipherSpecs』 :

- IBM MQ 支援 CipherSpecs 。
- 如何啟用已淘汰的 SSL 3.0 和 TLS 1.0 CipherSpecs。

IBM MQ 支援 RSA 和 Diffie-Hellman 金鑰交換及鑑別演算法。 TLS 信號交換期間使用的金鑰大小可以視您使用的數位憑證而定，但部分 CipherSpecs 包括信號交換金鑰大小的規格。 信號交換金鑰越大，所能提供的鑑別功能越強。 但金鑰越小，信號交換速度越快。

相關概念

第 16 頁的『CipherSpecs 和 CipherSuites』

加密安全通訊協定必須同意安全連線所使用的演算法。 CipherSpecs 和 CipherSuites 定義演算法的特定組合。

IBM MQ 中的 *NSA Suite B* 加密法

本主題提供如何配置 IBM MQ for AIX, Linux, and Windows 以符合套組 B 相容 TLS 1.2 設定檔的相關資訊。

隨著時間的推移， NSA Cryptography Suite B Standard 會更新，以反映針對加密演算法和通訊協定的新攻擊。 例如，部分 CipherSpecs 可能不再經過 Suite B 認證。 當發生這類變更時，也會更新 IBM MQ 來實作最新標準。 因此， 您可能會在套用維護項目之後看到行為的變更。 IBM MQ Readme 檔列出每一個產品維護層次所施行的套組 B 版本。 如果您配置 IBM MQ 以施行套組 B 相符性，在規劃套用維護時，請一律參閱 Readme 檔。 請參閱 IBM MQ、 WebSphere MQ 和 MQSeries 產品自述文件。

在 AIX, Linux, and Windows 系統上， IBM MQ 可以配置為符合表 1 所示安全層次的 Suite B 相容 TLS 1.2 設定檔。

表 *3:* 具有容許的 *CipherSpecs* 及數位簽章演算法的 *Suite B* 安全層次

安全層次	容許 CipherSpecs	容許的數位簽章演算法
128 位元	ECDHE_ECDSA_AES_128_GCM_SHA256 ECDHE_ECDSA_AES_256_GCM_SHA384	ECDSA，含 SHA-256 ECDSA，含 SHA-384
192 位元	ECDHE_ECDSA_AES_256_GCM_SHA384	ECDSA，含 SHA-384
兩者 [1]	ECDHE_ECDSA_AES_128_GCM_SHA256 ECDHE_ECDSA_AES_256_GCM_SHA384	ECDSA，含 SHA-256 ECDSA，含 SHA-384

1. 可以同時配置 128 位元及 192 位元安全層次。 由於套組 B 配置會決定可接受的最低加密演算法，因此配置這兩個安全層次相當於只配置 128 位元安全層次。 192 位元安全層次的加密演算法比 128 位元安全層次所需的最小值更強，因此即使未啟用 192 位元安全層次，也允許 128 位元安全層次使用它們。

註: 用於 安全層次 的命名慣例不一定代表 AES 加密演算法的橢圓曲線大小或金鑰大小。

CipherSpec 與套組 B 的構象

雖然 IBM MQ 的預設行為不符合套組 B 標準，但 IBM MQ 可以配置為符合 AIX, Linux, and Windows 系統上的其中一個或兩個安全層次。 在成功配置 IBM MQ 以使用套組 B 之後，嘗試使用 CipherSpec 不符合套組 B 的 CipherSpec 來啟動出埠通道會導致錯誤 AMQ9282。 此活動也會導致 MQI 用戶端傳回原因碼 MQRC_CIPHER_SPEC_NOT_SUITE_B。 同樣地，嘗試使用不符合套組 B 配置的 CipherSpec 來啟動入埠通道會導致錯誤 AMQ9616。

如需 IBM MQ CipherSpecs 的相關資訊，請參閱 第 348 頁的『啟用 CipherSpecs』

套組 B 和數位憑證

套組 B 限制可用來簽署數位憑證的數位簽章演算法。 套組 B 也會限制憑證可包含的公開金鑰類型。 因此，IBM MQ 必須配置為使用遠端夥伴的已配置套組 B 安全層次容許其數位簽章演算法及公開金鑰類型的憑證。 不符合安全層次需求的數位憑證會遭到拒絕，且連線失敗，錯誤為 AMQ9633 或 AMQ9285。

對於 128 位元 Suite B 安全層次，憑證主體的公開金鑰必須使用 NIST P-256 橢圓曲線或 NIST P-384 橢圓曲線，並使用 NIST P-256 橢圓曲線或 NIST P-384 橢圓曲線來簽署。 在 192 位元 Suite B 安全層次，需要憑證主體的公開金鑰才能使用 NIST P-384 橢圓曲線，並以 NIST P-384 橢圓曲線簽署。

若要取得適用於套組 B 相容作業的憑證，請使用 **runmqakm** 指令並指定 **-sig_alg** 參數，以要求適當的數位簽章演算法。 EC_ecdsa_with_SHA256 和 EC_ecdsa_with_SHA384 **-sig_alg** 參數值對應於由容許的套組 B 數位簽章演算法簽署的橢圓曲線金鑰。

如需 **runmqakm** 指令的相關資訊，請參閱 runmqckm 及 runmqakm 選項。

註: runmqckm 和 **strmqikm** 指令不支援為符合 Suite B 標準的作業建立數位憑證。

建立及要求數位憑證

若要建立用於套組 B 測試的自簽數位憑證，請參閱 第 246 頁的『在 AIX, Linux, and Windows 上建立自簽個人憑證』

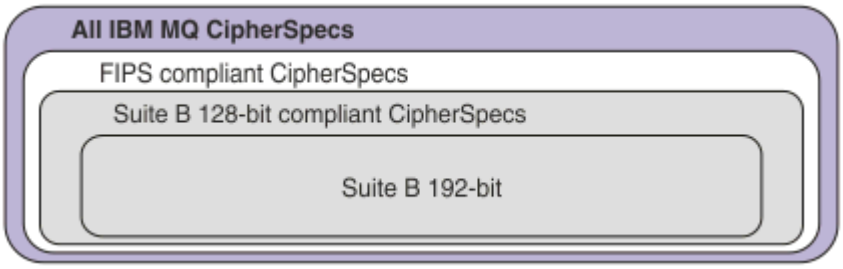
若要申請 CA 簽章的數位憑證以供 Suite B 正式作業使用，請參閱 第 248 頁的『在 AIX, Linux, and Windows 上要求個人憑證』。

註: 所使用的憑證管理中心必須產生數位憑證，以滿足 IETF RFC 6460 中說明的需求。

FIPS 140-2 和套組 B

註: 在 AIX, Linux, and Windows 上， IBM MQ 透過 IBM Crypto for C (ICC) 加密模組提供 FIPS 140-2 相符性。 此模組的憑證已移至「歷程」狀態。 客戶應該檢視 IBM Crypto for C (ICC) 憑證 ，並注意 NIST 提供的任何建議。 目前正在進行取代 FIPS 140-3 模組，您可以在 處理程序清單中的 NIST CMVP 模組中搜尋它，以檢視其狀態。

「套組 B」標準在概念上類似於 FIPS 140-2 ，因為它會限制已啟用的加密演算法集，以提供安全的保證層次。 當 IBM MQ 配置為符合 FIPS 140-2 標準的作業時，可以使用目前支援的 Suite B CipherSpecs 。 因此，可以同時配置 IBM MQ 以符合 FIPS 及套組 B 標準，在此情況下，這兩組限制都適用。



下圖說明這些子集之間的關係:

針對套組 B 相容作業配置 IBM MQ

如需如何在 AIX, Linux, and Windows 上配置 IBM MQ 以符合套組 B 標準作業的相關資訊，請參閱 第 35 頁的『為套組 B 配置 IBM MQ』。

IBM MQ 在 IBM i 和 z/OS 平台上不支援 Suite B 相容作業。 IBM MQ Java 和 JMS 用戶端也不支援 Suite B 相容作業。

相關概念

第 227 頁的『指定在執行時期於 MQI 用戶端上僅使用 FIPS 認證的 CipherSpecs』
使用符合 FIPS 標準的軟體來建立金鑰儲存庫，然後指定通道必須使用 FIPS 認證的 CipherSpecs。

ALW 為套組 *B* 配置 *IBM MQ*

IBM MQ 可以配置為符合 AIX, Linux, and Windows 平台上的 NSA Suite B 標準。

套組 B 會限制已啟用的加密演算法集，以提供安全的保證層次。 IBM MQ 可以配置為遵循 Suite B 來運作，以提供加強的安全層次。 如需套組 B 的進一步資訊，請參閱 第 17 頁的『國家安全域性 (NSA) Suite B 加密法』。 如需套組 B 配置及其對 TLS 通道的影響的相關資訊，請參閱 第 34 頁的『IBM MQ 中的 NSA Suite B 加密法』。

佇列管理程式

若為佇列管理程式，請搭配使用指令 **ALTER QMGR** 與參數 **SUITEB** ，以設定適合您所需安全層次的值。 如需進一步資訊，請參閱 ALTER QMGR。

您也可以搭配使用 PCF **MQCMD_CHANGE_Q_MGR** 指令與 **MQIA_SUITE_B_STRENGTH** 參數，以針對符合 Suite B 的作業配置佇列管理程式。

註: 如果您變更佇列管理程式的套組 B 設定，則必須重新啟動 MQXR 服務，這些設定才會生效。

MQI 用戶端

依預設， MQI 用戶端不會強制執行套組 B 相符性。 您可以執行下列其中一個選項，以啟用 MQI 用戶端的「套組 B」相符性:

1. 透過將 MQCONNX 呼叫 MQSCO 結構中的 **EncryptionPolicySuiteB** 欄位設為下列一或多個值:
   - MQ_SUITE_B_NONE
   - MQ_SUITE_B_128_BIT
   - MQ_SUITE_B_192_BIT

   搭配使用 MQ_SUITE_B_NONE 與任何其他值無效。
2. 將 MQSUITEB 環境變數設為下列一或多個值:
   - 無
   - 128_BIT
   - 192_BIT

   您可以使用逗點區隔清單來指定多個值。 將值 NONE 與任何其他值搭配使用無效。
3. 透過將 MQI 用戶端配置檔的 SSL 段落中的 **EncryptionPolicySuiteB** 屬性設為下列一或多個值:
   - 無
   - 128_BIT
   - 192_BIT

   您可以使用逗點區隔清單來指定多個值。 搭配使用 NONE 與任何其他值無效。

註: MQI 用戶端設定會依優先順序列出。 MQCONNX 呼叫上的 MSCO 結構會置換 MQSUITEB 環境變數上的設定，其會置換 SSL 段落中的屬性。

如需 MQSCO 結構的完整資料，請參閱 MQSCO-SSL 配置選項。

如需在用戶端配置檔中使用 Suite B 的相關資訊，請參閱 用戶端配置檔的 SSL 段落。

如需使用 MQSUITEB 環境變數的進一步資訊，請參閱 環境變數說明。

.NET

對於 .NET 未受管理的用戶端，內容 **MQC.ENCRYPTION_POLICY_SUITE_B** 指出所需的套組 B 安全類型。

如需在 IBM MQ classes for .NET 中使用套組 B 的相關資訊，請參閱 MQEnvironment .NET 類別。

AMQP

佇列管理程式的套組 B 屬性設定會套用至該佇列管理程式上的 AMQP 通道。 如果您修改佇列管理程式套組 B 設定，則必須重新啟動 AMQP 服務，變更才會生效。

IBM MQ 中的憑證驗證原則

憑證驗證原則決定憑證鏈驗證符合業界安全標準的嚴格程度。

憑證驗證原則取決於平台及環境，如下所示:

- 對於所有平台上的 Java 和 JMS 應用程式，憑證驗證原則取決於 Java 執行時期環境的 JSSE 元件。 如需憑證驗證原則的相關資訊，請參閱 JRE 的說明文件。
- 對於 IBM i 系統，憑證驗證原則取決於作業系統提供的 Secure Socket Library。 如需憑證驗證原則的相關資訊，請參閱作業系統的說明文件。
- 對於 z/OS 系統，憑證驗證原則取決於作業系統所提供的「系統 SSL」元件。 如需憑證驗證原則的相關資訊，請參閱作業系統的說明文件。
- 對於 AIX, Linux, and Windows 系統，憑證驗證原則由 IBM Global Security Kit (GSKit) 提供且可以配置。 支援兩個不同的憑證驗證原則:
  - 舊式憑證驗證原則，用於與不符合現行 IETF 憑證驗證標準的舊數位憑證保持最大舊版相容性及交互作業能力。 此原則稱為「基本」原則。
  - 嚴格符合標準的憑證驗證原則，施行 RFC 5280 標準。 此原則稱為「標準」原則。

如需如何在 AIX, Linux, and Windows 上配置憑證驗證原則的相關資訊，請參閱 第 37 頁的『在 IBM MQ 中配置憑證驗證原則』。 如需「基本」與「標準」憑證驗證原則之間差異的相關資訊，請參閱 AIX, Linux, and Windows 上的憑證驗證及信任原則設計。

在 *IBM MQ* 中配置憑證驗證原則

您可以用四種方式指定使用哪個 TLS 憑證驗證原則來驗證從遠端夥伴系統收到的數位憑證。

在佇列管理程式上，可以使用下列方式來設定憑證驗證原則:

- 使用佇列管理程式屬性 *CERTVPOL*。 如需設定此屬性的相關資訊，請參閱 ALTER QMGR。

在用戶端上，有數種方法可用來設定憑證驗證原則。 如果使用多個方法來設定原則，用戶端會以下列優先順序來使用設定:

1. 使用用戶端 MQSCO 結構中的 *CertificateVal* 原則 欄位。 如需使用此欄位的相關資訊，請參閱 MQSCO-SSL 配置選項。
2. 使用用戶端環境變數 *MQCERTVPOL*。 如需使用此變數的相關資訊，請參閱 MQCERTVPOL。
3. 使用用戶端 SSL 段落調整參數設定 *CertificateValPolicy*。 如需使用此設定的相關資訊，請參閱 用戶端配置檔的 SSL 段落。

如需憑證驗證原則的相關資訊，請參閱 第 36 頁的『IBM MQ 中的憑證驗證原則』。

IBM MQ 中的數位憑證及 *CipherSpec* 相容性

本主題提供如何透過概述 CipherSpecs 與 IBM MQ 中數位憑證之間的關係，為安全原則選擇適當的 CipherSpecs 及數位憑證的相關資訊。

只有一部分受支援的 CipherSpecs 可以與所有受支援的數位憑證類型搭配使用。 因此，必須為您的數位憑證選擇適當的 CipherSpec 。 同樣地，如果您組織的安全原則要求您使用特定的 CipherSpec ，則必須為該 CipherSpec 取得適當的數位憑證。

MD5 數位簽章演算法和 TLS 1.2

使用 TLS 1.2 通訊協定時，會拒絕使用 MD5 演算法簽署的數位憑證。 這是因為現在許多加密分析師都認為 MD5 演算法很弱，因此通常不建議使用它。 若要使用基於 TLS 1.2 通訊協定的較新的 CipherSpecs ，請確保數位憑證在其數位簽章中不會使用 MD5 演算法。 使用 TLS 1.0 通訊協定的較舊 CipherSpecs 不受此限制，可以繼續使用具有 MD5 數位簽章的憑證。

若要檢視特定憑證的數位簽章演算法，您可以使用 **runmqakm** 指令:

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

其中 *cert_label* 是要顯示之數位簽章演算法的憑證標籤。 如需詳細資料，請參閱 數位憑證標籤 。

註: 雖然可以使用 **runmqckm** (iKeycmd) 及 **strmqikm** (iKeyman) GUI 來檢視數位簽章演算法的選項，但 **runmqakm** 工具會提供更廣泛的範圍。

執行 **runmqakm** 指令會產生輸出，顯示使用指定的簽章演算法:

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

```
Public Key Type : RSA (1.2.840.113549.1.1.1)
Fingerprint : SHA1 :
    09 4E 4F F2 1B CB C1 F4 4F 15 C9 2A F7 32 0A 82
    DA 45 92 9F
Fingerprint : MD5 :
    44 54 81 7C 58 68 08 3A 5D 75 96 40 D5 8C 7A CB
Fingerprint : SHA256 :
    3B 47 C6 E7 7B B0 FF 85 34 E7 48 BE 11 F2 D4 35
    B7 9A 79 53 2B 07 F5 E7 65 E8 F7 84 E0 2E 82 55
Signature Algorithm : MD5WithRSASignature (1.2.840.113549.1.1.4)
Value
    3B B9 56 E6 F2 77 94 69 5B 3F 17 EA 7B 19 D0 A2
    D7 10 38 F1 88 A4 44 1B 92 35 6F 3B ED 99 9B 3A
    A5 A4 FC 72 25 5A A9 E3 B1 96 88 FC 1E 9F 9B F1
    C5 E8 8E CF C4 8F 48 7B 0E A6 BB 13 AE 2B BD D8
    63 2C 03 38 EF DC 01 E1 1F 7A 6F FB 2F 65 74 D0
    FD 99 94 BA B2 3A D5 B4 89 6C C1 2B 43 6D E2 39
    66 6A 65 CB C3 C4 E2 CC F5 49 39 A3 8B 93 5A DD
    B0 21 0B A8 B2 59 5B 24 59 50 44 89 DC 78 19 51
Trust Status : Enabled
```

`Signature Algorithm` 行顯示使用 `MD5WithRSASignature` 演算法。 此演算法基於 MD5 ， 因此此數位憑證無法與 TLS 1.2 CipherSpecs 搭配使用。

橢圓曲線和 RSA CipherSpecs 的交互作業能力

V 9.2.0 並非所有 CipherSpecs 都可以與所有數位憑證搭配使用。 CipherSpecs 以 CipherSpec 名稱字首表示。 每一種類型的 CipherSpec 都會對可以使用的數位憑證類型施加不同的限制。 這些限制適用於所有 IBM MQ TLS 連線，但特別適用於橢圓曲線加密法的使用者。

下表彙總 CipherSpecs 與數位憑證之間的關係:

表 *4: CipherSpecs* 與數位憑證之間的關係

類型	CipherSpec 名稱字首	說明	必要的公開金鑰類型	數位簽章加密演算法	秘密金鑰建立方法
1	ECDHE_ECDSA_	CipherSpecs ， 使用「橢圓曲線公開金鑰」、「橢圓曲線秘密金鑰」及「橢圓曲線數位簽章演算法」。	橢圓曲線	ECDSA	ECDHE
2	ECDHE_RSA_	使用 RSA 公開金鑰、橢圓曲線秘密金鑰及 RSA 數位簽章演算法的 CipherSpecs 。	RSA	RSA	ECDHE
V 9.2.0 3	(所有 TLS 1.3 CipherSpecs)	CipherSpecs ， 使用「橢圓曲線」或 RSA 公開金鑰、「橢圓曲線秘密金鑰」及「橢圓曲線」或 RSA 數位簽章演算法。	橢圓曲線或 RSA	ECDSA 或 RSA	ECDHE 或 RSA
4	(所有其他)	使用 RSA 公開金鑰和 RSA 數位簽章演算法的 CipherSpecs 。	RSA	RSA	RSA

必要的公開金鑰類型直欄會顯示使用每一種 CipherSpec 類型時， 個人憑證必須具有的公開金鑰類型。 個人憑證是向其遠端友機識別佇列管理程式或用戶端的終端實體憑證。

您必須確定憑證標籤中指定的憑證適用於通道 CipherSpec。 也就是說，如果您使用需要 Elliptic Curve (EC) 憑證的 CipherSpec 來配置通道，則無法在憑證標籤中命名 RSA 憑證。 如果您使用需要 RSA 憑證的 CipherSpec 來配置通道，則無法在憑證標籤中命名 EC 憑證。

假設您已正確配置 IBM MQ，則可以具有:

- 混合 RSA 和 EC 憑證的單一佇列管理程式。
- 相同佇列管理程式上使用 RSA 或 EC 憑證的不同通道。

數位簽章加密演算法是指用來驗證對等節點的加密演算法。 加密演算法與雜湊演算法 (例如 MD5、 SHA-1 或 SHA-256) 一起使用， 以計算數位簽章。 可以使用各種數位簽章演算法， 例如 RSA 與 MD5 或 ECDSA 與

SHA-256。 在表格中， ECDSA 是指使用 ECDSA 的數位簽章演算法集; RSA 是指使用 RSA 的數位簽章演算法集。 可以使用集合中任何受支援的數位簽章演算法，前提是它是基於指定的加密演算法。

類型 1 CipherSpecs 需要個人憑證必須具有「橢圓曲線」公開金鑰。 使用這些 CipherSpecs 時，會使用「橢圓曲線 Diffie Hellman 暫時金鑰協定」來建立連線的秘密金鑰。

類型 2 CipherSpecs 需要個人憑證具有 RSA 公開金鑰。 使用這些 CipherSpecs 時，會使用「橢圓曲線 Diffie Hellman 暫時金鑰協定」來建立連線的秘密金鑰。

類型 3 CipherSpecs 需要個人憑證必須具有 RSA 公開金鑰。 使用這些 CipherSpecs 時，會使用 RSA 金鑰交換來建立連線的秘密金鑰。

此限制清單並非詳盡無遺: 視配置而定，可能還有其他限制可進一步影響交互作業能力。 例如，如果 IBM MQ 配置為符合 FIPS 140-2 或 NSA Suite B 標準，則這也會限制容許配置的範圍。 如需相關資訊，請參閱下列小節。

如果您需要在相同佇列管理程式或用戶端應用程式上使用不同類型的 CipherSpec ，請在用戶端定義上配置適當的憑證標籤及 CipherSpec 組合。

三種類型的 CipherSpec 不會直接交互作業: 這是現行 TLS 標準的限制。 例如，假設您選擇將 ECDHE_ECDSA_AES_128_CBC_SHA256 CipherSpec 用於名為 QM1 的佇列管理程式上名為 TO.QM1 的接收端通道，則接收端應該具有具有橢圓曲線金鑰及 ECDSA 型數位簽章的個人憑證。 如果接收端通道不符合這些需求，則通道無法啟動。

連接到佇列管理器 QM1 其他通道可以使用其他 CipherSpecs,前提是每個通道使用該通道的 CipherSpec 的正確類型的憑證。 例如，假設 QM1 使用名為 TO.QM2 ，將訊息傳送至名為 QM2 的另一個佇列管理程式。 通道 TO.QM2 可以使用類型 4 CipherSpec TLS_RSA_WITH_AES_256_CBC_SHA256 ，前提是通道兩端都使用包含 RSA 公鑰的憑證。 憑證標籤通道屬性可用來為每一個通道配置不同的憑證。

規劃 IBM MQ 網路時，請仔細考量哪些通道需要 TLS ，並確保用於每個通道的憑證類型適合與該通道上的 CipherSpec 搭配使用。

若要檢視數位憑證的數位簽章演算法及公開金鑰類型，您可以使用 **runmqakm** 指令:

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

其中 *cert_label* 是您需要顯示其數位簽章演算法之憑證的標籤。 如需詳細資料，請參閱 數位憑證標籤 。

執行 **runmqakm** 指令會產生輸出，顯示「公開金鑰類型」:

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

```
    22 CC E6 1A 4E F4 61 CC 51 F9 EE A0 8E F4 DC B5
    0B B9 72 58 C3 C7 A4
Trust Status : Enabled
```

在此情況下，「公開金鑰類型」線會顯示憑證具有「橢圓曲線」公開金鑰。 在此情況下，「簽章演算法」行顯示正在使用 EC_ecdsa_with_SHA384 演算法: 這是根據 ECDSA 演算法。 因此，此憑證僅適用於類型 1 CipherSpecs。

您也可以搭配使用 **runmqckm** 指令與相同的參數。 此外，如果您開啟金鑰儲存庫並按兩下憑證的標籤，則可以使用 **strmqikm** GUI 來檢視數位簽章演算法。 不過，您應該使用 **runmqakm** 工具來檢視數位憑證，因為它支援範圍更廣的演算法。

TLS 1.3 CipherSpecs

V 9.2.0

TLS 1.3 CipherSpecs 同時支援 ECDSA 及 RSA 憑證。

橢圓曲線 CipherSpecs 與 NSA Suite B

當 IBM MQ 配置為符合 Suite B 相容 TLS 1.2 設定檔時，允許的 CipherSpecs 及數位簽章演算法會受到限制，如 第 34 頁的『IBM MQ 中的 NSA Suite B 加密法』中所述。 此外，可接受橢圓曲線金鑰的範圍會根據所配置的安全層次而減少。

在 128 位元 Suite B 安全層次，憑證主體的公開金鑰必須使用 NIST P-256 或 NIST P-384 橢圓曲線，並以 NIST P-256 橢圓曲線或 NIST P-384 橢圓曲線簽署。 **runmqakm** 指令可使用 EC_ecdsa_with_SHA256 或 EC_ecdsa_with_SHA384 的 -sig_alg 參數來要求此安全層次的數位憑證。

在 192 位元 Suite B 安全層次，需要憑證主體的公開金鑰才能使用 NIST P-384 橢圓曲線，並使用 NIST P-384 橢圓曲線簽署。 **runmqakm** 指令可使用 EC_ecdsa_with_SHA384 的 -sig_alg 參數來要求此安全層次的數位憑證。

支援的 NIST 橢圓曲線如下:

表 *5:* 支援的 *NIST* 橢圓曲線

NIST FIPS 186-3 曲線名稱	**RFC 4492 曲線名稱**	**橢圓曲線金鑰大小 (位元)**
P-256	secp256r1	256
P-384	secp384r1	384
P-521	secp521r1	521

註: NIST P-521 橢圓曲線無法用於「套組 B」相容作業。

相關概念

第 348 頁的『啟用 CipherSpecs』
在 **DEFINE CHANNEL** 或 **ALTER CHANNEL** MQSC 指令中使用 **SSLCIPH** 參數來啟用 CipherSpec 。

第 227 頁的『指定在執行時期於 MQI 用戶端上僅使用 FIPS 認證的 CipherSpecs』
使用符合 FIPS 標準的軟體來建立金鑰儲存庫，然後指定通道必須使用 FIPS 認證的 CipherSpecs。

第 34 頁的『IBM MQ 中的 NSA Suite B 加密法』
本主題提供如何配置 IBM MQ for AIX, Linux, and Windows 以符合套組 B 相容 TLS 1.2 設定檔的相關資訊。

第 17 頁的『國家安全域性 (NSA) Suite B 加密法』
美利堅合眾國政府就包括資料加密在內的 IT 系統和安全問題提供技術諮詢。 美國國家安全域性 (NSA) 在其 Suite B 標準中建議一組可交互作業的加密演算法。

通道鑑別記錄

若要在通道層次對授與連接系統的存取權進行更精確的控制，您可以使用通道鑑別記錄。

您可能會發現用戶端嘗試使用空白使用者 ID，或是容許用戶端執行不樂見動作的高階使用者 ID，來連接至佇列管理程式。 您可以使用通道鑑別記錄來封鎖對這些用戶端的存取。 此外，用戶端所主張的使用者 ID，

可能在用戶端平台上有效，但是在伺服器平台上則為不明或格式無效。 您可以使用通道鑑別記錄，將主張的使用者 ID 對映至有效使用者 ID。

您可能會發現連接至佇列管理程式的用戶端應用程式在某些方面行為不當。 若要保護伺服器免受此應用程式所造成問題的危害，則需要使用用戶端應用程式所在的 IP 位址來暫時封鎖此應用程式，直至更新防火牆規則或更正用戶端應用程式為止。 您可以使用通道鑑別記錄，來封鎖用戶端應用程式從其進行連接的 IP 位址。

如果已設定管理工具（例如 IBM MQ Explorer），以及該特定使用的通道，則您可能想要確保只有特定的用戶端電腦才能使用它。 您可以使用通道鑑別記錄，來容許只能從某些 IP 位址使用該通道。

如果您剛剛開始使用以用戶端身分執行的部分範例應用程式，請參閱 準備及執行範例程式 ，以取得使用通道鑑別記錄安全地設定佇列管理程式的範例。

若要取得通道鑑別記錄以控制入埠通道，請使用 MQSC 指令 **ALTER QMGR CHLAUTH(ENABLED)**。

CHLAUTH 規則適用於以回應新入埠連線所建立的通道 MCA。 對於在本端啟動的通道所建立的通道 MCA，未套用任何 **CHLAUTH** 規則。

表 *6:* 其中 *CHLAUTH* 規則適用於不同的通道配對

通道類型	**套用 CHLAUTH 規則的 MCA**
SDR-RCVR	RCVR
RQSTR-SVR（已啟動於 SVR）	RQSTR
RQSTR-SVR（已啟動於 RQSTR）	SVR
RQSTR-SDR（已啟動於 SDR）	RQSTR
RQSTR-SDR（啟動於 RQSTR）	初始連線的 SDR。 回呼連線的 RQSTR。

您可以建立通道鑑別記錄以執行下列功能：

- 封鎖來自特定 IP 位址的連線。
- 封鎖來自特定使用者 ID 的連線。
- 針對從特定 IP 位址連接的任何通道，設定要使用的 MCAUSER 值。
- 針對主張特定使用者 ID 的任何通道，設定要使用的 MCAUSER 值。
- 針對具有特定 SSL 或 TLS「識別名稱 (DN)」的任何通道，設定要使用的 MCAUSER 值。
- 針對從特定佇列管理程式連接的任何通道，設定要使用的 MCAUSER 值。
- 封鎖聲稱是來自某個佇列管理程式的連線，除非該連線來自特定的 IP 位址。
- 封鎖提供某個 SSL 或 TLS 憑證的連線，除非該連線來自特定的 IP 位址。

下列幾節會進一步說明這些用途。

您可以使用 MQSC 指令 **SET CHLAUTH** 或 PCF 指令 **Set Channel Authentication Record** 來建立、修改或移除通道鑑別記錄。

註: 大量通道鑑別記錄可能會對佇列管理程式的效能產生負面影響。

封鎖 IP 位址

防止從某些 IP 位址進行存取，通常是防火牆所扮演的角色。 不過，可能發生連線嘗試從 IP 位址無法存取 IBM MQ 系統，且必須暫時封鎖位址，才能更新防火牆之前的連線嘗試。 這些連線嘗試可能不是來自 IBM MQ 通道；這些連線嘗試可能是來自錯誤配置成目標 IBM MQ 接聽器的其他 Socket 應用程式。 透過設定 BLOCKADDR 類型的通道鑑別記錄，即可封鎖 IP 位址。 您可以指定一個以上的單一位址、位址範圍或包括萬用字元的型樣。

只要已啟用通道事件且佇列管理程式正在執行中，則每當入埠連線因以此方式封鎖 IP 位址而遭到拒絕時，就會發出一則事件訊息 MQRC_CHANNEL_BLOCKED，其中包含原因限定元 MQRQ_CHANNEL_BLOCKED_ADDRESS。 此外，連線會已保留開啟 30 秒之後才會傳回此錯誤，以確保接聽器不會因為不斷重複封鎖連線的嘗試而被癱瘓。

若只要封鎖特定通道上的 IP 位址，或避免在報告錯誤之前發生延遲，請使用 USERSRC(NOACCESS) 參數設定 ADDRESSMAP 類型的通道鑑別記錄。

只要已啟用通道事件且佇列管理程式正在執行中，則每當入埠連線因此原因而遭到拒絕時，會發出一則事件訊息 MQRC_CHANNEL_BLOCKED，其中包含原因限定元 MQRQ_CHANNEL_BLOCKED_NOACCESS。

如需範例，請參閱 第 319 頁的『封鎖特定 IP 位址』。

封鎖使用者 ID

若要防止某些使用者 ID 透過用戶端通道進行連接，請設定 BLOCKUSER 類型的通道鑑別記錄。 此類型的通道鑑別記錄僅適用於用戶端通道，不適用於訊息通道。 您可以指定一個以上要封鎖的個別使用者 ID，但不能使用萬用字元。

只要已啟用通道事件，則每當因此原因而拒絕入埠連線時，均會發出原因限定元為 MQRQ_CHANNEL_BLOCKED_USERID 的事件訊息 MQRC_CHANNEL_BLOCKED。

如需範例，請參閱 第 321 頁的『封鎖特定使用者 ID』。

您也可以透過使用 USERSRC(NOACCESS) 參數設定 USERMAP 類型的通道鑑別記錄，來封鎖指定使用者 ID 在某些通道上進行的任何存取。

只要已啟用通道事件且佇列管理程式正在執行中，則每當入埠連線因此原因而遭到拒絕時，會發出一則事件訊息 MQRC_CHANNEL_BLOCKED，其中包含原因限定元 MQRQ_CHANNEL_BLOCKED_NOACCESS。

如需範例，請參閱 第 323 頁的『封鎖存取用戶端使用者 ID』。

封鎖佇列管理程式名稱

若要將從指定佇列管理程式連接的所有通道，指定為沒有存取權，請使用 USERSRC(NOACCESS) 參數設定 QMGRMAP 類型的通道鑑別記錄。 您可以指定單一佇列管理程式名稱或包括萬用字元的型樣。 沒有同等的 BLOCKUSER 函數可封鎖從佇列管理程式進行的存取。

只要已啟用通道事件且佇列管理程式正在執行中，則每當入埠連線因此原因而遭到拒絕時，會發出一則事件訊息 MQRC_CHANNEL_BLOCKED，其中包含原因限定元 MQRQ_CHANNEL_BLOCKED_NOACCESS。

如需範例，請參閱 第 323 頁的『封鎖從遠端佇列管理程式存取』。

封鎖 SSL 或 TLS DN

若要將提供的 SSL 或 TLS 個人憑證包含指定 DN 的所有使用者，指定為沒有存取權，請使用 USERSRC(NOACCESS) 參數設定 SSLPEERMAP 類型的通道鑑別記錄。 您可以指定單一識別名稱或包括萬用字元的型樣。 沒有同等的 BLOCKUSER 函數可封鎖對 DN 的存取。

只要已啟用通道事件且佇列管理程式正在執行中，則每當入埠連線因此原因而遭到拒絕時，會發出一則事件訊息 MQRC_CHANNEL_BLOCKED，其中包含原因限定元 MQRQ_CHANNEL_BLOCKED_NOACCESS。

如需範例，請參閱 第 324 頁的『封鎖存取 SSL 或 TLS 識別名稱』。

將 IP 位址對映至要使用的使用者 ID

若要將從指定 IP 位址連接的所有通道，指定為使用特定的 MCAUSER，請設定 ADDRESSMAP 類型的通道鑑別記錄。 您可以指定單一位址、位址範圍或包括萬用字元的型樣。

如果您使用埠轉遞程式、DMZ 階段作業岔斷或變更提供給佇列管理程式的 IP 位址的任何其他設定，則對映 IP 位址不一定適合您使用。

如需範例，請參閱 第 324 頁的『將 IP 位址對映至 MCAUSER 使用者 ID』。

將佇列管理程式名稱對映至要使用的使用者 ID

若要指定從指定佇列管理程式連接的所有通道將使用特定的 MCAUSER，請設定 QMGRMAP 類型的通道鑑別記錄。 您可以指定單一佇列管理程式名稱或包括萬用字元的型樣。

如需範例，請參閱 第 321 頁的『將遠端佇列管理程式對映至 MCAUSER 使用者 ID』。

將用戶端主張的使用者 ID 對映至要使用的使用者 ID

若要指定，如果某個使用者 ID 是由 IBM MQ MQI 用戶端的連線所使用，則會使用不同的指定 MCAUSER，請設定 USERMOAP 類型的通道鑑別記錄。 使用者 ID 對映不會使用萬用字元。

如需範例，請參閱 第 322 頁的『將用戶端使用者 ID 對映至 MCAUSER 使用者 ID』 。

將 SSL 或 TLS DN 對映至要使用的使用者 ID

若要指定提供的 SSL/TLS 個人憑證包含指定 DN 的所有使用者將使用特定的 MCAUSER，請設定 SSLPEERMAP 類型的通道鑑別記錄。 您可以指定單一識別名稱或包括萬用字元的型樣。

如需範例，請參閱 第 322 頁的『將 SSL 或 TLS 識別名稱對映至 MCAUSER 使用者 ID』 。

根據 IP 位址對映佇列管理程式、用戶端或 SSL 或 TLS DN

在某些情況下，第三方可能會盜用佇列管理程式名稱。 也可能會竊取及重複使用 SSL 或 TLS 憑證或是金鑰資料庫檔。 若要避免受到這些威脅，您可以指定來自某個佇列管理程式或用戶端的連線，或是使用某個 DN 的連線必須從指定的 IP 位址進行連線。 設定類型為 USERMAP、QMGRMAP 或 SSLPEERMAP 的通道鑑別記錄，並使用 ADDRESS 參數指定允許的 IP 位址或 IP 位址型樣。

如需範例，請參閱 第 321 頁的『將遠端佇列管理程式對映至 MCAUSER 使用者 ID』 。

通道鑑別記錄之間的互動

嘗試建立連線的一個通道可能會符合多個通道鑑別記錄，而且這些記錄具有相互矛盾的效果。 例如，通道所主張的使用者 ID，可能會被 BLOCKUSER 通道鑑別記錄所封鎖，但它擁有的 SSL 或 TLS 憑證，卻符合設定不同使用者 ID 的 SSLPEERMAP 記錄。 此外，如果通道鑑別記錄使用萬用字元，則單一 IP 位址、佇列管理程式名稱或是 SSL 或 TLS DN，可能會符合數個型樣。 例如，IP 位址 192.0.2.6 符合型樣 192.0.2.0-24、192.0.2.* 及 192.0.*.6。 所採取的動作將按以下方式來決定。

- 所使用的通道鑑別記錄將以下列方式來選取:
  - 明確符合通道名稱的通道鑑別記錄，優先於使用萬用字元符合通道名稱的通道鑑別記錄。
  - 使用 SSL 或 TLS DN 的通道鑑別記錄，優先於使用使用者 ID、佇列管理程式名稱或 IP 位址的記錄。
  - 使用使用者 ID 或佇列管理程式名稱的通道鑑別記錄，優先於使用 IP 位址的記錄。
- 如果找到相符的通道鑑別記錄，而且指定了 MCAUSER，則會將此 MCAUSER 指派給通道。
- 如果找到相符的通道鑑別記錄，而且指定了通道沒有存取權，則會將 MCAUSER 值 *NOACCESS 指派給通道。 此值稍後可由安全結束程式加以變更。
- 如果找不到相符的通道鑑別記錄，或找到相符的通道鑑別記錄，而且它指定要使用通道的使用者 ID，則會檢查 MCAUSER 欄位。
  - 如果 MCAUSER 欄位是空白，則會將用戶端使用者 ID 指派給通道。
  - 如果 MCAUSER 欄位不是空白，則會將其指派給通道。
- 執行任何安全結束程式。 此結束程式可能會設定通道使用者 ID，或是決定是否封鎖存取。
- 如果封鎖該連線或 MCAUSER 已設定為 *NOACCESS，則通道會結束。
- 如果不封鎖連線，則會根據已封鎖使用者清單，檢查之前步驟中所決定的任何通道（用戶端通道除外）的通道使用者 ID。
  - 如果該使用者 ID 位在已封鎖的使用者清單中，則通道會結束。
  - 如果該使用者 ID 不在已封鎖的使用者清單中，則通道會執行。

如果有多個通道鑑別記錄與通道名稱、IP 位址、主機名稱、佇列管理程式名稱或者 SSL 或 TLS DN 相符，則會使用最具體的相符項。 此相符項被認為是:

- 最具體是指沒有萬用字元的名稱，例如:
  - 通道名稱 A.B.C
  - IP 位址 192.0.2.6
  - `hursley.ibm.com` 的主機名稱
  - 佇列管理程式名稱 192.0.2.6

- 最通用是指符合的單一星號 (*)，例如：
  - 所有通道名稱
  - 所有 IP 位址
  - 所有主機名稱
  - 所有佇列管理程式名稱
- 在字串開頭使用星號的型樣比在字串開頭使用定義值的型樣更通用：
  - 對於通道，*.B.C 比 A.* 更通用
  - 對於 IP 位址，*.0.2.6 比 192.* 更通用
  - 對於主機名稱，`*.ibm.com` 比 `hursley.*` 更通用
  - 對於佇列管理程式名稱，*QUEUEMANAGER 比 QUEUEMANAGER* 更通用
- 在字串中特定位置使用星號的型樣比在字串中相同位置使用定義值的型樣更通用，對於字串中後續的每個位置同樣如此：
  - 對於通道，A.*.C 比 A.B.* 更通用
  - 對於 IP 位址，192.*.2.6 比 192.0.*. 更通用
  - 對於主機名稱，`hursley.*.com` 比 `hursley.ibm.*` 更通用
  - 對於佇列管理程式名稱，Q*MANAGER 比 QUEUE* 更通用
- 如果兩個以上的型樣在字串中的特定位置使用星號，則星號後的節點更少的型樣更通用：
  - 若為通道，A.* 比 A.*.C 更通用
  - 若為 IP 位址，192.* 比 192.*.2.*. 更通用
  - 對於主機名稱，`hurlsey.*` 比 `hursley.*.com` 更通用
  - 對於佇列管理程式名稱，Q* 比 Q*MGR 更通用
- 此外，對於 IP 位址：
  - 使用連字號 (-) 指出的範圍比星號更具體。 因此，192.0.2.0-24 比 192.0.2.* 更具體。
  - 屬於另一個範圍子集的範圍，比較大的範圍更具體。 因此，192.0.2.5-15 比 192.0.2.0-24 更具體。
  - 不允許範圍重疊。 例如，您不得同時有 192.0.2.0-15 及 192.0.2.10-20 的通道鑑別記錄。
  - 型樣不能少於需要的部分數目，除非型樣結尾是一個尾端星號。 例如 192.0.2 無效，但 192.0.2 .* 有效。
  - 必須使用適當的部分分隔字元（點 (.) 適用於 IPv4，冒號 (:) 適用於 IPv6），來分隔尾端星號與位址的其餘部分。 例如，192.0* 無效，因為星號不在它自己的部分中。
  - 型樣可以包含額外的星號，前提是沒有星號與尾端星號相鄰。 例如，192.*.2.* 有效，但 192.0.*.* 是無效的。
  - IPv6 位址型樣不能包含一個雙冒號和一個尾端星號，因為產生的位址會不夠明確。 例如，2001::* 可以擴充為 2001:0000:*、2001:0000:0000:* 等
- 對於 SSL 或 TLS「識別名稱 (DN)」，子字串的優先順序如下：

表 7: 子字串的優先順序

訂購	**DN 子字串**	**名稱**
1	SERIALNUMBER=	憑證序號
2	MAIL=	電子郵件位址
3	E=	電子郵件位址（已淘汰，最好使用 MAIL）
4	UID=, USERID=	使用者 ID
5	CN=	通用名稱

表 7: 子字串的優先順序 (繼續)		
訂購	**DN 子字串**	**名稱**
6	T =	標題
7	OU=	組織單位
8	DC=	網域元件
9	O=	組織
10	STREET=	街道/地址的第一行
11	L=	地區
12	ST=, SP=, S=	州/省（縣/市）名稱
13	PC =	郵遞區號
14	C=	國家/地區
15	UNSTRUCTUREDNAME=	主機名稱
16	UNSTRUCTUREDADDRESS=	IP 位址
17	DNQ=	識別名稱限定元

因此，如果使用包含子字串 O=IBM 及 C=UK 的 DN 來呈現 SSL 或 TLS 憑證， IBM MQ 會使用 O=IBM 的通道鑑別記錄（若為 C=UK 的話），如果兩者都存在的話，則使用此記錄。

一個 DN 可以包含多個 OU，這些 OU 必須以階層式順序來指定（先指定大型組織單位）。 如果兩個 DN 除了其 OU 值以外，其餘所有方面皆相等，則按以下方式來決定更具體的 DN:

1. 如果它們具有不同數量的 OU 屬性，則 OU 值最多的 DN 更具體。 這是因為具有更多組織單位的 DN，可以更詳細地全面限定 DN，進而提供更符合的準則。 即使其最上層 OU 是萬用字元 (OU=*)，從整體上仍會將 OU 較多的 DN 視為更具體。
2. 如果它們具有相同數量的 OU 屬性，則會根據下列規則，按照從左到右的順序來比較相對應的 OU 值配對，其中最左側的 OU 是最高層次（最不具體）。
   a. 沒有萬用字元值的 OU 是最具體的 OU，因為它只會與一個字串完全相符。
   b. 開頭或結尾有單一萬用字元的 OU（例如，OU=ABC* 或 OU=*ABC），是下一個最具體的 OU。
   c. 有兩個萬用字元的 OU（例如，OU=*ABC*），是再下一個最具體的 OU。
   d. 只由一個星號組成的 OU (OU=*) 最不具體。
3. 如果字串比較發現兩個屬性值的具體程度相同，則較長的屬性字串更具體。
4. 如果字串比較發現兩個屬性值的具體程度和長度相同，則結果將由 DN 部分（不包含任何萬用字元）的不區分大小寫的字串比較來確定。

如果兩個 DN 在所有方面都相等，但其 DC 值除外，則相同相符規則會套用為 OU，但在 DC 值中，最左邊的 DC 是最低層次（最具體的），且比較排序也會因此而不同。

顯示通道鑑別記錄

若要顯示通道鑑別記錄，請使用 MQSC 指令 **DISPLAY CHLAUTH** 或 PCF 指令 **Inquire Channel Authentication Records**。 您可以選擇傳回符合所提供通道名稱的所有記錄，也可以選擇傳回明確的相符項。 明確的相符項會告訴您，如果某個通道嘗試從特定 IP 位址、特定佇列管理程式或使用特定使用者 ID 來建立連線，以及選擇性地提供包含指定 DN 的 SSL/TLS 個人憑證來建立連線，您將使用哪個通道鑑別記錄。

相關概念

第 81 頁的『遠端傳訊的安全』
本節處理安全的遠端傳訊層面。

CHLAUTH 與 *CONNAUTH* 的互動

在通道上單一交談的情況下，通道鑑別記錄 (CHLAUTH) 及連線鑑別 (CONNAUTH) 在 IBM MQ 中如何互動。

不同類型的連結

IBM MQ 支援兩種應用程式連接方法:

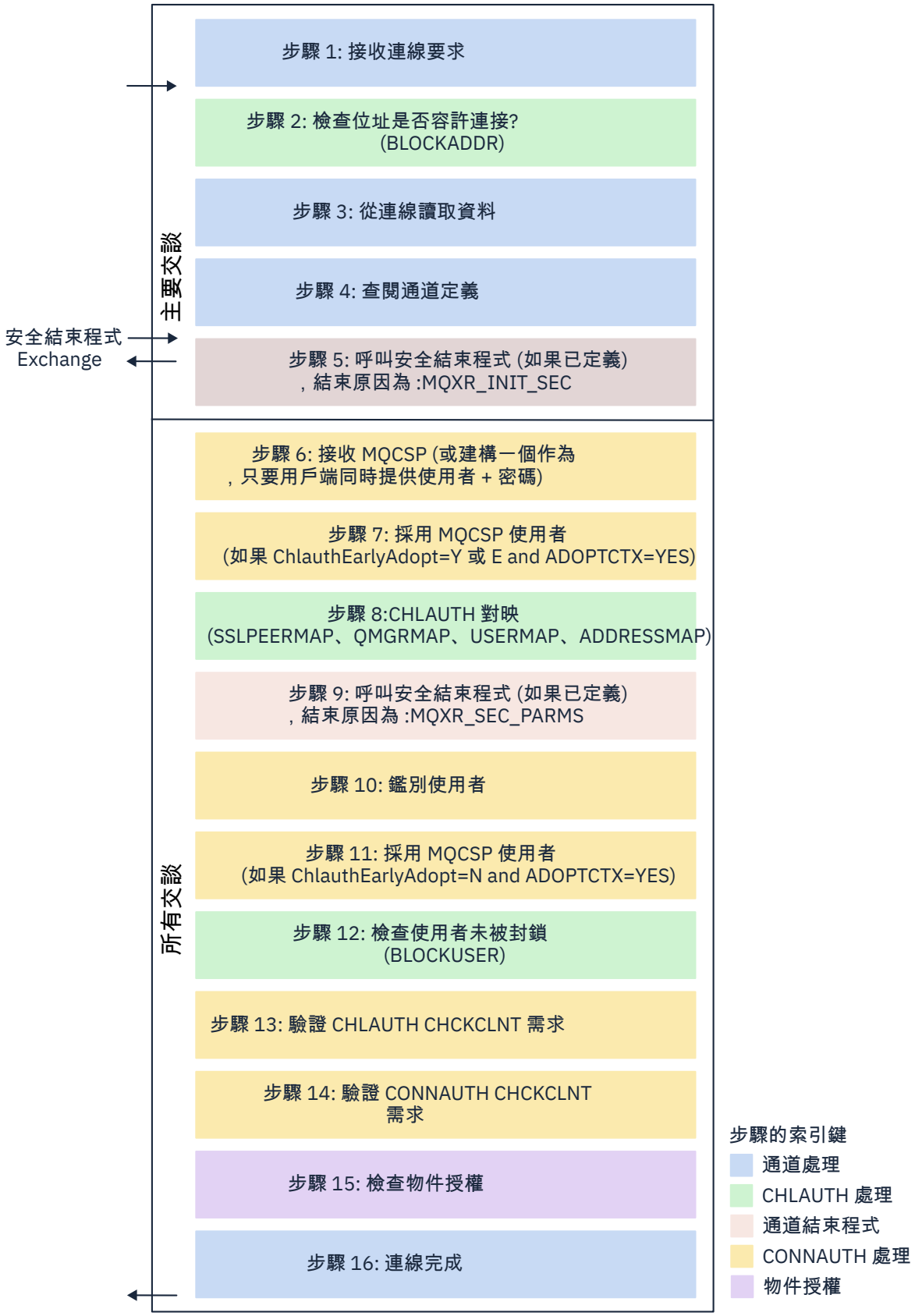
本端連結
當應用程式和佇列管理程式位於相同的作業映像檔時適用。 CHLAUTH 與此類型的應用程式連線無關。

用戶端連結
當應用程式和佇列管理程式使用網路進行通訊時適用。 應用程式和佇列管理程式可以在相同機器上執行，也可以在不同機器上執行。 在 IBM MQ 中，用戶端連線是以伺服器連線 (SVRCONN) 通道的形式處理，在此情況下， CONNAUTH 和 CHLAUTH 都適用。

通道接收端的連結步驟

當應用程式連接至佇列管理程式時，會執行大量檢查，以確保通道兩端瞭解另一端支援的內容。 通道的接收端會執行一些額外檢查，包括 CHLAUTH 及 CONNAUTH ，以確保容許用戶端連接，而且此處理程序也可能包括安全結束程式，因為這可能會影響結果。 此通道連接階段也稱為 連結階段。

下圖列出當伺服器端 (在佇列管理程式上) 啟動時 SVRCONN 通道所經歷的步驟:



步驟 1: 接收連線要求
通道起始程式或接聽器會從網路上的某個地方接收連線要求。

步驟 2: 是否容許位址連接?
在讀取任何資料之前， IBM MQ 會根據 CHLAUTH 規則檢查夥伴的 IP 位址，以查看位址是否在 *BLOCKADDR* 規則中。 如果找不到位址，因此未封鎖，則流程會繼續進行下一步。

步驟 3: 從通道讀取資料
IBM MQ 現在會將資料讀取至緩衝區，並開始處理已傳送的資訊。

步驟 4: 查閱通道定義
在第一個資料流程中，除其他事項外， IBM MQ 會傳送傳送端嘗試啟動的通道名稱。 然後，接收端佇列管理程式可以查閱通道定義，其具有為通道指定的所有設定。

步驟 5: 呼叫安全結束程式 (如果已定義)
如果通道已定義安全結束程式 (SCYEXIT) ，則會以結束原因 (MQCXP.**ExitReason**) 來呼叫它。 設為 MQXR_INIT_SEC。

步驟 6: 接收 MQCSP
必要的話，只要用戶端提供使用者 ID 和密碼，即可建構一個。

如果用戶端是在相容模式下執行的 Java 或 JMS 應用程式，則用戶端不會將 MQCSP 結構傳遞至佇列管理程式。 相反地，如果應用程式提供使用者 ID 和密碼，則會在這裡建構 MQCSP 結構。

步驟 7: 採用 MQCSP 使用者 (如果 ChlauthEarlyAdopt 是 *Y* 且 ADOPTCTX=YES)
鑑別用戶端所主張的使用者 ID。

如果 CONNAUTH 使用 LDAP 將主張的識別名稱對映至簡短使用者 ID ，則會在此步驟中進行對映。

如果鑑別成功，通道會採用使用者 ID ，並由 CHLAUTH 對映步驟使用。

註: 從 IBM MQ 9.0.4 中， **ChlauthEarlyAdopt=** *Y* 參數會自動新增至新佇列管理程式之 qm.ini 檔的通道段落。

步驟 8 :CHLAUTH 對映
重新檢查 CHLAUTH 快取，以尋找對映規則 *SSLPEERMAP*、 *USERMAP*、 *QMGRMAP* 及 *ADDRESSMAP*。

使用最明確符合送入通道的規則。 如果規則具有 **USERSRC***(CHANNEL)* 或 *(MAP)*，則通道會繼續進行連結。

如果 CHLAUTH 規則評估為具有 **USERSRC***(NOACCESS)*的規則，則會阻止應用程式連接至通道，除非隨後在步驟 9 中以有效使用者 ID 及密碼置換認證。

步驟 9: 呼叫安全結束程式 (如果已定義)
如果通道已定義安全結束程式 (SCYEXIT) ，則會以結束原因 (MQCXP.**ExitReason**) 來呼叫它。 設為 MQXR_SEC_PARMS。

MQCXP 結構的 **SecurityParms** 欄位中會呈現指向 MQCSP 的指標。

MQCSP 結構具有指向使用者 ID (MQCSP.**CSPUserIdPtr**) 的指標 及密碼 (MQCSP.**CSPPasswordPtr**)。

可以在結束程式中變更使用者 ID 和密碼。 下列範例顯示安全結束程式如何將使用者 ID 和密碼值列印至審核日誌:

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

結束程式可以透過在 MQCXP 中傳回 *MQXCC_CLOSE_CHANNEL* 來告知 IBM MQ 關閉通道。 **Exitresponse** 欄位。 否則，通道處理程序會繼續進行連線鑑別階段。

註: 如果安全結束程式變更主張的使用者，則不會將 CHLAUTH 對映規則重新套用至新使用者。

步驟 10: 鑑別使用者
如果在佇列管理程式上啟用 CONNAUTH ，則會發生鑑別階段。

若要檢查此狀況，請發出 MQSC 指令 'DISPLAY QMGR CONNAUTH'。

z/OS 下列範例顯示在 IBM MQ for z/OS 上執行的佇列管理程式中 **DISPLAY QMGR CONNAUTH** 指令的輸出。

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

Multi 下列範例顯示來自在 IBM MQ for Multiplatforms 上執行之佇列管理程式的指令 '**DISPLAY QMGR CONNAUTH**' 輸出。

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

CONNAUTH 值是 **AUTHINFO** IBM MQ 物件的名稱。

由於作業系統鑑別 (**AUTHTYPE**(*IDPWOS*)) 同時適用於 IBM MQ for Multiplatforms 和 IBM MQ for z/OS，因此範例使用作業系統鑑別。

z/OS 下列範例顯示在 IBM MQ for z/OS 上執行的佇列管理程式中 **AUTHTYPE**(*IDPWOS*) 的原廠預設物件。

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

Multi 下列範例顯示在 IBM MQ for Multiplatforms 上執行的佇列管理程式中 **AUTHTYPE**(*IDPWOS*) 的原廠預設物件。

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

AUTHINFO TYPE (IDPWOS) 具有稱為 CHCKCLNT 的屬性。 如果值變更為 *REQUIRED* ，則所有用戶端應用程式都必須提供有效的使用者 ID 和密碼。

如果已在步驟 7 中鑑別使用者，則不會再次鑑別使用者，除非步驟 9 中 MQCXP 結構的 SecurityParms 欄位中的使用者或密碼已由安全結束程式變更。

步驟 11: 採用 MQCSP 使用者的環境定義 (如果 ChlauthEarlyAdopt=*N* 且 ADOPTCTX=YES)

您可以設定 ADOPTCTX 屬性，以控制通道是在 MCAUSER 下執行，還是應用程式提供的使用者 ID。

如果已順利鑑別 MQCSP 或 MQXCP 結構的 **SecurityParms** 欄位中所主張的使用者 ID ，且 **ADOPTCTX** 是 *YES*，則會採用步驟 7 和 8 所產生的使用者環境定義作為要用於此應用程式的環境定義，除非步驟 9 中安全結束程式已變更 MQCXP 結構的 **SecurityParms** 欄位中的使用者或密碼。

這個主張的使用者 ID 是檢查授權使用 IBM MQ 資源的使用者 ID。

例如，您在 SVRCONN 通道上未設定 MCAUSER，且您的用戶端在 Linux 機器上的 'johndoe' 下執行。您的應用程式在 MQCSP 中指定使用者 'fred'，因此通道會開始以 'johndoe' 作為作用中 MCAUSER 來執行。 CONNAUTH 檢查之後，會採用使用者 'fred'，且通道會以 'fred' 作為作用中 MCAUSER 來執行。

步驟 12: 檢查使用者未被封鎖 (BLOCKUSER)
如果 **CONNAUTH** 檢查成功，則會重新檢查 CHLAUTH 快取，以檢查 *BLOCKUSER* 規則是否封鎖作用中的 MCAUSER。 如果使用者被封鎖，則通道會結束。

Step13: 驗證 CHLAUTH CHCKCLNT 需求
如果在步驟 8 中選取的 CHLAUTH 規則額外指定 REQUIRED 或 REQDADM 的 CHCKCLNT 值，則會執行驗證，以確保提供有效的 CONNAUTH 使用者 ID 符合需求。

- 如果設定 CHCKCLNT (REQUIRED)，則必須已在步驟 7 或 10 中鑑別使用者。 否則會拒絕連線。
- 如果設定 CHCKCLNT (REQDADM)，則在步驟 7 或 10 中必須已鑑別使用者 (如果此連線判定為特許)。否則會拒絕連線。
- 如果設定 CHCKCLNT (ASQMGR)，則會跳過此步驟。

附註:

1. 如果已設定 CHCKCLNT (REQUIRED) 或 CHCKCLNT (REQDADM)，但未在佇列管理程式上啟用 CONNAUTH，則由於配置中的衝突，連線會失敗並傳回 MQRC_SECURITY_ERROR (2063) 回覆碼。
2. 在此步驟中未重新鑑別使用者。

步驟 14: 驗證 CONNAUTH CHCKCLNT 需求。
如果在佇列管理程式上啟用 CONNAUTH，則會發生鑑別階段。

會檢查 CONNAUTH CHCKCLNT 值，以判定送入連線的需求:

- 如果設定 CHCKCLNT (NONE)，則會跳過此步驟
- 如果設定 CHCKCLNT (OPTIONAL)，則會跳過此步驟。
- 如果設定 CHCKCLNT (REQUIRED)，則必須已在步驟 7 或 10 中鑑別使用者。 否則會拒絕連線。
- 如果設定 CHCKCLNT (REQDADM)，則在步驟 7 或 10 中必須已鑑別使用者 (如果此連線判定為特許)。否則會拒絕連線。

註: 在此步驟中未重新鑑別使用者。

Multi **步驟 15: 檢查物件授權**
進行檢查以確保作用中 MCAUSER 具有適當的權限來連接至佇列管理程式。

ALW 如需相關資訊，請參閱 物件權限管理程式。

IBM i 如需相關資訊，請參閱 第 130 頁的『IBM i 上的物件權限管理程式』。

步驟 16: 連線完成
如果上述步驟順利完成，則連線會完成。

相關概念
CONNAUTH
佇列管理程式可以配置成使用提供的使用者 ID 和密碼，來檢查使用者是否有權存取資源。

相關參考
SET CHLAUTH
ALTER AUTHINFO

解決 *CHLAUTH* 存取問題
關於使用通道鑑別記錄 (CHLAUTH) 時如何解決特定存取問題的建議。

預設 CHLAUTH 規則

CHLAUTH 處理有三個預設規則:

- 任何 MQ-admin* 使用者都不存取所有通道

- 不存取所有 SYSTEM.* 所有使用者的通道
- 具有 SYSTEM.ADMIN.SVRCONN 通道 (非 MQ-admin 使用者)

前兩個規則會封鎖所有通道的存取。 第三個規則更具體，因此如果通道是 SYSTEM.ADMIN.SVRCONN 通道，因此容許存取該通道。

一般連線錯誤

CHLAUTH 規則用來判斷通道是否可以啟動，且它們容許透過 MCAUSER 對映至另一個使用者 ID。 如果無法啟動通道，通常會發生下列錯誤:

- RC 2035 MQRC_NOT_AUTHORIZED
- RC 2059 MQRC_Q_MGR_NOT_AVAILABLE
- AMQ4036 不允許存取
- AMQ9776: 使用者 ID 已封鎖通道
- AMQ9777: 已封鎖通道
- MQJE001: 發生 MQException: 完成碼 2 ，原因 2035
- MQJE036: 佇列管理程式拒絕連線嘗試

您應該嚴格封鎖存取，然後新增更多 CHLAUTH 規則來控制誰可以存取及啟動通道。 作為暫時措施，如果要對列出的錯誤進行疑難排解，您可以執行下列動作:

- 第 51 頁的『停用 CHLAUTH 規則』
- 第 51 頁的『修改或移除 CHLAUTH 規則』

停用 CHLAUTH 規則

作為暫時測量，也為了疑難排解上述錯誤，您可以停用 CHLAUTH 規則。 可以隨時重新啟用規則，如果停用 CHLAUTH 規則可解決連線問題，則您知道這是原因。

若要停用 CHLAUTH 規則，請發出下列指令:

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

請注意，您也可以將 CHLAUTH 設為 *WARN*，以容許存取及記載規則的結果。

修改或移除 CHLAUTH 規則

您也可以刪除或修改造成問題的一或多個 CHLAUTH 規則。

若要修改 CHLAUTH 規則，請搭配使用 SET CHLAUTH 指令與 ACTION (REPLACE)。 例如，若要修改導致任何 MQ-admin 使用者無法存取所有通道的預設規則 WARN ，而不是被封鎖，請發出下列指令:

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

若要刪除 CHLAUTH 規則，請搭配使用 SET CHLAUTH 指令與 ACTION (REMOVE)。 例如，若要刪除導致任何 MQ-admin 使用者無法存取所有通道的預設規則，請發出下列指令:

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

使用 MATCH (RUNCHECK) 來測試存取權

您可以使用 runmqsc 中 CHLAUTH 規則的 MATCH (*RUNCHECK*) 選項來測試 CHLAUTH 規則的結果。 如果特定入埠通道連接至此佇列管理程式，則 **MATCH** (*RUNCHECK*) 選項會傳回在執行時期符合特定入埠通道的記錄。 您必須提供:

- 通道名稱

- 「位址」屬性
- SSLPEER 屬性，只有在入埠通道使用 SSL 或 TLS 時
- QMNAME，如果入埠通道是佇列管理程式通道，或
- CLNTUSER 屬性 (如果入埠通道是用戶端通道)

下列範例會檢查具有預設規則的哪些 CHLAUTH 規則會導致 MQ-admin 使用者 johndoe 存取名為 CHAN1 的通道:

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

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

對於使用者 johndoe，通道不會執行，因為 *MQADMIN 使用者的 BLOCKUSER 規則會封鎖使用者。

下列範例會檢查具有預設規則的哪些 CHLAUTH 規則會導致非 MQ-admin 使用者的使用者 alice 存取名為 CHAN1 的通道:

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

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

對於使用者 alice，通道會執行，且通道會以 MCAUSER 身分傳入 alice。 MCAUSER 是用來檢查 IBM MQ 物件權限的使用者 ID。

相關參考

SET CHLAUTH

DISPLAY CHLAUTH

為使用者建立新的 *CHLAUTH* 規則

使用者的部分一般實務範例，以及用來達成這些目的的範例 CHLAUTH 規則。

本主題包含下列實務範例:

- 第 52 頁的『控制特定 MQ-admin 使用者的存取權』
- 第 53 頁的『控制特定使用者及 IBM MQ 用戶端應用程式的存取權』
- 第 53 頁的『使用特定使用者的憑證識別名稱 (DN) 來控制該使用者的存取權』
- 第 54 頁的『將特定使用者對映至 mqm 使用者』

控制特定 MQ-admin 使用者的存取權

在此情況下，請設定伺服器連線通道，以專門用於管理視景，亦即，從 IBM MQ Explorer 進行連接。 如果連線不是來自其中一個指定的 IP 位址，則您具有此用法的特定通道，並定義要從中接受連線的一或多個 IP 位址，以及 'mqm' ID 的封鎖存取權。

為稱為 ADMIN.CHAN 的 IBM MQ Explorer 及 MQ-admin 使用者建立 SVRCONN 通道:

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

若要測試，請確定您已定義 MQ-admin 群組中的使用者，但未定義。 在此實務範例中， mqadm 是在 MQ-admin 群組中，而 alice 不是。

預設 CHLAUTH 規則 已備妥。 新增三個規則以容許特定使用者存取 ADMIN.CHAN as MQ-admin :

- 從任何位址設定 NOACCESS
- 將此通道的 BLOCKUSER 設為僅封鎖使用者 nobody，這會置換 *MQADMIN BLOCKUSER

- 允許存取特定位址子網路上的使用者 mqadm，並對映至 mqadm 使用者權限

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

此時，使用者 mqadm 可以存取並啟動 ADMIN.CHAN 通道，從指定的 IP 位址範圍。

您可以隨時執行 MATCH (RUNCHECK)，以查看下列每一個指令的結果:

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

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

此時，只容許具有 CHLAUTH 記錄的使用者使用 ADMIN.CHAN。

控制特定使用者及 IBM MQ 用戶端應用程式的存取權

在此實務範例中，預設 CHLAUTH 規則已足夠，假設應該為特定使用者設定 IBM MQ 權限，以提供正確的 IBM MQ 權限 (使用 setmqaut)。

在此實務範例中，為非 MQ-admin 使用者的使用者 mqapp1 設定權限。 建立 SVRCONN 通道 APP1.CHAN，由特定應用程式及特定使用者使用。

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

在具備 預設 CHLAUTH 規則 的情況下，使用者 mqapp1 可以啟動 APP1.CHAN 通道。

來自 IBM MQ 用戶端應用程式的使用者 ID 用於 IBM MQ 物件權限檢查。 在此情況下，假設 'mqapp1' 使用者正在執行 IBM MQ 用戶端應用程式，這將用於 IBM MQ 物件權限檢查。 因此，如果 mqapp1 對應用程式所需的 IBM MQ 物件具有存取權，則一切正常; 如果沒有，則會發生權限錯誤。

您可以為 mqapp1 使用者 ID 建立特定的 CHLAUTH 規則，以進一步提高安全，但在預設規則下， MQ-admin 群組的任何成員都無法存取此通道。

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

使用特定使用者的憑證識別名稱 (DN) 來控制該使用者的存取權

在此實務範例中，使用者必須具有傳送至佇列管理程式的憑證。 然後， DN 會與 CHLAUTH 規則的 SSLPEER 設定進行比對，且 SSLPEER 可以使用萬用字元。

如果相符，也可以將使用者對映至不同的 MCAUSER，以檢查 IBM MQ 物件權限。 對映 MCAUSER 可將需要在 IBM MQ 物件權限管理程式 (OAM) 中管理的使用者數目減至最少。

您具有使用中憑證的 TLS 通道，且您需要規則來執行下列動作:

- 封鎖特定通道的所有使用者
- 僅容許具有特定 SSLPEER 的使用者使用該使用者的用戶端進行 IBM MQ OAM 存取。

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

通道上連接的用戶端使用者 ID 用於 IBM MQ 物件的 IBM MQ OAM 權限; 因此使用者 ID 必須具有適當的 IBM MQ 權限。

如果您想要的話，可以使用下列指令來對映至不同的 IBM MQ 使用者 ID:

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

而不是 USERSRC(CHANNEL)。

將特定使用者對映至 mqm 使用者

這是 第 52 頁的『控制特定 MQ-admin 使用者的存取權』的新增或修改。

新增下列 CHLAUTH 規則，以將特定使用者對映至在 IBM MQ OAM 中具有 IBM MQ 物件權限設定的 mqm 使用者或 MQ-admin 使用者 ID。

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

這容許並將 johndoe 使用者對映至 mqm 特定通道的使用者 ADMIN.CHAN。

相關概念

第 50 頁的『解決 CHLAUTH 存取問題』
關於使用通道鑑別記錄 (CHLAUTH) 時如何解決特定存取問題的建議。

第 54 頁的『建立通道的新 CHLAUTH 規則』
為了協助您建立自己的 CHLAUTH 規則，以下是通道的一些一般實務範例，以及用來達成這些目的的 CHLAUTH 規則範例。

相關參考

SET CHLAUTH
DISPLAY CHLAUTH

建立通道的新 *CHLAUTH* 規則
為了協助您建立自己的 CHLAUTH 規則，以下是通道的一些一般實務範例，以及用來達成這些目的的 CHLAUTH 規則範例。

本主題包含下列實務範例:

- 第 54 頁的『僅容許從特定 IP 位址範圍存取特定通道。』
- 第 55 頁的『對於特定通道，封鎖所有使用者，但容許特定使用者連接。』
- 第 55 頁的『將 CHLAUTH 用於接收端及傳送端通道』

僅容許從特定 IP 位址範圍存取特定通道。

針對此實務範例，您想要:

- 從任何地方設定通道無存取權
- 容許從特定 IP 位址或位址範圍存取

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

這只容許 APP2.CHAN 通道。

以 MCAUSER 身分連接的使用者會對映至 mqapp2，因此會取得該使用者的 IBM MQ OAM 權限。

對於特定通道，封鎖所有使用者，但容許特定使用者連接。

在此實務範例中，通道 MY.SVRCONN 的存取權已備妥 預設 CHLAUTH 規則 。

您需要新增下列:

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

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

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

程式碼的第一部分會封鎖任何人在 MY.SVRCONN 上進行連接，因此當連線來自特定使用者 ID johndoe 時，程式碼只容許啟動 MY.SVRCONN 通道。

在通道 johndoe 上連接的使用者用於 IBM MQ 物件的 IBM MQ OAM 權限。 因此，使用者 ID 必須具有適當的 IBM MQ 權限。

如果您想要的話，可以使用下列指令來對映至不同的 IBM MQ 使用者 ID:

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

而不是 USERSRC(CHANNEL)。

將 CHLAUTH 用於接收端及傳送端通道

您可以使用 CHLAUTH 規則，將額外安全新增至接收端及傳送端通道，以限制存取接收端通道。 請注意，如果您要新增或變更 CHLAUTH 規則，則只有在啟動通道時才會套用已更新的 CHLAUTH 規則，因此如果通道已在執行中，您需要停止並重新啟動它們，才能套用 CHLAUTH 更新項目。

CHLAUTH 規則可以在任何通道上使用，但有一些限制。 例如， USERMAP 規則僅適用於 SVRCONN 通道。

此範例僅容許來自特定 IP 位址的連線啟動 TO.MYSVR1 通道:

```
# First you could lock down the channel by disallowing all
# for channel 'TO.MYSVR1', RCVR channel
SET CHLAUTH('TO.MYSVR1') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)
DESCR('Back-stop rule')

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

此範例僅容許來自特定佇列管理程式的連線:

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

```
DESCR('Back-stop rule')

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

相關概念

第 50 頁的『解決 CHLAUTH 存取問題』
關於使用通道鑑別記錄 (CHLAUTH) 時如何解決特定存取問題的建議。

第 52 頁的『為使用者建立新的 CHLAUTH 規則』
使用者的部分一般實務範例，以及用來達成這些目的的範例 CHLAUTH 規則。

相關參考

SET CHLAUTH

DISPLAY CHLAUTH

建立 *CHLAUTH back-stop* 規則

當考慮控制佇列管理程式的入埠連線時，您有兩個選項。 您可以嘗試列出所有不容許的連線，或先說不容許所有連線，然後嘗試列出所有容許的連線。 這裡說明第二個選項。

關於這項作業

使用第二個選項的原因是，如果您嘗試列出所有不容許的連線，因此所有未列出的連線都容許在中，則清單中遺漏一個連線的結果是不應該容許的連線能夠連接，導致潛在的安全侵害。

相反地，如果您從不容許每一個連線開始，然後列出這些連線，則此清單中遺漏一個連線的結果不是安全侵害。 如果您的企業需要新增其他連線，這是一項相對簡單的作業，但沒有潛在的安全侵害。

要做的第一件事是建立 *back-stop* 規則，它是用來捕捉不符合更特定規則的任何連線的規則。 此規則會完全阻止任何遠端連線連接至佇列管理程式。

不過，如果您擔心此方法，則可以在警告模式下設定 *back-stop* 規則; 請參閱步驟 第 56 頁的『2』

程序

1. 若要建立後端規則，以停止連接至佇列管理程式的遠端連線，請發出下列指令:

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

   現在，您已關閉所有遠端連線上的門，您可以開始將更具體的規則放在適當的位置，以容許某些連線進入。 例如:

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

2. 如果您想要以警告模式建立 back-stop 規則，請發出下列指令:

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

   現在您可以繼續，並制定所有正面規則。 當您認為已建立所有需要的規則時，請發出下列指令來開啟頻道事件:

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

   並監視 SYSTEM.ADMIN.CHANNEL.EVENT 佇列，其中 **Reason** 設為 MQRC_CHANNEL_BLOCKED_WARNING。

   這些事件詳述符合您的支援停止規則的連線，但由於指令以警告模式執行，目前實際上並未被封鎖。

請檢閱每一個事件，並判斷此連線是否應該具有正面的規則來容許它進入，或它是否已正確符合 *back-stop* 規則。 您可以在此模式下執行，在建立事件時檢閱它們，直到您很高興看到所有入埠通道，並有適當的正向規則適用於所有這些通道。

此時，您可以發出下列指令來變更 *back-stop* 規則，以開始實際封鎖它所符合的連線:

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

相關資訊

SET CHLAUTH（建立或修改通道認證記錄）

建立非特許 *IBM MQ* 管理者
如何使用 CHLAUTH 建立非特許 IBM MQ 管理者。

關於這項作業

在此作業的環境定義中，術語如下:

特許使用者
表示使用者有權執行作業，但未明確授與執行該作業的存取權。 mqm 群組中的使用者是這些特許使用者的範例。

***IBM MQ* 管理者**
表示使用者需要對 IBM MQ 發出管理指令，例如 **DEFINE QLOCAL** 或 **START CHANNEL**。

下列步驟會建立非特許 IBM MQ 管理者。

程序

1. 使用您企業使用的平台或平台適用的指令，在佇列管理程式機器上建立使用者 ID。
   在此範例中使用使用者名稱 `alice` 。
2. 執行下列程序，授與這個新的使用者權限來發出所有 IBM MQ 管理指令:
   a) 使用特許使用者來啟動 IBM MQ Explorer 。
   b) 選取適當的佇列管理程式，然後選取 **物件權限** 及 **新增角色型權限**，以導覽至「角色型精靈」。
   c) 在蹦現的精靈畫面中，輸入您在第一個步驟中建立的使用者 ID，或者如果您偏好使用群組，請輸入您要成為非特許 IBM MQ 管理者的使用者或一組使用者的群組名稱。
   d) 設定精靈的完整管理存取權。
   e) 如果您想要讓非特許 IBM MQ 管理者能夠瀏覽佇列上的訊息，請同時選取該勾選框。
   f) 在精靈底端的預覽畫面中檢閱指令。
   您可以剪下並貼上這些指令，以建置您自己的 Script。

   您可能偏好使用自己的 Script 來執行此動作的原因之一，是要減少您提供給此使用者的存取權數量。您可能偏好只授與對特定物件群組的存取權，而不是授與對所有物件的存取權。

   在精靈上按 **確定** 會發出顯示的指令。
   g) 如果非特許 IBM MQ 管理者的需求也適用於遠端存取，則您需要設定一些 CHLAUTH 規則，以容許此使用者 ID 進行遠端存取。
   假設您的企業正在使用 第 56 頁的『建立 CHLAUTH back-stop 規則』中的指引，您只需要新增啟用規則即可。
   您建立的規則視您選擇如何鑑別遠端 IBM MQ 管理者而定。

   如果您使用弱式 TCP/IP 鑑別，則可以設定看起來如下的 CHLAUTH 規則:

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

   9. 如果您使用 TLS 鑑別，則可以設定如下所示的 CHLAUTH 規則:

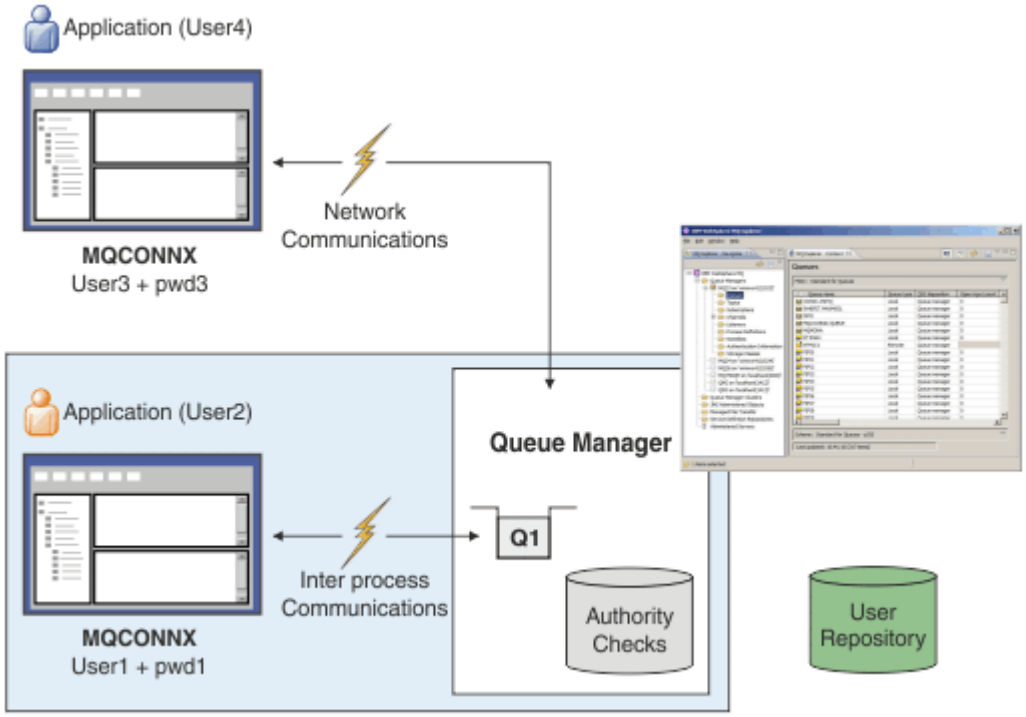
```
SET CHLAUTH(admin-channel-name)      TYPE(SSLPEERMAP)
SSLPEER('CN=Alice') ADDRESS('1.2.3.4')      USERSRC(MAP) MCAUSER('alice')
DESCR('Admin Channel - TLS authentication'
```

現在，當使用者連接至 admin-channel-name (並符合 CHLAUTH 規則) 時，他們能夠在佇列管理程式上的使用者 ID alice 下發出指令，因此不需要特許遠端存取。

連線鑑別

連線鑑別可以透過各種方式來達成:

- 應用程式可以提供使用者 ID 和密碼。 應用程式可以是用戶端，也可以使用本端連結。
- 佇列管理程式可以配置成處理所提供的使用者 ID 和密碼。
- 儲存庫可用來判斷使用者 ID 和密碼組合是否有效。



在圖表中，有兩個應用程式與佇列管理程式建立連線，一個應用程式作為用戶端，另一個使用本端連結。 應用程式可以使用各種 API 來連接至佇列管理程式，但都可以提供使用者 ID 和密碼。 在圖表中執行應用程式的使用者 ID User2 和 User4 (這是呈現給 IBM MQ 的一般作業系統使用者 ID) 可能與應用程式 User1 和 User3 所提供的使用者 ID 不同。

佇列管理程式會接收配置指令 (在圖表中，正在使用 IBM MQ Explorer)，並管理資源的開啟，以及檢查存取那些資源的權限。 IBM MQ 中有許多應用程式可能需要權限才能存取的不同資源。 此圖說明開啟佇列以進行輸出，但相同的原則也適用於其他資源。

如需用來檢查使用者 ID 和密碼之儲存庫的詳細資料，請參閱 使用者儲存庫 。

相關概念

第 58 頁的『連線鑑別: 配置』
佇列管理程式可以配置成使用提供的使用者 ID 和密碼，來檢查使用者是否有權存取資源。

第 62 頁的『連線鑑別: 應用程式變更』

第 62 頁的『連線鑑別: 使用者儲存庫』
對於每一個佇列管理程式，您可以選擇不同類型的鑑別資訊物件來鑑別使用者 ID 和密碼。

連線鑑別: 配置

佇列管理程式可以配置成使用提供的使用者 ID 和密碼，來檢查使用者是否有權存取資源。

在佇列管理程式上開啟連線鑑別

在佇列管理程式物件上， **CONNAUTH** 屬性可以設為鑑別資訊 (AUTHINFO) 物件的名稱。 此物件可以是兩種類型 (AUTHTYPE 屬性) 之一:

IDPWOS
指出佇列管理程式使用本端作業系統來鑑別使用者 ID 及密碼。

IDPWLDAP
指出佇列管理程式使用 LDAP 伺服器來鑑別使用者 ID 及密碼。

註: 您無法在 **CONNAUTH** 欄位中使用任何其他類型的鑑別資訊物件。

IDPWOS 和 IDPWLDAP 在它們的一些屬性中是類似的，如這裡所說明。 稍後會考量其他屬性。

若要檢查本端連線，請使用 AUTHINFO 屬性 **CHCKLOCL** (檢查本端連線)。 若要檢查用戶端連線，請使用 AUTHINFO 屬性 **CHCKCLNT** (檢查用戶端連線)。 必須先重新整理配置，然後佇列管理程式才能辨識變更。

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

其中 CONNAUTH 中的 USE.PW 是符合 AUTHINFO 定義的字串。

CHCKLOCL 接受 NONE 及 OPTIONAL 的值，且 **CHCKCLNT** 容許針對要配置的鑑別需求，使用 NONE 值:

NONE
關閉檢查。

選用項目
確保如果應用程式提供使用者 ID 和密碼，則它們是有效的配對，但不一定要提供它們。 例如，在移轉期間，此選項可能很有用。

如果您:

- 提供已鑑別的使用者名稱及密碼。
- 不提供使用者名稱和密碼，容許連線。
- 請提供使用者名稱，但不要提供您收到錯誤的密碼。

重要: OPTIONAL 是您可以設定的最小值，以便使用更嚴格的 CHLAUTH 規則。

如果您選取 NONE ，且用戶端連線符合具有 CHCKCLNT REQUIRED (或 z/OS 以外平台上的 REQDADM) 的 CHLAUTH 記錄，則連線會失敗。 您會在 z/OS 以外的平台上收到 AMQ9793 訊息，在 z/OS 上收到 CSQX793E 訊息。

必要
需要所有應用程式提供有效的使用者 ID 和密碼。 另請參閱下列附註。

REQDADM
特許使用者必須提供有效的使用者 ID 及密碼，但非特許使用者會被視為使用 OPTIONAL 設定。 另請參閱下列附註。 z/OS (在 z/OS 系統上不容許此設定。)

註:

將 **CHCKLOCL** 設為 REQUIRED 或 REQDADM 表示您無法使用 **runmqsc** (錯誤 AMQ8135: 未獲授權) 在本端管理佇列管理程式，除非使用者在 **runmqsc** 指令行上指定 -u UserId 參數。 設定之後， **runmqsc** 會在主控台提示輸入使用者的密碼。

同樣地，在本端系統上執行「IBM MQ 探險家」的使用者在嘗試連接至佇列管理程式時，會看到錯誤 AMQ4036 。 如果要指定使用者名稱和密碼，請用滑鼠右鍵按一下本端佇列管理程式物件，然後選取 **連線詳細資料 > 內容 ...** 功能表。 在 **使用者 ID** 區段中，輸入要使用的使用者名稱和密碼，然後按一下 **確定**。

類似的考量適用於與 **CHCKCLNT** 的遠端連線。

對於已移轉的佇列管理程式， **CONNAUTH** 為空白，但設為 *SYSTEM.DEFAULT.AUTHINFO.IDPWOS* ，適用於新的佇列管理程式。 依預設，前述 **AUTHINFO** 定義將 **CHCKCLNT** 設為 *REQDADM* 。

因此，您需要為使用特許使用者 ID 來連接的任何現有用戶端提供正確的作業系統密碼。

警告: 在某些情況下，用戶端應用程式的 MQCSP 結構中的密碼將透過網路以純文字傳送。 若要確保用戶端應用程式密碼受到適當保護，請參閱 第 25 頁的『MQCSP 密碼保護』。

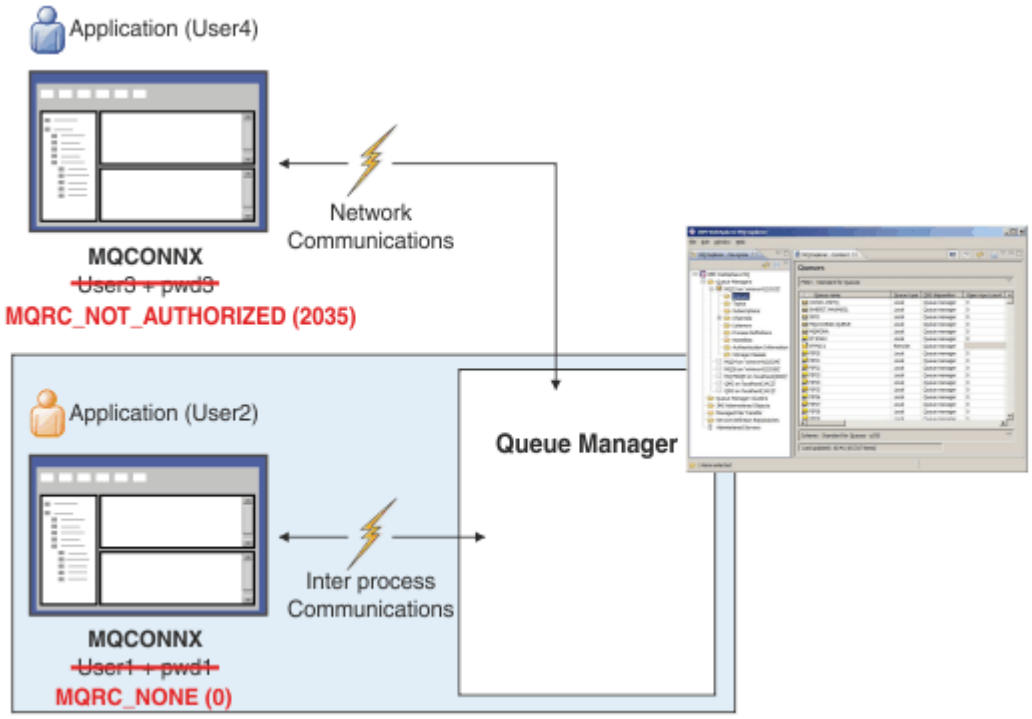
配置精度

除了用來開啟使用者 ID 及密碼檢查的 **CHCKLOCL** 及 **CHCKCLNT** 之外，還提供 CHLAUTH 規則的加強功能，以便使用 **CHCKCLNT** 進行更具體的配置。

例如，您可以將整體 **CHCKCLNT** 值設為 OPTIONAL，然後在 CHLAUTH 規則上將 **CHCKCLNT** 設為 REQUIRED 或 REQDADM ， 以針對特定通道將它升級至更嚴格。 依預設， CHLAUTH 規則將以 CHCKCLNT(ASQMGR) 執行，因此不需要使用此精度。 例如:

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

錯誤通知



如果應用程式在需要時未提供使用者 ID 和密碼，或提供不正確的組合 (即使它是選用的) ，則會記錄錯誤。

註: 當關閉密碼檢查時，如果在 **CHCKLOCL** 或 **CHCKCLNT** 上使用 NONE 選項，則不會偵測到無效密碼。

在將錯誤傳回至應用程式之前，會保留失敗鑑別達 **FAILDLAY** 屬性指定的秒數。 這可針對反覆嘗試連接的應用程式提供一些保護。

以多種方式記錄錯誤:

應用程式
應用程式會傳回標準 IBM MQ 安全錯誤 RC2035 -MQRC_NOT_AUTHORIZED。

管理者

IBM MQ 管理者會看到錯誤日誌中所報告的事件，因此可以看到應用程式遭到拒絕，因為使用者 ID 和密碼檢查失敗，而不是因為 (例如) 沒有連線權限 。

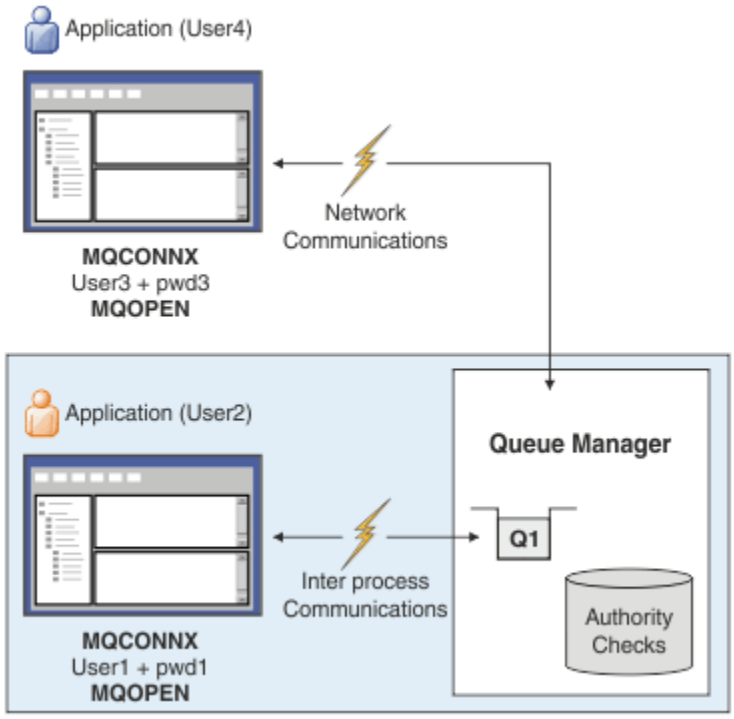
監視工具

如果您透過將事件訊息傳送至 SYSTEM.ADMIN.QMGR.EVENT 佇列:

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

此「未獲授權」事件是「類型 1」連接事件，並提供與其他「類型 1」事件相同的欄位，以及其他欄位，即所提供的 MQCSP 使用者 ID。 事件訊息中未提供密碼。 這表示事件訊息中有兩個使用者 ID: 用來執行應用程式的 ID ， 以及應用程式針對使用者 ID 和密碼檢查所呈現的 ID。

與授權的關係



您可以配置佇列管理程式，以強制特定應用程式提供使用者 ID 及密碼，因為執行應用程式的使用者 ID 可能不是應用程式所呈現的相同使用者 ID ， 以及應用程式開啟輸出佇列時的密碼，例如:

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

如何處理使用者 ID 和密碼是由鑑別資訊物件上的 **ADOPTCTX** 屬性所控制。

ADOPTCTX (YES)

應用程式的所有授權檢查都是使用您透過密碼所鑑別的相同使用者 ID 來進行，方法是選取在連線的剩餘生命期限內採用環境定義作為應用程式環境定義。

小心: 使用 ADOPTCTX (YES) 及 OS 使用者 ID 時，您必須確定採用的使用者 ID 不會超出使用者 ID 的長度上限。 如需相關資訊，請參閱 第 72 頁的『使用者 ID』。

ADOPTCTX (NO)

應用程式會提供使用者 ID 和密碼，以便在連線時鑑別它們，但之後會繼續使用應用程式在其下執行的使用者 ID 來進行未來授權檢查。 在移轉時，或您計劃使用其他機制 (例如通道鑑別記錄) 來指派 訊息通道代理程式使用者 ID (MCAUSER)時，您可能會發現這個選項很有用。

小心:

當您在鑑別資訊物件上使用 **ADOPTCTX(YES)** 參數時，除非您在 qm.ini 檔的通道段落中設定 **ChlauthEarlyAdopt** 參數，否則無法採用另一個安全環境定義。

例如，預設鑑別資訊物件設為 **ADOPTCTX(YES)**，且使用者 fred 已登入。 已配置下列兩個 CHLAUTH 規則:

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

發出下列指令，目的是將指令鑑別為使用者 bob 採用的安全環境定義:

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

事實上，佇列管理程式會使用 fred 而非 bob 的安全環境定義，且連線會失敗。

如需 **ChlauthEarlyAdopt** 的相關資訊，請參閱 通道段落的屬性。

相關概念

第 58 頁的『連線鑑別』

第 62 頁的『連線鑑別: 應用程式變更』

第 62 頁的『連線鑑別: 使用者儲存庫』

對於每一個佇列管理程式，您可以選擇不同類型的鑑別資訊物件來鑑別使用者 ID 和密碼。

連線鑑別: 應用程式變更

當呼叫 MQCONNX 時，應用程式可以在連線安全參數 (MQCSP) 結構內提供使用者 ID 和密碼。 使用者 ID 和密碼會傳遞給隨佇列管理程式一起提供的 物件權限管理程式 (OAM)，或隨 z/OS 系統上的佇列管理程式一起提供的授權服務元件。 您不需要撰寫自己的自訂介面。

如果應用程式以用戶端身分執行，則使用者 ID 及密碼也會傳遞至用戶端及伺服器端安全結束程式進行處理。 它們也可以用來設定通道實例的 訊息通道代理程式使用者 ID (MCAUSER) 屬性 。 針對此處理程序，會以結束原因 MQXR_SEC_PARMS 來呼叫安全結束程式。 用戶端安全結束程式及預先連接結束程式在傳送至佇列管理程式之前，可以對 MQCONN 進行變更。

警告: 在某些情況下，用戶端應用程式的 MQCSP 結構中的密碼將透過網路以純文字傳送。 若要確保用戶端應用程式密碼受到適當保護，請參閱 第 25 頁的『MQCSP 密碼保護』。

透過使用 XAOPEN 字串來提供使用者 ID 和密碼，您可以避免必須對應用程式碼進行變更。

註:

從 IBM WebSphere MQ 6.0 開始，安全結束程式已容許設定 MQCSP。 因此，此層次或更新版本的用戶端不需要升級。

不過，在 IBM MQ 8.0 之前的 IBM MQ 版本中， MQCSP 對應用程式提供的使用者 ID 和密碼沒有任何限制。 將這些值與 IBM MQ 提供的特性搭配使用時，會有一些限制適用於這些特性的使用，但如果您只將它們傳遞至自己的結束程式，則那些限制不適用。

相關概念

第 58 頁的『連線鑑別』

第 58 頁的『連線鑑別: 配置』

佇列管理程式可以配置成使用提供的使用者 ID 和密碼，來檢查使用者是否有權存取資源。

第 62 頁的『連線鑑別: 使用者儲存庫』

對於每一個佇列管理程式，您可以選擇不同類型的鑑別資訊物件來鑑別使用者 ID 和密碼。

連線鑑別: 使用者儲存庫

對於每一個佇列管理程式，您可以選擇不同類型的鑑別資訊物件來鑑別使用者 ID 和密碼。

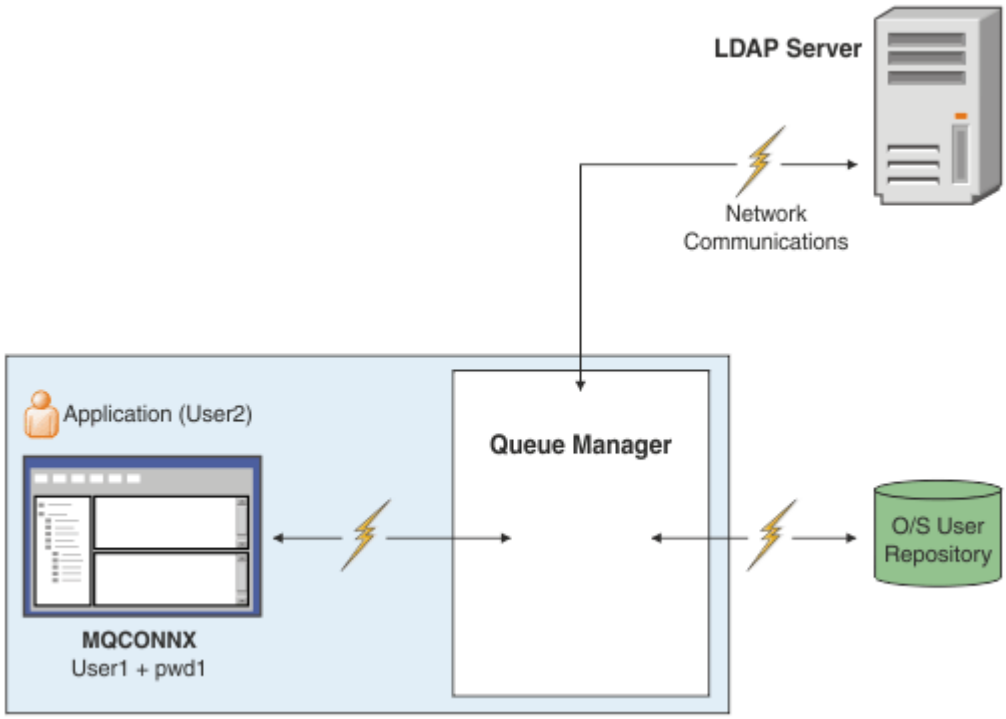


圖 7: 鑑別資訊物件的類型

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

鑑別資訊物件有兩種類型，如圖表中所示:

- IDPWOS 用來指出佇列管理程式使用本端作業系統來鑑別使用者 ID 及密碼。 如果您選擇使用本端作業系統，則需要設定共同屬性，如前述主題中所述。
- IDPWLDAP 用來指出佇列管理程式使用 LDAP 伺服器來鑑別使用者 ID 及密碼。 如果您選擇使用 LDAP 伺服器，本主題會提供更多資訊。

透過在佇列管理程式的 **CONNAUTH** 屬性中命名適當的物件，只能為每一個要使用的佇列管理程式選擇一種類型的鑑別資訊物件。

使用 LDAP 伺服器進行鑑別。

將 **CONNAME** 欄位設為佇列管理程式的 LDAP 伺服器位址。 您可以在以逗點區隔的清單中提供 LDAP 伺服器的其他位址，如果 LDAP 伺服器本身不提供此機能，則有助於備援。

在 **LDAPUSER** 及 **LDAPPWD** 欄位中設定必要的 LDAP 伺服器 ID 及密碼，讓佇列管理程式可以存取 LDAP 伺服器，並查閱使用者記錄的相關資訊。

LDAP 伺服器的安全連線

與通道不同，沒有 **SSLCIPH** 參數可開啟使用 TLS 與 LDAP 伺服器進行通訊。 在此情況下， IBM MQ 充當 LDAP 伺服器的用戶端，因此大部分配置都在 LDAP 伺服器上完成。 IBM MQ 中的部分現有參數用來配置該連線的運作方式。

設定 **SECCOMM** 欄位，以控制 LDAP 伺服器的連線功能是否使用 TLS。

除了此屬性之外，佇列管理程式屬性 **SSLFIPS** 及 **SUITEB** 還會限制所選擇的密碼規格集。 用來向 LDAP 伺服器識別佇列管理程式的憑證是佇列管理程式憑證 (`ibmwebspheremq` *`qmgr-name`* 或 **CERTLABL** 屬性的值)。 如需詳細資料，請參閱 數位憑證標籤 。

LDAP 使用者儲存庫

使用 LDAP 使用者儲存庫時，除了告訴佇列管理程式在何處尋找 LDAP 伺服器之外，還有一些要在佇列管理程式上執行的配置。

LDAP 伺服器中定義的使用者 ID 具有可唯一識別它們的階層式結構。 因此，應用程式可以連接至佇列管理程式，並以完整的階層式使用者 ID 來呈現其使用者 ID。

不過，為了簡化應用程式必須提供的資訊，可以將佇列管理程式配置成假設階層的第一部分是所有 ID 的共同部分，並在應用程式提供的縮短 ID 之前自動新增此部分。 然後，佇列管理程式可以向 LDAP 伺服器提供完整 ID。

將 BASEDNU 設為 LDAP 搜尋在 LDAP 階層中尋找 ID 的起始點。 當您設定 BASEDNU 時，必須確保在 LDAP 階層中搜尋 ID 時只會傳回一個結果。

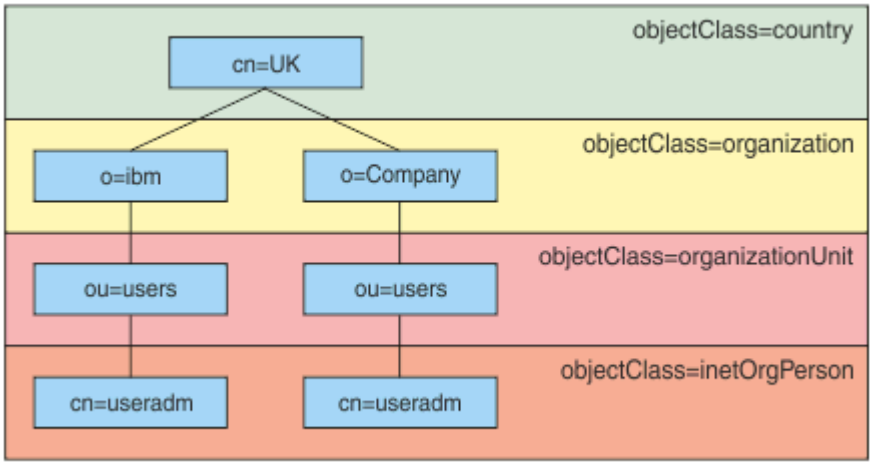


圖 *8: LDAP* 階層範例

例如，在 第 64 頁的圖 8 BASEDNU 中，可以設為 "ou=users， o=ibm， c = UK" 或 "， o=ibm， c = UK"。 不過，因為同時在 "o = ibm" 分支和 "o=Company" 分支中存在包含 "cn = useradm" 的識別名稱，所以 BASEDNU 無法設為 "c = UK"。 基於效能和安全理由，請使用 LDAP 階層中的最高點，您可以從中參照您需要的所有使用者 ID。 在此範例中，即 "ou=users， o=ibm， c = UK"。

例如，您的應用程式可能會向佇列管理程式提交使用者 ID，而不提供 LDAP 屬性名稱 CN= 。 如果您將 USRFIELD 設為 LDAP 屬性名稱，則會將此值作為字首新增至來自應用程式的使用者 ID。 當您從作業系統使用者 ID 移至 LDAP 使用者 ID 時，這可能是有用的移轉輔助工具，因為在這兩種情況下，應用程式可以呈現相同的字串，且您可以避免變更應用程式。

因此，呈現給 LDAP 伺服器的完整使用者 ID 看起來如下:

```
USRFIELD = ID_from_application BASEDNU
```

相關概念

第 58 頁的『連線鑑別』

第 58 頁的『連線鑑別: 配置』
佇列管理程式可以配置成使用提供的使用者 ID 和密碼，來檢查使用者是否有權存取資源。

第 62 頁的『連線鑑別: 應用程式變更』

用來插入使用者 *ID* 和密碼的用戶端安全結束程式 (*mqccred*)

如果您有任何用戶端應用程式需要傳送使用者 ID 或密碼，但您還無法變更來源，則您可以使用 IBM MQ 8.0 **mqccred** 隨附的安全結束程式。 **mqccred** 代表用戶端應用程式從 `.ini` 檔案提供使用者 ID 和密碼。 此使用者 ID 及密碼會傳送至佇列管理程式，如果配置為這樣做，則會對它們進行鑑別。

概觀

mqccred 是在與用戶端應用程式相同的機器上執行的安全結束程式。 它容許代表用戶端應用程式提供使用者 ID 及密碼資訊，其中應用程式本身未提供該資訊。 使用者 ID 和密碼資訊以稱為 連線安全參數 (MQCSP) 的結構提供，如果配置 連線鑑別 ，則由佇列管理程式進行鑑別。

使用者 ID 和密碼資訊擷取自用戶端機器上的 `.ini` 檔案。 檔案中的密碼受到 **runmqccred** 指令的模糊化保護，並確保 `.ini` 檔案上的檔案許可權已設定成只有執行用戶端應用程式 (因此結束程式) 的使用者 ID 能夠讀取它。

位置

已安裝 **mqccred** :

Windows 平台
在 *installation_directory*\Tools\c\Samples\mqccred\ 目錄中

AIX and Linux 平台
在 *installation_directory*/samp/mqccred 目錄中

附註: 結束程式:

1. 純粹作為安全通道結束程式，且必須是在通道上定義的唯一此類結束程式。
2. 通常是透過「用戶端通道定義表 (CCDT)」來命名，但 Java 用戶端可以直接在 JNDI 物件中提及結束程式，或為手動建構 MQCD 結構的應用程式配置結束程式。
3. 您必須將 **mqccred** 和 **mqccred_r** 程式複製到 `var/mqm/exits` 目錄。

   例如，在 64 位元 AIX 或 Linux 系統上，發出下列指令:

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

   如需相關資訊，請參閱 如何測試 mqccred 的逐步範例 。
4. 能夠在舊版 IBM MQ(遠至 IBM WebSphere MQ 7.0.1) 上執行。

設定使用者 ID 和密碼

`.ini` 檔案包含每一個佇列管理程式的段落，以及未指定佇列管理程式的廣域設定。 每一個段落都包含佇列管理程式的名稱、使用者 ID ，以及純文字或模糊化密碼。

您必須手動編輯 `.ini` 檔案 (使用您想要的任何編輯器) ，並將純文字密碼屬性新增至段落。 執行所提供的 **runmqccred** 程式，它會取得 `.ini` 檔，並將 **Password** 屬性取代為 **OPW** 屬性 (密碼的模糊化形式)。

如需指令及其參數的說明，請參閱 runmqccred 。

`mqccred.ini` 檔案包含您的使用者 ID 和密碼資訊。

在與結束程式相同的目錄中提供範本 `.ini` 檔案，以提供企業的起始點。

依預設，會在 `$HOME/.mqs/mqccred.ini` 中尋找此檔案。 如果您想要在其他位置找到它，您可以使用環境變數 *MQCCRED* 來指向它:

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

如果您使用 *MQCCRED*，則變數必須包括配置檔的完整名稱，包括任何 `.ini` 檔案類型。 因為此檔案包含密碼 (即使模糊化也一樣) ，所以您應該使用作業系統專用權來保護檔案，以確保未獲授權的人員無法讀取它。 如果您沒有正確的檔案許可權，結束程式將無法順利執行。

如果應用程式已提供 MQCSP 結構，結束程式通常會遵循這一點，且不會從 `.ini` 檔插入任何資訊。 不過，您可以在段落中使用 **Force** 屬性來置換此情況。

將 **Force** 設為值 *TRUE* 會移除應用程式提供的使用者 ID 及密碼，並將這些使用者 ID 及密碼取代為 ini 檔案版本。

您也可以在檔案的廣域區段中設定 **Force** 屬性，以設定該檔案的預設值。

Force 的預設值為 *FALSE*。

您可以為所有佇列管理程式或每個個別佇列管理程式提供使用者 ID 和密碼。 這是 `mqccred.ini` 檔的範例:

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

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

Force=TRUE

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

附註:

1. 個別佇列管理程式定義優先於廣域設定。
2. 屬性不區分大小寫。

限制

當使用這個結束程式時，執行應用程式之人員的本端使用者 ID 不會從用戶端流向伺服器。 唯一可用的身分資訊是來自 ini 檔案內容。

因此，您必須將佇列管理程式配置成使用 **ADOPTCTX(YES)**，或透過其中一個可用的機制 (例如 第 40 頁的『通道鑑別記錄』)，將入埠連線要求對映至適當的使用者 ID。

重要: 如果您新增密碼或更新舊密碼， **runmqccred** 指令只會處理任何純文字密碼，而不會改變模糊化的密碼。

除錯

當啟用時，結束程式會寫入標準 IBM MQ 追蹤。

為了協助除錯配置問題，結束程式也可以直接寫入 stdout。

無通道安全結束程式資料 (**SCYDATA**) 通道通常需要配置。 不過，您可以指定:

錯誤
僅列印符合錯誤狀況的資訊，例如找不到配置檔。

除錯
顯示這些錯誤狀況，以及一些其他追蹤陳述式。

NOCHECKS
略過檔案許可權的限制，以及 `.ini` 檔案不應包含任何未受保護密碼的進一步限制。

您可以在 **SCYDATA** 欄位中以任何順序放置一個以上這些元素 (以逗點區隔)。 例如，`SCYDATA=(NOCHECKS,DEBUG)`。

請注意，項目區分大小寫，且必須以大寫輸入。

使用 mqccred

設定檔案之後，您可以更新用戶端連線通道定義來包含 `SCYEXIT('mqccred(ChlExit)')` 屬性，以呼叫通道結束程式:

```
DEFINE CHANNEL(channelname) CHLTYPE(clntconn) +
CONNAME(remote machine) +
QMNAME(remote qmgr) +
```

```
SCYEXIT('mqccred(ChlExit)') +
REPLACE
```

相關參考

SCYDATA

SCYEXIT

runmqccred

與 *Java* 用戶端的連線鑑別

連線鑑別是 IBM MQ 中的一項特性，可讓您配置佇列管理程式，以便佇列管理程式可以使用提供的使用者 ID 和密碼來鑑別應用程式。 當應用程式是使用用戶端傳輸的 Java 應用程式時，可以在相容模式或 MQCSP 鑑別模式下執行連線鑑別。

應用程式使用下列其中一種方法來指定要鑑別的使用者 ID 和密碼:

- 在 IBM MQ classes for Java 應用程式中，在 `MQEnvironment` 類別中，或傳遞至 `com.ibm.mq.MQQueueManager` 建構子的內容 Hashtable 中。
- 在 IBM MQ classes for JMS 應用程式中，作為 `createConnection(String username, String Password)` 或 `createContext(String username, String password)` 方法的引數。

MQCSP 鑑別模式

在此模式中，執行應用程式的用戶端使用者 ID 會傳送至佇列管理程式，以及要鑑別的使用者 ID 和密碼。 IBM MQ classes for Java 和 IBM MQ classes for JMS 會將要鑑別的使用者 ID 和密碼傳送至 MQCSP 結構中的佇列管理程式。

使用者 ID 和密碼可供 MQCSP 結構內的伺服器連線安全結束程式使用。 MQCSP 結構位址可以在通道之 MQCXP 結構的 **SecurityParms** 欄位中找到。

MQCSP 鑑別模式具有下列好處:

- 要鑑別的使用者 ID 長度上限為 1024 個字元。
- 用於鑑別的密碼長度上限為 256 個字元。
- 當使用 ADOPTCTX (NO) 配置用來控制佇列管理程式上的連線鑑別的鑑別資訊物件時，可以使用應用程式執行所在的用戶端使用者 ID 來執行使用 IBM MQ 資源的存取權授權檢查。

相容模式

在 IBM MQ 8.0 之前， Java 用戶端可以透過用戶端連線通道將使用者 ID 及密碼傳送至伺服器連線通道，並將它們提供給 MQCD 結構的 **RemoteUserIdentifier** 及 **RemotePassword** 欄位中的安全結束程式。 在相容模式中，會保留此行為。

您可以將此模式與連線鑑別一起使用，並從先前用來執行相同工作的任何安全結束程式移轉出去。

此模式具有下列限制:

- 使用者 ID 和密碼的長度必須等於或小於 12 個字元。 長度超過 12 個字元的使用者 ID 會截斷為 12 個字元。 這可能會導致連線失敗，原因碼為 MQRC_NOT_AUTHORIZED。
- 用來執行應用程式的用戶端使用者 ID 不會傳送至佇列管理程式。 您必須在鑑別資訊物件上設定 ADOPTCTX (YES) ，以用來控制佇列管理程式上的連線鑑別，或使用其他方法 (例如基於 TLS 憑證的通道鑑別規則) 來設定通道 MCA 使用者 ID ，以檢查是否有權使用 IBM MQ 資源。

預設鑑別模式

IBM MQ classes for Java 或 IBM MQ classes for JMS 用戶端應用程式使用的預設鑑別模式視應用程式是否指定使用者 ID 及密碼而定。

- V 9.2.1 從 IBM MQ 9.2.1 開始，如果指定使用者 ID 和密碼，依預設會使用 MQCSP 鑑別。
- 在早於 IBM MQ 9.2.1 的版本中，如果指定使用者 ID 及密碼，則預設模式如下:
  - 依預設， MQCSP 鑑別由使用 IBM MQ classes for Java 的應用程式使用。

  - 依預設，相容模式由使用 IBM MQ classes for JMS 的應用程式使用。
- 如果使用者 ID 未指定密碼，依預設會使用相容模式。
- 如果未指定使用者 ID，則一律使用相容模式。

在指定使用者 ID 的情況下，應用程式可以針對每一個個別連線選擇特定的鑑別模式，或在啟動應用程式之前廣域設定鑑別模式，如 第 68 頁的『選擇鑑別模式』中所述。

註: V 9.2.1 使用 IBM MQ classes for JMS 的應用程式可能會受到 IBM MQ 9.2.1 中預設鑑別模式變更的影響。 將 IBM MQ classes for JMS 升級至 IBM MQ 9.2.1 之後，先前依預設使用相容模式的應用程式將改用 MQCSP 鑑別。 這可能會導致先前順利連接至佇列管理程式的應用程式無法連接包含原因碼 2035 (MQRC_NOT_AUTHORIZED) 的 `JMSException`。 如果發生這種情況，請使用 第 68 頁的『選擇鑑別模式』中說明的其中一種方法來指定應用程式使用相容模式。

使用本端連結連接至佇列管理程式的 Java 應用程式一律使用 MQCSP 鑑別模式。

選擇鑑別模式

使用下列其中一種方法，可以指定 Java 用戶端應用程式在連接至佇列管理程式時指定使用者 ID 所使用的鑑別模式。 這些方法以遞減優先順序列出。 如果未使用任何這些方法來指定鑑別模式，則會使用預設鑑別模式。

註: V 9.2.1 在 IBM MQ 9.2.1 中已明確使用這些方法來選取鑑別模式。 在某些情況下，當 IBM MQ classes for Java 或 IBM MQ classes for JMS 升級至 IBM MQ 9.2.1 時， Java 用戶端應用程式所使用的鑑別模式可能會變更。 這可能會導致先前順利連接至佇列管理程式的應用程式無法連接包含原因碼 2035 (MQRC_NOT_AUTHORIZED) 的 `JMSException`。 如果發生這種情況，請使用下列其中一種方法來選取所需的鑑別模式。

- 在連接至佇列管理程式之前，在應用程式中設定適當的內容，以指定每一個個別連線的鑑別模式。
  - 使用 IBM MQ classes for Java 時，請在傳遞至 `com.ibm.mq.MQQueueManager` 建構子的內容 Hashtable 中設定內容 *MQConstants.USE_MQCSP_AUTHENTICATION_PROPERTY*。
  - 使用 IBM MQ classes for JMS 時，請設定 *JmsConstants* 內容。在建立連線之前，請先適當 *Connection Factory* 上的 *USER_AUTHENTICATION_MQCSP*。

  將這些內容的值設為下列其中一個值:

  true
  向佇列管理程式進行鑑別時，請使用 MQCSP 鑑別模式。

  false
  向佇列管理程式進行鑑別時使用相容模式。
- 透過在啟動應用程式時設定 *com.ibm.mq.cfg.jmqi.useMQCSPauthentication* Java 系統內容，為應用程式所建立的所有用戶端連線指定鑑別模式。 將內容的值設為下列其中一個值:

  Y
  向佇列管理程式進行鑑別時，請使用 MQCSP 鑑別模式。

  N
  向佇列管理程式進行鑑別時使用相容模式。

  例如，下列指令會將內容設為選取相容模式，並啟動 Java 應用程式:

  ```
  java -Dcom.ibm.mq.cfg.jmqi.useMQCSPauthentication=N application_name
  ```
- 透過在啟動應用程式的環境中設定 *com.ibm.mq.jmqi.useMQCSPauthentication* 環境變數，指定在相同環境中啟動之應用程式所建立的所有用戶端連線的鑑別模式。 將環境變數的值設為下列其中一個值:

  Y
  向佇列管理程式進行鑑別時，請使用 MQCSP 鑑別模式。

  N
  向佇列管理程式進行鑑別時使用相容模式。

- 透過在用戶端配置檔的 JMQI 段落中指定 **useMQCSPauthentication** 屬性，為使用特定 IBM MQ MQI client 用戶端配置檔的所有應用程式指定鑑別模式。 將屬性值設為下列其中一個值:

  YES
  向佇列管理程式進行鑑別時，請使用 MQCSP 鑑別模式。

  NO
  向佇列管理程式進行鑑別時使用相容模式。

  如需 **useMQCSPauthentication** 屬性的相關資訊，請參閱 用戶端配置檔的 JMQI 段落。

在 IBM MQ Explorer 中選擇鑑別模式

IBM MQ Explorer 是 Java 應用程式，因此這兩種模式 (相容模式及 MQCSP 鑑別模式) 也適用於它。

從 IBM MQ 9.1.0 開始， MQCSP 鑑別模式是預設值。 在 IBM MQ 9.1 之前，相容模式是預設值。

在提供使用者識別的畫面上，有一個勾選框可啟用或停用相容模式:

- 從 IBM MQ 9.1.0 開始，依預設不會選取這個勾選框。 若要使用相容模式，請選取此勾選框。
- 在 IBM MQ 9.1.0 之前，依預設會啟用這個勾選框。 若要使用 MQCSP 鑑別，請清除勾選框。

相關概念
第 58 頁的『連線鑑別』
第 62 頁的『連線鑑別: 應用程式變更』
第 62 頁的『連線鑑別: 使用者儲存庫』
對於每一個佇列管理程式，您可以選擇不同類型的鑑別資訊物件來鑑別使用者 ID 和密碼。

IBM MQ 中的訊息安全

IBM MQ 基礎架構中的訊息安全由 Advanced Message Security 提供。

Advanced Message Security (AMS) 擴充 IBM MQ 安全服務，以提供訊息層次的資料簽署及加密。 展開的服務可保證訊息資料在最初放置在佇列上與擷取時之間未修改。 此外， AMS 還會驗證訊息資料的傳送端是否已獲授權將已簽署的訊息放置在目標佇列上。

相關概念
第 498 頁的『Advanced Message Security』
Advanced Message Security (AMS) 是 IBM MQ 的元件，可為流經 IBM MQ 網路的機密資料提供高階保護，同時不會影響一般應用程式。

規劃安全需求

此主題集合說明在 IBM MQ 環境中規劃安全時需要考量的事項。

您可以將 IBM MQ 用於各種平台上的各種應用程式。 每一個應用程式的安全需求可能不同。 對某些人來說，安全將是重要考量。

IBM MQ 提供一系列鏈結層次安全服務，包括支援「傳輸層安全 (TLS)」。

規劃安裝 IBM MQ 時，您必須考量安全的某些層面:

- Multi 在 多平台上，如果您忽略這些層面而不執行任何動作，則無法使用 IBM MQ。
- z/OS 在 z/OS 上，忽略這些層面的效果是您的 IBM MQ 資源未受保護。 亦即，所有使用者都可以存取及變更所有 IBM MQ 資源。

管理 IBM MQ 的權限

IBM MQ 管理者需要下列權限:

- 發出指令以管理 IBM MQ
- 使用 IBM MQ Explorer

- IBM i 使用 IBM i 管理畫面和指令。
- z/OS 使用 z/OS 上的作業及控制面板
- z/OS 在 z/OS 上使用 IBM MQ 公用程式 CSQUTIL
- z/OS 在 z/OS 上存取佇列管理程式資料集

如需相關資訊，請參閱：

- ALW 第 334 頁的『在 AIX, Linux, and Windows 上管理 IBM MQ 的權限』
- IBM i 第 73 頁的『在 IBM i 上管理 IBM MQ 的權限』
- z/OS 第 74 頁的『在 z/OS 上管理 IBM MQ 的權限』

使用 IBM MQ 物件的權限

應用程式可以透過發出 MQI 呼叫來存取下列 IBM MQ 物件：

- 佇列管理程式
- 佇列
- Processes
- 名單
- 主題

應用程式也可以使用「可程式化指令格式 (PCF)」指令來存取這些 IBM MQ 物件，以及存取通道和鑑別資訊物件。 這些物件可以由 IBM MQ 保護，因此與應用程式相關聯的使用者 ID 需要權限才能存取它們。

如需相關資訊，請參閱第 75 頁的『授權應用程式使用 IBM MQ』。

通道安全性

與訊息通道代理程式 (MCA) 相關聯的使用者 ID 需要存取各種 IBM MQ 資源的權限。 例如， MCA 必須能夠連接至佇列管理程式。 如果它是傳送端 MCA，則必須能夠開啟通道的傳輸佇列。 如果它是接收 MCA，則必須能夠開啟目的地佇列。 與需要管理通道、通道起始程式及接聽器的應用程式相關聯的使用者 ID，需要使用相關 PCF 指令的權限。 不過，大部分應用程式都不需要這類存取權。

如需相關資訊，請參閱第 92 頁的『通道授權』。

其他考量

只有在使用特定 IBM MQ 功能或基本產品延伸時，才需要考量下列安全層面：

- 第 102 頁的『佇列管理程式叢集的安全』
- 第 102 頁的『IBM MQ 發佈/訂閱的安全』
- 第 103 頁的『的安全 IBM MQ Internet Pass-Thru』

規劃識別及鑑別

決定要使用的使用者 ID，以及您要套用鑑別控制項的方式和層次。

您必須決定如何識別 IBM MQ 應用程式的使用者，請記住，不同的作業系統支援不同長度的使用者 ID。 您可以使用通道鑑別記錄，從一個使用者 ID 對映至另一個使用者 ID，或根據連線的某個屬性來指定使用者 ID。 使用 TLS 的 IBM MQ 通道使用數位憑證作為識別及鑑別的機制。 每一個數位憑證都有一個主體識別名稱，可使用通道鑑別記錄對映至特定身分。 此外，金鑰儲存庫中的 CA 憑證會決定哪些數位憑證可用來向 IBM MQ 進行鑑別。 如需相關資訊，請參閱：

- 第 321 頁的『將遠端佇列管理程式對映至 MCAUSER 使用者 ID』
- 第 322 頁的『將用戶端使用者 ID 對映至 MCAUSER 使用者 ID』

- 第 322 頁的『將 SSL 或 TLS 識別名稱對映至 MCAUSER 使用者 ID』
- 第 324 頁的『將 IP 位址對映至 MCAUSER 使用者 ID』

規劃用戶端應用程式的鑑別

您可以在四個層次套用鑑別控制: 通訊層次、安全結束程式、通道鑑別記錄，以及傳遞至安全結束程式的識別。

有四種安全等級需要考量。 此圖顯示已連接至伺服器的 IBM MQ MQI client 。 在四個層次上套用安全，如下列文字中所述。 MCA 是「訊息通道代理程式」。

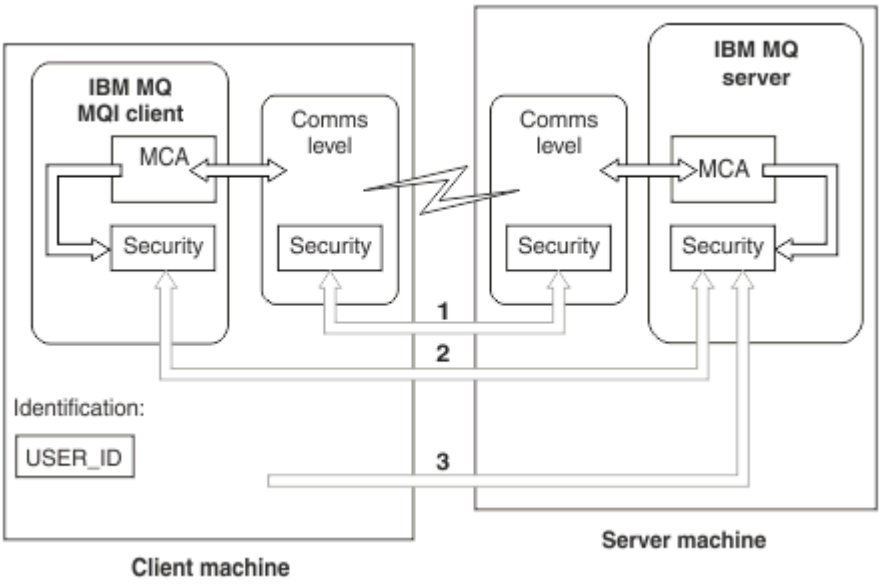


圖 9: 主從式連線中的安全

1. 通訊層次

   請參閱箭頭 1。 若要在通訊層次實作安全，請使用 TLS。 如需相關資訊，請參閱 第 13 頁的『加密安全通訊協定: TLS』

2. 通道鑑別記錄

   請參閱箭頭 2 和 3。 可以在安全層次使用 IP 位址或 TLS 識別名稱來控制鑑別。 也可以封鎖使用者 ID，或將主張的使用者 ID 對映至有效的使用者 ID。 第 40 頁的『通道鑑別記錄』中提供完整說明。

3. 連線鑑別

   請參閱箭頭 3。 用戶端會傳送 ID 和密碼。 如需相關資訊，請參閱第 58 頁的『連線鑑別: 配置』。

4. 通道安全結束程式

   請參閱箭頭 2。 用戶端至伺服器通訊的通道安全結束程式的運作方式與伺服器至伺服器通訊的運作方式相同。 可以撰寫通訊協定無關的結束程式配對，以提供用戶端及伺服器的交互鑑別。 在 通道安全結束程式中提供完整說明。

5. 傳遞至通道安全結束程式的識別

   請參閱箭頭 3。 在用戶端至伺服器通訊中，通道安全結束程式不需要成對運作。 可以省略 IBM MQ 用戶端上的結束程式。 在此情況下，使用者 ID 會放在通道描述子 (MQCD) 中，必要的話，伺服器端安全結束程式可以變更它。

   IBM MQ MQI clients 也會傳送額外資訊來協助識別。

   - 傳遞至伺服器的使用者 ID 是用戶端上目前登入的使用者 ID。
   - 目前登入使用者的安全 ID。

   伺服器安全結束程式可以使用使用者 ID 及安全 ID (如果有的話) 的值來建立 IBM MQ MQI client 的身分。

從 IBM MQ 8.0 開始，您可以傳送 MQCSP 結構中包含的密碼。

警告: 在某些情況下，用戶端應用程式的 MQCSP 結構中的密碼將透過網路以純文字傳送。 若要確保用戶端應用程式密碼受到適當保護，請參閱 第 25 頁的『MQCSP 密碼保護』。

使用者 *ID*

當您建立用戶端應用程式的使用者 ID 時，使用者 ID 不得超過允許的長度上限。 您不得使用保留使用者 ID UNKNOWN 及 NOBODY。 如果用戶端連接的伺服器是 IBM MQ for Windows 伺服器，您必須跳出使用 at 符號 @。 允許的使用者 ID 長度取決於用於伺服器的平台:

- z/OS Linux AIX 在 z/OS AIX and Linux 上，使用者 ID 的長度上限為 12 個字元。
- IBM i 在 IBM i 上，使用者 ID 的長度上限為 10 個字元。
- Windows 在 Windows 上，如果 IBM MQ MQI client 和 IBM MQ 伺服器都在 Windows 上，且伺服器有權存取定義用戶端使用者 ID 的網域，則使用者 ID 的長度上限為 20 個字元。 不過，如果 IBM MQ 伺服器不是 Windows 伺服器，則會將使用者 ID 截斷為 12 個字元。
- 如果您使用 MQCSP 結構來傳遞認證，則使用者 ID 的長度上限為 1024 個字元。 MQCSP 結構使用者 ID 無法用來規避 IBM MQ 用於授權的使用者 ID 長度上限。 如需 MQCSP 結構的相關資訊，請參閱 第 280 頁的『使用 MQCSP 結構來識別及鑑別使用者』。

在 AIX and Linux 系統上，預設值是使用使用者 ID 進行鑑別，並使用群組進行授權。 不過，您可以配置這些系統，以針對使用者 ID 進行授權。 如需相關資訊，請參閱 第 293 頁的『AIX and Linux 上的 OAM 使用者型許可權』。 Windows 系統可以同時將使用者 ID 用於鑑別和授權，並將群組用於授權。

如果您建立服務帳戶而不注意群組，並以不同方式授權所有使用者 ID，則每個使用者都可以存取每個其他使用者的資訊。

受限使用者 ID

使用者 ID UNKNOWN 和群組 nobody 對 IBM MQ 具有特殊意義。 在名為 UNKNOWN 的作業系統或名為 nobody 的群組中建立使用者 ID 可能會產生非預期的結果。

連接至 IBM MQ for Windows 伺服器時的使用者 ID

Windows

如果用戶端以包含 @ 字元 (例如， abc@d) 的使用者 ID 執行，則 IBM MQ for Windows 伺服器不支援 IBM MQ MQI client 的連線。 用戶端上 MQCONN 呼叫的回覆碼為 MQRC_NOT_AUTHORIZED。

不過，您可以使用兩個 @ 字元來指定使用者 ID，例如 abc@@d。 使用 id@domain 格式是偏好的作法，以確保在正確的網域中一致地解析使用者 ID; 因此 abc@@d@domain。

規劃授權

規劃將具有管理權限的使用者，並規劃如何授權應用程式使用者適當使用 IBM MQ 物件，包括從 IBM MQ MQI client 連接的使用者。

必須授與個人或應用程式存取權，才能使用 IBM MQ。 他們需要的存取權取決於他們所承擔的角色，以及他們需要執行的作業。 IBM MQ 中的授權可以細分為兩個主要種類:

- 執行管理作業的授權
- 授權應用程式使用 IBM MQ

兩個作業類別都由相同的元件控制，且可以授與個人執行兩個作業種類的權限。

下列主題提供您必須考量之特定授權區域的進一步相關資訊:

管理 IBM MQ 的權限

IBM MQ 管理者需要權限才能執行各種功能。 此權限在不同平台上以不同方式取得。

IBM MQ 管理者需要下列權限:

- 發出指令以管理 IBM MQ。
- Windows Linux 使用 IBM MQ Explorer。
- z/OS 使用 z/OS 上的作業及控制面板。
- z/OS 在 z/OS 上使用 IBM MQ 公用程式 CSQUTIL。
- z/OS 存取 z/OS 上的佇列管理程式資料集。

如需相關資訊，請參閱適合您作業系統的主題。

ALW 在 *AIX, Linux, and Windows* 系統上管理 *IBM MQ* 的權限

IBM MQ 管理者是 mqm 群組的成員。 此群組具有所有 IBM MQ 資源的存取權，並且可以發出 IBM MQ 控制指令。 管理者可以將特定權限授與其他使用者。

若要在 AIX, Linux, and Windows 系統上成為 IBM MQ 管理者，使用者必須是 *mqm* 群組的成員。 當您安裝 IBM MQ 時，會自動建立此群組。 若要容許使用者發出控制指令，您必須將它們新增至 mqm 群組。 這包括 AIX and Linux 上的 root 使用者。

非 mqm 群組成員的使用者可以獲授與管理專用權，但他們無法發出 IBM MQ 控制指令，且他們有權僅執行他們已獲授與存取權的指令。

此外，在 Windows 系統上， SYSTEM 及 Administrator 帳戶具有 IBM MQ 資源的完整存取權。

mqm 群組的所有成員都可以存取系統上的所有 IBM MQ 資源，包括能夠管理系統上執行的任何佇列管理程式。 只有從 mqm 群組中移除使用者，才能撤銷此存取權。 在 Windows 系統上， Administrators 群組的成員也具有所有 IBM MQ 資源的存取權。

管理者可以使用控制指令 **runmqsc** 來發出 IBM MQ Script (MQSC) 指令。 以間接模式使用 **runmqsc** 將 MQSC 指令傳送至遠端佇列管理程式時，每一個 MQSC 指令都會封裝在 Escape PCF 指令內。 管理者必須具有遠端佇列管理程式處理 MQSC 指令所需的權限。

IBM MQ Explorer 會發出 PCF 指令來執行管理作業。 管理者不需要其他權限，即可使用「 IBM MQ Explorer 」來管理本端系統上的佇列管理程式。 當使用「 IBM MQ Explorer 」來管理另一個系統上的佇列管理程式時，管理者必須具有遠端佇列管理程式處理 PCF 指令所需的權限。

如需處理 PCF 及 MQSC 指令時所執行之授權檢查的相關資訊，請參閱下列主題:

- 如需在佇列管理程式、佇列、通道、處理程序、名稱清單及鑑別資訊物件上運作的指令，請參閱 第 75 頁的『授權應用程式使用 IBM MQ』。
- 如需在通道、通道起始程式、接聽器及叢集上運作的指令，請參閱 通道安全。
- z/OS 如需 IBM MQ for z/OS 上由指令伺服器處理的 MQSC 指令，請參閱 第 74 頁的『z/OS 上的指令安全及指令資源安全』。

如需管理 IBM MQ for AIX, Linux, and Windows 系統所需之權限的相關資訊，請參閱相關資訊。

IBM i 在 *IBM i* 上管理 *IBM MQ* 的權限

若要成為 IBM i 上的 IBM MQ 管理者，您必須是 *QMQMADM* 群組的成員。 此群組具有與 AIX, Linux, and Windows 系統上 mqm 群組的內容類似的內容。 特別是當您安裝 IBM MQ for IBM i 時，會建立 QMQMADM 群組，且 QMQMADM 群組的成員可以存取系統上的所有 IBM MQ 資源。 如果您具有 *ALLOBJ 權限，則也可以存取所有 IBM MQ 資源。

管理者可以使用 CL 指令來管理 IBM MQ。其中一個指令是 GRTMQMAUT ，用來授與權限給其他使用者。 另一個指令 STRMQMMQSC 可讓管理者對本端佇列管理程式發出 MQSC 指令。

IBM MQ for IBM i 提供兩組 CL 指令:

群組 1
若要發出此種類的指令，使用者必須是 QMQMADM 群組的成員或具有 *ALLOBJ 權限。 例如，GRTMQMAUT 及 STRMQMQSC 屬於此種類。

群組 2

若要在此種類中發出指令，使用者不需要是 QMQMADM 群組的成員或具有 *ALLOBJ 權限。 相反地，需要兩個層次的權限:

- 使用者需要 IBM i 權限才能使用指令。 使用 GRTOBJAUT 指令授與此權限。
- 使用者需要 IBM MQ 權限，才能存取與指令相關聯的任何 IBM MQ 物件。 使用 GRTMQMAUT 指令授與此權限。

下列範例顯示此群組中的指令:

- CRTMQMQ，建立 MQM 佇列
- CHGMQMPRC，變更 MQM 處理程序
- DLTMQMNL，刪除 MQM 名單
- DSPMQMAUTI，顯示 MQM 鑑別資訊
- CRTMQMCHL，建立 MQM 通道

如需此指令群組的相關資訊，請參閱 第 75 頁的『授權應用程式使用 IBM MQ』。

如需群組 1 和群組 2 指令的完整清單，請參閱 第 131 頁的『IBM i 上 IBM MQ 物件的存取權』

如需在 IBM i 上管理 IBM MQ 所需之權限的相關資訊，請參閱 管理 IBM i。

z/OS 在 *z/OS* 上管理 *IBM MQ* 的權限

此主題集合說明您管理 IBM MQ for z/OS 所需之權限的各個層面。

z/OS *z/OS* 上的權限檢查

IBM MQ for z/OS 使用「系統授權機能 (SAF)」，將權限檢查要求遞送至外部安全管理程式 (ESM)，例如 z/OS Security Server 資源存取控制機能 (RACF)。 IBM MQ 不會自行執行任何授權檢查。

假設您使用 RACF 作為 ESM。 如果您使用不同的 ESM，則可能需要以與 ESM 相關的方式來解譯為 RACF 提供的資訊。

您可以指定是要個別開啟或關閉每一個佇列管理程式的權限檢查，還是佇列共用群組中每一個佇列管理程式的權限檢查。此控制層次稱為 子系統安全。 如果您關閉特定佇列管理程式的子系統安全，則不會對該佇列管理程式執行任何權限檢查。

如果您開啟特定佇列管理程式的子系統安全，則可以在兩個層次執行權限檢查:

佇列共用群組層次安全

權限檢查會使用佇列共用群組中所有佇列管理程式所共用的 RACF 設定檔。 這表示要定義及維護的設定檔較少，使安全管理更容易。

佇列管理程式層次安全 (queue manager level security)

權限檢查使用特定於佇列管理程式的 RACF 設定檔。

您可以使用佇列共用群組與佇列管理程式層次安全的組合。 例如，您可以安排佇列管理程式特定的設定檔，以置換其所屬佇列共用群組的設定檔。

透過定義 交換器設定檔來開啟或關閉子系統安全、佇列共用群組層次安全及佇列管理程式層次安全。 交換器設定檔是對 IBM MQ 具有特殊意義的一般 RACF 設定檔。

z/OS *z/OS* 上的指令安全及指令資源安全

指令安全與發出指令的權限相關; 指令資源權限與對資源執行作業的權限相關。 兩者都是使用 RACF 類別來實作 y。

當 IBM MQ 管理者發出 MQSC 指令時，會執行權限檢查。 這稱為 指令安全。

如果要實作指令安全，您必須定義某些 RACF 設定檔，並在必要層次提供這些設定檔的必要群組和使用者 ID 存取權。 指令安全的設定檔名稱包含 MQSC 指令的名稱。

部分 MQSC 指令會在 IBM MQ 資源上執行作業，例如 DEFINE QLOCAL 指令，以建立本端佇列。 當管理者發出 MQSC 指令時，會執行權限檢查，以判斷是否可以對指令中指定的資源執行所要求的作業。 這稱為 指令資源安全。

如果要實作指令資源安全，您必須定義某些 RACF 設定檔，並在必要層次提供這些設定檔的必要群組和使用者 ID 存取權。 指令資源安全的設定檔名稱包含 IBM MQ 資源的名稱及其類型 (QUEUE、PROCESS、NAMELIST、TOPIC、AUTHINFO 或 CHANNEL)。

指令安全和指令資源安全是獨立的。 例如，當管理者發出指令時:

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

會執行下列權限檢查:

- 指令安全會檢查管理者是否有權發出 DEFINE QLOCAL 指令。
- 指令資源安全會檢查管理者是否有權在稱為 MOON.EUROPA。

透過定義交換器設定檔，可以開啟或關閉指令安全及指令資源安全。

z/OS *z/OS 上的 MQSC 指令及系統指令輸入佇列*

請使用本主題來瞭解指令伺服器如何處理導向 z/OS 上系統指令輸入佇列的 MQSC 指令。

當指令伺服器從系統指令輸入佇列擷取包含 MQSC 指令的訊息時，也會使用指令安全和指令資源安全。用於權限檢查的使用者 ID 是在包含 MQSC 指令之訊息的訊息描述子的 *UserIdentifier* 欄位中找到的使用者 ID。此使用者 ID 必須對處理指令的佇列管理程式具有必要權限。 如需 *UserIdentifier* 欄位及其設定方式的相關資訊，請參閱 訊息環境定義。

在下列情況下，會將包含 MQSC 指令的訊息傳送至系統指令輸入佇列:

- 作業及控制台會將 MQSC 指令傳送至目標佇列管理程式的系統指令輸入佇列。 MQSC 指令對應於您在畫面上選擇的動作。 每一則訊息中的 *UserIdentifier* 欄位都會設為管理者的 TSO 使用者 ID。
- IBM MQ 公用程式 CSQUTIL 的 COMMAND 函數會將輸入資料集中的 MQSC 指令傳送至目標佇列管理程式的系統指令輸入佇列。 COPY 和 EMPTY 函數會傳送 DISPLAY QUEUE 和 DISPLAY STGCLASS 指令。 每一則訊息中的 *UserIdentifier* 欄位都會設為工作使用者 ID。
- CSQINPX 資料集中的 MQSC 指令會傳送至通道起始程式所連接之佇列管理程式的系統指令輸入佇列。 每一則訊息中的 *UserIdentifier* 欄位會設為通道起始程式位址空間使用者 ID。

  從 CSQINP1 及 CSQINP2 資料集發出 MQSC 指令時，不會執行任何權限檢查。 您可以使用 RACF 資料集保護來控制容許誰更新這些資料集。
- 在佇列共用群組內，通道起始程式可能會將 START CHANNEL 指令傳送至它所連接之佇列管理程式的系統指令輸入佇列。 當透過觸發來啟動使用共用傳輸佇列的出埠通道時，會傳送指令。 每一則訊息中的 *UserIdentifier* 欄位會設為通道起始程式位址空間使用者 ID。
- 應用程式可以將 MQSC 指令傳送至系統指令輸入佇列。 依預設，每一則訊息中的 *UserIdentifier* 欄位會設為與應用程式相關聯的使用者 ID。
- 在 AIX, Linux, and Windows 系統上， **runmqsc** 控制指令可以在間接模式下使用，將 MQSC 指令傳送至 z/OS 上佇列管理程式的系統指令輸入佇列。 每一則訊息中的 *UserIdentifier* 欄位都設為發出 **runmqsc** 指令之管理者的使用者 ID。

z/OS *z/OS 上佇列管理程式資料集的存取權*

IBM MQ for z/OS 管理者需要存取佇列管理程式資料集的權限。 使用本主題來瞭解哪些資料集需要 RACF 保護。

這些資料集包括:

- 在佇列管理程式的已啟動作業程序中， CSQINP1、 CSQINP2 及 CSQINPT 所參照的資料集。
- 佇列管理程式的頁面集、作用中日誌資料集、保存日誌資料集及引導資料集 (BSDS)
- 通道起始程式的已啟動作業程序中 CSQXLIB 及 CSQINPX 所參照的資料集

您必須保護資料集，以便沒有未獲授權的使用者可以啟動佇列管理程式或取得任何佇列管理程式資料的存取權。 若要這麼做，請使用 RACF 資料集保護。

授權應用程式使用 IBM MQ

當應用程式存取物件時，與應用程式相關聯的使用者 ID 需要適當的權限。

應用程式可以透過發出 MQI 呼叫來存取下列 IBM MQ 物件:

- 佇列管理程式
- 佇列
- Processes
- 名單
- 主題

應用程式也可以使用 PCF 指令來管理 IBM MQ 物件。 當處理 PCF 指令時，它會使用放置 PCF 訊息之使用者 ID 的權限環境定義。

在此環境定義中，應用程式包括由使用者及供應商撰寫的應用程式，以及隨 IBM MQ for z/OS 提供的應用程式。 IBM MQ for z/OS 隨附的應用程式包括:

- 作業及控制面板
- IBM MQ 公用程式 CSQUTIL
- 無法傳送郵件的佇列處理程式公用程式 CSQUDLQH

使用 IBM MQ classes for Java、 IBM MQ classes for JMS、 IBM MQ classes for .NET 或 Message Service Clients for C/C++ 及 .NET 的應用程式會間接使用 MQI。

MCA 也會發出 MQI 呼叫，以及與 MCA 相關聯的使用者 ID，需要存取這些 IBM MQ 物件的權限。 如需這些使用者 ID 及其所需權限的相關資訊，請參閱 第 92 頁的『通道授權』。

在 z/OS 上，應用程式也可以使用 MQSC 指令來存取這些 IBM MQ 物件，但在這些情況下，指令安全及指令資源安全會提供權限檢查。 z/OS 如需相關資訊，請參閱 第 74 頁的『z/OS 上的指令安全及指令資源安全』 和 第 75 頁的『z/OS 上的 MQSC 指令及系統指令輸入佇列』。

在 IBM i 上，在群組 2 中發出 CL 指令的使用者可能需要權限，才能存取與指令相關聯的 IBM MQ 物件。 如需相關資訊，請參閱 第 76 頁的『執行權限檢查時』。

執行權限檢查時

當應用程式嘗試存取佇列管理程式、佇列、處理程序或名單時，會執行權限檢查。

在 IBM i 上，當使用者在群組 2 中發出可存取任何這些 IBM MQ 物件的 CL 指令時，也可能會執行權限檢查。 在下列情況下會執行檢查:

當應用程式使用 MQCONN 或 MQCONNX 呼叫連接至佇列管理程式時

佇列管理程式會向作業系統詢問與應用程式相關聯的使用者 ID。 然後，佇列管理程式會檢查使用者 ID 是否已獲授權連接至該佇列管理程式，並保留該使用者 ID 以供未來檢查。

使用者不需要登入 IBM MQ。 IBM MQ 假設使用者已登入基礎作業系統，並由它進行鑑別。

當應用程式使用 MQOPEN 或 MQPUT1 呼叫開啟 IBM MQ 物件時

所有權限檢查都是在開啟物件時執行，而不是在稍後存取物件時執行。 例如，當應用程式開啟佇列時，會執行權限檢查。 當應用程式將訊息放入佇列或從佇列取得訊息時，不會執行這些動作。

當應用程式開啟物件時，它會指定需要對物件執行的作業類型。 例如，應用程式可能會開啟佇列以瀏覽其中的訊息、從其中取得訊息，但不會在其中放置訊息。 對於每一種類型的作業，佇列管理程式會檢查與應用程式相關聯的使用者 ID 是否具有執行該作業的權限。

當應用程式開啟佇列時，會對物件描述子的 `ObjectName` 欄位中指定的物件執行權限檢查。 `ObjectName` 欄位用於 `MQOPEN` 或 `MQPUT1` 呼叫。 如果物件是別名佇列或遠端佇列定義，則會對物件本身執行權限檢查。 它們不會在別名佇列或遠端佇列定義所解析的佇列上執行。 這表示使用者不需要許可權即可存取它。 將建立佇列的權限限制為特許使用者。 如果您不這麼做，使用者只要建立別名，就可以略過一般存取控制。

應用程式可以明確參照遠端佇列。 它會將物件描述子中的 `ObjectName` 及 `ObjectQMgr` **名稱** 欄位設為遠端佇列及遠端佇列管理程式的名稱。 對與遠端佇列管理程式同名的傳輸佇列執行權限檢查:

- z/OS 在 z/OS 上，會對符合遠端佇列管理程式名稱的 RACF 佇列設定檔進行檢查，且不論是否在本端定義此傳輸佇列，都將執行檢查。

- Multi 在 多平台上，如果正在使用叢集作業，則會對符合遠端佇列管理程式名稱的 RQMNAME 設定檔進行檢查。

應用程式可以明確參照叢集佇列，方法是將物件描述子中的 ObjectName 欄位設為叢集佇列的名稱。會對叢集傳輸佇列 SYSTEM.CLUSTER.TRANSMIT.QUEUE 執行權限檢查。

動態佇列的權限是根據其衍生來源的模型佇列，但不一定相同; 請參閱附註 1。

從作業系統取得佇列管理程式用於權限檢查的使用者 ID。 當應用程式連接至佇列管理程式時，即會取得使用者 ID。 適當授權的應用程式可以發出 MQOPEN 呼叫，並指定替代使用者 ID; 然後會對替代使用者 ID 進行存取控制檢查。 使用替代使用者 ID 不會變更與應用程式相關聯的使用者 ID，只會變更用於存取控制檢查的使用者 ID。

當應用程式使用 MQSUB 呼叫來訂閱主題時

當應用程式訂閱主題時，它會指定需要執行的作業類型。 它是建立訂閱、變更現有訂閱，或回復現有訂閱而不變更它。 對於每一種類型的作業，佇列管理程式會檢查與應用程式相關聯的使用者 ID 是否具有執行作業的權限。

當應用程式訂閱主題時，會針對在主題樹狀結構中找到的主題物件執行權限檢查。 主題物件位於或高於應用程式訂閱的主題樹狀結構中的點。 權限檢查可能涉及多個主題物件的檢查。 從作業系統取得佇列管理程式用於權限檢查的使用者 ID。 當應用程式連接至佇列管理程式時，即會取得使用者 ID。

佇列管理程式會對訂閱者佇列執行權限檢查，但不會對受管理佇列執行權限檢查。

當應用程式使用 MQCLOSE 呼叫來刪除永久動態佇列時

在 MQCLOSE 呼叫上指定的物件控點，不一定與建立永久動態佇列的 MQOPEN 呼叫所傳回的控點相同。如果不同，佇列管理程式會檢查與發出 MQCLOSE 呼叫的應用程式相關聯的使用者 ID。 它會檢查使用者 ID 是否已獲授權刪除佇列。

當關閉訂閱以移除它的應用程式未建立它時，需要適當的權限才能移除它。

當指令伺服器處理在 IBM MQ 物件上運作的 PCF 指令時

此規則包括 PCF 指令在鑑別資訊物件上運作的情況。

用於權限檢查的使用者 ID 是在 PCF 指令訊息描述子的 UserIdentifier 欄位中找到的使用者 ID。 此使用者 ID 必須對處理指令的佇列管理程式具有必要權限。 以相同方式處理封裝在 Escape PCF 指令內的對等 MQSC 指令。 如需 UserIdentifier 欄位及其設定方式的相關資訊，請參閱 第 78 頁的『訊息環境定義』。

IBM i 在 IBM i 上，當使用者在群組 2 中發出處理 IBM MQ 物件的 CL 指令時

此規則包括群組 2 中的 CL 指令在鑑別資訊物件上運作的情況。

執行檢查以判定使用者是否有權對與指令相關聯的 IBM MQ 物件進行操作。 除非使用者是 QMQMADM 群組的成員或具有 *ALLOBJ 權限，否則會執行檢查。 所需的權限視指令對物件執行的作業類型而定。 例如，指令 **CHGMQMQ**「變更 MQM 佇列」需要權限，才能變更指令所指定佇列的屬性。 相反地，指令 **DSPMQMQ**「顯示 MQM 佇列」需要權限，才能顯示指令所指定佇列的屬性。

許多指令會在多個物件上操作。 例如，若要發出指令 **DLTMQMQ**「刪除 MQM 佇列」，需要下列權限:

- 連接至指令所指定佇列管理程式的權限
- 刪除指令所指定佇列的權限

部分指令完全對無物件執行作業。 在此情況下，使用者只需要 IBM i 權限，即可發出下列其中一個指令。 **STRMQMLSR**，「啟動 MQM 接聽器」是這類指令的範例。

替代使用者權限

當應用程式開啟物件或訂閱主題時，應用程式可以在 MQOPEN、 MQPUT1 或 MQSUB 呼叫上提供使用者 ID。 它可以要求佇列管理程式使用此使用者 ID 進行權限檢查，而不是使用與應用程式相關聯的使用者 ID。

只有在同時符合下列兩個條件時，應用程式才會成功開啟物件:

- 與應用程式相關聯的使用者 ID 有權提供不同的使用者 ID 來進行權限檢查。 應用程式據說具有 替代使用者權限。
- 應用程式提供的使用者 ID 有權開啟所要求作業類型的物件，或訂閱主題。

訊息環境定義

訊息環境定義 資訊可讓擷取訊息的應用程式找出訊息發送端的相關資訊。 資訊保留在訊息描述子的欄位中，且欄位分成三個邏輯組件

這些部分如下:

身分環境定義 (identity context)
這些欄位包含將訊息放入佇列之應用程式使用者的相關資訊。

原始環境定義
這些欄位包含應用程式本身的相關資訊，以及訊息放入佇列的時間。

使用者環境定義
這些欄位包含應用程式可用來選取佇列管理程式應遞送之訊息的訊息內容。

當應用程式將訊息放入佇列時，應用程式可以要求佇列管理程式在訊息中產生環境定義資訊。 這是預設動作。 或者，它可以指定環境定義欄位不包含任何資訊。 與應用程式相關聯的使用者 ID 不需要特殊權限即可執行上述任一項。

應用程式可以在訊息中設定身分環境定義欄位，容許佇列管理程式產生原始環境定義，也可以設定所有環境定義欄位。 應用程式也可以將身分環境定義欄位從它擷取的訊息傳遞至它放置在佇列上的訊息，也可以傳遞所有環境定義欄位。 不過，與應用程式相關聯的使用者 ID 需要有設定或傳遞環境定義資訊的權限。 應用程式指定它在開啟即將放置訊息的佇列時，要設定或傳遞環境定義資訊，此時會檢查其權限。

以下是每一個環境定義欄位的簡要說明:

身分環境定義 (identity context)

UserIdentifier
與放置訊息的應用程式相關聯的使用者 ID。 如果佇列管理程式設定此欄位，則會設為應用程式連接至佇列管理程式時從作業系統取得的使用者 ID。

AccountingToken
可用來對因訊息而完成的工作收費的資訊。

ApplIdentityData
如果與應用程式相關聯的使用者 ID 有權設定身分環境定義欄位，或設定所有環境定義欄位，則應用程式可以將此欄位設為與身分相關的任何值。 如果佇列管理程式設定此欄位，則會設為空白。

原始環境定義

PutApplType
放置訊息的應用程式類型; 例如 CICS 交易。

PutApplName
放置訊息的應用程式名稱。

PutDate
放置訊息的日期。

PutTime
放置訊息的時間。

ApplOriginData
如果與應用程式相關聯的使用者 ID 有權設定所有環境定義欄位，則應用程式可以將此欄位設為與原點相關的任何值。 如果佇列管理程式設定此欄位，則會設為空白。

使用者環境定義

MQINQMP 或 **MQSETMP** 支援下列值:

MQPD_USER _CONTEXT

內容與使用者環境定義相關聯。

不需要特殊授權即可使用 MQSETMP 呼叫來設定與使用者環境定義相關聯的內容。

在 V7.0 或後續佇列管理程式上，會依照 MQOO_SAVE_ALL_CONTEXT 的說明來儲存與使用者環境定義相關聯的內容。 指定 MQOO_PASS_ALL_CONTEXT 的 MQPUT 會將內容從已儲存的環境定義複製到新訊息中。

MQPD_NO_CONTEXT

內容未與訊息環境定義相關聯。

MQRC_PD_ERROR 會拒絕無法辨識的值。 此欄位的起始值為 **MQPD_NO_CONTEXT**。

如需每一個環境定義欄位的詳細說明，請參閱 MQMD-訊息描述子。 如需如何使用訊息環境定義的相關資訊，請參閱 訊息環境定義。

IBM i ALW 在 IBM i *IBM i 、 AIX, Linux, and Windows 系統上使用 IBM MQ 物件的權限*

IBM MQ 隨附的授權服務元件稱為 物件權限管理程式 (OAM)。 它透過鑑別及授權檢查提供存取控制。

鑑別。

IBM MQ 隨附的 OAM 所執行的鑑別檢查是基本的，且僅在特定情況下執行。 它不是要符合在高度安全環境中預期的嚴格需求。

當應用程式連接至佇列管理程式時， OAM 會執行其鑑別檢查，且符合下列條件:

- 如果連接應用程式已提供 MQCSP 結構，且
- MQCSP 結構中的 *AuthenticationType* 屬性會獲得值 MQCSP_AUTH_USER_ID_AND_PWD ，以及
- 所配置 AUTHINFO 物件上的 CHCKLOCL 或 CHKCCLNT 值不是 'NONE'

OAM 中的鑑別步驟使用作業系統服務來驗證密碼，這些服務可能已配置為執行其他檢查，例如確保使用者名稱沒有太多不正確的密碼測試嘗試。

如果您撰寫新的授權服務元件，或從供應商取得授權服務元件，則可以使用替代鑑別機制。

授權。

授權檢查是綜合性的，旨在符合最正常的需求。

當應用程式發出 MQI 呼叫來存取佇列管理程式、佇列、處理程序、主題或名稱清單時，會執行授權檢查。 它們也會在其他時間執行，例如在「指令伺服器」執行指令時。

在 IBM i IBM i 、 AIX, Linux, and Windows 系統上，當應用程式發出 MQI 呼叫來存取 IBM MQ 物件 (即佇列管理程式、佇列、處理程序、主題或名單) 時， 授權服務 會提供存取控制。 這包括檢查替代使用者權限，以及設定或傳遞環境定義資訊的權限。

Windows 在 Windows 上， OAM 會授與 Administrators 群組成員存取所有 IBM MQ 物件的權限，即使已啟用 UAC 也一樣。 此外，在 Windows 系統上， SYSTEM 帳戶對 IBM MQ 資源具有完整存取權。

當 PCF 指令在其中一個 IBM MQ 物件或鑑別資訊物件上運作時，授權服務也會提供權限檢查。 以相同方式處理封裝在 Escape PCF 指令內的對等 MQSC 指令。

IBM i 在 IBM i 上，除非使用者是 QMQMADM 群組的成員或具有 *ALLOBJ 權限，否則當使用者在群組 2 中針對任何這些 IBM MQ 物件或鑑別資訊物件發出 CL 指令時，授權服務也會提供權限檢查。

授權服務是 可安裝的服務，這表示它由一或多個 可安裝的服務元件實作。 每一個元件都是使用記載的介面來呼叫。 這可讓使用者及供應商提供元件，以擴增或取代 IBM MQ 產品所提供的元件。

IBM MQ 隨附的授權服務元件稱為物件權限管理程式 (OAM)。 您建立的每一個佇列管理程式都會自動啟用 OAM。

OAM 會維護其控制存取的每一個 IBM MQ 物件的存取控制清單 (ACL)。 在 AIX and Linux 系統上，只有群組 ID 可以出現在 ACL 中。 這表示群組的所有成員都具有相同的權限。 在 IBM i IBM i 及 Windows 系統上，使用者 ID 及群組 ID 都可以出現在 ACL 中。 這表示可以將權限授與個別使用者和群組。

群組和使用者 ID 都有 12 個字元的限制。 UNIX 平台通常會將使用者 ID 的長度限制為 12 個字元。 AIX 和 Linux 已提高此限制，但 IBM MQ 繼續在所有 UNIX 平台上遵守 12 個字元的限制。 如果您使用大於 12 個字元的使用者 ID， IBM MQ 會將它取代為 "UNKNOWN"值。 請勿定義值為 "UNKNOWN"的使用者 ID。

OAM 可以鑑別使用者，並變更適當的身分環境定義欄位。 您可以透過在 MQCONNX 呼叫中指定連線安全參數結構 (MQCSP) 來啟用此功能。 結構會傳遞至 OAM Authenticate User 函數 (MQZ_AUTHENTICATE_USER) ，其會設定適當的身分環境定義欄位。 如果來自 IBM MQ 用戶端的

MQCONNX 連線，則 MQCSP 中的資訊會傳送至用戶端透過用戶端連線及伺服器連線通道連接的佇列管理程式。 如果安全結束程式定義在該通道上，則 MQCSP 會傳遞至每一個安全結束程式，並可由結束程式變更。 安全結束程式也可以建立 MQCSP。 如需在此環境定義中使用安全結束程式的詳細資料，請參閱 通道安全結束程式。

警告: 在某些情況下，用戶端應用程式的 MQCSP 結構中的密碼將透過網路以純文字傳送。 若要確保用戶端應用程式密碼受到適當保護，請參閱 IBM MQCSP 密碼保護。

在 AIX, Linux, and Windows 系統上，控制指令 **setmqaut** 會授與及撤銷權限，並用來維護 ACL。 例如，指令:

```
setmqaut -m JUPITER -t queue -n MOON.EUROPA -g VOYAGER +browse +get
```

容許群組 VOYAGER 的成員瀏覽佇列 MOON.EUROPA 。 它也可讓成員從佇列中取得訊息。 若要稍後撤銷這些權限，請輸入下列指令:

```
setmqaut -m JUPITER -t queue -n MOON.EUROPA -g VOYAGER -browse -get
```

指令:

```
setmqaut -m JUPITER -t queue -n MOON.* -g VOYAGER +put
```

容許群組 VOYAGER 的成員將訊息放置在名稱以字元 MOON. 開頭的任何佇列上。 MOON.* 是通用設定檔的名稱。通用設定檔 可讓您使用單一 **setmqaut** 指令來授與一組物件的權限。

控制指令 **dspmqaut** 可用來顯示使用者或群組對指定物件的現行權限。控制指令 **dmpmqaut** 也可用來顯示與通用設定檔相關的現行權限。

IBM i 在 IBM i 上，管理者使用 CL 指令 GRTMQMAUT 來授與權限，並使用 CL 指令 RVKMQMAUT 來撤銷權限。 也可以使用通用設定檔。例如， CL 指令:

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

提供與前一個 **setmqaut** 指令範例相同的功能; 它容許群組 VOYAGER 的成員將訊息放置在名稱以字元開頭的任何佇列上 MOON.

IBM i CL 指令 DSPMQMAUT 會顯示使用者或群組對指定物件的現行權限。 CL 指令 WRKMQMAUT 及 WRKMQMAUTD 也可使用與物件及同屬設定檔相關的現行權限。

如果您不想要任何權限檢查，例如在測試環境中，您可以停用 OAM。

Multi 使用 *PCF* 存取 *OAM* 指令

在 IBM i、 AIX, Linux, and Windows 系統上，您可以使用 PCF 指令來存取 OAM 管理指令。

PCF 指令及其對等的 OAM 指令如下:

表 *8: PCF* 指令及其對等的 *OAM* 指令

PCF 指令	**OAM 指令**
查詢權限記錄	dmpmqaut
查詢實體權限	dspmqaut
設定權限記錄	setmqaut
刪除權限記錄	setmqaut 與 -remove 選項

setmqaut 及 **dmpmqaut** 指令僅限於 mqm 群組的成員。 對等 PCF 指令可以由任何群組中的使用者執行，這些使用者已獲授與對佇列管理程式的 dsp 及 chg 權限。

如需使用這些指令的相關資訊，請參閱 可程式指令格式簡介。

z/OS 在 *z/OS* 上使用 *IBM MQ* 物件的權限

在 z/OS 上，有七個種類的權限檢查與對 MQI 的呼叫相關聯。 您必須定義特定 RACF 設定檔，並提供這些設定檔的適當存取權。 使用 *RESLEVEL* 設定檔來控制要檢查多少使用者 ID。

與 MQI 呼叫相關聯的七個權限檢查種類:

連線安全
應用程式連接至佇列管理程式時所執行的權限檢查

佇列安全
當應用程式開啟佇列或刪除永久動態佇列時所執行的權限檢查

處理程序安全
當應用程式開啟處理程序物件時所執行的權限檢查

名單安全
當應用程式開啟名單物件時所執行的權限檢查

替代使用者安全性 (alternate user security)
當應用程式在開啟物件時要求替代使用者權限時所執行的權限檢查

環境定義安全 (context security)
當應用程式開啟佇列並指定它要在放置在佇列上的訊息中設定或傳遞環境定義資訊時所執行的權限檢查

主題安全
應用程式開啟主題時所執行的權限檢查

每一個種類的權限檢查的實作方式與指令安全及指令資源安全的實作方式相同。 您必須定義某些 RACF 設定檔，並提供必要層次對這些設定檔的必要群組和使用者 ID 存取權。 基於佇列安全，存取層次會決定應用程式可以在佇列上執行的作業類型。 基於環境定義安全，存取層次會決定應用程式是否可以:

- 傳遞所有環境定義欄位
- 傳遞所有環境定義欄位並設定身分環境定義欄位
- 傳遞並設定所有環境定義欄位

每一個種類的權限檢查可以透過定義交換器設定檔來開啟或關閉。

所有種類 (連線安全除外) 統稱為 *API*-資源安全。

依預設，當由於來自使用批次連線之應用程式的 MQI 呼叫而執行 API 資源安全檢查時，只會檢查一個使用者 ID。 當由於來自 CICS 或 IMS 應用程式或通道起始程式的 MQI 呼叫而執行檢查時，會檢查兩個使用者 ID。

不過，透過定義 *RESLEVEL* 設定檔，您可以控制是否檢查零個、一個或兩個使用者 ID。 當應用程式連接至佇列管理程式時，以及使用者 ID 對 RESLEVEL 設定檔的存取層次時，所檢查的使用者 ID 數目取決於與連線類型相關聯的使用者 ID。 與每一種連線類型相關聯的使用者 ID 為:

- 批次連線之連接作業的使用者 ID
- CICS 連線的 CICS 位址空間使用者 ID
- IMS 連線的 IMS 區域位址空間使用者 ID
- 通道起始程式連線的通道起始程式位址空間使用者 ID

如需在 z/OS 上使用 IBM MQ 物件之權限的相關資訊，請參閱 第 74 頁的『在 z/OS 上管理 IBM MQ 的權限』。

遠端傳訊的安全

本節處理安全的遠端傳訊層面。

您必須提供使用者使用 IBM MQ 機能的權限。 這是根據要對物件和定義採取的動作來組織的。 例如:

- 授權使用者可以啟動及停止佇列管理程式
- 應用程式必須連接至佇列管理程式，並且具有使用佇列的權限
- 訊息通道必須由授權使用者建立及控制
- 物件保留在檔案庫中，且可以限制對這些檔案庫的存取權

遠端網站上的訊息通道代理程式必須檢查遞送的訊息是否源自有權在此遠端網站上執行此動作的使用者。 此外，由於 MCA 可以從遠端啟動，因此可能需要驗證嘗試啟動 MCA 的遠端處理程序是否已獲授權執行此動作。 您有四種可能的方法來處理此問題:

1. 適當地使用 RCVR、RQSTR 或 CLUSRCVR 通道定義的 PutAuthority 屬性，以控制在將送入訊息放入佇列時，使用哪個使用者進行授權檢查。 請參閱 MQSC 指令參考手冊中的 DEFINE CHANNEL 指令說明。
2. 實作通道鑑別記錄，以拒絕不想要的連線嘗試，或根據下列項目來設定 MCAUSER 值: 遠端 IP 位址、遠端使用者 ID、提供的 TLS 主體識別名稱 (DN) 或遠端佇列管理程式名稱。
3. 實作 使用者結束程式 安全檢查，以確定對應的訊息通道已獲授權。 管理對應通道之安裝的安全可確保所有使用者都已適當授權，因此您不需要檢查個別訊息。
4. 實作 使用者結束程式 訊息處理，以確保會檢查個別訊息以取得授權。

IBM i *IBM MQ for IBM i 物件的安全*

本節處理安全的遠端傳訊層面。

您必須提供使用者使用 IBM MQ for IBM i 機能的權限。 此權限是根據要對物件及定義採取的動作來組織。 例如:

- 授權使用者可以啟動及停止佇列管理程式
- 應用程式需要連接至佇列管理程式，並具有使用佇列的權限
- 訊息通道需要由授權使用者建立及控制

遠端網站上的訊息通道代理程式必須檢查遞送的訊息是否衍生自有權在此遠端網站上設定訊息的使用者。 此外，由於 MCA 可以從遠端啟動，因此可能需要驗證嘗試啟動 MCA 的遠端處理程序是否已獲授權執行此動作。 您有四種可能的方法來處理此問題:

- 通道定義中的法令，指出訊息必須包含可接受的 環境定義 權限，否則會捨棄訊息。
- 實作通道鑑別記錄以拒絕不想要的連線嘗試，或根據下列其中一項來設定 MCAUSER 值: 遠端 IP 位址、遠端使用者 ID、提供的 TLS 識別名稱 (DN) 或遠端佇列管理程式名稱。
- 實作使用者結束程式安全檢查，以確定對應的訊息通道已獲授權。 管理對應通道之安裝的安全可確保所有使用者都已適當授權，因此您不需要檢查個別訊息。
- 實作使用者結束程式訊息處理程序，以確保會檢查個別訊息以取得授權。

以下是 IBM MQ for IBM i 安全運作方式的一些事實:

- 使用者由 IBM i 識別及鑑別。
- 應用程式所呼叫的佇列管理程式服務會以佇列管理程式使用者設定檔的權限執行，但在使用者的處理程序中。
- 使用者指令所呼叫的佇列管理程式服務會以佇列管理程式使用者設定檔的權限來執行。

Linux AIX *AIX and Linux 上物件的安全*

如果此 ID 將使用 IBM MQ 管理指令，則管理使用者必須是系統上 mqm 群組的一部分 (包括 root)。

您應該一律以 "mqm" 使用者 ID 執行 amqcrsta。

AIX and Linux 上的使用者 ID

佇列管理程式會將所有大寫或大小寫混合格式的使用者 ID 轉換成小寫。 然後佇列管理程式會將使用者 ID 插入訊息的環境定義部分，或檢查其授權。 因此，授權僅基於小寫 ID。

Windows *Windows 系統上物件的安全*

如果此 ID 將使用 IBM MQ 管理指令，則管理使用者必須同時隸屬於 Windows 系統上的 mqm 群組及 administrators 群組。

Windows 系統上的使用者 ID

在 Windows 系統上， 如果未安裝任何訊息結束程式，佇列管理程式會將任何大寫或大小寫混合格式的使用者 ID 轉換為小寫。 然後佇列管理程式會將使用者 ID 插入訊息的環境定義部分，或檢查其授權。 因此，授權僅基於小寫 ID。

跨系統的使用者 *ID*

AIX, Linux, and Windows 系統以外的平台在訊息中對使用者 ID 使用大寫字元。 若要容許 AIX, Linux, and Windows 系統在訊息中使用小寫使用者 ID ， 訊息通道代理程式 (MCA) 必須執行英文字母的適當轉換。

為了容許 AIX, Linux, and Windows 系統在訊息中使用小寫使用者 ID ， 這些平台上的訊息通道代理程式 (MCA) 會執行下列轉換:

在傳送端
如果未安裝任何訊息結束程式，則所有使用者 ID 中的英文字母都會轉換為大寫字元。

在接收端
如果未安裝訊息結束程式，則所有使用者 ID 中的英文字母都會轉換為小寫字元。

如果您基於任何其他原因在 AIX, Linux, and Windows 上提供訊息結束程式，則不會執行自動轉換。

使用自訂授權服務

IBM MQ 提供可安裝的授權服務。 您可以選擇安裝替代服務。

IBM MQ 隨附的授權服務元件稱為「物件權限管理程式 (OAM)」。 如果 OAM 未提供您需要的授權機能，您可以撰寫自己的授權服務元件。 可安裝服務介面參照資訊中說明授權服務元件必須實作的可安裝服務功能。

用戶端的存取控制

存取控制是根據使用者 ID。 可以有許多要管理的使用者 ID ， 且使用者 ID 可以採用不同的格式。 您可以將伺服器連線通道內容 MCAUSER 設為特殊使用者 ID 值，供用戶端使用。

IBM MQ 中的存取控制基於使用者 ID。 通常會使用進行 MQI 呼叫之處理程序的使用者 ID。 對於 MQ MQI 用戶端，伺服器連線 MCA 會代表 MQ MQI 用戶端進行 MQI 呼叫。 您可以為伺服器連線 MCA 選取替代使用者 ID ， 以用於進行 MQI 呼叫。 替代使用者 ID 可以與用戶端工作站相關聯，也可以與您選擇組織及控制用戶端存取權的任何項目相關聯。 使用者 ID 需要在伺服器上配置必要的權限，才能發出 MQI 呼叫。 選擇替代使用者 ID 會比容許用戶端使用伺服器連線 MCA 的權限進行 MQI 呼叫更理想。

表 *9:* 伺服器連線通道使用的使用者 *ID*

使用者 ID	使用時
安全結束程式所設定的使用者 ID	除非被 **CHLAUTH TYPE(BLOCKUSER)** 規則封鎖，否則使用。 如需相關資訊，請參閱下列小節 第 84 頁的『在安全結束程式中設定使用者 ID』 。
由 CHLAUTH 規則設定的使用者 ID	除非被安全結束程式置換，否則使用。 如需相關資訊，請參閱 通道鑑別記錄 。
SVRCONN 通道定義中的 **MCAUSER** 屬性所定義的使用者 ID	除非由安全結束程式或 CHLAUTH 規則置換，否則使用。
從用戶端機器傳送的使用者 ID	在沒有任何其他方法設定使用者 ID 時使用。
啟動伺服器連線通道的使用者 ID	在未使用任何其他方法設定使用者 ID 且未傳送任何用戶端使用者 ID 時使用。 如需相關資訊，請參閱下列小節 第 84 頁的『執行通道程式的使用者 ID』 。

因為伺服器連線 MCA 代表遠端使用者進行 MQI 呼叫，所以請務必考量伺服器連線 MCA 代表遠端用戶端發出 MQI 呼叫的安全含意，以及如何管理大量使用者的存取權。

- 其中一種方法是讓伺服器連線 MCA 在其自己的權限上發出 MQI 呼叫。 但請注意，通常不想要伺服器連線 MCA (具有強大的存取功能) 代表用戶端使用者發出 MQI 呼叫。

- 另一種方法是使用來自用戶端的使用者 ID。 伺服器連線 MCA 可以使用用戶端使用者 ID 的存取功能來發出 MQI 呼叫。 這一方法提出了一些需要考慮的問題:
  1. 在不同平台上，使用者 ID 有不同的格式。 如果用戶端上的使用者 ID 格式與伺服器上可接受的格式不同，這有時會造成問題。
  2. 可能有許多用戶端具有不同的使用者 ID，且正在變更使用者 ID。 需要在伺服器上定義及管理 ID。
  3. 要信任使用者 ID 嗎? 任何使用者 ID 都可以從用戶端傳送，不一定是已登入使用者的 ID。 例如，基於安全理由，用戶端可能傳送具有完整 mqm 權限的 ID，而此 ID 是刻意在伺服器上定義的。
- 偏好的方法是在伺服器定義用戶端識別記號，因此限制用戶端連接應用程式的功能。 這通常是透過將伺服器連線通道內容 MCAUSER 設為用戶端要使用的特殊使用者 ID 值，以及定義少數 ID 供伺服器上具有不同授權層次的用戶端使用。

在安全結束程式中設定使用者 ID

對於 IBM MQ MQI clients，發出 MQI 呼叫的處理程序是伺服器連線 MCA。 伺服器連線 MCA 使用的使用者 ID 包含在 MQCD 的 `MCAUserIdentifier` 或 `LongMCAUserIdentifier` 欄位中。 這些欄位的內容由下列設定:

- 安全結束程式所設定的任何值
- 來自用戶端的使用者 ID
- MCAUSER (在伺服器連線通道定義中)

當呼叫安全結束程式時，它可以置換可見的值。

- 如果伺服器連線通道 MCAUSER 屬性設為非空白，則會使用 MCAUSER 值。
- 如果伺服器連線通道 MCAUSER 屬性空白，則會使用從用戶端收到的使用者 ID。
- 如果伺服器連線通道 MCAUSER 屬性空白，且未從用戶端收到任何使用者 ID，則會使用啟動伺服器連線通道的使用者 ID。

當使用用戶端安全結束程式時， IBM MQ 用戶端不會將主張的使用者 ID 傳送至伺服器。

執行通道程式的使用者 ID

當使用者 ID 欄位衍生自啟動伺服器連線通道的使用者 ID 時，會使用下列值:

- z/OS 對於 z/OS，由 z/OS 啟動程序表格指派給通道起始程式啟動作業的使用者 ID。
- 若為 TCP/IP (非 z/OS)，則是來自 `inetd.conf` 項目的使用者 ID，或啟動接聽器的使用者 ID。
- 對於 SNA (非 z/OS)，這是來自「SNA 伺服器」項目或 (如果沒有) 送入連接要求的使用者 ID，或啟動接聽器的使用者 ID。
- 若為 NetBIOS 或 SPX，為啟動接聽器的使用者 ID。

如果有任何伺服器連線通道定義將 MCAUSER 屬性設為空白，則用戶端可以使用此通道定義，以用戶端提供的使用者 ID 所決定的存取權限來連接至佇列管理程式。 如果執行佇列管理程式的系統容許未獲授權的網路連線，則這可能是安全暴露。 IBM MQ 預設伺服器連線通道 (SYSTEM.DEF.SVRCONN) 將 MCAUSER 屬性設為空白。 若要防止未獲授權的存取，請使用無權存取 IBM MQ MQ 物件的使用者 ID 來更新預設定義的 MCAUSER 屬性。

使用者 ID 的大小寫

當您使用 `runmqsc` 定義通道時，除非使用者 ID 包含在單引號內，否則 MCAUSER 屬性會變更為大寫。

ALW 對於 AIX, Linux, and Windows 上的伺服器，從用戶端收到的 `MCAUserIdentifier` 欄位內容會變更為小寫。

IBM i 對於 IBM i 上的伺服器，從用戶端收到的 `LongMCAUserIdentifier` 欄位內容會變更為大寫。

Linux AIX 對於 AIX and Linux 系統上的伺服器，從用戶端收到的 LongMCAUserIdentifier 欄位內容會變更為小寫。

依預設，使用 IBM MQ JMS 連結應用程式時所傳遞的使用者 ID，是應用程式執行所在之 JVM 的使用者 ID。

也可以透過 createQueueConnection 方法傳遞使用者 ID。

規劃機密性

規劃如何保持資料機密。

您可以在應用程式層次或鏈結層次實作機密性。 您可以選擇使用 TLS，在此情況下，您必須規劃數位憑證的使用。 如果標準機能無法滿足您的需求，您也可以使用通道結束程式。

相關概念

第 85 頁的『比較鏈結層次安全和應用程式層次安全』
這個主題包含鏈結層次安全和應用程式層次安全的各個層面的相關資訊，並比較兩個安全層次。

第 89 頁的『通道結束程式』
通道結束程式 是在 MCA 處理順序中的已定義位置呼叫的程式。 使用者和供應商可以撰寫自己的通道結束程式。 部分由 IBM 提供。

第 94 頁的『使用 SSL/TLS 保護通道』
IBM MQ 中的 TLS 支援使用佇列管理程式鑑別資訊物件及各種 MQSC 指令。 您也必須考量使用數位憑證。

比較鏈結層次安全和應用程式層次安全

這個主題包含鏈結層次安全和應用程式層次安全的各個層面的相關資訊，並比較兩個安全層次。

鏈結層次及應用程式層次安全在 第 85 頁的圖 10 中說明。

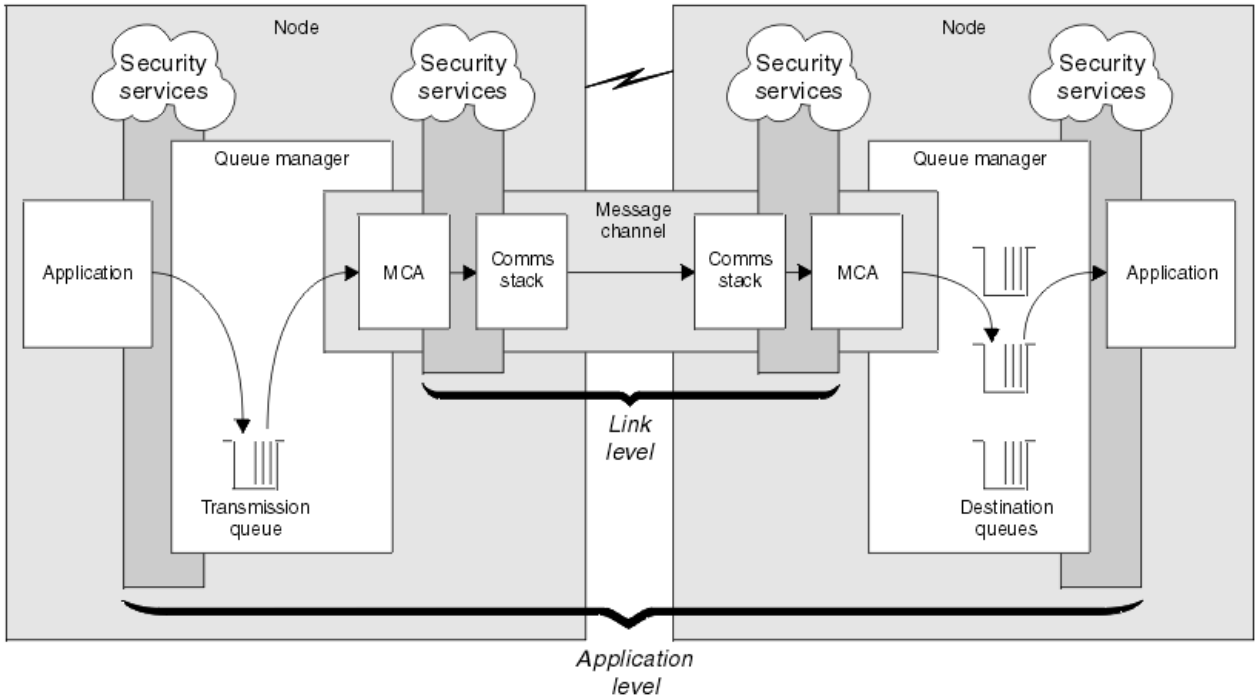


圖 *10:* 鏈結層次安全及應用程式層次安全

保護佇列中的訊息

當訊息從一個佇列管理程式傳送至另一個佇列管理程式時，鏈結層次安全可保護訊息。 當訊息是透過不安全的網路來傳輸時，尤其重要。 不過，當訊息儲存在來源佇列管理程式、目的地佇列管理程式或中間佇列管理程式的佇列中時，它無法保護訊息。

z/OS V 9.2.0 z/OS 資料集加密可以為儲存在佇列上的訊息提供部分保護，但僅適用於本端佇列管理程式上的靜態資料。 請參閱 IBM MQ for z/OS 上具有資料集加密之靜態資料的機密性 一節。 的文件以取得相關資訊。

相比之下，應用程式層次安全可以在訊息儲存在佇列時保護訊息，即使未使用分散式佇列也適用。 這是鏈結層次安全與應用程式層次安全之間的主要差異，如 第 85 頁的圖 10 中所示。

佇列管理程式未在受控制及信任的環境中執行

如果佇列管理程式在受控制且受信任的環境中執行，則 IBM MQ 提供的存取控制機制可能被視為足以保護其佇列上儲存的訊息。 如果只涉及本端佇列作業，且訊息永不離開佇列管理程式，則尤其如此。 在此情況下，應用程式層次安全可能被視為不必要。

如果訊息傳送至另一個佇列管理程式 (也在受管制且授信的環境中執行)，或從這類佇列管理程式接收到訊息，則應用程式層次安全也可能被視為不必要。 當訊息傳送至或從未在受控制且授信環境中執行的佇列管理程式接收時，應用程式層次安全的需求會變得更需要。

成本差異

就管理和效能而言，應用程式層次安全的成本可能超過鏈結層次安全。

管理成本可能會更高，因為配置及維護可能有更多限制。 例如，您可能需要確保特定使用者僅傳送特定類型的訊息，並僅將訊息傳送至特定目的地。 相反地，您可能需要確保特定使用者只接收特定類型的訊息，並只從特定來源接收訊息。 您可能需要為透過單一訊息通道交換訊息的每一對使用者配置及維護規則，而不是在該通道上管理鏈結層次安全服務。

如果每次應用程式放置或取得訊息時都呼叫安全服務，則可能會影響效能。

組織傾向於先考量鏈結層次安全，因為它可能更容易實作。 如果他們發現鏈結層次安全無法滿足其所有需求，則會考量應用程式層次安全。

元件可用性

一般而言，在分散式環境中，安全服務至少需要兩個系統上的元件。 例如，訊息可能在一個系統上加密，而在另一個系統上解密。 這同時適用於鏈結層次安全和應用程式層次安全。

在異質環境中，如果使用不同的平台，且每一個平台都有不同的安全功能層次，則安全服務的必要元件可能無法用於每一個需要它們的平台，且其形式容易使用。 這可能是應用程式層次安全的問題多於鏈結層次安全的問題，尤其是當您想要透過購買來自各種來源的元件來提供自己的應用程式層次安全時。

無法傳送的郵件佇列中的訊息

如果訊息受應用程式層次安全保護，則在訊息因任何原因而無法到達其目的地並置於無法傳送郵件的佇列時，可能會發生問題。 如果您無法解決如何從訊息描述子及無法傳送的郵件標頭中處理訊息，則可能需要檢查應用程式資料的內容。 如果應用程式資料已加密且只有預期的收件者可以解密，則您無法執行此動作。

應用程式層次安全無法執行的動作

應用程式層次安全不是完整的解決方案。 即使您實作應用程式層次安全，仍可能需要一些鏈結層次安全服務。 例如：

- 當通道啟動時，兩個 MCA 的交互鑑別仍可能是需求。 這只能由鏈結層次安全服務來執行。
- 應用程式層次安全無法保護包含內嵌訊息描述子的傳輸佇列標頭 MQXQH。 除了訊息資料之外，它也無法保護 IBM MQ 通道通訊協定流程中的資料。 只有鏈結層次安全可以提供此保護。
- 如果在 MQI 通道的伺服器端呼叫應用程式層次安全服務，則這些服務無法保護透過通道傳送之 MQI 呼叫的參數。 尤其是 MQPUT、 MQPUT1 或 MQGET 呼叫中的應用程式資料不受保護。 在此情況下，只有鏈結層次安全可以提供保護。

鏈結層次安全 *(link level security)*

鏈結層次安全 是指由 MCA、通訊子系統或兩者一起運作的組合直接或間接呼叫的那些安全服務。

鏈結層次安全在 第 85 頁的圖 10 中說明。

以下是一些鏈結層次安全服務的範例:

- 訊息通道每一端的 MCA 可以鑑別其夥伴。 當通道啟動且已建立通訊連線時，但在任何訊息開始傳送之前，即會執行此動作。 如果任一端鑑別失敗，則會關閉通道，且不會傳送任何訊息。 這是識別及鑑別服務的範例。
- 訊息可以在通道傳送端加密，並在接收端解密。 這是機密性服務的範例。
- 可以在通道的接收端檢查訊息，以判斷其內容是否在透過網路傳輸時刻意修改。 這是資料完整性服務的範例。

IBM MQ 提供的鏈結層次安全

IBM MQ 中提供機密性和資料完整性的主要方法是使用 TLS。 如需在 IBM MQ 中使用 TLS 的相關資訊，請參閱 第 20 頁的『IBM MQ 中的 TLS 安全通訊協定』。 對於鑑別， IBM MQ 提供使用通道鑑別記錄的機能。 通道鑑別記錄在個別通道或通道群組層次提供對授與連接系統之存取權的精確控制。 如需相關資訊，請參閱 第 40 頁的『通道鑑別記錄』。

提供您自己的鏈結層次安全

您可以提供自己的鏈結層次安全服務。 撰寫您自己的通道結束程式是提供您自己的鏈結層次安全服務的主要方式。

通道結束程式在 第 89 頁的『通道結束程式』中引進。 同一主題也說明 IBM MQ for Windows 隨附的通道結束程式 (SSPI 通道結束程式)。 此通道結束程式以來源格式提供，因此您可以修改原始碼以符合您的需求。 如果此通道結束程式或其他供應商提供的通道結束程式不符合您的需求，您可以自行設計及撰寫。 本主題建議通道結束程式提供安全服務的方式。 如需如何撰寫通道結束程式的相關資訊，請參閱 撰寫通道結束程式。

使用安全結束程式的鏈結層次安全

安全結束程式通常成對運作; 通道兩端各一個。 在通道啟動時完成起始資料協議之後，會立即呼叫它們。

安全結束程式可用來提供識別及鑑別、存取控制及機密性。

使用訊息結束程式的鏈結層次安全

訊息結束程式只能在訊息通道上使用，不能在 MQI 通道上使用。 它可以存取傳輸佇列標頭 MQXQH ，其中包括內嵌的訊息描述子，以及訊息中的應用程式資料。 它可以修改訊息的內容並變更其長度。

訊息結束程式可以用於任何需要存取整個訊息的目的，而不是其中的一部分。

訊息結束程式可用來提供識別及鑑別、存取控制、機密性、資料完整性及不可否認性，以及安全以外的原因。

使用傳送及接收結束程式的鏈結層次安全

傳送及接收結束程式可以在訊息及 MQI 通道上使用。 它們是針對在通道上流動的所有資料類型，以及雙向的流程所呼叫。

傳送及接收結束程式可以存取每一個傳輸區段。 他們可以修改其內容並變更其長度。

在訊息通道上，如果 MCA 需要分割訊息並在多個傳輸區段中傳送訊息，則會針對包含部分訊息的每一個傳輸區段呼叫傳送結束程式，並在接收端針對每一個傳輸區段呼叫接收結束程式。 如果 MQI 呼叫的輸入或輸出參數太大，無法在單一傳輸區段中傳送，則在 MQI 通道上也會發生相同情況。

在 MQI 通道上，傳輸區段的位元組 10 會識別 MQI 呼叫，並指出傳輸區段是否包含呼叫的輸入或輸出參數。 傳送及接收結束程式可以檢查此位元組，以判定 MQI 呼叫是否包含可能需要保護的應用程式資料。

第一次呼叫傳送結束程式時，若要獲得並起始設定它需要的任何資源，它可以要求 MCA 在保留傳輸區段的緩衝區中保留指定的空間量。 例如，當稍後呼叫它來處理傳輸區段時，它可以使用此空間來新增加密金鑰或數位簽章。 通道另一端的對應接收結束程式可以移除傳送結束程式所新增的資料，並使用它來處理傳輸區段。

傳送及接收結束程式最適合其不需要瞭解所處理之資料結構的用途，因此可以將每一個傳輸區段視為二進位物件。

傳送及接收結束程式可用來提供機密性和資料完整性，以及用於安全以外的其他用途。

相關工作

識別傳送或接收結束程式中的 API 呼叫

應用程式層次安全 *(application level security)*

應用程式層次安全 是指在應用程式與其所連接的佇列管理程式之間的介面上呼叫的那些安全服務。

當應用程式對佇列管理程式發出 MQI 呼叫時，會呼叫這些服務。 應用程式、佇列管理程式、支援 IBM MQ 的另一個產品，或任何這些產品一起運作的組合，可能會直接或間接呼叫這些服務。 第 85 頁的圖 10 中說明應用程式層次安全。

應用程式層次安全也稱為 端對端安全 或 訊息層次安全。

以下是應用程式層次安全服務的一些範例:

- 當應用程式將訊息放入佇列時，訊息描述子會包含與應用程式相關聯的使用者 ID。 不過，沒有可用來鑑別使用者 ID 的資料 (例如已加密密碼)。 安全服務可以新增此資料。 當接收端應用程式最終擷取訊息時，服務的另一個元件可以使用隨訊息一起傳送的資料來鑑別使用者 ID。 這是識別及鑑別服務的範例。
- 當應用程式將訊息放在佇列時，訊息可以加密，當接收端應用程式擷取訊息時，訊息可以解密。這是機密性服務的範例。
- 當接收端應用程式擷取訊息時，可以檢查訊息。 此檢查會判定自傳送應用程式第一次將其放入佇列之後，是否刻意修改其內容。這是資料完整性服務的範例。

規劃 *Advanced Message Security*

Advanced Message Security (AMS) 是 IBM MQ 的元件，可為流經 IBM MQ 網路的機密資料提供高階保護，同時不會影響一般應用程式。

如果您要移動高度機密或有價值的資訊，特別是機密或付款相關資訊 (例如病患記錄或信用卡詳細資料) ，您必須特別注意資訊安全。 確保在企業周圍移動的資訊保持其完整性，並防止未獲授權的存取，是持續的挑戰和責任。 您也可能需要遵守安全法規，但不遵守可能受到懲罰。

您可以開發自己的 IBM MQ 安全延伸規格。 然而，這類解決方案需要專業技能，且維護可能複雜且昂貴。 Advanced Message Security 在幾乎每種類型的商業 IT 系統之間移動資訊時，可協助解決這些挑戰。

Advanced Message Security 以下列方式延伸 IBM MQ 的安全特性:

- 它使用訊息的加密或數位簽署，為您的點對點傳訊基礎架構提供應用程式層次的端對端資料保護。
- 它提供綜合性的安全保護，無需撰寫複雜的安全程式碼或修改或重新編譯現有的應用程式。
- 它使用「公開金鑰基礎架構 (PKI)」技術，為訊息提供鑑別、授權、機密性及資料完整性服務。
- 它提供大型主機及分散式伺服器的安全原則管理。
- 它同時支援 IBM MQ 伺服器和用戶端。
- 它與 Managed File Transfer 整合，以提供端對端安全傳訊解決方案。

如需相關資訊，請參閱第 498 頁的『Advanced Message Security』。

提供您自己的應用程式層次安全

您可以提供自己的應用程式層次安全服務。 為了協助您實作應用程式層次安全， IBM MQ 提供兩個結束程式 :API 結束程式和 API 交互結束程式。

API 結束程式和跨 API 結束程式可以提供識別和鑑別、存取控制、機密性、資料完整性和不可否認性服務，以及與安全無關的其他功能。

如果您的系統環境不支援 API 結束程式或 API 交互結束程式，您可以考量其他方式來提供您自己的應用程式層次安全。 一種方法是開發封裝 MQI 的更高層次 API。 然後程式設計師會使用此 API 而非 MQI 來撰寫 IBM MQ 應用程式。

使用較高層次 API 的最常見原因如下:

- 向程式設計師隱藏 MQI 的更進階特性。
- 在使用 MQI 時施行標準。
- 將函數新增至 MQI。 這項額外功能可以是安全服務。

部分供應商產品使用此技術來提供 IBM MQ 的應用程式層次安全。

如果您計劃以這種方式提供安全服務，請注意下列有關資料轉換的事項:

- 如果安全記號 (例如數位簽章) 已新增至訊息中的應用程式資料，則任何執行資料轉換的程式碼都必須知道此記號是否存在。
- 安全記號可能衍生自應用程式資料的二進位映像檔。 因此，在轉換資料之前，必須先完成記號的任何檢查。
- 如果訊息中的應用程式資料已加密，則必須在資料轉換之前將它解密。

通道結束程式

通道結束程式 是在 MCA 處理順序中的已定義位置呼叫的程式。 使用者和供應商可以撰寫自己的通道結束程式。 部分由 IBM 提供。

通道結束程式有數種類型，但只有四種具有提供鏈結層次安全的角色:

- 安全結束程式
- 訊息結束
- 傳送結束程式
- 接收結束

這四種類型的通道結束程式在 第 89 頁的圖 11 中有說明，並在下列主題中說明。

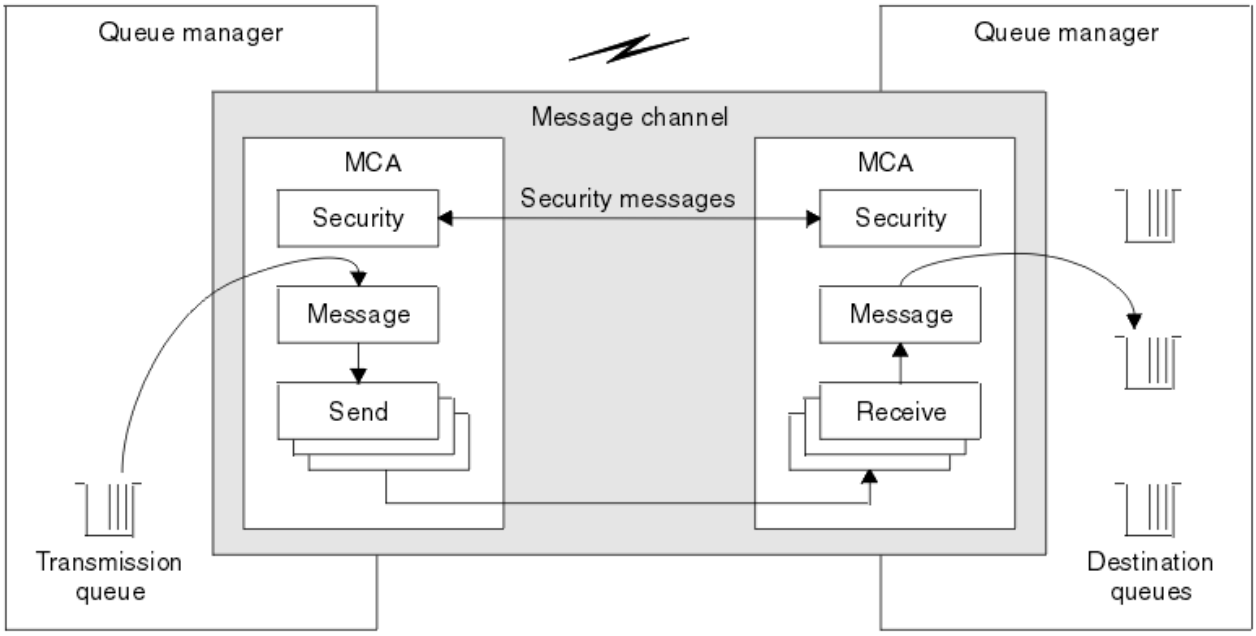


圖 *11:* 訊息通道上的安全、訊息、傳送及接收結束程式

相關概念

傳訊通道的通道結束程式

安全結束程式概觀

安全結束程式通常成對運作。 在訊息流程之前會呼叫它們，其目的是容許 MCA 鑑別其夥伴。

安全結束程式 通常成對運作; 通道兩端各有一個。 在通道啟動時完成起始資料協議之後，但在任何訊息開始流程之前，會立即呼叫它們。 安全結束程式的主要目的是在通道的每一端啟用 MCA ，以鑑別其夥伴。 不過，沒有任何項目可防止安全結束程式執行其他功能，即使功能與安全無關。

安全結束程式可以透過傳送 安全訊息來彼此通訊。 安全訊息的格式未定義，由使用者決定。 安全訊息交換的一個可能結果是其中一個安全結束程式可能決定不再繼續進行。 在該情況下，通道會關閉，且訊息不會流動。 如果通道的一端只有安全結束程式，則仍會呼叫該結束程式，並且可以選擇要繼續還是關閉通道。

可以在訊息及 MQI 通道上呼叫安全結束程式。 安全結束程式的名稱指定為通道每一端通道定義中的參數。

如需安全結束程式的相關資訊，請參閱 第 87 頁的『使用安全結束程式的鏈結層次安全』。

訊息結束

訊息結束程式僅在訊息通道上運作，且通常成對運作。 訊息結束程式可以對整個訊息進行操作，並對其進行各種變更。

通道傳送端和接收端的 訊息結束程式 通常成對運作。 在 MCA 從傳輸佇列取得訊息之後，會呼叫通道傳送端的訊息結束程式。 在通道的接收端，在 MCA 將訊息放入其目的地佇列之前，會先呼叫訊息結束程式。

訊息結束程式可以存取傳輸佇列標頭 MQXQH ，其中包括內嵌的訊息描述子，以及訊息中的應用程式資料。 訊息結束程式可以修改訊息的內容並變更其長度。 長度變更可能是壓縮、解壓縮、加密或解密訊息的結果。 它也可能是將資料新增至訊息或從中移除資料的結果。

訊息結束程式可以用於任何需要存取整個訊息的目的，而不是部分訊息，而且不一定要用於安全。

訊息結束程式可以判斷目前正在處理的訊息不應該進一步向其目的地前進。 然後 MCA 會將訊息放置在無法傳送的郵件佇列上。 訊息結束程式也可以關閉通道。

訊息結束程式只能在訊息通道上呼叫，而不能在 MQI 通道上呼叫。 這是因為 MQI 通道的目的是讓 MQI 呼叫的輸入及輸出參數在 IBM MQ MQI client 應用程式與佇列管理程式之間流動。

訊息結束程式的名稱指定為通道每一端的通道定義中的參數。 您也可以指定要連續執行的訊息結束程式清單。

如需訊息結束程式的相關資訊，請參閱 第 87 頁的『使用訊息結束程式的鏈結層次安全』。

傳送及接收結束程式

傳送和接收結束程式通常成對運作。 它們在傳輸區段上運作，且在它們正在處理的資料結構不相關的情況下最適合使用。

通道一端的 傳送結束程式 和另一端的 接收結束程式 通常成對運作。 在 MCA 發出通訊傳送以透過通訊連線傳送資料之前，會呼叫傳送結束程式。 接收結束程式會在 MCA 在通訊接收之後重新取得控制權，並從通訊連線接收資料之後呼叫。 如果正在使用共用交談，則會透過 MQI 通道，針對每一個交談呼叫傳送及接收結束程式的不同實例。

訊息通道上兩個 MCA 之間的 IBM MQ 通道通訊協定流程包含控制資訊及訊息資料。 同樣地，在 MQI 通道上，流程包含 MQI 呼叫的控制資訊及參數。 會針對所有類型的資料呼叫傳送及接收結束程式。

在訊息通道上，訊息資料只會在一個方向流動，但在 MQI 通道上， MQI 呼叫流程的輸入參數會在一個方向流動，而輸出參數則會在另一個方向流動。 在訊息及 MQI 通道上，控制雙向資訊流程。 因此，可以在通道兩端呼叫傳送及接收結束程式。

在兩個 MCA 之間的單一流程中傳輸的資料單元稱為 傳輸區段。 傳送及接收結束程式可以存取每一個傳輸區段。 他們可以修改其內容並變更其長度。 不過，傳送結束程式不得變更傳輸區段的前 8 個位元組。 這 8 個位元組構成 IBM MQ 通道通訊協定標頭的一部分。 傳送結束程式可以增加傳輸區段長度的程度也有一些限制。 尤其，傳送結束程式無法增加其長度，超出通道啟動時兩個 MCA 之間協議的長度上限。

在訊息通道上，如果訊息太大而無法在單一傳輸區段中傳送，則傳送端 MCA 會分割訊息，並在多個傳輸區段中傳送它。 因此，針對包含部分訊息的每個傳輸段呼叫傳送結束程式，並在接收端針對每個傳輸段呼叫接收結束程式。 接收 MCA 會在接收結束程式處理來自傳輸區段的訊息之後，重新建構來自傳輸區段的訊息。

同樣地，在 MQI 通道上，會在多個傳輸區段中傳送 MQI 呼叫的輸入或輸出參數 (如果它們太大)。 例如，如果應用程式資料足夠大，則可能會在 MQPUT、 MQPUT1 或 MQGET 呼叫上發生這種情況。

將這些考量納入考量，比較適合使用傳送及接收結束程式，因為它們不需要瞭解處理中資料的結構，因此可以將每一個傳輸區段視為二進位物件。

傳送或接收結束程式可以關閉通道。

傳送結束程式和接收結束程式的名稱指定為通道每一端的通道定義中的參數。 您也可以指定要連續執行的傳送結束程式清單。 同樣地，您可以指定接收結束程式清單。

如需傳送及接收結束程式的相關資訊，請參閱 第 87 頁的『使用傳送及接收結束程式的鏈結層次安全』。

規劃資料完整性

規劃如何保留資料的完整性。

您可以在應用程式層次或鏈結層次實作資料完整性。

在應用程式層次，如果標準機能不符合您的需求，您可以使用 API 結束程式。 您可以選擇使用 Advanced Message Security (AMS) 來數位簽署訊息，以防止未獲授權的修改。

在鏈結層次上，您可以選擇使用 TLS，在此情況下，您必須規劃數位憑證的使用。 如果標準機能無法滿足您的需求，您也可以使用通道結束程式。

相關概念

第 94 頁的『使用 SSL/TLS 保護通道』
IBM MQ 中的 TLS 支援使用佇列管理程式鑑別資訊物件及各種 MQSC 指令。 您也必須考量使用數位憑證。

第 19 頁的『IBM MQ 中的資料完整性』
您可以使用資料完整性服務來偵測訊息是否已修改。

第 88 頁的『規劃 Advanced Message Security』
Advanced Message Security (AMS) 是 IBM MQ 的元件，可為流經 IBM MQ 網路的機密資料提供高階保護，同時不會影響一般應用程式。

相關參考

API 結束程式參照

通道結束程式呼叫和資料結構

規劃審核

決定您需要審核哪些資料，以及您將如何擷取及處理審核資訊。 請考量如何檢查系統是否已正確配置。

活動監視有數個層面。 您必須考量的層面通常是由審核員需求所定義，而這些需求通常是由諸如 HIPAA (醫療保險轉移和責任法) 或 SOX (沙賓法案) 之類的法規標準所驅動。 IBM MQ 提供旨在協助符合這類標準的特性。

請考量您是否只對異常狀況感興趣，或您是否對所有系統行為感興趣。

審核的某些方面也可以視為作業監視; 審核的一個區別是您經常查看歷程資料，而不只是查看即時警示。 監視涵蓋在 監視及效能一節中。

要審核的資料

請考量您需要審核哪些類型的資料或活動，如下列各節所述:

使用 IBM MQ 介面對 IBM MQ 所做的變更
配置 IBM MQ 以發出檢測事件，特別是指令事件及配置事件。

在其控制之外對 IBM MQ 所做的變更
部分變更可能會影響 IBM MQ 的行為方式，但 IBM MQ 無法直接監視。 這類變更的範例包括 `mqs.ini`、 `qm.ini` 和 `mqclient.ini` 配置檔的變更、佇列管理程式的建立和刪除、二進位檔 (例如使用者結束程式) 的安裝，以及檔案許可權的變更。 若要監視這些活動，您必須使用在作業系統層次執行的工具。 不同的工具可用且適用於不同的作業系統。 您也可能具有由相關聯工具 (例如 *sudo*) 建立的日誌。

IBM MQ 的作業控制
您可能必須使用作業系統工具來審核活動，例如啟動及停止佇列管理程式。 在某些情況下， IBM MQ 可以配置成發出檢測事件。

IBM MQ 內的應用程式活動
若要審核應用程式的動作 (例如開啟佇列以及放置和取得訊息)，請配置 IBM MQ 以發出適當的事件。

侵入者警示
若要審核嘗試的安全侵害，請配置您的系統以發出授權事件。 頻道事件也可能有助於顯示活動，尤其是在頻道非預期地結束時。

規劃審核資料的擷取、顯示及保存

您需要的許多元素都會報告為 IBM MQ 事件訊息。 您必須選擇可讀取及格式化這些訊息的工具。 如果您對長期儲存體及分析感興趣，則必須將它們移至輔助儲存體機制 (例如資料庫)。 如果您不處理這些訊息，則它們會保留在事件佇列上，可能填滿佇列。 您可以決定實作工具，以根據部分事件自動採取動作; 例如，在發生安全失敗時發出警示。

驗證系統已正確配置

IBM MQ Explorer 隨附一組測試。 請使用這些來檢查物件定義是否有問題。

此外，請定期檢查系統配置是否如您預期。 雖然指令及配置事件可以在變更時報告，但傾出配置並將其與已知良好副本進行比較也很有用。

依拓蹼規劃安全

本節涵蓋特定狀況下的安全，即通道、佇列管理程式叢集、發佈/訂閱及多重播送應用程式，以及使用防火牆時。

如需相關資訊，請參閱下列子主題:

通道授權

當您透過通道傳送或接收訊息時，需要提供對各種 IBM MQ 資源的存取權。 「訊息通道代理程式 (MCA)」基本上是在佇列管理程式之間移動訊息的 IBM MQ 應用程式，因此需要存取各種 IBM MQ 資源才能正確運作。

若要在 MCA 的 PUT 時間接收訊息，您可以使用與 MCA 相關聯的使用者 ID，或與訊息相關聯的使用者 ID。

在 CONNECT 時，您可以使用 **CHLAUTH** 通道鑑別記錄，將主張的使用者 ID 對映至替代使用者。

在 IBM MQ 中，通道可以受到 TLS 支援的保護。

與傳送及接收通道相關聯的使用者 ID (不包括未使用 MCAUSER 屬性的傳送端通道) 需要存取下列資源:

- 與傳送端通道相關聯的使用者 ID 需要存取佇列管理程式、傳輸佇列、無法傳送郵件的佇列，以及存取通道結束程式所需的任何其他資源。
- 接收端通道的 MCAUSER 使用者 ID 需要 *+ setall* 權限。 原因是接收端通道必須使用它從遠端傳送端通道收到的資料來建立完整 MQMD，包括所有環境定義欄位。 因此，佇列管理程式需要執行此活動的使用者具有 *+ setall* 權限。 必須將此 *+ setall* 權限授與使用者:
  - 接收端通道有效放置訊息的所有佇列。
  - 佇列管理程式物件。 如需相關資訊，請參閱 環境定義授權。
- 發送端要求 COA 報告訊息之接收端通道的 MCAUSER 使用者 ID，在傳回報告訊息的傳輸佇列上需要 *+ passid* 權限。 如果沒有此權限，則會記載 AMQ8077 錯誤訊息。
- 使用與接收通道相關聯的使用者 ID，您可以開啟目標佇列，以將訊息放置在佇列上。 這涉及「訊息佇列作業介面 (MQI)」，因此如果您不是使用「IBM MQ 物件權限管理程式 (OAM)」，則可能需要進行其他存取控制檢查。 您可以指定是針對與 MCA 相關聯的使用者 ID 進行授權檢查 (如本主題所述)，還是針對與訊息相關聯的使用者 ID 進行授權檢查 (來自 MQMD UserIdentifier 欄位)。

  對於它所套用的通道類型，通道定義的 **PUTAUT** 參數會指定用於這些檢查的使用者 ID。
  - 通道預設為使用佇列管理程式的服務帳戶，其具有完整管理權限且不需要特殊授權。
  - 如果是伺服器連線通道，依預設， CHLAUTH 規則會封鎖管理連線，且需要明確供應。
  - 類型為接收端、要求端及叢集接收端的通道容許任何相鄰佇列管理程式進行本端管理，除非管理者採取步驟來限制此存取權。
  - 不需要授與 *dsp* 及 *ctrlx* 權限給接收端通道的 MCAUSER 使用者 ID。
  - 在 IBM MQ 8.0.0 Fix Pack 4 之前，如果您使用缺少 IBM MQ 管理專用權的使用者 ID，則必須將通道的 **dsp** 及 **ctrlx** 權限授與該使用者 ID，通道才能運作。

    從 IBM MQ 8.0.0 Fix Pack 4 開始，當通道重新同步化本身並更正序號時，不會進行任何權限檢查。

    不過，手動發出 RESET CHANNEL 指令仍需要所有版次的 **+dsp** 和 **+ctrlx**。

小心: 當訊息批次確認需要重設通道時， IBM MQ 會嘗試查詢通道，這需要 **+dsp** 權限。

- SDR 通道類型的 MCAUSER 屬性未使用。
- 如果您使用與訊息相關聯的使用者 ID，則該使用者 ID 可能來自遠端系統。 此遠端系統使用者 ID 必須由目標系統辨識。 下列指令是您可以發出以從遠端系統授與使用者 ID 權限之指令類型的範例:

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

```
setmqaut -m QMgrName -t chl -n Profile -g GroupName +dsp +ctrlx
```

其中 設定檔 是通道。

```
setmqaut -m QMgrName -t q -n Profile -g GroupName +put +setall
```

其中 設定檔 是無法傳送郵件的佇列 (如果已設定的話)。

```
setmqaut -m QMgrName -t q -n Profile -g GroupName +put +setall
```

其中 設定檔 是授權佇列的清單。

小心: 授權使用者 ID 將訊息放入「指令佇列」或其他機密系統佇列時，請小心。

與 MCA 相關聯的使用者 ID 取決於 MCA 類型。 MCA 有兩種類型:

呼叫程式 MCA
起始通道的 MCA。 呼叫程式 MCA 可以作為個別處理程序、通道起始程式的執行緒或處理程序儲存區的執行緒來啟動。 所使用的使用者 ID 是與母項處理程序 (通道起始程式) 相關聯的使用者 ID，或與啟動 MCA 的處理程序相關聯的使用者 ID。

回應者 MCA
回應者 MCA 是由於呼叫者 MCA 要求而啟動的 MCA。 回應者 MCA 可以作為個別處理程序、接聽器的執行緒或處理程序儲存區的執行緒來啟動。 使用者 ID 可以是下列任何一種類型 (依這個喜好設定順序):

1. 在 APPC 上，呼叫者 MCA 可以指出要用於回應者 MCA 的使用者 ID。 這稱為網路使用者 ID，且僅適用於以個別處理程序啟動的通道。 使用通道定義的 USERID 參數來設定網路使用者 ID。
2. 如果未使用 **USERID** 參數，則回應者 MCA 的通道定義可以指定 MCA 必須使用的使用者 ID。 使用通道定義的 **MCAUSER** 參數來設定使用者 ID。
3. 如果上述 (兩種) 方法都未設定使用者 ID，則會使用啟動 MCA 之程序的使用者 ID 或母項程序 (接聽器) 的使用者 ID。

相關概念
第 40 頁的『通道鑑別記錄』
若要在通道層次對授與連接系統的存取權進行更精確的控制，您可以使用通道鑑別記錄。

相關參考
通道鑑別記錄內容

保護通道起始程式定義
只有 mqm 群組的成員才能操作通道起始程式。

IBM MQ 通道起始程式不是 IBM MQ 物件; 對它們的存取權不受 OAM 控制。 IBM MQ 不容許使用者或應用程式操作這些物件，除非其使用者 ID 是 mqm 群組的成員。 如果您有應用程式發出 PCF 指令 **StartChannelInitiator**，則在 PCF 訊息的訊息描述子中指定的使用者 ID 必須是目標佇列管理程式上 mqm 群組的成員。

使用者 ID 也必須是目標機器上 mqm 群組的成員，才能透過 Escape PCF 指令或以間接模式使用 `runmqsc` 來發出對等 MQSC 指令。

傳輸佇列
佇列管理程式會自動將遠端訊息放置在傳輸佇列上; 這不需要特殊權限。

不過，如果您需要將訊息直接放置在傳輸佇列上，這需要特殊授權; 請參閱 第 108 頁的表 12。

通道結束程式

如果通道鑑別記錄不適用，您可以使用通道結束程式來增加安全。 安全結束程式會在兩個安全結束程式之間形成安全連線。 一個程式用於傳送訊息通道代理程式 (MCA)，另一個程式用於接收 MCA。

如需通道結束程式的相關資訊，請參閱 第 89 頁的『通道結束程式』。

使用 *SSL/TLS* 保護通道

IBM MQ 中的 TLS 支援使用佇列管理程式鑑別資訊物件及各種 MQSC 指令。 您也必須考量使用數位憑證。

數位憑證和金鑰儲存庫

最好設定佇列管理程式憑證標籤屬性 (**CERTLABL**) 要用於大部分通道的個人憑證名稱，並透過在那些需要不同憑證的通道上設定憑證標籤來置換該憑證以用於異常狀況。

如果您需要許多通道，其憑證與佇列管理程式上設定的預設憑證不同，您應該考慮在數個佇列管理程式之間劃分通道，或使用佇列管理程式前面的 MQIPT Proxy 來呈現不同的憑證。

您可以對每個通道使用不同的憑證，但如果您在金鑰儲存庫中儲存太多憑證，則在啟動 TLS 通道時可能會預期效能受到影響。 嘗試將金鑰儲存庫中的憑證數目保持小於大約 50，並將 100 視為最大數目，因為較大的金鑰儲存庫會大幅降低 IBM Global Security Kit (GSKit) 效能。

容許在相同佇列管理程式上使用多個憑證會增加在相同佇列管理程式上使用多個 CA 憑證的機會。 這會增加個別憑證管理中心所發出憑證的憑證「主旨識別名稱」名稱空間衝突的機率。

雖然專業憑證管理中心可能會更小心，但內部憑證管理中心通常缺乏明確的命名慣例，最後您可能會在一個 CA 與另一個 CA 之間產生非預期的相符項。

除了「主旨識別名稱」之外，您還應該檢查憑證發證者識別名稱。 如果要這麼做，請使用通道鑑別 SSLPEERMAP 記錄，並同時設定 **SSLPEER** 和 **SSLCERTI** 欄位，以分別符合「主體 DN」和「發證者 DN」。

自簽憑證和 CA 簽章憑證

在開發及測試應用程式時，以及在正式作業中使用應用程式時，請務必規劃數位憑證的使用。 您可以使用 CA 簽章憑證或自簽憑證，視佇列管理程式及用戶端應用程式的使用情形而定。

CA 簽章憑證
若為正式作業系統，請從授信憑證管理中心 (CA) 取得您的憑證。 當您從外部 CA 取得憑證時，您會支付服務的費用。

自簽憑證
在開發應用程式時，您可以使用自簽憑證或本端 CA 發出的憑證，視平台而定:

ALW 在 AIX, Linux, and Windows 系統上，您可以使用自簽憑證。 請參閱第 246 頁的『在 AIX, Linux, and Windows 上建立自簽個人憑證』以取得相關指示。

IBM i 在 IBM i 系統上，您可以使用本端 CA 所簽署的憑證。 請參閱第 234 頁的『在 IBM i 上要求伺服器憑證』以取得相關指示。

z/OS 在 z/OS 上，您可以使用自簽或本端 CA 簽章憑證。 如需指示，請參閱 第 269 頁的『在 z/OS 上建立自簽個人憑證』 或 第 270 頁的『在 z/OS 上要求個人憑證』。

由於下列原因，自簽憑證不適合正式作業使用:

- 無法撤銷自簽憑證，這可能容許攻擊者在私密金鑰受損之後盜用身分。 CA 可以撤銷已受損憑證，這會阻止其進一步使用。 因此，在正式作業環境中使用 CA 簽章憑證更安全，雖然自簽憑證對測試系統更方便。
- 自簽憑證永不到期。 在測試環境中，這既方便又安全，但在正式作業環境中，它會讓它們對最終安全侵害保持開放。 由於無法撤銷自簽憑證，因此風險更加嚴重。
- 自簽憑證同時用作個人憑證及主要 (或信任錨點) CA 憑證。 具有自簽個人憑證的使用者可能可以使用它來簽署其他個人憑證。 一般而言，這並不是由 CA 發出的個人憑證，而是大量曝光率。

CipherSpecs 和數位憑證

只有一部分受支援的 CipherSpecs 可以與所有受支援的數位憑證類型搭配使用。 因此，必須為您的數位憑證選擇適當的 CipherSpec 。 同樣地，如果您組織的安全原則要求使用特定的 CipherSpec ，則您必須取得適當的數位憑證。

如需 CipherSpecs 與數位憑證之間關係的相關資訊，請參閱 第 37 頁的『IBM MQ 中的數位憑證及 CipherSpec 相容性』

憑證驗證原則

IETF RFC 5280 標準指定一系列憑證驗證規則，符合標準的應用軟體必須實作這些規則，才能防止假冒攻擊。 一組憑證驗證規則稱為憑證驗證原則。 如需 IBM MQ 中憑證驗證原則的相關資訊，請參閱 第 36 頁的『IBM MQ 中的憑證驗證原則』。

規劃憑證撤銷檢查

容許來自不同憑證管理中心的多個憑證可能導致不必要的其他憑證撤銷檢查。

尤其，如果您已明確配置使用來自特定 CA 的撤銷伺服器 (例如使用 AUTHINFO 物件或鑑別資訊記錄 (MQAIR) 結構) ，則當呈現來自不同 CA 的憑證時，撤銷檢查會失敗。

您應該避免明確的憑證撤銷伺服器配置。 相反地，您應該在憑證延伸 (例如，「CRL 配送點」或 OCSP AuthorityInfo 存取) 中啟用隱含檢查，其中每一個憑證都包含自己的撤銷伺服器位置。

如需相關資訊，請參閱 OCSPCheckExtensions 和 CDPCheckExtensions。

TLS 支援的指令及屬性

「傳輸層安全 (TLS)」通訊協定提供通道安全，可防止竊聽、竄改及模擬。 IBM MQ 支援 TLS 可讓您在通道定義上指定特定通道使用 TLS 安全。 您也可以指定所需安全類型的詳細資料，例如您要使用的加密演算法。

- 下列 MQSC 指令支援 TLS:

  ALTER AUTHINFO
  修改鑑別資訊物件的屬性。

  DEFINE AUTHINFO
  建立鑑別資訊物件。

  DELETE AUTHINFO
  刪除鑑別資訊物件。

  DISPLAY AUTHINFO
  顯示特定鑑別資訊物件的屬性。

- 下列佇列管理程式參數支援 TLS:

  CERTLABL
  定義要使用的個人憑證標籤。

  SSLCRLNL
  SSLCRLNL 屬性指定鑑別資訊物件的名單，用來提供憑證撤銷位置以容許加強 TLS 憑證檢查。

  SSLCRYP
  在 AIX, Linux, and Windows 系統上，設定 **SSLCryptoHardware** 佇列管理程式屬性。 此屬性是參數字串的名稱，可用來配置系統上的加密硬體。

  SSLEV
  判定如果使用 TLS 的通道無法建立 TLS 連線，是否報告 TLS 事件訊息。

  SSLFIPS
  指定在 IBM MQ 而非加密硬體中執行加密法時，是否只使用 FIPS 認證的演算法。 如果已配置加密硬體，則會使用硬體產品所提供的加密模組，且這些模組可能經過 FIPS 認證達到特定層次。 這取決於使用中的硬體產品。

SSLKEYR
在 AIX, Linux, and Windows 系統上，將金鑰儲存庫與佇列管理程式相關聯。 金鑰資料庫保留在 *GSKit* 金鑰資料庫中。 GSKit 可讓您在 AIX, Linux, and Windows 系統上使用 TLS 安全。

SSLRKEYC
在重新協議秘密金鑰之前，在 TLS 交談中要傳送及接收的位元組數。 位元組數包括 MCA 所傳送的控制資訊。

- 下列通道參數支援 TLS:

CERTLABL
定義要使用的個人憑證標籤。

SSLCAUTH
定義 IBM MQ 是否需要並驗證來自 TLS 用戶端的憑證。

SSLCIPH
指定加密強度和功能 (CipherSpec) ，例如 TLS_RSA_WITH_AES_128_CBC_SHA。 通道兩端的 CipherSpec 必須相符。

SSLPEER
指定容許夥伴的識別名稱 (唯一 ID)。

本節說明 **setmqaut**、 **dspmqaut**、 **dmpmqaut**、 **rcrmqobj**、 **rcdmqimg** 及 **dspmqfls** 指令，以支援鑑別資訊物件。 它也說明 **runmqckm** (iKeycmd) 及 **runmqakm** 指令，用於在 AIX, Linux, and Windows 上管理憑證。 請參閱下列小節:

- setmqaut
- dspmqaut
- dmpmqaut
- rcrmqobj
- rcdmqimg
- dspmqfls
- 管理金鑰和憑證

如需使用 TLS 的通道安全概觀，請參閱

- 第 20 頁的『IBM MQ 中的 TLS 安全通訊協定』

如需與 TLS 相關聯之 MQSC 指令的詳細資料，請參閱

- ALTER AUTHINFO
- DEFINE AUTHINFO
- DELETE AUTHINFO
- DISPLAY AUTHINFO

如需與 TLS 相關聯之 PCF 指令的詳細資料，請參閱

- 變更、複製及建立鑑別資訊物件
- 刪除鑑別資訊物件
- 查詢鑑別資訊物件

z/OS *IBM MQ for z/OS* 伺服器連線通道

如果沒有實作通道鑑別或使用 TLS 新增安全結束程式，則 IBM MQ for z/OS SVRCONN 通道不安全。 依預設， SVRCONN 通道沒有定義安全結束程式。

安全考量

SVRCONN 通道並不如起始定義的 SYSTEM.DEF.SVRCONN 。 若要保護 SVRCONN 通道安全，您必須使用 SET CHLAUTH 指令來設定通道鑑別，或安裝安全結束程式並實作 TLS。

您必須使用公開可用的範例安全結束程式、自行撰寫安全結束程式，或購買安全結束程式。

您可以使用數個範例作為撰寫您自己的 SVRCONN 通道安全結束程式的良好起點。

在 IBM MQ for z/OS 中， hlq.SCSQC37S 程式庫中的成員 CSQ4BCX3 是以 C 語言撰寫的安全結束程式範例。 範例 CSQ4BCX3 也會在 hlq.SCSQAUTH 程式庫中預先編譯。

您可以將已編譯成員 hlq.SCSQAUTH(CSQ4BCX3) 複製到 CHIN Proc 中配置給 CSQXLIB DD 的載入程式庫，以實作 CSQ4BCX3 範例結束程式。 請注意， CHIN 需要將載入程式庫設為「程式控制」。

變更 SVRCONN 通道以將 CSQ4BCX3 設為安全結束程式。

V 9.2.0 當用戶端使用該 SVRCONN 通道連接時， CSQ4BCX3 會使用 MQCD 中的 **RemoteUserIdentifier** 及 **RemotePassword** 配對進行鑑別，或從 IBM MQ for z/OS 9.1.4 中，使用 MQCSP 中的 **CSPUserIdPtr** 及 **CSPPasswordPtr** 配對進行鑑別。 如果鑑別成功，它會將 **RemoteUserIdentifier** 複製到 **MCAUserIdentifier**，並變更執行緒的身分環境定義。

若為 Long Term Support 及 IBM MQ for z/OS 9.1.4 之前的 Continuous Delivery ，當用戶端使用該 SVRCONN 通道連接時， CSQ4BCX3 會使用 MQCD 中的 **RemoteUserIdentifier** 及 **RemotePassword** 配對進行鑑別。 如果鑑別成功，它會將 **RemoteUserIdentifier** 複製到 **MCAUserIdentifier**，並變更執行緒的身分環境定義。

如果您要撰寫 IBM MQ Java 用戶端，則可以使用蹦現畫面來查詢使用者，並設定 MQEnvironment.userID 和 MQEnvironment.password。 當建立連線時，會傳遞這些值。

現在您有了功能安全結束程式，另外還有一個問題，就是在建立連線時，使用者 ID 和密碼會以純文字在網路中傳輸，如同任何後續 IBM MQ 訊息的內容一樣。 您可以使用 TLS 來加密此起始連線資訊以及任何 IBM MQ 訊息的內容。

範例

保護 IBM MQ Explorer SVRCONN 通道 SYSTEM.ADMIN.SVRCONN 完成下列步驟:

1. 將 hlq.SCSQAUTH(CSQ4BCX3) 複製到配置給 CHINIT Proc 中 CSQXLIB DD 的載入程式庫。
2. 請驗證載入程式庫是否為「程式控制」。
3. 變更 SYSTEM ADMIN.SVRCONN 以使用安全結束程式 CSQ4BCX3。
4. 在 IBM MQ Explorer 中，用滑鼠右鍵按一下 z/OS 佇列管理程式名稱，選取 **連線詳細資料** > **內容** > **使用者 ID** ，然後輸入您的 z/OS 使用者 ID。
5. 輸入密碼，以連接至「 z/OS 佇列管理程式」。

其他資訊

若要在「程式控制」環境中執行結束程式 CSQ4BCX3 ，必須從「程式控制」程式庫載入所有載入至 CHIN 位址空間的項目，例如， STEPLIB 中的所有程式庫，以及 CSQXLIB DD 上的任何程式庫。 若要將載入程式庫設為「程式控制」，請發出 RACF 指令。 在下列範例中，載入程式庫名稱是 MY.TEST.LOADLIB。

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

若要變更 SVRCONN 通道來實作 CSQ4BCX3，請發出下列 IBM MQ 指令:

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

在上述範例中，所使用的 SVRCONN 通道名稱是 SYSTEM ADMIN.SVRCONN。

如需通道結束程式的相關資訊，請參閱 第 89 頁的『通道結束程式』 。

相關工作

在 z/OS 上撰寫通道結束程式

SNA LU 6.2 安全服務

SNA LU 6.2 提供階段作業層次加密法、階段作業層次鑑別及交談層次鑑別。

註: 此主題集合假設您對「系統網路架構 (SNA)」有基本瞭解。 本節提及的其他文件包含相關概念和術語的簡要介紹。 如果您需要更完整的 SNA 技術簡介，請參閱 *Systems Network Architecture Technical Overview*(GC30-3073)。

SNA LU 6.2 提供三種安全服務:

- 階段作業層次加密法
- 階段作業層次鑑別
- 交談層次鑑別

對於階段作業層次加密法和階段作業層次鑑別， SNA 會使用 資料加密標準 *(DES)* 演算法。 DES 演算法是一種區塊密碼演算法，使用對稱金鑰來加密及解密資料。 區塊及索引鍵的長度都是 8 個位元組。

階段作業層次加密法

階段作業層次加密法 會使用 DES 演算法來加密及解密階段作業資料。 因此，它可以用來在 SNA LU 6.2 通道上提供鏈結層次機密性服務。

邏輯單元 (LU) 可以提供必要 (或必要) 資料加密法、選擇性資料加密法或無資料加密法。

在 強制加密階段作業上， LU 會加密所有出埠資料要求單元，並解密所有入埠資料要求單元。

在 選擇性加密階段作業上， LU 只會加密傳送交易程式 (TP) 指定的資料要求單元。 傳送 LU 透過在要求標頭中設定指示器來發出資料已加密的信號。 透過檢查此指示器，接收端 LU 可以在將哪些要求單元傳遞給接收端 TP 之前，先告知要解密哪些要求單元。

在 SNA 網路中， IBM MQ MCA 是交易程式。 MCA 不會對它們傳送的任何資料要求加密。 因此，選擇性資料加密法不是一個選項; 在階段作業上只能使用強制資料加密法或沒有資料加密法。

如需如何實作必要資料加密法的相關資訊，請參閱 SNA 子系統的文件。 如需可在平台上使用的更強加密形式 (例如 z/OS 上的三重 DES 演算法 24 位元組加密) 的相關資訊，請參閱相同文件。

如需階段作業層次加密法的一般相關資訊，請參閱 *Systems Network Architecture LU 6.2 Reference: Peer Protocols*， SC31-6808。

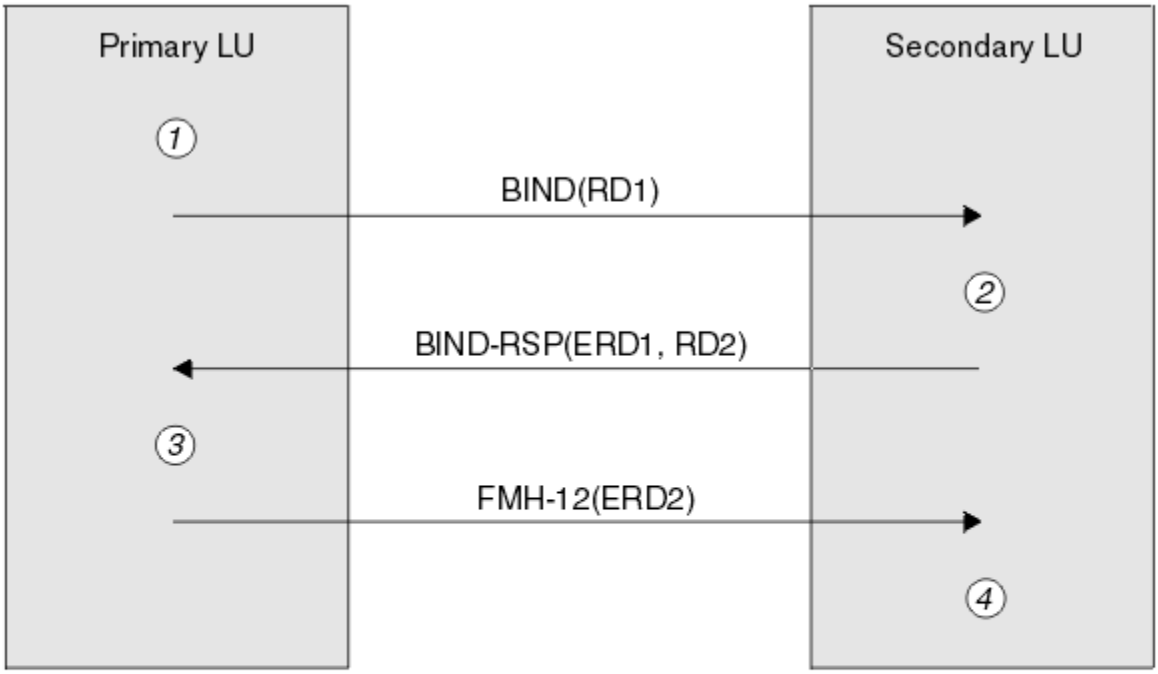
階段作業層次鑑別

階段作業層次鑑別 是階段作業層次安全通訊協定，可讓兩個 LU 在啟動階段作業時彼此鑑別。 它也稱為 *LU-LU* 驗證。

因為 LU 實際上是從網路進入系統的 "閘道" ，所以在某些情況下您可能會認為此鑑別層次已足夠。 例如，如果您的佇列管理程式需要與在受控制且授信環境中執行的遠端佇列管理程式交換訊息，則在鑑別 LU 之後，您可能準備好信任遠端系統其餘元件的身分。

階段作業層次鑑別是由每一個 LU 驗證其友機的密碼來達成。 此密碼稱為 *LU-LU* 密碼 ，因為在每對 LU 之間建立一個密碼。 建立 LU-LU 密碼的方式視實作而定，且不在 SNA 範圍內。

第 99 頁的圖 12 說明階段作業層次鑑別的流程。



Legend:

BIND	=	BIND request unit
BIND-RSP	=	BIND response unit
ERD	=	Encrypted random data
FMH-12	=	Function Management Header 12
RD	=	Random data

圖 *12:* 階段作業層次鑑別的流程

階段作業層次鑑別的通訊協定如下。 程序中的數字對應於 第 99 頁的圖 12 中的數字。

1. 主要 LU 會產生隨機資料值 (RD1)，並將它傳送至 BIND 要求中的次要 LU。
2. 當次要 LU 接收具有隨機資料的 BIND 要求時，它會使用 DES 演算法來加密資料，並以其 LU-LU 密碼副本作為金鑰。 然後，次要 LU 會產生第二個隨機資料值 (RD2)，並將其與加密資料 (ERD1) 一起傳送至 BIND 回應中的主要 LU。
3. 當主要 LU 收到 BIND 回應時，它會從它最初產生的隨機資料計算它自己的加密資料版本。 其作法是使用 DES 演算法，並以其 LU-LU 密碼副本作為金鑰。 然後，它會將其版本與 BIND 回應中收到的加密資料進行比較。 如果這兩個值相同，則主要 LU 知道次要 LU 具有與其相同的密碼，且會鑑別次要 LU。 如果這兩個值不相符，則主要 LU 會終止階段作業。

   然後，主要 LU 會加密它在 BIND 回應中收到的隨機資料，並將加密資料 (ERD2) 傳送至功能管理標頭 12 (FMH-12) 中的次要 LU。
4. 當次要 LU 接收 FMH-12 時，它會從它所產生的隨機資料計算它自己的加密資料版本。 然後，它會比較其版本與在 FMH-12 中收到的加密資料。 如果這兩個值相同，則會鑑別主要 LU。 如果這兩個值不相符，則次要 LU 會終止階段作業。

在加強版的通訊協定中 (在中間攻擊中提供更好的保護)，次要 LU 會使用其 LU-LU 密碼副本作為金鑰，計算來自 RD1、 RD2 的「DES 訊息鑑別碼 (MAC)」，以及次要 LU 的完整名稱。 次要 LU 會將 MAC 傳送至 BIND 回應中的主要 LU，而不是 ERD1。

主要 LU 透過計算其自己的 MAC 版本 (與 BIND 回應中接收的 MAC 相比較) 來鑑別次要 LU。 然後，主要 LU 會從 RD1 和 RD2 計算第二個 MAC，並將 MAC 傳送至 FMH-12 中的次要 LU，而不是 ERD2。

次要 LU 透過計算其自己版本的第二個 MAC (與 FMH-12 中接收的 MAC 相比較) 來鑑別主要 LU。

如需如何配置階段作業層次鑑別的相關資訊，請參閱 SNA 子系統的文件。 如需階段作業層次鑑別的一般資訊，請參閱 *Systems Network Architecture LU 6.2 Reference: Peer Protocols*， SC31-6808。

交談層次鑑別
當本端 TP 嘗試配置與友機 TP 的交談時，本端 LU 會將連接要求傳送至友機 LU，要求它連接友機 TP。 在特定情況下，連接要求可以包含安全資訊，友機 LU 可以用來鑑別本端 TP。 這稱為 交談層次鑑別或 一般使用者驗證。

下列主題說明 IBM MQ 如何提供交談層次鑑別的支援。

如需交談層次鑑別的相關資訊，請參閱 *Systems Network Architecture LU 6.2 參考:* 同層級通訊協定 (SC31-6808)。 如需 z/OS 特定的資訊，請參閱 *z/OS MVS Planning: APPC/MVS Management*(SA22-7599)。

如需 CPI-C 的相關資訊，請參閱 共用程式設計介面通訊 *CPI-C* 規格， SC31-6180。 如需 APPC/MVS TP Conversation Callable Services 的相關資訊，請參閱 *z/OS MVS Programming: Writing Transaction Programs for APPC/MVS*(SA22-7621)。

Multi *Multiplatforms* 上的交談層次鑑別支援
請利用這個主題來取得如何在 Multiplatforms 上進行交談層次鑑別的概觀。

第 100 頁的圖 13 說明 Multiplatforms 上的交談層次鑑別支援。 圖表中的數字對應於下列說明中的數字。

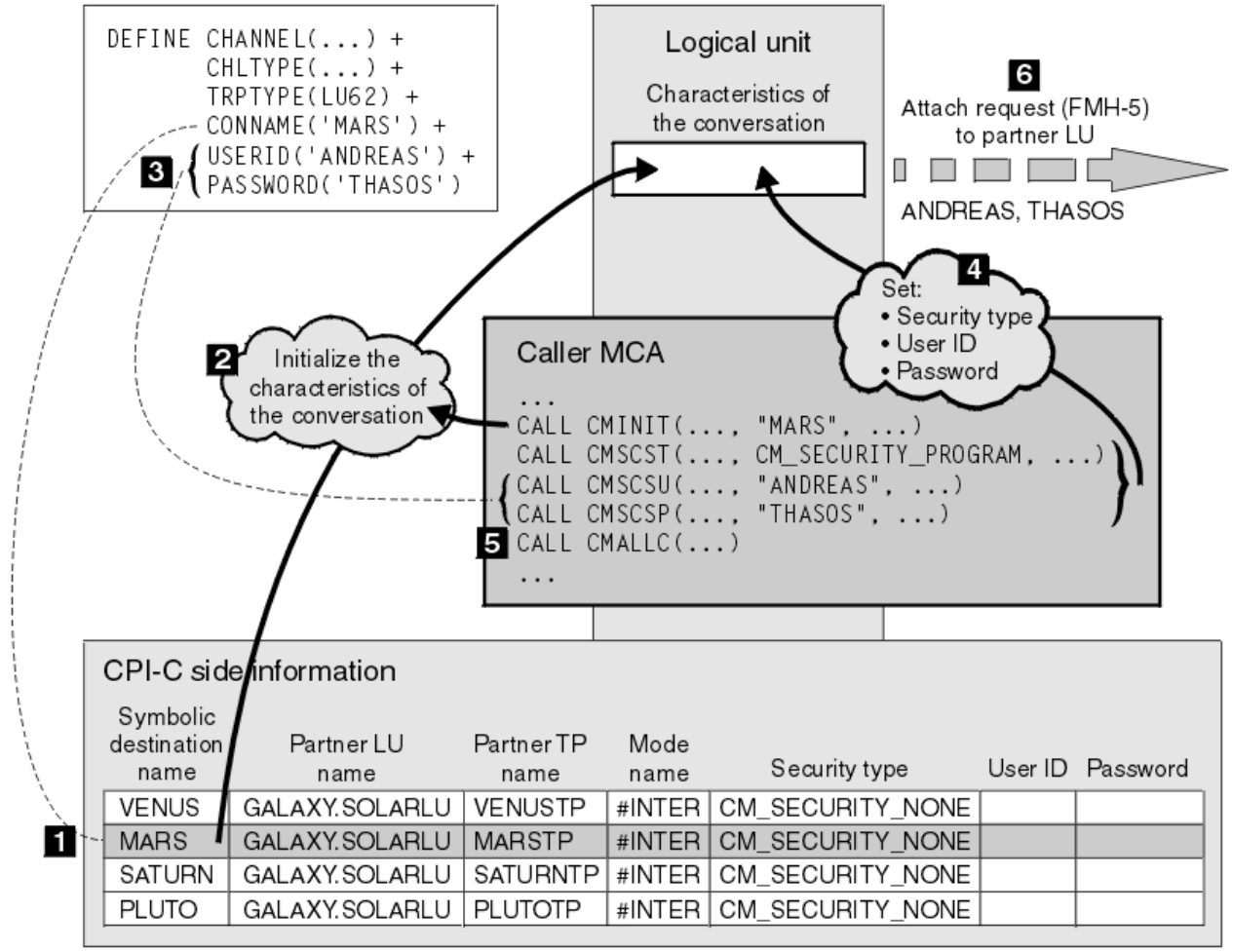


圖 *13: IBM MQ* 支援交談層次鑑別

在 Multiplatforms 上， MCA 會使用「共用程式設計介面通訊 (CPI-C)」呼叫，透過 SNA 網路與友機 MCA 進行通訊。 在通道呼叫端的通道定義中， CONNAME 參數的值是符號式目的地名稱，可識別 CPI-C 端資訊項目 (1)。 此項目指定:

- 友機 LU 的名稱

- 夥伴 TP 的名稱，這是回應者 MCA
- 要用於交談的模式名稱

側邊資訊項目也可以指定下列安全資訊:

- 安全類型。

  通常實作的安全類型為 CM_SECURITY_NONE、CM_SECURITY_PROGRAM 及 CM_SECURITY_SAME，但其他則定義在 CPI-C 規格中。
- 使用者 ID。
- 密碼。

呼叫者 MCA 準備透過發出 CPI-C 呼叫 CMINIT，並使用 CONNAME 值作為呼叫上的其中一個參數，來配置與回應者 MCA 的交談。 為了本端 LU 的好處， CMINIT 呼叫會識別 MCA 要用於交談的週邊資訊項目。 本端 LU 使用此登錄中的值來起始設定交談的性質 (2)。

然後，呼叫端 MCA 會檢查通道定義 (3) 中 USERID 及 PASSWORD 參數的值。 如果設定 USERID，則呼叫者 MCA 會發出下列 CPI-C 呼叫 (4):

- CMSCST，將交談的安全類型設為 CM_SECURITY_PROGRAM。
- CMSCSU，將交談的使用者 ID 設為 USERID 值。
- CMSCSP，將交談的密碼設為 PASSWORD 值。 除非設定 PASSWORD，否則不會呼叫 CMSCSP。

這些呼叫所設定的安全類型、使用者 ID 及密碼會置換先前從側邊資訊項目獲得的任何值。

然後，呼叫者 MCA 發出 CPI-C 呼叫 CMALLC 以配置交談 (5)。 為了回應此呼叫，本端 LU 將連接要求 (函數管理標頭 5 或 FMH-5) 傳送至友機 LU (6)。

如果友機 LU 將接受使用者 ID 及密碼，則附加要求中會包含 USERID 及 PASSWORD 值。 如果友機 LU 不接受使用者 ID 及密碼，則附加要求中不會包含這些值。 當 LU 連結以形成階段作業時，本端 LU 會探索友機 LU 是否接受使用者 ID 及密碼作為資訊交換的一部分。

在更新版本的附加要求中，密碼替代可以在 LU 之間流動，而不是清除密碼。 密碼替代是由密碼組成的「DES 訊息鑑別碼 (MAC)」或 SHA-1 訊息摘要。 只有在兩個 LU 都支援時，才能使用密碼替代。

當夥伴 LU 收到包含使用者 ID 和密碼的送入連接要求時，它可能會使用使用者 ID 和密碼來進行識別和鑑別。 透過參照存取控制清單，友機 LU 也可以判定使用者 ID 是否具有配置交談及連接回應者 MCA 的權限。

此外，回應者 MCA 可能以附加要求中包含的使用者 ID 來執行。 在此情況下，使用者 ID 會變成回應者 MCA 的預設使用者 ID，並在 MCA 嘗試連接至佇列管理程式時用於權限檢查。 當 MCA 嘗試存取佇列管理程式的資源時，也可以使用它來後續進行權限檢查。

連接要求中的使用者 ID 和密碼可用於識別、鑑別和存取控制的方式取決於實作。 如需 SNA 子系統特定的資訊，請參閱適當的文件。

如果未設定 USERID，則呼叫者 MCA 不會呼叫 CMSCST、CMSCSU 及 CMSCSP。 在此情況下，連接要求中流動的安全資訊僅由週邊資訊項目中指定的內容及友機 LU 將接受的內容來決定。

交談層次鑑別及 *IBM MQ for z/OS*

在 z/OS 上，請利用這個主題來取得交談層次鑑別如何運作的概觀。

在 IBM MQ for z/OS 上， MCA 不使用 CPI-C。 相反地，他們使用 APPC/MVS TP Conversation Callable Services，這是進階程式對程式通訊 (APPC) 的實作，具有部分 CPI-C 特性。 當呼叫端 MCA 配置交談時，會在通話中指定 SAME 安全類型。 因此，因為 APPC/MVS LU 只支援入埠交談的持續性驗證，而不支援出埠交談，所以有兩種可能性:

- 如果夥伴 LU 信任 APPC/MVS LU 並將接受已驗證的使用者 ID，則 APPC/MVS LU 會傳送包含下列內容的連接要求:
  - 通道起始程式位址空間使用者 ID
  - 安全設定檔名稱，如果使用 RACF，則為通道起始程式位址空間使用者 ID 的現行連接群組名稱
  - 已驗證指示器
- 如果友機 LU 不信任 APPC/MVS LU，且不接受已驗證的使用者 ID，則 APPC/MVS LU 會傳送不含安全資訊的連接要求。

在 IBM MQ for z/OS 上， DEFINE CHANNEL 指令上的 USERID 及 PASSWORD 參數無法用於訊息通道，且只在 MQI 通道的用戶端連線端才有效。 因此，來自 APPC/MVS LU 的連接要求絕不會包含這些參數指定的值。

佇列管理程式叢集的安全

雖然佇列管理程式叢集可以方便使用，但您必須特別注意其安全。

佇列管理程式叢集 是邏輯上以某種方式關聯的佇列管理程式網路。 屬於叢集成員的佇列管理程式稱為 叢集佇列管理程式。

叢集中的其他佇列管理程式可以知道屬於叢集佇列管理程式的佇列。 這類佇列稱為 叢集佇列。 叢集中的任何佇列管理程式都可以將訊息傳送至叢集佇列，而不需要下列任何一項:

- 每一個叢集佇列的明確遠端佇列定義
- 明確定義與每一個遠端佇列管理程式之間的通道
- 每一個出埠通道的個別傳輸佇列

您可以建立一個叢集，其中複製兩個以上佇列管理程式。 這表示它們具有相同本端佇列的實例，包括宣告為叢集佇列的任何本端佇列，並且可以支援相同伺服器應用程式的實例。

當連接至叢集佇列管理程式的應用程式將訊息傳送至在每一個複製的佇列管理程式上具有實例的叢集佇列時， IBM MQ 會決定要將它傳送至哪個佇列管理程式。 當許多應用程式將訊息傳送至叢集佇列時， IBM MQ 會在具有佇列實例的每一個佇列管理程式之間平衡工作量。 如果其中一個管理所複製佇列管理程式的系統失敗， IBM MQ 會繼續平衡其餘佇列管理程式之間的工作量，直到重新啟動失敗的系統為止。

如果您使用佇列管理程式叢集，則需要考量下列安全問題:

- 只容許選取的佇列管理程式將訊息傳送至佇列管理程式
- 只容許選取的遠端佇列管理程式使用者將訊息傳送至佇列管理程式上的佇列
- 容許連接至佇列管理程式的應用程式只將訊息傳送至選取的遠端佇列

即使您沒有使用叢集，這些考量也會相關，但如果您使用叢集，它們會變得更重要。

如果應用程式可以將訊息傳送至一個叢集佇列，則它可以將訊息傳送至任何其他叢集佇列，而不需要其他遠端佇列定義、傳輸佇列或通道。 因此，請考量是否需要限制存取佇列管理程式上的叢集佇列，以及限制應用程式可以傳送訊息的叢集佇列。

有一些其他安全考量，只有在您使用佇列管理程式叢集時才相關:

- 只容許選取的佇列管理程式加入叢集
- 強制不要的佇列管理程式離開叢集

如需所有這些考量的相關資訊，請參閱 保持叢集安全。 z/OS 如需 IBM MQ for z/OS 特定的考量，請參閱 第 222 頁的『z/OS 上佇列管理程式叢集中的安全』。

相關工作

第 402 頁的『防止佇列管理程式接收訊息』
您可以使用結束程式來防止叢集佇列管理程式接收未獲授權接收的訊息。

IBM MQ 發佈/訂閱的安全

如果您使用「 IBM MQ 發佈/訂閱」，則有其他安全考量。

在發佈/訂閱系統中，有兩種類型的應用程式: 發佈者和訂閱者。 發佈者 以 IBM MQ 訊息形式提供資訊。 當發佈者發佈訊息時，它會指定 主題，以識別訊息內資訊的主旨。

訂閱者 是已發佈資訊的消費者。 訂閱者透過訂閱來指定感興趣的主題。

佇列管理程式 是「 IBM MQ 發佈/訂閱」所提供的應用程式。 它接收來自發佈者的已發佈訊息及來自訂閱者的訂閱要求，並將已發佈訊息遞送給訂閱者。 訂閱者只會在其訂閱的那些主題上傳送訊息。

如需相關資訊，請參閱 發佈/訂閱安全。

多重播送安全

使用此資訊來瞭解 IBM MQ Multicast 可能需要安全處理程序的原因。

IBM MQ Multicast 沒有內建安全。 安全檢查在佇列管理程式的 MQOPEN 時間處理， MQMD 欄位設定由用戶端處理。 網路中的部分應用程式可能不是 IBM MQ 應用程式 (例如， LLM 應用程式，如需相關資訊，請參閱 具有 IBM MQ 低延遲傳訊的多重播送交互作業能力)，因此您可能需要實作自己的安全程序，因為接收應用程式無法確定環境定義欄位的有效性。

有三個安全處理程序需要考量:

存取控制

IBM MQ 中的存取控制基於使用者 ID。 如需此主題的相關資訊，請參閱 第 83 頁的『用戶端的存取控制』。

網路安全

隔離網路可能是防止偽造訊息的可行安全選項。 多重播送群組位址上的應用程式可以使用原生通訊功能來發佈惡意訊息，這些訊息與 MQ 訊息無法區分，因為它們來自相同多重播送群組位址上的應用程式。

在多重播送群組位址上的用戶端也可以接收預期用於相同多重播送群組位址上其他用戶端的訊息。

隔離多重播送網路可確保只有有效的用戶端及應用程式具有存取權。 此安全預防措施可以防止惡意訊息進入，並防止機密資訊進入。

如需多重播送群組網址的相關資訊，請參閱: 設定多重播送資料流量的適當網路

數位簽章

通過加密訊息的表示來形成數字簽名。 加密會使用簽署人的私密金鑰，為了效率，通常會對訊息摘要而非訊息本身進行操作。 在 MQPUT 之前對訊息進行數位簽署是良好的安全預防措施，但如果有大量訊息，此處理程序可能會對效能產生不利影響。

數位簽章會隨著所簽署的資料而不同。 如果相同實體以數位方式簽署兩個不同的訊息，則兩個簽章會不同，但可以使用相同的公開金鑰 (即簽署訊息之實體的公開金鑰) 來驗證這兩個簽章。

如本節先前所述，多重播送群組位址上的應用程式可能使用原生通訊功能來發佈惡意訊息，這些功能與 MQ 訊息無法區分。 數位簽章提供來源證明，且只有傳送者知道私密金鑰，這提供有力的證據證明傳送者是訊息的創始者。

如需此主題的相關資訊，請參閱 第 7 頁的『加密概念』。

防火牆和網際網路透通

您通常會使用防火牆來防止來自惡意 IP 位址的存取，例如在「阻斷服務」攻擊中。 不過，您可能需要暫時封鎖 IBM MQ 內的 IP 位址，可能是在您等待安全管理者更新防火牆規則時。

若要封鎖一個以上 IP 位址，請建立 BLOCKADDR 或 ADDRESSMAP 類型的通道鑑別記錄。 如需相關資訊，請參閱第 319 頁的『封鎖特定 IP 位址』。

的安全 IBM MQ Internet Pass-Thru

IBM MQ Internet Pass-Thru 可以簡化透過防火牆的通訊，但這有安全含意。

IBM MQ Internet Pass-Thru (MQIPT) 是 IBM MQ 的選用元件，可用來跨網際網路在遠端網站之間實作傳訊解決方案。

MQIPT 可讓兩個佇列管理程式透過網際網路交換訊息，或讓 IBM MQ 用戶端應用程式透過網際網路連接至佇列管理程式，而不需要直接 TCP/IP 連線。 如果防火牆禁止兩個系統之間的直接 TCP/IP 連線，這會很有用。 它透過在 HTTP 內或充當 Proxy，讓進入及離開防火牆的 IBM MQ 通道通訊協定流程更簡單且更易於管理。 使用「傳輸層安全 (TLS)」，也可以用來加密及解密透過網際網路傳送的訊息。

當 IBM MQ 系統與 MQIPT 通訊時，除非您在 MQIPT 中使用 SSL Proxy 模式，否則請確定 IBM MQ 使用的 CipherSpec 符合 MQIPT 使用的 CipherSuite :

- 當 MQIPT 充當 TLS 伺服器且 IBM MQ 作為 TLS 用戶端進行連接時， IBM MQ 所使用的 CipherSpec 必須對應於在相關 MQIPT 金鑰環中啟用的 CipherSuite 。

- 當 MQIPT 充當 TLS 用戶端並連接至 IBM MQ TLS 伺服器時， MQIPT CipherSuite 必須符合接收端 IBM MQ 通道上定義的 CipherSpec 。

如果從 MQIPT 移轉至整合式 IBM MQ TLS 支援，請使用 **mqiptKeyman** 或 **mqiptKeycmd** 從 MQIPT 金鑰環傳送數位憑證。

如需相關資訊，請參閱 IBM MQ Internet Pass-Thru。

z/OS IBM MQ for z/OS 安全實作核對清單

本主題提供逐步程序，您可以用來解決及定義每一個 IBM MQ 佇列管理程式的安全實作。

RACF 在其提供的靜態「類別描述子表格 (CDT)」中提供 IBM MQ 安全類別的定義。 當您完成核對清單時，您可以決定您的設定需要這些類別中的哪些類別。 您必須確保它們如 第 156 頁的『RACF 安全類別』中所述啟動。

如需詳細資料，特別是 第 165 頁的『用來控制 IBM MQ 資源存取權的設定檔』，請參閱其他小節。

如果您需要安全檢查，請遵循此核對清單來實作它:

1. 啟動 RACF MQADMIN (大寫設定檔) 或 MXADMIN (大小寫混合格式設定檔) 類別。
   - 您要佇列共用群組層次、佇列管理程式層次或兩者的組合中的安全嗎?

     請參閱 第 161 頁的『控制佇列共用群組或佇列管理程式層次安全的設定檔』。
2. 您需要連線安全嗎?
   - **是**: 啟動 MQCONN 類別。 請在 MQCONN 類別中的佇列管理程式層次或佇列共用群組層次定義適當的連線設定檔。 然後允許適當的使用者或群組存取這些設定檔。

     註: 只有 MQCONN API 要求或 CICS 或 IMS 位址空間使用者 ID 的使用者才需要具備對應連線設定檔的存取權。
   - **否**: 定義 hlq.NO.CONNECT.CHECKS 設定檔。
3. 您需要對指令進行安全檢查嗎?
   - **是**: 啟動 MQCMDS 類別。 請在 MQCMDS 類別中的佇列管理程式層次或佇列共用群組層次定義適當的指令設定檔。 然後允許適當的使用者或群組存取這些設定檔。

     如果您使用佇列共用群組，則可能需要包括佇列管理程式本身及通道起始程式所使用的使用者 ID。 請參閱 第 214 頁的『設定 IBM MQ for z/OS 資源安全』。
   - **否**: 定義 hlq.NO.CMD.CHECKS 設定檔。
4. 您需要指令中所使用資源的安全嗎?
   - **是**: 請確定 MQADMIN 或 MXADMIN 類別在作用中。 在 MQADMIN 或 MXADMIN 類別中的佇列管理程式層次或佇列共用群組層次上，定義適當的設定檔來保護指令上的資源。 然後允許適當的使用者或群組存取這些設定檔。 將 CSQ6SYSP 中的 CMDUSER 參數設為要用於指令安全檢查的預設使用者 ID。

     如果您使用佇列共用群組，則可能需要包括佇列管理程式本身及通道起始程式所使用的使用者 ID。 請參閱 第 214 頁的『設定 IBM MQ for z/OS 資源安全』。
   - **否**: 定義 hlq.NO.CMD.RESC.CHECKS 設定檔。
5. 您需要佇列安全嗎?
   - **是**: 啟動 MQQUEUE 或 MXQUEUE 類別。 在 MQQUEUE 或 MXQUEUEclass 中，為必要的佇列管理程式或佇列共用群組定義適當的佇列設定檔。 然後允許適當的使用者或群組存取這些設定檔。
   - **否**: 定義 hlq.NO.QUEUE.CHECKS 設定檔。
6. 您需要處理程序安全嗎?
   - **是**: 啟動 MQPROC 或 MXPROC 類別。 在佇列管理程式或佇列共用群組層次定義適當的程序設定檔，並允許適當的使用者或群組存取這些設定檔。
   - **否**: 定義 hlq.NO.PROCESS.CHECKS 設定檔。
7. 您需要名單安全嗎?

- **是**: 啟動 MQNLIST 或 MXNLISTclass。 在 MQNLIST 或 MXNLIST 類別中的佇列管理程式層次或佇列共用群組層次定義適當的名單設定檔。 然後允許適當的使用者或群組存取這些設定檔。
- **否**: 定義 hlq.NO.NLIST.CHECKS 設定檔。

8. 您需要主題安全嗎?
   - **是**: 啟動 MXTOPIC 類別。 在 MXTOPIC 類別中的佇列管理程式層次或佇列共用群組層次定義適當的主題設定檔。 然後允許適當的使用者或群組存取這些設定檔。
   - **否**: 定義 hlq.NO.TOPIC.CHECKS 設定檔。
9. 是否有任何使用者需要保護使用與使用環境定義相關的 MQOPEN 或 MQPUT1 選項?
   - **是**: 請確定 MQADMIN 或 MXADMIN 類別在作用中。 在 MQADMIN 或 MXADMIN 類別中的佇列、佇列管理程式或佇列共用群組層次定義 hlq.CONTEXT.queuename 設定檔。 然後允許適當的使用者或群組存取這些設定檔。
   - **否**: 定義 hlq.NO.CONTEXT.CHECKS 設定檔。
10. 您是否需要保護使用替代使用者 ID?
    - **是**: 請確定 MQADMIN 或 MXADMIN 類別在作用中。 定義適當的 hlq.ALTERNATE.USER。 必要佇列管理程式或佇列共用群組的 *alternateuserid* 設定檔，並允許必要使用者或群組存取這些設定檔。
    - **否**: 定義設定檔 hlq.NO.ALTERNATE.USER.CHECKS 。
11. 您是否需要透過 RESLEVEL 自訂要使用哪些使用者 ID 來進行資源安全檢查?
    - **是**: 請確定 MQADMIN 或 MXADMIN 類別在作用中。 在 MQADMIN 或 MXADMIN 類別中的佇列管理程式層次或佇列共用群組層次，定義 hlq.RESLEVEL 設定檔。 然後允許必要的使用者或群組存取設定檔。
    - **否**: 請確定 MQADMIN 或 MXADMIN 類別中沒有可套用至 hlq.RESLEVEL 的通用設定檔。 請為必要的佇列管理程式或佇列共用群組定義 hlq.RESLEVEL 設定檔，並確定沒有任何使用者或群組可以存取它。
12. 您需要從 IBM MQ 中「逾時」未用的使用者 ID 嗎?
    - **是**: 決定您要使用的逾時值，並發出 MQSC ALTER SECURITY 指令來變更 TIMEOUT 及 INTERVAL 參數。
    - **否**: 發出 MQSC ALTER SECURITY 指令，將 INTERVAL 值設為零。

    註: 更新子系統所使用的 CSQINP1 起始設定輸入資料集，以便在啟動佇列管理程式時自動發出 MQSC ALTER SECURITY 指令。
13. 您使用分散式佇列嗎?
    - **是**: 使用通道鑑別記錄。 如需相關資訊，請參閱 第 40 頁的『通道鑑別記錄』。
    - 您也可以決定每一個通道的適當 MCAUSER 屬性值，或提供適當的通道安全結束程式。
14. 您要使用「傳輸層安全 (TLS)」嗎?
    - **是**: 若要指定提供包含指定 DN 之 TLS 個人憑證的任何使用者使用特定的 MCAUSER ，請設定 SSLPEERMAP 類型的通道鑑別記錄。 您可以指定單一識別名稱或包括萬用字元的型樣。
    - 規劃 TLS 基礎架構。 安裝 z/OS 的「系統 SSL」特性。 在 RACF 中，設定憑證名稱過濾器 (CNF) (如果您使用它們的話) 以及數位憑證。 設定 SSL 金鑰環。 請確定 SSLKEYR 佇列管理程式屬性非空白，且指向您的 SSL 金鑰環。 另請確保 SSLTASKS 的值至少為 2。
    - **否**: 請確定 SSLKEYR 為空白，且 SSLTASKS 為零。

    如需 TLS 的進一步詳細資料，請參閱 第 20 頁的『IBM MQ 中的 TLS 安全通訊協定』。
15. 您使用客戶嗎?
    - **是**: 使用通道鑑別記錄。
    - 您也可以決定每一個伺服器連線通道的適當 MCAUSER 屬性值，或在必要時提供適當的通道安全結束程式。
16. 請檢查您的交換器設定。

當佇列管理程式已啟動且顯示您的安全設定時， IBM MQ 會發出訊息。 請使用這些訊息來判斷是否已正確設定交換器。

17. 您是否從用戶端應用程式傳送密碼?
    - **是**: 請確定已安裝 z/OS 特性，且已啟動「整合加密服務機能 (ICSF)」，以取得最佳保護。
    - **否**: 您可以忽略報告 ICSF 尚未啟動的錯誤訊息。

    如需 ICSF 的進一步相關資訊，請參閱 第 221 頁的『使用「整合加密服務機能 (ICSF)」』

設定安全

此主題集合包含不同作業系統及用戶端使用的特定資訊。

ALW 在 AIX, Linux, and Windows 上設定安全

AIX, Linux, and Windows 系統特有的安全考量。

IBM MQ 佇列管理程式會傳送具有潛在價值的資訊，因此您需要使用權限系統來確保未獲授權的使用者無法存取您的佇列管理程式。 請考量下列類型的安全控制:

誰可以管理 IBM MQ
您可以定義一組可以發出指令來管理 IBM MQ 的使用者。

誰可以使用 IBM MQ 物件
您可以定義哪些使用者 (通常是應用程式) 可以使用 MQI 呼叫及 PCF 指令來執行下列動作:

- 誰可以連接至佇列管理程式。
- 誰可以存取物件 (佇列、程序定義、名稱清單、通道、用戶端連線通道、接聽器、服務及鑑別資訊物件)，以及他們對那些物件具有何種類型的存取權。
- 誰可以存取 IBM MQ 訊息。
- 誰可以存取與訊息相關聯的環境定義資訊。

通道安全性
您需要確保用來將訊息傳送至遠端系統的通道可以存取所需的資源。

您可以使用標準作業機能來授與對程式庫、MQI 鏈結程式庫及指令的存取權。 不過，包含佇列及其他佇列管理程式資料的目錄是 IBM MQ 專用的; 請不要使用標準作業系統指令來授與或撤銷對 MQI 資源的授權。

ALW 授權在 AIX, Linux, and Windows 上的運作方式

本節主題中的授權規格表格精確定義授權的運作方式及適用的限制。

這些表格適用於下列狀況:

- 發出 MQI 呼叫的應用程式
- 以跳出 PCF 形式發出 MQSC 指令的管理程式
- 發出 PCF 指令的管理程式

在此區段中，資訊會呈現為一組指定下列項目的表格:

要執行的動作
MQI 選項、MQSC 指令或 PCF 指令。

存取控制物件
佇列、處理程序、佇列管理程式、名單、鑑別資訊、通道、用戶端連線通道、接聽器或服務。

需要授權
以 MQZAO_ 常數表示。

在表格中，字首為 MQZAO_ 的常數對應於特定實體的 `setmqaut` 指令授權清單中的關鍵字。 For example, MQZAO_BROWSE corresponds to the keyword `+browse`, MQZAO_SET_ALL_CONTEXT corresponds to the keyword `+setall`, and so on. 這些常數定義在產品隨附的標頭檔 cmqzc.h 中。

ALW *MQI* 呼叫的授權

MQCONN、 **MQOPEN**、 **MQPUT1** 和 **MQCLOSE** 可能需要授權檢查。 本主題中的表格彙總每一個呼叫所需的授權。

只有在執行應用程式的使用者 ID (或其授權可以假設) 已獲授與相關授權時，才容許應用程式發出特定的 MQI 呼叫及選項。

四個 MQI 呼叫可能需要授權檢查: **MQCONN**、 **MQOPEN**、 **MQPUT1** 及 **MQCLOSE**。

對於 **MQOPEN** 和 **MQPUT1**，會對所開啟物件的名稱進行權限檢查，而不是對名稱進行權限檢查，在解析名稱之後所產生的名稱。 例如，應用程式可能被授與開啟別名佇列的權限，而沒有開啟別名所解析成的基本佇列的權限。 規則是除非直接開啟佇列管理程式別名定義，否則會對在解析非佇列管理程式別名的名稱過程中所發現的第一個定義執行檢查; 亦即，其名稱會顯示在物件描述子的 *ObjectName* 欄位中。 所開啟的物件一律需要權限。 在某些情況下，需要透過佇列管理程式物件的授權取得其他與佇列無關的權限。

第 107 頁的表 10、 第 107 頁的表 11、 第 108 頁的表 12 和 第 108 頁的表 13 彙總每一個呼叫所需的授權。 在表格 不適用 中，表示授權檢查與這項作業無關; 不檢查 表示不執行授權檢查。

註: 您將在這些表格中找不到名稱清單、通道、用戶端連線通道、接聽器、服務或鑑別資訊物件的提及項目。 這是因為除了 MQOO_INQUIRE 之外，沒有任何授權適用於這些物件，其適用的授權與適用於其他物件的授權相同。

特殊授權 MQZAO_ALL_MQI 包括表格中與物件類型相關的所有授權，但分類為管理授權的 MQZAO_DELETE 及 MQZAO_DISPLAY 除外。

若要修改任何訊息環境定義選項，您必須具有適當的授權才能發出呼叫。 例如，若要使用 MQOO_SET_IDENTITY_CONTEXT 或 MQPMO_SET_IDENTITY_CONTEXT，您必須具有 `+setid` 許可權。

表 *10: MQCONN* 呼叫所需的安全授權

需要授權:	**佇列物件 (第 109 頁的『1』)**	**程序物件**	**佇列管理程式物件**
MQCONN	不適用	不適用	MQZAO_CONNECT

表 *11: MQOPEN* 呼叫所需的安全授權

需要授權:	**佇列物件 (第 109 頁的『1』)**	**程序物件**	**佇列管理程式物件**
MQOO_INQUIRE	MQZAO_INQUIRE	MQZAO_INQUIRE	MQZAO_INQUIRE
MQ 瀏覽	MQ 導覽 _ 瀏覽	不適用	不檢查
MQOO_INPUT_ *	MQZAO_ 輸入	不適用	不檢查
MQOO_SAVE_ ALL_CONTEXT **(第 109 頁的『2』)**	MQZAO_ 輸入	不適用	不適用
MQOO_OUTPUT (正常佇列) **(第 109 頁的『3』)**	MQZAO_OUTPUT	不適用	不適用
MQOO_PASS_ IDENTITY_CONTEXT **(第 109 頁的『4』)**	MQZAO_PASS_ IDENTITY_CONTEXT	不適用	不檢查
MQOO_PASS_ALL_ CONTEXT **(第 109 頁的『4』， 第 109 頁的『5』)**	MQZAO_PASS _ALL_CONTEXT	不適用	不檢查

表 *11: MQOPEN* 呼叫所需的安全授權 *(繼續)*			
需要授權:	**佇列物件 (第 109 頁的『1』)**	**程序物件**	**佇列管理程式物件**
MQOO_SET_IDENTITY_CONTEXT **(第 109 頁的『4』， 第 109 頁的『5』)**	MQZAO_SET_IDENTITY_CONTEXT	不適用	MQZAO_SET_IDENTITY_CONTEXT **(第 109 頁的『6』)**
MQOO_SET_ALL_CONTEXT **(第 109 頁的『4』， 第 109 頁的『7』)**	MQZAO_SET_ALL_CONTEXT	不適用	MQZAO_SET_ALL_CONTEXT **(第 109 頁的『6』)**
MQOO_OUTPUT (傳輸佇列) **(第 109 頁的『8』)**	MQZAO_SET_ALL_CONTEXT	不適用	MQZAO_SET_ALL_CONTEXT **(第 109 頁的『6』)**
MQOO_SET	MQZAO_SET	不適用	不檢查
MQOO_ALTERNATE_USER_AUTHORITY	**(第 109 頁的『9』)**	**(第 109 頁的『9』)**	MQZAO_ALTERNATE_USER_AUTHORITY **(第 109 頁的『9』， 第 109 頁的『10』)**

表 *12: MQPUT1* 呼叫所需的安全授權			
需要授權:	**佇列物件 (第 109 頁的『1』)**	**程序物件**	**佇列管理程式物件**
MQPMO_PASS_IDENTITY_CONTEXT	MQZAO_PASS_IDENTITY_CONTEXT **(第 109 頁的『11』)**	不適用	不檢查
MQPMO_PASS_ALL_CONTEXT	MQZAO_PASS_ALL_CONTEXT **(第 109 頁的『11』)**	不適用	不檢查
MQPMO_SET_IDENTITY_CONTEXT	MQZAO_SET_IDENTITY_CONTEXT **(第 109 頁的『11』)**	不適用	MQZAO_SET_IDENTITY_CONTEXT **(第 109 頁的『6』)**
MQPMO_SET_ALL_CONTEXT	MQZAO_SET_ALL_CONTEXT **(第 109 頁的『11』)**	不適用	MQZAO_SET_ALL_CONTEXT **(第 109 頁的『6』)**
(傳輸佇列) **(第 109 頁的『8』)**	MQZAO_SET_ALL_CONTEXT	不適用	MQZAO_SET_ALL_CONTEXT **(第 109 頁的『6』)**
MQPMO_ALTERNATE_USER_AUTHORITY	**(第 109 頁的『12』)**	不適用	MQZAO_ALTERNATE_USER_AUTHORITY **(第 109 頁的『10』)**

表 *13: MQCLOSE* 呼叫所需的安全授權			
需要授權:	**佇列物件 (第 109 頁的『1』)**	**程序物件**	**佇列管理程式物件**
MQCO_DELETE	MQZAO_DELETE **(第 109 頁的『13』)**	不適用	不適用

表 *13: MQCLOSE* 呼叫所需的安全授權 *(繼續)*			
需要授權:	**佇列物件 (第 109 頁的『1』)**	**程序物件**	**佇列管理程式物件**
MQCO_DELETE _PURGE	MQZAO_DELETE **(第 109 頁的『13』)**	不適用	不適用

表格注意事項:

1. 如果開啟模型佇列:
   - 除了為您開啟的存取權類型開啟模型佇列的權限之外，還需要模型佇列的 MQZAO_DISPLAY 權限。
   - 不需要 MQZAO_CREATE 權限即可建立動態佇列。
   - 用來開啟模型佇列的使用者 ID 會自動授與所建立動態佇列的所有佇列特定權限 (相當於 MQZAO_ALL)。
2. 也必須指定 MQOO_INPUT_ *。 這適用於本端、模型或別名佇列。
3. 此檢查是針對所有輸出案例執行，但傳輸佇列除外 (請參閱附註 第 109 頁的『8』)。
4. 也必須指定 MQOO_OUTPUT。
5. 此選項也隱含 MQOO_PASS_IDENTITY_CONTEXT。
6. 佇列管理程式物件及特定佇列都需要此權限。
7. 此選項也隱含 MQOO_PASS_IDENTITY_CONTEXT、MQOO_PASS_ALL_CONTEXT 及 MQOO_SET_IDENTITY_CONTEXT。
8. 針對 *Usage* 佇列屬性為 MQUS_TRANSMISSION 且直接開啟以供輸出的本端或模型佇列執行此檢查。 如果正在開啟遠端佇列 (透過指定遠端佇列管理程式及遠端佇列的名稱，或透過指定遠端佇列的本端定義名稱) ，則此不適用。
9. 至少必須指定 MQOO_INQUIRE (適用於任何物件類型) 或 MQOO_BROWSE、MQOO_INPUT_ *、MQOO_OUTPUT 或 MQOO_SET (適用於佇列) 其中之一。 所執行的檢查與其他指定選項一樣，使用所提供的替代使用者 ID (針對特定命名物件權限) ，以及現行應用程式權限 (針對 MQZAO_ALTERNATE_USER_IDID 檢查)。
10. 此授權容許指定任何 *AlternateUserId* 。
11. 如果佇列沒有 MQUS_TRANSMISSION 的 *Usage* 佇列屬性，則也會執行 MQZAO_OUTPUT 檢查。
12. 所執行的檢查與其他指定選項一樣，使用所提供的替代使用者 ID (針對特定命名的佇列權限) ，以及現行應用程式權限 (針對 MQZAO_ALTERNATE_USER_ID 檢查)。
13. 只有在下列兩個陳述式都成立時，才會執行檢查:
    - 正在關閉並刪除永久動態佇列。
    - 佇列不是由傳回所使用物件控點的 MQOPEN 呼叫所建立。

    否則，不會有任何檢查。

ALW 跳出 *PCF* 中 *MQSC* 指令的授權

此資訊彙總 Escape PCF 中包含的每一個 MQSC 指令所需的授權。

不適用 表示此作業與此物件類型無關。

提交指令的程式所使用的使用者 ID 也必須具有下列權限:

- 佇列管理程式的 MQZAO_CONNECT 權限
- 佇列管理程式上的 MQZAO_DISPLAY 權限，以便執行 PCF 指令
- 在 Escape PCF 指令文字內發出 MQSC 指令的權限

ALTER 物件

物件	需要授權
佇列	MQZAO_CHANGE
主題	MQZAO_CHANGE
處理程序	MQZAO_CHANGE
佇列管理程式	MQZAO_CHANGE
名稱清單	MQZAO_CHANGE
鑑別資訊	MQZAO_CHANGE
通道	MQZAO_CHANGE
用戶端連線通道	MQZAO_CHANGE
接聽器	MQZAO_CHANGE
服務	MQZAO_CHANGE
通訊資訊	MQZAO_CHANGE

CLEAR 物件

物件	需要授權
佇列	MQZAO_clear
主題	MQZAO_clear
處理程序	不適用
佇列管理程式	不適用
名稱清單	不適用
鑑別資訊	不適用
通道	不適用
用戶端連線通道	不適用
接聽器	不適用
服務	不適用
通訊資訊	不適用

DEFINE 物件 NOREPLACE (第 114 頁的『1』)

物件	需要授權
佇列	MQZAO_CREATE (第 114 頁的『2』)
主題	MQZAO_CREATE (第 114 頁的『2』)
處理程序	MQZAO_CREATE (第 114 頁的『2』)
佇列管理程式	不適用
名稱清單	MQZAO_CREATE (第 114 頁的『2』)
鑑別資訊	MQZAO_CREATE (第 114 頁的『2』)
通道	MQZAO_CREATE (第 114 頁的『2』)

物件	需要授權
用戶端連線通道	MQZAO_CREATE (第 114 頁的『2』)
接聽器	MQZAO_CREATE (第 114 頁的『2』)
服務	MQZAO_CREATE (第 114 頁的『2』)
通訊資訊	MQZAO_CREATE (第 114 頁的『2』)

DEFINE 物件 REPLACE (第 114 頁的『1』， 第 114 頁的『3』)

物件	需要授權
佇列	MQZAO_CHANGE
主題	MQZAO_CHANGE
處理程序	MQZAO_CHANGE
佇列管理程式	不適用
名稱清單	MQZAO_CHANGE
鑑別資訊	MQZAO_CHANGE
通道	MQZAO_CHANGE
用戶端連線通道	MQZAO_CHANGE
接聽器	MQZAO_CHANGE
服務	MQZAO_CHANGE
通訊資訊	MQZAO_CHANGE

DELETE 物件

物件	需要授權
佇列	MQZAO_DELETE
主題	MQZAO_DELETE
處理程序	MQZAO_DELETE
佇列管理程式	不適用
名稱清單	MQZAO_DELETE
鑑別資訊	MQZAO_DELETE
通道	MQZAO_DELETE
用戶端連線通道	MQZAO_DELETE
接聽器	MQZAO_DELETE
服務	MQZAO_DELETE
通訊資訊	MQZAO_DELETE

DISPLAY *object*

物件	需要授權
佇列	MQZAO_DISPLAY
主題	MQZAO_DISPLAY

物件	需要授權
處理程序	MQZAO_DISPLAY
佇列管理程式	MQZAO_DISPLAY
名稱清單	MQZAO_DISPLAY
鑑別資訊	MQZAO_DISPLAY
通道	MQZAO_DISPLAY
用戶端連線通道	MQZAO_DISPLAY
接聽器	MQZAO_DISPLAY
服務	MQZAO_DISPLAY
通訊資訊	MQZAO_DISPLAY

START 物件

物件	需要授權
佇列	不適用
主題	不適用
處理程序	不適用
佇列管理程式	不適用
名稱清單	不適用
鑑別資訊	不適用
通道	MQZAO_CONTROL
用戶端連線通道	不適用
接聽器	MQZAO_CONTROL
服務	MQZAO_CONTROL
通訊資訊	不適用

STOP 物件

物件	需要授權
佇列	不適用
主題	不適用
處理程序	不適用
佇列管理程式	不適用
名稱清單	不適用
鑑別資訊	不適用
通道	MQZAO_CONTROL
用戶端連線通道	不適用
接聽器	MQZAO_CONTROL
服務	MQZAO_CONTROL

物件	需要授權
通訊資訊	不適用

通道指令

指令	物件	需要授權
Ping 通道	通道	MQZAO_CONTROL
重設通道	通道	已延伸 MQZAO_CONTROL_EXTENDED
解析通道	通道	已延伸 MQZAO_CONTROL_EXTENDED

訂閱指令

指令	物件	需要授權
ALTER SUB	主題	MQZAO_CONTROL
DEFINE SUB	主題	MQZAO_CONTROL
DELETE SUB	主題	MQZAO_CONTROL
DISPLAY SUB	主題	MQZAO_DISPLAY

安全指令

指令	物件	需要授權
SET AUTHREC	佇列管理程式	MQZAO_CHANGE
DELETE AUTHREC	佇列管理程式	MQZAO_CHANGE
DISPLAY AUTHREC	佇列管理程式	MQZAO_DISPLAY
DISPLAY AUTHSERV	佇列管理程式	MQZAO_DISPLAY
DISPLAY ENTAUTH	佇列管理程式	MQZAO_DISPLAY
SET CHLAUTH	佇列管理程式	MQZAO_CHANGE
DISPLAY CHLAUTH	佇列管理程式	MQZAO_DISPLAY
REFRESH SECURITY	佇列管理程式	MQZAO_CHANGE

狀態顯示畫面

指令	物件	需要授權
DISPLAY CHSTATUS	佇列管理程式	MQZAO_DISPLAY 請注意，如果通道類型是 CLUSSDR，則傳輸佇列上需要 +inq 權限 (或同等的 MQZAO_INQUIRE)。
DISPLAY LSSTATUS	佇列管理程式	MQZAO_DISPLAY
DISPLAY PUBSUB	佇列管理程式	MQZAO_DISPLAY
DISPLAY SBSTATUS	佇列管理程式	MQZAO_DISPLAY
DISPLAY SVSTATUS	佇列管理程式	MQZAO_DISPLAY

指令	物件	需要授權
DISPLAY TPSTATUS	佇列管理程式	MQZAO_DISPLAY

叢集指令

指令	物件	需要授權
DISPLAY CLUSQMGR	佇列管理程式	MQZAO_DISPLAY
重新整理叢集	需要 'mqm' 群組成員資格	
重設叢集	需要 'mqm' 群組成員資格	
SUSPEND 佇列管理程式	需要 'mqm' 群組成員資格	
回復佇列管理程式	需要 'mqm' 群組成員資格	

其他管理指令

指令	物件	需要授權
PING 佇列管理程式	佇列管理程式	MQZAO_DISPLAY
重新整理佇列管理程式	佇列管理程式	MQZAO_CHANGE
RESET QMGR	佇列管理程式	MQZAO_CHANGE
DISPLAY CONN	佇列管理程式	MQZAO_DISPLAY
STOP CONN	佇列管理程式	MQZAO_CHANGE

註:

1. 對於 DEFINE 指令， LIKE 物件也需要 MQZAO_DISPLAY 權限 (如果已指定)，或在適當的 SYSTEM.DEFAULT.xxx 物件 (如果省略 LIKE)。
2. MQZAO_CREATE 權限不是特定物件或物件類型所特有。 透過在 `setmqaut` 指令上指定物件類型 QMGR，授與指定佇列管理程式的所有物件建立權限。
3. 如果要取代的物件已存在，則適用此情況。 如果沒有，則檢查是針對 DEFINE 物件 NOREPLACE。

相關資訊

叢集作業：使用 REFRESH CLUSTER 最佳作法

ALW *PCF* 指令的授權

本節彙總每一個 PCF 指令所需的授權。

不檢查 表示不執行授權檢查; 不適用 表示此作業與此物件類型無關。

提交指令的程式所使用的使用者 ID 也必須具有下列權限:

- 佇列管理程式的 MQZAO_CONNECT 權限
- 佇列管理程式上的 MQZAO_DISPLAY 權限，以便執行 PCF 指令

特殊授權 MQZAO_ALL_ADMIN 包括下列清單中與物件類型相關的所有授權，但非特定物件或物件類型專用的 MQZAO_CREATE 除外。

變更 物件

物件	需要授權
佇列	MQZAO_CHANGE
主題	MQZAO_CHANGE
程序	MQZAO_CHANGE

物件	需要授權
佇列管理程式	MQZAO_CHANGE
名稱清單	MQZAO_CHANGE
鑑別資訊	MQZAO_CHANGE
通道	MQZAO_CHANGE
用戶端連線通道	MQZAO_CHANGE
接聽器	MQZAO_CHANGE
服務	MQZAO_CHANGE
通訊資訊	MQZAO_CHANGE

清除 物件

物件	需要授權
佇列	MQZAO_clear
主題	MQZAO_clear
處理程序	不適用
佇列管理程式	不適用
名稱清單	不適用
鑑別資訊	不適用
通道	不適用
用戶端連線通道	不適用
接聽器	不適用
服務	不適用
通訊資訊	不適用

複製 物件 (不取代) (1)

物件	需要授權
佇列	MQZAO_CREATE **(2)**
主題	MQZAO_CREATE **(2)**
程序	MQZAO_CREATE **(2)**
佇列管理程式	不適用
名稱清單	MQZAO_CREATE **(2)**
鑑別資訊	MQZAO_CREATE **(2)**
通道	MQZAO_CREATE **(2)**
用戶端連線通道	MQZAO_CREATE **(2)**
接聽器	MQZAO_CREATE **(2)**
服務	MQZAO_CREATE **(2)**
通訊資訊	MQZAO_CREATE **(第 120 頁的『2』)**

複製 物件 (含取代) (1, 4)

物件	需要授權
佇列	MQZAO_CHANGE
主題	MQZAO_CHANGE
程序	MQZAO_CHANGE
佇列管理程式	不適用
名稱清單	MQZAO_CHANGE
鑑別資訊	MQZAO_CHANGE
通道	MQZAO_CHANGE
用戶端連線通道	MQZAO_CHANGE
接聽器	MQZAO_CHANGE
服務	MQZAO_CHANGE
通訊資訊	MQZAO_CHANGE

建立 物件 (不取代) (3)

物件	需要授權
佇列	MQZAO_CREATE **(2)**
主題	MQZAO_CREATE **(2)**
程序	MQZAO_CREATE **(2)**
佇列管理程式	不適用
名稱清單	MQZAO_CREATE **(2)**
鑑別資訊	MQZAO_CREATE **(2)**
通道	MQZAO_CREATE **(2)**
用戶端連線通道	MQZAO_CREATE **(2)**
接聽器	MQZAO_CREATE **(2)**
服務	MQZAO_CREATE **(2)**
通訊資訊	MQZAO_CREATE **(2)**

建立 物件 (含取代) (3, 4)

物件	需要授權
佇列	MQZAO_CHANGE
主題	MQZAO_CHANGE
程序	MQZAO_CHANGE
佇列管理程式	不適用
名稱清單	MQZAO_CHANGE
鑑別資訊	MQZAO_CHANGE
通道	MQZAO_CHANGE

物件	需要授權
用戶端連線通道	MQZAO_CHANGE
接聽器	MQZAO_CHANGE
服務	MQZAO_CHANGE
通訊資訊	MQZAO_CHANGE

刪除 物件

物件	需要授權
佇列	MQZAO_DELETE
主題	MQZAO_DELETE
程序	MQZAO_DELETE
佇列管理程式	不適用
名稱清單	MQZAO_DELETE
鑑別資訊	MQZAO_DELETE
通道	MQZAO_DELETE
用戶端連線通道	MQZAO_DELETE
接聽器	MQZAO_DELETE
服務	MQZAO_DELETE
通訊資訊	MQZAO_DELETE

查詢 物件

物件	需要授權
佇列	MQZAO_DISPLAY
主題	MQZAO_DISPLAY
程序	MQZAO_DISPLAY
佇列管理程式	MQZAO_DISPLAY
名稱清單	MQZAO_DISPLAY
鑑別資訊	MQZAO_DISPLAY
通道	MQZAO_DISPLAY
用戶端連線通道	MQZAO_DISPLAY
接聽器	MQZAO_DISPLAY
服務	MQZAO_DISPLAY
通訊資訊	MQZAO_DISPLAY

查詢 *object* 名稱

物件	需要授權
佇列	不檢查
主題	不檢查

物件	需要授權
處理程序	不檢查
佇列管理程式	不檢查
名稱清單	不檢查
鑑別資訊	不檢查
通道	不檢查
用戶端連線通道	不檢查
接聽器	不檢查
服務	不檢查
通訊資訊	不檢查

啟動 物件

物件	需要授權
佇列	不適用
主題	不適用
處理程序	不適用
佇列管理程式	不適用
名稱清單	不適用
鑑別資訊	不適用
通道	MQZAO_CONTROL
用戶端連線通道	不適用
接聽器	MQZAO_CONTROL
服務	MQZAO_CONTROL
通訊資訊	不適用

停止 物件

物件	需要授權
佇列	不適用
主題	不適用
處理程序	不適用
佇列管理程式	不適用
名稱清單	不適用
鑑別資訊	不適用
通道	MQZAO_CONTROL
用戶端連線通道	不適用
接聽器	MQZAO_CONTROL
服務	MQZAO_CONTROL

物件	需要授權
通訊資訊	不適用

通道指令

指令	物件	需要授權
Ping 通道	通道	MQZAO_CONTROL
重設通道	通道	已延伸 MQZAO_CONTROL_EXTENDED
解析通道	通道	已延伸 MQZAO_CONTROL_EXTENDED

訂閱指令

指令	物件	需要授權
變更訂閱	主題	MQZAO_CONTROL
建立訂閱	主題	MQZAO_CONTROL
刪除訂閱	主題	MQZAO_CONTROL
查詢訂閱	主題	MQZAO_DISPLAY

安全指令

指令	物件	需要授權
設定權限記錄	佇列管理程式	MQZAO_CHANGE
刪除權限記錄	佇列管理程式	MQZAO_CHANGE
查詢權限記錄	佇列管理程式	MQZAO_DISPLAY
查詢權限服務	佇列管理程式	MQZAO_DISPLAY
查詢實體權限	佇列管理程式	MQZAO_DISPLAY
設定通道鑑別記錄	佇列管理程式	MQZAO_CHANGE
查詢通道鑑別記錄	佇列管理程式	MQZAO_DISPLAY
重新整理安全	佇列管理程式	MQZAO_CHANGE

狀態顯示畫面

指令	物件	需要授權
查詢通道狀態	佇列管理程式	MQZAO_DISPLAY 請注意，如果通道類型是 CLUSSDR，則傳輸佇列上需要 +inq 權限 (或同等的 MQZAO_INQUIRE)。
查詢通道接聽器狀態	佇列管理程式	MQZAO_DISPLAY
查詢發佈/訂閱狀態	佇列管理程式	MQZAO_DISPLAY
查詢訂閱狀態	佇列管理程式	MQZAO_DISPLAY
查詢服務狀態	佇列管理程式	MQZAO_DISPLAY

指令	物件	需要授權
查詢主題狀態	佇列管理程式	MQZAO_DISPLAY

叢集指令

指令	物件	需要授權
查詢叢集佇列管理程式	佇列管理程式	MQZAO_DISPLAY
重新整理叢集	需要 'mqm' 群組成員資格	需要 'mqm' 群組成員資格
重設叢集	需要 'mqm' 群組成員資格	需要 'mqm' 群組成員資格
暫停佇列管理程式叢集	需要 'mqm' 群組成員資格	需要 'mqm' 群組成員資格
回復佇列管理程式叢集	需要 'mqm' 群組成員資格	需要 'mqm' 群組成員資格

其他管理指令

指令	物件	需要授權
Ping 佇列管理程式	佇列管理程式	MQZAO_DISPLAY
重新整理佇列管理程式	佇列管理程式	MQZAO_CHANGE
重設佇列管理程式	佇列管理程式	MQZAO_CHANGE
重設佇列統計資料	佇列	MQZAO_DISPLAY 和 MQZAO_CHANGE
查詢連線	佇列管理程式	MQZAO_DISPLAY
停止連線	佇列管理程式	MQZAO_CHANGE

註:

1. 對於 Copy 指令， From 物件也需要 MQZAO_DISPLAY 權限。
2. MQZAO_CREATE 權限不是特定物件或物件類型所特有。 透過在 `setmqaut` 指令上指定物件類型 QMGR，授與指定佇列管理程式的所有物件建立權限。
3. 若為「建立」指令，適當的 SYSTEM.DEFAULT.* 物件。
4. 如果要取代的物件已存在，則適用此情況。 如果不存在，則檢查適用於「複製」或「建立而不取代」。

AIX 在 AIX 上建立及管理群組

在 AIX 上，如果您不是使用 NIS 或 NIS +，請使用 SMITTY 來使用群組。

關於這項作業

在 AIX 上，您可以使用 SMITTY 來建立群組、將使用者新增至群組、顯示群組中的使用者清單，以及從群組中移除使用者。

程序

1. 從 SMITTY 中，選取 **安全及使用者**，然後按 Enter 鍵。
2. 選取 **群組**，然後按 Enter 鍵。
3. 若要建立群組，請完成下列步驟:
   a) 選取 **新增群組**，然後按 Enter 鍵。
   b) 輸入群組名稱，以及您要新增至群組的任何使用者名稱 (以逗點區隔)。
   c) 按 Enter 鍵以建立群組。

4. 若要將使用者新增至群組，請完成下列步驟:
   a) 選取 **變更/顯示群組性質** ，然後按 Enter 鍵。
   b) 輸入群組名稱，以顯示群組成員的清單。
   c) 新增您要新增至群組的使用者名稱，以逗點區隔。
   d) 按 Enter 鍵將名稱新增至群組。
5. 若要顯示群組中的人員，請完成下列步驟:
   a) 選取 **變更/顯示群組性質** ，然後按 Enter 鍵。
   b) 輸入群組名稱，以顯示群組成員的清單。
6. 若要從群組中移除使用者，請完成下列步驟:
   a) 選取 **變更/顯示群組性質** ，然後按 Enter 鍵。
   b) 輸入群組名稱，以顯示群組成員的清單。
   c) 刪除您要從群組中移除的使用者名稱。
   d) 按 Enter 鍵以從群組中移除名稱。

Linux 在 Linux 上建立及管理群組

在 Linux 上，如果您不是使用 NIS 或 NIS + ，請使用 `/etc/group` 檔案來使用群組。

關於這項作業

在 Linux 上，群組資訊保留在 `/etc/group` 檔案中。 您可以使用指令來建立群組、將使用者新增至群組、顯示群組中的使用者清單，以及從群組中移除使用者。

程序

1. 若要建立新的群組，請使用 **groupadd** 指令。
   請鍵入下列指令:

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

   其中 *group-ID* 是群組的數值 ID ，而 *group-name* 是群組的名稱。
2. 若要將成員新增至增補群組，請使用 **usermod** 指令來列出使用者目前是其成員的增補群組，以及使用者將成為其成員的增補群組。
   例如，如果使用者已是群組 `groupa` 的成員，且要成為 `groupb` 的成員，請使用下列指令:

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

   其中 *user-name* 是使用者名稱。
3. 若要顯示誰是群組成員，請使用 **getent** 指令。
   請鍵入下列指令:

   ```
   getent group group-name
   ```

   其中 *group-name* 是群組的名稱。
4. 若要從增補群組中移除成員，請使用 **usermod** 指令來列出您希望使用者保留其成員的增補群組。
   例如，如果使用者的主要群組是 `users` ，且使用者也是群組 `mqm`、 `groupa` 及 `groupb` 的成員，則若要從 `mqm` 群組中移除使用者，請使用下列指令:

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

   其中 *user-name* 是使用者名稱。

Windows 在 Windows 上建立及管理群組

在 Windows 上，您可以使用「電腦管理」特性來管理工作站或成員伺服器機器上的群組。

關於這項作業

對於網域控制站，使用者和群組是透過 Active Directory 來管理。 如需使用 Active Directory 的詳細資料，請參閱適當的作業系統指示。

在重新啟動佇列管理程式或您發出 MQSC 指令 **REFRESH SECURITY** (或 PCF 對等項目) 之前，無法辨識您對主體群組成員資格所做的任何變更。

使用「Windows 電腦管理」畫面來處理使用者和群組。 在使用者重新登入之前，對現行登入使用者所做的任何變更可能都不會生效。

Windows 在 *Windows* 上建立群組

使用控制台來建立群組。

程序

1. 開啟控制面板
2. 按兩下 **系統管理工具**。
   即會開啟「系統管理工具」畫面。
3. 按兩下 **電腦管理**。
   即會開啟「電腦管理」畫面。
4. 展開**本機使用者和群組**。
5. 用滑鼠右鍵按一下 **群組**，然後選取 **新建群組 ...**。
   即會顯示「新建群組」畫面。
6. 在「群組名稱」欄位中鍵入適當的名稱，然後按一下 **建立**。
7. 按一下**關閉**。

Windows 將使用者新增至 *Windows* 上的群組

使用控制台將使用者新增至群組。

程序

1. 開啟控制面板
2. 按兩下 **系統管理工具**。
   即會開啟「系統管理工具」畫面。
3. 按兩下 **電腦管理**。
   即會開啟「電腦管理」畫面。
4. 從「電腦管理」畫面中，展開 **本端使用者和群組**。
5. 選取 **使用者**
6. 按兩下您要新增至群組的使用者。
   即會顯示使用者內容畫面。
7. 選取 **成員隸屬** 標籤。
8. 選取您要將使用者新增至其中的群組。 如果您想要的群組不可見:
   a) 按一下**新增...**。
      即會顯示「選取群組」畫面。
   b) 按一下 **位置 ...**。
      即會顯示「位置」畫面。
   c) 從清單中選取您要新增使用者的群組位置，然後按一下 **確定**。
   d) 在提供的欄位中鍵入群組名稱。

      或者，按一下 **進階 ...** 然後 **立即尋找**，以列出目前所選取位置中可用的群組。 從這裡，選取您要新增使用者的群組，然後按一下 **確定**。

e) 按一下**確定**。
即會顯示使用者內容畫面，其中顯示您所新增的群組。
f) 選取群組。
9. 按一下**確定**。
即會顯示「電腦管理」畫面。

Windows 在 *Windows* 上顯示群組中的人員

使用控制面板來顯示群組成員。

程序

1. 開啟控制面板
2. 按兩下 **系統管理工具**。
即會開啟「系統管理工具」畫面。
3. 按兩下 **電腦管理**。
即會開啟「電腦管理」畫面。
4. 從「電腦管理」畫面中，展開 **本端使用者和群組**。
5. 選取 **群組**。
6. 按兩下群組。 即會顯示群組內容畫面。
即會顯示群組內容畫面。

結果

即會顯示群組成員。

Windows 在 *Windows* 上從群組中移除使用者

使用控制面板從群組中移除使用者。

程序

1. 開啟控制面板
2. 按兩下 **系統管理工具**。
即會開啟「系統管理工具」畫面。
3. 按兩下 **電腦管理**。
即會開啟「電腦管理」畫面。
4. 從「電腦管理」畫面中，展開 **本端使用者和群組**。
5. 選取**使用者**。
6. 按兩下您要新增至群組的使用者。
即會顯示使用者內容畫面。
7. 選取 **成員隸屬** 標籤。
8. 選取您要從中移除使用者的群組，然後按一下 **移除**。
9. 按一下**確定**。
即會顯示「電腦管理」畫面。

結果

您現在已從群組中移除使用者。

Windows Windows 上安全的特殊考量

部分安全功能在不同版本的 Windows 上的行為不同。

IBM MQ 安全依賴於對作業系統 API 的呼叫，以取得使用者授權及群組成員資格的相關資訊。 部分功能在 Windows 系統上的行為並不相同。 這個主題集合包括說明當您在 Windows 環境中執行 IBM MQ 時，這些差異可能如何影響 IBM MQ 安全。

Windows *IBM MQ Windows* 服務的本端及網域使用者帳戶

當 IBM MQ 執行時，它必須確認只有已獲授權的使用者可以存取佇列管理程式或佇列。 這需要特殊使用者帳戶， IBM MQ 可以使用該帳戶來查詢任何嘗試此類存取之使用者的相關資訊。

- 第 124 頁的『使用 Prepare IBM MQ Wizard 配置特殊使用者帳戶』
- 第 124 頁的『搭配使用 IBM MQ 與 Active Directory』
- 第 124 頁的『IBM MQ Windows 服務所需的使用者權限』

使用 Prepare IBM MQ Wizard 配置特殊使用者帳戶

Prepare IBM MQ Wizard 會建立特殊使用者帳戶，以便需要使用它的處理程序可以共用 Windows 服務 (請參閱 使用 PPrepare IBM MQ Wizard 來配置 IBM MQ)。

Windows 服務在 IBM MQ 安裝的用戶端程序之間共用。 每個安裝都會建立一個服務。 每一個服務都命名為 MQ_*InstallationName* ，且顯示名稱為 IBM MQ(*InstallationName*)。

因為每一個服務必須在非互動式及互動式登入階段作業之間共用，所以您必須在特殊使用者帳戶下啟動每一個服務。 您可以對所有服務使用一個特殊使用者帳戶，或建立不同的特殊使用者帳戶。 每一個特殊使用者帳戶都必須具有 **以服務方式登入**的使用者權限，如需相關資訊，請參閱 第 125 頁的表 14。 如果使用者 ID 沒有執行服務的權限，則服務不會啟動，且會在 Windows 系統事件日誌中傳回錯誤。 通常，您已執行 Prepare IBM MQ Wizard，並正確地設定使用者 ID。 不過，如果您已手動配置使用者 ID ，是否可能有您需要解決的問題。

當您第一次安裝 IBM MQ 並執行 Prepare IBM MQ Wizard 時，它會為稱為 MUSR_MQADMIN 的服務建立本端使用者帳戶，並具有必要的設定和許可權，包括 **以服務方式登入**。

對於後續安裝， Prepare IBM MQ Wizard 會建立名為 MUSR_MQADMIN*x* 的使用者帳戶，其中 *x* 是下一個可用的號碼，代表不存在的使用者 ID。 建立帳戶時，會隨機產生 MUSR_MQADMIN*x* 的密碼，並用來配置服務的登入環境。 產生的密碼不會到期。

此 IBM MQ 帳戶不受系統上設定的任何帳戶原則所影響，這些原則會要求在特定期間之後變更帳戶密碼。

密碼在此一次性處理之外不明，並由 Windows 作業系統儲存在登錄的安全部分中。

搭配使用 IBM MQ 與 Active Directory

在部分網路配置中，在使用 Active Directory 目錄服務的網域控制站上定義使用者帳戶，執行 IBM MQ 的本端使用者帳戶可能沒有查詢其他網域使用者帳戶的群組成員資格所需的權限。 當您安裝 IBM MQ 時，Prepare IBM MQ Wizard 會執行測試並詢問您網路配置的相關問題，以識別是否為這種情況。

如果執行 IBM MQ 的本端使用者帳戶沒有必要的權限，則 Prepare IBM MQ Wizard 會提示您輸入具有特定使用者權限之網域使用者帳戶的帳戶詳細資料。 如需如何建立及設定 Windows 網域帳戶的相關資訊，請參閱為 IBM MQ。 如需網域使用者帳戶所需的使用者權限，請參閱 第 125 頁的表 14。

當您在 Prepare IBM MQ Wizard 中輸入網域使用者帳戶的有效帳戶詳細資料時，精靈會將 IBM MQ Windows 服務配置成在新帳戶下執行。 帳戶詳細資料保留在「登錄」的安全部分中，使用者無法讀取。

當服務在執行中，只要 IBM MQ Windows 服務在執行中，該服務即會啟動並保持執行中。 啟動 Windows 服務之後登入伺服器的 IBM MQ 管理者可以使用 IBM MQ Explorer 來管理伺服器上的佇列管理程式。 這會將 IBM MQ Explorer 連接至現有的 Windows 服務程序。 這兩個動作需要不同的許可權層次才能運作:

- 啟動程序需要啟動許可權。
- IBM MQ 管理者需要存取權。

IBM MQ Windows 服務所需的使用者權限

下表列出執行 IBM MQ 安裝之 Windows 服務的本端及網域使用者帳戶所需的使用者權限。

表 *14: IBM MQ Windows* 服務所需的使用者權限	
許可權	**說明**
以批次工作登入	啟用 IBM MQ Windows 服務，以在此使用者帳戶下執行。
以服務方式登入	可讓使用者設定 IBM MQ Windows 服務，以使用已配置的帳戶登入。
關閉系統	容許 IBM MQ Windows 服務在服務回復失敗時重新啟動伺服器 (如果已配置的話)。
增加配額	作業系統 `CreateProcessAsUser` 呼叫的必要項目。
作為作業系統的一部分	作業系統 `LogonUser` 呼叫的必要項目。
略過遍訪檢查	作業系統 `LogonUser` 呼叫的必要項目。
更換程序層記號	作業系統 `LogonUser` 呼叫的必要項目。

註: 在執行 ASP 及 IIS 應用程式的環境中可能需要除錯程式權限。

您的網域使用者帳戶必須將這些 Windows 使用者權限設為「本機安全性原則」應用程式中列出的有效使用者權限。 如果沒有，請在伺服器本端使用「本機安全性原則」應用程式，或使用「網域安全性應用程式」網域範圍來設定它們。

Windows *Windows* 伺服器安全許可權

視本端使用者或網域使用者執行安裝而定， IBM MQ 的安裝在 Windows Server 上的行為有所不同。

如果 本端 使用者安裝 IBM MQ，則 Prepare IBM MQ Wizard 會偵測到為 IBM MQ Windows 服務建立的本端使用者可以擷取安裝使用者的群組成員資格資訊。 Prepare IBM MQ Wizard 會詢問使用者關於網路配置的問題，以判定在 Windows 2000 或更新版本上執行的網域控制站上是否定義了其他使用者帳戶。 如果是這樣，則 IBM MQ Windows 服務需要在具有特定設定及權限的網域使用者帳戶下執行。 Prepare IBM MQ Wizard 會提示使用者輸入此使用者的帳戶詳細資料，如 使用 Prepare IBM MQ Wizard 來配置 IBM MQ 中所述。

如果 網域 使用者安裝 IBM MQ，則 Prepare IBM MQ Wizard 會偵測到為 IBM MQ Windows 服務建立的本端使用者無法擷取安裝使用者的群組成員資格資訊。 在此情況下， Prepare IBM MQ Wizard 一律提示使用者輸入網域使用者帳戶的帳戶詳細資料，以供 IBM MQ Windows 服務使用。

當 IBM MQ Windows 服務需要使用網域使用者帳戶時，在使用 Prepare IBM MQ Wizard 配置之前， IBM MQ 無法正確運作。 在使用適當的帳戶配置 Windows 服務之前， Prepare IBM MQ Wizard 不容許使用者繼續執行其他作業。

如需相關資訊，請參閱 建立及設定 IBM MQ 的網域帳戶。

Windows 變更與 *IBM MQ* 服務相關聯的使用者名稱

您可以透過使用 Prepare IBM MQ Wizard 建立新帳戶並輸入其詳細資料，來變更與 IBM MQ 服務相關聯的使用者名稱。

關於這項作業

當您第一次安裝 IBM MQ 並執行 Prepare IBM MQ Wizard 時，它會為稱為 MUSR_MQADMIN 的服務建立本端使用者帳戶。 對於後續安裝， Prepare IBM MQ Wizard 會建立名為 MUSR_MQADMIN*x* 的使用者帳戶，其中 *x* 是下一個可用的號碼，代表不存在的使用者 ID。

您可能需要將與 IBM MQ 服務相關聯的使用者名稱從 MUSR_MQADMIN 或 MUSR_MQADMIN*x* 變更為其他名稱。 例如，如果佇列管理程式與 Db2 相關聯，則您可能需要執行此動作，因為不接受超過 8 個字元的使用者名稱。

程序

1. 建立新的使用者帳戶 (例如 **NEW_NAME**)
2. 使用 Prepare IBM MQ Wizard 來輸入新使用者帳戶的詳細資料。

相關工作

使用 Prepare IBM MQ Wizard 來配置 IBM MQ

Windows 變更 *IBM MQ Windows* 服務本端使用者帳戶的密碼

您可以使用「電腦管理」畫面來變更 IBM MQ Windows 服務本端使用者帳戶的密碼。

關於這項作業

若要變更 IBM MQ Windows 服務本端使用者帳戶的密碼，請執行下列步驟:

程序

1. 識別執行服務的使用者。
2. 從「電腦管理」畫面停止 IBM MQ 服務。
3. 變更所需密碼的方式與您變更個人密碼的方式相同。
4. 從「電腦管理」畫面移至 IBM MQ 服務的內容。
5. 選取 **登入** 頁面。
6. 請確認指定的帳戶名稱符合已修改密碼的使用者。
7. 在 **密碼** 和 **確認密碼** 欄位中鍵入密碼，然後按一下 **確定**。

Windows 針對以網域使用者帳戶執行的安裝，變更 *IBM MQ Windows* 服務的密碼

除了使用 Prepare IBM MQ Wizard 來輸入網域使用者帳戶的帳戶詳細資料之外，您也可以使用「電腦管理」畫面來變更安裝特定 IBM MQ 服務的 **登入** 詳細資料。

關於這項作業

如果安裝的 IBM MQ Windows 服務在網域使用者帳戶下執行，您可以變更帳戶的密碼，如下所示:

程序

1. 變更網域控制站上網域帳戶的密碼。 您可能需要要求網域管理者為您執行此動作。
2. 完成下列步驟，以修改 IBM MQ 服務的「 **登入** 」頁面。
   a) 識別執行服務的使用者。
   b) 從「電腦管理」畫面停止 IBM MQ 服務。
   c) 變更所需密碼的方式與您變更個人密碼的方式相同。
   d) 從「電腦管理」畫面移至 IBM MQ 服務的內容。
   e) 選取 **登入** 頁面。
   f) 請確認指定的帳戶名稱符合已修改密碼的使用者。
   g) 在 **密碼** 和 **確認密碼** 欄位中鍵入密碼，然後按一下 **確定**。

   執行 IBM MQ Windows 服務的使用者帳戶會執行使用者介面應用程式所發出的任何 MQSC 指令，或在系統啟動、關閉或服務回復時自動執行的任何 MQSC 指令。 因此，此使用者帳戶必須具有 IBM MQ 管理權限。 依預設，它會新增至伺服器上的本端 mqm 群組。 如果移除此成員資格，則 IBM MQ Windows 服務無法運作。 如需使用者權限的相關資訊，請參閱 第 124 頁的『IBM MQ Windows 服務所需的使用者權限』。

   如果執行 IBM MQ Windows 服務的使用者帳戶發生安全問題，則錯誤訊息及說明會出現在系統事件日誌中。

相關工作

使用 Prepare IBM MQ Wizard 來配置 IBM MQ

Windows 將 *Windows* 伺服器升級至網域控制站時的考量

將 Windows 伺服器升級至網域控制站時，您應該考量與使用者和群組許可權相關的安全設定是否適當。 在伺服器與網域控制站之間變更 Windows 機器的狀態時，您應該考量這可能會影響 IBM MQ 的作業，因為 IBM MQ 使用本端定義的 mqm 群組。

與網域使用者和群組許可權相關的安全設定

IBM MQ 依賴群組成員資格資訊來實作其安全原則，這表示執行 IBM MQ 作業的使用者 ID 可以決定其他使用者的群組成員資格，這很重要。

當您將 Windows 伺服器升級至網域控制站時，您會看到與使用者和群組許可權相關的安全設定選項。 這個選項控制任意使用者是否能夠從作用中目錄擷取群組成員資格。 如果設定網域控制站，以便本端帳戶有權查詢網域使用者帳戶的群組成員資格，則 IBM MQ 在安裝程序期間建立的預設使用者 ID 可以根據需要取得其他使用者的群組成員資格。 不過，如果設定網域控制站，使得本端帳戶無權查詢網域使用者帳戶的群組成員資格，則這會阻止 IBM MQ 完成其檢查，確認在網域上定義的使用者已獲授權存取佇列管理程式或佇列，且存取失敗。 如果您在以這種方式設定的網域控制站上使用 Windows ，則必須使用具有必要許可權的特殊網域使用者帳戶。

在此情況下，您需要知道:

- Windows 版本的安全許可權行為方式。
- 如何容許網域 mqm 群組成員讀取群組成員資格。
- 如何將 IBM MQ Windows 服務配置為在網域使用者下執行。

如需相關資訊，請參閱 配置 IBM MQ 的使用者帳戶。

IBM MQ 對本端 mqm 群組的存取權

當 Windows 伺服器升級至網域控制站或從網域控制站降級時， IBM MQ 會失去本端 mqm 群組的存取權。

當伺服器提升為網域控制站時，範圍會從本端變更為本端網域。 當機器降級至伺服器時，會移除所有網域本端群組。 這表示將機器從伺服器變更為網域控制站，然後再變更回伺服器會失去本端 mqm 群組的存取權。 症狀是指出缺少本端 mqm 群組的錯誤，例如:

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

若要補救此問題，請使用標準 Windows 管理工具重建本端 mqm 群組。 因為遺失所有群組成員資格資訊，所以您必須在新建立的本端 mqm 群組中恢復特許 IBM MQ 使用者。 如果機器是網域成員，則還必須將網域 mqm 群組新增至本端 mqm 群組，以授與特許網域 IBM MQ 使用者 ID 必要的權限層次。

Windows *Windows* 上巢狀群組的限制

使用巢狀群組有一些限制。 這些結果部分來自網域功能層次，部分來自 IBM MQ 限制。

視「網域」功能層次而定， Active Directory 可以支援「網域」環境定義內的不同群組類型。 依預設，Windows 2003 網域位於 " Windows 2000 混合功能層次。 (Windows Server 2008 和 Windows Server 2012 遵循 Windows 2003 網域模型。) 網域功能層次決定在網域環境中配置使用者 ID 時容許的受支援群組類型及巢狀層次。 如需群組範圍及併入準則的詳細資料，請參閱 Active Directory 文件。

除了 Active Directory 需求之外，還會對 IBM MQ 使用的 ID 施加進一步限制。 IBM MQ 使用的網路 API 不支援網域功能層次所支援的所有配置。 因此， IBM MQ 無法查詢「網域本端」群組中呈現的任何「網域 ID」的群組成員資格，該群組隨後會巢套在本端群組中。 此外，不支援多個巢狀內嵌廣域及通用群組。 不過，支援立即巢狀的廣域或通用群組。

Windows 授權使用者從遠端使用 *IBM MQ*

如果您需要在遠端連接至 IBM MQ 時建立並啟動佇列管理程式，則必須具有 **建立廣域物件** 使用者存取權。

關於這項作業

註: 依預設，管理者具有 **建立廣域物件** 使用者存取權，因此如果您是管理者，則可以在遠端連接時建立並啟動佇列管理程式，而無需變更您的使用者權限。

如果您使用「終端機服務」或「遠端桌面連線」連接至 Windows 機器，且您在建立、啟動或刪除佇列管理程式時發生問題，這可能是因為您沒有使用者存取權 **建立廣域物件**。

建立廣域物件 使用者存取權會限制獲授權在廣域名稱空間中建立物件的使用者。 為了讓應用程式建立廣域物件，它必須在廣域名稱空間中執行，或執行應用程式的使用者必須已套用 **建立廣域物件** 使用者存取權。

當您使用「終端機服務」或「遠端桌面連線」從遠端連接至 Windows 機器時，應用程式會在自己的本端名稱空間中執行。 如果您嘗試使用 IBM MQ Explorer 或 **crtmqm** 或 **dltmqm** 指令來建立或刪除佇列管理程式，或使用 **strmqm** 指令來啟動佇列管理程式，則會導致授權失效。 這會建立具有探測 ID XY132002 的 IBM MQ FDC。

使用「 IBM MQ Explorer」或使用 **amqmdain qmgr start** 指令來啟動佇列管理程式會正確運作，因為這些指令不會直接啟動佇列管理程式。 相反地，指令會將啟動佇列管理程式的要求傳送至在廣域名稱空間中執行的個別處理程序。

當您使用終端機服務時，如果各種管理 IBM MQ 的方法都無法運作，請嘗試設定 **建立廣域物件** 使用者權限。

程序

1. 開啟「系統管理工具」畫面:

   Windows Server 2008 及 Windows Server 2012
   使用 **控制台** > **系統和維護** > **系統管理工具**來存取此畫面。

   Windows 8.1
   使用 **系統管理工具** > **電腦管理** 存取此畫面
2. 按兩下**本機安全性原則**。
3. 展開 **本機原則**。
4. 按一下**使用者權限指派**。
5. 將新的使用者或群組新增至 **建立廣域物件** 原則。

Windows *Windows* 上的 *SSPI* 通道結束程式

IBM MQ for Windows 提供可在訊息及 MQI 通道上使用的安全結束程式。 結束程式作為來源及物件程式碼提供，並提供單向及雙向鑑別。

安全結束程式使用「安全支援提供者介面 (SSPI)」，其提供 Windows 平台的整合安全機能。

安全結束程式提供下列識別及鑑別服務:

單向鑑別 (one way authentication)
這會使用 Windows NT LAN Manager (NTLM) 鑑別支援。 NTLM 容許伺服器鑑別其用戶端。 它不容許用戶端鑑別伺服器，或一個伺服器鑑別另一個伺服器。 NTLM 是針對網路環境而設計，其中假設伺服器是真實的。 IBM WebSphere MQ 7.0 支援的所有 Windows 平台都支援 NTLM。

此服務通常在 MQI 通道上使用，讓伺服器佇列管理程式能夠鑑別 IBM MQ MQI client 應用程式。 用戶端應用程式由與執行中處理程序相關聯的使用者 ID 識別。

為了執行鑑別，通道用戶端的安全結束程式會從 NTLM 取得鑑別記號，並將安全訊息中的記號傳送至通道另一端的夥伴。 夥伴安全結束程式會將記號傳遞至 NTLM ，這會檢查記號是否真實。 如果夥伴安全結束程式不滿意記號的確實性，它會指示 MCA 關閉通道。

雙向或交互鑑別
這會使用 Kerberos 鑑別服務。 Kerberos 通訊協定不假設網路環境中的伺服器是真實的。 伺服器可以鑑別用戶端及其他伺服器，而用戶端可以鑑別伺服器。 在 IBM WebSphere MQ 7.0 支援的所有 Windows 平台上都支援 Kerberos 。

此服務可以在訊息及 MQI 通道上使用。 在訊息通道上，它提供兩個佇列管理程式的交互鑑別。 在 MQI 通道上，它可讓伺服器佇列管理程式及 IBM MQ MQI client 應用程式彼此鑑別。 佇列管理程式由字首為字串 `ibmMQSeries/`的名稱來識別。 用戶端應用程式由與執行中處理程序相關聯的使用者 ID 識別。

為了執行交互鑑別，起始安全結束程式會從 Kerberos 安全伺服器獲得鑑別記號，並將安全訊息中的記號傳送給其夥伴。 夥伴安全結束程式會將記號傳遞至 Kerberos 伺服器，伺服器會檢查記號是否真實。 Kerberos 安全伺服器會產生第二個記號，夥伴會在安全訊息中傳送給起始安全結束程式。 然後起始安全結束程式會要求 Kerberos 伺服器檢查第二個記號是否真實。 在此交換期間，如果任一安全結束程式不滿意另一個安全結束程式所傳送記號的確實性，則會指示 MCA 關閉通道。

以來源及物件格式提供安全結束程式。 您可以使用原始碼作為起始點來撰寫您自己的通道結束程式，也可以使用所提供的物件模組。 物件模組有兩個進入點，一個用於使用 NTLM 鑑別支援進行單向鑑別，另一個用於使用 Kerberos 鑑別服務進行雙向鑑別。

如需 SSPI 通道結束程式如何運作的相關資訊，以及如何實作它的指示，請參閱 在 Windows 系統上使用 SSPI 安全結束程式。

Windows 在 *Windows* 上套用安全範本檔案

套用範本可能會影響套用至 IBM MQ 檔案及目錄的安全設定。 如果您使用高度安全的範本，請先套用它，然後再安裝 IBM MQ。

Windows 支援文字型安全範本檔案，您可以使用這些範本檔案，將統一安全設定套用至具有安全配置及分析 MMC 嵌入式管理單元的一部以上電腦。 特別是， Windows 提供數個範本，其中包括一系列安全設定，以提供特定安全層次。 這些範本包括「相容」、「安全」及「高度安全」。

套用其中一個範本可能會影響套用至 IBM MQ 檔案和目錄的安全設定。 如果您要使用「高度安全」範本，請先配置您的機器，然後再安裝 IBM MQ。

如果您將高度安全的範本套用至已安裝 IBM MQ 的機器，則會移除您在 IBM MQ 檔案及目錄上設定的所有許可權。 因為已移除這些許可權，所以您會失去 *Administrator*、 *mqm* 及 *Everyone* 群組從錯誤目錄的存取權 (如果適用的話)。

Windows 為連接至 *IBM MQ* 的 *Windows* 應用程式配置額外權限

執行 IBM MQ 處理程序的帳戶可能需要額外授權，才能授與對應用程式程序的 SYNCHRONIZE 存取權。

關於這項作業

如果您有 Windows 應用程式 (例如 ASP 頁面) 連接至配置為在高於平常的安全層次執行的 IBM MQ ，則可能會遇到問題。

IBM MQ 需要應用程式程序的「同步化」存取權，才能協調特定動作。 當伺服器應用程式第一次嘗試連接至佇列管理程式時， IBM MQ 會修改程序以授與 SYNCHRONIZE 權限給 IBM MQ 管理者。 不過，執行 IBM MQ 處理程序所使用的帳戶可能需要其他授權，才能授與所要求的存取權。

若要配置對執行 IBM MQ 處理程序之使用者 ID 的其他權限，請完成下列步驟:

程序

1. 啟動「本機安全性原則」工具，按一下 **安全性設定->本機原則->使用者權限指派**，然後按一下 **程式除錯**。
2. 按兩下 **程式除錯**，然後將您的 IBM MQ 使用者 ID 新增至清單

   如果系統位於 Windows 網域中，且仍未設定有效原則設定，則即使已設定本端原則設定，也必須使用「網域安全原則」工具，以相同的方式在網域層次授權使用者 ID。

IBM i 在 IBM i 上設定安全

IBM i 上的安全是使用「 IBM MQ 物件權限管理程式 (OAM)」及 IBM i 物件層次安全來實作。

在決定對 IBM MQ 物件的存取權時必須考量的安全。

對企業中的使用者設定權限時，您需要考量下列要點:

1. 使用 IBM i GRTOBJAUT 及 RVKOBJAUT 指令來授與及撤銷 IBM MQ for IBM i 指令的權限。

   在 QMQM 檔案庫中，某些非指令 (* cmd) 物件設為具有 ***USE** 的 ***PUBLIC** 權限。 請勿變更這些物件的權限，或使用授權清單來提供權限。 任何不正確的權限可能危及 IBM MQ 功能。
2. 在安裝 IBM MQ for IBM i 期間，會建立下列特殊使用者設定檔:

   QMQM
   主要用於內部僅限產品功能。 不過，它可用來使用 MQCNO_FASTPATH_BINDINGS 來執行授信應用程式。 請參閱 使用 MQCONNX 呼叫來連接佇列管理程式。

   QMQMADM
   用作 IBM MQ 管理者的群組設定檔。 群組設定檔可讓您存取 CL 指令及 IBM MQ 資源。

   使用 SBMJOB 提交呼叫 IBM MQ 指令的程式時， USER 不得明確設為 QMQMADM。 相反地，請將 USER 設為 QMQM ，或將 QMQMADM 指定為群組的另一個使用者設定檔。
3. 如果您要將通道指令傳送至遠端佇列管理程式，請確定您的使用者設定檔是目標系統上群組 QMQMADM 的成員。 如需 PCF 及 MQSC 通道指令的清單，請參閱 IBM MQ for IBM i CL 指令。
4. 當 OAM 計算群組授權時，會快取與使用者相關聯的群組集。

   在快取群組集之後對使用者群組成員資格所做的任何變更，在您重新啟動佇列管理程式或執行 RFRMQMAUT 以重新整理安全之前，都無法辨識。
5. 限制有權使用特別機密指令的使用者數目。 這些指令包括:
   - 建立訊息佇列管理程式 (CRTMQM)
   - 刪除訊息佇列管理程式 (DLTMQM)
   - 啟動訊息佇列管理程式 (STRMQM)
   - 結束訊息佇列管理程式 (ENDMQM)
   - 啟動指令伺服器 (STRMQMCSVR)
   - 結束指令伺服器 (ENDMQMCSVR)
6. 通道定義包含安全結束程式規格。 通道建立及修改需要特殊考量。 第 89 頁的『安全結束程式概觀』中提供安全結束程式的詳細資料。
7. 可以替代通道結束程式及觸發監視器程式。 這類替換項目的安全是程式設計師的責任。

IBM i IBM i 上的物件權限管理程式

物件權限管理程式 (OAM) 會管理使用者操作 IBM MQ 物件 (包括佇列及程序定義) 的授權。 它也提供指令介面，您可以透過它來授與或撤銷特定使用者群組的物件存取權。 容許存取資源的決策由 OAM 做出，且佇列管理程式遵循該決策。 如果 OAM 無法做出決策，佇列管理程式會阻止存取該資源。

透過 OAM ，您可以控制:

- 透過 MQI 存取 IBM MQ 物件。 當應用程式嘗試存取物件時， OAM 會檢查提出要求的使用者設定檔是否具有所要求作業的授權。

  特別是，這表示佇列及佇列上的訊息可以受到保護，不會遭到未獲授權的存取。
- 使用 PCF 及 MQSC 指令的許可權。

不同的使用者群組可以對相同的物件具有不同的存取權。 例如，對於特定佇列，一個群組可以同時執行 put 及 get 作業; 另一個群組只能瀏覽佇列 (具有瀏覽選項的 MQGET)。 同樣地，部分群組可能具有佇列的取得及放置權限，但不容許變更或刪除佇列。

IBM MQ for IBM i 指令及對 IBM MQ for IBM i 物件執行作業

IBM i IBM i 上的 IBM MQ 權限

若要存取 IBM MQ 物件，您需要權限才能發出指令及存取參照的物件。 管理者有權存取所有 IBM MQ 資源。

IBM MQ 物件的存取權由下列權限控制:

1. 發出 IBM MQ 指令
2. 存取指令所參照的 IBM MQ 物件

所有 IBM MQ for IBM i CL 指令都隨附於 QMQM 的擁有者，且管理設定檔 (QMQMADM) 具有 *USE 權限，且 *PUBLIC 存取權設為 *EXCLUDE。

註: IBM MQ for IBM i 授權程式安裝程式使用 QSRDUPER 程式複製 QSYS 中的指令 (*CMD) 物件。 在 IBM i V5R4 以及更新版本中， QSRDUPER 程式已變更，因此預設行為是建立 Proxy 指令，而不是複製原始指令。 Proxy 指令會將指令執行重新導向至另一個指令，且屬性為 PRX。 如果檔案庫 QSYS 中存在與所複製指令同名的 Proxy 指令，則不會將 Proxy 指令的專用權限授與產品檔案庫中的指令。 嘗試在 QSYS 中提示或執行 Proxy 指令，請檢查產品檔案庫中目標指令的權限。 因此，必須在產品檔案庫 (QMQM) 中完成對 *CMD 物件的任何權限變更，且不需要修改 QSYS 中的那些變更。 例如:

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

如果您對 IBM MQ 物件具有進行這些變更所需的 OAM 權限，則部分產品 CL 指令的權限結構變更容許公開使用這些指令。

若要成為 IBM i 上的 IBM MQ 管理者，您必須是 *QMQMADM* 群組的成員。 此群組具有類似 AIX, Linux, and Windows 系統上 mqm 群組的內容。 特別是當您安裝 IBM MQ for IBM i 時，會建立 QMQMADM 群組，且 QMQMADM 群組的成員可以存取系統上的所有 IBM MQ 資源。 如果您具有 *ALLOBJ 權限，則也可以存取所有 IBM MQ 資源。

管理者可以使用 CL 指令來管理 IBM MQ。 其中一個指令是 GRTMQMAUT，用來授與權限給其他使用者。 另一個指令 STRMQMMQSC 可讓管理者對本端佇列管理程式發出 MQSC 指令。

相關概念

第 73 頁的『在 IBM i 上管理 IBM MQ 的權限』

IBM i *IBM i 上 IBM MQ 物件的存取權*

執行 IBM MQ CL 指令所需的存取權。

IBM MQ for IBM i 將產品的 CL 指令分類為兩個群組:

群組 1

使用者必須在 QMQMADM 使用者群組中，或具有 *ALLOBJ 權限，才能處理這些指令。 具有這些權限之一的使用者可以處理所有種類中的所有指令，而不需要任何額外權限。

註: 這些權限會置換任何 OAM 權限。

這些指令可以分組如下:

- 指令伺服器指令
  - ENDMQMCSVR，結束 IBM MQ 指令伺服器
  - STRMQCSVR，啟動 IBM MQ 指令伺服器
- 無法傳送郵件的佇列處理程式指令
  - STRMQMDLQ，啟動 IBM MQ 無法傳送郵件的佇列處理程式
- 接聽器指令
  - ENDMQMLSR，結束 IBM MQ 接聽器
  - STRMQMLSR，啟動非物件接聽器
- 媒體回復指令
  - RCDMQMIMG，記錄 IBM MQ 物件影像
  - RCRMQMOBJ，重建 IBM MQ 物件
  - WRKMQMTRN，使用 IBM MQ Q 交易
- 佇列管理程式指令
  - CRTMQM，建立訊息佇列管理程式
  - DLTMQM，刪除訊息佇列管理程式

- ENDMQM，結束訊息佇列管理程式
- STRMQM，啟動訊息佇列管理程式

- 安全指令
  - GRTMQMAUT，授與 IBM MQ 物件權限
  - RVKMQMAUT，撤銷 IBM MQ 物件權限
- 追蹤指令
  - TRCMQM，追蹤 IBM MQ 工作
- 異動指令
  - RSVMQMTRN，解析 IBM MQ 交易
- 觸發監視器指令
  - STRMQMTRM，啟動觸發監視器
- IBM MQSC 指令
  - RUNMQSC，執行 IBM MQSC 指令
  - STRMQMQSC，啟動 IBM MQSC 指令

群組 2

其餘指令，需要兩個層次的權限:

1. 執行指令的 IBM i 權限。 IBM MQ 管理者使用 **GRTOBJAUT** 指令來設定此項，以置換使用者或使用者群組的 *PUBLIC (*EXCLUDE) 限制。

   例如:

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

2. IBM MQ 權限，以在步驟 1 中獲得正確的 IBM i 權限，來操作與指令相關聯的 IBM MQ 物件。

   此權限由具有必要動作之適當 OAM 權限的使用者控制，由 IBM MQ 管理者使用 **GRTMQMAUT** 指令設定。

   例如:

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

指令可以分組如下:

- 通道指令
  - CHGMQMCHL，變更 IBM MQ 通道

    這需要 * 對佇列管理程式的連接權限，以及 * 對通道的 admchg 權限。
  - CPYMQMCHL，複製 IBM MQ 通道

    這需要對佇列管理程式的 * connect 和 * admcrt 權限，對要複製的預設通道類型的 * admdsp 權限，以及對通道物件類別的 * admcrt 權限。

    例如，複製「傳送端」通道需要 * admdsp 權限至 SYSTEM.DEF.SENDER 通道
  - CRTMQMCHL，建立 IBM MQ 通道

    這需要對佇列管理程式的 * connect 和 * admcrt 權限，對要建立的預設通道類型的 * admdsp 權限，以及對通道物件類別的 * admcrt 權限。

    例如，建立「傳送端」通道需要對 SYSTEM.DEF.SENDER 通道
  - DLTMQMCHL，刪除 IBM MQ 通道

    這需要 * 對佇列管理程式的連接權限，以及 * 對通道的 admdlt 權限。
  - RSVMQMCHL，解析 IBM MQ 通道

這需要 * 對佇列管理程式的連接權限，以及 * 對通道的 ctrlx 權限。

- 顯示指令

  若要處理 DSP 指令，您必須授與使用者 *connect 及 *admdsp 對佇列管理程式的權限，以及列出的任何特定選項:

  – DSPMQM，顯示訊息佇列管理程式
  – DSPMQMAUT，顯示 IBM MQ 物件權限
  – DSPMQMAUTI，顯示 IBM MQ 鑑別資訊- *admdsp 至鑑別資訊物件
  – DSPMQMCHL，顯示通道的 IBM MQ 通道- *admdsp
  – DSPMQMCSVR，顯示 IBM MQ 指令伺服器
  – DSPMQMNL，顯示 IBM MQ 名單- *admdsp 至名單
  – DSPMQMOBJN，顯示 IBM MQ 物件名稱
  – DSPMQMPRC，顯示 IBM MQ 處理程序- *admdsp 至處理程序
  – DSPMQMQ，顯示 IBM MQ 佇列- *admdsp 至佇列
  – DSPMQMTOP，顯示 IBM MQ 主題- *admdsp 至主題

- 使用指令

  如果要處理 WRK 指令並顯示選項畫面，您必須授與使用者 *connect 和 *admdsp 對佇列管理程式的權限，以及列出的任何特定選項:

  – WRKMQM，使用訊息佇列管理程式
  – WRKMQMAUT，使用 IBM MQ 物件權限
  – WRKMQMAUTD，使用 IBM MQ 物件權限資料
  – WRKMQMAUTI，使用 IBM MQ 鑑別資訊
    - *admchg 代表「變更 IBM MQ 鑑別資訊物件」指令。
    - *admcrt 代表「建立及複製 IBM MQ 鑑別資訊物件」指令。
    - *admdlt 代表「刪除 IBM MQ 鑑別資訊物件」指令。
    - *admdsp 代表「顯示 IBM MQ 鑑別資訊物件」指令。
  – WRKMQMCHL，使用 IBM MQ 通道

    這需要下列權限:

    - *admchg 代表「變更 IBM MQ 通道」指令。
    - *admclr 代表「清除 IBM MQ 通道」指令。
    - *admcrt 代表「建立及複製 IBM MQ 通道」指令。
    - *admdlt 代表「刪除 IBM MQ 通道」指令。
    - *admdsp 代表「顯示 IBM MQ 通道」指令。
    - *ctrl 代表「啟動 IBM MQ 通道」指令。
    - *ctrl 代表「結束 IBM MQ 通道」指令。
    - *ctrl (適用於「連線測試 IBM MQ 通道」指令)。
    - *ctrlx 代表「重設 IBM MQ 通道」指令。
    - *ctrlx for the Resolve IBM MQ Channel 指令。
  – WRKMQMCHST，使用 IBM MQ 通道狀態

    這需要通道的 *admdsp 權限。
  – WRKMQMCL，使用 IBM MQ 叢集
  – WRKMQMCLQ，使用 IBM MQ 叢集佇列
  – WRKMQMCLQM，使用 IBM MQ 叢集佇列管理程式
  – WRKMQMLSR，使用 IBM MQ 接聽器

- WRKMQMMSG，使用 IBM MQ 訊息

  這需要佇列的 *browse 權限
- WRKMQMNL，使用 IBM MQ 名稱清單

  這需要下列權限:

  - *admchg 代表「變更 IBM MQ 名單」指令。
  - *admcrt 代表「建立及複製 IBM MQ 名單」指令。
  - *admdlt 代表「刪除 IBM MQ 名單」指令。
  - *admdsp，用於「顯示 IBM MQ 名單」指令。
- WRKMQMPRC，使用 IBM MQ 處理程序

  這需要下列權限:

  - *admchg，表示「變更 IBM MQ 處理程序」指令。
  - *admcrt 代表「建立及複製 IBM MQ 處理程序」指令。
  - *admdlt 代表「刪除 IBM MQ 處理程序」指令。
  - *admdsp 代表「顯示 IBM MQ 處理程序」指令。
- WRKMQMQ，使用 IBM MQ 佇列

  這需要下列權限:

  - *admchg 代表「變更 IBM MQ 佇列」指令。
  - *admclr 代表「清除 IBM MQ 佇列」指令。
  - *admcrt 代表「建立及複製 IBM MQ 佇列」指令。
  - *admdlt 代表「刪除 IBM MQ 佇列」指令。
  - *admdsp 代表「顯示 IBM MQ 佇列」指令。
- WRKMQMQSTS，使用 IBM MQ 佇列狀態
- WRKMQMTOP，使用 IBM MQ 主題

  這需要下列權限

  - *admchg 表示「變更 IBM MQ 主題」指令。
  - *admcrt 代表「建立及複製 IBM MQ 主題」指令。
  - *admdlt 代表「刪除 IBM MQ 主題」指令。
  - *admdsp 代表「顯示 IBM MQ 主題」指令。
- WRKMQMSUB，使用 IBM MQ 訂閱

- 其他通道指令

  若要處理通道指令，您必須授與使用者列出的特定權限:

  - ENDMQMCHL，結束 IBM MQ 通道

    這需要對佇列管理程式的 *connect 權限，以及對與通道相關聯的傳輸佇列的 *allmqi 權限。
  - ENDMQMLSR，結束 IBM MQ 接聽器

    這需要對佇列管理程式的 *connect 權限，以及對指定接聽器物件的 *ctrl 權限。
  - PNGMQMCHL，連線測試 IBM MQ 通道

    這需要對佇列管理程式的 *connect 及 *inq 權限，以及對通道物件的 *ctrl 權限。
  - RSTMQMCHL，重設 IBM MQ 通道

    這需要佇列管理程式的 *connect 權限。
  - STRMQCHL，啟動 IBM MQ 通道

    這需要對佇列管理程式的 *connect 權限，以及對通道物件的 *ctrl 權限。

- STRMQMCHLI，啟動 IBM MQ 通道起始程式

  這需要對佇列管理程式的 *connect 及 *inq 權限，以及對與通道傳輸佇列相關聯之起始佇列的 *allmqi 權限。
- STRMQLSR，啟動 IBM MQ 接聽器

  這需要 * 對佇列管理程式的連接權限，以及 * 對具名接聽器物件的 ctrl 權限。

- 其他指令:

若要處理下列指令，您必須授與使用者列出的特定權限:

- CCTMQM，連接至訊息佇列管理程式

  這不需要 IBM MQ 物件權限。
- CHGMQM，變更訊息佇列管理程式

  這需要佇列管理程式的 *connect 及 *admchg 權限。
- CHGMQMAUTI，變更 IBM MQ 鑑別資訊

  這需要對佇列管理程式的 *connect 權限，以及對鑑別資訊物件的 *admchg 和 *admdsp 權限。
- CHGMQMNL，變更 IBM MQ 名單

  這需要對佇列管理程式的 *connect 權限，以及對名單的 *admchg 權限。
- CHGMQMPRC，變更 IBM MQ 處理程序

  這需要對佇列管理程式的 *connect 權限，以及對處理程序的 *admchg 權限。
- CHGMQMQ，變更 IBM MQ 佇列

  這需要對佇列管理程式的 *connect 權限，以及對佇列的 *admchg 權限。
- CLRMQMQ，清除 IBM MQ 佇列

  這需要對佇列管理程式的 *connect 權限，以及對佇列的 *admclr 權限。
- CPYMQMAUTI，複製 IBM MQ 鑑別資訊

  這需要對佇列管理程式的 *connect 權限，以及對鑑別資訊物件的 *admdsp 權限，以及對鑑別資訊物件類別的 *admcrt 權限。
- CPYMQMNL，複製 IBM MQ 名單

  這需要佇列管理程式的 *connect 及 *admcrt 權限。
- CPYMQMPRC，複製 IBM MQ 處理程序

  這需要佇列管理程式的 *connect 及 *admcrt 權限。
- CPYMQMQ，複製 IBM MQ 佇列

  這需要佇列管理程式的 *connect 及 *admcrt 權限。
- CRTMQMAUTI，建立 IBM MQ 鑑別資訊

  這需要對佇列管理程式的 *connect 權限，以及對鑑別資訊物件的 *admdsp 權限，以及對鑑別資訊物件類別的 *admcrt 權限。
- CRTMQMNL，建立 IBM MQ 名單

  這需要佇列管理程式的 *connect 及 *admcrt 權限，以及預設名單的 *admdsp 權限。
- CRTMQMPRC，建立 IBM MQ 處理程序

  這需要對佇列管理程式的 *connect 及 *admcrt 權限，以及對預設處理程序的 *admdsp 權限。
- CRTMQMQ，建立 IBM MQ 佇列

  這需要對佇列管理程式的 *connect 及 *admcrt 權限，以及對預設佇列的 *admdsp 權限。
- CVTMQMDTA，轉換 IBM MQ 資料類型指令

  這不需要 IBM MQ 物件權限。
- DLTMQMAUTI，刪除 IBM MQ 鑑別資訊

這需要對佇列管理程式的 *connect 權限，以及對鑑別資訊物件的 *ctrlx 權限。

- DLTMQMNL，刪除 IBM MQ 名單

  這需要對佇列管理程式的 *connect 權限，以及對名單的 *admdlt 權限。
- DLTMQMPRC，刪除 IBM MQ 處理程序

  這需要對佇列管理程式的 *connect 權限，以及對處理程序的 *admdlt 權限。
- DLTMQMQ，刪除 IBM MQ 佇列

  這需要對佇列管理程式的 *connect 權限，以及對佇列的 *admdlt 權限。
- DSCMQM，切斷與訊息佇列管理程式的連線

  這不需要 IBM MQ 物件權限。
- RFRMQMAUT，重新整理安全

  這需要佇列管理程式的 *connect 權限。
- RFRMQMCL，重新整理叢集

  這需要佇列管理程式的 *connect 權限。
- RSMMQMCLQM，回復叢集佇列管理程式

  這需要佇列管理程式的 *connect 權限。
- RSTMQMCL，重設叢集

  這需要佇列管理程式的 *connect 權限。
- SPDMQMCLQM，暫停叢集佇列管理程式

  這需要佇列管理程式的 *connect 權限。

IBM i *IBM i 上的存取授權*

使用此資訊來瞭解存取授權指令。

GRTMQMAUT 及 RVKMQMAUT 指令上 AUT 關鍵字所定義的授權可以分類如下:

- 與 MQI 呼叫相關的授權
- 授權相關管理指令
- 環境定義授權
- 一般授權，亦即，適用於 MQI 呼叫及/或指令

下表使用 AUT 參數來列出 MQI 呼叫、環境定義呼叫、MQSC 及 PCF 指令及一般作業的不同權限。

表 *15: MQI* 呼叫的授權

AUT	**說明**
*ALTUSR	容許另一個使用者的權限用於 MQOPEN 和 MQPUT1 呼叫。
*XX_ENCODE_CASE_ONE Browse	使用 BROWSE 選項發出 MQGET 呼叫，從佇列中擷取訊息。
*CONNECT	透過發出 MQCONN 呼叫，將應用程式連接至指定的佇列管理程式。
*GET	透過發出 MQGET 呼叫，從佇列中擷取訊息。
*INQ	透過發出 MQINQ 呼叫，對特定佇列進行查詢。
*PUB	開啟主題以使用 MQPUT 呼叫來發佈訊息。
*PUT	透過發出 MQPUT 呼叫，將訊息放置在特定佇列上。
*RESUME	使用 MQSUB 呼叫回復訂閱。

表 *15: MQI* 呼叫的授權 *(繼續)*	
AUT	**說明**
*SET	透過發出 MQSET 呼叫，從 MQI 設定佇列上的屬性。 如果您開啟多個選項的佇列，您必須獲得每一個選項的授權。
*SUB	使用 MQSUB 呼叫來建立、變更或回復主題的訂閱。

表 *16:* 環境定義呼叫的授權	
AUT	**說明**
*PASSALL	在指定的佇列上傳遞所有環境定義。 從原始要求複製所有環境定義欄位。
*PASSID	在指定的佇列上傳遞身分環境定義。 身分環境定義與要求的環境定義相同。
*SETALL	設定指定佇列上的所有環境定義。 這是由特殊系統公用程式使用。
*SETID	在指定的佇列上設定身分環境定義。 這是由特殊系統公用程式使用。

表 *17: MQSC* 及 *PCF* 呼叫的授權	
AUT	**說明**
*ADMCHG	變更指定物件的屬性。
*ADMCLR	清除指定的物件 (僅限 PCF 清除物件指令)。
*ADMCRT	建立指定類型的物件。
*ADMDLT	刪除指定的物件。
*ADMDSP	顯示指定物件的屬性。

表 *18:* 一般作業的授權	
AUT	**說明**
*ALL	使用適用於物件的所有作業。 all 權限相當於適用於物件類型之權限 alladm、allmqi 及 system 的聯集。
*ALLADM	執行適用於物件的所有管理作業。
*ALLMQI	使用適用於物件的所有 MQI 呼叫。
*CTRL	控制通道、接聽器及服務的啟動及關閉。
*CTRLX	重設序號並解決不確定的通道。

IBM i 在 *IBM i* 上使用存取授權指令

使用此資訊來瞭解存取授權指令，並使用指令範例。

使用 GRTMQMAUT 指令

如果您具有必要的授權，則可以使用 GRTMQMAUT 指令來授與使用者設定檔或使用者群組存取特定物件的權限。 下列範例說明如何使用 GRTMQMAUT 指令:

1. ```
   GRTMQMAUT OBJ(RED.LOCAL.QUEUE) OBJTYPE(*LCLQ) USER(GROUPA) +
             AUT(*BROWSE *PUT) MQMNAME('saturn.queue.manager')
   ```

   在此範例中:

   - RED.LOCAL.QUEUE 是物件名稱。
   ```

- *LCLQ (本端佇列) 是物件類型。
- GROUPA 是系統上要變更其授權的使用者設定檔名稱。 此設定檔可用作其他使用者的群組設定檔。
- *BROWSE 和 *PUT 是授與指定佇列的授權。

  *BROWSE 會新增授權來瀏覽佇列上的訊息 (使用瀏覽選項發出 MQGET)。

  *PUT 會新增佇列上放置 (MQPUT) 訊息的授權。
- saturn.queue.manager 是佇列管理程式名稱。

2. 下列指令會將預設佇列管理程式的所有適用授權授與使用者 JACK 及 JILL。

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*PRC) USER(JACK JILL) AUT(*ALL)
```

3. 下列指令授與使用者 GEORGE 權限，可以將訊息放置在佇列管理程式 TRENT 的佇列 ORDERS 上。

```
GRTMQMAUT OBJ(TRENT) OBJTYPE(*MQM) USER(GEORGE) AUT(*CONNECT) MQMNAME (TRENT)
GRTMQMAUT OBJ(ORDERS) OBJTYPE(*Q) USER(GEORGE) AUT(*PUT) MQMNAME (TRENT)
```

使用 RVKMQMAUT 指令

如果您具有必要的授權，則可以使用 RVKMQMAUT 指令來移除先前授與使用者設定檔或使用者群組存取特定物件的授權。 下列範例說明如何使用 RVKMQMAUT 指令:

1. ```
   RVKMQMAUT OBJ(RED.LOCAL.QUEUE) OBJTYPE(*LCLQ) USER(GROUPA) +
   AUT(*PUT) MQMNAME('saturn.queue.manager')
   ```

   已針對 GROUPA 移除前一個範例中所授與之將訊息放入指定佇列的權限。

2. ```
   RVKMQMAUT OBJ(PAY*) OBJTYPE(*Q) USER(*PUBLIC) AUT(*GET) +
   MQMNAME(PAYROLLQM)
   ```

   從任何佇列中取得訊息 (名稱以字元 PAY 開頭，由佇列管理程式 PAYROLLQM 所擁有) 的權限，會從系統的所有使用者中移除，除非他們或他們所屬的群組已個別獲得授權。

使用 DSPMQMAUT 指令

顯示 MQM 權限 (DSPMQMAUT) 指令會針對指定的物件和使用者，顯示使用者對該物件的授權清單。 下列範例說明如何使用指令:

```
DSPMQMAUT OBJ(ADMINNL) OBJTYPE(*NMLIST) USER(JOE) OUTPUT(*PRINT) +
MQMNAME(ADMINQM)
```

使用 RFRMQMAUT 指令

重新整理 MQM 安全 (RFRMQMAUT) 指令可讓您立即更新 OAM 的授權群組資訊，以反映在作業系統層次所做的變更，而不需要停止並重新啟動佇列管理程式。 下列範例說明如何使用指令:

```
RFRMQMAUT MQMNAME(ADMINQM)
```

IBM i IBM i 上的授權規格表格

使用此資訊來判定使用特定 API 呼叫所需的授權，以及那些呼叫的特定選項、佇列物件、處理程序物件及佇列管理程式物件。

從 第 139 頁的表 19 開始的授權規格表格會精確定義授權的運作方式，以及適用的限制。 這些表格適用於下列狀況:

- 發出 MQI 呼叫的應用程式
- 以跳出 PCF 形式發出 MQSC 指令的管理程式
- 發出 PCF 指令的管理程式

在此區段中，資訊會呈現為一組指定下列資料的表格:

要執行的動作
MQI 選項、MQSC 指令或 PCF 指令。

存取控制物件
佇列、程序定義、佇列管理程式、名單、通道、用戶端連線通道、接聽器、服務或鑑別資訊物件。

需要授權
以 MQZAO_ 常數表示。

在表格中，字首為 MQZAO_ 的常數對應於特定實體之 **GRTMQMAUT** 及 **RVKMQMAUT** 指令授權清單中的關鍵字。 例如， MQZAO_BROWSE 對應於關鍵字 *BROWSE ; 同樣地，關鍵字 MQZAO_SET_ALL_CONTEXT 對應於關鍵字 *SETALL，依此類推。 這些常數定義在產品隨附的標頭檔 cmqzc.h 中。

MQI 授權

只有在執行應用程式的使用者 ID (或其授權可以假設) 已獲授與相關授權時，才容許應用程式發出特定的 MQI 呼叫及選項。

四個 MQI 呼叫需要授權檢查 :MQCONN、MQOPEN、 MQPUT1 及 MQCLOSE。

對於 MQOPEN 和 MQPUT1，會對所開啟物件的名稱進行權限檢查，而不是對名稱進行權限檢查，在解析名稱之後所產生的名稱。 例如，可以授與應用程式開啟別名佇列的權限，而不具有開啟別名所解析的基本佇列的權限。 規則是除非直接開啟佇列管理程式別名定義，否則會對在名稱解析 (不是佇列管理程式別名) 處理程序期間所發現的第一個定義執行檢查; 亦即，其名稱會出現在物件描述子的 *ObjectName* 欄位中。 所開啟的特定物件一律需要權限; 在某些情況下，需要透過佇列管理程式物件的授權取得其他與佇列無關的權限。

第 139 頁的表 19、 第 139 頁的表 20、 第 140 頁的表 21 和 第 141 頁的表 22 彙總每一個呼叫所需的授權。

註: 這些表格未提及名稱清單、通道、用戶端連線通道、接聽器、服務或鑑別資訊物件。 這是因為除了 MQOO_INQUIRE 之外，沒有任何授權適用於這些物件，其適用的授權與適用於其他物件的授權相同。

表 *19: MQCONN* 呼叫所需的安全授權

需要授權:	**佇列物件 (第 141 頁的『1』)**	**程序物件**	**佇列管理程式物件**
MQCONN 選項	不適用	不適用	MQZAO_CONNECT

表 *20: MQOPEN* 呼叫所需的安全授權

需要授權:	**佇列物件 (第 141 頁的『1』)**	**程序物件**	**佇列管理程式物件**
MQOO_INQUIRE	MQZAO_INQUIRE **(第 141 頁的『2』)**	MQZAO_INQUIRE **(第 141 頁的『2』)**	MQZAO_INQUIRE **(第 141 頁的『2』)**
MQ 瀏覽	MQ 導覽 _ 瀏覽	不適用	不檢查
MQOO_INPUT_ *	MQZAO_ 輸入	不適用	不檢查
MQOO_SAVE_ ALL_CONTEXT **(第 141 頁的『3』)**	MQZAO_ 輸入	不適用	不適用
MQOO_OUTPUT (正常佇列) **(第 141 頁的『4』)**	MQZAO_OUTPUT	不適用	不適用
MQOO_PASS_ IDENTITY_CONTEXT **(第 141 頁的『5』)**	MQZAO_PASS_ IDENTITY_CONTEXT	不適用	不檢查

表 *20: MQOPEN* 呼叫所需的安全授權 *(*繼續*)*			
需要授權:	**佇列物件 (第 141 頁的『1』)**	**程序物件**	**佇列管理程式物件**
MQOO_PASS_ALL_CONTEXT **(第 141 頁的『5』， 第 141 頁的『6』)**	MQZAO_PASS_ALL_CONTEXT	不適用	不檢查
MQOO_SET_IDENTITY_CONTEXT **(第 141 頁的『5』， 第 141 頁的『6』)**	MQZAO_SET_IDENTITY_CONTEXT	不適用	MQZAO_SET_IDENTITY_CONTEXT **(第 141 頁的『7』)**
MQOO_SET_ALL_CONTEXT **(第 141 頁的『5』， 第 141 頁的『8』)**	MQZAO_SET_ALL_CONTEXT	不適用	MQZAO_SET_ALL_CONTEXT **(第 141 頁的『7』)**
MQOO_OUTPUT (傳輸佇列) **(第 141 頁的『9』)**	MQZAO_SET_ALL_CONTEXT	不適用	MQZAO_SET_ALL_CONTEXT **(第 141 頁的『7』)**
MQOO_SET	MQZAO_SET	不適用	不檢查
MQOO_ALTERNATE_USER_AUTHORITY	**(第 141 頁的『10』)**	**(第 141 頁的『10』)**	MQZAO_ALTERNATE_USER_AUTHORITY **(第 141 頁的『10』， 第 141 頁的『11』)**

表 *21: MQPUT1* 呼叫所需的安全授權			
需要授權:	**佇列物件 (第 141 頁的『1』)**	**程序物件**	**佇列管理程式物件**
MQPMO_PASS_IDENTITY_CONTEXT	MQZAO_PASS_IDENTITY_CONTEXT **(第 141 頁的『12』)**	不適用	不檢查
MQPMO_PASS_ALL_CONTEXT	MQZAO_PASS_ALL_CONTEXT **(第 141 頁的『12』)**	不適用	不檢查
MQPMO_SET_IDENTITY_CONTEXT	MQZAO_SET_IDENTITY_CONTEXT **(第 141 頁的『12』)**	不適用	MQZAO_SET_IDENTITY_CONTEXT **(第 141 頁的『7』)**
MQPMO_SET_ALL_CONTEXT	MQZAO_SET_ALL_CONTEXT **(第 141 頁的『12』)**	不適用	MQZAO_SET_ALL_CONTEXT **(第 141 頁的『7』)**
(傳輸佇列) **(第 141 頁的『9』)**	MQZAO_SET_ALL_CONTEXT	不適用	MQZAO_SET_ALL_CONTEXT **(第 141 頁的『7』)**
MQPMO_ALTERNATE_USER_AUTHORITY	**(第 141 頁的『13』)**	不適用	MQZAO_ALTERNATE_USER_AUTHORITY **(第 141 頁的『11』)**

表 *22: MQCLOSE* 呼叫所需的安全授權			
需要授權:	**佇列物件 (第 141 頁的『1』)**	**程序物件**	**佇列管理程式物件**
MQCO_DELETE	MQZAO_DELETE **(第 141 頁的『14』)**	不適用	不適用
MQCO_DELETE _PURGE	MQZAO_DELETE **(第 141 頁的『14』)**	不適用	不適用

表格注意事項:

1. 如果正在開啟模型佇列:
   - 除了為您開啟的存取權類型開啟模型佇列的權限之外，還需要模型佇列的 MQZAO_DISPLAY 權限。
   - 不需要 MQZAO_CREATE 權限即可建立動態佇列。
   - 用來開啟模型佇列的使用者 ID 會自動授與所建立動態佇列的所有佇列特定權限 (相當於 MQZAO_ALL)。
2. 視開啟的物件類型而定，會檢查佇列、處理程序、名單或佇列管理程式物件。
3. 也必須指定 MQOO_INPUT_ *。 此選項適用於本端、模型或別名佇列。
4. 此檢查會針對所有輸出觀察值執行，但附註 第 141 頁的『9』中指定的觀察值除外。
5. 也必須指定 MQOO_OUTPUT。
6. 此選項也隱含 MQOO_PASS_IDENTITY_CONTEXT。
7. 佇列管理程式物件及特定佇列都需要此權限。
8. 此選項也隱含 MQOO_PASS_IDENTITY_CONTEXT、MQOO_PASS_ALL_CONTEXT 及 MQOO_SET_IDENTITY_CONTEXT。
9. 針對 *Usage* 佇列屬性為 MQUS_TRANSMISSION 且直接開啟以供輸出的本端或模型佇列執行此檢查。 如果正在開啟遠端佇列 (透過指定遠端佇列管理程式及遠端佇列的名稱，或透過指定遠端佇列的本端定義名稱) ，則此不適用。
10. 至少必須指定 MQOO_INQUIRE (適用於任何物件類型) 或 MQOO_BROWSE、MQOO_INPUT_ *、MQOO_OUTPUT 或 MQOO_SET 其中之一。 所執行的檢查與其他指定選項一樣，使用所提供的替代使用者 ID (針對特定命名物件權限) ，以及現行應用程式權限 (針對 MQZAO_ALTERNATE_USER_IDID 檢查)。
11. 此授權容許指定任何 *AlternateUserId* 。
12. 如果佇列沒有 MQUS_TRANSMISSION 的 *Usage* 佇列屬性，則也會執行 MQZAO_OUTPUT 檢查。
13. 所執行的檢查與其他指定選項一樣，使用具名佇列權限所提供的替代使用者 ID ，以及 MQZAO_ALTERNATE_USER_IDID 檢查的現行應用程式權限。
14. 只有在下列兩個陳述式都成立時，才會執行檢查:
    - 正在關閉並刪除永久動態佇列。
    - 佇列不是由傳回所使用物件控點的 MQOPEN 所建立。

    否則，不會有任何檢查。

一般注意事項:

1. 特殊授權 MQZAO_ALL_MQI 包括與物件類型相關的下列所有授權:
   - MQZAO_CONNECT
   - MQZAO_INQUIRE
   - MQZAO_SET
   - MQ 導覽 _ 瀏覽
   - MQZAO_ 輸入
   - MQZAO_OUTPUT

- MQZAO_PASS_IDENTITY_CONTEXT
- MQZAO_PASS_ALL_CONTEXT
- MQZAO_SET_IDENTITY_CONTEXT
- MQZAO_SET_ALL_CONTEXT
- MQZAO_ALTERNATE_USER_AUTHORITY

2. MQZAO_DELETE (請參閱附註 第 141 頁的『14』) 及 MQZAO_DISPLAY 會分類為管理授權。 因此它們不會包含在 MQZAO_ALL_MQI 中。
3. 不檢查 表示不執行授權檢查。
4. 不適用 表示授權檢查與此作業無關。 例如，您無法對程序物件發出 MQPUT 呼叫。

IBM i *IBM i 上跳出 PCF 中 MQSC 指令的授權*

這些授權可讓使用者發出管理指令作為跳出 PCF 訊息。 這些方法可讓程式將管理指令當作訊息傳送至佇列管理程式，以代表該使用者執行。

本節彙總 Escape PCF 中包含的每一個 MQSC 指令所需的授權。

不適用 表示授權檢查與此作業無關。

提交指令的程式所使用的使用者 ID 也必須具有下列權限:

- 佇列管理程式的 MQZAO_CONNECT 權限
- 佇列管理程式上執行 PCF 指令的 DISPLAY 權限
- 在 Escape PCF 指令文字內發出 MQSC 指令的權限

ALTER 物件

物件	需要授權
佇列	MQZAO_CHANGE
主題	MQZAO_CHANGE
處理程序	MQZAO_CHANGE
佇列管理程式	MQZAO_CHANGE
名稱清單	MQZAO_CHANGE
鑑別資訊	MQZAO_CHANGE
通道	MQZAO_CHANGE
用戶端連線通道	MQZAO_CHANGE
接聽器	MQZAO_CHANGE
服務	MQZAO_CHANGE

CLEAR 物件

物件	需要授權
佇列	MQZAO_clear
主題	MQZAO_clear
處理程序	不適用
佇列管理程式	不適用
名稱清單	不適用
鑑別資訊	不適用

物件	需要授權
通道	不適用
用戶端連線通道	不適用
接聽器	不適用
服務	不適用

DEFINE 物件 NOREPLACE (第 146 頁的『1』)

物件	需要授權
佇列	MQZAO_CREATE (第 146 頁的『2』)
主題	MQZAO_CREATE (第 146 頁的『2』)
處理程序	MQZAO_CREATE (第 146 頁的『2』)
佇列管理程式	不適用
名稱清單	MQZAO_CREATE (第 146 頁的『2』)
鑑別資訊	MQZAO_CREATE (第 146 頁的『2』)
通道	MQZAO_CREATE (第 146 頁的『2』)
用戶端連線通道	MQZAO_CREATE (第 146 頁的『2』)
接聽器	MQZAO_CREATE (第 146 頁的『2』)
服務	MQZAO_CREATE (第 146 頁的『2』)

DEFINE 物件 REPLACE (第 146 頁的『1』， 第 146 頁的『3』)

物件	需要授權
佇列	MQZAO_CHANGE
主題	MQZAO_CHANGE
處理程序	MQZAO_CHANGE
佇列管理程式	不適用
名稱清單	MQZAO_CHANGE
鑑別資訊	MQZAO_CHANGE
通道	MQZAO_CHANGE
用戶端連線通道	MQZAO_CHANGE
接聽器	MQZAO_CHANGE
服務	MQZAO_CHANGE

DELETE 物件

物件	需要授權
佇列	MQZAO_DELETE
主題	MQZAO_DELETE
處理程序	MQZAO_DELETE
佇列管理程式	不適用

物件	需要授權
名稱清單	MQZAO_DELETE
鑑別資訊	MQZAO_DELETE
通道	MQZAO_DELETE
用戶端連線通道	MQZAO_DELETE
接聽器	MQZAO_DELETE
服務	MQZAO_DELETE

DISPLAY *object*

物件	需要授權
佇列	MQZAO_DISPLAY
主題	MQZAO_DISPLAY
處理程序	MQZAO_DISPLAY
佇列管理程式	MQZAO_DISPLAY
名稱清單	MQZAO_DISPLAY
鑑別資訊	MQZAO_DISPLAY
通道	MQZAO_DISPLAY
用戶端連線通道	MQZAO_DISPLAY
接聽器	
服務	

Ping 通道

物件	需要授權
佇列	不適用
主題	不適用
處理程序	不適用
佇列管理程式	不適用
名稱清單	不適用
鑑別資訊	不適用
通道	MQZAO_CONTROL
用戶端連線通道	不適用
接聽器	不適用
服務	不適用

重設通道

物件	需要授權
佇列	不適用
主題	不適用

物件	需要授權
處理程序	不適用
佇列管理程式	不適用
名稱清單	不適用
鑑別資訊	不適用
通道	已延伸 MQZAO_CONTROL_EXTENDED
用戶端連線通道	不適用
接聽器	不適用
服務	不適用

解析通道

物件	需要授權
佇列	不適用
主題	不適用
處理程序	不適用
佇列管理程式	不適用
名稱清單	不適用
鑑別資訊	不適用
通道	已延伸 MQZAO_CONTROL_EXTENDED
用戶端連線通道	不適用
接聽器	不適用
服務	不適用

START 物件

物件	需要授權
佇列	不適用
主題	不適用
處理程序	不適用
佇列管理程式	不適用
名稱清單	不適用
鑑別資訊	不適用
通道	MQZAO_CONTROL
用戶端連線通道	不適用
接聽器	MQZAO_CONTROL
服務	MQZAO_CONTROL

STOP 物件

物件	需要授權
佇列	不適用
主題	不適用
處理程序	不適用
佇列管理程式	不適用
名稱清單	不適用
鑑別資訊	不適用
通道	MQZAO_CONTROL
用戶端連線通道	不適用
接聽器	MQZAO_CONTROL
服務	MQZAO_CONTROL

註:

1. 對於 DEFINE 指令， LIKE 物件也需要 MQZAO_DISPLAY 權限 (如果已指定) ，或在適當的 SYSTEM.DEFAULT.xxx 物件 (如果省略 LIKE)。
2. MQZAO_CREATE 權限不是特定物件或物件類型所特有。 透過在 GRTMQMAUT 指令上指定 QMGR 物件類型，授與指定佇列管理程式的所有物件建立權限。
3. 如果要置換的物件已存在，則此選項適用。 如果沒有，則檢查是針對 DEFINE 物件 NOREPLACE。

IBM i *IBM i 上 PCF 指令的授權*

這些授權可讓使用者發出管理指令作為 PCF 指令。 這些方法可讓程式將管理指令當作訊息傳送至佇列管理程式，以代表該使用者執行。

本節彙總每一個 PCF 指令所需的授權。

不檢查 表示不執行任何授權檢查; 不適用 表示授權檢查與此作業無關。

提交指令的程式所使用的使用者 ID 也必須具有下列權限:

- 佇列管理程式的 MQZAO_CONNECT 權限
- 佇列管理程式上執行 PCF 指令的 DISPLAY 權限

特殊授權 MQZAO_ALL_ADMIN 包括下列授權:

- MQZAO_CHANGE
- MQZAO_clear
- MQZAO_DELETE
- MQZAO_DISPLAY
- MQZAO_CONTROL
- 已延伸 MQZAO_CONTROL_EXTENDED

不包括 MQZAO_CREATE ，因為它不是特定物件或物件類型特有的

變更 物件

物件	需要授權
佇列	MQZAO_CHANGE
主題	MQZAO_CHANGE
處理程序	MQZAO_CHANGE

物件	需要授權
佇列管理程式	MQZAO_CHANGE
名稱清單	MQZAO_CHANGE
鑑別資訊	MQZAO_CHANGE
通道	MQZAO_CHANGE
用戶端連線通道	MQZAO_CHANGE
接聽器	MQZAO_CHANGE
服務	MQZAO_CHANGE

清除 物件

物件	需要授權
佇列	MQZAO_clear
主題	MQZAO_clear
處理程序	不適用
佇列管理程式	不適用
名稱清單	不適用
鑑別資訊	不適用
通道	不適用
用戶端連線通道	不適用
接聽器	不適用
服務	不適用

複製 物件 (不取代) (**第 152 頁的『1』** **)**

物件	需要授權
佇列	MQZAO_CREATE (**第 152 頁的『2』**)
主題	MQZAO_CREATE (**第 152 頁的『2』**)
處理程序	MQZAO_CREATE (**第 152 頁的『2』**)
佇列管理程式	不適用
NamelistMQZAO_CREATE	MQZAO_CREATE (**第 152 頁的『2』**)
鑑別資訊	MQZAO_CREATE (**第 152 頁的『2』**)
通道	MQZAO_CREATE (**第 152 頁的『2』**)
用戶端連線通道	MQZAO_CREATE (**第 152 頁的『2』**)
接聽器	MQZAO_CREATE (**第 152 頁的『2』**)
服務	MQZAO_CREATE (**第 152 頁的『2』**)

複製 物件 (含取代) (**第 152 頁的『1』****,** **第 152 頁的『4』** **)**

物件	需要授權
佇列	MQZAO_CHANGE

物件	需要授權
主題	MQZAO_CHANGE
處理程序	MQZAO_CHANGE
佇列管理程式	不適用
名稱清單	MQZAO_CHANGE
鑑別資訊	MQZAO_CHANGE
通道	MQZAO_CHANGE
用戶端連線通道	MQZAO_CHANGE
接聽器	MQZAO_CHANGE
服務	MQZAO_CHANGE

建立 物件 (不取代) (第 152 頁的『3』)

物件	需要授權
佇列	MQZAO_CREATE (第 152 頁的『2』)
主題	MQZAO_CREATE (第 152 頁的『2』)
處理程序	MQZAO_CREATE (第 152 頁的『2』)
佇列管理程式	不適用
名稱清單	MQZAO_CREATE (第 152 頁的『2』)
鑑別資訊	MQZAO_CREATE (第 152 頁的『2』)
通道	MQZAO_CREATE (第 152 頁的『2』)
用戶端連線通道	MQZAO_CREATE (第 152 頁的『2』)
接聽器	MQZAO_CHANGE
服務	MQZAO_CHANGE

建立 物件 (含取代) (第 152 頁的『3』， 第 152 頁的『4』)

物件	需要授權
佇列	MQZAO_CHANGE
主題	MQZAO_CHANGE
處理程序	MQZAO_CHANGE
佇列管理程式	不適用
名稱清單	MQZAO_CHANGE
鑑別資訊	MQZAO_CHANGE
通道	MQZAO_CHANGE
用戶端連線通道	MQZAO_CHANGE
接聽器	MQZAO_CHANGE
服務	MQZAO_CHANGE

刪除 物件

物件	**需要授權**
佇列	MQZAO_DELETE
主題	MQZAO_DELETE
處理程序	MQZAO_DELETE
佇列管理程式	MQZAO_DELETE
名稱清單	MQZAO_DELETE
鑑別資訊	MQZAO_DELETE
通道	MQZAO_DELETE
用戶端連線通道	MQZAO_DELETE
接聽器	MQZAO_DELETE
服務	MQZAO_DELETE

查詢 物件

物件	**需要授權**
佇列	MQZAO_DISPLAY
主題	MQZAO_DISPLAY
處理程序	MQZAO_DISPLAY
佇列管理程式	MQZAO_DISPLAY
名稱清單	MQZAO_DISPLAY
鑑別資訊	MQZAO_DISPLAY
通道	MQZAO_DISPLAY
用戶端連線通道	MQZAO_DISPLAY
接聽器	MQZAO_DISPLAY
服務	MQZAO_DISPLAY

查詢 *object* 名稱

物件	**需要授權**
佇列	不檢查
主題	不檢查
處理程序	不檢查
佇列管理程式	不檢查
名稱清單	不檢查
鑑別資訊	不檢查
通道	不檢查
用戶端連線通道	不檢查
接聽器	不檢查

物件	需要授權
服務	不檢查

Ping 通道

物件	需要授權
佇列	不適用
主題	不適用
處理程序	不適用
佇列管理程式	不適用
名稱清單	不適用
鑑別資訊	不適用
通道	MQZAO_CONTROL
用戶端連線通道	不適用
接聽器	不適用
服務	不適用

重設通道

物件	需要授權
佇列	不適用
主題	不適用
處理程序	不適用
佇列管理程式	不適用
名稱清單	不適用
鑑別資訊	不適用
通道	已延伸 MQZAO_CONTROL_EXTENDED
用戶端連線通道	不適用
接聽器	不適用
服務	不適用

重設佇列統計資料

物件	需要授權
佇列	MQZAO_DISPLAY 和 MQZAO_CHANGE
主題	不適用
處理程序	不適用
佇列管理程式	不適用
名稱清單	不適用
鑑別資訊	不適用
通道	不適用

物件	需要授權
用戶端連線通道	不適用
接聽器	
服務	

解析通道

物件	需要授權
佇列	不適用
主題	不適用
處理程序	不適用
佇列管理程式	不適用
名稱清單	不適用
鑑別資訊	不適用
通道	已延伸 MQZAO_CONTROL_EXTENDED
用戶端連線通道	不適用
接聽器	不適用
服務	不適用

啟動通道

物件	需要授權
佇列	不適用
主題	不適用
處理程序	不適用
佇列管理程式	不適用
名稱清單	不適用
鑑別資訊	不適用
通道	MQZAO_CONTROL
用戶端連線通道	不適用
接聽器	不適用
服務	不適用

停止通道

物件	需要授權
佇列	不適用
主題	不適用
處理程序	不適用
佇列管理程式	不適用
名稱清單	不適用

物件	需要授權
鑑別資訊	不適用
通道	MQZAO_CONTROL
用戶端連線通道	不適用
接聽器	不適用
服務	不適用

註:

1. 對於 Copy 指令， From 物件也需要 MQZAO_DISPLAY 權限。
2. MQZAO_CREATE 權限不是特定物件或物件類型所特有。 透過在 GRTMQMAUT 指令上指定 QMGR 物件類型，授與指定佇列管理程式的所有物件建立權限。
3. 若為「建立」指令，適當的 SYSTEM.DEFAULT.* 物件。
4. 如果要置換的物件已存在，則此選項適用。 如果不存在，則檢查適用於「複製」或「建立而不取代」。

IBM i IBM i 上的通用 OAM 設定檔

物件權限管理程式 (OAM) 通用設定檔可讓您一次設定使用者對許多物件的權限，而不必在建立時針對每一個個別物件發出個別 **GRTMQMAUT** 指令。 在 **GRTMQMAUT** 指令中使用通用設定檔，可讓您針對所有建立符合該設定檔的未來物件設定通用權限。

本節其餘部分更詳細說明通用設定檔的用法:

- 第 152 頁的『使用萬用字元』
- 第 153 頁的『設定檔優先順序』

使用萬用字元

使設定檔成為通用的是在設定檔名稱中使用特殊字元 (萬用字元)。 例如，問號 (?) 萬用字元符合名稱中的任何單一字元。 因此，如果您指定 ABC.?EF，您提供給該設定檔的授權會套用至以 ABC.DEF、 ABC.CEF、 ABC.BEF 等名稱建立的任何物件。

可用的萬用字元如下:

?
請使用問號 (?)，而不是任何單一字元。 例如， AB.?D 將套用至物件 AB.CD、 AB.ED 及 AB.FD。

*****
使用星號 (*) 作為:

- 設定檔名稱中的 限定元 ，符合物件名稱中的任何一個限定元。 限定元為物件名稱的一部分，以句點區隔。 例如，在 ABC.DEF.GHI 中，限定元為 ABC、DEF 及 GHI。

  例如， ABC.*.JKL 會套用至物件 ABC.DEF.JKL 及 ABC.GHI.JKL。 (請注意，它 **不** 適用於 ABC.JKL ; * used in this context always indicates one qualifier.)
- 設定檔名稱中限定元內的字元，符合物件名稱中限定元內零個以上的字元。

  例如， ABC.DE*.JKL 將套用至物件 ABC.DE.JKL、 ABC.DEF.JKL 及 ABC.DEGH.JKL。

******
在設定檔名稱中使用雙星號 (**) ***once*** ，如下所示:

- 符合所有物件名稱的整個設定檔名稱。 例如，如果您使用關鍵字 OBJTYPE （*PRC）來識別處理程序，然後使用 ** 作為設定檔名稱，則會變更所有處理程序的授權。
- 作為設定檔名稱中的開始、中間或結束限定元，以符合物件名稱中的零個以上限定元。 例如， **.ABC 會識別具有最終限定元 ABC 的所有物件。

設定檔優先順序

當使用通用設定檔時，要瞭解的重要點是在決定要套用至所建立物件的權限時，提供設定檔的優先順序。 例如，假設您已發出下列指令:

```
GRTMQMAUT OBJ(AB.*) OBJTYPE(*Q) USER(FRED) AUT(*PUT) MQMNAME(MYQMGR)
GRTMQMAUT OBJ(AB.C*) OBJTYPE(*Q) USER(FRED) AUT(*GET) MQMNAME(MYQMGR)
```

第一個會提供對主體 FRED 的所有佇列的放置權限，這些佇列的名稱符合設定檔 AB. *; 第二個會提供取得權限給符合設定檔 AB.C*。

假設您現在建立名為 AB.CD。 根據萬用字元比對的規則， GRTMQMAUT 可以套用至該佇列。 所以它是有權力還是有權力?

若要尋找答案，您可以套用規則，每當多個設定檔可以套用至物件時， **只會套用最特定的**。 您套用此規則的方式是從左到右比較設定檔名稱。 無論它們有何不同，非同屬字元比同屬字元更具體。 因此，在前一個範例中，是佇列 AB.CD 具有 **get** 權限 (AB.C* 比 AB. *) 更具體。

當您比較一般字元時， 特定性 的順序如下:

1. ?
2. *
3. **

IBM i 指定 IBM i 上已安裝的授權服務

您可以指定要使用的授權服務元件。

GRTMQMAUT 和 **RVKMQMAUT** 上的參數 **Service Component name** 可讓您指定已安裝授權服務元件的名稱。

在起始畫面上選取 **F24** ，然後在任一指令的下一個畫面上選取 **F9=All 參數** ，可讓您指定已安裝的授權元件 (*DFT) ，或在佇列管理程式 qm.ini 檔的「服務」段落中指定的必要授權服務元件名稱。

DSPMQMAUT 也具有此額外參數。 此參數可讓您在所有已安裝的授權元件 (*DFT) 或指定的授權服務元件名稱中搜尋指定的物件名稱、物件類型及使用者

IBM i 在 IBM i 上使用及不使用權限設定檔

使用此資訊來瞭解如何使用權限設定檔，以及如何在沒有權限設定檔的情況下工作。

您可以使用權限設定檔 (如 第 153 頁的『使用權限設定檔』中所說明) ，也可以不使用它們，如這裡所說明:

若要在沒有權限設定檔的情況下工作，請使用 *NONE 作為 **GRTMQMAUT** 上的「權限」參數，以在沒有權限的情況下建立設定檔。 這會維持任何現有的設定檔不變。

在 **RVKMQMAUT** 上，使用 *REMOVE 作為「權限」參數，以移除現有的權限設定檔。

使用權限設定檔

權限側寫有兩個相關聯的指令:

- **WRKMQMAUT**
- **WRKMQMAUTD**

您可以直接從指令行或 WRKMQM 畫面存取這些指令，方法為:

1. 鍵入佇列管理程式名稱，並按 Enter 鍵以存取 **WRKMQM** 結果畫面。
2. 在此畫面上選取 F23=More options 。

選項 24 會選取 **WRKMQMAUT** 指令 的結果畫面，而選項 25 會選取 **WRKMQMAUTI** 指令，以與 SSL 連結層搭配使用。

WRKMQMAUT

此指令可讓您使用保留在權限佇列中的權限資料。

註: 若要執行此指令，您必須具有佇列管理程式的 *connect 及 *admdsp 權限。 不過，若要建立或刪除設定檔，您需要 QMQMADM 權限。

如果您將資訊輸出至畫面，則會顯示權限設定檔名稱及其類型的清單。 如果您列印輸出，則會收到所有權限資料、已登錄使用者及其權限的詳細清單。

在此畫面上輸入物件或設定檔名稱，然後按 ENTER 鍵，會將您帶到 **WRKMQMAUT** 的結果畫面。

如果您選取 4=Delete，則會跳至新的畫面，您可以從中確認要刪除向您指定的通用權限設定檔名稱登錄的所有使用者名稱。 此選項會針對所有使用者執行 **RVKMQMAUT** 與選項 *REMOVE，並將 **僅** 套用至通用設定檔名稱。

如果您選取 12=Work with profile，則會跳至 **WRKMQMAUTD** 指令結果畫面，如 第 154 頁的「WRKMQMAUTD」中所述。

WRKMQMAUTD

此指令可讓您顯示以特定權限設定檔名稱及物件類型登錄的所有使用者。 若要執行此指令，您必須具有佇列管理程式的 *connect 及 *admdsp 權限。 不過，若要授與、執行、建立或刪除設定檔，您需要 QMQMADM 權限。

從起始輸入畫面中選取 F24=More keys，後面接著選項 F9=All Parameters，會顯示 **GRTMQMAUT** 和 **RVKMQMAUT** 的「服務元件名稱」。

註: F11=Display Object Authorizations 金鑰會在下列類型的權限之間切換:

- 物件授權
- 環境定義授權
- MQI 授權

畫面上的選項如下:

2=Grant
將您帶至 **GRTMQMAUT** 畫面，以新增至現行權限。

3=Revoke
帶您到 **RVKMQMAUT** 畫面，以移除部分現行定義

4=Delete
將您帶至可讓您刪除指定使用者之權限資料的畫面。 這會使用選項 *REMOVE 來執行 **RVKMQMAUT**。

5=Display
將您帶至現有的 **DSPMQMAUT** 指令

F6=Create
將您帶到可讓您建立設定檔權限記錄的 **GRTMQMAUT** 畫面。

IBM i IBM i 上的物件權限管理程式準則

使用物件權限管理程式 (OAM) 的其他提示和要訣

限制機密作業的存取權

部分作業是機密作業; 請將它們限制為特許使用者。 例如,

- 存取部分特殊佇列，例如傳輸佇列或指令佇列 SYSTEM.ADMIN.COMMAND.QUEUE
- 執行使用完整 MQI 環境定義選項的程式
- 建立及複製應用程式佇列

佇列管理程式目錄

包含佇列及其他佇列管理程式資料的目錄及檔案庫是產品專用的。 請勿使用標準作業系統指令來授與或撤銷對 MQI 資源的授權。

佇列

動態佇列的權限是以衍生它的模型佇列為基礎，但不一定與衍生它的模型佇列相同。

對於別名佇列及遠端佇列，授權是物件本身的授權，而不是別名或遠端佇列所解析成的佇列。 可以授權使用者設定檔存取別名佇列，該別名佇列解析為使用者設定檔沒有存取權的本端佇列。

將建立佇列的權限限制為特許使用者。 如果沒有，使用者可以透過建立別名來略過一般存取控制。

替代使用者權限

替代使用者權限控制當存取 IBM MQ 物件時，一個使用者設定檔是否可以使用另一個使用者設定檔的權限。 當伺服器接收來自程式的要求，且伺服器想要確保程式具有要求的必要權限時，此技術非常重要。 伺服器可能具有必要的權限，但它需要知道程式是否具有它所要求之動作的權限。

例如：

- 在使用者設定檔 PAYSERV 下執行的伺服器程式會從使用者設定檔 USER1 放置在佇列上的佇列中擷取要求訊息。
- 當伺服器程式取得要求訊息時，它會處理要求，並將回覆放回要求訊息所指定的回覆佇列中。
- 伺服器可以指定其他使用者設定檔 (在此情況下為 USER1)，而不是使用自己的使用者設定檔 (PAYSERV) 來授權開啟回覆目的地佇列。 在此範例中，您可以使用替代使用者權限來控制是否容許 PAYSERV 在開啟回覆目的地佇列時指定 USER1 作為替代使用者設定檔。

在物件描述子的 *AlternateUserId* 欄位上指定替代使用者設定檔。

註: 您可以在任何 IBM MQ 物件上使用替代使用者設定檔。 使用替代使用者設定檔不會影響任何其他資源管理程式所使用的使用者設定檔。

環境定義權限

環境定義是適用於特定訊息的資訊，包含在訊息的訊息描述子 MQMD 中。

如需與環境定義相關之訊息描述子欄位的說明，請參閱 MQMD 概觀。

如需環境定義選項的相關資訊，請參閱 訊息環境定義。

遠端安全考量

對於遠端安全，請考量：

放置權限
為了確保佇列管理程式之間的安全，您可以指定當通道接收從另一個佇列管理程式傳送的訊息時所使用的放置權限。

此參數僅對 RCVR、RQSTR 或 CLUSRCVR 通道類型有效。 指定通道屬性 PUTAUT，如下所示：

DEF
預設使用者設定檔。 這是執行訊息通道代理程式的 QMQM 使用者設定檔。

CTX
訊息環境定義中的使用者設定檔。

傳輸佇列
佇列管理程式會自動將遠端訊息放入傳輸佇列中; 不需要特殊權限。 不過，將訊息直接放置在傳輸佇列上需要特殊授權。

通道結束程式
通道結束程式可用來增加安全。

通道鑑別記錄
用來在通道層次對授與連接系統的存取權進行更精確的控制。

如需遠端安全的相關資訊，請參閱 第 92 頁的『通道授權』。

使用 SSL/TLS 保護通道

「傳輸層安全 (TLS)」通訊協定提供通道安全，可防止竊聽、竄改及模擬。 IBM MQ 支援 TLS 可讓您在通道定義上指定特定通道使用 TLS 安全。 您也可以指定所需安全的詳細資料，例如您要使用的加密演算法。

IBM MQ 中的 TLS 支援使用佇列管理程式 鑑別資訊物件 及各種 CL 和 MQSC 指令，以及佇列管理程式和通道參數，這些參數可詳細定義所需的 TLS 支援。

下列 CL 指令支援 TLS:

WRKMQMAUTI
使用鑑別資訊物件的屬性。

CHGMQMAUTI
修改鑑別資訊物件的屬性。

CRTMQMAUTI
建立鑑別資訊物件。

CPYMQMAUTI
複製現有的鑑別資訊物件來建立鑑別資訊物件。

DLTMQMAUTI
刪除鑑別資訊物件。

DSPMQMAUTI
顯示特定鑑別資訊物件的屬性。

如需使用 TLS 的通道安全概觀，請參閱

- 使用 TLS 保護通道

如需與 TLS 相關聯之 PCF 指令的詳細資料，請參閱

- 變更、複製及建立鑑別資訊物件
- 刪除鑑別資訊物件
- 查詢鑑別資訊物件

z/OS 在 z/OS 上設定安全

z/OS 特有的安全考量。

IBM MQ for z/OS 中的安全是使用 RACF 或對等的外部安全管理程式 (ESM) 來控制。

下列指示假設您使用 RACF。

相關參考
安全實務範例: z/OS 上的兩個佇列管理程式
安全實務範例: z/OS 上的佇列共用群組

z/OS RACF 安全類別

RACF 類別用來保留 IBM MQ 安全檢查所需的設定檔。 許多成員類別都有相等的群組類別。 您必須啟動類別，並讓它們接受通用設定檔。

每一個 RACF 類別都會保留在檢查順序中某個點使用的一個以上設定檔，如 第 157 頁的表 23 中所示。

表 *23: IBM MQ* 使用的 *RACF* 類別

成員類別	群組類別	內容
MQADMIN	GMQADMIN	主要用於管理功能的設定檔。 例如: • IBM MQ 安全交換器的設定檔。 • RESLEVEL 安全設定檔。 • 替代使用者安全的設定檔。 • 環境定義安全的設定檔。 • 指令資源安全的設定檔。 此類別只能保留大寫 RACF 設定檔。
MXADMIN	GMXADMIN	主要用於管理功能的設定檔。 例如: • IBM MQ 安全交換器的設定檔。 • RESLEVEL 安全設定檔。 • 替代使用者安全的設定檔。 • 環境定義安全的設定檔。 • 指令資源安全的設定檔。 此類別可以同時保留大寫及大小寫混合格式的 RACF 設定檔。
MQCONN		用於連線安全的設定檔。
MQCMDS		用於指令安全的設定檔。
MQQUEUE	GMQQUEUE	佇列資源安全中使用的大寫設定檔。
MXQUEUE	GMXQUEUE	在佇列資源安全中使用大小寫混合的設定檔。
MQPROC	GMQPROC	處理程序資源安全中使用的大寫設定檔。
MXPROC	GMXPROC	在處理程序資源安全中使用大小寫混合的設定檔。
MQNLIST	GMQNLIST	在名單資源安全中使用的大寫設定檔。
MXNLIST	GMXNLIST	在名單資源安全中使用大小寫混合的設定檔。
MXTOPIC	GMXTOPIC	主題安全中使用的大小寫混合設定檔。

部分類別具有相關的 群組類別 ，可讓您將具有類似存取需求的資源群組組合在一起。 如需成員與群組類別之間的差異以及何時使用成員或群組類別的詳細資料，請參閱 z/OS Security Server RACF Security Administrator 's Guide。

必須先啟動類別，才能進行安全檢查。 若要啟動所有 IBM MQ 類別，您可以使用下列 RACF 指令:

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

您也應該確定您已設定類別，以便它們可以接受通用設定檔。 您也可以使用 RACF 指令 **SETROPTS** 來執行此動作，例如:

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

z/OS RACF 設定檔

IBM MQ 使用的所有 RACF 設定檔都包含字首，它是佇列管理程式名稱或佇列共用群組名稱。 使用百分比符號作為萬用字元時請小心。

IBM MQ 使用的所有 RACF 設定檔都包含字首。 對於佇列共用群組層次安全，這是佇列共用群組名稱。 對於佇列管理程式層次安全，字首是佇列管理程式名稱。 如果您混合使用佇列管理程式及佇列共用群組層次安全，則會使用具有這兩種字首類型的設定檔。 IBM MQ for z/OS 中的安全控制項和選項說明佇列共用群組和佇列管理程式層次安全。

例如，如果您要在佇列共用群組層次保護佇列共用群組 QSG1 中稱為 QUEUE_FOR_SUBSCRIBER_LIST 的佇列，則適當的設定檔會定義為 RACF：

```
RDEFINE MQQUEUE QSG1.QUEUE_FOR_SUBSCRIBER_LIST
```

如果您想要保護名為 QUEUE_FOR_LOST_CARD_LIST 的佇列 (屬於佇列管理程式層次的佇列管理程式 STCD)，則會將適當的設定檔定義為 RACF：

```
RDEFINE MQQUEUE STCD.QUEUE_FOR_LOST_CARD_LIST
```

這表示不同的佇列管理程式和佇列共用群組可以共用相同的 RACF 資料庫，但具有不同的安全選項。

請勿在設定檔中使用通用佇列管理程式名稱，以避免非預期的使用者存取。

IBM MQ 容許在物件名稱中使用百分比符號 (%)。 不過， RACF 會使用% 字元作為單一字元萬用字元。 這表示當您定義名稱中含有% 字元的物件名稱時，必須在定義對應的設定檔時考量此情況。

例如，對於佇列管理程式 CRDP 上的 CREDIT_CARD_ %_RATE_INQUIRY 佇列，設定檔將定義為 RACF，如下所示:

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

此佇列無法受到同屬設定檔 (例如 CRDP.* *) 的保護。

IBM MQ 容許在物件名稱中使用大小寫混合的字元。 您可以透過定義下列項目來保護這些物件:

1. 適當大小寫混合的 RACF 類別中的大小寫混合設定檔，或
2. 適當大寫 RACF 類別的通用設定檔。

若要使用大小寫混合的設定檔及大小寫混合的 RACF 類別，您必須遵循 第 225 頁的『將 z/OS 佇列管理程式移轉至大小寫混合格式安全』中說明的步驟。

只有在 IBM MQ 提供值時，部分設定檔或部分設定檔才會保持大寫。 它們是:

- 切換設定檔。
- 所有高階限定元 (HLQ)，包括子系統及佇列共用群組 ID。
- SYSTEM 物件的設定檔。
- 預設物件的設定檔。
- **MQCMDS** 類別，因此所有指令設定檔都只能大寫。
- **MQCONN** 類別，因此所有連線設定檔都僅限大寫。
- **RESLEVEL** 設定檔。
- 指令資源設定檔中的 'object' 資格; 例如 hlq.QUEUE.queuename。 資源名稱僅大小寫混合。
- 動態佇列設定檔 hlq.CSQOREXX.*、 hlq.CSQUTIL.*及 CSQXCMD.*。
- hlq.CONTEXT.resourcename 的 'CONTEXT' 部分。
- hlq.ALTERNATE.USER.userid 的 'ALTERNATE.USER' 部分。

例如，您可以使用下列其中一種方式來定義設定檔，以授與對佇列管理程式 QM01 上稱為 PAYROLL.Dept1 的佇列的存取權。

- 如果您使用大小寫混合的設定檔，您可以使用下列指令在 IBM MQ RACF 類別 MXQUEUE 中定義設定檔:

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

- 如果您使用大寫設定檔，則可以使用下列指令在 IBM MQ RACF 類別 MQQUEUE 中定義設定檔:

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

第一個範例 (使用大小寫混合的設定檔) 可讓您更精細地控制授與存取資源的權限。

z/OS 交換器設定檔

若要控制 IBM MQ 所執行的安全檢查，您可以使用 切換設定檔。 交換器設定檔是對 IBM MQ 具有特殊意義的一般 RACF 設定檔。 IBM MQ 不會使用交換器設定檔中的存取清單。

IBM MQ 會針對表格 子系統層次安全的交換器設定檔、 佇列共用群組或佇列管理程式層次安全的交換器設定檔及 資源檢查的交換器設定檔中顯示的每一種交換器類型維護一個內部交換器。 交換器設定檔可以在佇列共用群組層次、佇列管理程式層次或兩者的組合中維護。 使用一組佇列共用群組安全切換設定檔，您可以控制佇列共用群組內所有佇列管理程式的安全。

當安全交換器設為開啟時，會執行與交換器相關聯的安全檢查。 當安全開關設為關閉時，會略過與該開關相關聯的安全檢查。 預設值是所有安全交換器都設為開啟。

z/OS 交換器及類別

當您啟動佇列管理程式或重新整理安全時， IBM MQ 會根據各種 RACF 類別的狀態來設定切換。

當啟動佇列管理程式時 (或當 IBM MQ REFRESH SECURITY 指令重新整理 MQADMIN 或 MXADMIN 類別時)， IBM MQ 會先檢查 RACF 及適當類別的狀態:

- MQADMIN 類別 (如果您使用大寫設定檔)
- MXADMIN 類別 (如果您使用大小寫混合格式設定檔)。

如果下列任何條件成立，它會將子系統安全開關設為關閉:

- RACF 非作用中或未安裝。
- 未定義 MQADMIN 或 MXADMIN 類別 (這些類別一律針對 RACF 定義，因為它們包含在類別描述子表格 (CDT) 中)。
- 尚未啟動 MQADMIN 或 MXADMIN 類別。

如果 RACF 及 MQADMIN 或 MXADMIN 類別都處於作用中，則 IBM MQ 會檢查 MQADMIN 或 MXADMIN 類別，以查看是否已定義任何交換器設定檔。 它會先檢查 第 160 頁的『控制子系統安全的設定檔』中說明的設定檔。 如果不需要子系統安全， IBM MQ 會將內部子系統安全開關設為關閉，且不會執行進一步檢查。

這些設定檔會決定對應的 IBM MQ 開關是設為開啟或關閉。

- 如果關閉開關，則會取消啟動該安全類型。
- 如果有任何 IBM MQ 開關設為開啟， IBM MQ 會檢查與 IBM MQ 開關對應之安全類型相關聯的 RACF 類別狀態。 如果類別未安裝或非作用中，則會將 IBM MQ 開關設為關閉。 例如，如果尚未啟動 MQPROC 或 MXPROC 類別，則不會執行處理程序安全檢查。 非作用中類別相當於為使用此 RACF 資料庫的每個佇列管理程式及佇列共用群組定義 NO.PROCESS.CHECKS 設定檔。

z/OS 交換器如何運作

若要設定安全開關，請定義 NO.* 交換器設定檔。 您可以置換 NO.* 透過定義 YES.* 來設定佇列共用群組層次的設定檔。 佇列管理程式的設定檔。

若要設定安全開關，您需要定義 NO.* 交換器設定檔。 存在 NO.* 設定檔表示 **不會** 針對該類型的資源執行安全檢查，除非您選擇置換特定佇列管理程式上的佇列共用群組層次設定。 這說明於第 160 頁的『置換佇列共用群組層次設定』。

如果您的佇列管理程式不是佇列共用群組的成員，則不需要定義任何佇列共用群組層次設定檔或任何置換設定檔。 不過，如果佇列管理程式日後加入佇列共用群組，您必須記得定義這些設定檔。

每一個 NO.* IBM MQ 偵測到的交換器設定檔會關閉該類型資源的檢查。 在佇列管理程式啟動期間，會啟動交換器設定檔。 如果您在任何受影響的佇列管理程式執行時變更交換器設定檔，則可以發出 IBM MQ REFRESH SECURITY 指令，讓 IBM MQ 辨識這些變更。

交換器設定檔必須一律定義在 MQADMIN 或 MXADMIN 類別中。 請勿在 GMQADMIN 或 GMXADMIN 類別中定義它們。 表格 子系統層次安全的交換器設定檔 及 資源檢查的交換器設定檔 顯示有效的交換器設定檔及其控制的安全類型。

置換佇列共用群組層次設定

您可以針對屬於該群組成員的特定佇列管理程式，置換佇列共用群組層次安全設定。 如果您要對未在群組中其他佇列管理程式上執行的個別佇列管理程式執行佇列管理程式檢查，請使用 (qmgr-name.YES. *) 交換器設定檔。

相反地，如果您不想對佇列共用群組內的某個特定佇列管理程式執行特定檢查，請定義一個 (qmgr-name.NO. *) 佇列管理程式上該特定資源類型的設定檔，且不定義佇列共用群組的設定檔。 (只有在找不到佇列管理程式層次設定檔時， IBM MQ 才會檢查佇列共用群組層次設定檔。)

z/OS 控制子系統安全的設定檔

IBM MQ 會檢查子系統、佇列管理程式及佇列共用群組是否需要子系統安全檢查。

IBM MQ 進行的第一個安全檢查是用來判斷整個 IBM MQ 子系統是否需要安全檢查。 如果您指定不要子系統安全，則不會進行進一步檢查。

系統會檢查下列交換器設定檔，以判定是否需要子系統安全。 第 160 頁的圖 14 顯示檢查它們的順序。

表 *24:* 子系統層次安全的交換器設定檔

交換器設定檔名稱	**受控制的資源類型或檢查**
qmgr-name.NO.SUBSYS.SECURITY	此佇列管理程式的子系統安全
qsg-name.NO.SUBSYS.SECURITY	此佇列共用群組的子系統安全
qmgr-name.YES.SUBSYS.SECURITY	此佇列管理程式的子系統安全置換

如果您的佇列管理程式不是佇列共用群組的成員， IBM MQ 只會檢查 qmgr-name.NO.SUBSYS.SECURITY 交換器設定檔。

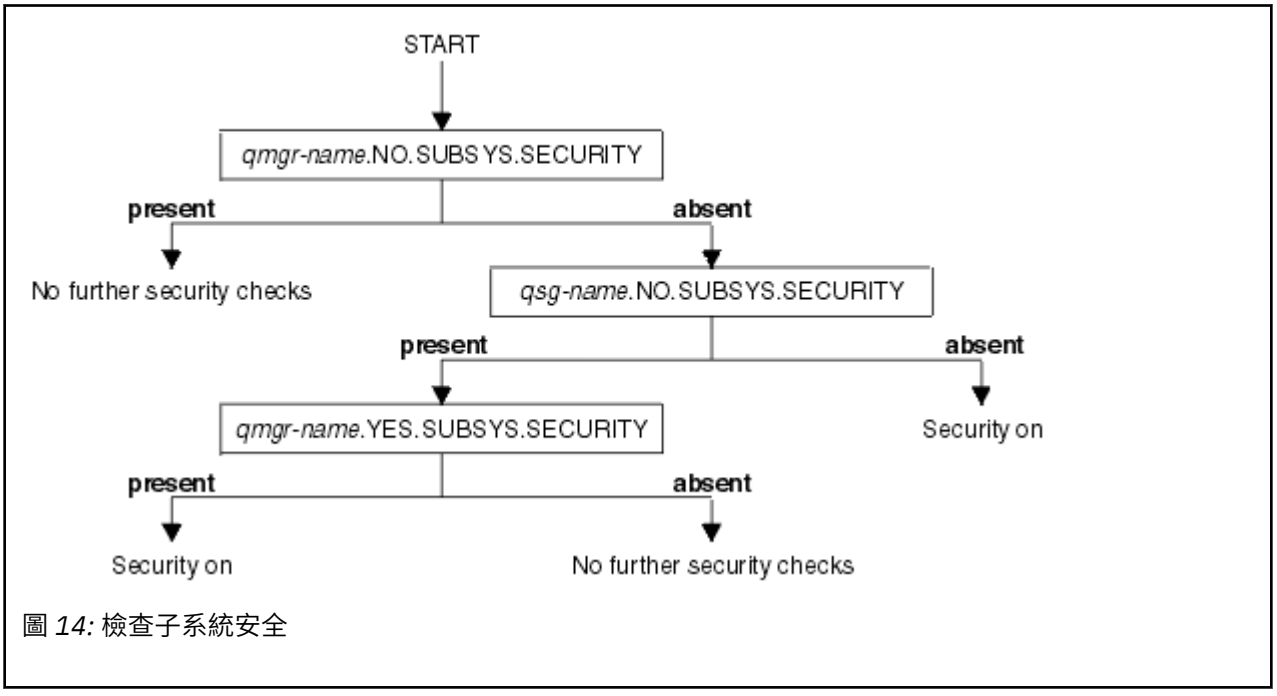


圖 *14:* 檢查子系統安全

z/OS 控制佇列共用群組或佇列管理程式層次安全的設定檔

如果需要子系統安全檢查， IBM MQ 會檢查在佇列共用群組或佇列管理程式層次是否需要安全檢查。

當 IBM MQ 判定需要安全檢查時，它會判定是否需要在佇列共用群組及/或佇列管理程式層次進行檢查。 如果您的佇列管理程式不是佇列共用群組的成員，則不會執行這些檢查。

會檢查下列交換器設定檔，以判定所需的層次。 第 161 頁的圖 15 和 第 162 頁的圖 16 會顯示檢查它們的順序。

表 *25:* 切換佇列共用群組或佇列管理程式層次安全的設定檔

交換器設定檔名稱	**受控制的資源類型或檢查**
qmgr-name.NO.QMGR.CHECKS	此佇列管理程式沒有佇列管理程式層次檢查
qsg-name.NO.QMGR.CHECKS	沒有此佇列共用群組的佇列管理程式層次檢查
qmgr-name.YES.QMGR.CHECKS	此佇列管理程式的佇列管理程式層次檢查置換
qmgr-name.NO.QSG.CHECKS	沒有此佇列管理程式的佇列共用群組層次檢查
qsg-name.NO.QSG.CHECKS	沒有此佇列共用群組的佇列共用群組層次檢查
qmgr-name.YES.QSG.CHECKS	此佇列管理程式的佇列共用群組層次檢查置換

如果子系統安全處於作用中，則無法同時關閉佇列共用群組及佇列管理程式層次安全。 如果您嘗試這樣做，IBM MQ 會在兩個層次設定安全檢查。

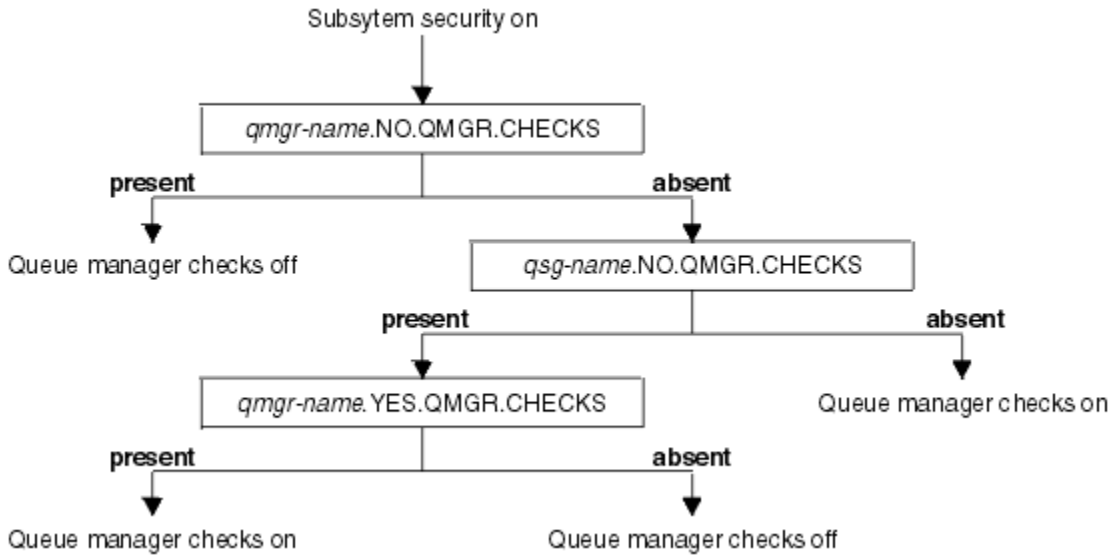


圖 *15:* 檢查佇列管理程式層次安全

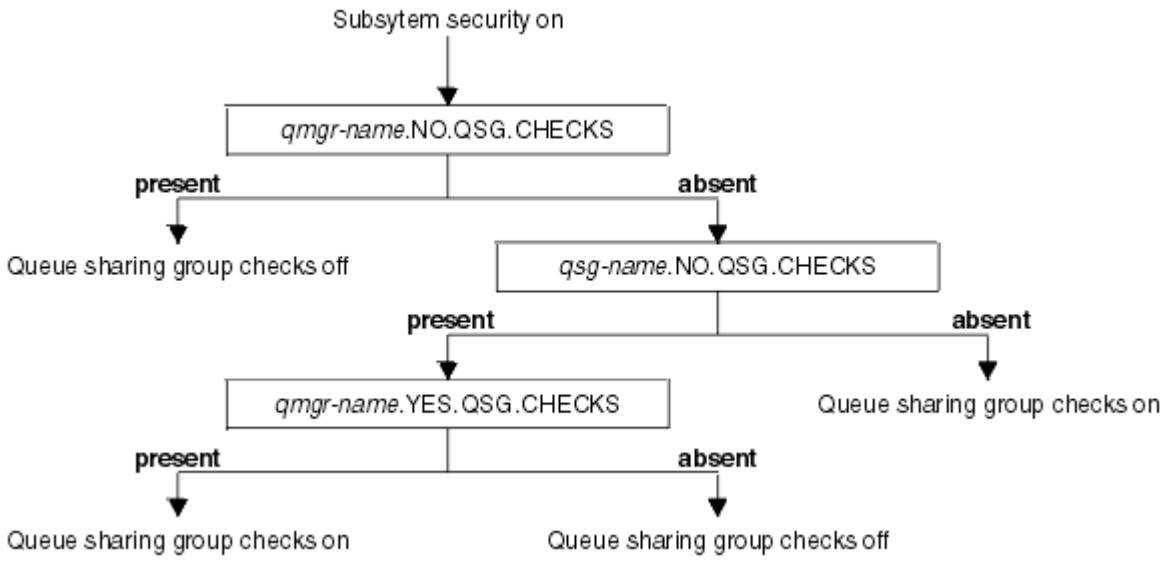


圖 *16:* 檢查佇列共用群組層次安全

z/OS 安全交換器的有效組合

只有某些交換器組合是有效的。 如果您使用無效的交換器設定組合，則會發出訊息 CSQH026I ， 並在佇列共用群組及佇列管理程式層次設定安全檢查。

第 162 頁的表 26、 第 162 頁的表 27、 第 163 頁的表 28 及 第 163 頁的表 29 會顯示適用於每一種安全層次類型的交換器設定組合。

表 *26:* 佇列管理程式層次安全的有效安全切換組合
組合
qmgr-name.NO.QSG.CHECKS
qsg-name.NO.QSG.CHECKS
qmgr-name.NO.QSG.CHECKS qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS
qsg-name.NO.QSG.CHECKS qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS

表 *27:* 佇列共用群組層次安全的有效安全切換組合
組合
qmgr-name.NO.QMGR.CHECKS
qsg-name.NO.QMGR.CHECKS
qmgr-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS

表 *27:* 佇列共用群組層次安全的有效安全切換組合 *(繼續)*
組合
qsg-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS

表 *28:* 佇列管理程式和佇列共用群組層次安全的有效安全切換組合
組合
qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS 無 QSG.* 定義設定檔
無 QMGR.* 定義設定檔 qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS
未定義任一交換器的設定檔

表 *29:* 其他有效的安全切換組合，可切換兩個層次的檢查 **開啟**。
組合
qmgr-name.NO.QMGR.CHECKS qmgr-name.NO.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS
qmgr-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qmgr-name.NO.QSG.CHECKS

z/OS 資源層次檢查

使用許多交換器設定檔來控制對資源的存取權。 在佇列管理程式或佇列共用群組上執行部分停止檢查。 這些可以被啟用特定佇列管理程式檢查的設定檔置換。

第 164 頁的表 30 顯示用來控制 IBM MQ 資源存取權的交換器設定檔。

如果您的佇列管理程式是佇列共用群組的一部分，且您同時有作用中的佇列管理程式和佇列共用群組安全，您可以使用 YES.*。 切換設定檔以置換佇列共用群組層次設定檔，並特別開啟特定佇列管理程式的安全。

部分設定檔同時適用於佇列管理程式及佇列共用群組。 這些是以 *hlq* 字串作為字首，您應該在適用時替換佇列共用群組或佇列管理程式的名稱。 以 *qmgr-name* 為字首顯示的設定檔名稱是佇列管理程式置換設定檔; 您應該取代佇列管理程式的名稱。

表 *30:* 用於資源檢查的交換器設定檔		
受控制的資源檢查類型	**交換器設定檔名稱**	**置換特定佇列管理程式的設定檔**
連線安全	hlq.NO.CONNECT.CHECKS	qmgr-name.YES.CONNECT.CHECKS
佇列安全	hlq.NO.QUEUE.CHECKS	qmgr-name.YES.QUEUE.CHECKS
處理程序安全	hlq.NO.PROCESS.CHECKS	qmgr-name.YES.PROCESS.CHECKS
名單安全	hlq.NO.NLIST.CHECKS	qmgr-name.YES.NLIST.CHECKS
環境定義安全 (context security)	hlq.NO.CONTEXT.CHECKS	qmgr-name.YES.CONTEXT.CHECKS
替代使用者安全性 (alternate user security)	hlq.NO.ALTERNATE.USER.CHECKS	qmgr-name.YES.ALTERNATE.USER.CHECKS
指令安全	hlq.NO.CMD.CHECKS	qmgr-name.YES.CMD.CHECKS
指令資源安全	hlq.NO.CMD.RESC.CHECKS	qmgr-name.YES.CMD.RESC.CHECKS
主題安全	hlq.NO.TOPIC.CHECKS	qmgr-name.YES.TOPIC.CHECKS
註: 通用交換器設定檔，例如 hlq.NO。* * 由 IBM MQ 忽略		

例如，如果您要對佇列管理程式 QM01(其為佇列共用群組 QSG3 的成員) 執行處理程序安全檢查，但不想對群組中任何其他佇列管理程式執行處理程序安全檢查，請定義下列切換設定檔:

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

如果您想要對佇列共用群組中 QM02 以外的所有佇列管理程式執行佇列安全檢查，請定義下列切換設定檔:

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

(不需要定義佇列共用群組的設定檔，因為如果未定義設定檔，則會自動啟用檢查。)

z/OS 定義交換器的範例

不同的 IBM MQ 子系統有不同的安全需求，可以使用不同的交換器設定檔來實作。

已定義四個 IBM MQ 子系統:

- MQP1 (正式作業系統)
- MQP2 (正式作業系統)
- MQD1 (開發系統)
- MQT1 (測試系統)

所有四個佇列管理程式都是佇列共用群組 QS01 的成員。 已定義並啟動所有 IBM MQ RACF 類別。

這些子系統具有不同的安全需求:

- 正式作業系統需要完整 IBM MQ 安全檢查在兩個系統上的佇列共用群組層次都處於作用中狀態。

  這是透過指定下列設定檔來完成:

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

  這會針對佇列共用群組中的所有佇列管理程式，設定佇列共用群組層次檢查。 您不需要為正式作業佇列管理程式定義任何其他交換器設定檔，因為您想要檢查這些系統的所有項目。

- 測試佇列管理程式 MQT1 也需要完整安全檢查。 不過，因為您稍後可能想要變更此項，所以可以在佇列管理程式層次定義安全，以便您可以變更此佇列管理程式的安全設定，而不會影響佇列共用群組的其他成員。

  這是透過定義 MQT1 的 NO.QSG.CHECKS 設定檔來完成，如下所示:

```
RDEFINE MQADMIN MQT1.NO.QSG.CHECKS
```

- 開發佇列管理程式 MQD1 與佇列共用群組的其餘部分具有不同的安全需求。 它只需要連線及佇列安全處於作用中。

  作法是定義此佇列管理程式的 MQD1.YES.QMGR.CHECKS 設定檔，然後定義下列設定檔來關閉不需要檢查之資源的安全檢查:

```
RDEFINE MQADMIN MQD1.NO.CMD.CHECKS
RDEFINE MQADMIN MQD1.NO.CMD.RESC.CHECKS
RDEFINE MQADMIN MQD1.NO.PROCESS.CHECKS
RDEFINE MQADMIN MQD1.NO.NLIST.CHECKS
RDEFINE MQADMIN MQD1.NO.CONTEXT.CHECKS
RDEFINE MQADMIN MQD1.NO.ALTERNATE.USER.CHECKS
```

當佇列管理程式處於作用中狀態時，您可以發出 DISPLAY SECURITY MQSC 指令來顯示現行安全設定。

您也可以透過在 MQADMIN 類別中定義或刪除適當的交換器設定檔，在佇列管理程式執行時變更交換器設定。 若要對交換器設定進行作用中的變更，您必須針對 MQADMIN 類別發出 REFRESH SECURITY 指令。

如需使用 DISPLAY SECURITY 和 REFRESH SECURITY 指令的詳細資料，請參閱 第 210 頁的『重新整理 z/OS 上的佇列管理程式安全』 。

z/OS 用來控制 IBM MQ 資源存取權的設定檔

除了可能已定義的交換器設定檔之外，您還必須定義 RACF 設定檔來控制對 IBM MQ 資源的存取權。 此主題集合包含不同 IBM MQ 資源類型的 RACF 設定檔相關資訊。

如果您未針對特定安全檢查定義資源設定檔，且使用者發出涉及進行該檢查的要求，則 IBM MQ 會拒絕存取。 您不需要定義與已取消啟動的任何安全交換器相關之安全類型的設定檔。

z/OS 連線安全的設定檔

如果連線安全在作用中，您必須在 MQCONN 類別中定義設定檔，並允許必要的群組或使用者 ID 存取這些設定檔，以便它們可以連接至 IBM MQ。

若要建立連線，您必須授與使用者 RACF 對適當設定檔的 READ 存取權。 (如果佇列管理程式層次設定檔不存在，且您的佇列管理程式是佇列共用群組的成員，則在安全設定為執行此動作時，可能會針對佇列共用群組層次設定檔進行檢查。)

以佇列管理程式名稱限定的連線設定檔會控制對特定佇列管理程式的存取權，且獲授與此設定檔存取權的使用者可以連接至該佇列管理程式。 以佇列共用群組名稱限定的連線設定檔會控制該連線類型之佇列共用群組內所有佇列管理程式的存取權。 例如，具有 QS01.BATCH 存取權的使用者可以對佇列共用群組 QS01 中未定義佇列管理程式層次設定檔的任何佇列管理程式使用批次連線。

註:

1. 如需針對不同安全要求所檢查之使用者 ID 的相關資訊，請參閱 第 200 頁的『z/OS 上用於安全檢查的使用者 ID』 。
2. 也會在連線時進行資源層次安全 (RESLEVEL) 檢查。 如需詳細資料，請參閱 第 196 頁的『RESLEVEL 安全設定檔』 。

IBM MQ 安全可辨識下列不同類型的連線:

- 批次 (及批次類型) 連線，包括:
  - z/OS 批次工作
  - TSO 應用程式
  - z/OS UNIX System Services 登入

  - Db2 儲存程序
- CICS 連線
- 來自控制項和應用程式處理區域的 IMS 連線
- IBM MQ 通道起始程式

z/OS 批次連線的連線安全設定檔

用於檢查批次類型連線的設定檔由佇列管理程式或佇列共用群組名稱後面接著單字 *BATCH* 組成。 將連線設定檔的 READ 存取權提供給與連接位址空間相關聯的使用者 ID。

用於檢查批次和批次類型連線的設定檔格式如下:

```
hlq.BATCH
```

其中 `hlq` 可以是 `qmgr-name` (佇列管理程式名稱) 或 `qsg-name` (佇列共用群組名稱)。 如果您同時使用佇列管理程式及佇列共用群組層次安全， IBM MQ 會檢查以佇列管理程式名稱為字首的設定檔。 如果找不到，它會尋找以佇列共用群組名稱為字首的設定檔。 如果找不到任一設定檔，連線要求會失敗。

對於批次或批次類型連線要求，您必須允許與連接位址空間相關聯的使用者 ID 存取連線設定檔。 例如，下列 RACF 指令容許 CONNTQM1 群組中的使用者連接至佇列管理程式 TQM1; 將允許這些使用者 ID 使用任何批次或批次類型連線。

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

z/OS 在本端連結的應用程式上使用 ***CHCKLOCL***

CHCKLOCL 僅適用於透過 BATCH 連線建立的連線，不適用於從 CICS 或 IMS 建立的連線。 透過通道起始程式建立的連線由 **CHCKCLNT** 控制。

概觀

如果您要配置 z/OS 佇列管理程式，以強制使用者 ID 及密碼檢查部分 (而非全部) 本端連結的應用程式，則需要執行一些其他配置。

原因是一旦配置 **CHCKLOCL** (*REQUIRED*)，使用 MQCONN API 呼叫的舊式批次應用程式就無法再連接至佇列管理程式。

僅適用於 z/OS，基於位址空間連線安全的更精細機制可用來針對明確定義的使用者 ID，將廣域 CHCKLOCL (REQUIRED) 配置降級至 CHCKLOCL (OPTIONAL)。 所使用的機制與範例一起在下列文字中說明。

為了讓 **CHCKLOCL** (*REQUIRED*) 比僅 EVERYONE 更精細，您修改 **CHCKLOCL** 的方式與修改 MQCONN 類別中 `hlq.batch` 連線設定檔之連接位址空間相關聯使用者 ID 的存取層次相同。

如果位址空間使用者 ID 僅具有 READ 存取權 (這是您完全能夠連接所需的最低讀取權)，則 **CHCKLOCL** 配置會以書面方式套用。

如果位址空間使用者 ID 具有 UPDATE 存取權 (或更高版本)，則 **CHCKLOCL** 配置會以 *OPTIONAL* 模式運作。 也就是說，您不需要提供使用者 ID 和密碼，但如果您提供，使用者 ID 和密碼必須是有效的配對。

已針對 z/OS 佇列管理程式配置連線安全

如果您已配置 z/OS 佇列管理程式的連線安全，且想要 **CHCKLOCL** (*REQUIRED*) 套用至 WAS 本端連結應用程式，但不套用其他應用程式，請執行下列步驟:

1. 以 **CHCKLOCL** (*OPTIONAL*) 作為配置開頭。 這表示會檢查所提供的任何使用者 ID 和密碼是否有效，但不會強制。
2. 透過發出下列指令，列出有權存取連線安全設定檔的所有使用者:

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

   此指令會顯示，例如:

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

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

3. 對於每一個列出為具有 READ 存取權的使用者 ID，將存取權變更為

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

4. 將 IBM MQ 配置更新為 **CHCKLOCL** (*REQUIRED*)。

   MQ23.BATCH 的 UPDATE 存取權與現行設定的組合表示您正在使用 **CHCKLOCL** (*OPTIONAL*)。

5. 現在，將 **CHCKLOCL** (*REQUIRED*) 行為套用至某個特定使用者 ID (例如 WASUSER)，因此來自該區域的所有連線都必須提供使用者 ID 及密碼。

   透過發出下列指令來反轉您先前所做的變更，以執行此動作:

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

未配置 z/OS 佇列管理程式的連線安全

在此情況下，您必須:

1. 透過發出下列指令，在 MQCONN 類別中建立 hlq.BATCH 的連線設定檔:

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

2. 授權所有建立佇列管理程式批次連線的使用者 ID，讓他們具有此設定檔的 UPDATE 存取權。 這樣做會在連線時略過使用者 ID 及密碼的 **CHCKLOCL** (*REQUIRED*) 需求。

   透過發出下列指令來執行此動作:

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

   這些包括使用者 ID:

   a. 用於 CSQUUTIL、 ISPF 面板和其他本地綁定工具。

   b. 與批次 (例如佇列管理程式的連線) 相關聯。 例如， Advanced Message Security、 IBM Integration Bus、 Db2 儲存程序、 z/OS UNIX System Services 和 TSO 使用者，以及 Java 應用程式

3. 發出下列指令，以刪除佇列管理程式的交換器設定檔:

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

4. 現在，將 **CHCKLOCL** (*REQUIRED*) 行為套用至某個特定使用者 ID (例如 WASUSER)，因此來自該區域的所有連線都必須提供使用者 ID 及密碼。

   透過發出下列指令來反轉您先前所做的變更，以執行此動作:

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

z/OS *CICS* 連線的連線安全設定檔

用於檢查 CICS 連線的設定檔由佇列管理程式或佇列共用群組名稱後面接著單字 *CICS* 組成。 將連線設定檔的 READ 存取權提供給與 CICS 位址空間相關聯的使用者 ID。

用於檢查來自 CICS 的連線的設定檔採用下列格式:

```
hlq.CICS
```

其中 hlq 可以是 qmgr-name (佇列管理程式名稱) 或 qsg-name (佇列共用群組名稱)。 如果您同時使用佇列管理程式及佇列共用群組層次安全， IBM MQ 會檢查以佇列管理程式名稱為字首的設定檔。 如果找不到，它會尋找以佇列共用群組名稱為字首的設定檔。 如果它找不到任一設定檔，則連線要求會失敗

對於 CICS 的連線要求，您只需要允許 CICS 位址空間使用者 ID 存取連線設定檔。

例如， 下列 RACF 指令容許 CICS 位址空間使用者 ID KCBCICS 連接至佇列管理程式 TQM1:

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

z/OS *IMS* 連線的連線安全設定檔

用於檢查 IMS 連線的設定檔由佇列管理程式或佇列共用群組名稱後面接著單字 *IMS* 組成。 授與 IMS 控制項和相依區域使用者 ID 對連線設定檔的 READ 存取權。

用於檢查來自 IMS 的連線的設定檔採用下列格式:

```
hlq.IMS
```

其中 hlq 可以是 qmgr-name (佇列管理程式名稱) 或 qsg-name (佇列共用群組名稱)。 如果您同時使用佇列管理程式及佇列共用群組層次安全， IBM MQ 會檢查以佇列管理程式名稱為字首的設定檔。 如果找不到，它會尋找以佇列共用群組名稱為字首的設定檔。 如果它找不到任一設定檔，則連線要求會失敗

對於 IMS 的連線要求，允許存取 IMS 控制項及相依區域使用者 ID 的連線設定檔。

例如， 下列 RACF 指令容許:

- IMS 區域使用者 ID IMSREG , 以連接至佇列管理程式 TQM1。
- 群組 BMPGRP 中用來提交 BMP 工作的使用者。

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

z/OS 通道起始程式的連線安全設定檔

用於檢查來自通道起始程式之連線的設定檔由佇列管理程式或佇列共用群組名稱後面接著單字 *CHIN* 組成。 將連線設定檔的 READ 存取權提供給通道起始程式作業位址空間所使用的使用者 ID。

用於從通道起始程式檢查連線的設定檔格式如下:

```
hlq.CHIN
```

其中 hlq 可以是 qmgr-name (佇列管理程式名稱) 或 qsg-name (佇列共用群組名稱)。 如果您同時使用佇列管理程式及佇列共用群組層次安全， IBM MQ 會檢查以佇列管理程式名稱為字首的設定檔。 如果找不到，它會尋找以佇列共用群組名稱為字首的設定檔。 如果它找不到任一設定檔，則連線要求會失敗

對於通道起始程式的連線要求，請定義通道起始程式啟動作業位址空間所使用之使用者 ID 的連線設定檔存取權。

例如， 下列 RACF 指令容許以使用者 ID DQCTRL 執行的通道起始程式位址空間連接至佇列管理程式 TQM1:

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

z/OS 佇列安全的設定檔

如果佇列安全在作用中，您必須在適當的類別中定義設定檔，並允許必要的群組或使用者 ID 存取這些設定檔。 佇列安全設定檔是以佇列管理程式或佇列共用群組及要開啟的佇列來命名。

如果佇列安全在作用中，您必須:

- 如果使用大寫設定檔，請在 **MQQUEUE** 或 **GMQQUEUE** 類別中定義設定檔。
- 如果使用大小寫混合格式設定檔，請在 **MXQUEUE** 或 **GMXQUEUE** 類別中定義設定檔。
- 允許必要的群組或使用者 ID 存取這些設定檔，以便它們可以發出使用佇列的 IBM MQ API 要求。

佇列安全的設定檔採用下列格式:

```
hlq.queuename
```

其中 hlq 可以是 qmgr-name (佇列管理程式名稱) 或 qsg-name (佇列共用群組名稱)，而 queuename 是所開啟佇列的名稱，如 MQOPEN 或 MQPUT1 呼叫的物件描述子中所指定。

以佇列管理程式名稱為字首的設定檔會控制對該佇列管理程式上單一佇列的存取權。 以佇列共用群組名稱為字首的設定檔可控制存取佇列共用群組內所有佇列管理程式上具有該佇列名稱的一或多個佇列，或群組內任何佇列管理程式對共用佇列的存取權。 透過在個別佇列管理程式上定義該佇列的佇列管理程式層次設定檔，可以在該佇列管理程式上置換此存取權。

如果您的佇列管理程式是佇列共用群組的成員，且您同時使用佇列管理程式及佇列共用群組層次安全，則 IBM MQ 會先檢查字首為佇列管理程式名稱的設定檔。 如果找不到，它會尋找以佇列共用群組名稱為字首的設定檔。

如果您使用共用佇列，建議您使用佇列共用群組層次安全。

如需佇列名稱為別名或模型佇列 z/OS 的佇列安全運作方式的詳細資料，請參閱 第 170 頁的『別名佇列的考量』 及 第 171 頁的『模型佇列的考量』 。

開啟佇列所需的 RACF 存取權取決於指定的 MQOPEN 或 MQPUT1 選項。 如果多個 MQOO_ * 及 MQPMO_ * 選項已編碼，則會針對所需的最高 RACF 權限執行佇列安全檢查。

表 *31:* 使用 *MQOPEN* 或 *MQPUT1* 呼叫的佇列安全存取層次

MQOPEN 或 MQPUT1 選項	**RACF hlq.queuename 所需的存取層次**
MQ 瀏覽	READ
MQOO_INQUIRE	READ
MQOO_BIND_ *	UPDATE
MQOO_INPUT_ *	UPDATE
MQOO_OUTPUT 或 MQPUT1	UPDATE
MQOO_PASS_ALL_CONTEXT MQPMO_PASS_ALL_CONTEXT	UPDATE
MQOO_PASS_IDENTITY_CONTEXT MQPMO_PASS_IDENTITY_CONTEXT	UPDATE
MQOO_SAVE_ALL_CONTEXT	UPDATE
MQOO_SET_IDENTITY_CONTEXT MQPMO_SET_IDENTITY_CONTEXT	UPDATE
MQOO_SET_ALL_CONTEXT MQPMO_SET_ALL_CONTEXT	UPDATE
MQOO_SET	ALTER

例如，在 IBM MQ 佇列管理程式 QM77 上，將為 RACF 群組 PAYGRP 中的所有使用者 ID 提供存取權，以從所有名稱以 'PAY.' 開頭的佇列取得訊息，或將訊息放置到所有名稱以 'PAY.' 開頭的佇列。 您可以使用下列 RACF 指令來執行此動作:

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

此外， PAYGRP 群組中的所有使用者 ID 都必須具有存取權，才能將訊息放置在未遵循 PAY 命名慣例的佇列上。 例如:

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

若要這樣做，您可以在 GMQQUEUE 類別中定義這些佇列的設定檔，並授與該類別的存取權，如下所示:

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

註:

1. 如果變更應用程式對佇列安全設定檔的 RACF 存取層次，則變更只會對該佇列取得的任何新物件控點 (即新的 MQOPEN) 生效。 變更時已存在的那些控點會保留其對佇列的現有存取權。 如果應用程式需要使用其對佇列的已變更存取層次，而不是其現有的存取層次，則它必須針對每一個需要變更的物件控點關閉並重新開啟佇列。
2. 在此範例中，佇列管理程式名稱 QM77 也可以是佇列共用群組的名稱。

開啟佇列時也可能會發生其他類型的安全檢查，視指定的開啟選項及作用中的安全類型而定。 z/OS 另請參閱 第 183 頁的『環境定義安全的設定檔』 和 第 181 頁的『替代使用者安全的設定檔』。 如需摘要表格，其中顯示當佇列、環境定義及替代使用者安全都在作用中時所需的開啟選項及安全授權，請參閱 第 175 頁的表 36。

如果您是使用發佈/訂閱，則必須考量下列各項。 處理 MQSUB 要求時，會執行安全檢查，以確保提出要求的使用者 ID 具有將訊息放置到目標 IBM MQ 佇列的必要存取權，以及訂閱 IBM MQ 主題的必要存取權。

表 *32:* 使用 *MQSUB* 呼叫的佇列安全存取層次

MQSUB 選項	**RACF hlq.queuename 所需的存取層次**
MQSO_ALTER、MQSO_CREATE 及 MQSO_RESUME	UPDATE

註:

1. hlq.queuename 是發佈的目的地佇列。 當這是受管理佇列時，您需要存取要用於所建立受管理佇列及動態佇列的適當模型佇列。
2. 如果您想要區分進行訂閱的使用者，以及從目的地佇列擷取發佈的使用者，您可以對 MQSUB API 呼叫所提供的目的地佇列使用這類技術。

z/OS 別名佇列的考量

當您對別名佇列發出 MQOPEN 或 MQPUT1 呼叫時， IBM MQ 會對呼叫上物件描述子 (MQOD) 中指定的佇列名稱進行資源檢查。 它不會檢查是否容許使用者存取目標佇列名稱。

例如，稱為 PAYROLL.REQUEST 解析為 PAY.REQUEST。 如果佇列安全處於作用中，則只需要授權您存取佇列 PAYROLL.REQUEST。 不會檢查您是否已獲授權存取佇列 PAY.REQUEST。

z/OS 使用別名佇列來區分 *MQGET* 與 *MQPUT* 要求

如果您想要將佇列的存取限制為僅容許 **MQPUT** 呼叫或僅容許 **MQGET** 呼叫，則在一個存取層次中可用的 MQI 呼叫範圍可能會導致問題。 可以透過定義兩個解析至該佇列的別名來保護佇列: 一個可讓應用程式從佇列取得訊息，另一個可讓應用程式將訊息放置在佇列上。

下列文字提供一個範例，說明如何將佇列定義給 IBM MQ:

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

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

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

您也必須建立下列 RACF 定義:

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

然後確保沒有使用者可以存取佇列 hlq.MUST_USE_ALIAS_TO_ACCESS，並將別名的存取權提供給適當的使用者或群組。 您可以使用下列 RACF 指令來執行此動作:

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

這表示只容許群組 GETGRP 中的使用者 ID GETUSER 及使用者 ID 透過別名佇列 USE_THIS_ONE_FOR_GETS 取得 MUST_USE_ALIAS_TO_ACCESS 上的訊息; 而群組 PUTGRP 中的使用者 ID PUTUSER 及使用者 ID 只容許透過別名佇列 USE_THIS_ONE_FOR_PUTS 放置訊息。

註:

1. 如果您想要使用這樣的技術，則必須通知您的應用程式開發人員，以便他們可以適當地設計其程式。
2. 如果您想要區分進行訂閱的使用者與從目的地佇列「取得」發佈的使用者，您可以對 MQSUB API 要求所提供的目的地佇列使用類似的技術。

z/OS 模型佇列的考量

若要開啟模型佇列，您必須能夠同時開啟模型佇列本身及其解析成的動態佇列。 定義動態佇列的通用 RACF 設定檔，包括 IBM MQ 公用程式所使用的動態佇列。

當您開啟模型佇列時， IBM MQ 安全會進行兩項佇列安全檢查:

1. 您是否已獲授權存取模型佇列?
2. 您是否已獲授權存取模型佇列解析成的動態佇列?

如果動態佇列名稱包含尾端星號 (*) 字元，則此 * 會取代為 IBM MQ 所產生的字串，以建立具有唯一名稱的動態佇列。 不過，因為會使用完整名稱 (包括這個產生的字串) 來檢查權限，所以您應該定義這些佇列的通用設定檔。

例如， MQOPEN 呼叫使用模型佇列名稱 CREDIT.CHECK.REPLY.MODEL 及 CREDIT.REPLY.* 在佇列管理程式 (或佇列共用群組) 上 MQSP。

若要這樣做，您必須發出下列 RACF 指令，以定義必要的佇列設定檔:

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

您也必須發出對應的 RACF PERMIT 指令，以容許使用者存取這些設定檔。

MQOPEN 所建立的一般動態佇列名稱類似於 CREDIT.REPLY.A346EF00367849A0。 無法預期最後一個限定元的精確值; 這就是您應該對這類佇列名稱使用通用設定檔的原因。

許多 IBM MQ 公用程式在動態佇列上放置訊息。 您應該定義下列動態佇列名稱的設定檔，並提供相關使用者 ID 的 RACF UPDATE 存取權 (如需正確的使用者 ID，請參閱 第 200 頁的『z/OS 上用於安全檢查的使用者 ID』):

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

您也可以考慮定義設定檔，以控制應用程式設計副本成員中預設使用的動態佇列名稱。 IBM MQ 提供的記錄定義檔包含預設 *DynamicQName*，即 CSQ. *。 這可讓您建立適當的 RACF 設定檔。

註: 不容許應用程式設計師為動態佇列名稱指定單一 *。 如果您這麼做，則必須定義 hlq. ** MQQUEUE 類別中的設定檔，您必須提供廣泛的存取權。 這表示此設定檔也可以用於沒有更具體 RACF 設定檔的其他非動態佇列。 因此，您的使用者可以存取您不希望他們存取的佇列。

z/OS 關閉永久動態佇列上的選項

如果應用程式開啟由另一個應用程式建立的永久動態佇列，然後嘗試使用 MQCLOSE 選項來刪除該佇列，則在進行嘗試時，會套用一些額外的安全檢查。

表 *33:* 永久動態佇列上關閉選項的存取層次

MQCLOSE 選項	**RACF hlq.queuename 所需的存取層次**
MQCO_DELETE	ALTER
MQCO_DELETE_PURGE	ALTER

z/OS 安全及遠端佇列

當訊息放置在遠端佇列上時，本端佇列管理程式所實作的佇列安全取決於遠端佇列在開啟時的指定方式。

會套用下列規則:

1. 如果已透過 IBM MQ DEFINE QREMOTE 指令在本端佇列管理程式上定義遠端佇列，則所檢查的佇列是遠端佇列的名稱。 例如，如果在佇列管理程式 MQS1 上定義遠端佇列，如下所示:

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

   在此情況下，是 BANK7.CREDIT.REFERENCE 。
2. 如果請求的 *ObjectQMgrName* 未解析為本機佇列管理器，則會針對解析的（遠端）佇列管理器名稱執行安全性檢查，但在叢集佇列的情況下，根據叢集佇列名稱進行檢查。

   例如，傳輸佇列 BANK1.TO.BANK7 定義在佇列管理程式 MQS1 上。 然後，在 MQS1 上發出 MQPUT1 請求，將 *ObjectName* 指定為 BANK1.INTERBANK.TRANSFERS 並將 *ObjectQMgrName* 指定為 BANK1.TO.BANK7 。 在此情況下，執行要求的使用者必須具有 BANK1.TO.BANK7。
3. 如果您對佇列提出 MQPUT 要求，並指定 *ObjectQMgrName* 作為本端佇列管理程式的別名，則只會檢查佇列名稱是否安全，而不會檢查佇列管理程式的安全。

當訊息到達遠端佇列管理程式時，可能需要進行額外的安全處理。 如需相關資訊，請參閱 第 81 頁的『遠端傳訊的安全』。

z/OS 無法傳送郵件的佇列安全

特殊考量適用於無法傳送郵件的佇列，因為許多使用者必須能夠在其中放置訊息，但必須嚴格限制擷取訊息的存取權。 您可以將不同的 RACF 權限套用至無法傳送郵件的佇列及別名佇列，以達到此目的。

無法遞送的訊息可以放在稱為無法傳送郵件之佇列的特殊佇列中。 如果您具有可能在此佇列上結束的機密資料，則必須考量此作業的安全含意，因為您不想要未獲授權的使用者擷取此資料。

必須容許下列每一個項目將訊息放入無法傳送郵件的佇列:

- 應用程式。
- 通道起始程式位址空間及任何 MCA 使用者 ID。 (如果 RESLEVEL 設定檔不存在，或已定義為檢查通道使用者 ID，則通道使用者 ID 也需要權限，才能將訊息放置在無法傳送郵件的佇列上。)
- CKTI， CICS 提供的 CICS 作業起始器。
- CSQQTRMN， IBM MQ 提供的 IMS 觸發監視器。

唯一可以從無法傳送郵件的佇列擷取訊息的應用程式應該是處理這些訊息的「特殊」應用程式。 不過，如果您提供應用程式 RACF UPDATE 權限給 MQPUT 的無法傳送郵件的佇列，則會發生問題，因為它們隨後可以使用 MQGET 呼叫自動從佇列中擷取訊息。 您無法針對取得作業停用無法傳送郵件的佇列，因為如果您這麼做，即使「特殊」應用程式也無法擷取訊息。

此問題的一個解決方案是對無法傳送郵件的佇列設定兩層存取權。 CKTI、訊息通道代理程式交易或通道起始程式位址空間，以及「特殊」應用程式具有直接存取權; 其他應用程式只能透過別名佇列來存取無法傳送郵件的佇列。 此別名定義為容許應用程式將訊息放置在無法傳送郵件的佇列中，但不容許應用程式從中取得訊息。

這就是它的運作方式:

1. 使用屬性 PUT (ENABLED) 及 GET (ENABLED) 來定義實際無法傳送郵件的佇列，如範例 thlqual.SCSQPROC(CSQ4INYG) 所示。
2. 將無法傳送郵件之佇列的 RACF UPDATE 權限授與下列使用者 ID:
   - CKTI 及 MCA 或通道起始程式位址空間執行所在的使用者 ID。
   - 與「特殊」無法傳送郵件的佇列處理應用程式相關聯的使用者 ID。
3. 定義別名佇列以解析為實際無法傳送郵件的佇列，但將下列屬性提供給別名佇列 :PUT (ENABLED) 及 GET (DISABLED)。 為別名佇列提供與無法傳送郵件的佇列名稱具有相同詞幹的名稱，但將字元 ". PUT" 附加至此詞幹。 例如，如果無法傳送郵件的佇列名稱是 hlq.DEAD.QUEUE，別名佇列名稱會是 hlq.DEAD.QUEUE.PUT。
4. 如果要將訊息放在無法傳送郵件的佇列中，應用程式會使用別名佇列。 您的應用程式必須執行下列動作:
   - 擷取實際無法傳送郵件的佇列名稱。 為此，它會使用 MQOPEN 開啟佇列管理程式物件，然後發出 MQINQ 以取得無法傳送郵件的佇列名稱。
   - 透過將字元 '.PUT' 附加至此名稱 (在本例中為 hlq.DEAD.QUEUE.PUT。
   - 開啟別名佇列 hlq.DEAD.QUEUE.PUT。
   - 針對別名佇列發出 MQPUT，將訊息放置在實際無法傳送郵件的佇列中。
5. 將與應用程式相關聯的使用者 ID RACF UPDATE 權限授與別名，但對實際無法傳送郵件的佇列沒有存取權 (權限 NONE)。 這表示:
   - 應用程式可以使用別名佇列將訊息放入無法傳送郵件的佇列。
   - 應用程式無法使用別名佇列從無法傳送郵件的佇列取得訊息，因為別名佇列已停用取得作業。

   應用程式無法從實際無法傳送郵件的佇列取得任何訊息，因為它確實具有正確的 RACF 權限。

第 173 頁的表 34 彙總此解決方案中各種參與者所需的 RACF 權限。

表 *34:* 對無法傳送郵件之佇列及其別名的 *RACF* 權限

相關聯的使用者 ID	**實際無法傳送郵件的佇列 (hlq.DEAD.QUEUE)**	**別名無法傳送郵件的佇列 (hlq.DEAD.QUEUE.PUT)**
MCA 或通道起始程式位址空間及 CKTI	UPDATE	無
「特殊」應用程式 (適用於無法傳送郵件的佇列處理)	UPDATE	無

表 *34:* 對無法傳送郵件之佇列及其別名的 *RACF* 權限 *(*繼續*)*		
相關聯的使用者 ID	**實際無法傳送郵件的佇列 (hlq.DEAD.QUEUE)**	**別名無法傳送郵件的佇列 (hlq.DEAD.QUEUE.PUT)**
使用者撰寫的應用程式使用者 ID	無	UPDATE

如果您使用此方法，則應用程式無法判定無法傳送郵件之佇列的訊息長度上限 (MAXMSGL)。 這是因為無法從別名佇列擷取 MAXMSGL 屬性。 因此，您的應用程式應該假設訊息長度上限為 100 MB，即 IBM MQ for z/OS 支援的大小上限。 實際無法傳送郵件的佇列也應該以 100 MB 的 MAXMSGL 屬性來定義。

註: 使用者撰寫的應用程式通常不會使用替代使用者權限，將訊息放置在無法傳送郵件的佇列上。 這會減少可存取無法傳送郵件的佇列的使用者 ID 數目。

z/OS 系統佇列安全

您必須設定 RACF 存取權，以容許特定使用者 ID 存取特定系統佇列。

IBM MQ 的輔助組件會存取許多系統佇列:

- CSQUTIL 公用程式
- 訊息安全原則公用程式 (CSQ0UTIL)
- 作業及控制面板
- 通道起始程式位址空間 (包括「排入佇列的發佈/訂閱常駐程式」)
- mqweb 伺服器，由 MQ Console 和 REST API 使用。

必須為這些執行所使用的使用者 ID 提供這些佇列的 RACF 存取權，如 第 174 頁的表 35 中所示。

表 *35: IBM MQ* 需要存取 *SYSTEM* 佇列					
系統佇列	**CSQUTIL**	**CSQ0UTIL**	**mqweb 伺服器**	**作業及控制面板**	**分散式佇列的通道起始程式**
SYSTEM.ADMIN.CHANNEL.EVENT	-	-	-	-	UPDATE
SYSTEM.ADMIN.COMMAND.QUEUE	-	-	UPDATE	-	-
SYSTEM.BROKER.ADMIN.STREAM	-	-	-	-	ALTER
SYSTEM.BROKER.CONTROL.QUEUE	-	-	-	-	ALTER
SYSTEM.BROKER.DEFAULT.STREAM	-	-	-	-	ALTER
SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS	-	-	-	-	UPDATE
SYSTEM.CHANNEL.INITQ	-	-	-	-	UPDATE
SYSTEM.CHANNEL.SYNCQ	-	-	-	-	UPDATE
SYSTEM.CLUSTER.COMMAND.QUEUE	-	-	-	-	ALTER
SYSTEM.CLUSTER.REPOSITORY.QUEUE	-	-	-	-	UPDATE
SYSTEM.CLUSTER.TRANSMIT.QUEUE	-	-	-	-	ALTER
SYSTEM.COMMAND.INPUT	UPDATE	-	-	UPDATE	UPDATE
SYSTEM.COMMAND.REPLY.*	-	-	-	-	UPDATE
SYSTEM.COMMAND.REPLY.MODEL	UPDATE	-	-	UPDATE	UPDATE
SYSTEM.CSQOREXX.*	-	-	-	UPDATE	-
SYSTEM.CSQUTIL.*	UPDATE	-	-	-	-
SYSTEM.CSQXCMD.*	-	-	-	-	UPDATE

表 35: IBM MQ 需要存取 SYSTEM 佇列 (繼續)					
系統佇列	**CSQUTIL**	**CSQ0UTIL**	**mqweb 伺服器**	**作業及控制面板**	**分散式佇列的通道起始程式**
SYSTEM.HIERARCHY.STATE	-	-	-	-	UPDATE
SYSTEM.INTER.QMGR.CONTROL	-	-	-	-	UPDATE
SYSTEM.INTER.QMGR.PUBS	-	-	-	-	UPDATE
SYSTEM.INTER.QMGR.FANREQ	-	-	-	-	UPDATE
SYSTEM.PROTECTION.ERROR.QUEUE	-	-	-	-	UPDATE
SYSTEM.PROTECTION.POLICY.QUEUE	-	更新第 175 頁的『1』	-	-	READ
SYSTEM.QSG.CHANNEL.SYNCQ	-	-	-	-	UPDATE
SYSTEM.QSG.TRANSMIT.QUEUE	-	-	-	-	UPDATE
SYSTEM.REST.REPLY.QUEUE	-	-	UPDATE	-	-
SYSTEM.BLUEMIX.REGISTRATION.QUEUE	-	-	-	-	UPDATE

附註:

1. Advanced Message Security 位址空間使用者也需要此佇列的 READ 存取權。

z/OS *API*-資源安全存取快速參照

MQOPEN、 **MQPUT1**、 **MQSUB** 及 **MQCLOSE** 選項的摘要，以及不同資源安全類型所需的存取權。

表 36: *MQOPEN*、 *MQPUT1*、*MQSUB* 和 *MQCLOSE* 選項以及所需的安全授權. 如此顯示的圖說文字 **(1)** ，請參閱此表格後面的附註。				
	需要最低 RACF 存取層次			
RACF 類別:	**MXTOPIC**	**MQQUEUE 或 MXQUEUE (1)**	**MQADMIN 或 MXADMIN**	**MQADMIN 或 MXADMIN**
RACF profile:	**(15 或 16)**	**(2)**	**(3)**	**(4)**
MQOPEN 選項				
MQOO_INQUIRE		READ **(5)**	不檢查	不檢查
MQ 瀏覽		READ	不檢查	不檢查
MQOO_INPUT_ *		UPDATE	不檢查	不檢查
MQOO_SAVE_ALL_CONTEXT **(6)**		UPDATE	不檢查	不檢查
MQOO_OUTPUT (USAGE = NORMAL) **(7)**		UPDATE	不檢查	不檢查
MQOO_PASS_IDENTITY_CONTEXT **(8)**		UPDATE	READ	不檢查
MQOO_PASS_ALL_CONTEXT **(8) (9)**		UPDATE	READ	不檢查
MQOO_SET_IDENTITY_CONTEXT **(8) (9)**		UPDATE	UPDATE	不檢查
MQOO_SET_ALL_CONTEXT **(8) (10)**		UPDATE	CONTROL	不檢查
MQOO_OUTPUT (USAGE (XMITQ) **(11)**		UPDATE	CONTROL	不檢查

表 *36: MQOPEN*、 *MQPUT1*、*MQSUB* 和 *MQCLOSE* 選項以及所需的安全授權. 如此顯示的圖說文字 **(1)** ，請參閱此表格後面的附註。 *(繼續)*

	需要最低 RACF 存取層次			
RACF 類別:	**MXTOPIC**	**MQQUEUE 或 MXQUEUE (1)**	**MQADMIN 或 MXADMIN**	**MQADMIN 或 MXADMIN**
RACF profile:	**(15 或 16)**	**(2)**	**(3)**	**(4)**
MQOO_OUTPUT (主題物件)	UPDATE **(16)**			
MQOO_OUTPUT (主題物件的別名佇列)	UPDATE **(16)**	UPDATE		
MQOO_SET		ALTER	不檢查	不檢查
MQOO_ALTERNATE_USER_AUTHORITY		**(12)**	**(12)**	UPDATE
MQPUT1 選項				
放入一般佇列 **(7)**		UPDATE	不檢查	不檢查
MQPMO_PASS_IDENTITY_CONTEXT		UPDATE	READ	不檢查
MQPMO_PASS_ALL_CONTEXT		UPDATE	READ	不檢查
MQPMO_SET_IDENTITY_CONTEXT		UPDATE	UPDATE	不檢查
MQPMO_SET_ALL_CONTEXT		UPDATE	CONTROL	不檢查
MQOO_OUTPUT 放置在傳輸佇列 **(11)**		UPDATE	CONTROL	不檢查
MQOO_OUTPUT (主題物件)	UPDATE **(16)**			
MQOO_OUTPUT (主題物件的別名佇列)	UPDATE **(16)**	UPDATE		
MQPMO_ALTERNATE_USER_AUTHORITY		**(13)**	**(13)**	UPDATE
MQCLOSE 選項				
MQCO_DELETE **(14)**		ALTER	不檢查	不檢查
MQCO_DELETE_PURGE **(14)**		ALTER	不檢查	不檢查
MQCO_REMOVE_SUB	ALTER **(15)**			
MQSUB 選項				
MQSO_CREATE	ALTER **(15)**	**(17)**	**(18)**	
MQSO_ALTER	ALTER **(15)**	**(17)**	**(18)**	
MQ 回復	READ **(15)**	**(17)**	不檢查	
MQSO_ALTERNATE_USER_AUTHORITY				UPDATE
MQSO_SET_IDENTITY_CONTEXT			**(18)**	

註:

1. 此選項不限於佇列。 將 MQNLIST 或 MXNLIST 類別用於名稱清單，並將 MQPROC 或 MXPROC 類別用於處理程序。
2. 使用 RACF 設定檔: hlq.resourcename
3. 使用 RACF 設定檔: hlq.CONTEXT.queuename
4. 使用 RACF 設定檔: hlq.ALTERNATE.USER。 `alternateuserid`

alternateuserid 是在物件描述子的 *AlternateUserId* 欄位中指定的使用者 ID。 請注意，此檢查最多使用 *AlternateUserId* 欄位的 12 個字元，不像其他檢查只使用使用者 ID 的前 8 個字元。

5. 開啟佇列管理程式進行查詢時，不會進行任何檢查。
6. 也必須指定 MQOO_INPUT_ *。 這適用於本端、模型或別名佇列。
7. 這項檢查是針對 **Usage** 佇列屬性為 MQUS_NORMAL 的本端或模型佇列，以及別名或遠端佇列 (定義給連接的佇列管理程式)。 如果佇列是在明確指定 *ObjectQMgrName* (不是所連接佇列管理程式的名稱) 的情況下開啟的遠端佇列，則會對與 *ObjectQMgrName* 同名 (必須是 **Usage** 佇列屬性為 MQUS_TRANSMISSION 的本端佇列) 的佇列執行檢查。
8. 也必須指定 MQOO_OUTPUT。
9. 此選項也隱含 MQOO_PASS_IDENTITY_CONTEXT。
10. 此選項也隱含 MQOO_PASS_IDENTITY_CONTEXT、MQOO_PASS_ALL_CONTEXT 及 MQOO_SET_IDENTITY_CONTEXT。
11. 針對 **Usage** 佇列屬性為 MQUS_TRANSMISSION 且正在直接開啟以供輸出的本端或模型佇列執行此檢查。 如果正在開啟遠端佇列，則它不適用。
12. 至少還必須指定 MQOO_INQUIRE、MQOO_BROWSE、MQOO_INPUT_ *、MQOO_OUTPUT 或 MQOO_SET 其中之一。 所執行的檢查與其他指定選項的檢查相同。
13. 所執行的檢查與其他指定選項的檢查相同。
14. 這僅適用於已直接開啟 (即未透過模型佇列開啟) 的永久動態佇列。 刪除暫時動態佇列不需要安全。
15. 使用 RACF 設定檔 hlq.SUBSCRIBE.topicname。
16. 使用 RACF profile hlq.PUBLISH.topicname。
17. 如果您在 MQSUB 要求上指定要將發佈傳送至的目的地佇列，則會針對該佇列執行安全檢查，以確保您具有該佇列的放置權限。
18. 在 MQSUB 要求上，如果已指定 MQSO_CREATE 或 MQSO_ALTER 選項，則您想要在 MQSD 結構中設定任何身分環境定義欄位，您也需要指定 MQSO_SET_IDENTITY_CONTEXT 選項，而且也需要對目的地佇列的環境定義設定檔具有適當的權限。

z/OS 主題安全的設定檔

如果主題安全處於作用中狀態，您必須在適當的類別中定義設定檔，並允許必要的群組或使用者 ID 存取那些設定檔。

發佈/訂閱安全中說明主題樹狀結構內主題安全的概念。

如果主題安全在作用中，您必須執行下列動作:

- 在 **MXTOPIC** 或 **GMXTOPIC** 類別中定義設定檔。
- 允許必要的群組或使用者 ID 存取這些設定檔，以便他們可以發出使用主題的 IBM MQ API 要求。

主題安全的設定檔採用下列格式:

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

其中

- hlq 是 qmgr-name (佇列管理程式名稱) 或 qsg-name (佇列共用群組名稱)。
- topicname 是主題樹狀結構中主題管理節點的名稱，與透過 MQSUB 呼叫訂閱或透過 MQOPEN 呼叫發佈至的主題相關聯。

以佇列管理程式名稱為字首的設定檔會控制對該佇列管理程式上單一主題的存取權。 以佇列共用群組名稱為字首的設定檔會控制對佇列共用群組內所有佇列管理程式上具有該主題名稱的一或多個主題的存取權。 透過在個別佇列管理程式上定義該主題的佇列管理程式層次設定檔，可以在該佇列管理程式上置換此存取權。

如果您的佇列管理程式是佇列共用群組的成員，且您同時使用佇列管理程式及佇列共用群組層次安全，則 IBM MQ 會先檢查字首為佇列管理程式名稱的設定檔。 如果找不到，它會尋找以佇列共用群組名稱為字首的設定檔。

訂閱

如果要訂閱主題，您需要同時存取您嘗試訂閱的主題，以及發佈資訊的目的地佇列。

當您發出 MQSUB 要求時，會進行下列安全檢查:

- 您是否具有適當的存取層次來訂閱該主題，以及是否開啟目的地佇列 (如果已指定) 以進行輸出
- 您是否具有該目的地佇列的適當存取層次。

表 *37:* 主題安全訂閱所需的存取層次

MQSUB 選項	**需要對 MXTOPIC 類別中 hlq.SUBSCRIBE.topicname 設定檔的 RACF 存取權**
MQSO_CREATE 及 MQSO_ALTER	ALTER
MQ 回復	READ

表 *38:* 使用非受管理目的地佇列訂閱所需的其他權限

MQSUB 選項	**需要對 MQADMIN 或 MXADMIN 類別中 hlq.CONTEXT.queuename 設定檔的 RACF 存取權**
MQSO_CREATE、MQSO_ALTER 及 MQSO_RESUME	UPDATE
	MQQUEUE 或 MXQUEUE 類別中的 hlq.queuename 設定檔需要 RACF 存取權
MQSO_CREATE 及 MQSO_ALTER	UPDATE
	MQADMIN 或 MXADMIN 類別中的 hlq.ALTERNATE.USER.alternateuserid 設定檔需要 RACF 存取權
MQSO_ALTERNATE_USER_AUTHORITY	UPDATE

訂閱的受管理佇列考量

會執行安全檢查，以查看是否容許您訂閱主題。 不過，當建立受管理佇列時，並不會執行安全檢查，或判斷您是否有權將訊息放入這個目的地佇列。

您無法關閉刪除受管理佇列。

使用的模型佇列為: SYSTEM.DURABLE.MODEL.QUEUE 和 SYSTEM.NDURABLE.MODEL.QUEUE。

從這些模型佇列建立的受管理佇列格式為 SYSTEM.MANAGED.DURABLE.A346EF00367849A0 及 SYSTEM.MANAGED.NDURABLE.A346EF0036785EA0，其中最後一個限定元無法預期。

不提供任何使用者對這些佇列的存取權。 可以使用 SYSTEM.MANAGED.DURABLE.* 和 SYSTEM.MANAGED.NDURABLE.* 格式的通用設定檔來保護佇列，而不授與任何權限。

可以使用 MQSUB 要求所傳回的控點，從這些佇列擷取訊息。

如果您明確針對已指定 MQCO_REMOVE_SUB 選項的訂閱發出 MQCLOSE 呼叫，且未建立您在此控點下關閉的訂閱，則會在關閉時執行安全檢查，以確保您具有執行作業的正確權限。

表 *39:* 關閉訂閱作業所需的主題安全設定檔存取層次

MQCLOSE 選項	**需要對 MXTOPIC 類別中 hlq.SUBSCRIBE.topicname 設定檔的 RACF 存取權**
MQCO_REMOVE_SUB	ALTER

發佈

若要發佈主題，您需要存取主題，如果您使用別名佇列，也需要存取別名佇列。

表 *40:* 主題安全發佈所需的存取層次	
MQOPEN 或 MQPUT1 選項	**需要對 MXTOPIC 類別中 hlq.PUBLISH.topicname 設定檔的 RACF 存取權**
MQOO_OUTPUT 或 MQPUT1	UPDATE

表 *41:* 開啟解析為主題的別名佇列所需的存取層次	
MQOPEN 或 MQPUT1 選項	**需要 RACF 存取別名佇列之 MQQUEUE 或 MXQUEUE 類別中的 hlq.queuename 設定檔**
MQOO_OUTPUT 或 MQPUT1	UPDATE

如需在開啟解析為主題名稱的別名佇列進行發佈時主題安全如何運作的詳細資料，請參閱 第 179 頁的『解析為發佈作業主題之別名佇列的考量』。

當您考量 PUT 或 GET 限制的目的地佇列所使用的別名佇列時，請參閱 第 170 頁的『別名佇列的考量』。

如果變更應用程式對主題安全設定檔的 RACF 存取層次，則變更只會對該主題取得的任何新物件控點 (即新的 MQSUB 或 MQOPEN) 生效。 這些控點在變更時已存在，保留其對主題的現有存取權。 此外，現有訂閱者仍可存取他們已建立的任何訂閱。

解析為發佈作業主題之別名佇列的考量

當您對解析為主題的別名佇列發出 MQOPEN 或 MQPUT1 呼叫時， IBM MQ 會進行兩項資源檢查:

- 第一個針對 MQOPEN 或 MQPUT1 呼叫上物件描述子 (MQOD) 中指定的別名佇列名稱。
- 針對別名佇列所解析的主題的第二個

您必須注意，此行為與您在別名佇列解析為其他佇列時所取得的行為不同。 您需要正確存取這兩個設定檔，才能繼續執行發佈動作。

系統主題安全

通道起始程式位址空間會存取下列系統主題。

必須將這些佇列的 RACF 存取權提供給用來執行此動作的使用者 ID ，如 第 179 頁的表 42 所示。

表 *42:* 需要存取 *SYSTEM* 主題		
SYSTEM 主題	**設定檔**	**分散式佇列的通道起始程式**
SYSTEM.BROKER.ADMIN.STREAM	hlq.PUBLISH.topicname	UPDATE
SYSTEM.BROKER.ADMIN.STREAM	hlq.SUBSCRIBE.topicname	ALTER

z/OS 程序的設定檔

如果處理程序安全在作用中，您必須在適當的類別中定義設定檔，並允許必要的群組或使用者 ID 存取那些設定檔。

如果處理程序安全處於作用中狀態，則您必須:

- 如果使用大寫設定檔，請在 **MQPROC** 或 **GMQPROC** 類別中定義設定檔。
- 如果使用大小寫混合格式設定檔，請在 **MXPROC** 或 **GMXPROC** 類別中定義設定檔。
- 允許必要的群組或使用者 ID 存取這些設定檔，以便他們可以發出使用處理程序的 IBM MQ API 要求。

程序的設定檔採用下列格式:

```
hlq.processname
```

其中 hlq 可以是 qmgr-name (佇列管理程式名稱) 或 qsg-name (佇列共用群組名稱)，而 processname 是正在開啟的處理程序名稱。

以佇列管理程式名稱為字首的設定檔會控制對該佇列管理程式上單一程序定義的存取權。 以佇列共用群組名稱為字首的設定檔可控制在佇列共用群組內所有佇列管理程式上使用該名稱的一或多個程序定義的存取權。 透過在個別佇列管理程式上定義該程序定義的佇列管理程式層次設定檔，可以在該佇列管理程式上置換此存取權。

如果您的佇列管理程式是佇列共用群組的成員，且您同時使用佇列管理程式及佇列共用群組層次安全，則 IBM MQ 會先檢查字首為佇列管理程式名稱的設定檔。 如果找不到，它會尋找以佇列共用群組名稱為字首的設定檔。

下表顯示開啟程序所需的存取權。

表 *43:* 處理程序安全的存取層次

MQOPEN 選項	**RACF hlq.processname 所需的存取層次**
MQOO_INQUIRE	READ

例如，在佇列管理程式 MQS9 上， RACF 群組 INQVPRC 必須能夠查詢 (MQINQ) 在所有以字母 V 開頭的處理程序上。 此的 RACF 定義如下:

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

根據開啟程序定義物件時指定的開啟選項，替代使用者安全也可能處於作用中。

z/OS 名稱清單的設定檔

如果名單安全在作用中，您可以在適當的類別中定義設定檔，並提供這些設定檔的必要群組或使用者 ID 存取權。

如果名單安全在作用中，您必須:

- 如果使用大寫設定檔，請在 **MQNLIST** 或 **GMQNLIST** 類別中定義設定檔。
- 如果使用大小寫混合格式設定檔，請在 **MXNLIST** 或 **GMXNLIST** 類別中定義設定檔。
- 允許必要的群組或使用者 ID 存取這些設定檔。

名稱清單的設定檔採用下列格式:

```
hlq.namelistname
```

其中 hlq 可以是 qmgr-name (佇列管理程式名稱) 或 qsg-name (佇列共用群組名稱)， namelistname 是所開啟名稱清單的名稱。

以佇列管理程式名稱為字首的設定檔會控制該佇列管理程式上單一名單的存取權。 以佇列共用群組名稱為字首的設定檔可控制存取佇列共用群組內所有佇列管理程式上具有該名稱之一或多個名稱清單的存取權。 透過在個別佇列管理程式上定義該名稱清單的佇列管理程式層次設定檔，可以在該佇列管理程式上置換此存取權。

如果您的佇列管理程式是佇列共用群組的成員，且您同時使用佇列管理程式及佇列共用群組層次安全，則 IBM MQ 會先檢查字首為佇列管理程式名稱的設定檔。 如果找不到，它會尋找以佇列共用群組名稱為字首的設定檔。

下表顯示開啟名單所需的存取權。

表 *44:* 名單安全的存取層次	
MQOPEN 選項	**RACF hlq.namelistname 所需的存取層次**
MQOO_INQUIRE	READ

例如，在佇列管理程式 (或佇列共用群組) PQM3 上， RACF 群組 DEPT571 必須能夠查詢 (MQINQ) 在這些名稱清單上:

- 以 "DEPT571"開頭的所有名稱清單。
- PRINTER/DESTINATIONS/DEPT571
- AGENCY/REQUEST/QUEUES
- WAREHOUSE.BROADCAST

要執行此動作的 RACF 定義如下:

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

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

視開啟名單物件時指定的選項而定，替代使用者安全可能處於作用中。

系統名單安全

IBM MQ 的輔助組件會存取許多系統名稱清單:

- CSQUTIL 公用程式
- 作業及控制面板
- 通道起始程式位址空間 (包括「排入佇列的發佈/訂閱常駐程式」)

這些執行所使用的使用者 ID 必須獲得這些名稱清單的 RACF 存取權，如 第 181 頁的表 45 所示。

表 *45: IBM MQ* 對 *SYSTEM* 名稱清單所需的存取權			
系統名單	**CSQUTIL**	**作業及控制面板**	**分散式佇列的通道起始程式**
SYSTEM.QPUBSUB.QUEUE.NAMELIST	-	-	READ
SYSTEM.QPUBSUB.SUBPOINT.NAMELIST	-	-	READ

z/OS 替代使用者安全的設定檔

如果替代使用者安全處於作用中狀態，您必須在適當的類別中定義設定檔，並允許必要的群組或使用者 ID 存取那些設定檔。

有關 *AlternateUserId* 的更多信息，請參閱 AlternateUserID (MQCHAR12) 。

如果替代使用者安全在作用中，您必須:

- 如果您使用大寫設定檔，請在 MQADMIN 或 GMQADMIN 類別中定義設定檔。
- 如果您使用大小寫混合的設定檔，請在 MXADMIN 或 GMXADMIN 類別中定義設定檔。

允許必要的群組或使用者 ID 存取這些設定檔，以便在開啟物件時使用 ALTERNATE_USER_AUTHORITY 選項。

替代使用者安全的設定檔可以在子系統層次或佇列共用群組層次指定，並採用下列格式:

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

其中 hlq 可以是 qmgr-name (佇列管理程式名稱) 或 qsg-name (佇列共用群組名稱)，而 alternateuserid 是物件描述子中 *AlternateUserId* 欄位的值。

以佇列管理程式名稱為字首的設定檔可控制在該佇列管理程式上使用替代使用者 ID。 以佇列共用群組名稱為字首的設定檔可控制在佇列共用群組內的所有佇列管理程式上使用替代使用者 ID。 具有正確存取權的使用者可以在佇列共用群組內的任何佇列管理程式上使用此替代使用者 ID。 透過在個別佇列管理程式上定義該替代使用者 ID 的佇列管理程式層次設定檔，可以在該佇列管理程式上置換此存取權。

如果您的佇列管理程式是佇列共用群組的成員，且您同時使用佇列管理程式及佇列共用群組層次安全，則 IBM MQ 會先檢查字首為佇列管理程式名稱的設定檔。 如果找不到，它會尋找以佇列共用群組名稱為字首的設定檔。

下表顯示指定替代使用者選項時的存取權。

表 *46:* 替代使用者安全的存取層次

MQOPEN、MQSUB 或 MQPUT1 選項	**需要 RACF 存取層次**
MQOO_ALTERNATE_USER_AUTHORITY MQSO_ALTERNATE_USER_AUTHORITY MQPMO_ALTERNATE_USER_AUTHORITY	UPDATE

除了替代使用者安全檢查之外，還可以對佇列、處理程序、名單及環境定義安全進行其他安全檢查。 替代使用者 ID (如果有提供的話) 僅用於佇列、程序定義或名單資源上的安全檢查。 對於替代使用者和環境定義安全檢查，使用要求檢查的使用者 ID。 如需如何處理使用者 ID 的詳細資料，請參閱 第 200 頁的『z/OS 上用於安全檢查的使用者 ID』。 如需摘要表格，其中顯示當佇列、環境定義及替代使用者安全都在作用中時所需的開啟選項及安全檢查，請參閱 第 175 頁的表 36。

替代使用者設定檔可讓要求的使用者 ID 存取與替代使用者 ID 中指定的使用者 ID 相關聯的資源。 例如，在佇列管理程式 QMPY 上以使用者 ID PAYSERV 身分執行的薪資伺服器會處理來自人事使用者 ID 的要求，所有這些要求都以 PS 開頭。 若要讓薪資伺服器所執行的工作以發出要求之使用者的使用者 ID 來執行，則使用替代使用者權限。 薪資伺服器知道要指定哪個使用者 ID 作為替代使用者 ID，因為要求程式會使用 MQPMO_DEFAULT_CONTEXT 放置訊息選項產生訊息。 如需從何處取得替代使用者 ID 的詳細資料，請參閱 第 200 頁的『z/OS 上用於安全檢查的使用者 ID』。

下列範例 RACF 定義可讓伺服器程式指定以 PS 字元開頭的替代使用者 ID:

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

註:

1. 物件描述子及訂閱描述子中的 *AlternateUserId* 欄位長度為 12 個位元組。 所有 12 個位元組都在設定檔檢查中使用，但 IBM MQ 只會使用前 8 個位元組作為使用者 ID。 如果不需要此使用者 ID 截斷，則提出要求的應用程式必須將超過 8 個位元組的任何替代使用者 ID 轉換為更適當的使用者 ID。
2. 如果您指定 MQOO_ALTERNATE_USER_AUTHORITY、MQSO_ALTERNATE_USER_AUTHORITY 或 MQPMO_ALTERNATE_USER_AUTHORITY，且未在物件描述子中指定 *AlternateUserId* 欄位，則會使用空白的使用者 ID。 基於替代使用者安全的目的，請檢查用於 *AlternateUserId* 限定元的使用者 ID 是 -BLANK-。例如 RDEF MQADMIN hlq.ALTERNATE.USER.-BLANK-。

   如果容許使用者存取此設定檔，則會以空白的使用者 ID 進行所有進一步檢查。 如需空白使用者 ID 的詳細資料，請參閱 第 207 頁的『空白使用者 ID 和 UACC 層次』。

如果您有使用者 ID 的命名慣例，可讓您使用一般替代使用者設定檔，則替代使用者 ID 的管理會更容易。 如果沒有，您可以使用 RACF RACVARS 特性。 如需使用 RACVARS 的詳細資料，請參閱 *z/OS SecureWay Security Server RACF Security Administrator 's Guide*。

將訊息放入已使用替代使用者權限開啟的佇列，且佇列管理程式已產生訊息的環境定義時，MQMD_USER_IDENTIFIER 欄位會設為替代使用者 ID。

z/OS 環境定義安全的設定檔

如果環境定義安全在作用中，為了控制訊息環境定義資訊的存取權，您必須在適當的類別中定義設定檔，並允許必要的群組或使用者 ID 存取這些設定檔。 訊息環境定義包含在訊息描述子 (MQMD) 中。

使用環境定義安全的設定檔

如果環境定義安全在作用中，為了允許使用者存取特定佇列上訊息的環境定義資訊，或在發佈至特定主題時，您必須在下列其中一個類別中定義設定檔:

- MQADMIN 類別 (如果使用大寫設定檔)。
- MXADMIN 類別 (如果使用大小寫混合格式設定檔)。

環境定義安全的設定檔可以在子系統層次或佇列共用群組層次指定，並採用下列格式:

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

其中 *hlq* 可以是佇列管理程式名稱或佇列共用群組名稱，而 *queuename* 和 *topicname* 可以是您要定義其環境定義設定檔之佇列或主題的完整或通用名稱。

以佇列管理程式名稱作為字首，並將 ** 指定為佇列或主題名稱的設定檔，容許控制屬於該佇列管理程式之所有佇列及主題的環境定義安全。 在個別佇列或主題上，可以透過定義該佇列或主題上環境定義的特定設定檔來置換此設定檔。

以佇列共用群組名稱為字首，並將 ** 指定為佇列或主題名稱的設定檔，可讓您控制佇列共用群組內屬於佇列管理程式之所有佇列和主題的環境定義。 這可以在個別佇列管理程式上置換，方法是在該佇列管理程式上定義環境定義的佇列管理程式層次設定檔，並指定以佇列管理程式名稱為字首的設定檔。 您也可以指定以佇列或主題名稱為字尾的設定檔，以在個別佇列或主題上置換它。

如果您的佇列管理程式是佇列共用群組的成員，且您同時使用佇列管理程式及佇列共用群組層次安全，則 IBM MQ 會先檢查字首為佇列管理程式名稱的設定檔。 如果找不到，它會尋找以佇列共用群組名稱為字首的設定檔。

您必須允許必要的群組或使用者 ID 存取此設定檔。 下表顯示所需的存取層次，視開啟佇列時環境定義選項的規格而定。

表 *47:* 環境定義安全的存取層次

MQOPEN 或 MQPUT1 選項	**RACF hlq.CONTEXT.queuename 或 hlq.CONTEXT.topicname 所需的存取層次**
MQPMO_NO_CONTEXT	無環境定義安全檢查
MQPMO_DEFAULT_CONTEXT	無環境定義安全檢查
MQOO_SAVE_ALL_CONTEXT	無環境定義安全檢查
MQOO_PASS_IDENTITY_CONTEXT MQPMO_PASS_IDENTITY_CONTEXT	READ
MQOO_PASS_ALL_CONTEXT MQPMO_PASS_ALL_CONTEXT	READ
MQOO_SET_IDENTITY_CONTEXT MQPMO_SET_IDENTITY_CONTEXT	UPDATE
MQOO_SET_ALL_CONTEXT MQPMO_SET_ALL_CONTEXT	CONTROL
MQOO_OUTPUT 或 MQPUT1(USAGE (XMITQ))	CONTROL
MQSUB 選項	
MQSO_SET_IDENTITY_CONTEXT **(附註 2)**	UPDATE

註:

1. 用於分散式佇列的使用者 ID 需要對 hlq.CONTEXT.queuename 的 CONTROL 存取權，才能將訊息放置在目的地佇列上。 如需所使用使用者 ID 的相關資訊，請參閱 第 203 頁的『通道起始程式使用的使用者 ID』 。
2. 如果在 MQSUB 要求上指定 MQSO_CREATE 或 MQSO_ALTER 選項，您想要在 MQSD 結構中設定任何身分環境定義欄位，則需要指定 MQSO_SET_IDENTITY_CONTEXT 選項。 您也需要目的地佇列之環境定義設定檔的適當權限。

如果您將指令放置在系統指令輸入佇列上，請使用預設環境定義放置訊息選項，將正確的使用者 ID 與指令相關聯。

例如， IBM MQ 提供的公用程式 CSQUTIL 可用來卸載及重新載入佇列中的訊息。 當卸載訊息還原至佇列時， CSQUTIL 公用程式會使用 MQOO_SET_ALL_CONTEXT 選項，將訊息傳回其原始狀態。 除了此開啟選項所需的佇列安全之外，還需要環境定義權限。 例如，如果佇列管理程式 MQS1 上的群組 BACKGRP 需要此權限，則會由下列項目定義:

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

視指定的選項及執行的安全類型而定，開啟佇列時也可能會發生其他類型的安全檢查。 這些包括佇列安全 (請參閱 第 169 頁的『佇列安全的設定檔』) 及替代使用者安全 (請參閱 第 181 頁的『替代使用者安全的設定檔』)。 如需摘要表格，其中顯示當佇列、環境定義及替代使用者安全都在作用中時所需的開啟選項及安全檢查，請參閱 第 175 頁的表 36。

系統佇列環境定義安全

許多系統佇列都由 IBM MQ 的輔助組件存取，例如通道起始程式位址空間，以及 IBM MQ Console 和 REST API 所使用的 mqweb 伺服器。

執行這些作業所使用的使用者 ID 必須獲得 RACF 對這些佇列的存取權，如 第 184 頁的表 48 所示。

表 *48:* 環境定義作業需要 *SYSTEM* 佇列的存取權

系統佇列	**分散式佇列的通道起始程式**	**mqweb 伺服器**
SYSTEM.ADMIN.COMMAND.QUEUE	-	CONTROL
SYSTEM.BROKER.CONTROL.QUEUE	CONTROL	-
SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS	CONTROL	-
SYSTEM.CHANNEL.SYNCQ	CONTROL	-
SYSTEM.CLUSTER.COMMAND.QUEUE	CONTROL	-
SYSTEM.CLUSTER.TRANSMIT.QUEUE	CONTROL	-

z/OS 指令安全的設定檔

若要啟用指令的安全檢查，請將設定檔新增至 MQCMDS 類別。 設定檔名稱基於 MQSC 指令，但同時控制 MQSC 和 PCF 指令。 設定檔可以套用至佇列管理程式或佇列共用群組。

如果您想要對指令進行安全檢查 (因此您尚未定義指令安全切換設定檔 hlq.NO.CMD.CHECKS) 您必須將設定檔新增至 MQCMDS 類別。

相同的安全設定檔同時控制 MQSC 及 PCF 指令。 用於指令安全檢查的 RACF 設定檔名稱基於 MQSC 指令名稱本身。 這些設定檔採用下列格式:

```
hlq.verb.pkw
```

其中 hlq 可以是 qmgr-name (佇列管理程式名稱) 或 qsg-name (佇列共用群組名稱) ， verb 是指令名稱的動詞部分 (例如 ALTER) ， 而 pkw 是物件類型 (例如本端佇列的 QLOCAL)。

因此，子系統 CSQ1 中 ALTER QLOCAL 指令的設定檔名稱如下:

```
CSQ1.ALTER.QLOCAL
```

您可以使用通用設定檔來保護指令集，以便您可以維護較少的設定檔，從而減少存取清單。 請考慮建立通用設定檔，以套用至未受特定設定檔保護的所有指令。 使用 UACC (NONE) 定義此設定檔，並只授與 ALTER 存取權給包含管理者的 RACF 群組。 然後，您可以建立適用於所有 DISPLAY 指令的通用設定檔，並授與廣泛的存取權。 在這些極端值之間，您可以識別需要存取某些指令集的使用者群組，在此情況下，您可以建立那些指令集的設定檔，並將存取權授與代表那些使用者類別的 RACF 群組。 避免讓使用者存取他們不需要的指令: 套用最小專用權原則，讓使用者只能存取其工作所需的指令。

以佇列管理程式名稱作為字首的設定檔可控制在該佇列管理程式上使用指令。 以佇列共用群組名稱為字首的設定檔可控制在佇列共用群組內的所有佇列管理程式上使用指令。 透過在個別佇列管理程式上定義該指令的佇列管理程式層次設定檔，可以在該佇列管理程式上置換此存取權。

如果您的佇列管理程式是佇列共用群組的成員，且您同時使用佇列管理程式及佇列共用群組層次安全，則 IBM MQ 會檢查字首為佇列管理程式名稱的設定檔。 如果找不到，它會尋找以佇列共用群組名稱為字首的設定檔。

透過在佇列管理程式層次設定指令設定檔，可以限制使用者在特定佇列管理程式上發出指令。 或者，您可以針對每一個指令動詞為佇列共用群組定義一個設定檔，並針對該設定檔而非個別佇列管理程式進行所有安全檢查。

如果子系統安全及佇列共用群組安全都在作用中，且找不到本端設定檔，則會執行指令安全檢查，以查看使用者是否有權存取佇列共用群組設定檔。

如果您使用 CMDSCOPE 屬性，將指令遞送至佇列共用群組中的其他佇列管理程式，則會在執行指令的每一個佇列管理程式上檢查安全，但不一定在輸入指令的佇列管理程式上。

第 185 頁的表 49 針對每一個 IBM MQ MQSC 指令，顯示要執行指令安全檢查所需的設定檔，以及 MQCMDS 類別中每一個設定檔的對應存取層次。

第 190 頁的表 50 針對每一個 IBM MQ PCF 指令，顯示執行指令安全檢查所需的設定檔，以及 MQCMDS 類別中每一個設定檔的對應存取層次。

表 *49: MQSC* 指令、設定檔及其存取層次

指令	**MQCMDS 的指令設定檔**	**MQCMDS 的存取層次**	**MQADMIN 或 MXADMIN 的指令資源設定檔**	**MQADMIN 或 MXADMIN 的存取層次**
ALTER AUTHINFO	hlq.ALTER.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
ALTER BUFFPOOL	hlq.ALTER.BUFFPOOL	ALTER	不檢查	-
ALTER CFSTRUCT	hlq.ALTER.CFSTRUCT	ALTER	不檢查	-
ALTER CHANNEL	hlq.ALTER.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
ALTER NAMELIST	hlq.ALTER.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
ALTER PROCESS	hlq.ALTER.PROCESS	ALTER	hlq.PROCESS.process	ALTER
ALTER PSID	hlq.ALTER.PSID	ALTER	不檢查	-
ALTER QALIAS	hlq.ALTER.QALIAS	ALTER	hlq.QUEUE.queue	ALTER
ALTER QLOCAL	hlq.ALTER.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
ALTER QMGR	hlq.ALTER.QMGR	ALTER	不檢查	-
ALTER QMODEL	hlq.ALTER.QMODEL	ALTER	hlq.QUEUE.queue	ALTER
ALTER QREMOTE	hlq.ALTER.QREMOTE	ALTER	hlq.QUEUE.queue	ALTER

表 49: MQSC 指令、設定檔及其存取層次 (繼續)

指令	MQCMDS 的指令設定檔	MQCMDS 的存取層次	MQADMIN 或 MXADMIN 的指令資源設定檔	MQADMIN 或 MXADMIN 的存取層次
ALTER SECURITY	hlq.ALTER.SECURITY	ALTER	不檢查	-
ALTER SMDS	hlq.ALTER.SMDS	ALTER	不檢查	-
ALTER STGCLASS	hlq.ALTER.STGCLASS	ALTER	不檢查	-
ALTER SUB	hlq.ALTER.SUB	ALTER	不檢查	-
ALTER TOPIC	hlq.ALTER.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
ALTER TRACE	hlq.ALTER.TRACE	ALTER	不檢查	-
保存日誌	hlq.ARCHIVE.LOG	CONTROL	不檢查	-
備份 CFSTRUCT	hlq.BACKUP.CFSTRUCT	CONTROL	不檢查	-
CLEAR QLOCAL	hlq.CLEAR.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
CLEAR TOPICSTR 第 190 頁的『3』	hlq.CLEAR.TOPICSTR	ALTER	hlq.TOPIC.topic	ALTER
DEFINE AUTHINFO	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
DEFINE BUFFPOOL	hlq.DEFINE.BUFFPOOL	ALTER	不檢查	-
DEFINE CFSTRUCT	hlq.DEFINE.CFSTRUCT	ALTER	不檢查	-
定義通道	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
DEFINE LOG	hlq.DEFINE.LOG	ALTER	不檢查	-
DEFINE MAXSMSGS	hlq.DEFINE.MAXSMSGS	ALTER	不檢查	-
DEFINE NAMELIST	hlq.DEFINE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
DEFINE PROCESS	hlq.DEFINE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
DEFINE PSID	hlq.DEFINE.PSID	ALTER	不檢查	-
DEFINE QALIAS	hlq.DEFINE.QALIAS	ALTER	hlq.QUEUE.queue	ALTER
DEFINE QLOCAL	hlq.DEFINE.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
DEFINE QMODEL	hlq.DEFINE.QMODEL	ALTER	hlq.QUEUE.queue	ALTER
DEFINE QREMOTE	hlq.DEFINE.QREMOTE	ALTER	hlq.QUEUE.queue	ALTER
DEFINE STGCLASS	hlq.DEFINE.STGCLASS	ALTER	不檢查	-
DEFINE SUB	hlq.DEFINE.SUB	ALTER	不檢查	-
DEFINE TOPIC	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
DELETE AUTHINFO	hlq.DELETE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
DELETE BUFFPOOL	hlq.DELETE.BUFFPOOL	ALTER	不檢查	-
DELETE CFSTRUCT	hlq.DELETE.CFSTRUCT	ALTER	不檢查	-
刪除通道	hlq.DELETE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER

表 *49: MQSC* 指令、設定檔及其存取層次 *(*繼續*)*

指令	MQCMDS 的指令設定檔	MQCMDS 的存取層次	MQADMIN 或 MXADMIN 的指令資源設定檔	MQADMIN 或 MXADMIN 的存取層次
刪除名單	hlq.DELETE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
刪除處理程序	hlq.DELETE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
DELETE PSID	hlq.DELETE.PSID	ALTER	不檢查	-
DELETE QALIAS	hlq.DELETE.QALIAS	ALTER	hlq.QUEUE.queue	ALTER
DELETE QLOCAL	hlq.DELETE.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
DELETE QMODEL	hlq.DELETE.QMODEL	ALTER	hlq.QUEUE.queue	ALTER
DELETE QREMOTE	hlq.DELETE.QREMOTE	ALTER	hlq.QUEUE.queue	ALTER
DELETE STGCLASS	hlq.DELETE.STGCLASS	ALTER	不檢查	-
DELETE SUB	hlq.DELETE.SUB	ALTER	不檢查	-
刪除主題	hlq.DELETE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
DISPLAY ARCHIVE 第 189 頁的『1』	hlq.DISPLAY.ARCHIVE	READ	不檢查	-
DISPLAY AUTHINFO	hlq.DISPLAY.AUTHINFO	READ	不檢查	-
DISPLAY CFSTATUS	hlq.DISPLAY.CFSTATUS	READ	不檢查	-
DISPLAY CFSTRUCT	hlq.DISPLAY.CFSTRUCT	READ	不檢查	-
顯示通道	hlq.DISPLAY.CHANNEL	READ	不檢查	-
顯示 CHINIT	hlq.DISPLAY.CHINIT	READ	不檢查	-
DISPLAY CHLAUTH	hlq.DISPLAY.CHLAUTH	READ	不檢查	-
DISPLAY CHSTATUS	hlq.DISPLAY.CHSTATUS	READ	不檢查	-
DISPLAY CLUSQMGR	hlq.DISPLAY.CLUSQMGR	READ	不檢查	-
DISPLAY CMDSERV	hlq.DISPLAY.CMDSERV	READ	不檢查	-
DISPLAY CONN 第 189 頁的『1』	hlq.DISPLAY.CONN	READ	不檢查	-
顯示群組	hlq.DISPLAY.GROUP	READ	不檢查	-
DISPLAY LOG 第 189 頁的『1』	hlq.DISPLAY.LOG	READ	不檢查	-
DISPLAY MAXSMSGS	hlq.DISPLAY.MAXSMSGS	READ	不檢查	-
顯示名單	hlq.DISPLAY.NAMELIST	READ	不檢查	-
DISPLAY PROCESS	hlq.DISPLAY.PROCESS	READ	不檢查	-
DISPLAY PUBSUB	hlq.DISPLAY.PUBSUB	READ	不檢查	-
DISPLAY QALIAS	hlq.DISPLAY.QALIAS	READ	不檢查	-
DISPLAY QCLUSTER	hlq.DISPLAY.QCLUSTER	READ	不檢查	-
DISPLAY QLOCAL	hlq.DISPLAY.QLOCAL	READ	不檢查	-

表 49: MQSC 指令、設定檔及其存取層次 (繼續)

指令	MQCMDS 的指令設定檔	MQCMDS 的存取層次	MQADMIN 或 MXADMIN 的指令資源設定檔	MQADMIN 或 MXADMIN 的存取層次
DISPLAY QMGR	hlq.DISPLAY.QMGR	READ	不檢查	-
DISPLAY QMODEL	hlq.DISPLAY.QMODEL	READ	不檢查	-
DISPLAY QREMOCE	hlq.DISPLAY.QREMOTE	READ	不檢查	-
DISPLAY QSTATUS	hlq.DISPLAY.QSTATUS	READ	不檢查	-
顯示佇列	hlq.DISPLAY.QUEUE	READ	不檢查	-
DISPLAY SBSTATUS	hlq.DISPLAY.SBSTATUS	READ	不檢查	-
顯示 SMDS	hlq.DISPLAY.SMDS	READ	不檢查	-
DISPLAY SMDSCONN	hlq.DISPLAY.SMDSCONN	READ	不檢查	-
DISPLAY SUB	hlq.DISPLAY.SUB	READ	不檢查	-
DISPLAY SECURITY	hlq.DISPLAY.SECURITY	READ	不檢查	-
DISPLAY STGCLASS	hlq.DISPLAY.STGCLASS	READ	不檢查	-
DISPLAY SYSTEM 第 189 頁的『1』	hlq.DISPLAY.SYSTEM	READ	不檢查	-
顯示執行緒	hlq.DISPLAY.THREAD	READ	不檢查	-
DISPLAY TPSTATUS	hlq.DISPLAY.TPSTATUS	READ	不檢查	-
顯示主題	hlq.DISPLAY.TOPIC	READ	不檢查	-
DISPLAY TPSTATUS	hlq.DISPLAY.TPSTATUS	READ	不檢查	-
顯示追蹤	hlq.DISPLAY.TRACE	READ	不檢查	-
顯示使用情形第 189 頁的『1』	hlq.DISPLAY.USAGE	READ	不檢查	-
MOVE QLOCAL	hlq.MOVE.QLOCAL	ALTER	hlq.QUEUE.from-queue hlq.QUEUE.to-queue	ALTER
Ping 通道	hlq.PING.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
回復 BSDS	hlq.RECOVER.BSDS	CONTROL	不檢查	-
回復 CFSTRUCT	hlq.RECOVER.CFSTRUCT	CONTROL	不檢查	-
重新整理叢集	hlq.REFRESH.CLUSTER	ALTER	不檢查	-
重新整理佇列管理程式	hlq.REFRESH.QMGR	ALTER	不檢查	-
REFRESH SECURITY	hlq.REFRESH.SECURITY	ALTER	不檢查	-
RESET CFSTRUCT	hlq.RESET.CFSTRUCT	CONTROL	不檢查	-
重設通道	hlq.RESET.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
重設叢集	hlq.RESET.CLUSTER	CONTROL	不檢查	-
RESET QMGR	hlq.RESET.QMGR	CONTROL	不檢查	-
重設 QSTATS	hlq.RESET.QSTATS	CONTROL	hlq.QUEUE.queue	CONTROL

表 *49: MQSC* 指令、設定檔及其存取層次 *(*繼續*)*

指令	MQCMDS 的指令設定檔	MQCMDS 的存取層次	MQADMIN 或 MXADMIN 的指令資源設定檔	MQADMIN 或 MXADMIN 的存取層次
重設 SMDS	hlq.RESET.SMDS	CONTROL	不檢查	-
重設 Tpipe	hlq.RESET.TPIPE	CONTROL	不檢查	-
解析通道	hlq.RESOLVE.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
解析不確定	hlq.RESOLVE.INDOUBT	CONTROL	不檢查	-
回復佇列管理程式	hlq.RESUME.QMGR	CONTROL	不檢查	-
RVerify 安全	hlq.RVERIFY.SECURITY	ALTER	不檢查	-
設定保存	hlq.SET.ARCHIVE	CONTROL	不檢查	-
SET CHLAUTH	hlq.SET.CHLAUTH	CONTROL	不檢查	-
設定日誌	hlq.SET.LOG	CONTROL	不檢查	-
設定系統	hlq.SET.SYSTEM	CONTROL	不檢查	-
啟動通道	hlq.START.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
開始 CHINIT 第 190 頁的『4』	hlq.START.CHINIT	CONTROL	不檢查	-
START CMDSERV	hlq.START.CMDSERV	CONTROL	不檢查	-
啟動接聽器	hlq.START.LISTENER	CONTROL	不檢查	-
開始佇列管理程式	無第 189 頁的『2』	-	-	-
START SMDSCONN	hlq.START.SMDSCONN	CONTROL	不檢查	-
啟動追蹤	hlq.START.TRACE	CONTROL	不檢查	-
停止通道	hlq.STOP.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
停止 CHINIT	hlq.STOP.CHINIT	CONTROL	不檢查	-
STOP CMDSERV	hlq.STOP.CMDSERV	CONTROL	不檢查	-
停止接聽器	hlq.STOP.LISTENER	CONTROL	不檢查	-
停止佇列管理程式	hlq.STOP.QMGR	CONTROL	不檢查	-
STOP SMDSCONN	hlq.STOP.SMDSCONN	CONTROL	不檢查	-
停止追蹤	hlq.STOP.TRACE	CONTROL	不檢查	-
SUSPEND 佇列管理程式	hlq.SUSPEND.QMGR	CONTROL	不檢查	-

附註:

1. 這些指令可能由佇列管理程式在內部發出; 在這些情況下不會檢查任何權限。
2. IBM MQ 不會檢查發出 START QMGR 指令之使用者的權限。 不過，您可以使用 RACF 或您的替代安全機能，來控制對 START xxxxMSTR 指令的存取權，該指令是因 START QMGR 指令而發出的。 這是透過控制對 RACF 操作員指令 (OPERCMDS) 類別中 MVS.START.STC.xxxxMSTR 設定檔的存取權來完成。 如需此程序的詳細資料，請參閱 *z/OS SecureWay Security Server RACF Security Administrator 's Guide*。 如果您使用此技術，且未獲授權的使用者嘗試啟動佇列管理程式，則它會終止，原因碼為 00F30216。

3. **hlq.TOPIC.topic** 資源是指衍生自 TOPICSTR 的 Topic 物件。 如需詳細資料，請參閱 第 404 頁的『發佈/訂閱安全』
4. 在 IBM MQ for z/OS V6 之前的版本中，安全檢查適用於 MVS.START.STC.CSQ1CHIN。 在 IBM MQ for z/OS V6 以及更新版本中，資源名稱附加了額外的 JOBNAME 限定元。 這可能會在啟動通道起始程式時造成問題。

   若要解決此問題，請取代 MVS.START.STC。 *ssid* CHIN ，具有名為 MVS.START.STC 之資源的設定檔。 *ssid* CHIN **.*** 或 MVS.START.STC。 *ssid* CHIN。 *ssid* CHIN ，其中 *ssid* 是佇列管理程式的子系統 ID。 這需要 RACF UPDATE 權限。 如需詳細資料，請參閱 z/OS 產品文件 for 作業規劃、*MVS* 指令、 *RACF* 存取權及資源名稱。

   ssid MSTR 的 START 不包含 JOBNAME= 參數。 為了一致性，您可能想要將 MVS.START.STC.ssidMSTR 的設定檔更新為 MVS.START.STC.ssidMSTR. *。

表 *50: PCF* 指令、設定檔及其存取層次

指令	MQCMDS 的指令設定檔	MQCMDS 的存取層次	MQADMIN 或 MXADMIN 的指令資源設定檔	MQADMIN 或 MXADMIN 的存取層次
備份 CF 結構	hlq.BACKUP.CFSTRUCT	CONTROL	不檢查	-
變更鑑別資訊物件	hlq.ALTER.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
變更 CF 結構	hlq.ALTER.CFSTRUCT	ALTER	不檢查	-
變更通道	hlq.ALTER.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
變更名單	hlq.ALTER.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
變更處理程序	hlq.ALTER.PROCESS	ALTER	hlq.PROCESS.process	ALTER
變更佇列	hlq.ALTER.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
變更佇列管理程式	hlq.ALTER.QMGR	ALTER	不檢查	-
變更安全	hlq.ALTER.SECURITY	ALTER	不檢查	-
變更 SMDS	hlq.ALTER.SMDS	ALTER	不檢查	-
變更儲存類別	hlq.ALTER.STGCLASS	ALTER	不檢查	-
變更訂閱	hlq.ALTER.SUB	ALTER	不檢查	-
變更主題	hlq.ALTER.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
清除佇列	hlq.CLEAR.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
清除主題字串第 193 頁的『1』	hlq.CLEAR.TOPICSTR	ALTER	hlq.TOPIC.topic	ALTER
複製鑑別資訊物件	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
複製 CF 結構	hlq.DEFINE.CFSTRUCT	ALTER	不檢查	-
複製通道	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
複製名單	hlq.DEFINE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
複製處理程序	hlq.DEFINE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
複製佇列	hlq.DEFINE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
複製訂閱	hlq.DEFINE.SUB	ALTER	不檢查	-
複製儲存類別	hlq.DEFINE.STGCLASS	ALTER	不檢查	-
複製主題	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
建立鑑別資訊物件	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
建立 CF 結構	hlq.DEFINE.CFSTRUCT	ALTER	不檢查	-

表 *50: PCF* 指令、設定檔及其存取層次 *(*繼續*)*

指令	MQCMDS 的指令設定檔	MQCMDS 的存取層次	MQADMIN 或 MXADMIN 的指令資源設定檔	MQADMIN 或 MXADMIN 的存取層次
建立通道	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
建立名單	hlq.DEFINE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
建立處理程序	hlq.DEFINE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
建立佇列	hlq.DEFINE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
建立儲存類別	hlq.DEFINE.STGCLASS	ALTER	不檢查	-
建立訂閱	hlq.DEFINE.SUB	ALTER	不檢查	-
建立主題	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
刪除鑑別資訊物件	hlq.DELETE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
刪除 CF 結構	hlq.DELETE.CFSTRUCT	ALTER	不檢查	-
刪除通道	hlq.DELETE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
刪除名單	hlq.DELETE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
刪除處理程序	hlq.DELETE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
刪除佇列	hlq.DELETE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
刪除儲存類別	hlq.DELETE.STGCLASS	ALTER	不檢查	-
刪除訂閱	hlq.DELETE.SUB	ALTER	不檢查	-
刪除主題	hlq.DELETE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
查詢保存	hlq.DISPLAY.ARCHIVE	READ	不檢查	-
查詢鑑別資訊物件	hlq.DISPLAY.AUTHINFO	READ	不檢查	-
查詢鑑別資訊物件名稱	hlq.DISPLAY.AUTHINFO	READ	不檢查	-
查詢 CF 結構	hlq.DISPLAY.CFSTRUCT	READ	不檢查	-
查詢 CF 結構名稱	hlq.DISPLAY.CFSTRUCT	READ	不檢查	-
查詢 CF 結構狀態	hlq.DISPLAY.CFSTATUS	READ	不檢查	-
查詢通道	hlq.DISPLAY.CHANNEL	READ	不檢查	-
查詢通道鑑別記錄	hlq.DISPLAY.CHLAUTH	READ	不檢查	-
查詢通道起始程式	hlq.DISPLAY.CHINIT	READ	不檢查	-
查詢通道名稱	hlq.DISPLAY.CHANNEL	READ	不檢查	-
查詢通道狀態	hlq.DISPLAY.CHSTATUS	READ	不檢查	-
查詢叢集佇列管理程式	hlq.DISPLAY.CLUSQMGR	READ	不檢查	-
查詢連線	hlq.DISPLAY.CONNPCF	READ	不檢查	-
查詢群組	hlq.DISPLAY.GROUP	READ	不檢查	-
查詢日誌	hlq.DISPLAY.LOG	READ	不檢查	-
查詢名單	hlq.DISPLAY.NAMELIST	READ	不檢查	-
查詢名單名稱	hlq.DISPLAY.NAMELIST	READ	不檢查	-
查詢處理程序	hlq.DISPLAY.PROCESS	READ	不檢查	-
查詢處理程序名稱	hlq.DISPLAY.PROCESS	READ	不檢查	-

表 *50: PCF* 指令、設定檔及其存取層次 *(*繼續*)*

指令	MQCMDS 的指令設定檔	MQCMDS 的存取層次	MQADMIN 或 MXADMIN 的指令資源設定檔	MQADMIN 或 MXADMIN 的存取層次
查詢發佈/訂閱狀態	hlq.DISPLAY.PUBSUB	READ	不檢查	-
查詢佇列	hlq.DISPLAY.QUEUE	READ	不檢查	-
查詢佇列管理程式	hlq.DISPLAY.QMGR	READ	不檢查	-
查詢佇列名稱	hlq.DISPLAY.QUEUE	READ	不檢查	-
查詢佇列狀態	hlq.DISPLAY.QSTATUS	READ	不檢查	-
查詢安全	hlq.DISPLAY.SECURITY	READ	不檢查	-
查詢 SMDS	hlq.DISPLAY.SMDS	READ	不檢查	-
查詢 SMDSCONN	hlq.DISPLAY.SMDSCONN	READ	不檢查	-
查詢儲存類別	hlq.DISPLAY.STGCLASS	READ	不檢查	-
查詢儲存類別名稱	hlq.DISPLAY.STGCLASS	READ	不檢查	-
查詢訂閱	hlq.INQUIRE.SUB	READ	不檢查	-
查詢訂閱狀態	hlq.INQUIRE.SBSTATUS	READ	不檢查	-
查詢系統	hlq.DISPLAY.SYSTEM	READ	不檢查	-
查詢主題	hlq.DISPLAY.TOPIC	READ	不檢查	-
查詢主題名稱	hlq.DISPLAY.TOPIC	READ	不檢查	-
查詢主題狀態	hlq.DISPLAY.TPSTATUS	READ	不檢查	-
查詢使用情形	hlq.DISPLAY.USAGE	READ	不檢查	-
移動佇列	hlq.MOVE.QLOCAL	ALTER	hlq.QUEUE.from-queue hlq.QUEUE.to-queue	ALTER
Ping 通道	hlq.PING.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
回復 CF 結構	hlq.RECOVER.CFSTRUCT	CONTROL	不檢查	-
重新整理叢集	hlq.REFRESH.CLUSTER	ALTER	不檢查	-
重新整理佇列管理程式	hlq.REFRESH.QMGR	ALTER	不檢查	-
重新整理安全	hlq.REFRESH.SECURITY	ALTER	不檢查	-
重設 CF 結構	hlq.RESET.CFSTRUCT	CONTROL	不檢查	-
重設通道	hlq.RESET.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
重設叢集	hlq.RESET.CLUSTER	CONTROL	不檢查	-
重設佇列管理程式	hlq.RESET.QMGR	CONTROL	不檢查	-
重設佇列統計資料	hlq.RESET.QSTATS	CONTROL	hlq.QUEUE.queue	CONTROL
重設 SMDS	hlq.RESET.SMDS	CONTROL	不檢查	-
解析通道	hlq.RESOLVE.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
回復佇列管理程式	hlq.RESUME.QMGR	CONTROL	不檢查	-
回復佇列管理程式叢集	hlq.RESUME.QMGR	CONTROL	不檢查	-
重新驗證安全	hlq.RVERIFY.SECURITY	ALTER	不檢查	-
設定保存	hlq.SET.ARCHIVE	CONTROL	不檢查	-

表 50: PCF 指令、設定檔及其存取層次 (繼續)				
指令	**MQCMDS 的指令設定檔**	**MQCMDS 的存取層次**	**MQADMIN 或 MXADMIN 的指令資源設定檔**	**MQADMIN 或 MXADMIN 的存取層次**
設定通道鑑別記錄	hlq.SET.CHLAUTH	CONTROL	不檢查	-
設定日誌	hlq.SET.LOG	CONTROL	不檢查	-
設定系統	hlq.SET.SYSTEM	CONTROL	不檢查	-
啟動通道	hlq.START.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
啟動通道起始程式	hlq.START.CHINIT	CONTROL	不檢查	-
啟動通道接聽器	hlq.START.LISTENER	CONTROL	不檢查	-
啟動 SMDS 連線	hlq.START.SMDSCONN	CONTROL	不檢查	-
停止通道	hlq.STOP.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
停止通道起始程式	hlq.STOP.CHINIT	CONTROL	不檢查	-
停止通道接聽器	hlq.STOP.LISTENER	CONTROL	不檢查	-
停止 SMDS 連線	hlq.STOP.SMDSCONN	CONTROL	不檢查	-
暫停佇列管理程式	hlq.SUSPEND.QMGR	CONTROL	不檢查	-
暫停佇列管理程式叢集	hlq.SUSPEND.QMGR	CONTROL	不檢查	-

附註:

1. **hlq.TOPIC.topic** 資源是指衍生自 TOPICSTR 的 Topic 物件。 如需詳細資料，請參閱 第 404 頁的『發佈/訂閱安全』

如需使用 IBM MQ Console 時所需 IBM MQ PCF 設定檔的詳細資料，請參閱 第 193 頁的『IBM MQ Console -必要的指令安全設定檔』 。

z/OS *IBM MQ Console -必要的指令安全設定檔*

在 IBM MQ Console 中由 MQWebAdmin 或 MQWebAdminRO 角色中的使用者所執行的作業，會在 mqweb 伺服器啟動作業使用者 ID 的安全環境定義下進行。 如果您想要使用 IBM MQ Console，則 mqweb 伺服器已啟動作業使用者 ID 需要授權才能發出特定 PCF 指令。

第 193 頁的表 51 針對每一個 IBM MQ PCF 指令，顯示所需的指令安全設定檔，以及 IBM MQ Console 所需 MQCMDS 類別中每一個設定檔的對應存取層次。

表 51: IBM MQ Console PCF 指令、設定檔及其存取層次				
指令	**MQCMDS 的指令設定檔**	**MQCMDS 的存取層次**	**MQADMIN 或 MXADMIN 的指令資源設定檔**	**MQADMIN 或 MXADMIN 的存取層次**
變更鑑別資訊物件	hlq.ALTER.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
變更通道	hlq.ALTER.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
變更佇列	hlq.ALTER.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
變更佇列管理程式	hlq.ALTER.QMGR	ALTER	不檢查	-
變更主題	hlq.ALTER.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
清除佇列	hlq.CLEAR.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
建立鑑別資訊物件	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
建立通道	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER

表 51: IBM MQ Console PCF 指令、設定檔及其存取層次 (繼續)

指令	MQCMDS 的指令設定檔	MQCMDS 的存取層次	MQADMIN 或 MXADMIN 的指令資源設定檔	MQADMIN 或 MXADMIN 的存取層次
建立佇列	hlq.DEFINE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
建立訂閱	hlq.DEFINE.SUB	ALTER	不檢查	-
建立主題	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
刪除鑑別資訊物件	hlq.DELETE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
刪除通道	hlq.DELETE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
刪除佇列	hlq.DELETE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
刪除訂閱	hlq.DELETE.SUB	ALTER	不檢查	-
刪除主題	hlq.DELETE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
查詢鑑別資訊物件	hlq.DISPLAY.AUTHINFO	READ	不檢查	-
查詢鑑別資訊物件名稱	hlq.DISPLAY.AUTHINFO	READ	不檢查	-
查詢通道	hlq.DISPLAY.CHANNEL	READ	不檢查	-
查詢通道鑑別記錄	hlq.DISPLAY.CHLAUTH	READ	不檢查	-
查詢通道起始程式	hlq.DISPLAY.CHINIT	READ	不檢查	-
查詢通道名稱	hlq.DISPLAY.CHANNEL	READ	不檢查	-
查詢通道狀態	hlq.DISPLAY.CHSTATUS	READ	不檢查	-
查詢佇列	hlq.DISPLAY.QUEUE	READ	不檢查	-
查詢佇列管理程式	hlq.DISPLAY.QMGR	READ	不檢查	-
查詢佇列名稱	hlq.DISPLAY.QUEUE	READ	不檢查	-
查詢佇列狀態	hlq.DISPLAY.QSTATUS	READ	不檢查	-
查詢訂閱	hlq.INQUIRE.SUB	READ	不檢查	-
查詢訂閱狀態	hlq.INQUIRE.SBSTATUS	READ	不檢查	-
查詢主題	hlq.DISPLAY.TOPIC	READ	不檢查	-
查詢主題名稱	hlq.DISPLAY.TOPIC	READ	不檢查	-
查詢主題狀態	hlq.DISPLAY.TPSTATUS	READ	不檢查	-
Ping 通道	hlq.PING.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
重新整理叢集	hlq.REFRESH.CLUSTER	ALTER	不檢查	-
重新整理安全	hlq.REFRESH.SECURITY	ALTER	不檢查	-
重設通道	hlq.RESET.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
解析通道	hlq.RESOLVE.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
設定通道鑑別記錄	hlq.SET.CHLAUTH	CONTROL	不檢查	-
啟動通道	hlq.START.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
停止通道	hlq.STOP.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL

z/OS 指令資源安全的設定檔

如果您尚未定義指令資源安全切換設定檔，因為您想要對與指令相關聯的資源進行安全檢查，您必須將每一個資源的資源設定檔新增至適當的類別。 相同的安全設定檔同時控制 MQSC 及 PCF 指令。

如果您尚未定義指令資源安全切換設定檔 hlq.NO.CMD.RESC.CHECKS，因為您想要對與指令相關聯的資源進行安全檢查，您必須:

- 針對每一個資源，在 **MQADMIN** 類別中新增資源設定檔 (如果使用大寫設定檔)。
- 針對每一個資源，在 **MXADMIN** 類別中新增資源設定檔 (如果使用大小寫混合格式的設定檔)。

相同的安全設定檔同時控制 MQSC 及 PCF 指令。

用於指令資源安全檢查的設定檔格式如下:

```
hlq.type.resourcename
```

其中 hlq 可以是 qmgr-name (佇列管理程式名稱) 或 qsg-name (佇列共用群組名稱)。

以佇列管理程式名稱為字首的設定檔可控制對該佇列管理程式上與指令相關聯之資源的存取權。 以佇列共用群組名稱為字首的設定檔可控制存取與佇列共用群組內所有佇列管理程式上的指令相關聯的資源。 透過在個別佇列管理程式上定義該指令資源的佇列管理程式層次設定檔，可以在該佇列管理程式上置換此存取權。

如果您的佇列管理程式是佇列共用群組的成員，且您同時使用佇列管理程式及佇列共用群組層次安全，則 IBM MQ 會先檢查字首為佇列管理程式名稱的設定檔。 如果找不到，它會尋找以佇列共用群組名稱為字首的設定檔。

例如，針對子系統 CSQ1 中的模型佇列 CREDIT.WORTHY 進行指令資源安全檢查的 RACF 設定檔名稱為:

```
CSQ1.QUEUE.CREDIT.WORTHY
```

因為所有指令資源類型的設定檔都保留在 MQADMIN 類別中，所以在設定檔中需要設定檔名稱的 "type" 部分，以區分具有相同名稱之不同類型的資源。 設定檔名稱的 "type" 部分可以是 CHANNEL、QUEUE、TOPIC、PROCESS 或 NAMELIST。 例如，使用者可能已獲授權定義 hlq.QUEUE.PAYROLL.ONE，但未獲授權定義 hlq.PROCESS.PAYROLL.ONE

如果資源類型是佇列，且設定檔是佇列共用群組層次設定檔，則它會控制對佇列共用群組內一個以上本端佇列的存取權，或從佇列共用群組中的任何佇列管理程式存取單一共用佇列。

z/OS MQSC 指令、設定檔及其存取層次 會針對每一個 IBM MQ MQSC 指令，顯示要執行指令安全檢查所需的設定檔，以及 MQCMDS 類別中每一個設定檔的對應存取層次。

z/OS PCF 指令、設定檔及其存取層次 會針對每一個 IBM MQ PCF 指令，顯示執行指令安全檢查所需的設定檔，以及 MQCMDS 類別中每一個設定檔的對應存取層次。

z/OS 別名佇列及遠端佇列的指令資源安全檢查
別名佇列和遠端佇列都提供間接給另一個佇列。 當您考量這些佇列的安全檢查時，會套用其他點。

別名佇列

當您定義別名佇列時，只會針對別名佇列的名稱執行指令資源安全檢查，而不是針對別名所解析的目標佇列名稱。

別名佇列可以解析為本端及遠端佇列。 如果您不想允許使用者存取特定本端或遠端佇列，則必須執行下列兩項:

1. 不容許使用者存取這些本端及遠端佇列。
2. 限制使用者無法定義這些佇列的別名。 亦即，防止它們能夠發出 DEFINE QALIAS 及 ALTER QALIAS 指令。

遠端佇列

當您定義遠端佇列時，只會對遠端佇列名稱執行指令資源安全檢查。 不會對遠端佇列物件定義中 RNAME 或 XMITQ 屬性中指定的佇列名稱執行任何檢查。

z/OS RESLEVEL 安全設定檔

您可以在 MQADMIN 或 MXADMIN 類別中定義特殊設定檔，以控制針對 API 資源安全所檢查的使用者 ID 數目。 此設定檔稱為 RESLEVEL 設定檔。 此設定檔如何影響 API-資源安全取決於您存取 IBM MQ 的方式。

當應用程式嘗試連接至 IBM MQ 時， IBM MQ 會檢查與連線相關聯的使用者 ID 對 MQADMIN 或 MXADMIN 類別中稱為之設定檔的存取權:

```
hlq.RESLEVEL
```

其中 hlq 可以是 ssid (子系統 ID) 或 qsg (佇列共用群組 ID)。

與每一個連線類型相關聯的使用者 ID 如下:

- 批次連線之連接作業的使用者 ID
- CICS 連線的 CICS 位址空間使用者 ID
- IMS 連線的 IMS 區域位址空間使用者 ID
- 通道起始程式連線的通道起始程式位址空間使用者 ID

小心: RESLEVEL 是一個非常強大的選項; 它可能會導致略過特定連線的所有資源安全檢查。

如果您未定義 RESLEVEL 設定檔，則必須小心 MQADMIN 類別中沒有其他設定檔符合 hlq.RESLEVEL。 例如，如果您在 MQADMIN 中有一個稱為 hlq. * * 的設定檔 且沒有 hlq.RESLEVEL 設定檔，請注意 hlq. * * 因為它用於 RESLEVEL 檢查。

定義 hlq.RESLEVEL 設定檔，並將 UACC 設為 NONE ， 而不是完全沒有 RESLEVEL 設定檔。 在存取清單中擁有儘可能少的使用者或群組。 如需如何審核 RESLEVEL 存取權的詳細資料，請參閱 第 217 頁的『z/OS 上的審核考量』。

如果您只使用佇列管理程式層次安全， IBM MQ 會針對 qmgr-name.RESLEVEL 設定檔執行 RESLEVEL 檢查。 如果您只使用佇列共用群組層次安全， IBM MQ 會對 qsg-name.RESLEVEL 設定檔執行 RESLEVEL 檢查。 如果您同時使用佇列管理程式及佇列共用群組層次安全的組合， IBM MQ 會先檢查佇列管理程式層次的 RESLEVEL 設定檔是否存在。 如果找不到，它會在佇列共用群組層次檢查 RESLEVEL 設定檔。

如果找不到 RESLEVEL 設定檔， IBM MQ 會啟用檢查 CICS 或 IMS 連線的工作及作業 (或替代使用者) ID。 對於批次連線， IBM MQ 會啟用工作 (或替代) 使用者 ID 的檢查。 對於通道起始程式， IBM MQ 會啟用通道使用者 ID 及 MCA (或替代) 使用者 ID 的檢查。

如果有 RESLEVEL 設定檔，則檢查層次取決於環境及設定檔的存取層次。

請記住，如果您的佇列管理程式是佇列共用群組的成員，且您未在佇列管理程式層次定義此設定檔，則可能會在佇列共用群組層次定義一個會影響檢查層次的設定檔。 若要啟動兩個使用者 ID 的檢查，您可以使用 UACC (NONE) 來定義 RESLEVEL 設定檔 (字首為佇列共用群組名稱的佇列管理程式名稱) ， 並確保相關使用者沒有對此設定檔授與的存取權。

當您考量通道起始程式使用者 ID 對 RESLEVEL 的存取權時，請記住通道起始程式所建立的連線也是通道所使用的連線。 此設定會導致略過通道起始程式使用者 ID 的所有資源安全檢查，有效地略過所有通道的安全檢查。 如果通道起始程式對 RESLEVEL 的使用者 ID 存取權不是 NONE ， 則只會檢查一個使用者 ID (若為 READ 或 UPDATE 存取層次) ， 或不檢查任何使用者 ID (若為 CONTROL 或 ALTER 存取層次)。 如果您授與通道起始程式的使用者 ID NONE 以外的存取層次給 RESLEVEL ， 請確定您瞭解此設定對通道所執行安全檢查的影響。

使用 RESLEVEL 設定檔表示不會取得一般安全審核記錄。 例如，如果您將 UAUDIT 放置在使用者身上，則不會審核 MQADMIN 中 hlq.RESLEVEL 設定檔的存取權。

如果您在 hlq.RESLEVEL 設定檔上使用 RACF WARNING 選項，則不會針對 RESLEVEL 類別中的設定檔產生任何 RACF 警告訊息。

報告訊息 (例如 COD) 的安全檢查由與原始應用程式相關聯的 RESLEVEL 設定檔控制。 例如，如果批次工作的使用者 ID 對 RESLEVEL 設定檔具有 CONTROL 或 ALTER 權限，則會略過批次工作所執行的所有資源檢查，包括報告訊息的安全檢查。

如果您變更 RESLEVEL 設定檔，則在進行變更之前，使用者必須切斷並重新連接。 (這包括如果分散式佇列位址空間使用者 ID 對 RESLEVEL 設定檔的存取權已變更，則停止並重新啟動通道起始程式。)

若要關閉 RESLEVEL 審核，請使用 RESAUDIT 系統參數。

z/OS *RESLEVEL* 及批次連線

依預設，當透過批次和批次類型連線來存取 IBM MQ 資源時，使用者必須獲得授權來存取特定作業的該資源。 您可以設定適當的 RESLEVEL 定義，以略過安全檢查。

是否檢查使用者取決於連接時使用的使用者 ID，以及用於連線檢查的相同使用者 ID。

例如，您可以設定 RESLEVEL，以便當您信任的使用者透過批次連線存取特定資源時，不會執行任何 API 資源安全檢查; 但當您不信任的使用者嘗試存取相同的資源時，安全檢查會正常執行。 只有在充分信任使用者及該使用者所執行的程式時，您才應該設定 RESLEVEL 檢查來略過 API 資源安全檢查。

下表顯示對批次連線進行的檢查。

表 *52:* 在不同的 *RACF* 存取層次進行批次連線的檢查

RACF 存取層次	**檢查層次**
無	已執行資源檢查
READ	已執行資源檢查
UPDATE	已執行資源檢查
CONTROL	沒有支票
ALTER	沒有支票

z/OS *RESLEVEL* 和系統功能

RESLEVEL 套用至作業及控制台，以及套用至 CSQUTIL。

作業及控制面板及 CSQUTIL 公用程式是批次類型應用程式，可對佇列管理程式的指令伺服器提出要求，因此它們會遵循 第 197 頁的『RESLEVEL 及批次連線』中說明的考量。 您可以使用 RESLEVEL 來略過 SYSTEM.COMMAND.INPUT 和 SYSTEM.COMMAND.REPLY.MODEL 佇列，但不適用於動態佇列 SYSTEM.CSQXCMD. * ， SYSTEM.CSQOREXX.*, 及 SYSTEM.CSQUTIL. *。

指令伺服器是佇列管理程式的一部分，因此沒有連線或 RESLEVEL 檢查與它相關聯。 因此，為了維護安全，指令伺服器必須確認發出要求之應用程式的使用者 ID 有權開啟用於回覆的佇列。 對於作業及控制面板，這是 SYSTEM.CSQOREXX. *。 對於 CSQUTIL，它是 SYSTEM.CSQUTIL. *。 除了所提供的 RESLEVEL 授權之外，使用者還必須獲得授權來使用這些佇列 (如 第 174 頁的『系統佇列安全』中所述)。

對於其他使用指令伺服器的應用程式，它是它們命名為回覆目的地佇列的佇列。 這類其他應用程式可能會藉由傳遞 (在訊息環境定義中) 比其本身更信任的使用者 ID 給指令伺服器，來欺騙指令伺服器將訊息放置在未獲授權的佇列上。 若要防止此情況，請使用 CONTEXT 設定檔來保護放置在 SYSTEM.COMMAND.INPUT。

z/OS *RESLEVEL* 及 *CICS* 連線

依預設，在 CICS 連線上進行 API 資源安全檢查時，會檢查兩個使用者 ID。 您可以設定 RESLEVEL 設定檔來變更要檢查哪些使用者 ID。

第一個檢查的使用者 ID 是 CICS 位址空間的使用者 ID。 這是 CICS 工作的工作卡上的使用者 ID，或 z/OS STARTED 類別或啟動程序表格指派給 CICS 啟動作業的使用者 ID。 (它不是 CICS DFLTUSER。)

第二個檢查的使用者 ID 是與 CICS 交易相關聯的使用者 ID。

如果其中一個使用者 ID 沒有資源的存取權，則要求會失敗，完成碼為 MQRC_NOT_AUTHORIZED。 CICS 位址空間使用者 ID 及執行 CICS 交易之人員的使用者 ID 都必須具有正確層次資源的存取權。

RESLEVEL 如何影響所執行的檢查

視您設定 RESLEVEL 設定檔的方式而定，您可以在要求存取資源時變更要檢查哪些使用者 ID。 如需相關資訊，請參閱 第 198 頁的表 53。

所檢查的使用者 ID 取決於連線時使用的使用者 ID，即 CICS 位址空間使用者 ID。 此控制項可讓您針對來自某個系統 (例如，測試系統、TESTCICS) 的 IBM MQ 要求略過 API 資源安全檢查，但針對另一個系統 (例如，正式作業系統、PRODCICS) 實作它們。

註: 如果您以 STARTED 類別或 RACF 啟動程序表格 ICHRIN03 中的 "trusted" 屬性來設定 CICS 位址空間使用者 ID，這會置換任何使用者 ID，以檢查佇列管理程式 RESLEVEL 設定檔所建立的 CICS 位址空間 (亦即，佇列管理程式不會對 CICS 位址空間執行安全檢查)。 如需相關資訊，請參閱 *CICS Transaction Server for z/OS V3.2 RACF* 安全手冊。

下表顯示對 CICS 連線進行的檢查。

表 *53:* 在 *CICS* 連線的不同 *RACF* 存取層次進行的檢查

RACF 存取層次	**檢查層次**
無	IBM MQ 會檢查 CICS 位址空間使用者 ID 及交易使用者 ID。
READ	IBM MQ 只會檢查 CICS 位址空間使用者 ID。
UPDATE	如果交易定義為具有 RESSEC (YES) 的 CICS， IBM MQ 會檢查 CICS 位址空間使用者 ID 及交易使用者 ID。
UPDATE	如果將交易定義給具有 RESSEC (NO) 的 CICS，則 IBM MQ 只會檢查 CICS 位址空間使用者 ID。
CONTROL 或 ALTER	IBM MQ 不會檢查任何使用者 ID。

z/OS *RESLEVEL 及 IMS 連線*

依預設，對 IMS 連線進行 API 資源安全檢查時，會檢查兩個使用者 ID。 您可以設定 RESLEVEL 設定檔來變更要檢查哪些使用者 ID。

依預設，當對 IMS 連線進行 API 資源安全檢查時，會檢查兩個使用者 ID，以查看是否容許存取資源。

第一個檢查的使用者 ID 是 IMS 區域的位址空間。 這取自工作卡中的 USER 欄位，或從 z/OS STARTED 類別或啟動程序表格 (SPT) 指派給區域的使用者 ID。

第二個檢查的使用者 ID 與在相依區域中執行的工作相關聯。 它是根據相依區域的類型來決定，如 如何判定 IMS(tm) 連線的第二個使用者 ID 中所示。

如果第一個或第二個 IMS 使用者 ID 無法存取資源，則要求會失敗，完成碼為 MQRC_NOT_AUTHORIZED。

IBM MQ RESLEVEL 設定檔的設定無法變更從 IBM 提供的 MQ-IMS 觸發監視器程式 CSQQTRMN 排定 IMS 交易的使用者 ID。 此使用者 ID 是該觸發監視器的 PSBNAME，依預設為 CSQQTRMN。

RESLEVEL 如何影響所執行的檢查

視您設定 RESLEVEL 設定檔的方式而定，您可以在要求存取資源時變更要檢查哪些使用者 ID。 可能的檢查如下:

- 請檢查 IMS 區域位址空間使用者 ID 及第二個使用者 ID 或替代使用者 ID。
- 僅檢查 IMS 區域位址空間使用者 ID。
- 不檢查任何使用者 ID。

下表顯示對 IMS 連線進行的檢查。

表 *54:* 在 *IMS* 連線的不同 *RACF* 存取層次進行的檢查

RACF 存取層次	**檢查層次**
無	請檢查 IMS 位址空間使用者 ID 及 IMS 第二個使用者 ID 或替代使用者 ID。
READ	請檢查 IMS 位址空間使用者 ID。
UPDATE	請檢查 IMS 位址空間使用者 ID。

表 *54:* 在 *IMS* 連線的不同 *RACF* 存取層次進行的檢查 *(*繼續*)*	
RACF 存取層次	**檢查層次**
CONTROL	沒有支票
ALTER	沒有支票

z/OS *RESLEVEL* 及通道起始程式連線

依預設，當通道起始程式進行 API 資源安全檢查時，會檢查兩個使用者 ID。 您可以設定 RESLEVEL 設定檔來變更要檢查哪些使用者 ID。

依預設，當通道起始程式進行 API 資源安全檢查時，會檢查兩個使用者 ID，以查看是否容許存取資源。

檢查的使用者 ID 可以是 MCAUSER 通道屬性指定的使用者 ID、從網路接收的使用者 ID、通道起始程式位址空間的使用者 ID，或訊息描述子的替代使用者 ID。 要檢查哪些使用者 ID，視您使用的通訊協定及 PUTAUT 通道屬性的設定而定。 如需相關資訊，請參閱 第 203 頁的『通道起始程式使用的使用者 ID』。

如果其中一個使用者 ID 沒有資源的存取權，則要求會失敗，完成碼為 MQRC_NOT_AUTHORIZED。

RESLEVEL 如何影響所執行的檢查

視您設定 RESLEVEL 設定檔的方式而定，您可以變更在要求存取資源時要檢查哪些使用者 ID，以及要檢查多少使用者 ID。

下表顯示對通道起始程式的連線所進行的檢查，以及自使用此連線以來對所有通道所進行的檢查。

表 *55:* 在不同 *RACF* 存取層次對通道起始程式連線進行的檢查	
RACF 存取層次	**檢查層次**
無	請檢查兩個使用者 ID。
READ	請檢查一個使用者 ID。
UPDATE	請檢查一個使用者 ID。
CONTROL	沒有支票
ALTER	沒有支票
註: 如需所檢查使用者 ID 的定義，請參閱 第 203 頁的『通道起始程式使用的使用者 ID』	

z/OS *RESLEVEL* 及內部群組佇列作業

依預設，當內部群組佇列作業代理程式進行 API 資源安全檢查時，會檢查兩個使用者 ID，以查看是否容許存取資源。 您可以設定 RESLEVEL 設定檔來變更要檢查哪些使用者 ID。

檢查的使用者 ID 可以是接收端佇列管理程式的 IGQUSER 屬性所決定的使用者 ID，這是將訊息放入 SYSTEM.QSG.TRANSMIT.QUEUE，或在訊息之訊息描述子的 *UserIdentifier* 欄位中指定的替代使用者 ID。 如需相關資訊，請參閱 第 207 頁的『內部群組佇列作業代理程式使用的使用者 ID』。

因為內部群組佇列作業代理程式是內部佇列管理程式作業，所以它不會發出明確連接要求，並在佇列管理程式的使用者 ID 下執行。 在佇列管理程式起始設定時，會啟動內部群組佇列作業代理程式。 在起始設定群組內佇列作業代理程式期間， IBM MQ 會檢查與佇列管理程式相關聯的使用者 ID 對 MQADMIN 類別中的設定檔所具備的存取權，該設定檔稱為:

```
hlq.RESLEVEL
```

除非已設定 hlq.NO.SUBSYS.SECURITY 參數，否則一律會執行這項檢查。

如果沒有 RESLEVEL 設定檔， IBM MQ 會啟用兩個使用者 ID 的檢查。 如果有 RESLEVEL 設定檔，則檢查層次取決於授與設定檔之佇列管理程式使用者 ID 的存取層次。 在不同 RACF® 存取層次對內部群組佇列作業代理程式進行的檢查 會顯示對內部群組佇列作業代理程式進行的檢查。

表 56: 在不同 *RACF* 存取層次對內部群組佇列作業代理程式進行的檢查	
RACF 存取層次	**檢查層次**
無	請檢查兩個使用者 ID。
READ	請檢查一個使用者 ID。
UPDATE	請檢查一個使用者 ID。
CONTROL	沒有支票
ALTER	沒有支票
註: 如需所檢查使用者 ID 的定義，請參閱 第 207 頁的『內部群組佇列作業代理程式使用的使用者 ID』	

如果已變更授與佇列管理程式使用者 ID 的 RESLEVEL 設定檔的許可權，則必須停止並重新啟動內部群組佇列作業代理程式，以取得新的許可權。 因為無法獨立停止並重新啟動群組內佇列作業代理程式，所以必須停止並重新啟動佇列管理程式才能達到此目的。

z/OS *RESLEVEL* 和使用者 *ID* 已勾選

設定 RESLEVEL 設定檔並授與其存取權的範例。

針對批次連線的設定檔名稱檢查使用者 ID 至 針對 LU 6.2 及 TCP/IP 伺服器連線通道的設定檔名稱檢查使用者 ID 顯示 RESLEVEL 如何影響針對不同 MQI 要求來檢查哪些使用者 ID。

例如，您具有稱為 QM66 且具有下列需求的佇列管理程式:

- 使用者 WS21B 將免於資源安全。
- 在位址空間使用者 ID CICSWXN 下執行的 CICS 啟動作業 WXNCICS ，只會針對以 RESSEC (YES) 定義的交易執行完整資源檢查。

若要定義適當的 RESLEVEL 設定檔，請發出下列 RACF 指令:

```
RDEFINE MQADMIN QM66.RESLEVEL UACC(NONE)
```

然後使用下列指令，授與使用者對此設定檔的存取權:

```
PERMIT QM66.RESLEVEL CLASS(MQADMIN) ID(WS21B) ACCESS(CONTROL)
PERMIT QM66.RESLEVEL CLASS(MQADMIN) ID(CICSWXN) ACCESS(UPDATE)
```

如果您在使用者 ID 連接至佇列管理程式 QM66 時進行這些變更，則在進行變更之前，使用者必須中斷連線並重新連接。

當使用者連接時，如果子系統安全不在作用中，但當此使用者仍連接時，子系統安全會變成作用中，則會對使用者套用完整資源安全檢查。 使用者必須重新連接，才能取得正確的 RESLEVEL 處理程序。

z/OS z/OS 上用於安全檢查的使用者 ID

IBM MQ 會根據與使用者、終端機、應用程式及其他資源相關聯的使用者 ID 來起始安全檢查。 這個主題集合列出每一種安全檢查類型所使用的使用者 ID。

z/OS 連線安全的使用者 *ID*

用於連線安全的使用者 ID 視連線類型而定。

連線類型	**使用者 ID 內容**
批次連線	連接作業的使用者 ID。 例如: • TSO 使用者 ID • 由 USER JCL 參數指派給批次工作的使用者 ID • STARTED 類別或已啟動程序表格指派給已啟動作業的使用者 ID

連線類型	使用者 ID 內容
CICS connection	CICS 位址空間使用者 ID。
IMS connection	IMS 區域位址空間使用者 ID。
通道起始程式連線	通道起始程式位址空間使用者 ID。

z/OS 指令及指令資源安全的使用者 *ID*

用於指令安全或指令資源安全的使用者 ID 取決於從中發出指令的位置。

發行自 ...	使用者 ID 內容
CSQINP1、 CSQINP2 或 CSQINPT	未進行任何檢查。
系統指令輸入佇列	在包含指令之訊息的訊息描述子的 *UserIdentifier* 中找到的使用者 ID。 如果訊息不包含 *UserIdentifier*，則會將空白的使用者 ID 傳遞給安全管理程式。
主控台	登入主控台的使用者 ID。 如果主控台未登入，則為 CSQ6SYSP 中 CMDUSER 系統參數所設定的預設使用者 ID。 如果要從主控台發出指令，主控台必須具有 z/OS SYS AUTHORITY 屬性。
SDSF/TSO 主控台	TSO 或工作使用者 ID。
作業及控制面板	TSO 使用者 ID。 如果您要使用作業及控制台，則必須具有適當的權限，才能發出對應於您所選擇動作的指令。 此外，您必須具備所有 hlq.DISPLAY 的 READ 存取權。 MQCMDS 類別中的 物件 設定檔，因為畫面會使用各種 DISPLAY 指令來收集它們所呈現的資訊。
MGCRE	如果 MGCRE 與 UTOKEN 一起使用，則為 UTOKEN 中的使用者 ID。 如果在沒有 UTOKEN 的情況下發出 MGCRE ，則會使用 TSO 或工作使用者 ID。
CSQ0UTIL	工作使用者 ID。
CSQUTIL	工作使用者 ID。
CSQINPX	通道起始程式位址空間的使用者 ID。

z/OS 資源安全的使用者 *ID (MQOPEN*、*MQSUB* 和 *MQPUT1)*

此資訊顯示每一種連線類型的一般及替代使用者 ID 的使用者 ID 內容。 RESLEVEL 設定檔定義檢查數目。 所檢查的使用者 ID 是用於 **MQOPEN**、 **MQSUB** 或 **MQPUT1** 呼叫。

註: 所有使用者 ID 欄位都會在收到時完全檢查。 不會進行任何轉換，例如，包含 "Bob"、 "BOB"及 "bob" 的三個使用者 ID 欄位並不相等。

z/OS 檢查批次連線的使用者 *ID*

檢查批次連線的使用者 ID 取決於作業的執行方式，以及是否已指定替代使用者 ID。

表 *57:* 針對批次連線的設定檔名稱檢查使用者 *ID*

開啟時指定的替代使用者 ID?	hlq.ALTERNATE.USER.userid 設定檔	hlq.CONTEXT.queuename 設定檔	hlq.resourcename 設定檔
否	-	工作	工作
是	工作	工作	ALT

索引鍵:

ALT
替代使用者 ID。

工作

- TSO 或 z/OS UNIX System Services 登入的使用者 ID。
- 指派給批次工作的使用者 ID。
- STARTED 類別或已啟動程序表格指派給已啟動作業的使用者 ID。
- 與執行中 Db2 儲存程序相關聯的使用者 ID

「批次」工作正在對稱為 Q1 且 RESLEVEL 設為 READ 及替代使用者 ID 檢查已關閉的佇列執行 MQPUT1 。

在批次連線的不同 RACF(r) 存取層次所進行的檢查 和 根據批次連線的設定檔名稱來檢查使用者 ID 會顯示根據設定檔 hlq.Q1 來檢查工作使用者 ID。

z/OS 已檢查 *CICS* 連線的使用者 *ID*

針對 CICS 連線檢查的使用者 ID 取決於是否要執行一或兩項檢查，以及是否指定替代使用者 ID。

表 *58:* 針對 *CICS-type* 使用者 *ID* 的設定檔名稱檢查使用者 *ID*

開啟時指定的替代使用者 ID?	hlq.ALTERNATE.USER.userid 設定檔	hlq.CONTEXT.queuename 設定檔	hlq.resourcename 設定檔
否， *1* 檢查	-	ADS	ADS
否， *2* 個檢查	-	ADS + TXN	ADS + TXN
是， *1* 個檢查	ADS	ADS	ADS
是， *2* 個檢查	ADS + TXN	ADS + TXN	ADS + ALT

索引鍵:

ALT
替代使用者 ID

ADS
與 CICS 批次工作相關聯的使用者 ID ，或者如果 CICS 以已啟動的作業形式執行，則透過 STARTED 類別或已啟動的程序表格。

TXN
與 CICS 交易相關聯的使用者 ID。 這通常是啟動交易之終端機使用者的使用者 ID。 它可以是 CICS DFLTUSER、PRESET 安全終端機或手動登入使用者。

判定針對下列條件所檢查的使用者 ID:

- RESLEVEL 設定檔的 RACF 存取層次 (針對 CICS 位址空間使用者 ID) 設為 NONE。
- 對具有 MQOO_OUTPUT 及 MQOO_PASS_IDENTITY_CONTEXT 的佇列發出 MQOPEN 呼叫。

首先，請查看根據 RESLEVEL 設定檔的 CICS 位址空間使用者 ID 存取權來檢查多少 CICS 使用者 ID。 從主題 第 197 頁的『RESLEVEL 及 CICS 連線』中的 第 198 頁的表 53 ，如果 RESLEVEL 設定檔設為 NONE ，則會檢查兩個使用者 ID。 然後，從 第 202 頁的表 58 開始，會執行下列檢查:

- hlq.ALTERNATE.USER.userid 設定檔。
- 同時使用 CICS 位址空間使用者 ID 和 CICS 交易使用者 ID 來檢查 hlq.CONTEXT.queuename 設定檔。
- 同時使用 CICS 位址空間使用者 ID 和 CICS 交易使用者 ID 來檢查 hlq.resourcename 設定檔。

這表示對此 MQOPEN 呼叫進行四個安全檢查。

z/OS 已檢查 *IMS* 連線的使用者 *ID*

針對 IMS 連線檢查的使用者 ID 取決於是否要執行一或兩項檢查，以及是否指定替代使用者 ID。 如果勾選第二個使用者 ID ，則取決於相依區域的類型以及可用的使用者 ID。

表 59: 針對 *IMS-type* 使用者 *ID* 的設定檔名稱檢查使用者 *ID*			
開啟時指定的替代使用者 ID?	**hlq.ALTERNATE.USER.userid 設定檔**	**hlq.CONTEXT.queuename 設定檔**	**hlq.resourcename 設定檔**
否， *1* 檢查	-	REG	REG
否， *2* 個檢查	-	REG + SEC	REG + SEC
是， *1* 個檢查	REG	REG	REG
是， *2* 個檢查	REG + SEC	REG + SEC	REG + ALT

索引鍵:

ALT
替代使用者 ID。

REG
使用者 ID 通常是透過 STARTED 類別或已啟動的程序表格來設定，如果 IMS 在執行中，則由 USER JCL 參數從提交的工作來設定。

秒
第二個使用者 ID 與在相依區域中執行的工作相關聯。 它是根據 第 203 頁的表 60 來決定。

表 60: 如何判定 *IMS* 連線的第二個使用者 *ID*	
相依區域的類型	**用來決定第二個使用者 ID 的階層**
• BMP 訊息驅動及順利發出 GET UNIQUE。 • 已發出 RFP 和 GET UNIQUE。 • MPP。	與 IMS 交易相關聯的使用者 ID (如果使用者已登入)。 LTERM 名稱 (如果有的話)。 PSBNAME。
• 未發出 BMP 訊息驅動及順利完成 GET UNIQUE。 • BMP 不是訊息驅動。 • 未發出 RFP 和 GET UNIQUE。	如果這不是全部空白或全部為零，則與 IMS 相依區域位址空間相關聯的使用者 ID。 PSBNAME。

z/OS 通道起始程式使用的使用者 *ID*

此主題集合說明針對接收通道及透過伺服器連線通道發出的用戶端 MQI 要求所使用及檢查的使用者 ID。 提供 TCP/IP 及 LU6.2 的相關資訊

您可以使用接收端通道定義的 PUTAUT 參數來決定使用的安全檢查類型。 若要在整個 IBM MQ 網路中取得一致的安全檢查，您可以使用 ONLYMCA 及 ALTMCA 選項。

您可以使用 DISPLAY CHSTATUS 指令來決定 MCA 所使用的使用者 ID。

z/OS 使用 *TCP/IP* 的接收通道

所檢查的使用者 ID 取決於通道的 PUTAUT 選項，以及是否要執行一或兩項檢查。

表 61: 針對 *TCP/IP* 通道的設定檔名稱檢查使用者 *ID*			
在接收端或要求端通道上指定的 PUTAUT 選項	**hlq.ALTERNATE.USER.userid 設定檔**	**hlq.CONTEXT.queuename 設定檔**	**hlq.resourcename 設定檔**
***DEF*， *1* 檢查**	-	CHL	CHL
***DEF*， *2* 個檢查**	-	CHL + MCA	CHL + MCA
***CTX*， *1* 檢查**	CHL	CHL	CHL

表 61: 針對 *TCP/IP* 通道的設定檔名稱檢查使用者 *ID* (繼續)			
在接收端或要求端通道上指定的 PUTAUT 選項	**hlq.ALTERNATE.USER.userid 設定檔**	**hlq.CONTEXT.queuename 設定檔**	**hlq.resourcename 設定檔**
***CTX*， *2* 個檢查**	CHL + MCA	CHL + MCA	CHL + ALT
ONLYMCA*， *1 check	-	MCA	MCA
***ONLYMCA*， *2* 個檢查**	-	MCA	MCA
***ALTMCA*， *1* 個檢查**	MCA	MCA	MCA
***ALTMCA*， *2* 個檢查**	MCA	MCA	MCA + ALT

索引鍵:

MCA (MCA 使用者 ID)
在接收端指定給 MCAUSER 通道屬性的使用者 ID; 如果空白，則會使用接收端或要求端的通道起始程式位址空間使用者 ID。

CHL (通道使用者 ID)
在 TCP/IP 上，通道的通訊系統不支援安全。 如果正在使用「傳輸層安全 (TLS)」，且已從夥伴傳送數位憑證，則會使用與此憑證相關聯的使用者 ID (如果已安裝)，或與使用「 RACF 憑證名稱過濾 (CNF)」找到的相符過濾器相關聯的使用者 ID。 如果找不到相關聯的使用者 ID，或未使用 TLS，則會使用接收端或要求端之通道起始程式位址空間的使用者 ID，作為 PUTAUT 參數設為 DEF 或 CTX 之通道上的通道使用者 ID。

註: 使用「 RACF 憑證名稱過濾 (CNF)」可讓您將相同的 RACF 使用者 ID 指派給多個遠端使用者，例如相同組織單位中的所有使用者，他們自然都具有相同的安全權限。 這表示伺服器不必擁有全球每一個可能的遠端使用者的憑證副本，並大幅簡化憑證管理及配送。

如果通道的 PUTAUT 參數設為 ONLYMCA 或 ALTMCA，則會忽略通道使用者 ID，並使用接收端或要求端的 MCA 使用者 ID。 這也適用於使用 TLS 的 TCP/IP 通道。

ALT (替代使用者 ID)
來自訊息的訊息描述子內環境定義資訊 (即 *UserIdentifier* 欄位) 的使用者 ID。 在對目標目的地佇列發出 **MQOPEN** 或 **MQPUT1** 呼叫之前，此使用者 ID 會移至物件描述子中的 *AlternateUserID* 欄位。

z/OS 使用 *LU 6.2* 接收通道
所檢查的使用者 ID 取決於通道的 PUTAUT 選項，以及是否要執行一或兩項檢查。

表 62: 針對 *LU 6.2* 通道的設定檔名稱檢查使用者 *ID*			
在接收端或要求端通道上指定的 PUTAUT 選項	**hlq.ALTERNATE.USER.userid 設定檔**	**hlq.CONTEXT.queuename 設定檔**	**hlq.resourcename 設定檔**
***DEF*， *1* 檢查**	-	CHL	CHL
***DEF*， *2* 個檢查**	-	CHL + MCA	CHL + MCA
***CTX*， *1* 檢查**	CHL	CHL	CHL
***CTX*， *2* 個檢查**	CHL + MCA	CHL + MCA	CHL + ALT
ONLYMCA*， *1 check	-	MCA	MCA
***ONLYMCA*， *2* 個檢查**	-	MCA	MCA
***ALTMCA*， *1* 個檢查**	MCA	MCA	MCA
***ALTMCA*， *2* 個檢查**	MCA	MCA	MCA + ALT

索引鍵:

MCA (MCA 使用者 ID)
在接收端指定給 MCAUSER 通道屬性的使用者 ID; 如果空白，則會使用接收端或要求端的通道起始程式位址空間使用者 ID。

CHL (通道使用者 ID)

要求端-伺服器通道
如果通道是從要求端啟動，則沒有機會接收網路使用者 ID (通道使用者 ID)。

如果要求端通道上的 PUTAUT 參數設為 DEF 或 CTX，則通道使用者 ID 是要求端通道起始程式位址空間的通道使用者 ID，因為沒有從網路收到任何使用者 ID。

如果 PUTAUT 參數設為 ONLYMCA 或 ALTMCA，則會忽略通道使用者 ID，並使用要求端的 MCA 使用者 ID。

其他通道類型
如果在接收端或要求端通道上將 PUTAUT 參數設為 DEF 或 CTX，則通道使用者 ID 是起始通道時從通訊系統收到的使用者 ID。

- 如果傳送端通道是在 z/OS 上，則接收的通道使用者 ID 是傳送端的通道起始程式位址空間使用者 ID。
- 如果傳送端通道位於不同的平台 (例如， AIX)，則通常由通道定義的 USERID 參數提供接收的通道使用者 ID。

如果收到的使用者 ID 是空白，或未收到任何使用者 ID，則會使用空白的通道使用者 ID。

ALT (替代使用者 ID)
來自訊息的訊息描述子內環境定義資訊 (即 *UserIdentifier* 欄位) 的使用者 ID。 在針對目標目的地佇列發出 MQOPEN 或 MQPUT1 呼叫之前，此使用者 ID 會移至物件描述子中的 *AlternateUserID* 欄位。

z/OS 用戶端 *MQI* 要求
視已設定的使用者 ID 和環境變數而定，可以使用各種使用者 ID。 視使用的 PUTAUT 選項以及是否指定替代使用者 ID 而定，會根據各種設定檔來檢查這些使用者 ID。

本節說明針對 TCP/IP 及 LU 6.2 透過伺服器連線通道發出的用戶端 MQI 要求所檢查的使用者 ID。 MCA 使用者 ID 及通道使用者 ID 適用於前述各節所說明的 TCP/IP 及 LU 6.2 通道。

對於伺服器連線通道，如果 MCAUSER 屬性空白，則會使用從用戶端收到的使用者 ID。

如需相關資訊，請參閱 第 83 頁的『用戶端的存取控制』。

對於用戶端 **MQOPEN**、 **MQSUB** 和 **MQPUT1** 要求，請使用下列規則來決定所檢查的設定檔:

- 如果要求指定替代使用者權限，則會針對 *hlq*.ALTERNATE.USER。 *userid* 設定檔。
- 如果要求指定環境定義權限，則會對 *hlq* 進行檢查。環境定義。 *queuename* 設定檔。
- 對於所有 **MQOPEN**、 **MQSUB** 及 **MQPUT1** 要求，會對 *hlq.resourcename* 設定檔進行檢查。

當您決定要檢查哪些設定檔時，請使用下表來決定要根據這些設定檔來檢查哪些使用者 ID。

表 *63:* 針對 *LU 6.2* 及 *TCP/IP* 伺服器連線通道的設定檔名稱檢查使用者 *ID*

在伺服器連線通道上指定的 PUTAUT 選項	開啟時指定的替代使用者 ID?	hlq.ALTERNATE.USER.userid 設定檔	hlq.CONTEXT.queuename 設定檔	hlq.resourcename 設定檔
***DEF*，*1* 檢查**	否	-	CHL	CHL
***DEF*，*1* 檢查**	是	CHL	CHL	CHL

表 63: 針對 *LU 6.2* 及 *TCP/IP* 伺服器連線通道的設定檔名稱檢查使用者 *ID* (繼續)				
在伺服器連線通道上指定的 PUTAUT 選項	**開啟時指定的替代使用者 ID?**	**hlq.ALTERNATE.USER.userid 設定檔**	**hlq.CONTEXT.queuename 設定檔**	**hlq.resourcename 設定檔**
***DEF*， *2* 個檢查**	否	-	CHL + MCA	CHL + MCA
***DEF*， *2* 個檢查**	是	CHL + MCA	CHL + MCA	CHL + ALT
ONLYMCA*， *1 check	否	-	MCA	MCA
ONLYMCA*， *1 check	是	MCA	MCA	MCA
***ONLYMCA*， *2* 個檢查**	否	-	MCA	MCA
***ONLYMCA*， *2* 個檢查**	是	MCA	MCA	MCA + ALT

索引鍵:

MCA (MCA 使用者 ID)
在 server-connection 上為 MCAUSER 通道屬性指定的使用者 ID; 如果空白，則會使用通道起始程式位址空間使用者 ID。

CHL (通道使用者 ID)
在 TCP/IP 上，通道的通訊系統不支援安全。 如果正在使用「傳輸層安全 (TLS)」，且已從夥伴傳送數位憑證，則會使用與此憑證相關聯的使用者 ID (如果已安裝)，或與使用「 RACF 憑證名稱過濾 (CNF)」找到的相符過濾器相關聯的使用者 ID。 如果找不到相關聯的使用者 ID，或未使用 TLS，則會使用通道起始程式位址空間的使用者 ID，作為在 PUTAUT 參數設為 DEF 或 CTX 的通道上定義的通道使用者 ID。

註: 使用「 RACF 憑證名稱過濾 (CNF)」可讓您將相同的 RACF 使用者 ID 指派給多個遠端使用者，例如相同組織單位中的所有使用者，他們自然都具有相同的安全權限。 這表示伺服器不必擁有全球每一個可能的遠端使用者的憑證副本，並大幅簡化憑證管理及配送。

如果通道的 PUTAUT 參數設為 ONLYMCA 或 ALTMCA，則會忽略通道使用者 ID，並使用伺服器連線通道的 MCA 使用者 ID。 這也適用於使用 TLS 的 TCP/IP 通道。

ALT (替代使用者 ID)
來自訊息的訊息描述子內環境定義資訊 (即 *UserIdentifier* 欄位) 的使用者 ID。 在代表用戶端應用程式發出 **MQOPEN**、 **MQSUB** 或 **MQPUT1** 呼叫之前，此使用者 ID 會移至物件或訂閱描述子中的 *AlternateUserID* 欄位。

z/OS 通道起始程式範例
如何根據 RACF 設定檔來檢查使用者 ID 的範例。

使用者對佇列管理程式 QM01 上的佇列執行 **MQPUT1** 作業，該佇列管理程式會解析為佇列管理程式 QM02 上稱為 QB 的佇列。 訊息會在稱為 QM01.TO.QM02。 RESLEVEL 設為 NONE，並使用替代使用者 ID 及環境定義檢查來執行開啟。 接收端通道定義具有 PUTAUT (CTX)，且已設定 MCA 使用者 ID。 在接收通道上使用哪些使用者 ID 將訊息放入佇列 QB?

回答: 第 199 頁的表 55 顯示已檢查兩個使用者 ID，因為 RESLEVEL 設為 NONE。

第 203 頁的表 61 顯示當 PUTAUT 設為 CTX 及 2 個檢查時，會檢查下列使用者 ID:

- 會根據 hlq.ALTERNATE.USER.userid 設定檔。
- 會根據 hlq.CONTEXT.queuename 設定檔來檢查通道起始程式使用者 ID 及 MCAUSER 使用者 ID。

- 會根據 hlq.Q2 設定檔來檢查訊息描述子 (MQMD) 中指定的通道起始程式使用者 ID 及替代使用者 ID。

z/OS 內部群組佇列作業代理程式使用的使用者 *ID*

當內部群組佇列作業代理程式開啟目的地佇列時所檢查的使用者 ID，由 IGQAUT 及 IGQUSER 佇列管理程式屬性的值決定。

可能的使用者 ID 為:

內部群組佇列作業使用者 ID (IGQ)
由接收端佇列管理程式的 IGQUSER 屬性決定的使用者 ID。 如果設為空白，則會使用接收端佇列管理程式的使用者 ID。 不過，因為接收端佇列管理程式有權存取定義給它的所有佇列，所以不會對接收端佇列管理程式的使用者 ID 執行安全檢查。 在此情況下:

- 如果只檢查一個使用者 ID，且使用者 ID 是接收端佇列管理程式的使用者 ID，則不會進行安全檢查。 當 IGQAUT 設為 ONLYIGQ 或 ALTIGQ 時，可能會發生此情況。
- 如果要檢查兩個使用者 ID，且其中一個使用者 ID 是接收端佇列管理程式的使用者 ID，則只會對另一個使用者 ID 進行安全檢查。 當 IGQAUT 設為 DEF、CTX 或 ALTIGQ 時，可能會發生這種情況。
- 如果要檢查兩個使用者 ID，且這兩個使用者 ID 都是接收端佇列管理程式的使用者 ID，則不會進行安全檢查。 當 IGQAUT 設為 ONLYIGQ 時，可能會發生這種情況。

傳送佇列管理程式使用者 ID (SND)
將訊息放入 SYSTEM.QSG.TRANSMIT.QUEUE。

替代使用者 ID (ALT)
在訊息描述子的 *UserIdentifier* 欄位中指定的使用者 ID。

表 *64:* 針對內部群組佇列作業的設定檔名稱檢查使用者 *ID*

在接收佇列管理程式上指定 IGQAUT 選項	hlq.ALTERNATE.USER.userid 設定檔	hlq.CONTEXT.queuename 設定檔	hlq.resourcename 設定檔
***DEF*，*1* 檢查**	-	SND	SND
***DEF*，*2* 個檢查**	-	SND + IGQ	SND + IGQ
***CTX*，*1* 檢查**	SND	SND	SND
***CTX*，*2* 個檢查**	SND + IGQ	SND + IGQ	SND + ALT
ONLYIGQ*，*1 check	-	IGQ	IGQ
***ONLYIGQ*，*2* 個檢查**	-	IGQ	IGQ
***ALTIGQ*，*1* 檢查**	-	IGQ	IGQ
***ALTIGQ*，*2* 個檢查**	IGQ	IGQ	IGQ + ALT

索引鍵:

ALT
替代使用者 ID。

IGQ
IGQ 使用者 ID。

SND
正在傳送佇列管理程式使用者 ID。

z/OS 空白使用者 *ID* 和 *UACC* 層次

如果出現空白使用者 ID，則會登入 RACF 未定義的使用者。 請勿將廣泛存取權授與未定義的使用者。

當使用者使用環境定義或替代使用者安全來操作訊息時，或當 IBM MQ 傳遞空白使用者 ID 時，可能存在空白使用者 ID。 例如，當訊息寫入沒有環境定義的系統指令輸入佇列時，會使用空白使用者 ID。

註: 使用者 ID 為 " * " (亦即，星號字元後接七個空格) 被視為未定義的使用者 ID。

IBM MQ 會將空白使用者 ID 傳遞至 RACF，且會登入 RACF 未定義的使用者。 然後所有安全檢查都會使用相關設定檔的通用存取權 (UACC)。 視您設定存取層次的方式而定， UACC 可能會提供廣泛的存取權給未定義的使用者。

例如，如果您從 TSO 發出此 RACF 指令:

```
RDEFINE MQQUEUE Q.AVAILABLE.TO.EVERYONE UACC(UPDATE)
```

您定義的設定檔可讓 z/OS 定義的使用者 ID (尚未放入存取清單中) 和 RACF 未定義的使用者 ID 將訊息放置在該佇列上，並從中取得訊息。

為了防範空白使用者 ID，您必須小心規劃存取層次，並限制可以使用環境定義和替代使用者安全的人員數目。 您必須防止使用 RACF 未定義使用者 ID 的人員存取他們不得存取的資源。 不過，同時，您必須容許存取具有已定義使用者 ID 的人員。 若要這樣做，您可以在 RACF 指令 PERMIT 中指定星號 (*) 的使用者 ID，以授與所有已定義使用者 ID 的資源存取權。 因此，所有未定義的使用者 ID (例如 " * ") 拒絕存取。 例如，這些 RACF 指令會阻止 RACF 未定義的使用者 ID 取得佇列的存取權，以放置或取得訊息:

```
RDEFINE MQQUEUE Q.AVAILABLE.TO.RACF.DEFINED.USERS.ONLY UACC(NONE)
PERMIT Q.AVAILABLE.TO.RACF.DEFINED.USERS.ONLY CLASS(MQQUEUE) ACCESS(UPDATE) ID(*)
```

z/OS z/OS 使用者 ID 及多因子鑑別 (MFA)

IBM Multi-Factor Authentication for z/OS 可讓 z/OS 安全管理者加強 SAF 鑑別，方法是要求已識別使用者使用多個鑑別因素 (例如，密碼和加密記號) 來登入 z/OS 系統。 IBM MFA 也支援基於時間的一次性密碼產生技術，例如 RSA SecureId。

在大部分情況下， IBM MQ 並不知道使用者如何「登入」 CICS 或驅動 IBM MQ 工作的批次系統，登入的使用者 ID 認證會與 z/OS 作業或位址空間相關聯，且 IBM MQ 會使用此認證來檢查資源的授權。 針對 MFA 啟用的使用者 ID 可以用於對 IBM MQ 資源的授權，以及透過與 CICS 及 IMS 橋接器搭配使用的通行證進行鑑別。

重要: 不過，當使用應用程式 (例如 IBM MQ Explorer) 時，會以 *MQCSP_AUTH_USER_ID_AND_PWD* 選項傳遞 MQCONNX API 呼叫的使用者 ID 和密碼認證。 IBM MQ 沒有機能在此 API 要求上傳遞其他認證。

下列文字說明限制及可能的暫行解決方法。

IBM MQ Explorer

IBM MQ Explorer 無法用來以已啟用 MFA 的使用者 ID 登入 z/OS 系統，因為沒有將第二鑑別因素從 IBM MQ Explorer 傳遞至 z/OS 的機能。

此外， IBM MQ Explorer 使用兩種不同的機制來重複使用使用者 ID 和密碼認證，當一次使用密碼生效時需要特別注意:

1. IBM MQ Explorer 能夠在本端機器上以模糊化格式儲存密碼，以供稍後登入。 每次建立與 z/OS 佇列管理程式的連線時，都必須讓瀏覽器提示輸入密碼，以停用此功能。

   如果要執行這個動作，請使用下列程序:

   a. 選取 **佇列管理程式**。
   b. 從顯示的清單中，選擇您需要的佇列管理程式，然後用滑鼠右鍵按一下該佇列管理程式。
   c. 從出現的功能表清單中選取 **連線詳細資料** 。
   d. 從下一個功能表清單中選取 **內容** ，然後選擇 **使用者 ID** 標籤。

      請確定您選取 **提示輸入密碼** 圓鈕。
2. IBM MQ Explorer 中的各種作業 (例如瀏覽佇列上的訊息、測試訂閱等) 會啟動新的執行緒，以使用登入時第一次使用的認證來鑑別 IBM MQ 。 由於無法重複使用密碼認證，因此您無法使用這些作業。

這些問題在 MFA 配置層次有兩個可能的暫行解決方法:

- 使用 MFA 的應用程式 ID 排除，從 MFA 處理中完全排除 IBM MQ 作業。

  若要執行此動作，請發出下列指令:

1. `RDEFINE MFADEF MFABYPASS.USERID.chinuser`

   其中 *chinuser* 是通道起始程式位址空間層次使用者 ID (透過 STC 類別與通道起始程式相關聯)

2. `PERMIT MFABYPASS.USERID.chinuser CLASS MFADEF ACCESS(READ) ID(explorer user)`

如需此方法的相關資訊，請參閱 略過 IBM 應用程式的 MFA。

- 在 IBM MFA 1.2 引進的 MFA 上使用頻外支援。 使用此方法，您可以向 IBM MFA Web 伺服器預先鑑別，並且除了您的使用者 ID 和密碼之外，還指定透過原則判定的其他鑑別。 IBM MFA 伺服器會產生快取記號認證，然後您在 IBM MQ Explorer 鑑別對話框上指定該認證。 安全管理者可以容許在合理期間內重播此認證，因此啟用正常 IBM MQ Explorer 使用。

如需此方法的相關資訊，請參閱 IBM MFA 簡介。

z/OS IBM MQ for z/OS 安全管理

IBM MQ 使用儲存體內表格來保留每一個使用者的相關資訊，以及每一個使用者所提出的存取要求。 為了有效地管理此表格，並減少從 IBM MQ 向外部安全管理程式 (ESM) 提出的要求數，提供了一些控制項。

這些控制項可透過作業及控制面板及 IBM MQ 指令來使用。

z/OS 使用者 *ID* 重新驗證

如果使用 IBM MQ 資源之使用者的 RACF 定義已變更 (例如透過將使用者連接至新群組)，則您可以告知佇列管理程式在下次嘗試存取 IBM MQ 資源時再次登入此使用者。 您可以使用 IBM MQ 指令 RVERIFY SECURITY 來執行此動作。

- 使用者 HX0804 正在取得訊息並將訊息放入佇列管理程式 PRD1 上的 PAYROLL 佇列。 不過， HX0804 現在需要存取相同佇列管理程式 (PRD1) 上的部分 PENSION 佇列。
- 資料安全管理者將使用者 HX0804 連接至容許存取 PENSION 佇列的 RACF 群組。
- 為了讓 HX0804 可以立即存取 PENSION 佇列 (亦即，不需要關閉佇列管理程式 PRD1 或等待 HX0804 逾時)， 您必須使用 IBM MQ 指令:

```
RVERIFY SECURITY(HX0804)
```

註: 當佇列管理程式正在執行時，如果您長時間關閉使用者 ID 逾時 (天或甚至週)，則必須記得針對在該時間內已撤銷或刪除的任何使用者執行 RVerify SECURITY 指令。

z/OS 使用者 *ID* 逾時

在閒置一段時間之後，您可以讓 IBM MQ 讓使用者登出佇列管理程式。

當使用者存取 IBM MQ 資源時，佇列管理程式會嘗試將此使用者登入佇列管理程式 (如果子系統安全作用中)。 這表示會向 ESM 鑑別使用者。 此使用者會保持登入 IBM MQ，直到佇列管理程式關閉，或直到使用者 ID 逾時 (鑑別失效) 或重新驗證 (重新鑑別) 為止。

當使用者逾時，佇列管理程式內的使用者 ID 會 登出，且會捨棄為這個使用者保留的任何安全相關資訊。 對於應用程式或使用者而言，登入及登出佇列管理程式內的使用者並不明顯。

如果使用者在預先決定的時間量內未使用任何 IBM MQ 資源，則有資格逾時。 此時段由 MQSC ALTER SECURITY 指令設定。

在 ALTER SECURITY 指令中可以指定兩個值:

逾時
未用使用者 ID 及其相關聯資源可以保留在 IBM MQ 佇列管理程式內的時段 (分鐘)。

INTERVAL
檢查使用者 ID 及其相關聯資源之間的時段 (分鐘)，以判定 *TIMEOUT* 是否已過期。

例如，如果 *TIMEOUT* 值為 30 且 *INTERVAL* 值為 10，則每 10 分鐘 IBM MQ 會檢查使用者 ID 及其相關聯資源，以判斷是否有任何使用者 ID 及其相關聯資源已 30 分鐘未使用。 如果找到逾時的使用者 ID，則會將該使用者 ID 登出佇列管理程式。 如果找到任何與非逾時使用者 ID 相關聯的逾時資源資訊，則會捨棄該資源資訊。 如果您不想讓使用者 ID 逾時，請將 *INTERVAL* 值設為零。 不過，如果 *INTERVAL* 值為零，則除非您發

出 **REFRESH SECURITY** 或 **RVERIFY SECURITY** 指令，否則不會釋放使用者 ID 所佔用的儲存體及其相關聯資源。

如果您有許多一次性使用者，則調整此值可能很重要。 如果您設定小間隔及逾時值，則會釋放不再需要的資源。

註: 如果您使用非預設值的 *INTERVAL* 或 *TIMEOUT* 值，則必須在每次啟動佇列管理程式時重新輸入指令。 您可以將 **ALTER SECURITY** 指令放入該佇列管理程式的 CSQINP1 資料集，以自動執行此動作。

z/OS 重新整理 *z/OS* 上的佇列管理程式安全

IBM MQ for z/OS 會快取 RACF 資料以增進效能。 當您變更特定安全類別時，必須重新整理此快取資訊。 基於效能原因，不常重新整理安全。 您也可以選擇只重新整理 TLS 安全資訊。

當第一次開啟佇列 (或在安全重新整理之後第一次開啟) 時， IBM MQ 會執行 RACF 檢查以取得使用者的存取權，並將此資訊放在快取中。 快取的資料包括已執行安全檢查的使用者 ID 和資源。 如果相同使用者再次開啟佇列，則存在快取資料表示 IBM MQ 不需要發出 RACF 檢查，這會增進效能。 安全重新整理的動作是捨棄任何快取的安全資訊，因此強制 IBM MQ 對 RACF 進行新的檢查。 每當您新增、變更或刪除保留在 MQADMIN、MXADMIN、MQPROC、MXPROC、MQQUEUE、MXQUEUE、MQNLIST、MXNLIST 或 MXTOPIC 類別中的 RACF 資源設定檔時，您必須告知使用此類別的佇列管理程式，以重新整理它們所保留的安全資訊。 若要執行此動作，請發出下列指令:

- RACF SETROPTS RACLIST (classname) REFRESH 指令，以在 RACF 層次重新整理。
- IBM MQ REFRESH SECURITY 指令可重新整理佇列管理程式所保留的安全資訊。 此指令需要由存取已變更設定檔的每一個佇列管理程式發出。 如果您有佇列共用群組，您可以使用指令範圍屬性，將指令引導至群組中的所有佇列管理程式。

  註: 如果您已將新使用者連接至現有群組，則需要執行 IBM MQ RVerify SECURITY(使用者 ID) 指令。 REFRESH SECURITY (*) 指令不會讓佇列管理程式在下次嘗試存取 IBM MQ 資源時再次登入此使用者。

如果您在任何 IBM MQ 類別中使用通用設定檔，當您變更、新增或刪除任何通用設定檔時，也必須發出一般 RACF 重新整理指令。 例如， SETROPTS GENERIC (classname) 重新整理。

不過，如果新增、變更或刪除 RACF 資源設定檔，且尚未存取它所套用的資源 (因此不會快取任何資訊) ，則 IBM MQ 會使用新的 RACF 資訊，而不會發出 REFRESH SECURITY 指令。

如果開啟 RACF 審核 (例如，使用 RACF RALTER AUDIT (access-attempt (audit_access_level)) 指令) ，則不會進行快取，因此 IBM MQ 會直接參照每次檢查的 RACF 資料空間。 因此會立即挑選變更，且不需要 REFRESH SECURITY 即可存取變更。 您可以使用 RACF RLIST 指令來確認 RACF 審核是否已開啟。 例如，您可以發出指令

```
RLIST MQQUEUE (qmgr.SYSTEM.COMMAND.INPUT) GEN
```

並接收結果

```
CLASS      NAME
-----      ----
MQQUEUE    QP*.SYSTEM.COMMAND.*.** (G)
    AUDITING
    --------
    FAILURES(READ)
```

這指出已設定審核。 如需相關資訊，請參閱 *z/OS Security Server RACF* 審核員手冊 和 *z/OS Security Server RACF* 指令語言參考手冊。

第 211 頁的圖 17 彙總了快取安全資訊及使用快取資訊的狀況。

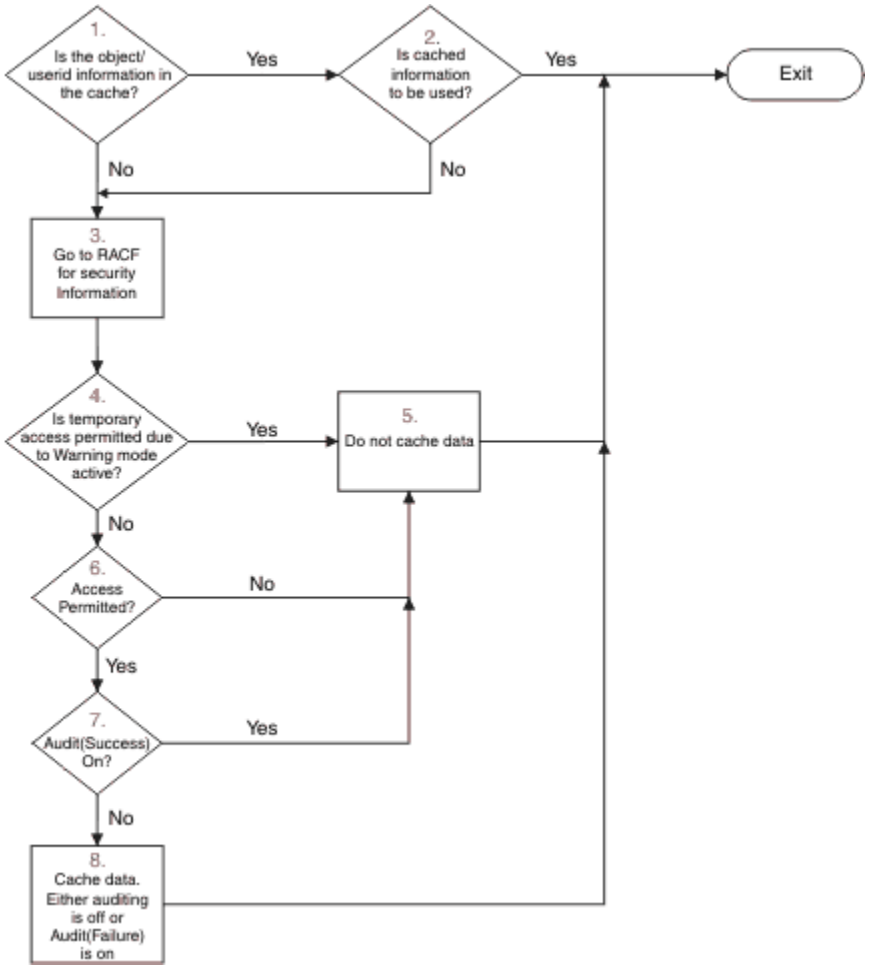


圖 *17: IBM MQ* 安全快取的邏輯流程

如果您透過在 MQADMIN 或 MXADMIN 類別中新增或刪除交換器設定檔來變更安全設定，請使用下列其中一個指令來動態挑選這些變更:

REFRESH SECURITY (*)
重新整理安全 (MQADMIN)
重新整理安全 (MXADMIN)

這表示您可以啟動新的安全類型，或取消啟動它們，而不需要重新啟動佇列管理程式。

基於效能原因，這些是唯一受 REFRESH SECURITY 指令影響的類別。 如果您變更 MQCONN 或 MQCMDS 類別中的設定檔，則不需要使用 REFRESH SECURITY。

註: 如果您變更 RESLEVEL 安全設定檔，則不需要重新整理 MQADMIN 或 MXADMIN 類別。

基於效能原因，請儘可能不常使用 REFRESH SECURITY ，最好在離峰時間使用。 您可以透過將使用者連接至已在 IBM MQ 設定檔存取清單中的 RACF 群組，而不是將個別使用者置於存取清單中，來最小化安全重新整理的次數。 以此方式，您可以變更使用者，而不是資源設定檔。 您也可以將 RVERIFY SECURITY 作為適當的使用者，而不是重新整理安全。

作為 REFRESH SECURITY 的範例，假設您定義新的設定檔，以保護對佇列管理程式 PRMQ 上以 INSURANCE.LIFE 開頭之佇列的存取權。 您可以使用下列 RACF 指令:

```
RDEFINE MQQUEUE PRMQ.INSURANCE.LIFE.** UACC(NONE)
PERMIT PRMQ.INSURANCE.LIFE.** ID(LIFEGRP) ACCESS(UPDATE)
```

您必須發出下列指令，以告知 RACF 重新整理它所保留的安全資訊，例如:

```
SETROPTS RACLIST(MQQUEUE) REFRESH
```

因為這些設定檔是通用的，所以您必須告知 RACF 重新整理 MQQUEUE 的通用設定檔。 例如：

```
SETROPTS GENERIC(MQQUEUE) REFRESH
```

然後，您必須使用此指令，告知佇列管理程式 PRMQ 佇列設定檔已變更：

```
REFRESH SECURITY(MQQUEUE)
```

重新整理 SSL/TLS 安全

若要重新整理「TLS 金鑰儲存庫」的快取視圖，請發出具有選項 TYPE (SSL) 的 REFRESH SECURITY 指令。 這可讓您更新部分 TLS 設定，而無需重新啟動通道起始程式。

z/OS 顯示安全狀態

若要顯示安全開關及其他安全控制項的狀態，請發出 MQSC DISPLAY SECURITY 指令。

下圖顯示 DISPLAY SECURITY ALL 指令的一般輸出。

```
CSQH015I +CSQ1 Security timeout = 54 MINUTES
CSQH016I +CSQ1 Security interval = 12 MINUTES
CSQH030I +CSQ1 Security switches ...
CSQH034I +CSQ1 SUBSYSTEM: ON, 'SQ05.NO.SUBSYS.SECURITY' not found
CSQH032I +CSQ1 QMGR: ON, 'CSQ1.YES.QMGR.CHECKS' found
CSQH031I +CSQ1 QSG: OFF, 'SQ05.NO.QSG.CHECKS' found
CSQH031I +CSQ1 CONNECTION: OFF, 'CSQ1.NO.CONNECT.CHECKS' found
CSQH034I +CSQ1 COMMAND: ON, 'CSQ1.NO.COMMAND.CHECKS' not found
CSQH031I +CSQ1 CONTEXT: OFF, 'CSQ1.NO.CONTEXT.CHECKS' found
CSQH034I +CSQ1 ALTERNATE USER: ON, 'CSQ1.NO.ALTERNATE.USER.CHECKS' not found
CSQH034I +CSQ1 PROCESS: ON, 'CSQ1.NO.PROCESS.CHECKS' not found
CSQH034I +CSQ1 NAMELIST: ON, 'CSQ1.NO.NLIST.CHECKS' not found
CSQH034I +CSQ1 QUEUE: ON, 'CSQ1.NO.QUEUE.CHECKS' not found
CSQH034I +CSQ1 TOPIC: ON, 'CSQ1.NO.TOPIC.CHECKS' not found
CSQH031I +CSQ1 COMMAND RESOURCES: OFF, 'CSQ1.NO.CMD.RESC.CHECKS' found
CSQ9022I +CSQ1 CSQHPDTC ' DISPLAY SECURITY' NORMAL COMPLETION
```

圖 *18: DISPLAY SECURITY* 指令的一般輸出

此範例顯示回覆指令的佇列管理程式在佇列管理程式層次具有作用中的子系統、指令、替代使用者、處理程序、名單及佇列安全，但不在佇列共用群組層次。 連線、指令資源及環境定義安全不在作用中。 它也會顯示使用者 ID 逾時處於作用中，且每 12 分鐘佇列管理程式會檢查此佇列管理程式中是否有 54 分鐘未使用的使用者 ID ，並移除它們。

註: 此指令會顯示現行安全狀態。 它不一定會反映定義給 RACF 之交換器設定檔的現行狀態，或 RACF 類別的狀態。 例如，自前次重新啟動此佇列管理程式或 REFRESH SECURITY 指令後，交換器設定檔可能已變更。

z/OS z/OS 的安全安裝作業

在安裝及自訂 IBM MQ 之後，請授權啟動型作業程序給 RACF，授權存取各種資源，以及設定 RACF 定義。 選擇性地配置系統以使用 TLS。

第一次安裝及自訂 IBM MQ 時，您必須執行下列安全相關作業：

1. 透過下列方式設定 IBM MQ 資料集及系統安全：
   - 授權佇列管理程式啟動型作業程序 xxxxMSTR 及分散式佇列啟動型作業程序 xxxxCHIN 在 RACF 下執行。
   - 授權存取佇列管理程式資料集。
   - 授權存取將使用佇列管理程式及公用程式之使用者 ID 的資源。
   - 授權存取將使用連結機能清單結構的那些佇列管理程式。
   - 授權存取將使用 Db2 的那些佇列管理程式。
2. 針對 IBM MQ 安全設定 RACF 定義。

3. 如果您想要使用「傳輸層安全 (TLS)」，請準備系統以使用憑證和金鑰。

z/OS 設定 *IBM MQ for z/OS* 資料集安全

IBM MQ 使用者有多種類型。 使用 RACF 來控制其對系統資料集的存取權。

IBM MQ 資料集的可能使用者包括下列實體:

- 佇列管理程式本身。
- 通道起始程式
- 需要建立 IBM MQ 資料集、執行公用程式及類似作業的 IBM MQ 管理者。
- 需要使用 IBM MQ 所提供之記錄定義檔的應用程式設計師，包括資料集、巨集及類似資源。
- 涉及下列一個以上的應用程式:
  - 批次工作
  - TSO 使用者
  - CICS 地區
  - IMS 地區
- 資料集 CSQOUTX 及 CSQSNAP
- 動態佇列 SYSTEM.CSQXCMD.*

對於所有這些潛在使用者，使用 RACF 來保護 IBM MQ 資料集。

您也必須控制對所有 'CSQINP' 資料集的存取權。

z/OS *RACF* 已啟動作業程序的授權

部分 IBM MQ 資料集用於佇列管理程式的專用。 如果您使用 RACF 來保護 IBM MQ 資料集，則也必須使用 RACF 來授權佇列管理程式啟動型作業程序 xxxxMSTR 及分散式佇列啟動型作業程序 xxxxCHIN。 如果要這麼做，請使用 STARTED 類別。 或者，您可以使用已啟動的程序表格 (ICHRIN03)，但您必須先執行 z/OS 系統的 IPL，變更才會生效。

如需相關資訊，請參閱 *z/OS Security Server RACF System Programmer 's Guide*。

所識別的 RACF 使用者 ID 必須具有已啟動作業程序中資料集的必要存取權。 例如，如果您將稱為 CSQ1MSTR 的佇列管理程式啟動作業程序與 RACF 使用者 ID QMGRCSQ1 相關聯，則使用者 ID QMGRCSQ1 必須能夠存取 CSQ1 佇列管理程式所存取的 z/OS 資源。

此外，佇列管理程式使用者 ID 中 GROUP 欄位的內容必須與該佇列管理程式 STARTED 設定檔中 GROUP 欄位的內容相同。 如果每一個 GROUP 欄位中的內容不相符，則會阻止適當的使用者 ID 進入系統。 此狀況會導致以未定義的使用者 ID 執行 IBM MQ，並因此因安全違規而關閉。

與佇列管理程式及通道起始程式啟動作業程序相關聯的 RACF 使用者 ID 不得設定 TRUSTED 屬性。

z/OS 授權存取資料集

IBM MQ 資料集應該受到保護，以便沒有未獲授權的使用者可以執行佇列管理程式實例，或取得任何佇列管理程式資料的存取權。 若要這麼做，請使用一般 z/OS RACF 資料集保護。

第 214 頁的表 65 彙總佇列管理程式啟動型作業程序對不同資料集必須具備的 RACF 存取權。

表 65: *RACF* 存取與佇列管理程式相關聯的資料集	
RACF 存取	**資料集**
READ	• `thlqual.SCSQAUTH` 和 `thlqual.SCSQANLx` (其中 x 是國家語言的語言字母)。 • 在佇列管理程式的已啟動作業程序中， CSQINP1、 CSQINP2 及 CSQXLIB 所參照的資料集。 • 群組中其他佇列管理程式所擁有的 SMDS 資料集。 • 群組中其他佇列管理程式的日誌、BSDS 及保存日誌資料集。
UPDATE	• 所有頁面集及日誌和 BSDS 資料集。 • 佇列管理程式所擁有的 SMDS 資料集 • 群組中其他佇列管理程式所擁有的 SMDS 資料集，適用於佇列管理程式執行 RECOVER CFSTRUCT 指令的結構。
ALTER	• 所有保存日誌資料集。

第 214 頁的表 66 彙總分散式佇列的啟動型作業程序必須對不同資料集具有的 RACF 存取權。

表 66: *RACF* 存取與分散式佇列相關聯的資料集	
RACF 存取	**資料集**
READ	• thlqual.SCSQAUTH、 thlqual.SCSQANLx (其中 x 是國家語言的語言字母) 及 thlqual.SCSQMVR1。 • LE 程式庫資料集。 • 通道起始程式啟動作業程序中 CSQXLIB 及 CSQINPX 所參照的資料集。
UPDATE	• 資料集 CSQOUTX 及 CSQSNAP

如需相關資訊，請參閱 *z/OS Security Server RACF Security Administrator 's Guide*。

z/OS V 9.2.0 加密資料集

IBM MQ 資料集可以使用 z/OS 資料集加密進行加密，以便資料受到保護，或基於法規原因。

您可以使用 z/OS 資料集加密來保護所有頁面集、作用中日誌、保存日誌及引導 (BSDS) 資料集。

小心: 您無法使用 IBM MQ for z/OS 9.1.4 或更早版本的 z/OS 資料集加密來保護共用訊息資料集 (SMDS)。

請參閱 IBM MQ for z/OS 上具有資料集加密之靜態資料的機密性 一節。 的文件以取得相關資訊。

z/OS 設定 *IBM MQ for z/OS* 資源安全

IBM MQ 使用者有多種類型。 使用 RACF 來控制其對 IBM MQ 資源的存取權。

IBM MQ 資源 (例如佇列及通道) 的可能使用者包括下列實體:

- 佇列管理程式本身。
- 通道起始程式
- IBM MQ 管理者，需要建立 IBM MQ 資料集、執行公用程式及類似作業
- 需要使用 IBM MQ 所提供之記錄定義檔的應用程式設計師，包括資料集、巨集及類似資源。
- 涉及下列一個以上的應用程式:
  - 批次工作
  - TSO 使用者

  - CICS 地區
  - IMS 地區
- 資料集 CSQOUTX 及 CSQSNAP
- 動態佇列 SYSTEM.CSQXCMD.*

對於所有這些潛在使用者，使用 RACF 來保護 IBM MQ 資源。 尤其請注意，通道起始程式需要存取各種資源 (如 第 220 頁的『z/OS 上通道起始程式的安全考量』中所述)，因此執行它的使用者 ID 必須獲得授權才能存取這些資源。

如果您使用佇列共用群組，佇列管理程式可能會在內部發出各種指令，因此它所使用的使用者 ID 必須獲得授權，才能發出這類指令。 指令如下:

- 對每一個具有 QSGDISP (GROUP) 的物件進行 DEFINE、ALTER 及 DELETE
- 與 CHLDISP (SHARED) 搭配使用之每個通道的 START 及 STOP CHANNEL

z/OS 配置 *z/OS* 系統以使用 *TLS*

使用本主題作為範例，說明如何使用 RACF 指令來配置 IBM MQ for z/OS with Transport Layer Security (TLS)。

如果您要將 TLS 用於通道安全，則需要在系統上執行一些作業。 (如需針對憑證和金鑰儲存庫 (金鑰環) 使用 RACF 指令的詳細資料，請參閱 在 z/OS 上使用 TLS。)

1. 在 RACF 中建立金鑰環，以使用 RACF RACDCERT 指令來保留系統的所有金鑰和憑證。 例如:

   ```
   RACDCERT ID(CHINUSER) ADDRING(QM1RING)
   ```

   ID 必須是通道起始程式位址空間使用者 ID，或是您想要擁有金鑰環的使用者 ID (如果它是共用金鑰環的話)。
2. 使用 RACF RACDCERT 指令為每一個佇列管理程式建立數位憑證。

   憑證的標籤必須是 IBM MQ **CERTLABL** 屬性的值 (如果已設定的話)，或預設 `ibmWebSphereMQ` 並附加佇列管理程式或佇列共用群組的名稱。 如需詳細資料，請參閱 數位憑證標籤。 在此範例中為 `ibmWebSphereMQQM1`。

   例如:

   ```
   RACDCERT ID(USERID) GENCERT
   SUBJECTSDN(CN('username') O('IBM') OU('departmentname') C('England'))
   WITHLABEL('ibmWebSphereMQQM1')
   ```
3. 使用 RACF RACDCERT 指令，將 RACF 中的憑證連接至金鑰環。 例如:

   ```
   RACDCERT CONNECT(ID(USERID) LABEL('ibmWebSphereMQQM1') RING(QM1RING))
   CONNECT ID(CHINUSER)
   ```

   您也需要將任何相關簽章者憑證 (從憑證管理中心) 連接至金鑰環。 亦即，此佇列管理程式的 TLS 憑證的所有憑證管理中心，以及與此佇列管理程式通訊之所有 TLS 憑證的所有憑證管理中心。 例如:

   ```
   RACDCERT ID(CHINUSER)
   CONNECT(CERTAUTH LABEL('My CA') RING(QM1RING) USAGE(CERTAUTH))
   ```
4. 在每一個佇列管理程式上，使用 IBM MQ ALTER QMGR 指令來指定佇列管理程式需要指向的金鑰儲存庫。 比方說，如果金鑰環是由通道起始程式位址空間所擁有:

```
ALTER QMGR SSLKEYR(QM1RING)
```

或者如果您使用共用金鑰環:

```
ALTER QMGR SSLKEYR(userid/QM1RING)
```

其中 *userid* 是擁有共用金鑰環的使用者 ID。

5. 「憑證撤銷清冊 (CRL)」可讓憑證管理中心撤銷不再信任的憑證。 CRL 儲存在 LDAP 伺服器中。 若要在 LDAP 伺服器上存取此清單，您首先需要使用 IBM MQ DEFINE AUTHINFO 指令建立 AUTHTYPE CRLLDAP 的 AUTHINFO 物件。 例如:

```
DEFINE AUTHINFO(LDAP1)
AUTHTYPE(CRLLDAP)
CONNAME(ldap.server(389))
LDAPUSER('')
LDAPPWD('')
```

在此範例中，憑證撤銷清冊儲存在 LDAP 伺服器的公用區域中，因此不需要 LDAPUSER 和 LDAPPWD 欄位。

接下來，使用 IBM MQ DEFINE NAMELIST 指令，將 AUTHINFO 物件放入名單中。 例如:

```
DEFINE NAMELIST(LDAPNL) NAMES(LDAP1)
```

最後，使用 IBM MQ ALTER QMGR 指令，將名稱清單與每一個佇列管理程式相關聯。 例如:

```
ALTER QMGR SSLCRLNL(LDAPNL)
```

6. 使用 IBM MQ ALTER QMGR 指令，設定佇列管理程式以執行 TLS 呼叫。 這會定義僅處理 SSL 呼叫的伺服器子作業，這會讓一般分派器繼續正常處理，而不受任何 SSL 呼叫影響。 您必須至少有兩個子作業。 例如:

```
ALTER QMGR SSLTASKS(8)
```

只有在重新啟動通道起始程式時，此變更才會生效。

7. 使用 IBM MQ DEFINE CHANNEL 或 ALTER CHANNEL 指令，指定要用於每一個通道的密碼規格。 例如:

```
ALTER CHANNEL(LDAPCHL)
CHLTYPE(SDR)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
```

通道兩端必須指定相同的密碼規格。

z/OS 管理 QSG 中的通道鑑別記錄

通道鑑別記錄適用於建立它們的佇列管理程式，不會在整個佇列共用群組 (QSG) 中共用它們。 因此，如果佇列共用群組中的所有佇列管理程式都需要具有相同的規則，則需要執行部分管理，以保持所有規則一致。

1. 一律將 CMDSCOPE(*) 選項新增至所有 SET CHLAUTH 指令。 這會將指令傳送至佇列共用群組中所有執行中的佇列管理程式

2. 搭配使用 DISPLAY CHLAUTH 指令與 CMDSCOPE(*) 選項，然後分析回應，以查看所有佇列管理程式中的記錄是否相同。 當發現不一致時，可以發出包含 CMDSCOPE(*) 或 CMDSCOPE(*qmgr-name*)之相同規則的 SET CHLAUTH 指令。
3. 將成員新增至具有完整規則集的佇列管理程式 CSQINP2 連結 (如需詳細資料，請參閱 起始設定指令)。 在佇列管理程式的起始設定程序中，會讀取這些項目。 如果 SET CHLAUTH 指令使用 ACTION(ADD)，則只有在規則不存在時才會新增規則。 使用 ACTION(REPLACE) 將取代現有規則 (如果已存在) 或新增現有規則 (如果未存在)。 然後，相同的成員可以放置在佇列共用群組中所有佇列管理程式的 CSQINP2 連結中。
4. 使用 CSQUTIL 公用程式 (如需詳細資料，請參閱 發出指令至 IBM MQ (COMMAND))，以使用 MAKEDEF 或 MAKEREP 選項從一個佇列管理程式擷取規則。 然後使用 CSQUTIL 將輸出重播至目標佇列管理程式。

相關概念

通道鑑別記錄

若要在通道層次對授與連接系統的存取權進行更精確的控制，您可以使用通道鑑別記錄。

z/OS z/OS 上的審核考量

一般 RACF 審核控制可用於處理佇列管理程式的安全審核。 IBM MQ 不會收集自己的任何安全統計資料。 唯一的統計資料是可以透過審核建立的統計資料。

RACF 審核可以根據:

- 使用者 ID
- 資源類別
- 設定檔

如需詳細資料，請參閱 *z/OS Security Server RACF* 審核員手冊。

註: 審核會降低效能; 您實作的審核越多，效能越降低。 這也是使用 RACF WARNING 選項的考量。

z/OS 審核 *RESLEVEL*

請使用 RESAUDIT 系統參數來控制 RESLEVEL 審核記錄的產生。 RACF 會產生一般審核記錄。

將 RESAUDIT 系統參數設為 YES，以產生 RESLEVEL 審核記錄。 如果 RESAUDIT 參數設為 NO，則不會產生審核記錄。 如需設定此參數的詳細資料，請參閱 使用 CSQ6SYSP。

如果 RESAUDIT 設為 YES，則在執行 RESLEVEL 檢查時，不會取得一般 RACF 審核記錄，以查看位址空間使用者 ID 對 hlq.RESLEVEL 設定檔具有哪些存取權。 相反地， IBM MQ 會要求 RACF 建立 GENERAL 審核記錄 (事件號碼 27)。 這些檢查僅在連接時執行，因此效能成本是最小的。

小心: RACFRW 不再是建議用來處理 RACF 審核記錄的公用程式。 您應該使用 RACF SMF 資料卸載公用程式，因為這是偏好的報告方法。

您可以使用 RACF 報告寫出器 (RACFRW) 來報告 IBM MQ 一般審核記錄。 您可以使用下列 RACFRW 指令來報告 RESLEVEL 存取權:

```
RACFRW
SELECT PROCESS
EVENT GENERAL
LIST
END
```

第 218 頁的圖 19 中顯示 RACFRW 的範例報告 (不含 *Date*、 *Time* 和 *SYSID* 欄位)。

```
       RACF REPORT - LISTING OF PROCESS RECORDS                                  PAGE   4
                                 E
                                 V  Q
                                 E  U
*JOB/USER *STEP/  --TERMINAL--   N  A
   NAME    GROUP     ID    LVL  T  L

 WS21B    MQMGRP IGJZM000   0   27 0  JOBID=(WS21B 05.111 09:44:57),USERDATA=()
    TRUSTED USER                      AUTH=(NONE),REASON=(NONE)
                                  SESSION=TSOLOGON,TERMINAL=IGJZM000,
                                  LOGSTR='CSQH RESLEVEL CHECK PERFORMED AGAINST
PROFILE(QM66.RESLEVEL),
                                  CLASS(MQADMIN), ACCESS EQUATES TO
(CONTROL)',RESULT=SUCCESS,MQADMIN
```

圖 *19: RACFRW* 的範例輸出顯示 *RESLEVEL* 一般審核記錄

從檢查此範例輸出中的 LOGSTR 資料，您可以看到 TSO 使用者 WS21B 具有 QM66.RESLEVEL。 這表示當使用者 WS21B 存取 QM66 資源時，會略過所有資源安全檢查。

如需使用 RACFRW 的相關資訊，請參閱 *z/OS Security Server RACF* 審核員手冊。

z/OS 自訂安全

如果您想要變更 IBM MQ 安全的運作方式，您必須透過 SAF 結束程式 (ICHRFR00) 或外部安全管理程式中的結束程式來執行。

若要進一步瞭解 RACF 結束程式，請參閱 *z/OS Security Server RACROUTE Macro Reference* 手冊。

註: 因為 IBM MQ 會最佳化對 ESM 的呼叫，所以可能不會對 (例如，特定使用者針對特定佇列的每次開啟) 提出 RACROUTE 要求。

z/OS z/OS 上的安全違規訊息

應用程式中的回覆碼 MQRC_NOT_AUTHORIZED 或工作日誌中的訊息指出安全違規。

由於下列原因， MQRC_NOT_AUTHORIZED 回覆碼可以傳回給應用程式:

- 不容許使用者連接至佇列管理程式。 在此情況下，您會在 Batch/TSO、 CICS 或 IMS 工作日誌中取得 ICH408I 訊息。
- 使用者登入佇列管理程式失敗，例如，工作使用者 ID 無效或適當，或作業使用者 ID 或替代使用者 ID 無效。 其中一個以上使用者 ID 可能無效，因為它們已被撤銷或刪除。 在此情況下，您會在佇列管理程式工作日誌中取得 ICHxxxx 訊息及可能的 IRRxxxx 訊息，以提供登入失敗的原因。 例如:

```
ICH408I USER(NOTDFND ) GROUP(        ) NAME(???                 )
  LOGON/JOB INITIATION - USER AT TERMINAL          NOT RACF-DEFINED
IRR012I  VERIFICATION FAILED. USER PROFILE NOT FOUND
```

- 已要求替代使用者，但工作或作業使用者 ID 沒有替代使用者 ID 的存取權。 針對此失敗，您會在相關佇列管理程式的工作日誌中收到違規訊息。
- 環境定義選項已使用或隱含在開啟輸出的傳輸佇列中，但工作使用者 ID 或作業或替代使用者 ID (如果適用的話) 沒有環境定義選項的存取權。 在此情況下，會將違規訊息放置在相關佇列管理程式的工作日誌中。
- 未獲授權的使用者已嘗試存取安全佇列管理程式物件 (例如，佇列)。 在此情況下，會將違規的 ICH408I 訊息放置在相關佇列管理程式的工作日誌中。 此違規可能是由於工作或作業或替代使用者 ID (如果適用的話)。

在佇列管理程式的工作日誌中，也可以找到指令安全及指令資源安全的違規訊息。

如果 ICH408I 違規訊息顯示佇列管理程式工作名稱而非使用者 ID ，這通常是指定空白替代使用者 ID 的結果。 例如:

```
ICH408I JOB(MQS1MSTR) STEP(MQS1MSTR)
          MQS1.PAYROLL.REQUEST CL(MQQUEUE)
          INSUFFICIENT ACCESS AUTHORITY
          ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
```

您可以檢查 MQADMIN 設定檔 hlq.ALTERNATE.USER.-BLANK-。

也可以透過下列方式產生 ICH408I 違規訊息:

- 傳送至系統指令輸入佇列的指令，不含環境定義。 寫入系統指令輸入佇列的使用者撰寫程式應該一律使用環境定義選項。 如需相關資訊，請參閱 第 183 頁的『環境定義安全的設定檔』。
- 當存取 IBM MQ 資源的工作沒有相關聯的使用者 ID 時，或當 IBM MQ 配接器無法從配接器環境擷取使用者 ID 時。

如果您同時使用佇列共用群組和佇列管理程式層次安全，也可能會發出違規訊息。 您可能會收到訊息，指出在佇列管理程式層次找不到任何設定檔，但由於佇列共用群組層次設定檔，仍被授與存取權。

```
ICH408I JOB(MQS1MSTR) STEP(MQS1MSTR)
          MQS1.PAYROLL.REQUEST CL(MQQUEUE)
          PROFILE NOT FOUND - REQUIRED FOR AUTHORITY CHECKING
          ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
```

z/OS 容許或不正確禁止存取時要執行的動作

除了 *z/OS Security Server RACF Security Administrator 's Guide* 中詳述的步驟之外，如果對資源的存取權似乎未正確控制，請使用此核對清單。

- 是否已正確設定交換器設定檔?
  - RACF 是否處於作用中?
  - IBM MQ RACF 類別是否已安裝且在作用中?

    請使用 RACF 指令 SETROPTS LIST 來檢查此項。
  - 請使用 IBM MQ DISPLAY SECURITY 指令來顯示佇列管理程式中的現行交換器狀態。
  - 請檢查 MQADMIN 類別中的交換器設定檔。

    為此，請使用 RACF 指令 SEARCH 及 RLIST。
  - 發出 IBM MQ REFRESH SECURITY (MQADMIN) 指令來重新檢查 RACF 交換器設定檔。
- RACF 資源設定檔是否已變更? 例如，設定檔的通用存取權是否已變更，或設定檔的存取清單是否已變更?
  - 設定檔是通用的嗎?

    如果是，請發出 RACF 指令 SETROPTS GENERIC (classname) 重新整理。
  - 您已重新整理此佇列管理程式上的安全嗎?

    必要的話，請發出 RACF 指令 SETROPTS RACLIST (classname) 重新整理。

    必要的話，請發出 IBM MQ REFRESH SECURITY (*) 指令。
- 使用者的 RACF 定義是否已變更? 例如，使用者是否已連接至新群組，或使用者存取權是否已撤銷?
  - 您是否已發出 IBM MQ RVERIFY SECURITY (userid) 指令來重新驗證使用者?
- 由於 RESLEVEL 而略過安全檢查嗎?
  - 請檢查連接使用者 ID 對 RESLEVEL 設定檔的存取權。 請使用 RACF 審核記錄來判斷 RESLEVEL 的設定。
  - 對於通道，請記住通道起始程式使用者 ID 對 RESLEVEL 的存取層次是由所有通道所繼承，因此導致略過所有檢查的存取層次 (例如 ALTER) 會導致略過所有通道的安全檢查。
  - 如果您是從 CICS 執行，請檢查交易的 RESSEC 設定。

- 如果在連接使用者時已變更 RESLEVEL，則必須先中斷連線並重新連接，新的 RESLEVEL 設定才會生效。

- 您正在使用佇列共用群組嗎?
  - 如果您同時使用佇列共用群組和佇列管理程式層次安全，請檢查您是否已定義所有正確的設定檔。 如果未定義佇列管理程式設定檔，則會將一則訊息傳送至日誌，指出找不到設定檔。
  - 您是否使用了無效的交換器設定組合，以便將完整安全檢查設定為開啟?
  - 您需要定義安全切換參數來置換佇列管理程式的部分佇列共用群組設定嗎?
  - 佇列管理程式層次設定檔是否優先於佇列共用群組層次設定檔?

z/OS z/OS 上通道起始程式的安全考量

如果您在分散式佇列環境中使用資源安全，則通道起始程式位址空間需要對各種 IBM MQ 資源的適當存取權。 您可以使用「整合 Cryptographic Support 機能 (ICSF)」來植入密碼保護演算法。

使用資源安全

如果您使用資源安全，如果您使用分散式佇列，請考量下列要點:

系統佇列
通道起始程式位址空間需要對 第 174 頁的『系統佇列安全』所列出的系統佇列，以及所有使用者目的地佇列和無法傳送郵件的佇列的 RACF UPDATE 存取權 (但請參閱 第 172 頁的『無法傳送郵件的佇列安全』)。

傳輸佇列
通道起始程式位址空間需要對所有使用者傳輸佇列的 ALTER 存取權。

環境定義安全 (context security)
通道使用者 ID (以及 MCA 使用者 ID，如果已指定的話) 需要 MQADMIN 類別中 hlq.CONTEXT.queuename 設定檔的 RACF CONTROL 存取權。 視 RESLEVEL 設定檔而定，通道使用者 ID 也可能需要這些設定檔的 CONTROL 存取權。

所有通道都需要 MQADMIN hlq.CONTEXT 的 CONTROL 存取權。 無法傳送郵件的佇列設定檔。 所有通道 (不論是起始或回應) 都可以產生報告，因此它們需要 hlq.CONTEXT.reply-q 設定檔的 CONTROL 存取權。

SENDER、CLUSSDR 及 SERVER 通道需要 hlq.CONTEXT.xmit-queue-name 設定檔的 CONTROL 存取權，因為訊息可以放入傳輸佇列中，以喚醒通道循序結束。

註: 如果通道使用者 ID 或通道使用者 ID 所連接的 RACF 群組具有 hlq.RESLEVEL 的 CONTROL 或 ALTER 存取權，則不會對通道起始程式或其任何通道進行資源檢查。

如需相關資訊，請參閱 第 183 頁的『環境定義安全的設定檔』 第 199 頁的『RESLEVEL 及通道起始程式連線』 和 第 200 頁的『z/OS 上用於安全檢查的使用者 ID』。

CSQINPX
如果您使用 CSQINPX 輸入資料集，通道起始程式也需要 CSQINPX 的 READ 存取權，以及資料集 CSQOUTX 和動態佇列 SYSTEM.CSQXCMD. *。

連線安全
通道起始程式位址空間連線要求使用必須設定適當存取安全的 CHIN 連線類型，請參閱 第 168 頁的『通道起始程式的連線安全設定檔』。

資料集
通道起始程式位址空間需要適當存取佇列管理程式資料集，請參閱 第 213 頁的『授權存取資料集』。

指令
分散式佇列作業指令 (例如， DEFINE CHANNEL、START CHINIT、START LISTENER 及其他通道指令) 必須已設定適當的指令安全，請參閱 第 185 頁的表 49。

如果您使用佇列共用群組，通道起始程式可能會在內部發出各種指令，因此它所使用的使用者 ID 必須獲得授權，才能發出這類指令。 這些指令是與 CHLDISP (SHARED) 搭配使用之每個通道的 START 及 STOP CHANNEL。

如果佇列管理程式的 PSMODE 不是 DISABLED，則通道起始程式必須具有 DISPLAY PUBSUB 指令的 READ 存取權。

通道安全性

通道 (特別是接收端及伺服器連線) 需要設定適當的安全; 如需相關資訊，請參閱 第 200 頁的『z/OS 上用於安全檢查的使用者 ID』。

您也可以使用「傳輸層安全 (TLS)」通訊協定來提供通道安全。 如需搭配使用 TLS 與 IBM MQ 的相關資訊，請參閱 第 20 頁的『IBM MQ 中的 TLS 安全通訊協定』。

另請參閱 第 83 頁的『用戶端的存取控制』，以取得伺服器連線安全的相關資訊。

使用者 ID

第 203 頁的『通道起始程式使用的使用者 ID』 和 第 207 頁的『內部群組佇列作業代理程式使用的使用者 ID』 中說明的使用者 ID 需要下列存取權:

- RACF 對適當目的地佇列及無法傳送郵件之佇列的 UPDATE 存取權
- RACF 如果在接收端執行環境定義檢查，則對 `hlq.CONTEXT.queuename` 設定檔的 CONTROL 存取權
- 對 hlq.ALTERNATE.USER.userid 設定檔。
- 對於用戶端，這是要使用之資源的適當 RACF 存取權。

APPC 安全

如果您使用 LU 6.2 傳輸通訊協定，請設定適當的 APPC 安全。 (例如，使用 APPCLU RACF 類別。) 如需設定 APPC 安全的相關資訊，請參閱下列手冊:

- *z/OS V1R2.0 MVS* 規劃*: APPC* 管理
- *Multiplatform APPC Configuration Guide*， IBM Redbooks 出版品

出埠傳輸使用 "安全 (SAME)" APPC 選項。 因此，通道起始程式位址空間及其預設設定檔 (RACF GROUP) 的使用者 ID 會透過網路傳送至接收端，並具有已驗證使用者 ID 的指示器 (ALREADYV)。

如果接收端也是 z/OS， APPC 會驗證使用者 ID 和設定檔，並將使用者 ID 呈現給接收端通道，作為通道使用者 ID。

在佇列管理程式使用 APPC 與相同或另一個 z/OS 系統上的另一個佇列管理程式進行通訊的環境中，您需要確保:

- 通訊 LU 的 VTAM 定義指定 SETACPT (ALREADYV)
- 對於指定 CONVSEC (ALREADYV) 的 LU 之間的連線，有一個 RACF APPCLU 設定檔

變更安全設定

如果通道使用者 ID 或 MCA 使用者 ID 對目的地佇列具有的 RACF 存取層次已變更，則此變更僅對目的地佇列的新物件控點 (即新的 MQOPEN) 生效。 MCA 開啟及關閉佇列的次數是可變的; 如果在進行這類存取變更時通道已在執行中，則 MCA 可以使用使用者 ID 的現有安全存取權，而非更新的安全存取權，繼續將訊息放置在目的地佇列上。 停止並重新啟動通道以施行更新的存取層次可避免此情況。

自動重新啟動

如果您使用 z/OS Automatic Restart Manager (ARM) 來重新啟動通道起始程式，則必須授權與 XCFAS 位址空間相關聯的使用者 ID，才能發出 IBM MQ START CHINIT 指令。

使用「整合加密服務機能 (ICSF)」

如果未使用 TLS，當植入密碼保護演算法來模糊化流經用戶端通道的密碼時，通道起始程式可以使用 ICSF 來產生亂數。 產生亂數的程序稱為 熵。

如果您已安裝 z/OS 特性，但尚未啟動 ICSF，則會看到訊息 CSQX213E，且通道起始程式會將 STCK 用於熵。

訊息 CSQX213E 警告您密碼保護演算法未盡可能安全。 不過，您可以繼續處理程序; 對執行時期沒有其他影響。

如果您未安裝 z/OS 特性，通道起始程式會自動使用 STCK。

附註:

1. 將 ICSF 用於熵會產生比使用 STCK 更多的隨機序列。
2. 如果您啟動 ICSF，則必須重新啟動通道起始程式。
3. 特定 CipherSpecs 需要 ICSF。 如果您嘗試使用其中一個 CipherSpecs，但未安裝 ICSF，則會收到訊息 CSQX629E。

z/OS z/OS 上佇列管理程式叢集中的安全

叢集的安全考量與非叢集佇列管理程式及通道的安全考量相同。 通道起始程式需要存取部分其他系統佇列，而部分其他指令則需要適當的安全集。

您可以使用 MCA 使用者 ID、通道鑑別記錄、TLS 及安全結束程式來鑑別叢集通道 (與傳統通道一樣)。 與叢集接收端通道相關的通道鑑別記錄或安全結束程式必須檢查是否允許遠端佇列管理程式存取伺服器佇列管理程式的叢集佇列。 您可以在不變更現有佇列存取安全的情況下開始使用 IBM MQ 叢集支援。 不過，您必須容許叢集中的其他佇列管理程式寫入 SYSTEM.CLUSTER.COMMAND.QUEUE 。

IBM MQ 叢集支援不提供將叢集成員限制為僅用戶端角色的機制。 因此，您必須確定信任任何容許進入叢集的佇列管理程式。 如果叢集中的任何佇列管理程式建立具有特定名稱的佇列，它可以接收該佇列的訊息，不論將訊息放入該佇列的應用程式是否預期如此。

若要限制叢集的成員資格，請採取您為防止佇列管理程式連接至接收端通道所採取的相同動作。 您可以使用通道鑑別記錄或在接收端通道上寫入安全結束程式，來限制叢集的成員資格。 您也可以撰寫結束程式，以防止未獲授權的佇列管理程式寫入 SYSTEM.CLUSTER.COMMAND.QUEUE。

註: 不建議允許應用程式開啟 SYSTEM.CLUSTER.TRANSMIT.QUEUE 。 也不建議允許應用程式直接開啟任何其他傳輸佇列。

如果您使用資源安全，除了 第 220 頁的『z/OS 上通道起始程式的安全考量』中包含的考量之外，請考量下列要點:

系統佇列

通道起始程式需要下列系統佇列的 RACF ALTER 存取權:

- SYSTEM.CLUSTER.COMMAND QUEUE
- SYSTEM.CLUSTER.TRANSMIT.QUEUE.

以及對 SYSTEM.CLUSTER.REPOSITORY.QUEUE

對於用於叢集作業的任何名稱清單，它也需要 READ 存取權。

指令

設定適當的指令安全 (如 第 185 頁的表 49 中所述) 適用於叢集支援指令 (REFRESH 和 RESET CLUSTER、SUSPEND 和 RESUME QMGR)。

z/OS 搭配使用 IBM MQ 與 CICS 時的安全考量

IBM MQ 9.0.0 以及更新版本所支援的所有 CICS 版本，都使用 CICS 提供的配接器及橋接器版本。

如需安全考量的詳細資料，請參閱:

- CICS-MQ 配接器的安全。
- CICS 的安全-MQ 橋接器。

z/OS 搭配使用 IBM MQ 與 IMS 時的安全考量

當您搭配使用 IBM MQ 與 IMS 時，請使用本主題來規劃您的安全需求。

使用 OPERCMDS 類別

如果您使用 RACF 來保護 OPERCMDS 類別中的資源，請確定與 IBM MQ 佇列管理程式位址空間相關聯的使用者 ID 有權向它可以連接的任何 IMS 系統發出 MODIFY 指令。

IMS 橋接器的安全考量

在決定 IMS 橋接器的安全需求時，您必須考量四個層面:

- 將 IBM MQ 連接至 IMS 所需的安全授權
- 在使用橋接器來存取 IMS 的應用程式上執行的安全檢查量
- 容許這些應用程式使用哪些 IMS 資源
- 橋接器放置及取得的訊息要使用的權限

當您定義 IMS 橋接器的安全需求時，必須考量下列事項:

- 透過橋接器傳遞的訊息可能源自未提供強大安全特性之平台上的應用程式
- 透過橋接器傳遞的訊息可能源自不受相同企業或組織控制的應用程式

z/OS 連接至 *IMS* 時的安全考量

授與 IBM MQ 佇列管理程式位址空間的使用者 ID 對 OTMA 群組的存取權。

IMS 橋接器是 OTMA 用戶端。 IMS 的連線會以 IBM MQ 佇列管理程式位址空間的使用者 ID 來運作。 這通常定義為已啟動作業群組的成員。 必須授與此使用者 ID 對 OTMA 群組的存取權 (除非 /SECURE OTMA 設定為 NONE)。

如果要這麼做，請在 FACILITY 類別中定義下列設定檔:

```
IMSXCF.xcfgname.mqxcfmname
```

其中 `xcfgname` 是 XCF 群組名稱， `mqxcfmname` 是 IBM MQ 的 XCF 成員名稱。

您必須提供此設定檔的讀取權給 IBM MQ 佇列管理程式使用者 ID。

註:

1. 如果您變更 FACILITY 類別中的權限，您必須發出 RACF 指令 SETROPTS RACLIST (FACILITY) REFRESH 來啟動變更。
2. 如果設定檔 hlq.NO.SUBSYS.SECURITY 存在於 MQADMIN 類別中，除非 /SECURE OTMA 設定為 NONE ，否則不會將任何使用者 ID 傳遞至 IMS ，且連線會失敗。

z/OS *IMS* 橋接器的應用程式存取控制

在每個 IMS 系統的 FACILITY 類別中定義 RACF 設定檔。 授與適當層次的存取權給 IBM MQ 佇列管理程式使用者 ID。

對於 IMS 橋接器所連接的每一個 IMS 系統，您可以在 FACILITY 類別中定義下列 RACF 設定檔，以決定要對傳給 IMS 系統的每一則訊息執行多少安全檢查。

```
IMSXCF.xcfgname.imsxcfmname
```

其中 `xcfgname` 是 XCF 群組名稱，而 `imsxcfmname` 是 IMS 的 XCF 成員名稱。 (您需要為每一個 IMS 系統定義個別設定檔。)

當 IMS 橋接器連接至 IMS 時，此設定檔中 IBM MQ 佇列管理程式使用者 ID 所容許的存取層次會傳回給 IBM MQ ，並指出後續交易所需的安全層次。 對於後續交易， IBM MQ 會向 RACF 要求適當的服務，並在使用者 ID 已獲授權的情況下，將訊息傳遞至 IMS。

OTMA 不支援 IMS /SIGN 指令; 不過， IBM MQ 可讓您設定每一則訊息的存取權檢查，以便能夠實作必要的控制層次。

可以傳回下列存取層次資訊:

無或找不到設定檔
這些值指出需要最大安全，亦即，每個交易都需要鑑別。 進行檢查以驗證 MQMD 結構的 *UserIdentifier* 欄位中指定的使用者 ID ，以及 MQIIH 結構的 *Authenticator* 欄位中的密碼或 PassTicket 是否為 RACF 已知且有效的組合。 UTOKEN 使用密碼或 PassTicket,並傳遞給 IMS ； UTOKEN 不會被緩存。

註: 如果 MQADMIN 類別中存在設定檔 hlq.NO.SUBSYS.SECURITY，則此安全層次會置換設定檔中定義的任何層次。

READ
此值指出在下列情況下要執行與 NONE 相同的鑑別:

- 第一次發現特定使用者 ID 時
- 之前已發現使用者 ID，但未使用密碼或 PassTicket 建立快取的 UTOKEN

IBM MQ 會在必要時要求 UTOKEN，並將它傳遞至 IMS。

註: 如果已處理重新驗證安全的要求，則會遺失所有快取資訊，並在稍後第一次發現每一個使用者 ID 時要求 UTOKEN。

UPDATE
會檢查 MQMD 結構的 *UserIdentifier* 欄位中的使用者 ID 是否為 RACF 已知。

UTOKEN 已建置並傳遞至 IMS；會快取 UTOKEN。

CONTROL/ALTER
這些值指出不需要針對此 IMS 系統的任何使用者 ID 提供安全 UTOKENs。 (您可能只會將此選項用於開發及測試系統。)

小心: 請注意，仍會針對 **CONTROL/ALTER** 傳遞 MQMD 結構的 *UserIdentifier* 欄位中包含的使用者 ID。

註:

1. 此存取權是在 IBM MQ 連接至 IMS 時定義，並在連線期間持續。 如果要變更安全層次，必須變更安全設定檔的存取權，然後停止並重新啟動橋接器 (例如，停止並重新啟動 OTMA)。
2. 如果您變更 FACILITY 類別中的權限，您必須發出 RACF 指令 SETROPTS RACLIST (FACILITY) REFRESH 來啟動變更。
3. 您可以使用密碼或 PassTicket,但必須記住 IMS 橋不會加密資料。 有關使用 PassTickets,請參閱第 225 頁的『在 IMS 標頭中使用 RACF PassTickets』。
4. 其中部分結果可能會受到 IMS 中使用 /SECURE OTMA 指令的安全設定的影響。
5. 在 IBM MQ ALTER SECURITY 指令的 INTERVAL 和 TIMEOUT 參數所定義的期間，會保留快取的 UTOKEN 資訊。
6. RACF WARNING 選項對 IMSXCF.xcfgname.imsxcfmname 設定檔沒有影響。 使用它不會影響所授與的存取層次，且不會產生任何 RACF WARNING 訊息。

z/OS *IMS* 上的安全檢查

通過橋接器的訊息包含安全資訊。 所進行的安全檢查取決於 IMS 指令 /SECURE OTMA 的設定。

每一個通過橋接器的 IBM MQ 訊息都包含下列安全資訊:

- MQMD 結構的 *UserIdentifier* 欄位中包含的使用者 ID
- MQIIH 結構的 *SecurityScope* 欄位中包含的安全範圍 (如果 MQIIH 結構存在的話)
- UTOKEN (除非 IBM MQ 子系統對相關 `IMSXCF.xcfgname.imsxcfmname` 設定檔具有 CONTROL 或 ALTER 存取權)

所進行的安全檢查取決於 IMS 指令 /SECURE OTMA 的設定，如下所示:

/SECURE OTMA NONE
不會對交易進行安全檢查。

/SECURE OTMA CHECK
MQMD 結構的 *UserIdentifier* 欄位會傳遞至 IMS，以進行交易或指令權限檢查。

ACEE (Accessor Environment Element) 建置在 IMS 控制區域中。

/secure OTMA FULL
MQMD 結構的 *UserIdentifier* 欄位會傳遞至 IMS，以進行交易或指令權限檢查。

ACEE 建置在 IMS 相依區域及 IMS 控制區域中。

/secure OTMA 設定檔
MQMD 結構的 *UserIdentifier* 欄位會傳遞至 IMS，以進行交易或指令權限檢查

MQIIH 結構中的 *SecurityScope* 欄位用來決定是否在 IMS 相依區域及控制區域中建置 ACEE。

註:

1. 如果您變更 TIMS 或 CIMS 類別或相關聯群組類別 GIMS 或 DIMS 中的權限，則必須發出下列 IMS 指令來啟動變更:
   - /MODIFY PREPARE RACF
   - /XX_ENCODE_CASE_CAPS_LOCK_ON modify commit
2. 如果您不使用 /SECURE OTMA PROFILE，則會忽略 MQIIH 結構的 *SecurityScope* 欄位中指定的任何值。

z/OS *IMS* 橋接器完成的安全檢查

視所執行的動作而定，會使用不同的權限。

當橋接器放置或取得訊息時，會使用下列權限:

從橋接器佇列取得訊息
不執行任何安全檢查。

放置異常狀況或 COA 報告訊息
在 MQMD 結構的 *UserIdentifier* 欄位中使用使用者 ID 的權限。

放置回覆訊息
使用原始訊息 MQMD 結構之 *UserIdentifier* 欄位中的使用者 ID 權限

將訊息放入無法傳送郵件的佇列
不執行任何安全檢查。

註:

1. 如果您變更 IBM MQ 類別設定檔，則必須發出 IBM MQ REFRESH SECURITY (*) 指令來啟動變更。
2. 如果您變更使用者的權限，則必須發出 MQSC RVERIFY SECURITY 指令來啟動變更。

z/OS 在 *IMS* 標頭中使用 *RACF PassTickets*

您可以使用 PassTicket 來取代 IMS 標頭中的密碼。

如果您想要在 IMS 標頭 (MQIIH) 中使用 PassTicket 而非密碼，請在訊息要遞送至之 IMS 橋接器佇列的 STGCLASS 定義的 PASSTKTA 屬性中，指定用來驗證 PassTicket 的應用程式名稱。

如果 PASSTKTA 值保留空白，則您必須安排產生 PassTicket。 在此情況下，應用程式名稱的格式必須是 MVSxxxx，其中 xxxx 是執行目標佇列管理程式之 z/OS 系統的 SMFID。

PassTicket 是從使用者 ID、目標應用程式名稱及秘密金鑰建置而成。 它是包含大寫英文字母及數值字元的 8 位元組值。 它只能使用一次，有效期間為 20 分鐘。 如果 PassTicket 是由本端 RACF 系統所產生，則 RACF 只會檢查設定檔是否存在，而不會檢查使用者對設定檔是否具有權限。 如果在遠端系統上產生 PassTicket， RACF 會驗證使用者 ID 對設定檔的存取權。 如需 PassTickets 的完整資訊，請參閱 *z/OS SecureWay Security Server RACF Security Administrator 's Guide*。

IMS 標頭中的 PassTickets 是由 IBM MQ 提供給 RACF，而不是 IMS。

z/OS 將 z/OS 佇列管理程式移轉至大小寫混合格式安全

請遵循下列步驟，將佇列管理程式移轉至大小寫混合的安全。 您可以檢閱所使用的安全產品層次，並啟動新的 IBM MQ 外部安全管理程式類別。 執行 **REFRESH SECURITY** 指令，以啟動大小寫混合格式的設定檔。

開始之前

1. 確定所有 IBM MQ 外部安全管理程式類別都已啟動。
2. 請確定您的佇列管理程式已啟動。

關於這項作業

請遵循下列步驟，將佇列管理程式轉換成大小寫混合格式的安全。

程序

1. 將所有現有的設定檔和存取層次從大寫類別複製到相等的大小寫混合格式外部安全管理程式類別。
   a) MQADMIN 至 MXADMIN。
   b) MQPROC 至 MXPROC。
   c) MQNLIST 至 MXNLIST。
   d) MQQUEUE 至 MXQUEUE。
2. 發出下列指令，將 SCYCASE 佇列管理程式屬性值變更為 MIXED 。

```
ALTER QMGR SCYCASE(MIXED)
```

3. 發出下列指令來啟動安全設定檔。

```
REFRESH SECURITY(*) TYPE(CLASSES)
```

4. 測試安全設定檔是否正常運作。

下一步

請檢閱您的物件定義，並視需要使用 **REFRESH SECURITY** 指令來啟動設定檔，以建立新的大小寫混合格式設定檔。

設定 IBM MQ MQI client 安全

您必須考量 IBM MQ MQI client 安全，讓用戶端應用程式無法無限制存取伺服器上的資源。

執行用戶端應用程式時，請不要使用存取權超過必要權限的使用者 ID 來執行應用程式; 例如， mqm 群組中的使用者，甚至 mqm 使用者本身。

透過以具有太多存取權的使用者身分執行應用程式，您會面臨應用程式存取及變更佇列管理程式組件的風險 (意外或惡意)。

用戶端應用程式與其佇列管理程式伺服器之間的安全有兩個層面: 鑑別和存取控制。

- 鑑別可用來確保以特定使用者身分執行的用戶端應用程式是他們所稱的使用者。 透過使用鑑別，您可以防止攻擊者假冒您的其中一個應用程式來取得佇列管理程式的存取權。

  從 IBM MQ 8.0 開始，鑑別由下列兩個選項之一提供:

  – 連線鑑別特性。

    如需連線鑑別的相關資訊，請參閱 第 58 頁的『連線鑑別』。

  – 在 TLS 內使用交互鑑別。

    如需 TLS 的相關資訊，請參閱 第 230 頁的『使用 SSL/TLS』。

- 存取控制可用來提供或移除特定使用者或使用者群組的存取權。 透過使用特別建立的使用者 (或特定群組中的使用者) 執行用戶端應用程式，您可以使用存取控制來確保應用程式無法存取應用程式不應該存取的佇列管理程式部分。

  設定存取控制時，您必須考量通道鑑別規則及通道上的 MCAUSER 欄位。 這兩個特性都能夠變更用來驗證存取控制權限的使用者 ID。

  如需存取控制的相關資訊，請參閱 第 292 頁的『授權存取物件』。

如果您已設定用戶端應用程式連接至具有受限 ID 的特定通道，但通道在其 MCAUSER 欄位中已設定管理者 ID ，則只要用戶端應用程式順利連接，就會使用管理者 ID 進行存取控制檢查。 因此，用戶端應用程式將具有佇列管理程式的完整存取權。

如需 MCAUSER 屬性的相關資訊，請參閱 第 322 頁的『將用戶端使用者 ID 對映至 MCAUSER 使用者 ID』。

通道鑑別規則也可以用來作為控制佇列管理程式存取權的方法，方法是設定要接受連線的特定規則及準則。

如需通道鑑別規則的相關資訊，請參閱: 第 40 頁的『通道鑑別記錄』。

指定在執行時期於 MQI 用戶端上僅使用 FIPS 認證的 CipherSpecs

使用符合 FIPS 標準的軟體來建立金鑰儲存庫，然後指定通道必須使用 FIPS 認證的 CipherSpecs。

註: 在 AIX, Linux, and Windows 上， IBM MQ 透過 IBM Crypto for C (ICC) 加密模組提供 FIPS 140-2 相符性。 此模組的憑證已移至「歷程」狀態。 客戶應該檢視 IBM Crypto for C (ICC) 憑證 ， 並注意 NIST 提供的任何建議。 目前正在進行取代 FIPS 140-3 模組，您可以在 處理程序清單中的 NIST CMVP 模組中搜尋它，以檢視其狀態。

為了在執行時期符合 FIPS 標準，必須僅使用符合 FIPS 標準的軟體 (例如具有 -fips 選項的 runmqakm) 來建立及管理金鑰儲存庫。

您可以指定 TLS 通道必須以三種方式僅使用 FIPS 認證的 CipherSpecs ， 依優先順序列出:

1. 將 MQSCO 結構中的 FipsRequired 欄位設為 MQSSL_FIPS_YES。
2. 將環境變數 MQSSLFIPS 設為 YES。
3. 在用戶端配置檔中，將 SSLFipsRequired 屬性設為 YES。

依預設，不需要 FIPS 認證的 CipherSpecs 。

這些值的意義與 ALTER QMGR SSLFIPS 上的對等參數值相同 (請參閱 ALTER QMGR)。 如果用戶端處理程序目前沒有作用中的 TLS 連線，並且在 SSL MQCONNX 上有效指定了 FipsRequired 值，則與此處理程序相關聯的所有後續 TLS 連線都必須僅使用與此值相關聯的 CipherSpecs 。 除非此連線及所有其他 TLS 連線都已停止，在此階段後續 MQCONNX 可以為 FipsRequired 提供新值。

如果存在加密硬體，則 IBM MQ 所使用的加密模組可以配置為硬體產品所提供的那些模組，且這些模組可能經過 FIPS 認證達到特定層次。 可配置模組以及它們是否經過 FIPS 認證取決於使用中的硬體產品。

可能的話，如果已配置僅 FIPS CipherSpecs ，則 MQI 用戶端會拒絕使用 MQRC_SSL_INITIALIZATION_ERROR 指定非 FIPS CipherSpec 的連線。 IBM MQ 不保證拒絕所有這類連線，您必須負責判斷您的 IBM MQ 配置是否符合 FIPS 標準。

相關概念

第 28 頁的『AIX, Linux, and Windows 的聯邦資訊存取安全標準 (FIPS)』
當 AIX, Linux, and Windows 系統上的 SSL/TLS 通道需要加密法時， IBM MQ 會使用稱為 IBM Crypto for C (ICC)的加密法套件。 在 AIX, Linux, and Windows 平台上， ICC 軟體已通過美國國家標準與技術機構 (US National Institute of Standards and Technology) 的 Federal Information Processing Standards (FIPS) Cryptomodule Validation Program ， 層次 140-2。

相關參考

FipsRequired (MQLONG)
MQSSLFIPS
用戶端配置檔的 SSL 段落

AIX 在 AIX 上執行具有多個 GSKit 8.0 安裝的 TLS 用戶端應用程式

在具有多個 IBM Global Security Kit (GSKit) 8.0 版安裝的 AIX 系統上執行時， AIX 上的 TLS 用戶端應用程式可能會遇到 MQRC_CHANNEL_CONFIG_ERROR 及錯誤 AMQ6175 。

在具有多個 GSKit 8.0 安裝的 AIX 系統上執行用戶端應用程式時，使用 TLS 時用戶端連接呼叫可能會傳回 MQRC_CHANNEL_CONFIG_ERROR 。 失敗用戶端應用程式的 /var/mqm/errors 日誌記錄錯誤 AMQ6175 及 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 --------------------------------------------------------
```

此錯誤的常見原因是 LIBPATH 或 LD_LIBRARY_PATH 環境變數的設定已導致 IBM MQ 用戶端從兩個不同的 GSKit 8.0 安裝載入一組混合的程式庫。 在 Db2 環境中執行 IBM MQ 用戶端應用程式可能會導致此錯誤。

若要避免此錯誤，請在媒體庫路徑前面併入 IBM MQ 媒體庫目錄，以便優先使用 IBM MQ 媒體庫。 可以使用 **setmqenv** 指令搭配 **-k** 參數來達成此目的，例如:

```
. /usr/mqm/bin/setmqenv -s -k
```

如需使用 **setmqenv** 指令的相關資訊，請參閱 setmqenv (設定 IBM MQ 環境)

IBM i 在 IBM i 上設定 SSL 或 TLS 的通訊

使用 SSL 或 TLS 加密安全通訊協定的安全通訊包括設定通訊通道，以及管理您將用於鑑別的數位憑證。

若要設定 SSL 或 TLS 安裝，您必須定義通道以使用 SSL 或 TLS。 您也必須建立及管理數位憑證。 在某些作業系統上，您可以使用自簽憑證來執行測試。 不過，在 IBM i 上，您必須使用本端 CA 所簽署的個人憑證。

如需建立及管理憑證的完整資訊，請參閱 第 230 頁的『在 IBM i 上使用 SSL/TLS』。

此主題集合介紹設定 SSL 或 TLS 通訊所涉及的部分作業，並提供完成這些作業的逐步指引

您也可能想要測試 SSL 或 TLS 用戶端鑑別，它們是 SSL 和 TLS 通訊協定的選用組件。 在 SSL 或 TLS 信號交換期間， SSL 或 TLS 用戶端一律會從伺服器取得並驗證數位憑證。 使用 IBM MQ 實作， SSL 或 TLS 伺服器一律會從用戶端要求憑證。

在 IBM i 上，只有在 SSL 或 TLS 用戶端具有以正確 IBM MQ 格式標示的憑證時，才會傳送憑證:

- 對於佇列管理程式，後面接著佇列管理程式名稱的 ibmwebspheremq 會變更為小寫。 例如，若為 QM1，則為 ibmwebspheremqqm1。
- 對於 IBM MQ C Client for IBM i， ibmwebspheremq 後面接著您的登入使用者 ID 已變更為小寫，例如 ibmwebspheremqmyuserid。

IBM MQ 會使用標籤上的 ibmwebspheremq 字首，以避免與其他產品的憑證混淆。 請確保以小寫形式指定整個憑證標籤。

如果傳送用戶端憑證， SSL 或 TLS 伺服器一律會驗證用戶端憑證。 如果 SSL 或 TLS 用戶端未傳送憑證，則只有在使用 SSLCAUTH 參數設為 REQUIRED 或 SSLPEER 參數值來定義作為 SSL 或 TLS 伺服器的通道結尾時，鑑別才會失敗。 如需相關資訊，請參閱 使用 SSL 或 TLS 連接兩個佇列管理程式。

ALW 在 AIX, Linux, and Windows 上設定 SSL 或 TLS 的通訊

使用 SSL 或 TLS 加密安全通訊協定的安全通訊包括設定通訊通道，以及管理您將用於鑑別的數位憑證。

若要設定 SSL 或 TLS 安裝，您必須定義通道以使用 SSL 或 TLS。 您也必須建立及管理數位憑證。 在 AIX, Linux, and Windows 系統上，您可以使用自簽憑證來執行測試。

小心: 在您要使用啟用 TLS 的通道結合在一起的佇列管理程式上，無法混合使用橢圓曲線簽署憑證和 RSA 簽署憑證。

使用已啟用 TLS 通道的佇列管理程式必須全部使用 RSA 簽署憑證，或全部使用 EC 簽署憑證，而不是兩者混合使用。

如需相關資訊，請參閱 第 37 頁的『IBM MQ 中的數位憑證及 CipherSpec 相容性』。

無法撤銷自簽憑證，這可能容許攻擊者在私密金鑰受損之後盜用身分。 CA 可以撤銷已受損憑證，這會阻止其進一步使用。 因此，在正式作業環境中使用 CA 簽章憑證更安全，雖然自簽憑證對測試系統更方便。

如需建立及管理憑證的完整資訊，請參閱 第 240 頁的『在 AIX, Linux, and Windows 上使用 SSL/TLS』。

此主題集合介紹設定 SSL 通訊所涉及的部分作業，並提供完成這些作業的逐步指引。

您也可能想要測試 SSL 或 TLS 用戶端鑑別，這是通訊協定的選用部分。 在 SSL 或 TLS 信號交換期間， SSL 或 TLS 用戶端一律會從伺服器取得並驗證數位憑證。 使用 IBM MQ 實作， SSL 或 TLS 伺服器一律會從用戶端要求憑證。

在 AIX, Linux, and Windows 上，只有在憑證具有正確 IBM MQ 格式的標籤時， SSL 或 TLS 用戶端才會傳送憑證:

- 對於佇列管理程式，格式為 `ibmwebspheremq`，後面接著佇列管理程式的名稱已變更為小寫。 例如，對於 `QM1`， `ibmwebspheremqqm1`
- 對於 IBM MQ 用戶端，後面接著登入使用者 ID 的 `ibmwebspheremq` 已變更為小寫，例如 `ibmwebspheremqmyuserid`。

IBM MQ 會使用標籤上的 `ibmwebspheremq` 字首，以避免與其他產品的憑證混淆。 請確保以小寫形式指定整個憑證標籤。

如果傳送用戶端憑證， SSL 或 TLS 伺服器一律會驗證用戶端憑證。 如果用戶端未傳送憑證，則只有在使用 SSLCAUTH 參數設為 REQUIRED 或設定 SSLPEER 參數值來定義作為 SSL 或 TLS 伺服器的通道結尾時，鑑別才會失敗。 如需相關資訊，請參閱 使用 SSL 或 TLS 連接兩個佇列管理程式。

z/OS 在 z/OS 上設定 SSL 或 TLS 的通訊

使用 SSL 或 TLS 加密安全通訊協定的安全通訊包括設定通訊通道，以及管理您將用於鑑別的數位憑證。

若要設定 SSL 或 TLS 安裝，您必須定義通道以使用 SSL 或 TLS。 您也必須建立及管理數位憑證。 在 z/OS 上，您可以使用自簽憑證或本端憑證管理中心 (CA) 所簽署的個人憑證來執行測試。

無法撤銷自簽憑證，這可能容許攻擊者在私密金鑰受損之後盜用身分。 CA 可以撤銷已受損憑證，這會阻止其進一步使用。 因此，在正式作業環境中使用 CA 簽章憑證更安全，雖然自簽憑證對測試系統更方便。

如需建立及管理憑證的完整資訊，請參閱 第 266 頁的『在 z/OS 上使用 SSL/TLS』。

如需相關資訊，請參閱 ALTER QMGR 指令的 CERTLABL 及 CERTQSGL 參數，以及 DEFINE CHANNEL 指令的 CERLABL 參數。

優先順序如下:

- 通道 CERTLABL 參數
- 如果通道是共用的，則為 QMGR CERTQSGL 參數。

  對於傳送端通道，這表示傳輸佇列 (XMITQ) 是共用的。 對於接收端通道，這表示通道透過共用接聽器 (即具有 INDISP (GROUP) 的接聽器) 啟動。
- QMGR CERTLABL
- `ibmWebSphereMQ` 的預設標籤，後面接著共用通道的佇列共用群組名稱，或佇列管理程式的名稱。

此主題集合介紹設定 SSL 或 TLS 通訊所涉及的部分作業，並提供完成這些作業的逐步指引。

您也可能想要測試 SSL 或 TLS 用戶端鑑別，這是通訊協定的選用部分。 在 SSL 或 TLS 信號交換期間， SSL 或 TLS 用戶端一律會從伺服器取得並驗證數位憑證。 使用 IBM MQ 實作， SSL 或 TLS 伺服器一律會從用戶端要求憑證。

如果通道是共用的，通道會先嘗試尋找佇列共用群組的憑證。 如果找不到佇列共用群組的憑證，它會嘗試尋找佇列管理程式的憑證。

在 z/OS 上， IBM MQ 會在標籤上使用 `ibmWebSphereMQ` 字首，以避免與其他產品的憑證混淆。

如果傳送用戶端憑證， SSL 或 TLS 伺服器一律會驗證用戶端憑證。 如果 SSL 或 TLS 用戶端未傳送憑證，則只有在使用 SSLCAUTH 參數設為 REQUIRED 或 SSLPEER 參數值來定義作為 SSL 或 TLS 伺服器的通道結尾時，鑑別才會失敗。 如需相關資訊，請參閱 使用 SSL 或 TLS 連接兩個佇列管理程式。

使用 SSL/TLS

這些主題提供如何執行與搭配使用 TLS 與 IBM MQ 相關的單一作業的指示。

其中許多是用作下列各節所說明的較高層次作業中的步驟:

- 第 277 頁的『識別及鑑別使用者』
- 第 292 頁的『授權存取物件』
- 第 348 頁的『訊息機密性』
- 第 397 頁的『訊息的資料完整性』
- 第 398 頁的『保持叢集安全』

IBM i 在 IBM i 上使用 SSL/TLS

此主題集合提供在 IBM MQ for IBM i 中使用「傳輸層安全 (TLS)」之個別作業的指示。

對於 IBM i， TLS 支援是作業系統不可或缺的。 確保您已安裝 IBM i 上的軟硬體需求中列出的必備項目。

在 IBM i 上，您可以使用「數位 Certificate Manager (DCM)」工具來管理金鑰及數位憑證。

存取 *DCM*

請遵循下列指示來存取 DCM 介面。

關於這項作業

在支援頁框的 Web 瀏覽器中執行下列步驟。

程序

1. 移至 `http://machine.domain:2001` 或 `https://machine.domain:2010`，其中 *machine* 是電腦的名稱。
2. 當要求時，請鍵入有效的使用者設定檔及密碼。

   請確定您的使用者設定檔具有 *ALLOBJ 及 *SECADM 特殊權限，可讓您建立新的憑證儲存庫。 如果您沒有特殊權限，則只能管理您的個人憑證，或檢視您已獲授權之物件的物件簽章。 如果您已獲授權使用物件簽署應用程式，則也可以從 DCM 簽署物件。
3. 在「網際網路配置」頁面上，按一下 **數位 Certificate Manager**。

   即會顯示「數位 Certificate Manager 」頁面。

將憑證指派給 *IBM i* 上的佇列管理程式

使用 DCM 將憑證指派給佇列管理程式。

使用傳統 IBM i 數位憑證管理，將憑證指派給佇列管理程式。 這表示您可以指定佇列管理程式使用系統憑證儲存庫，並登錄佇列管理程式以作為使用「數位 Certificate Manager」的應用程式。 若要這麼做，請將佇列管理程式 **SSLKEYR** 屬性的值變更為 *SYSTEM。

當 **SSLKEYR** 參數變更為 *SYSTEM 時， IBM MQ 會將佇列管理程式登錄為具有唯一應用程式標籤 QIBM_WEBSPHERE_MQ_QMGRNAME 的伺服器應用程式，以及具有說明 Qmgrname (WMQ) 的標籤。 請注意，如果您使用 *SYSTEM 憑證儲存庫，則不會使用通道 **CERTLABL** 屬性。 然後佇列管理程式會在「數位 Certificate Manager」中顯示為伺服器應用程式，您可以將系統儲存庫中的任何伺服器或用戶端憑證指派給此應用程式。

因為佇列管理程式已登錄為應用程式，所以可以執行 DCM 的進階功能，例如定義 CA 信任清單。

如果 **SSLKEYR** 參數變更為 *SYSTEM 以外的值，則 IBM MQ 會將佇列管理程式取消登錄為具有「數位 Certificate Manager」的應用程式。 如果刪除佇列管理程式，也會從 DCM 取消登錄它。 具有足夠 *SECADM 權限的使用者也可以在 DCM 手動新增或移除應用程式。

在 *IBM i* 上設定金鑰儲存庫

必須在連線兩端設定金鑰儲存庫。 可以使用預設憑證儲存庫，也可以建立您自己的憑證儲存庫。

TLS 連線在連線的每一端都需要 金鑰儲存庫 。 每一個佇列管理程式及 IBM MQ MQI client 都必須具有金鑰儲存庫的存取權。 如果您要使用檔名及密碼來存取金鑰儲存庫 (亦即，不使用 *SYSTEM 選項) ，請確定 QMQM 使用者設定檔具有下列權限:

- 包含金鑰儲存庫之目錄的執行權限
- 包含金鑰儲存庫之檔案的讀取權限

如需相關資訊，請參閱第 21 頁的『SSL/TLS 金鑰儲存庫』。 請注意，如果您使用 *SYSTEM 憑證儲存庫，則不會使用通道 **CERTLABL** 屬性。

在 IBM i 上，數位憑證儲存在使用 DCM 管理的憑證儲存庫中。 這些數位憑證具有標籤，可將憑證與佇列管理程式或 IBM MQ MQI client 相關聯。 TLS 會使用憑證來進行鑑別。

標籤是 **CERTLABL** 屬性 (如果有設定的話) 的值，或是附加佇列管理程式名稱或 IBM MQ MQI client 使用者登入 ID (全部小寫) 的預設 `ibmwebspheremq` 。 如需詳細資料，請參閱 數位憑證標籤 。

佇列管理程式或 IBM MQ MQI client 憑證儲存庫名稱包含路徑和詞幹名稱。 預設路徑為 `/QIBM/UserData/ICSS/Cert/Server/` ，預設詞幹名稱為 `Default`。 在 IBM i 上，預設憑證儲存庫 `/QIBM/UserData/ICSS/Cert/Server/Default.kdb` 也稱為 `*SYSTEM`。 您可以選擇性地定義自己的路徑和詞幹名稱。

如果您定義自己的路徑或檔名，請設定檔案的許可權，以嚴格控制對檔案的存取權。

第 232 頁的『在 IBM i 上變更佇列管理程式的金鑰儲存庫位置』 告訴您如何指定憑證儲存庫名稱。 您可以在建立憑證儲存庫之前或之後指定憑證儲存庫名稱。

註: 您可以使用 DCM 執行的作業可能受到使用者設定檔權限的限制。 例如，您需要 *ALLOBJ 及 *SECADM 權限才能建立 CA 憑證。

在 *IBM i* 上建立憑證儲存庫

如果您不想使用預設憑證儲存庫，請遵循此程序來建立您自己的憑證儲存庫。

關於這項作業

只有在您不想使用 IBM i 預設憑證儲存庫時，才建立新的憑證儲存庫。

若要指定要使用 IBM i 系統憑證儲存庫，請將佇列管理程式的 SSLKEYR 屬性值變更為 *SYSTEM。 此值指出佇列管理程式使用系統憑證儲存庫，且佇列管理程式已登錄為使用「數位 Certificate Manager (DCM)」的應用程式。

程序

1. 存取 DCM 介面，如 第 230 頁的『存取 DCM』 所述
2. 在導覽畫面中，按一下 **建立新的憑證庫**。
   「建立新的憑證庫」頁面會顯示在作業頁框中。
3. 在作業框架中，選取 **其他系統憑證庫** ，然後按一下 **繼續**。
   「在新憑證庫中建立憑證」頁面會顯示在作業頁框中。

4. 選取 **否-不要在憑證儲存庫中建立憑證**，然後按一下 **繼續**。
   「憑證庫名稱和密碼」頁面會顯示在作業頁框中。
5. 在 **憑證儲存庫路徑和檔名** 欄位中，輸入 IFS 路徑和檔名，例如 `/QIBM/UserData/mqm/qmgrs/qm1/key.kdb`
6. 在 **密碼** 欄位中鍵入密碼，然後在 **確認密碼** 欄位中再次鍵入該密碼。 按一下**繼續**。
   請記下密碼 (區分大小寫)，因為您在隱藏儲存庫金鑰時需要它。
7. 若要結束 DCM，請關閉瀏覽器視窗。

下一步

當您使用 DCM 建立憑證儲存庫時，請確保隱藏密碼，如 第 232 頁的『在 IBM i 系統上隱藏憑證儲存庫密碼』 中所述。

相關工作

第 236 頁的『將憑證匯入 IBM i 上的金鑰儲存庫』
請遵循此程序來匯入憑證。

在 *IBM i* 系統上隱藏憑證儲存庫密碼

使用 CL 指令隱藏憑證庫密碼。

下列指示適用於在 IBM i 上隱藏佇列管理程式的憑證儲存庫密碼。 或者，若為 IBM MQ MQI client，如果您未使用 *SYSTEM 憑證儲存庫 (亦即， MQSSLKEYR 環境設為 *SYSTEM 以外的值)，請遵循 第 238 頁的『IBM MQ SSL 用戶端公用程式 (amqrsslc) for IBM i』的 第 239 頁的『隱藏憑證儲存庫密碼』 一節中說明的程序。

如果您已指定要使用 *SYSTEM 憑證儲存庫 (透過將佇列管理程式的 SSLKEYR 屬性值變更為 *SYSTEM)，則不得遵循這些步驟。

當您使用 DCM 建立憑證儲存庫時，請使用下列指令來隱藏密碼:

```
STRMQM MQMNAME('queue_manager_name')
CHGMQM MQMNAME('queue_manager_name') SSLKEYRPWD('password')
```

密碼會區分大小寫。 輸入的單引號必須與您在 第 231 頁的『在 IBM i 上建立憑證儲存庫』的步驟 6 中輸入的單引號完全相同。

註: 如果您不是使用預設系統憑證儲存庫，且未隱藏密碼，則嘗試啟動 TLS 通道會失敗，因為它們無法取得存取憑證儲存庫所需的密碼。

在 *IBM i* 上尋找佇列管理程式的金鑰儲存庫

使用此程序來取得佇列管理程式的憑證儲存庫位置。

程序

1. 使用下列指令顯示佇列管理程式的屬性:

   ```
   DSPMQM MQMNAME('queue manager name')
   ```

2. 請檢查指令輸出，以找出憑證儲存庫的路徑和系統名稱。
   例如: `/QIBM/UserData/ICSS/Cert/Server/Default`，其中 `/QIBM/UserData/ICSS/Cert/Server` 是路徑， `Default` 是詞幹名稱。

在 *IBM i* 上變更佇列管理程式的金鑰儲存庫位置

使用 CHGMQM 或 ALTER QMGR 變更佇列管理程式的憑證儲存庫位置。

程序

請使用 CHGMQM 指令或 ALTER QMGR MQSC 指令來設定佇列管理程式的金鑰儲存庫屬性。

a) 使用 CHGMQM: `CHGMQM MQMNAME('qm1') SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey')`

b) 使用 ALTER QMGR: `ALTER QMGR SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey')`
在任一情況下，憑證儲存庫都具有完整檔名: `/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb`

下一步

當您變更佇列管理程式憑證儲存庫的位置時，不會從舊位置傳送憑證。 如果在建立憑證儲存庫時預先安裝的 CA 憑證不足，則必須在新的憑證儲存庫中移入憑證，如 第 236 頁的『將憑證匯入 IBM i 上的金鑰儲存庫』中所述。 您也必須隱藏新位置的密碼，如 第 232 頁的『在 IBM i 系統上隱藏憑證儲存庫密碼』中所述。

建立憑證管理中心及憑證以在 *IBM i* 上進行測試

使用此程序來建立本端 CA 憑證以簽署憑證申請，以及建立並安裝 CA 憑證。

開始之前

本主題中的指示假設本端憑證管理中心 (CA) 不存在。 如果本端 CA 確實存在，請跳至 第 234 頁的『在 IBM i 上要求伺服器憑證』。

關於這項作業

安裝 TLS 時提供的 CA 憑證由發出 CA 簽署。 在 IBM i 上，您可以產生本端憑證管理中心，以簽署伺服器憑證來測試系統上的 TLS 通訊。 在 Web 瀏覽器中遵循下列步驟，以建立本端 CA 憑證:

程序

1. 存取 DCM 介面，如 第 230 頁的『存取 DCM』中所述。
2. 在導覽畫面中，按一下 **建立憑證管理中心**。
   「建立憑證管理中心」頁面會顯示在作業頁框中。
3. 在 **憑證儲存庫密碼** 欄位中鍵入密碼，然後在 **確認密碼** 欄位中再次鍵入該密碼。
4. 在 **憑證管理中心 (CA) 名稱** 欄位中鍵入名稱，例如 `TLS Test Certificate Authority`。
5. 在 **通用名稱** 和 **組織** 欄位中鍵入適當的值，然後選取國家/地區。 對於其餘選用欄位，請鍵入您需要的值。
6. 在 **有效性期間** 欄位中輸入本端 CA 的有效期間。
   預設值為 1095 天。
7. 按一下**繼續**。
   即會建立 CA，且 DCM 會為您的本端 CA 建立憑證儲存庫及 CA 憑證。
8. 按一下 **安裝憑證**。
   即會顯示下載管理程式對話框。
9. 鍵入您要儲存 CA 憑證之暫存檔的完整路徑名稱，然後按一下 **儲存**。
10. 下載完成時，請按一下 **開啟**。
    即會顯示「憑證」視窗。
11. 按一下 **安裝憑證**。
    即會顯示「憑證匯入」精靈。
12. 按**下一步**。
13. 選取 **根據憑證類型自動選取憑證儲存庫**，然後按 **下一步**。
14. 按一下**完成**。
    會顯示「確認」視窗。
15. 按一下**確定**。
16. 在「憑證」視窗中，按一下 **確定**。
17. 按一下**繼續**。
    「憑證管理中心原則」頁面會顯示在作業頁框中。
18. 在 **容許建立使用者憑證** 欄位中，選取 **是**。
19. 在 **有效性期間** 欄位中，輸入本端 CA 發出的憑證有效期間。

預設值為 365 天。

20. 按一下**繼續**。
    「在新憑證庫中建立憑證」頁面會顯示在作業頁框中。
21. 請檢查是否未選取任何應用程式。
22. 按一下 **繼續** 以完成本端 CA 的設定。

下一步

如果您需要更新現有憑證，請參閱 IBM i 文件中的 更新現有憑證 。

在 *IBM i* 上要求伺服器憑證

數位憑證可防止假冒，認證公開金鑰屬於指定的實體。 可以使用「數位 Certificate Manager (DCM)」，從憑證管理中心要求新的伺服器憑證。

關於這項作業

在 Web 瀏覽器中執行下列步驟:

程序

1. 存取 DCM 介面，如 第 230 頁的『存取 DCM』中所述。
2. 在導覽畫面中，按一下 **選取憑證庫**。
   作業頁框中會顯示「選取憑證庫」頁面。
3. 選取您要使用的憑證儲存庫，然後按一下 **繼續**。
4. 選擇性的: 如果您在步驟 3 中選取 ***SYSTEM** ，請輸入系統儲存庫密碼，然後按一下 **繼續**。
5. 選擇性的: 如果您在步驟 3 中選取 **其他系統憑證儲存庫** ，請在 **憑證儲存庫路徑和檔名** 欄位中輸入您建立憑證儲存庫時所設定的 IFS 路徑和檔名。 同時在 **憑證庫密碼** 欄位中鍵入密碼。 然後按一下 **繼續**
6. 在導覽畫面中，按一下 **建立憑證**。
7. 在作業框架中，選取 **伺服器或用戶端憑證** 圓鈕，然後按一下 **繼續**。
   作業頁框中會顯示「選取憑證管理中心 (CA)」頁面。
8. 如果工作站上有本端 CA ，請選擇本端 CA 或商業 CA 來簽署憑證。 選取所需 CA 的圓鈕，然後按一下 **繼續**。
   「建立憑證」頁面會顯示在作業頁框中。
9. 選擇性的: 若為佇列管理程式，請在 **憑證標籤** 欄位中輸入憑證標籤。
   標籤是 **CERTLABL** 屬性的值 (如果已設定的話) ，或預設 `ibmwebspheremq` 並附加佇列管理程式的名稱 (全部為小寫)。 如需詳細資料，請參閱 數位憑證標籤 。
   例如，若為佇列管理程式 QM1，請鍵入 `ibmwebspheremqqm1` 以使用預設值。
10. 選擇性的: 若為 IBM MQ MQI client，請在 **憑證標籤** 欄位中鍵入 `ibmwebspheremq` ，後面接著將登入使用者 ID 轉換成小寫。
    例如，鍵入 `ibmwebspheremqmyuserID`
11. 在 **通用名稱** 和 **組織** 欄位中鍵入適當的值，然後選取國家/地區。 對於其餘選用欄位，請鍵入您需要的值。

結果

如果您選取商業 CA 來簽署憑證， DCM 會建立 PEM (Privacy-Enhanced Mail) 格式的憑證申請。 將要求轉遞至您選擇的 CA。

如果您選取本端 CA 來簽署憑證， DCM 會通知您憑證已在憑證儲存庫中建立且可以使用。

在 *IBM i* 上要求 *IBM Key Manager* 的伺服器憑證

請遵循此程序來建立本端憑證管理中心 (CA) 所簽署的憑證，或套用商業 CA 所簽署的伺服器憑證，以匯入至 IBM 金鑰管理 (iKeyman) 公用程式。

關於這項作業

當數位 Certificate Manager (DCM) 在多個平台上充當 IBM MQ 的憑證管理程式時，必須使用使用者憑證。對於配送至其他平台的個人憑證，以及要匯入至 iKeyman 公用程式的個人憑證，請在 Web 瀏覽器中執行下列步驟:

程序

1. 存取 DCM 介面，如 第 230 頁的『存取 DCM』 中所述。
2. 在 **導覽** 窗格中，按一下 **建立憑證**。
   「 **建立憑證** 」頁面會顯示在作業頁框中。
3. 在「 **建立憑證** 」畫面上，選取 **使用者憑證** 圓鈕，然後按一下 **繼續**。
   即會顯示「 **建立使用者憑證** 」頁面。
4. 在「 **建立使用者憑證** 」畫面上，完成 **組織名稱**、 **州/省** 或 **州/省**、 **國家/地區** 或 **區域**的「憑證資訊」下的必要欄位。 選擇性地將值放置在 **組織單位** 和 **地區** 或 **城市** 欄位中。 按一下**繼續**。
   通用名稱 會自動設為您用來登入 iSeries 系統的使用者 ID。
5. 在下一個「 **建立使用者憑證** 」畫面上，按一下 **安裝憑證** ，然後按一下 **繼續**。
   即會顯示一則訊息，指出 **已安裝您的個人憑證。 您應該保留此憑證的備份副本。**
6. 按一下**確定**。
7. 視您用來存取 DCM 的網際網路瀏覽器而定，執行下列步驟:
   a) 對於 Microsoft Edge ，請選擇: **工具> 網際網路選項> 內容標籤> 憑證按鈕> 個人標籤>**。 選取憑證，然後按一下 **匯出**。
   b) 若為 Mozilla Firefox ，請選擇: **工具> 選項> 進階> 加密標籤> 檢視憑證按鈕> 您的憑證標籤>**。 選取憑證，然後按一下 **備份**。 選取路徑和檔名，然後按一下 **確定**。
8. 使用 FTP 以二進位格式將匯出的憑證傳送至遠端系統。
9. 將從步驟 7 匯出的憑證新增至金鑰資料庫中的 iKeyman 公用程式。
   a) 如果使用 Microsoft Edge 儲存憑證，請使用 從 Microsoft `.pfx` 匯入 檔案中說明的指示。
   b) 如果使用 Mozilla Firefox 儲存憑證，請使用 將個人憑證匯入金鑰儲存庫中說明的指示。
   在匯入期間，請確定個人憑證和簽章者憑證的標籤名稱已變更為 IBM MQ 所預期的名稱。 標籤必須是 IBM MQ **CERTLABL** 屬性的值 (如果已設定的話) ，或預設 `ibmwebspheremq` 並附加佇列管理程式名稱 (全部小寫)。 如需詳細資料，請參閱 數位憑證標籤 。

將伺服器憑證新增至 *IBM i* 上的金鑰儲存庫

請遵循此程序，將所要求的憑證新增至金鑰儲存庫。

關於這項作業

在 CA 傳送新的伺服器憑證給您之後，您可以將它新增至從中產生要求的憑證儲存庫。 如果 CA 在電子郵件訊息中傳送憑證，請將憑證複製到個別檔案。

註:

- 如果伺服器憑證由本端 CA 簽署，則不需要執行此程序。
- 在將 PKCS #12 格式的伺服器憑證匯入 DCM 之前，您必須先匯入對應的 CA 憑證。

使用下列程序，將伺服器憑證接收至佇列管理程式憑證儲存庫:

程序

1. 存取 DCM 介面，如 第 230 頁的『存取 DCM』 中所述。
2. 在導覽畫面的 **管理憑證** 作業種類中，按一下 **匯入憑證**。
   「匯入憑證」頁面會顯示在作業頁框中。
3. 選取憑證類型的圓鈕，然後按一下 **繼續**。
   「匯入伺服器或用戶端憑證」頁面或「匯入憑證管理中心 (CA) 憑證」頁面會顯示在作業頁框中。

4. 在 **匯入檔案** 欄位中，鍵入您要匯入之憑證的檔名，然後按一下 **繼續**。
   DCM 會自動決定檔案的格式。
5. 如果憑證是 **伺服器或用戶端** 憑證，請在作業頁框中鍵入密碼，然後按一下 **繼續**。
   DCM 會通知您已匯入憑證。

從 *IBM i* 上的金鑰儲存庫匯出憑證

匯出憑證會同時匯出公開和私密金鑰。 採取此動作時應格外小心，因為傳遞私密金鑰會完全危及您的安全。

開始之前

當您與另一個使用者共用使用者的憑證時，您會交換公開金鑰。 此處理程序在 **作業 5 中說明**。 **第 510 頁的『AIX and Linux 上 AMS 的快速入門手冊』的 共用憑證 區段中的 共用憑證** 。 當您依照這裡的說明來匯出憑證時，您會同時匯出公開和私密金鑰。 採取此動作時應格外小心，因為傳遞私密金鑰會完全危及您的安全。

關於這項作業

在您要從中匯出憑證的電腦上執行下列步驟:

程序

1. 存取 DCM 介面，如 第 230 頁的『存取 DCM』中所述。
2. 在導覽畫面中，按一下 **選取憑證庫**。
   作業頁框中會顯示「選取憑證庫」頁面。
3. 選取您要使用的憑證儲存庫，然後按一下 **繼續**。
4. 選擇性的: 如果您在步驟 3 中選取 ***SYSTEM** ，請輸入系統儲存庫密碼，然後按一下 **繼續**。
5. 選擇性的: 如果您在步驟 3 中選取 **其他系統憑證庫** ，請在 **憑證庫路徑和檔名** 欄位中輸入您建立憑證庫時所設定的 IFS 路徑和檔名，並在 **憑證庫密碼** 欄位中輸入密碼。 然後按一下 **繼續**
6. 在導覽畫面的 **管理憑證** 作業種類中，按一下 **匯出憑證**。
   「匯出憑證」頁面會顯示在作業頁框中。
7. 選取憑證類型的圓鈕，然後按一下 **繼續**。
   「匯出伺服器或用戶端憑證」頁面或「匯出憑證管理中心 (CA) 憑證」頁面會顯示在作業頁框中。
8. 選取您要匯出的憑證。
9. 選取圓鈕，以指定您要將憑證匯出至檔案，還是直接匯出至另一個憑證儲存庫。
10. 如果您選擇將伺服器或用戶端憑證匯出至檔案，請提供下列資訊:
    - 您要儲存所匯出憑證之位置的路徑和檔名。
    - 對於個人憑證，這是用來加密已匯出憑證及目標版次的密碼。 對於 CA 憑證，您不需要指定密碼。
11. 如果您選擇將憑證直接匯出至另一個憑證儲存庫，請指定目標憑證儲存庫及其密碼。
12. 按一下**繼續**。

將憑證匯入 *IBM i* 上的金鑰儲存庫

請遵循此程序來匯入憑證。

開始之前

在將 PKCS #12 格式的個人憑證匯入 DCM 之前，您必須先匯入對應的 CA 憑證。

關於這項作業

在您要匯入憑證的機器上執行這些步驟。

程序

1. 存取 DCM 介面，如 第 230 頁的『存取 DCM』中所述。

2. 在導覽畫面中，按一下 **選取憑證庫**。
   作業頁框中會顯示「選取憑證庫」頁面。
3. 選取您要使用的憑證儲存庫，然後按一下 **繼續**。
4. 選擇性的: 如果您在步驟 3 中選取 ***SYSTEM** ，請輸入系統儲存庫密碼，然後按一下 **繼續**。
5. 選擇性的: 如果您在步驟 3 中選取 **其他系統憑證庫** ，請在 **憑證庫路徑和檔名** 欄位中輸入您建立憑證庫時所設定的 IFS 路徑和檔名，並在 **憑證庫密碼** 欄位中輸入密碼。 然後按一下 **繼續**
6. 在導覽畫面的 **管理憑證** 作業種類中，按一下 **匯入憑證**。
   「匯入憑證」頁面會顯示在作業頁框中。
7. 選取憑證類型的圓鈕，然後按一下 **繼續**。
   「匯入伺服器或用戶端憑證」頁面或「匯入憑證管理中心 (CA) 憑證」頁面會顯示在作業頁框中。
8. 在 **匯入檔案** 欄位中，鍵入您要匯入之憑證的檔名，然後按一下 **繼續**。
   DCM 會自動決定檔案的格式。
9. 如果憑證是 **伺服器或用戶端** 憑證，請在作業頁框中鍵入密碼，然後按一下 **繼續**。 DCM 會通知您已匯入憑證。

在 *IBM i* 中移除憑證

使用此程序來移除個人憑證。

程序

1. 存取 DCM 介面，如 第 230 頁的『存取 DCM』中所述。
2. 在導覽畫面中，按一下 **選取憑證庫**。
   作業頁框中會顯示「選取憑證庫」頁面。
3. 選取 **其他系統憑證庫** 勾選框，然後按一下 **繼續**。
   即會顯示「憑證儲存庫及密碼」頁面。
4. 在 **憑證儲存庫路徑和檔名** 欄位中，輸入您建立憑證儲存庫時所設定的 IFS 路徑和檔名。
5. 在 **憑證庫密碼** 欄位中輸入密碼。 按一下**繼續**。
   「現行憑證庫」頁面會顯示在作業頁框中。
6. 在導覽畫面中的 **管理憑證** 作業種類中，按一下 **刪除憑證**。
   作業頁框中會顯示「確認刪除憑證」頁面。
7. 選取您要刪除的憑證。 按一下**刪除**。
8. 按一下 **是** ，以確認您要刪除憑證。 否則，按一下**否**。
   DCM 會通知您它是否已刪除憑證。

在 *IBM i* 上使用 **SYSTEM* 憑證儲存庫進行單向鑑別

請遵循下列指示來設定單向鑑別。

開始之前

- 建立佇列管理程式、通道及傳輸佇列。
- 在伺服器佇列管理程式上建立伺服器或用戶端憑證。
- 將 CA 憑證傳送至用戶端佇列管理程式，並將它匯入至金鑰儲存庫。
- 在伺服器和用戶端佇列管理程式上啟動接聽器。

關於這項作業

若要使用單向鑑別，請使用執行 IBM i 作為 TLS 伺服器的電腦，將「SSL 金鑰儲存庫 (SSLKEYR)」參數設為 *SYSTEM。 此設定會將 IBM MQ 佇列管理程式登錄為應用程式。 然後，您可以將憑證指派給佇列管理程式，以啟用單向鑑別。

您也可以在金鑰儲存庫中為用戶端佇列管理程式建立虛擬憑證，以使用私密金鑰儲存庫來實作單向鑑別。

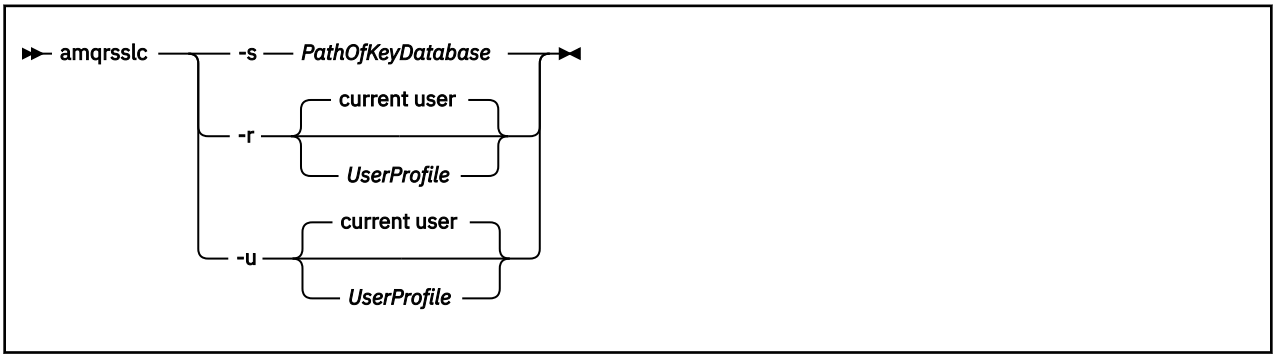
程序

1. 在伺服器和用戶端佇列管理程式上執行下列步驟:
   a) 發出指令 CHGMQM MQMNAME(SSL) SSLKEYR(*SYSTEM)來變更佇列管理程式，以設定 SSLKEYR 參數。
   b) 發出指令 CHGMQM MQMNAME(SSL) SSLKEYRPWD('xxxxxxx')來隱藏預設金鑰儲存庫的密碼。
   密碼必須以單引號括住。
   c) 變更通道，以在 SSLCIPHER 參數中具有正確的 CipherSpec 。
   d) 發出指令 RFRMQMAUT QMNAME(QMGRNAME) TYPE(*SSL)來重新整理 TLS 安全。
2. 使用 DCM 將憑證指派給伺服器佇列管理程式，如下所示:
   a) 存取 DCM 介面，如 第 230 頁的『存取 DCM』中所述。
   b) 在導覽畫面中，按一下 **選取憑證庫**。
   作業頁框中會顯示「選取憑證庫」頁面。
   c) 選取 *SYSTEM 憑證儲存庫，然後按一下 **繼續**。
   d) 在左畫面中，展開 **管理應用程式**。
   e) 選取 **檢視應用程式** 定義，以檢查佇列管理程式是否已登錄為應用程式。
   表格中會列出 SSL （WMQ）。
   f) 選取 **更新憑證指派**。
   g) 選取 **伺服器** ，然後按一下 **繼續**。
   h) 選取 QMGRNAME (WMQ) ，然後按一下 **更新憑證指派**。
   i) 選取憑證，然後按一下 **指派新憑證**。 即會開啟一個視窗，指出憑證已指派給應用程式。

IBM MQ SSL 用戶端公用程式 *(amqrsslc) for IBM i*

在 IBM i 系統上， IBM MQ MQI client 會使用 IBM i 的 IBM MQ SSL 用戶端公用程式 (amqrsslc) 來登錄或取消登錄用戶端使用者設定檔，或隱藏憑證儲存庫密碼。 此公用程式只能由具有 *ALLOBJ 特殊權限的設定檔，或具有在「數位 Certificate Manager (DCM)」中建立或刪除應用程式登錄之選項的 QMQMADM 成員來執行。

語法圖



登錄用戶端使用者設定檔

如果 IBM MQ MQI client 使用 *SYSTEM 憑證儲存庫，則您必須登錄用戶端使用者設定檔 (登入使用者) ，以使用 數位 Certificate Manager (DCM)作為應用程式。

如果您要登錄用戶端使用者設定檔，請使用 -r 選項與 *UserProfile* 來執行 **amqrsslc** 程式。 呼叫 **amqrsslc** 時所使用的使用者設定檔必須具有 *USE 權限。 為 *UserProfile* 提供-r 選項會將 *UserProfile* 註冊為伺服器應用程序，並具有唯一的應用程式標籤 Q IBM _WEBSPHERE_MQ_ *UserProfile* 和帶有 *UserProfile* (WMQ) 描述的標籤。 然後，此伺服器應用程式會顯示在 DCM 中，您可以將系統儲存庫中的任何伺服器或用戶端憑證指派給此應用程式。

註: 如果未使用 -r 選項指定使用者設定檔，則會登錄執行 **amqrsslc** 工具之使用者的使用者設定檔。

下列程式碼使用 **amqrsslc** 來登錄使用者設定檔。 在第一個範例中，會登錄指定的使用者設定檔; 在第二個範例中，它是已登入使用者的設定檔:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-r' UserProfile)
CALL PGM(QMQM/AMQRSSLC) PARM('-r')
```

取消登錄用戶端使用者設定檔

若要取消登錄用戶端設定檔，請搭配使用 -u 選項與 *UserProfile* 來執行 **amqrsslc** 程式。 呼叫 **amqrsslc** 時所使用的使用者設定檔必須具有 *USE 權限。 提供 *UserProfile* -u 選項將從 DCM 取消註冊帶有標籤 Q IBM _WEBSPHERE_MQ_ *UserProfile* 的 *UserProfile* 。

註: 如果未使用 -u 選項指定使用者設定檔，則會取消登錄執行 **amqrsslc** 工具之使用者的使用者設定檔。

下列程式碼使用 **amqrsslc** 來取消登錄使用者設定檔。 在第一個範例中，指定的使用者設定檔已取消登錄; 在第二個範例中，它是已登入使用者的設定檔:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-u' UserProfile)
CALL PGM(QMQM/AMQRSSLC) PARM('-u')
```

隱藏憑證儲存庫密碼

如果 IBM MQ MQI client 未使用 *SYSTEM 憑證儲存庫及使用另一個憑證儲存庫 (亦即， MQSSLKEYR 設為 *SYSTEM 以外的值) ，則必須隱藏金鑰資料庫的密碼。 使用 -s 選項來隱藏金鑰資料庫的密碼。

在下列程式碼中，憑證儲存庫的完整檔名是 /Path/Of/KeyDatabase/MyKey.kdb:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-s' '/Path/Of/KeyDatabase/MyKey')
```

執行此程式碼會導致要求此金鑰資料庫的密碼。 此密碼隱藏在與金鑰資料庫同名且副檔名為 .sth 的檔案中。 此檔案儲存在與金鑰資料庫相同的路徑上。 程式碼範例會產生 /Path/Of/KeyDatabase/MyKey.sth 的隱藏檔。 QMQM 是使用者擁有者，而 QMQMADM 是此檔案的群組擁有者。 QMQM 和 QMQMADM 具有讀取權、寫入權，而其他設定檔僅具有讀取權。

當憑證或憑證儲存庫的變更在 *IBM i* 上生效時

當您變更憑證儲存庫中的憑證或憑證儲存庫的位置時，變更會根據通道類型及通道執行的方式而生效。

在下列狀況下，對憑證儲存庫中的憑證及金鑰儲存庫屬性所做的變更會生效:

- 當新的出埠單一通道處理程序第一次執行 TLS 通道時。
- 當新的入埠 TCP/IP 單一通道處理程序第一次收到啟動 TLS 通道的要求時。
- 當發出 MQSC 指令 REFRESH SECURITY TYPE (SSL) 來重新整理 IBM MQ TLS 環境時。
- 對於用戶端應用程式程序，當程序中的最後一個 TLS 連線關閉時。 下一個 TLS 連線會挑選憑證變更。
- 對於作為處理程序儲存區處理程序 (amqrmppa) 的執行緒執行的通道，當啟動或重新啟動處理程序儲存區處理程序並先執行 TLS 通道時。 如果處理程序儲存區作業處理程序已執行 TLS 通道，且您想要變更立即生效，請執行 MQSC 指令 REFRESH SECURITY TYPE (SSL)。
- 對於作為通道起始程式的執行緒執行的通道，當啟動或重新啟動通道起始程式並先執行 TLS 通道時。 如果通道起始程式處理程序已執行 TLS 通道，且您想要變更立即生效，請執行 MQSC 指令 REFRESH SECURITY TYPE (SSL)。
- 對於作為 TCP/IP 接聽器的執行緒執行的通道，當接聽器啟動或重新啟動並首先收到啟動 TLS 通道的要求時。 如果接聽器已執行 TLS 通道，且您想要變更立即生效，請執行 MQSC 指令 REFRESH SECURITY TYPE (SSL)。

在 *IBM i* 上配置加密硬體

使用此程序在 IBM i 上配置加密輔助處理器

開始之前

請確定您的使用者設定檔具有 *ALLOBJ 及 *SECADM 特殊權限，可讓您配置輔助處理器硬體。

程序

1. 移至 `http://machine.domain:2001` 或 `https://machine.domain:2010`，其中 *machine* 是電腦的名稱。

   即會顯示一個對話框，要求使用者名稱和密碼。
2. 鍵入有效的 IBM i 使用者設定檔及密碼。
3. 請跳至 加密法 ，並遵循適當的鏈結以取得進一步資訊。

下一步

如需配置 4767 Cryptographic Coprocessor 的特定相關資訊，請參閱 4767 Cryptographic Coprocessor。

ALW 在 AIX, Linux, and Windows 上使用 SSL/TLS

在 AIX, Linux, and Windows 系統上，傳輸層安全 (TLS) 支援與 IBM MQ 一起安裝。

如需憑證驗證原則的詳細資訊，請參閱 憑證驗證及信任原則設計。

ALW 使用 *runmqckm*、 *runmqakm* 和 *strmqikm* 來管理數位憑證

在 AIX, Linux, and Windows 系統上，使用 **strmqikm** (iKeyman) 來管理金鑰和數位憑證 GUI ，或從指令行使用 **runmqckm** (iKeycmd) 或 **runmqakm** (GSKCapiCmd)。

小心: runmqckm 和 **strmqikm** 指令都依賴 IBM MQ Java 執行時期環境 (JRE)。 從 IBM MQ 9.1 開始，如果未安裝 JRE ，您會收到訊息 AMQ9183。

- Linux AIX 若為 **AIX and Linux** 系統:
  - 使用 **strmqikm** (iKeyman) 指令來啟動 iKeyman GUI。
  - 使用 **runmqckm** 指令，以使用指令行介面來執行作業。
  - 使用 **runmqakm** (GSKCapiCmd) 指令，以使用 runmqakm 指令行介面執行作業。 **runmqakm** 的指令語法與 **runmqckm** 的語法相同。

    如果您需要以符合 FIPS 標準的方式管理 TLS 憑證，請使用 **runmqakm** 指令而非 **runmqckm** 或 **strmqikm** 指令。

  如需 **runmqckm** 及 **runmqakm** 指令之指令行介面的完整說明，請參閱 管理金鑰及憑證 。

  如果您使用 PKCS #11 加密硬體上儲存的憑證或金鑰，請注意 **runmqckm** 和 iKeyman 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 Windows 和 Linux x86 32 位元平台是唯一的例外，因為 iKeyman 和 **runmqckm** 程式在這些平台上是 32 位元。

  如需進一步資訊，請參閱 GSKit: PKCS#11 及 IBM MQ JRE 定址模式 。

  在執行 **strmqikm** 指令以啟動 iKeyman GUI 之前，請確保您在能夠執行 X Window 系統的機器上工作，並且執行下列動作:
  - 設定 DISPLAY 環境變數，例如:

    ```
    export DISPLAY=mypc:0
    ```
  - 請確定 PATH 環境變數包含 **/usr/bin** 及 **/bin**。 這也是 **runmqckm** 和 **runmqakm** 指令的必要項目。 例如:

    ```
    export PATH=$PATH:/usr/bin:/bin
    ```
- Windows 若為 **Windows** 系統:
  - 使用 **strmqikm** 指令來啟動 iKeyman GUI。
  - 使用 **runmqckm** 指令，以使用指令行介面來執行作業。

    如果您需要以符合 FIPS 標準的方式管理 TLS 憑證，請使用 **runmqakm** 指令而非 **runmqckm** 或 **strmqikm** 指令。

– 搭配使用 **runmqakm -keydb** 指令與 *stashpw* 或 *stash* 選項。

  以這種方式使用 **runmqakm -keydb** 指令時，例如:

  ```
  runmqakm -keydb -create -db key.kdb -pw secretpwd -stash
  ```

  產生的 .sth 檔案未啟用 mqm 群組的讀取權。

  只有建立者可以讀取檔案。 使用 **runmqakm** 指令建立隱藏檔之後，請檢查檔案許可權，並授與許可權給執行佇列管理程式的服務帳戶或群組 (例如本端 mqm)。

ALW 若要在 AIX, Linux, and Windows 系統上要求 TLS 追蹤，請參閱 strmqtrc。

相關參考

第 451 頁的『AIX, Linux, and Windows 上的 runmqckm 及 runmqakm 指令』
本節根據指令的物件來說明 **runmqckm** 和 **runmqakm** 指令。

ALW 在 *AIX, Linux, and Windows* 上設定金鑰儲存庫

您可以使用 **strmqikm** (iKeyman) 來設定金鑰儲存庫 GUI，或從指令行使用 **runmqckm** (iKeycmd) 或 **runmqakm** (GSKCapiCmd) 指令。

關於這項作業

TLS 連線在連線的每一端都需要 金鑰儲存庫 。 每一個 IBM MQ 佇列管理程式及 IBM MQ MQI client 都必須具有金鑰儲存庫的存取權。 如需相關資訊，請參閱第 21 頁的『SSL/TLS 金鑰儲存庫』。

在 AIX, Linux, and Windows 系統上，數位憑證儲存在使用 **strmqikm** 使用者介面或使用 **runmqckm** 或 **runmqakm** 指令管理的金鑰資料庫檔中。 這些數位憑證具有標籤。 特定標籤會將個人憑證與佇列管理程式或 IBM MQ MQI client 相關聯。 TLS 會使用該憑證進行鑑別。 在 AIX, Linux, and Windows 系統上， IBM MQ 會使用 **CERTLABL** 屬性的值 (如果已設定的話)，或使用預設 ibmwebspheremq 並附加佇列管理程式或 IBM MQ MQI client 使用者登入 ID 的名稱 (全部都是小寫)。 如需詳細資料，請參閱 數位憑證標籤 。

金鑰資料庫檔名包含路徑和系統名稱:

- 在 AIX and Linux 系統上，佇列管理程式 (建立佇列管理程式時設定) 的預設路徑為 /var/mqm/qmgrs/*queue_manager_name*/ssl。

  在 Windows 系統上，預設路徑為 *MQ_INSTALLATION_PATH*\Qmgrs*queue_manager_name*\ssl，其中 *MQ_INSTALLATION_PATH* 是 IBM MQ 的安裝目錄。 例如， C:\Program Files\IBM\MQ\Qmgrs\QM1\ssl。

  預設詞幹名稱為 key。 您可以選擇性地選擇自己的路徑和詞幹名稱，但副檔名必須是 .kdb。

  如果您選擇自己的路徑或檔名，請設定檔案的許可權，以嚴格控制對檔案的存取權。
- 對於 IBM MQ 用戶端，沒有預設路徑或詞幹名稱。 嚴格控制對此檔案的存取權。 副檔名必須是 .kdb。

請勿在不支援檔案層次鎖定的檔案系統上建立金鑰儲存庫，例如 Linux 系統上的 NFS 第 2 版。

如需檢查及指定金鑰資料庫檔名的相關資訊，請參閱 第 245 頁的『在 AIX, Linux, and Windows 上變更佇列管理程式的金鑰儲存庫位置』。 您可以在建立金鑰資料庫檔之前或之後指定金鑰資料庫檔名稱。

您從中執行 **strmqikm** 或 **runmqckm** 指令的使用者 ID 必須對建立或更新金鑰資料庫檔所在的目錄具有寫入權。 對於使用預設 ssl 目錄的佇列管理程式，您從中執行 **strmqikm** 或 **runmqckm** 的使用者 ID 必須是 mqm 群組的成員。 若為 IBM MQ MQI client，如果您從不同於執行用戶端的使用者 ID 執行 **strmqikm** 或 **runmqckm**，則必須變更檔案許可權，讓 IBM MQ MQI client 能夠在執行時期存取金鑰資料庫檔案。 如需相關資訊，請參閱第 243 頁的『在 Windows 上存取金鑰資料庫檔案並保護其安全』 或第 243 頁的『在 AIX and Linux 系統上存取金鑰資料庫檔並保護其安全』。

在 **strmqikm** 或 **runmqckm** for IBM Global Security Kit (GSKit) 7.0 版中，新的金鑰資料庫會自動移入一組預先定義的憑證管理中心 (CA) 憑證。 在 **strmqikm** 或 **runmqckm** for GSKit 8.0 中，不會自動移入金鑰資料庫，使起始設定更安全，因為您只會在金鑰資料庫檔中包含您想要的 CA 憑證。

註: 因為 GSKit 8.0 行為中的這項變更會導致 CA 憑證不再自動新增至儲存庫，所以您必須手動新增偏好的 CA 憑證。 此行為變更可讓您更精確地控制所使用的 CA 憑證。 請參閱 第 243 頁的『使用 GSKit 8.0 將預設 CA 憑證新增至 AIX, Linux, and Windows 上的空金鑰儲存庫』。

您可以使用指令行或使用 **strmqikm** (iKeyman) 使用者介面來建立金鑰資料庫。

註: 如果您必須以符合 FIPS 標準的方式管理 TLS 憑證，請使用 **runmqakm** 指令。 **strmqikm** 使用者介面不提供符合 FIPS 標準的選項。

程序

使用指令行建立金鑰資料庫。

1. 執行下列其中一個指令:
   - 使用 **runmqckm**:

     ```
     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 檔名
   指定 CMS 金鑰資料庫的完整檔名，且副檔名必須是 .kdb。

   -pw *password*
   指定 CMS 金鑰資料庫的密碼。

   -type *cms*
   指定資料庫的類型。 (對於 IBM MQ，它必須是 cms。)

   -stash
   將金鑰資料庫密碼儲存至檔案。

   -fips
   指定以 FIPS 模式執行指令。 處於 FIPS 模式時， IBM Crypto for C (ICC) 元件會使用 FIPS 140-2 已驗證的演算法。 如果 ICC 元件未在 FIPS 模式下起始設定，則 **runmqakm** 指令會失敗。

   -強烈
   檢查輸入的密碼是否滿足密碼強度的最低需求。 密碼的最低需求如下:
   - 密碼長度下限必須為 14 個字元。
   - 密碼必須至少包含一個小寫字元、一個大寫字元，以及一個數字或特殊字元。 特殊字元包括星號 (*)、錢幣符號 ($)、數字符號 (#) 及百分比符號 (%)。 空格被分類為特殊字元。
   - 每一個字元在密碼中最多可以出現三次。
   - 密碼中最多可以有兩個連續字元相同。
   - 所有字元都在標準 ASCII 可列印字集內，範圍為 0x20 - 0x7E。

或者，使用 **strmqikm** (iKeyman) 使用者介面來建立金鑰資料庫。

2. 在 AIX and Linux 系統上，以 root 使用者身分登入。 在 Windows 系統上，以管理者身分或 MQM 群組成員身分登入。
3. 透過執行 **strmqikm** 指令來啟動使用者介面。
4. 從 **金鑰資料庫檔** 功能表中，按一下 **新建**。
   即會開啟「新建」視窗。
5. 按一下**金鑰資料庫類型**然後選取 **CMS**（憑證管理系統）。
6. 在 **檔名** 欄位中，鍵入檔名。
   此欄位已包含文字 key.kdb。 如果您的詞幹名稱是 key，請保留此欄位不變。 如果您指定不同的詞幹名稱，請將 key 取代為您的詞幹名稱。 不過，您不得變更 .kdb 副檔名。
7. 在 **位置** 欄位中，輸入路徑。

例如:

- 若為佇列管理程式: `/var/mqm/qmgrs/QM1/ssl` (在 AIX and Linux 系統上) 或 `C:\ProgramData\IBM \ MQ\qmgrs\QM1\ssl` (在 Windows 系統上)。

  路徑必須符合佇列管理程式的 **SSLKeyRepository** 屬性值。
- 若為 IBM MQ 用戶端: `/var/mqm/ssl` (在 AIX and Linux 系統上) 或 `C:\mqm\ssl` (在 Windows 系統上)。

8. 按一下**確定**。

   這時會開啟「密碼提示」視窗。
9. 在 **密碼** 欄位中鍵入密碼，然後在 **確認密碼** 欄位中再次鍵入密碼。
10. 選取 **將密碼隱藏至檔案** 勾選框。

    註: 如果您未隱藏密碼，則嘗試啟動 TLS 通道會失敗，因為它們無法取得存取金鑰資料庫檔所需的密碼。
11. 按一下**確定**。

    即會開啟「個人憑證」視窗。
12. 設定存取權，如 第 243 頁的『在 Windows 上存取金鑰資料庫檔案並保護其安全』 或 第 243 頁的『在 AIX and Linux 系統上存取金鑰資料庫檔並保護其安全』中所述。

Windows 在 *Windows* 上存取金鑰資料庫檔案並保護其安全

金鑰資料庫檔可能沒有適當的存取權。 您必須設定這些檔案的適當存取權。

設定 *key*.kdb、 *key*.sth、 *key*.crl 及 *key*.rdb 檔案的存取控制，其中 *key* 是金鑰資料庫的詞幹名稱，以將權限授與受限使用者集。

請考量授與存取權，如下所示:

完整權限
BUILTIN\ADMINISTRATORS、NT AUTHORITY\SYSTEM 及建立資料庫檔案的使用者。

讀取權限
若為佇列管理程式，則僅限本端 mqm 群組。 這會假設 MCA 是以 mqm 群組中的使用者 ID 來執行。

對於用戶端，這是用來執行用戶端處理程序的使用者 ID。

Linux AIX 在 *AIX and Linux* 系統上存取金鑰資料庫檔並保護其安全

金鑰資料庫檔可能沒有適當的存取權。 您必須設定這些檔案的適當存取權。

對於佇列管理程式，請設定金鑰資料庫檔的許可權，以便必要時佇列管理程式及通道處理程序可以讀取它們，但其他使用者無法讀取或修改它們。 通常， mqm 使用者需要讀取權。 如果您已透過以 mqm 使用者身分登入來建立金鑰資料庫檔，則許可權可能已足夠; 如果您不是 mqm 使用者，而是 mqm 群組中的另一個使用者，則可能需要將讀取權授與 mqm 群組中的其他使用者。

同樣地，對於用戶端，請設定金鑰資料庫檔的許可權，以便用戶端應用程式程序可以在必要時讀取它們，但其他使用者無法讀取或修改它們。 一般而言，執行用戶端程序的使用者需要讀取權。 如果您已透過以該使用者身分登入來建立金鑰資料庫檔，則許可權可能已足夠; 如果您不是用戶端處理程序使用者，而是該群組中的另一個使用者，則可能需要將讀取權授與群組中的其他使用者。

設定對檔案 *key*.kdb、 *key*.sth、 *key*.crl 及 *key*.rdb 的許可權，其中 *key* 是金鑰資料庫的詞幹名稱，對檔案擁有者設定 read 及 write ，對 mqm 或用戶端使用者群組設定 read (-rw-r -----)。

ALW 使用 *GSKit 8.0* 將預設 *CA* 憑證新增至 *AIX, Linux, and Windows* 上的空金鑰儲存庫

遵循此程序，將一或多個預設 CA 憑證新增至 IBM Global Security Kit (GSKit) 8.0 版的空金鑰儲存庫。

在 GSKit 7.0 中，建立新金鑰儲存庫時的行為是自動新增一組常用憑證管理中心的預設 CA 憑證。 對於 GSKit 8.0，此行為已變更，因此 CA 憑證不再自動新增至儲存庫。 現在，使用者必須手動將 CA 憑證新增至金鑰儲存庫。

使用 strmqikm

請在您要新增 CA 憑證的機器上執行下列步驟:

1. 使用 **strmqikm** 指令來啟動 GUI (在 AIX, Linux, and Windows 上)。
2. 從**金鑰資料庫檔**功能表，按一下**開啟**。 這時會開啟「開啟舊檔」視窗。
3. 按一下**金鑰資料庫類型**然後選取 **CMS**（憑證管理系統）。
4. 按一下**瀏覽**以導覽至包含金鑰資料庫檔的目錄。
5. 選取您要新增憑證的金鑰資料庫檔，例如 key.kdb。
6. 按一下**開啟** 。 這時會開啟「密碼提示」視窗。
7. 鍵入您在建立金鑰資料庫時設定的密碼，然後按一下**確定**。 您的金鑰資料庫檔名稱會顯示在**檔名**欄位中。
8. 在**金鑰資料庫內容**欄位中，選取**簽章者憑證**。
9. 按一下 **移入**。 即會開啟「新增 CA 的憑證」視窗。
10. 可新增至儲存庫的 CA 憑證會以階層式樹狀結構顯示。 選取您要信任其 CA 憑證的組織最上層項目，以檢視有效 CA 憑證的完整清單。
11. 從清單中選取您要信任的 CA 憑證，然後按一下 **確定**。 憑證會新增至金鑰儲存庫。

使用指令行

使用下列指令來列出，然後使用 **runmqckm** 來新增 CA 憑證:

- 發出下列指令，以列出預設 CA 憑證以及發出它們的組織:

```
runmqckm -cert -listsigners
```

- 發出下列指令，以新增 *label* 欄位中所指定組織的所有 CA 憑證:

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

其中:

-db *filename*	是金鑰資料庫的完整路徑名稱。
-pw *password*	是金鑰資料庫的密碼。
-label *label*	是附加至憑證的標籤。

註: 將 CA 憑證新增至金鑰儲存庫會導致 IBM MQ 信任該 CA 憑證所簽署的所有個人憑證。 請仔細考量您要信任哪些「憑證管理中心」，並僅新增鑑別用戶端及管理程式所需的 CA 憑證集。 除非這是安全原則的最終需求，否則不建議新增完整的預設 CA 憑證集。

ALW 在 *AIX, Linux, and Windows* 上尋找佇列管理程式的金鑰儲存庫

使用此程序來取得佇列管理程式的金鑰資料庫檔位置

程序

1. 使用下列其中一個 MQSC 指令，顯示佇列管理程式的屬性:

```
DISPLAY QMGR ALL
DISPLAY QMGR SSLKEYR
```

   您也可以使用 IBM MQ Explorer 或 PCF 指令來顯示佇列管理程式的屬性。
2. 請檢查指令輸出，以找出金鑰資料庫檔的路徑和詞幹名稱。
   例如,
   a. 在 AIX and Linux 上: /var/mqm/qmgrs/QM1/ssl/key，其中 /var/mqm/qmgrs/QM1/ssl 是路徑， key 是詞幹名稱

b. 在 Windows 上: *MQ_INSTALLATION_PATH*\qmgrs\QM1\ssl\key，其中 *MQ_INSTALLATION_PATH*\qmgrs\QM1\ssl 是路徑， key 是詞幹名稱。 *MQ_INSTALLATION_PATH* 代表 IBM MQ 安裝所在的高階目錄。

ALW 在 *AIX, Linux, and Windows* 上變更佇列管理程式的金鑰儲存庫位置

您可以透過各種方法 (包括 MQSC 指令 ALTER QMGR) 來變更佇列管理程式的金鑰資料庫檔位置。

您可以使用 MQSC 指令 ALTER QMGR 來設定佇列管理程式的金鑰儲存庫屬性，以變更佇列管理程式的金鑰資料庫檔位置。 例如，在 AIX and Linux 上:

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

金鑰資料庫檔具有完整檔名: `/var/mqm/qmgrs/QM1/ssl/MyKey.kdb`

在 Windows 上:

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

金鑰資料庫檔具有完整檔名: `C:\Program Files\IBM\MQ\Qmgrs\QM1\ssl\Mykey.kdb`

小心: 請確定您未在 SSLKEYR 關鍵字的檔名中包含 `.kdb` 副檔名，因為佇列管理程式會自動附加此副檔名。

您也可以使用「IBM MQ 探險家」或 PCF 指令來變更佇列管理程式的屬性。

當您變更佇列管理程式金鑰資料庫檔的位置時，不會從舊位置傳送憑證。 如果您現在存取的金鑰資料庫檔是新的金鑰資料庫檔，則必須將您需要的 CA 及個人憑證移入其中，如 第 257 頁的『將個人憑證匯入 AIX, Linux, and Windows 上的金鑰儲存庫』中所述。

ALW 在 *AIX, Linux, and Windows* 上尋找 *IBM MQ MQI client* 的金鑰儲存庫

金鑰儲存庫的位置由 MQSSLKEYR 變數提供，或在 MQCONNX 呼叫中指定。

檢查 MQSSLKEYR 環境變數，以尋找 IBM MQ MQI client 的金鑰資料庫檔位置。 例如:

```
echo $MQSSLKEYR
```

也請檢查您的應用程式，因為金鑰資料庫檔名也可以在 MQCONNX 呼叫中設定，如第 245 頁的『在 AIX, Linux, and Windows 上指定 IBM MQ MQI client 的金鑰儲存庫位置』中所述。 MQCONNX 呼叫中設定的值會置換 MQSSLKEYR 的值。

ALW 在 *AIX, Linux, and Windows* 上指定 *IBM MQ MQI client* 的金鑰儲存庫位置

IBM MQ MQI client 沒有預設金鑰儲存庫。 您可以使用兩種方式之一來指定其位置。 請確定只有預期的使用者或管理者才能存取金鑰資料庫檔，以防止未獲授權複製到其他系統。

您可以使用兩種方式來指定 IBM MQ MQI client 的金鑰資料庫檔位置:

- 設定 MQSSLKEYR 環境變數。 例如，在 AIX and Linux 上:

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

  金鑰資料庫檔具有完整檔名:

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

  在 Windows 上:

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

  金鑰資料庫檔具有完整檔名:

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

  註: `.kdb` 副檔名是檔名的必要部分，但不會併入作為環境變數值的一部分。

- 當應用程式發出 MQCONNX 呼叫時，在 MQSCO 結構的 *KeyRepository* 欄位中提供金鑰資料庫檔的路徑和系統名稱。 如需在 MQCONNX 中使用 MQSCO 結構的相關資訊，請參閱 MQSCO 概觀。

ALW 當憑證或憑證儲存庫的變更在 *AIX, Linux, and Windows* 上生效時

當您變更憑證儲存庫中的憑證或憑證儲存庫的位置時，變更會根據通道類型及通道執行的方式而生效。

在下列情況下，對金鑰資料庫檔中的憑證及金鑰儲存庫屬性所做的變更會生效:

- 當新的出埠單一通道處理程序第一次執行 TLS 通道時。
- 當新的入埠 TCP/IP 單一通道處理程序第一次收到啟動 TLS 通道的要求時。
- 當發出 MQSC 指令 REFRESH SECURITY TYPE (SSL) 來重新整理 TLS 環境時。
- 對於用戶端應用程式程序，當程序中的最後一個 TLS 連線關閉時。 下一個 TLS 連線將採用憑證變更。
- 對於作為處理程序儲存區處理程序 (amqrmppa) 的執行緒執行的通道，當啟動或重新啟動處理程序儲存區處理程序並先執行 TLS 通道時。 如果處理程序儲存區作業處理程序已執行 TLS 通道，且您想要變更立即生效，請執行 MQSC 指令 REFRESH SECURITY TYPE (SSL)。
- 對於作為通道起始程式的執行緒執行的通道，當啟動或重新啟動通道起始程式並先執行 TLS 通道時。 如果通道起始程式處理程序已執行 TLS 通道，且您想要變更立即生效，請執行 MQSC 指令 REFRESH SECURITY TYPE (SSL)。
- 對於作為 TCP/IP 接聽器的執行緒執行的通道，當接聽器啟動或重新啟動並首先收到啟動 TLS 通道的要求時。 如果接聽器已執行 TLS 通道，且您希望變更立即生效，請執行 MQSC 指令 REFRESH SECURITY TYPE (SSL)。

您也可以使用 IBM MQ Explorer 或 PCF 指令來重新整理 IBM MQ TLS 環境。

重要: . 對金鑰儲存庫配置檔及/或 AMS MCA 攔截程式 (以及一般用戶端中的 AMS) 所使用的金鑰儲存庫所做的變更，會在佇列管理程式或應用程式重新啟動時挑選。

ALW 在 *AIX, Linux, and Windows* 上建立自簽個人憑證

您可以使用 **strmqikm** (iKeyman) 來建立自簽憑證 GUI，或從指令行使用 **runmqckm** (iKeycmd) 或 **runmqakm** (GSKCapiCmd)。

註: IBM MQ 不支援 SHA-3 或 SHA-5 演算法。 您可以使用數位簽章演算法名稱 SHA384WithRSA 及 SHA512WithRSA，因為這兩個演算法都是 SHA-2 系列的成員。

數位簽章演算法名稱 SHA3WithRSA 和 SHA5WithRSA 已淘汰，因為它們分別是 SHA384WithRSA 和 SHA512WithRSA 縮寫形式。

如需為何您可能想要使用自簽憑證的相關資訊，請參閱 使用自簽憑證進行兩個佇列管理程式的交互鑑別。

並非所有數位憑證都可以與所有 CipherSpecs 搭配使用。 請確定您建立的憑證與您需要使用的 CipherSpecs 相容。 IBM MQ 支援三種不同類型的 CipherSpec。 如需詳細資料，請參閱 第 37 頁的『IBM MQ 中的數位憑證及 CipherSpec 相容性』 主題中的 第 38 頁的『橢圓曲線和 RSA CipherSpecs 的交互作業能力』。

若要使用類型 1 CipherSpecs (名稱以 ECDHE_ECDSA_開頭)，您必須使用 **runmqakm** 指令來建立憑證，並且必須指定「橢圓曲線 ECDSA」簽章演算法參數; 例如 **-sig_alg** EC_ecdsa_with_SHA384。

如需 **-sig_alg** 雜湊演算法可用的選項清單，請參閱 第 464 頁的『AIX, Linux, and Windows 上的 runmqckm 及 runmqakm 選項』。

如果您使用:

- GUI，請參閱 第 246 頁的『使用 strmqikm 使用者介面』
- 指令行，請參閱 第 247 頁的『使用指令行』

ALW 使用 *strmqikm* 使用者介面

您可以使用 **strmqikm** (iKeyman) 來建立個人憑證 GUI。

關於這項作業

strmqikm 不提供符合 FIPS 標準的選項。 如果您需要以符合 FIPS 標準的方式管理 TLS 憑證，請使用 **runmqakm** 指令。

程序

請完成下列步驟，以使用圖形使用者介面來建立佇列管理程式或 IBM MQ MQI client 的個人憑證:

1. 使用 **strmqikm** 指令來啟動 GUI。
2. 從**金鑰資料庫檔**功能表，按一下**開啟**。
   即會顯示「**開啟**」視窗。
3. 按一下**金鑰資料庫類型**然後選取 **CMS**（憑證管理系統）。
4. 按一下**瀏覽**以導覽至包含金鑰資料庫檔的目錄。
5. 選取您要從中產生要求的金鑰資料庫檔; 例如， `key.kdb`。
6. 按一下**確定**。
   即會開啟「**密碼提示**」視窗。
7. 鍵入您在建立金鑰資料庫時設定的密碼，然後按一下**確定**。
   金鑰資料庫檔的名稱會顯示在 **檔名** 欄位中。
8. 從 **建立** 功能表中，按一下 **新建自簽憑證**。 即會顯示「建立新的自簽憑證」視窗。
9. 在 **金鑰標籤** 欄位中，輸入憑證標籤。
   標籤是 **CERTLABL** 屬性的值 (如果有設定的話)，或預設 `ibmwebspheremq` 並附加佇列管理程式或 IBM MQ MQI client 登入使用者 ID 的名稱 (全部都是小寫)。 如需詳細資料，請參閱 數位憑證標籤 。
10. 在 **識別名稱** 欄位或任何 **主旨替代名稱** 欄位中鍵入或選取任何欄位的值。
11. 對於其餘欄位，請接受預設值，或鍵入或選取新值。
    如需「識別名稱」的相關資訊，請參閱 第 10 頁的『識別名稱』。
12. 按一下**確定**。
    個人憑證 清單會顯示您所建立之自簽個人憑證的標籤。

ALW 使用指令行

您可以從指令行使用 **runmqckm** (iKeycmd) 或 **runmqakm** (GSKCapiCmd) 指令來建立個人憑證。 如果您需要以符合 FIPS 標準的方式管理 SSL 或 TLS 憑證，請使用 **runmqakm** 指令。

程序

使用 **runmqckm** 或 **runmqakm** (GSKCapiCmd) 指令來建立自簽個人憑證。

- 使用 **runmqckm**:

```
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_dnsname` *DNS_names*、 `-san_emailaddr` *email_addresses* 或 `-san_ipaddr` *IP_addresses* 來取代 `-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
```

其中:

-db 檔名
指定 CMS 金鑰資料庫的完整檔名。

-pw *password*
指定 CMS 金鑰資料庫的密碼。

-label *label*
指定附加至憑證的金鑰標籤。 標籤是 **CERTLABL** 屬性的值 (如果有設定的話)，或預設 `ibmwebspheremq` 並附加佇列管理程式名稱或 IBM MQ MQI client 登入使用者 ID (全部都是小寫)。 如需詳細資料，請參閱 第 22 頁的『數位憑證標籤，瞭解需求』。

-dn *distinguished_name*
指定以雙引號括住的 X.500 識別名稱。 至少需要一個屬性。 您可以提供多個 OU 及 DC 屬性。

註: runmqckm 及 **runmqakm** 工具將郵遞區號屬性稱為 POSTALCODE，而不是 PC。 當您使用這些憑證管理指令來要求具有郵遞區號的憑證時，請一律在 **-dn** 參數中指定 POSTALCODE 。

-size *key_size*
指定金鑰大小。 如果您使用 **runmqckm**，值可以是 512 或 1024。 如果您使用 **runmqakm**，值可以是 512、 1024 或 2048。

x509version 版本
要建立的 X.509 憑證版本。 值可以是 1、2 或 3。 預設是 3。

-file *filename*
指定憑證申請的檔名。

-expire *days*
憑證的有效期限 (以天為單位)。 憑證的預設值為 365 天。

-fips
指定以 FIPS 模式執行指令。 只會使用 FIPS IBM Crypto for C (ICC) 元件，且必須以 FIPS 模式順利起始設定此元件。 處於 FIPS 模式時， ICC 元件會使用已驗證 FIPS 140-2 的演算法。 如果 ICC 元件未在 FIPS 模式下起始設定，則 **runmqakm** 指令會失敗。

-sig_alg
對於 **runmqckm**，指定用於建立項目金鑰組的非對稱簽章演算法。 值可以是 MD2_WITH_RSA、 MD2WithRSA、 MD5_WITH_RSA、 MD5WithRSA、 SHA1WithDSA、 SHA1WithECDSA、 SHA1WithRSA、 SHA2/ECDSA、 SHA224WithECDSA、 SHA256_WITH_RSA、 SHA256WithECDSA, SHA256WithRSA, SHA2WithECDSA, SHA3/ECDSA, SHA384_WITH_RSA, SHA384WithECDSA, SHA384WithRSA, SHA3WithECDSA, SHA5/ECDSA, SHA512_WITH_RSA, SHA512WithECDSA、 SHA512WithRSA、 SHA5WithECDSA、 SHA_WITH_DSA、 SHA_WITH_RSA、 SHAWithDSA、 SHAWithRSA。 預設值為 SHA1WithRSA。

-sig_alg
對於 **runmqakm**，指定在建立憑證申請期間使用的雜湊演算法。 此雜湊演算法用來建立與新建立的憑證申請相關聯的簽章。 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。 預設值為 SHA1WithRSA。

-san_dnsname *DNS_names*
指定要建立之項目的 DNS 名稱清單 (以逗點定界或空格定界)。

-san_emailaddr *email_addresses*
指定所建立項目的電子郵件位址清單 (以逗點定界或空格定界)。

-san_ipaddr *IP_adds*
指定要建立之項目的 IP 位址清單 (以逗點定界或空格定界)。

ALW 在 *AIX, Linux, and Windows* 上要求個人憑證

您可以使用 **strmqikm** (iKeyman) 來要求個人憑證 GUI，或從指令行使用 **runmqckm** (iKeycmd) 或 **runmqakm** (GSKCapiCmd) 指令。 如果您需要以符合 FIPS 標準的方式管理 SSL 或 TLS 憑證，請使用 **runmqakm** 指令。

關於這項作業

您可以使用 **strmqikm** GUI 或從指令行要求個人憑證，但需遵循下列考量:

- IBM MQ 不支援 SHA-3 或 SHA-5 演算法。 您可以使用數位簽章演算法名稱 SHA384WithRSA 及 SHA512WithRSA ，因為這兩個演算法都是 SHA-2 系列的成員。

- 數位簽章演算法名稱 SHA3WithRSA 和 SHA5WithRSA 已淘汰，因為它們分別是 SHA384WithRSA 和 SHA512WithRSA 縮寫形式。
- 並非所有數位憑證都可以與所有 CipherSpecs 搭配使用。 請確定您要求的憑證與您需要使用的 CipherSpecs 相容。 IBM MQ 支援三種不同類型的 CipherSpec。 如需詳細資料，請參閱 第 37 頁的『IBM MQ 中的數位憑證及 CipherSpec 相容性』 主題中的 第 38 頁的『橢圓曲線和 RSA CipherSpecs 的交互作業能力』 。
- 若要使用類型 1 CipherSpecs (名稱以 ECDHE_ECDSA_開頭) ，您必須使用 **runmqakm** 指令來要求憑證，並且必須指定橢圓曲線 ECDSA 簽章演算法參數; 例如 **-sig_alg** EC_ecdsa_with_SHA384。

  如需 **-sig_alg** 雜湊演算法可用的選項清單，請參閱 第 464 頁的『AIX, Linux, and Windows 上的 runmqckm 及 runmqakm 選項』 。
- 只有 **runmqakm** 指令提供符合 FIPS 標準的選項。
- 如果您使用加密硬體，請參閱 第 264 頁的『要求 PKCS #11 硬體的個人憑證』 。

如果您使用:

- GUI ，請參閱 第 249 頁的『使用 strmqikm 使用者介面』
- 指令行，請參閱 第 250 頁的『使用指令行』

ALW 使用 ***strmqikm*** 使用者介面

您可以使用 **strmqikm** (iKeyman) 來要求個人憑證 GUI。 如果您需要以符合 FIPS 標準的方式管理 SSL 或 TLS 憑證，請使用 **runmqakm** 指令。

關於這項作業

strmqikm 不提供符合 FIPS 標準的選項。 如果您需要以符合 FIPS 標準的方式管理 TLS 憑證，請使用 **runmqakm** 指令。

程序

請完成下列步驟，以使用 iKeyman 使用者介面來套用個人憑證:

1. 使用 **strmqikm** 指令來啟動使用者介面。
2. 從**金鑰資料庫檔**功能表，按一下**開啟**。

   這時會開啟「**開啟**」視窗。
3. 按一下**金鑰資料庫類型**然後選取 **CMS**（憑證管理系統）。
4. 按一下**瀏覽**以導覽至包含金鑰資料庫檔的目錄。
5. 選取您要從中產生要求的金鑰資料庫檔; 例如， key.kdb。
6. 按一下**開啟**。

   即會開啟「 **密碼提示** 」視窗。
7. 鍵入您在建立金鑰資料庫時設定的密碼，然後按一下**確定**。

   金鑰資料庫檔的名稱會顯示在 **檔名** 欄位中。
8. 從 **建立** 功能表中，按一下 **新建憑證申請**。 即會開啟「 **建立新的金鑰和憑證申請** 」視窗。
9. 在 **金鑰標籤** 欄位中，輸入憑證標籤。

   標籤是 **CERTLABL** 屬性的值 (如果有設定的話) ，或預設 ibmwebspheremq 並附加佇列管理程式或 IBM MQ MQI client 登入使用者 ID 的名稱 (全部都是小寫)。 如需詳細資料，請參閱 數位憑證標籤 。
10. 在 **識別名稱** 欄位或任何 **主旨替代名稱** 欄位中鍵入或選取任何欄位的值。 對於其餘欄位，請接受預設值，或鍵入或選取新值。

    如需「識別名稱」的相關資訊，請參閱 第 10 頁的『識別名稱』 。
11. 在 **輸入要在其中儲存憑證申請的檔案名稱** 欄位中，接受預設值 certreq.arm，或鍵入具有完整路徑的新值。
12. 按一下**確定**。

    會顯示「確認」視窗。
13. 按一下**確定**。

個人憑證申請 清單會顯示您所建立之新個人憑證申請的標籤。 憑證申請儲存在您在步驟 第 249 頁的『11』中選擇的檔案中。

14. 透過將檔案傳送至憑證管理中心 (CA)，或將檔案複製到 CA 網站上的要求表單，來要求新的個人憑證。

ALW 使用指令行

您可以使用 **runmqckm** (iKeycmd) 或 **runmqakm** (GSKCapiCmd) 指令，從指令行要求個人憑證。 如果您需要以符合 FIPS 標準的方式管理 SSL 或 TLS 憑證，請使用 **runmqakm** 指令。

程序

使用 **runmqckm** 或 **runmqakm** (GSKCapiCmd) 指令來要求個人憑證。

- 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* 來取代 -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 檔名
指定 CMS 金鑰資料庫的完整檔名。

-pw *password*
指定 CMS 金鑰資料庫的密碼。

-label *label*
指定附加至憑證的金鑰標籤。 標籤是 **CERTLABL** 屬性的值 (如果有設定的話)，或預設 ibmwebspheremq 並附加佇列管理程式名稱或 IBM MQ MQI client 登入使用者 ID (全部都是小寫)。 如需詳細資料，請參閱第 22 頁的『數位憑證標籤，瞭解需求』。

-dn *distinguished_name*
指定以雙引號括住的 X.500 識別名稱。 至少需要一個屬性。 您可以提供多個 OU 及 DC 屬性。

註: **runmqckm** 及 **runmqakm** 工具將郵遞區號屬性稱為 POSTALCODE，而不是 PC。 當您使用這些憑證管理指令來要求具有郵遞區號的憑證時，請一律在 **-dn** 參數中指定 POSTALCODE 。

-size *key_size*
指定金鑰大小。 如果您使用 **runmqckm**，值可以是 512 或 1024。 如果您使用 **runmqakm**，值可以是 512、 1024 或 2048。

-file *filename*
指定憑證申請的檔名。

-fips
指定以 FIPS 模式執行指令。 處於 FIPS 模式時， IBM Crypto for C (ICC) 元件會使用 FIPS 140-2 已驗證的演算法。 如果 ICC 元件未在 FIPS 模式下起始設定，則 **runmqakm** 指令會失敗。

-sig_alg
對於 **runmqckm**，指定用於建立項目金鑰組的非對稱簽章演算法。 數值可以是 MD2_WITH_RSA 、 MD2WithRSA 、 MD5_WITH_RSA 、 MD5WithRSA 、 SHA1WithDSA 、 SHA1WithECDSA 、 SHA1WithRSA 、 SHA2/ECDSA 、 SHA224WithECDSA 、 SHA2WithECDSA /ECDSW 2556425 2564252545 256425254 2545625736250000 ECDSA SHA256WithRSA SHA3/ECDSA 、 SHA384_WITH_RSA 、 SHA256_WITH_RSA 、 SHA256WithECDSA SHA3WithECDSA SHA384WithECDSA SHA384WithRSA SHA5/ECDSA 、 SHA512_WITH_RSA 、 SHA512WithECDSA 、

SHA512WithRSA、 SHA5WithECDSA、 SHA_WITH_DSA、 SHA_WITH_RSA、 SHAWithDSA、 SHAWithRSA。 預設值為 SHA1WithRSA。

-sig_alg
對於 **runmqakm**，指定在建立憑證申請期間使用的雜湊演算法。 此雜湊演算法用來建立與新建立的憑證申請相關聯的簽章。 該值可以是 md5、 MD5_WITH_RSA、 MD5WithRSA、 SHA_WITH_DSA、 SHA_WITH_RSA、 sha1、 SHA1WithDSA、 SHA1WithECDSA、 SHA1WithRSA、 sha224、 SHA224_WITH_RSA SHA256_WITH_RSA SHA2WithRSA SHA256WithECDSA sha256 SHA224WithRSA SHA224WithDSA SHA224WithECDSA SHA256WithDSA SHA256WithRSA、 sha384、 SHA384_WITH_RSA、 SHA384WithECDSA SHA384WithRSA、 sha512、 SHA512_WITH_RSA、 SHA512WithECDSA、 SHA512WithRSA EC_ecdsa_with_SHA384 SHAWithDSA、 SHAWithRSA、 EC_ecdsa_with_SHA1、 EC_ecdsa_with_SHA256 % EC_ecdsa_with_SHA224 EC_ecdsa_with_SHA512。 預設值為 SHA1WithRSA。

-san_dnsname *DNS_names*
指定要建立之項目的 DNS 名稱清單 (以逗點定界或空格定界)。

-san_emailaddr *email_addresses*
指定所建立項目的電子郵件位址清單 (以逗點定界或空格定界)。

-san_ipaddr *IP_adds*
指定要建立之項目的 IP 位址清單 (以逗點定界或空格定界)。

下一步

向 CA 提交憑證申請。 如需進一步資訊，請參閱第 252 頁的『將個人憑證接收至 AIX, Linux, and Windows 上的金鑰儲存庫』。

ALW 在 *AIX, Linux, and Windows* 上更新現有的個人憑證

您可以使用 **strmqikm** (iKeyman) 來更新個人憑證 GUI，或從指令行使用 **runmqckm** (iKeycmd) 或 **runmqakm** (GSKCapiCmd) 指令。

關於這項作業

如果您需要對個人憑證使用較大的金鑰大小，則無法更新現有憑證。 您必須遵循第 248 頁的『在 AIX, Linux, and Windows 上要求個人憑證』中說明的步驟來取代現有金鑰，以建立使用所需金鑰大小的新憑證申請。

個人憑證具有到期日，在此日期之後無法再使用憑證。 此作業說明如何在現有個人憑證到期之前更新它。

使用 ***strmqikm*** 使用者介面

關於這項作業

strmqikm 不提供符合 FIPS 標準的選項。 如果您需要以符合 FIPS 標準的方式管理 TLS 憑證，請使用 **runmqakm** 指令。

程序

請完成下列步驟，以使用 **strmqikm** 使用者介面來套用個人憑證:

1. 在 AIX, Linux, and Windows 上使用 **strmqikm** 指令來啟動使用者介面。
2. 從**金鑰資料庫檔**功能表，按一下**開啟**。
   這時會開啟「**開啟**」視窗。
3. 按一下**金鑰資料庫類型**然後選取 **CMS**（憑證管理系統）。
4. 按一下**瀏覽**以導覽至包含金鑰資料庫檔的目錄。
5. 選取您要從中產生要求的金鑰資料庫檔; 例如， key.kdb。
6. 按一下**開啟**。
   即會開啟「 **密碼提示** 」視窗。
7. 鍵入您在建立金鑰資料庫時設定的密碼，然後按一下**確定**。

金鑰資料庫檔的名稱會顯示在 **檔名** 欄位中。

8. 從下拉選項功能表中選取 **個人憑證**，然後從清單中選取您要更新的憑證。
9. 按一下 **重建要求 ...** 按鈕。
   即會開啟視窗，讓您輸入檔名及檔案位置資訊。
10. 在 **檔名** 欄位中，接受預設 `certreq.arm`，或鍵入新值 (包括完整檔案路徑)。
11. 按一下**確定**。 憑證申請儲存在您在步驟 第 252 頁的『9』中選取的檔案中。
12. 透過將檔案傳送至憑證管理中心 (CA)，或將檔案複製到 CA 網站上的要求表單，來要求新的個人憑證。

使用指令行

程序

使用下列指令，透過 **runmqckm** 或 **runmqakm** 指令來要求個人憑證:

- 使用 **runmqckm**:

```
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 檔名
指定 CMS 金鑰資料庫的完整檔名。

-pw *password*
指定 CMS 金鑰資料庫的密碼。

-target 檔名
指定憑證申請的檔名。

註: 由於舊憑證資訊位於記憶體快取中，您必須執行指令 REFRESH SECURITY TYPE (SSL)。

下一步

從憑證管理中心收到已簽署的個人憑證之後，您可以使用 第 252 頁的『將個人憑證接收至 AIX, Linux, and Windows 上的金鑰儲存庫』中說明的步驟將它新增至金鑰資料庫。

ALW 將個人憑證接收至 *AIX, Linux, and Windows* 上的金鑰儲存庫

使用此程序將個人憑證接收至金鑰資料庫檔。 金鑰儲存庫必須是您在其中建立憑證申請的相同儲存庫。

在 CA 傳送新的個人憑證給您之後，您可以將它新增至您從中產生新憑證申請的金鑰資料庫檔。 如果 CA 在電子郵件訊息中傳送憑證，請將憑證複製到個別檔案。

使用 strmqikm

如果您需要以符合 FIPS 標準的方式管理 TLS 憑證，請使用 **runmqakm** 指令。 **strmqikm** 不提供符合 FIPS 標準的選項。

請確定要匯入的憑證檔對現行使用者具有寫入權，然後對佇列管理程式或 IBM MQ MQI client 使用下列程序，以將個人憑證接收至金鑰資料庫檔:

1. 使用 **strmqikm** 指令啟動 GUI。
2. 從**金鑰資料庫檔**功能表，按一下**開啟**。 這時會開啟「開啟舊檔」視窗。
3. 按一下**金鑰資料庫類型**然後選取 **CMS**（憑證管理系統）。
4. 按一下**瀏覽**以導覽至包含金鑰資料庫檔的目錄。

5. 選取您要新增憑證的金鑰資料庫檔，例如 key.kdb。
6. 按一下 **開啟**，然後按一下 **確定**。 這時會開啟「密碼提示」視窗。
7. 鍵入您在建立金鑰資料庫時設定的密碼，然後按一下**確定**。 金鑰資料庫檔的名稱會顯示在 **檔名** 欄位中。 選取 **個人憑證** 視圖。
8. 按一下 **接收**。 即會開啟「從檔案接收憑證」視窗。
9. 鍵入新個人憑證的憑證檔名及位置，或按一下 **瀏覽** 以選取名稱及位置。
10. 按一下**確定**。 如果您在金鑰資料庫中已有個人憑證，則會開啟一個視窗，詢問您是否要將您新增的金鑰設為資料庫中的預設金鑰。
11. 按一下 **是** 或 **否**。 這時會開啟「輸入標籤」視窗。
12. 按一下**確定**。 **個人憑證** 欄位會顯示您新增之個人憑證的標籤。

使用指令行

若要將個人憑證新增至金鑰資料庫檔，請使用下列其中一個指令:

- 使用 **runmqckm**:

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

- 使用 **runmqakm**:

```
runmqakm -cert -receive -file filename -db filename -pw password -fips
```

其中:

-file *filename*
指定個人憑證的完整檔名。

-db 檔名
指定 CMS 金鑰資料庫的完整檔名。

-pw *password*
指定 CMS 金鑰資料庫的密碼。

-format *ascii*
指定憑證格式。 值可以是 ascii（代表 Base64 編碼 ASCII）或 binary（代表二進位 DER 資料）。 預設值是 ascii。

-fips
指定以 FIPS 模式執行指令。 處於 FIPS 模式時， IBM Crypto for C (ICC) 元件會使用已驗證 FIPS 140-2 的演算法。 如果 ICC 元件未在 FIPS 模式下起始設定，則 **runmqakm** 指令會失敗。

如果您使用加密硬體，請參閱 第 265 頁的『將個人憑證接收至 PKCS #11 硬體』。

ALW 從 *AIX, Linux, and Windows* 上的金鑰儲存庫擷取 *CA* 憑證

請遵循此程序來擷取 CA 憑證。

使用 strmqikm

如果您需要以符合 FIPS 標準的方式管理 TLS 憑證，請使用 **runmqakm** 指令。 **strmqikm** (iKeyman) 不提供符合 FIPS 標準的選項。

在您要從中擷取 CA 憑證的機器上執行下列步驟:

1. 使用 **strmqikm** 指令啟動 GUI。
2. 從**金鑰資料庫檔**功能表，按一下**開啟**。 這時會開啟「開啟舊檔」視窗。
3. 按一下**金鑰資料庫類型**然後選取 **CMS**（憑證管理系統）。
4. 按一下**瀏覽**以導覽至包含金鑰資料庫檔的目錄。
5. 選取您要從中擷取的金鑰資料庫檔，例如 key.kdb。

6. 按一下**開啟**。 這時會開啟「密碼提示」視窗。
7. 鍵入您在建立金鑰資料庫時設定的密碼，然後按一下**確定**。 金鑰資料庫檔的名稱會顯示在 **檔名** 欄位中。
8. 在 **金鑰資料庫內容** 欄位中，選取 **簽章者憑證**，然後選取您要擷取的憑證。
9. 按一下 **擷取**。 即會開啟「將憑證擷取至檔案」視窗。
10. 針對副檔名為 `.arm` 的檔案，選取憑證的 **資料類型**，例如 **Base64-encoded ASCII 資料**。
11. 鍵入您要儲存憑證的憑證檔名及位置，或按一下 **瀏覽** 以選取名稱及位置。
12. 按一下**確定**。 憑證會寫入您指定的檔案。

使用指令行

使用下列指令，以使用 **runmqckm** 指令或 **runmqakm** 指令來擷取 CA 憑證:

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

or

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

其中:

`-db` *filename*	是 CMS 金鑰資料庫的完整路徑名稱。
`-pw` *password*	是 CMS 金鑰資料庫的密碼。
`-label` *label*	是附加至憑證的標籤。
`-target` *filename*	是目的地檔案的名稱。
`-format` *ascii*	是憑證的格式。 值可以是 `ascii`（代表 Base64 編碼 ASCII）或 `binary`（代表二進位 DER 資料）。 預設值是 `ascii`。
`-fips`	指定以 FIPS 模式執行指令。 處於 FIPS 模式時， IBM Crypto for C (ICC) 元件會使用已驗證 FIPS 140-2 的演算法。 如果 ICC 元件未在 FIPS 模式下起始設定，則 **runmqakm** 指令會失敗。

ALW 從 *AIX, Linux, and Windows* 上的金鑰儲存庫擷取自簽憑證的公用部分

請遵循此程序來擷取自簽憑證的公用部分。

使用 strmqikm

如果您需要以符合 FIPS 標準的方式管理 TLS 憑證，請使用 **runmqakm** 指令。 **strmqikm** (iKeyman) 不提供符合 FIPS 標準的選項。

在您要從中擷取自簽憑證公用部分的機器上執行下列步驟:

1. 使用 **strmqikm** 指令啟動 GUI。
2. 從**金鑰資料庫檔**功能表，按一下**開啟**。 這時會開啟「開啟舊檔」視窗。
3. 按一下**金鑰資料庫類型**然後選取 **CMS**（憑證管理系統）。
4. 按一下**瀏覽**以導覽至包含金鑰資料庫檔的目錄。
5. 選取您要從中擷取憑證的金鑰資料庫檔，例如 `key.kdb`。
6. 按一下**確定**。 這時會開啟「密碼提示」視窗。
7. 鍵入您在建立金鑰資料庫時設定的密碼，然後按一下**確定**。 金鑰資料庫檔的名稱會顯示在 **檔名** 欄位中。
8. 在 **金鑰資料庫內容** 欄位中，選取 **個人憑證**，然後選取憑證。
9. 按一下 **擷取憑證**。 即會開啟「將憑證擷取至檔案」視窗。

10. 針對副檔名為 .arm 的檔案，選取憑證的 **資料類型** ，例如 **Base64-encoded ASCII 資料** 。
11. 鍵入您要儲存憑證的憑證檔名及位置，或按一下 **瀏覽** 以選取名稱及位置。
12. 按一下**確定**。 憑證會寫入您指定的檔案。 請注意，當您擷取 (而非匯出) 憑證時，只會包含憑證的公用部分，因此不需要密碼。

使用指令行

使用下列指令，以使用 **runmqckm** 或 **runmqakm** 擷取自簽憑證的公用部分:

- 使用 runmqckm:

```
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 金鑰資料庫的完整路徑名稱。
-pw *password*	是 CMS 金鑰資料庫的密碼。
-label *label*	是附加至憑證的標籤。
-target *filename*	是目的地檔案的名稱。
-format *ascii*	是憑證的格式。 值可以是 ascii（代表 Base64 編碼 ASCII）或 binary（代表二進位 DER 資料）。 預設值是 ascii。
-fips	指定以 FIPS 模式執行指令。 處於 FIPS 模式時， IBM Crypto for C (ICC) 元件會使用已驗證 FIPS 140-2 的演算法。 如果 ICC 元件未在 FIPS 模式下起始設定，則 **runmqakm** 指令會失敗。

ALW 將 *CA* 憑證或自簽憑證的公用部分新增至 *AIX, Linux, and Windows* 上的金鑰儲存庫

遵循此程序可將 CA 憑證或自簽憑證的公用部分新增至金鑰儲存庫。

如果您要新增的憑證是在憑證鏈中，則也必須新增在其鏈結中位置上方的所有憑證。 您必須以嚴格遞減順序新增憑證，從主要憑證開始，接著是鏈結中緊接著它之下的 CA 憑證，依此類推。

下列指示不只適用於 CA 憑證，它們也適用於自簽憑證的公用部分。

註: 您必須確定憑證採用 ASCII (UTF-8) 或二進位 (DER) 編碼，因為 IBM Global Security Kit (GSKit) 不支援具有其他編碼類型的憑證。

使用 strmqikm

如果您需要以符合 FIPS 標準的方式管理 TLS 憑證，請使用 **runmqakm** 指令。 **strmqikm** 不提供符合 FIPS 標準的選項。

請在您要新增 CA 憑證的機器上執行下列步驟:

1. 使用 **strmqikm** 指令啟動 GUI。
2. 從**金鑰資料庫檔**功能表，按一下**開啟**。 這時會開啟「開啟舊檔」視窗。
3. 按一下**金鑰資料庫類型**然後選取 **CMS**（憑證管理系統）。
4. 按一下**瀏覽**以導覽至包含金鑰資料庫檔的目錄。
5. 選取您要新增憑證的金鑰資料庫檔，例如 key.kdb。
6. 按一下**確定**。 這時會開啟「密碼提示」視窗。

7. 鍵入您在建立金鑰資料庫時設定的密碼，然後按一下**確定**。 您的金鑰資料庫檔名稱會顯示在**檔名**欄位中。
8. 在**金鑰資料庫內容**欄位中，選取**簽章者憑證**。
9. 按一下**新增**。 這時會開啟「從檔案新增 CA 憑證」視窗。
10. 鍵入憑證檔名以及儲存憑證的位置，或按一下**瀏覽**以選取名稱及位置。
11. 按一下**確定**。 這時會開啟「輸入標籤」視窗。
12. 在「輸入標籤」視窗中，鍵入憑證的名稱。
13. 按一下**確定**。 憑證已新增至金鑰資料庫。

使用指令行

若要將 CA 憑證新增至金鑰資料庫，請使用下列其中一個指令:

- 使用 **runmqckm**:

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

- 使用 **runmqakm**:

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

其中:

-db 檔名
指定 CMS 金鑰資料庫的完整檔名。

-pw *password*
指定 CMS 金鑰資料庫的密碼。

-label *label*
指定附加至憑證的標籤。

-file *filename*
指定包含憑證的檔案名稱。

-format *ascii*
指定憑證格式。 值可以是 `ascii`（代表 Base64 編碼 ASCII）或 `binary`（代表二進位 DER 資料）。 預設值是 `ascii`。

-fips
指定以 FIPS 模式執行指令。 處於 FIPS 模式時， IBM Crypto for C (ICC) 元件會使用已驗證 FIPS 140-2 的演算法。 如果 ICC 元件未在 FIPS 模式下起始設定，則 **runmqakm** 指令會失敗。

ALW 從 *AIX, Linux, and Windows* 上的金鑰儲存庫匯出個人憑證

請遵循此程序來匯出個人憑證。

使用 strmqikm

如果您需要以符合 FIPS 標準的方式管理 TLS 憑證，請使用 **runmqakm** 指令。 **strmqikm** (iKeyman) 不提供符合 FIPS 標準的選項。

在您要從中匯出個人憑證的機器上執行下列步驟:

1. 使用 **strmqikm** 指令啟動 GUI。
2. 從**金鑰資料庫檔**功能表，按一下**開啟**。 這時會開啟「開啟舊檔」視窗。
3. 按一下**金鑰資料庫類型**然後選取 **CMS**（憑證管理系統）。
4. 按一下**瀏覽**以導覽至包含金鑰資料庫檔的目錄。
5. 選取您要從中匯出憑證的金鑰資料庫檔，例如 `key.kdb`。
6. 按一下**開啟** 。 這時會開啟「密碼提示」視窗。

7. 鍵入您在建立金鑰資料庫時設定的密碼，然後按一下**確定**。 金鑰資料庫檔的名稱會顯示在 **檔名** 欄位中。
8. 在 **金鑰資料庫內容** 欄位中，選取 **個人憑證** ，然後選取您要匯出的憑證。
9. 按一下 **匯出/匯入**。 即會開啟「匯出/匯入金鑰」視窗。
10. 選取 **匯出金鑰**。
11. 選取您要匯出之憑證的 **金鑰檔類型** ，例如 **PKCS12**。
12. 鍵入您要將憑證匯出至其中的檔名及位置，或按一下 **瀏覽** 以選取名稱及位置。
13. 按一下**確定**。 這時會開啟「密碼提示」視窗。 請注意，當您匯出 (而非擷取) 憑證時，會包括憑證的公用及專用部分。 這就是匯出檔受到密碼保護的原因。 當您擷取憑證時，只會包含憑證的公用部分，因此不需要密碼。
14. 在 **密碼** 欄位中鍵入密碼，然後在 **確認密碼** 欄位中再次鍵入密碼。
15. 按一下**確定**。 憑證會匯出至您指定的檔案。

使用指令行

使用 **runmqckm** 指令或 **runmqakm** 指令匯出個人憑證:

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

or

```
runmqakm -cert -export -db filename -pw password -label label -type cms
        -target filename -target_pw password -target_type pkcs12
        -encryption strong | weak -fips
```

其中:

-db *filename*	是 CMS 金鑰資料庫的完整路徑名稱。
-encryption	是憑證匯出指令中使用的加密強度。 值可以是 **強** 或 **弱**。 預設值是 strong。
-fips	指定以 FIPS 模式執行指令。 處於 FIPS 模式時， IBM Crypto for C (ICC) 元件會使用已驗證 FIPS 140-2 的演算法。 如果 ICC 元件未在 FIPS 模式下起始設定，則 **runmqakm** 指令會失敗。
-pw *password*	是 CMS 金鑰資料庫的密碼。
-label *label*	是附加至憑證的標籤。
-type *cms*	是資料庫的類型。
-target *filename*	是目的地檔案的完整路徑名稱。
-target_pw *password*	是用於加密憑證的密碼。
-target_type *pkcs12*	是憑證的類型。

ALW 將個人憑證匯入 *AIX, Linux, and Windows* 上的金鑰儲存庫

請遵循此程序來匯入個人憑證

將 PKCS #12 格式的個人憑證匯入金鑰資料庫檔之前，您必須先將發出 CA 憑證的完整有效鏈新增至金鑰資料庫檔 (請參閱 第 255 頁的『將 CA 憑證或自簽憑證的公用部分新增至 AIX, Linux, and Windows 上的金鑰儲存庫』)。

PKCS #12 檔案應視為暫時檔案，並在使用後刪除。

使用 strmqikm

如果您需要以符合 FIPS 標準的方式管理 TLS 憑證，請使用 **runmqakm** 指令。 **strmqikm** 不提供符合 FIPS 標準的選項。

在您要匯入個人憑證的機器上執行下列步驟:

1. 使用 **strmqikm** 指令啟動 GUI。
2. 從**金鑰資料庫檔**功能表，按一下**開啟**。 即會顯示「開啟」視窗。
3. 按一下**金鑰資料庫類型**然後選取 **CMS**（憑證管理系統）。
4. 按一下**瀏覽**以導覽至包含金鑰資料庫檔的目錄。
5. 選取您要新增憑證的金鑰資料庫檔，例如 key.kdb。
6. 按一下**開啟**。 即會顯示「密碼提示」視窗。
7. 鍵入您在建立金鑰資料庫時設定的密碼，然後按一下**確定**。 您的金鑰資料庫檔名稱會顯示在**檔名**欄位中。
8. 在 **金鑰資料庫內容** 欄位中，選取 **個人憑證**。
9. 如果「個人憑證」視圖中有憑證，請遵循下列步驟:
   a. 按一下 **匯出/匯入**。 即會顯示「匯出/匯入金鑰」視窗。
   b. 選取 **匯入金鑰**。
10. 如果「個人憑證」視圖中沒有憑證，請按一下 **匯入**。
11. 選取您要匯入之憑證的 **金鑰檔類型** ，例如 PKCS12。
12. 鍵入憑證檔名以及儲存憑證的位置，或按一下**瀏覽**以選取名稱及位置。
13. 按一下**確定**。 即會顯示「密碼提示」視窗。
14. 在 **密碼** 欄位中，鍵入匯出憑證時使用的密碼。
15. 按一下**確定**。 即會顯示「變更標籤」視窗。 例如，如果目標金鑰資料庫中已存在具有相同標籤的憑證，則您可以變更所匯入憑證的標籤。 變更憑證標籤不會影響憑證鏈驗證。 若要將憑證與特定佇列管理程式或 IBM MQ MQI client 相關聯， IBM MQ 會使用 **CERTLABL** 屬性的值 (如果已設定) ，或使用預設 ibmwebspheremq 並附加佇列管理程式或 IBM MQ MQI client 使用者登入 ID 的名稱 (全部小寫)。 如需詳細資料，請參閱 數位憑證標籤 。
16. 若要變更標籤，請從 **選取要變更的標籤** 清單中選取所需的標籤。 標籤會複製到 **輸入新標籤** 輸入欄位。 將標籤文字取代為新標籤的文字，然後按一下 **套用**。
17. **輸入新標籤** 輸入欄位中的文字會複製回 **選取要變更的標籤** 欄位，取代原先選取的標籤，因此重新標示對應的憑證。
18. 當您已變更所有需要變更的標籤時，請按一下 **確定**。 即會關閉「變更標籤」視窗，且原始 IBM 金鑰管理視窗會重新出現，並以正確標示的憑證更新 **個人憑證** 及 **簽章者憑證** 欄位。
19. 憑證會匯入至目標金鑰資料庫。

使用指令行

若要使用 **runmqckm** 匯入個人憑證，請使用下列指令:

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

若要使用 **runmqakm** 匯入個人憑證，請使用下列指令:

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

其中:

-file *filename*	是包含 PKCS #12 憑證之檔案的完整檔名。
-pw *password*	是 PKCS #12 憑證的密碼。
-type *pkcs12*	是檔案的類型。
-target *filename*	是目的地 CMS 金鑰資料庫的名稱。
-target_pw *password*	是 CMS 金鑰資料庫的密碼。

-target_type *cms*	是 -target 指定的資料庫類型
-label *label*	是要從來源金鑰資料庫匯入的憑證標籤。
-new_label *label*	是將在目標資料庫中指派憑證的標籤。 如果您省略 -new_label 選項，預設值是使用與 -label 選項相同的。
-fips	指定以 FIPS 模式執行指令。 處於 FIPS 模式時， IBM Crypto for C (ICC) 元件會使用已驗證 FIPS 140-2 的演算法。 如果 ICC 元件未在 FIPS 模式下起始設定，則 **runmqakm** 指令會失敗。

runmqckm 不提供直接變更憑證標籤的指令。 請使用下列步驟來變更憑證標籤:

1. 使用 **-cert -export** 指令將憑證匯出至 PKCS #12 檔案。 指定 -label 選項的現有憑證標籤。
2. 使用 **-cert -delete** 指令，從原始金鑰資料庫中移除憑證的現有副本。
3. 使用 **-cert -import** 指令從 PKCS #12 檔案匯入憑證。 指定 -label 選項的舊標籤，以及 -new_label 選項的必要新標籤。 憑證將匯入回具有必要標籤的金鑰資料庫。

ALW 從 *Microsoft.pfx* 檔案匯入個人憑證

請遵循此程序，從 AIX, Linux, and Windows 上的 Microsoft.pfx 檔案匯入。

.pfx 檔可以包含兩個與相同金鑰相關的憑證。 一個是個人或網站憑證 (同時包含公開和私密金鑰)。 另一個是 CA (簽章者) 憑證 (僅包含公開金鑰)。 這些憑證不能同時存在於相同的 CMS 金鑰資料庫檔中，因此只能匯入其中一個憑證。 此外， 「一般名稱」或標籤僅附加至簽章者憑證。

個人憑證由系統產生的「唯一使用者 ID (UUID)」識別。 本節顯示從 pfx 檔案匯入個人憑證，同時使用先前指派給 CA (簽章者) 憑證的一般名稱來標示該個人憑證。 發出 CA (簽章者) 憑證應該已新增至目標金鑰資料庫。 請注意， PKCS#12 檔案應該視為暫時檔案，並在使用之後刪除。

請遵循下列步驟，從來源 pfx 金鑰資料庫匯入個人憑證:

1. 使用 **strmqikm** 指令啟動 GUI。 即會顯示「 IBM 金鑰管理」視窗。
2. 從**金鑰資料庫檔**功能表，按一下**開啟**。 即會顯示「開啟」視窗。
3. 選取 **PKCS12** 的金鑰資料庫類型。
4. **建議您在執行此步驟之前先備份 pfx 資料庫**。 選取您要匯入的 pfx 金鑰資料庫。 按一下**開啟** 。 即會顯示「密碼提示」視窗。
5. 輸入金鑰資料庫密碼，然後按一下 **確定**。 即會顯示「 IBM 金鑰管理」視窗。 標題列會顯示所選取 pfx 金鑰資料庫檔的名稱，指出檔案已開啟且備妥。
6. 從清單中選取 **簽章者憑證** 。 必要憑證的「一般名稱」會顯示為「簽章者憑證」畫面中的標籤。
7. 選取標籤項目，然後按一下 **刪除** 以移除簽章者憑證。 即會顯示「確認」視窗。
8. 按一下 **是**。 選取的標籤不再顯示在「簽章者憑證」畫面中。
9. 針對所有簽章者憑證，重複步驟 6、7 和 8。
10. 從**金鑰資料庫檔**功能表，按一下**開啟**。 即會顯示「開啟」視窗。
11. 選取要將 pfx 檔案匯入其中的目標索引鍵 CMS 資料庫。 按一下**開啟** 。 即會顯示「密碼提示」視窗。
12. 輸入金鑰資料庫密碼，然後按一下 **確定**。 即會顯示「 IBM 金鑰管理」視窗。 標題列會顯示所選金鑰資料庫檔的名稱，指出檔案已開啟且備妥。
13. 從清單中選取 **個人憑證** 。
14. 如果「個人憑證」視圖中有憑證，請遵循下列步驟:
    a. 按一下 **匯出/匯入金鑰**。 即會顯示「匯出/匯入金鑰」視窗。
    b. 從「選擇動作類型」中選取 **匯入** 。
15. 如果「個人憑證」視圖中沒有憑證，請按一下 **匯入**。
16. 選取 PKCS12 檔案。
17. 輸入在步驟 4 中使用的 pfx 檔案名稱。 按一下**確定**。 即會顯示「密碼提示」視窗。
18. 指定您在刪除簽章者憑證時指定的相同密碼。 按一下**確定**。

19. 即會顯示「變更標籤」視窗 (因為應該只有單一憑證可用於匯入)。 憑證的標籤應該是 UUID，其格式為 xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx。
20. 如果要變更標籤，請從 **選取要變更的標籤:** 畫面中選取 UUID。 標籤將抄寫至 **輸入新標籤:** 欄位。 將標籤文字取代為步驟 7 中所刪除一般名稱的標籤文字，然後按一下 **套用**。 一般名稱必須是 IBM MQ **CERTLABL** 屬性的值 (如果已設定)，或預設 ibmwebspheremq 並附加佇列管理程式或 IBM MQ MQI client 使用者登入 ID 的名稱 (全部為小寫)。 如需詳細資料，請參閱 數位憑證標籤 。
21. 按一下**確定**。 現在會移除「變更標籤」視窗，原始 IBM 金鑰管理視窗會重新出現，並以正確標示的個人憑證更新「個人憑證和簽章者憑證」畫面。
22. pfx 個人憑證現在已匯入至 (目標) 資料庫。

無法使用 **runmqckm** 或 **runmqakm** 來變更憑證標籤。

使用指令行

若要使用 **runmqckm** 匯入個人憑證，請使用下列指令:

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

若要使用 **runmqakm** 匯入個人憑證，請使用下列指令:

```
runmqakm -cert -import -file filename -pw password -type pkcs12 -target filename
-target_pw password -target_type cms -label label -fips -pfx
```

其中:

-file *filename*	是包含 PKCS #12 憑證之檔案的完整檔名。
-pw *password*	是 PKCS #12 憑證的密碼。
-type *pkcs12*	是檔案的類型。
-target *filename*	是目的地 CMS 金鑰資料庫的名稱。
-target_pw *password*	是 CMS 金鑰資料庫的密碼。
-target_type *cms*	是 -target 指定的資料庫類型
-label *label*	是要從來源金鑰資料庫匯入的憑證標籤。
-new_label *label*	是將在目標資料庫中指派憑證的標籤。 如果您省略 -new_label 選項，預設值是使用與 -label 選項相同的。
-fips	指定以 FIPS 模式執行指令。 處於 FIPS 模式時， IBM Crypto for C (ICC) 元件會使用已驗證 FIPS 140-2 的演算法。 如果 ICC 元件未在 FIPS 模式下起始設定，則 **runmqakm** 指令會失敗。
-pfx	指出 PFX 檔案格式。

runmqckm 不提供直接變更憑證標籤的指令。 請使用下列步驟來變更憑證標籤:

1. 使用 **-cert -export** 指令將憑證匯出至 PKCS #12 檔案。 指定 -label 選項的現有憑證標籤。
2. 使用 **-cert -delete** 指令，從原始金鑰資料庫中移除憑證的現有副本。
3. 使用 **-cert -import** 指令從 PKCS #12 檔案匯入憑證。 指定 -label 選項的舊標籤，以及 -new_label 選項的必要新標籤。 憑證將匯入回具有必要標籤的金鑰資料庫。

ALW 從 *PKCS #7* 檔案匯入個人憑證

strmqikm (iKeyman) 和 **runmqckm** (iKeycmd) 工具不支援 PKCS #7 (.p7b) 檔案。 使用 **runmqakm** 工具，從 AIX, Linux, and Windows 上的 PKCS #7 檔案匯入憑證。

使用下列指令，從 PKCS #7 檔案新增 CA 憑證:

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

-db *filename*	是 CMS 金鑰資料庫的完整檔名。
-pw *password*	是金鑰資料庫的密碼。
-type *cms*	是金鑰資料庫的類型。
-file *filename*	是 PKCS #7 檔案的名稱。
-label *label*	是在目標資料庫中指派憑證的標籤。 第一個憑證採用給定的標籤。 所有其他憑證 (如果有的話) 都會以其主旨名稱標示。

使用下列指令，從 PKCS #7 檔案匯入個人憑證:

```
runmqakm -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 憑證之檔案的完整檔名。
-pw *password*	是 PKCS #7 憑證的密碼。
-type *pkcs7*	是檔案的類型。
-target *filename*	是目的地金鑰資料庫的名稱。
-target_pw *password*	是目的地金鑰資料庫的密碼。
-target_type *cms*	是 -target 指定的資料庫類型
-label *label*	是要匯入之憑證的標籤。
-new_label *label*	是將在目標資料庫中指派憑證的標籤。 如果您省略 -new_label 選項，預設值是使用與 -label 選項相同的。

ALW 從 *AIX, Linux, and Windows* 上的金鑰儲存庫中刪除憑證

使用此程序來移除個人或 CA 憑證。

使用 strmqikm

如果您需要以符合 FIPS 標準的方式管理 TLS 憑證，請使用 **runmqakm** 指令。 **strmqikm** (iKeyman) 不提供符合 FIPS 標準的選項。

1. 使用 **strmqikm** 指令啟動 GUI。
2. 從**金鑰資料庫檔**功能表，按一下**開啟**。 這時會開啟「開啟舊檔」視窗。
3. 按一下**金鑰資料庫類型**然後選取 **CMS**（憑證管理系統）。
4. 按一下**瀏覽**以導覽至包含金鑰資料庫檔的目錄。
5. 選取您要從中刪除憑證的金鑰資料庫檔，例如 key.kdb。
6. 按一下**開啟** 。 這時會開啟「密碼提示」視窗。
7. 鍵入您在建立金鑰資料庫時設定的密碼，然後按一下**確定**。 金鑰資料庫檔的名稱會顯示在 **檔名** 欄位中。
8. 從下拉清單中，選取 **個人憑證** 或 **簽章者憑證**
9. 選取您要刪除的憑證。
10. 如果您還沒有憑證副本且想要儲存它，請按一下 **匯出/匯入** 並匯出它 (請參閱 第 256 頁的『從 AIX, Linux, and Windows 上的金鑰儲存庫匯出個人憑證』)。
11. 選取憑證之後，按一下 **刪除**。 即會開啟「確認」視窗。
12. 按一下 **是**。 **個人憑證** 欄位不再顯示您已刪除之憑證的標籤。

使用指令行

使用下列指令，以使用 **runmqckm** 指令或 **runmqakm** 指令來刪除憑證:

使用 runmqckm:

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

使用 runmqakm:

```
runmqakm -cert -delete -db filename -pw password -label label -fips
```

其中:

-db *filename*	是 CMS 金鑰資料庫的完整檔名。
-pw *password*	是 CMS 金鑰資料庫的密碼。
-label *label*	是附加至個人憑證的標籤。
-fips	指定以 FIPS 模式執行指令。 處於 FIPS 模式時， IBM Crypto for C (ICC) 元件會使用已驗證 FIPS 140-2 的演算法。 如果 ICC 元件未在 FIPS 模式下起始設定，則 **runmqakm** 指令會失敗。

ALW 在 *AIX, Linux, and Windows* 上產生金鑰儲存庫保護的高保護性密碼

您可以使用 **runmqakm** (GSKCapiCmd) 指令產生金鑰儲存庫保護的高保護性密碼。

您可以搭配使用 **runmqakm** 指令與下列參數，以產生高保護性密碼:

```
runmqakm -random -create -length 14 -strong -fips
```

在後續憑證管理指令的 **-pw** 參數上使用產生的密碼時，請一律以雙引號括住密碼。 在 AIX and Linux 系統上，如果下列字元出現在密碼字串中，您也必須使用反斜線字元來跳出這些字元:

```
! \ " '
```

當輸入密碼以回應來自 **runmqckm**、 **runmqakm** 或 **strmqikm** GUI 的提示時，不需要引用或跳出密碼。 這是不必要的，因為在這些情況下，作業系統 Shell 不會影響資料輸入。

ALW 在 *AIX, Linux, and Windows* 上配置加密硬體

您可以使用多種方式來配置佇列管理程式或用戶端的加密硬體。

您可以使用下列其中一種方法，為 AIX, Linux, and Windows 上的佇列管理程式配置加密硬體:

- 搭配使用 ALTER QMGR MQSC 指令與 SSLCRYP 參數，如 ALTER QMGR 中所述。
- 使用 IBM MQ Explorer 來配置 AIX, Linux, and Windows 系統上的加密硬體。 如需相關資訊，請參閱線上說明。

您可以使用下列其中一種方法，為 AIX, Linux, and Windows 上的 IBM MQ 用戶端配置加密硬體:

- 設定 MQSSLCRYP 環境變數。 MQSSLCRYP 的允許值與 SSLCRYP 參數的允許值相同，如 ALTER QMGR 中所述。

  如果您使用 SSLCRYP 參數的 GSK_PKCS11 版本，則 PKCS #11 記號標籤必須符合您配置的硬體標籤。
- 在 IBM MQ client 配置檔的 SSL 段落中設定 SSLCryptographicHardware 屬性。 允許的值與 SSLCRYP 參數的值相同，如 ALTER QMGR 中所述。

  如果您使用 SSLCRYP 參數的 GSK_PKCS11 版本，則 PKCS #11 記號標籤必須符合您配置的硬體標籤。
- 在 MQCONNX 呼叫中設定 SSL 配置選項結構 MQSCO 的 **CryptoHardware** 欄位。 如需相關資訊，請參閱 MQSCO 概觀。

小心: V 9.2.3 透過 MQSSLCRYP 環境變數或 **SSLCryptoHardware** 屬性提供加密硬體的配置時，您應該在儲存之前保護密碼。 如需相關資訊，請參閱 第 476 頁的『使用加密硬體的 IBM MQ 用戶端』。

如果您已使用下列任何方法來配置使用 PKCS #11 介面的加密硬體，則必須將個人憑證儲存在您所配置加密記號的金鑰資料庫檔中，以便在通道上使用。 這說明於第 263 頁的『在 PKCS #11 硬體上管理憑證』。

ALW 在 *PKCS #11* 硬體上管理憑證

您可以在支援 PKCS #11 介面的加密硬體上管理數位憑證。

關於這項作業

您必須建立金鑰資料庫來準備 IBM MQ 環境，即使您不打算在其中儲存憑證管理中心 (CA) 憑證，但會將所有憑證儲存在加密硬體上。 需要有金鑰資料庫，佇列管理程式才能在其 SSLKEYR 欄位中參照，或用戶端應用程式才能在 MQSSLKEYR 環境變數中參照。 如果您要建立憑證申請，則也需要此金鑰資料庫。

您可以使用指令行或使用 **strmqikm** (iKeyman) 使用者介面來建立金鑰資料庫。

程序

使用指令行建立金鑰資料庫。

1. 執行下列其中一個指令：

   - 使用 **runmqckm**：

     ```
     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 檔名
   指定 CMS 金鑰資料庫的完整檔名，且副檔名必須是 .kdb。

   -pw *password*
   指定 CMS 金鑰資料庫的密碼。

   -type *cms*
   指定資料庫的類型。 (對於 IBM MQ，它必須是 cms。)

   -stash
   將金鑰資料庫密碼儲存至檔案。

   -fips
   指定以 FIPS 模式執行指令。 處於 FIPS 模式時， IBM Crypto for C (ICC) 元件會使用 FIPS 140-2 已驗證的演算法。 如果 ICC 元件未在 FIPS 模式下起始設定，則 **runmqakm** 指令會失敗。

   -強烈
   檢查輸入的密碼是否滿足密碼強度的最低需求。 密碼的最低需求如下：

   - 密碼長度下限必須為 14 個字元。
   - 密碼必須至少包含一個小寫字元、一個大寫字元，以及一個數字或特殊字元。 特殊字元包括星號 (*)、錢幣符號 ($)、數字符號 (#) 及百分比符號 (%)。 空格被分類為特殊字元。
   - 每一個字元在密碼中最多可以出現三次。
   - 密碼中最多可以有兩個連續字元相同。
   - 所有字元都在標準 ASCII 可列印字集內，範圍為 0x20 - 0x7E。

或者，使用 **strmqikm** (iKeyman) 使用者介面來建立金鑰資料庫。

2. 在 AIX and Linux 系統上，以 root 使用者身分登入。 在 Windows 系統上，以管理者身分或 MQM 群組成員身分登入。
3. 開啟 Java 安全內容檔 java.security。

   - 在 AIX and Linux 系統上， Java 安全內容檔位於 IBM MQ 安裝目錄的 java/jre64/jre/lib/security 子目錄中。

- 在 Windows 系統上， Java 安全內容檔位於 IBM MQ 安裝目錄的 `java\jre\lib\security` 子目錄中。

如果檔案中還沒有它，請新增 IBMPKCS11Impl 安全提供者。 例如，新增下列這一行:

```
security.provider.12=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
```

4. 透過執行 **strmqikm** 指令來啟動使用者介面。
5. 按一下 **金鑰資料庫檔 > 開啟**。
6. 按一下 **金鑰資料庫類型** ，然後選取 **PKCS11Direct**。
7. 在 **檔名** 欄位中，輸入用來管理加密硬體的模組名稱; 例如 `PKCS11_API.so`。

   如果使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 **runmqckm** 及 **strmqikm** 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 僅 Windows 及 Linux x86 32 位元平台例外，因為在這些平台， **strmqikm** 和 **runmqckm** 程式都是 32 位元的。
8. 在 **位置** 欄位中，輸入路徑:
   - 例如，在 AIX and Linux 系統上，這可能是 `/usr/lib/pksc11`。
   - 在 Windows 系統上，您可以鍵入媒體庫名稱; 例如 `cryptoki`。

   按一下**確定**。 即會開啟「開啟加密記號」視窗。
9. 選取您要用來儲存憑證的加密裝置記號標籤。
10. 在 **加密記號密碼** 欄位中，鍵入您在配置加密硬體時所設定的密碼。
11. 如果您的加密硬體有能力保留接收或匯入個人憑證所需的簽章者憑證，請清除這兩個次要金鑰資料庫勾選框，並從步驟 第 264 頁的『15』繼續進行。

    如果您需要次要 CMS 金鑰資料庫來保留簽章者憑證，請選取 **開啟現有的次要金鑰資料庫檔** 或 **建立新的次要金鑰資料庫檔**。
12. 在 **檔名** 欄位中，鍵入檔名。 此欄位已包含文字 `key.kdb`。 如果您的詞幹名稱是 `key`，請保留此欄位不變。 如果您指定不同的詞幹名稱，請將 `key` 取代為您的詞幹名稱。 您不得變更 `.kdb` 字尾。
13. 在 **位置** 欄位中，鍵入路徑，例如:
    - 若為佇列管理程式: `/var/mqm/qmgrs/QM1/ssl`
    - 若為 IBM MQ MQI client: `/var/mqm/ssl`

    按一下**確定**。 這時會開啟「密碼提示」視窗。
14. 請輸入密碼。

    如果您在步驟 第 264 頁的『11』中選取 **開啟現有的次要金鑰資料庫檔** ，請在 **密碼** 欄位中輸入密碼。

    如果您在步驟 第 264 頁的『11』中選取 **建立新的次要金鑰資料庫檔** ，請完成下列子步驟:

    a) 在 **密碼** 欄位中鍵入密碼，然後在 **確認密碼** 欄位中再次鍵入密碼。

    b) 選取 **將密碼隱藏至檔案**。 請注意，如果您不隱藏密碼，則嘗試啟動 TLS 通道會失敗，因為它們無法取得存取金鑰資料庫檔所需的密碼。

    c) 按一下**確定**。 即會顯示一個視窗，確認密碼位於檔案 `key.sth` 中 (除非您指定不同的詞幹名稱)。
15. 按一下**確定**。

    即會顯示金鑰資料庫內容頁框。

ALW 要求 *PKCS #11* 硬體的個人憑證

針對佇列管理程式或 IBM MQ MQI client 使用此程序，以要求加密硬體的個人憑證。

關於這項作業

此作業說明如何使用 **strmqikm** 使用者介面來要求個人憑證。 如果您使用指令行介面，請參閱 第 250 頁的『使用指令行』。

註: IBM MQ 不支援 SHA-3 或 SHA-5 演算法。 您可以使用數位簽章演算法名稱 SHA384WithRSA 及 SHA512WithRSA，因為這兩個演算法都是 SHA-2 系列的成員。

數位簽章演算法名稱 SHA3WithRSA 和 SHA5WithRSA 已淘汰，因為它們分別是 SHA384WithRSA 和 SHA512WithRSA 縮寫形式。

程序

若要從 **strmqikm** (iKeyman) 使用者介面要求個人憑證，請完成下列步驟:

1. 完成步驟以使用加密硬體。 請參閱 第 263 頁的『在 PKCS #11 硬體上管理憑證』。
2. 從 **建立** 功能表中，按一下 **新建憑證申請**。
   即會開啟「建立新的金鑰和憑證申請」視窗。
3. 在 **金鑰標籤** 欄位中，輸入憑證標籤。
   標籤是 **CERTLABL** 屬性的值 (如果有設定的話)，或預設 `ibmwebspheremq` 並附加佇列管理程式或 IBM MQ MQI client 登入使用者 ID 的名稱 (全部都是小寫)。 如需詳細資料，請參閱 數位憑證標籤 。
4. 選取您需要的 **金鑰大小** 和 **簽章演算法** 。
5. 輸入 **通用名稱** 和 **組織**的值，然後選取 **國家/地區**。 對於其餘選用欄位，請接受預設值，或鍵入或選取新值。
   請注意，您只能在 **組織單位** 欄位中提供一個名稱。 如需這些欄位的相關資訊，請參閱第 10 頁的『識別名稱』。
6. 在 **輸入要在其中儲存憑證申請的檔案名稱** 欄位中，接受預設值 `certreq.arm`，或鍵入具有完整路徑的新值。
7. 按一下**確定**。
   就會開啟確認視窗。
8. 按一下**確定**。
   個人憑證申請 清單會顯示您所建立之新個人憑證申請的標籤。 憑證申請儲存在您在步驟 第 265 頁的『6』中選擇的檔案中。
9. 透過將檔案傳送至憑證管理中心 (CA)，或將檔案複製到 CA 網站上的要求表單，來要求新的個人憑證。

ALW 將個人憑證接收至 *PKCS #11* 硬體

針對佇列管理程式或 IBM MQ MQI client 使用此程序，以接收加密硬體的個人憑證。

開始之前

新增簽署個人憑證之 CA 的 CA 憑證。 將它新增至加密硬體或次要 CMS 金鑰資料庫。 在您將已簽章的憑證接收到加密硬體之前，請執行此動作。 若要將 CA 憑證新增至金鑰環，請遵循 第 255 頁的『將 CA 憑證或自簽憑證的公用部分新增至 AIX, Linux, and Windows 上的金鑰儲存庫』中的程序。

程序

- 若要使用 **strmqikm** (iKeyman) 使用者介面來接收個人憑證，請完成下列步驟:
  a) 完成步驟以使用加密硬體。 請參閱 第 263 頁的『在 PKCS #11 硬體上管理憑證』。
  b) 按一下 **接收**。 即會開啟「從檔案接收憑證」視窗。
  c) 鍵入新個人憑證的憑證檔名及位置，或按一下 **瀏覽** 以選取名稱及位置。
  d) 按一下**確定**。 如果您在金鑰資料庫中已有個人憑證，則會開啟一個視窗，詢問您是否要將您要新增的金鑰設為資料庫中的預設金鑰。
  e) 請按一下**是**或**否**。 這時會開啟「輸入標籤」視窗。
  f) 按一下**確定**。 **個人憑證** 清單會顯示您新增之個人憑證的標籤。 此標籤是透過在您提供的標籤之前新增加密記號標籤所形成。
- 若要使用 **runmqakm** (GSKCapiCmd) 接收個人憑證，請完成下列步驟:
  a) 開啟針對您環境所配置的指令視窗。
  b) 使用 **runmqakm** (GSKCapiCmd) 指令來接收個人憑證:

```
runmqakm -cert -receive -file filename -crypto module_name
         -tokenlabel hardware_token -pw hardware_password
         -format cert_format -fips
         -secondaryDB filename -secondaryDBpw password
```

其中:

-file *filename*
指定包含個人憑證之檔案的完整檔名。

-crypto *module_name*
指定加密硬體隨附的 PKCS #11 檔案庫的完整名稱。

-tokenlabel *hardware_token*
指定 PKCS #11 加密裝置記號標籤。

-pw *hardware_password*
指定用於存取加密硬體的密碼。

-format *cert_format*
指定憑證格式。 值可以是 `ascii`（代表 Base64 編碼 ASCII）或 `binary`（代表二進位 DER 資料）。 預設值為 ASCII。

-fips
指定以 FIPS 模式執行指令。 處於 FIPS 模式時， IBM Crypto for C (ICC) 元件會使用 FIPS 140-2 已驗證的演算法。 如果 ICC 元件未在 FIPS 模式下起始設定，則 **runmqakm** 指令會失敗。

-secondaryDB 檔案名稱
指定 CMS 金鑰資料庫的完整檔名。

-secondaryDBpw 密碼
指定 CMS 金鑰資料庫的密碼。

MQ Appliance 在 IBM MQ Appliance 上使用 SSL/TLS

IBM MQ Appliance 具有傳輸層安全 (TLS) 支援。

IBM MQ Appliance 具有用於管理憑證的不同指令。 如需憑證管理的詳細資訊，請參閱 IBM MQ Appliance 文件: TLS 憑證管理

z/OS 在 z/OS 上使用 SSL/TLS

本資訊說明如何在 z/OS 上設定及使用「傳輸層安全 (TLS)」。

每一個主題都包括使用 RACF 執行每一個作業的範例。 您可以使用其他外部安全管理程式來執行類似作業。

在 z/OS 上，您還必須設定每一個佇列管理程式用於處理 TLS 呼叫的伺服器子作業數，如 第 267 頁的『在 z/OS 上設定 SSLTASKS 參數』中所述。

z/OS TLS 支援是作業系統不可或缺的部分，稱為 系統 *SSL*。 系統 SSL 是 z/OS 的 Cryptographic Services Base 元素的一部分。 「加密服務基本程式」成員安裝在 *pdsname* 中。 SIEALNKE 分割資料集 (PDS)。 當您安裝 System SSL 時，請確定您選擇適當的選項來提供您需要的 CipherSpecs 。

如果您需要更新自簽憑證，請參閱 在 RACF 中更新自簽憑證的步驟 ，以取得相關資訊。

z/OS *z/OS 上 TLS 的其他使用者 ID 需求*

本資訊說明您的使用者 ID 在 z/OS 上設定及使用 TLS 所需的其他需求。

確保您在系統上具有所有適當的「高影響或通用 (HIPER)」更新項目。

如果金鑰儲存庫是由 CHINIT 使用者 ID 所擁有，則此使用者 ID 需要對 IRR.DIGTCERT.LISTRING 設定檔，以其他方式更新存取權，以及 IRR.DIGTCERT.LIST 設定檔。 視情況使用 PERMIT 指令搭配 ACCESS (UPDATE) 或 ACCESS (READ) 來授與存取權

請確定您已設定下列必要條件:

- *ssidCHIN* 使用者 ID 在 RACF 中已正確定義，且 *ssidCHIN* 使用者 ID 具有下列設定檔的適當存取權:

– IRR.DIGTCERT.LIST
– IRR.DIGTCERT.LISTRING

這些變數定義在 RACF FACILITY 類別中。

- *ssidCHIN* 使用者 ID 是金鑰環的擁有者。
- 佇列管理程式的個人憑證 (如果由 RACDCERT 指令建立) 是以憑證類型使用者 ID 建立，該 ID 也與 *ssidCHIN* 使用者 ID 相同。
- 通道起始程式會回收，或發出 **REFRESH SECURITY TYPE(SSL)** 指令，以挑選您對金鑰環所做的任何變更。
- 「IBM MQ 通道起始程式」程序可透過鏈結清單、LPA 或 STEPLIB DD 陳述式來存取系統 SSL 執行時期程式庫 *pdsname*.SIEALNKE。 此媒體庫必須經過 APF 授權。
- 通道起始程式執行所使用之權限的使用者 ID 配置為使用 z/OS UNIX System Services (z/OS UNIX)，如 z/OS UNIX System Services Planning 說明文件中所述。

  不想要通道起始程式使用訪客/預設 UID 及 OMVS 區段來呼叫 z/OS UNIX 的使用者，只需要根據預設區段來建立新的 OMVS 區段模型，因為通道起始程式不需要特殊許可權，且不會以超級使用者身分在 UNIX 內執行。

  如需一些範例指令，請參閱 第 268 頁的『授與通道起始程式對 z/OS 的正確存取權』。

z/OS 在 *z/OS* 上設定 *SSLTASKS* 參數

使用 ALTER QMGR 指令來設定用於處理 TLS 呼叫的伺服器子作業數

若要使用 TLS 通道，請使用 ALTER QMGR 指令設定 SSLTASKS 參數，以確保至少有兩個伺服器子作業。 例如:

```
ALTER QMGR SSLTASKS(5)
```

為了避免儲存體配置的問題，在沒有「憑證撤銷清冊 (CRL)」檢查的環境中，請勿將 SSLTASKS 屬性設為大於 8 的值。

如果使用 CRL 檢查，則相關通道會在該檢查期間保留 SSLTASK。 因為每一個 SSLTASK 都是 z/OS 作業控制區塊，所以在聯絡相關 LDAP 伺服器時，這可能會經歷很長的經歷時間。

如果您變更 SSLTASKS 屬性的值，則必須重新啟動通道起始程式。

z/OS 在 *z/OS* 上設定金鑰儲存庫

在連線兩端設定金鑰儲存庫。 建立每一個金鑰儲存庫與其佇列管理程式的關聯。

TLS 連線在連線的每一端都需要 金鑰儲存庫 。 每一個佇列管理程式都必須具有金鑰儲存庫的存取權。 在 ALTER QMGR 指令上使用 SSLKEYR 參數，以建立金鑰儲存庫與佇列管理程式的關聯。 如需相關資訊，請參閱 第 21 頁的『SSL/TLS 金鑰儲存庫』。

在 z/OS 上，數位憑證儲存在「外部安全管理程式 (ESM)」所管理的 金鑰環 中。 這些數位憑證具有標籤，可將憑證與佇列管理程式相關聯。 TLS 會使用這些憑證來進行鑑別。 下列所有範例都使用 RACF 指令。 其他 ESM 程式有同等的指令。

在 z/OS 上， IBM MQ 會使用 **CERTLABL** 屬性的值 (如果已設定的話)，或使用預設 `ibmWebSphereMQ` 並附加佇列管理程式名稱。 如需詳細資料，請參閱 數位憑證標籤 。

佇列管理程式的金鑰儲存庫名稱是 RACF 資料庫中的金鑰環名稱。 您可以在建立金鑰環之前或之後指定金鑰環名稱。

使用下列程序來建立佇列管理程式的新金鑰環:

1. 請確定您具有適當的權限來發出 RACDCERT 指令 (如需詳細資料，請參閱 *SecureWay Security Server RACF Command Language Reference*)。
2. 發出下列指令:

   ```
   RACDCERT ID( userid1 ) ADDRING( ring-name )
   ```

其中:

- *userid1* 是通道起始程式位址空間的使用者 ID，或將擁有金鑰環的使用者 ID (如果金鑰環是共用的)。
- *ring-name* 是您要提供給金鑰環的名稱。 此名稱的長度最多可以為 237 個字元。 這個名稱會區分大小寫。 請以大寫字元指定 *ring-name*，以避免發生問題。

z/OS 讓 CA 憑證可供 z/OS 上的佇列管理程式使用

建立金鑰環之後，請將任何相關 CA 憑證連接至該金鑰環。

如果您在資料集中具有 CA 憑證，則必須先使用下列指令將憑證新增至 RACF 資料庫:

```
RACDCERT ID( userid1 ) ADD( input-data-set-name ) WITHLABEL( 'My CA' )
```

然後，若要將 My CA 的 CA 憑證連接至金鑰環，請使用下列指令:

```
RACDCERT ID(userid1)
CONNECT(CERTAUTH LABEL('My CA') RING(ring-name) USAGE(CERTAUTH))
```

其中 *userid1* 是通道起始程式使用者 ID 或共用金鑰環的擁有者。

如需 CA 憑證的相關資訊，請參閱 第 9 頁的『數位憑證』。

z/OS 在 z/OS 上尋找佇列管理程式的金鑰儲存庫

使用此程序來取得佇列管理程式金鑰環的位置。

1. 使用下列其中一個 MQSC 指令，顯示佇列管理程式的屬性:

   ```
   DISPLAY QMGR ALL
   DISPLAY QMGR SSLKEYR
   ```

2. 請檢查指令輸出，以找出金鑰環的位置。

z/OS 在 z/OS 上指定佇列管理程式的金鑰儲存庫位置

若要指定佇列管理程式的金鑰環位置，請使用 ALTER QMGR MQSC 指令來設定佇列管理程式的金鑰儲存庫屬性。

例如:

```
ALTER QMGR SSLKEYR(CSQ1RING)
```

如果金鑰環是由通道起始程式位址空間所擁有，或:

```
ALTER QMGR SSLKEYR(userid1/CSQ1RING)
```

如果它是共用金鑰環，其中 *userid1* 是擁有金鑰環的使用者 ID。

z/OS 授與通道起始程式對 z/OS 的正確存取權

通道起始程式 (CHINIT) 需要存取金鑰儲存庫及某些安全設定檔。

授與 CHINIT 存取權來讀取金鑰儲存庫

如果金鑰儲存庫由 CHINIT 使用者 ID 所擁有，則此使用者 ID 需要對 IRR.DIGTCERT.LISTRING 設定檔，以其他方式更新存取權，以及 IRR.DIGTCERT.LIST 設定檔。 視情況搭配使用 PERMIT 指令與 ACCESS (UPDATE) 或 ACCESS (READ) 來授與存取權:

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID( userid ) ACCESS(UPDATE)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID( userid ) ACCESS(READ)
```

其中 *userid* 是通道起始程式位址空間的使用者 ID。

授與 CHINIT 對適當 CSF* 設定檔的讀取權

如需透過要使用的「整合加密服務機能 (ICSF)」提供的硬體支援，請使用下列指令，確定您的 CHINIT 使用者 ID 具有 CSFSERV 類別中適當 CSF* 設定檔的讀取權:

```
PERMIT csf-resource CLASS(CSFSERV) ID( userid ) ACCESS(READ)
```

其中 *csf-resource* 是 CSF* 設定檔的名稱，而 *userid* 是通道起始程式位址空間的使用者 ID。

針對下列每一個 CSF* 設定檔，重複此指令:

- CSFDSG
- CSFDSV
- CSFPKD
- CSFPKE
- CSFPKI

您的 CHINIT 使用者 ID 可能也需要其他 CSF* 設定檔的讀取權。 例如，如果您使用 ECDHE_RSA_AES_256_GCM_SHA384 密碼規格，則您的 CHINIT 使用者 ID 也需要下列 CSF* 設定檔的讀取權:

- CSF1DVK
- CSF1GAV
- CSF1GKP
- CSF1SKE
- CSF1TRC
- CSF1TRD

如需相關資訊，請參閱 RACF CSFSERV 資源需求。

如果您的憑證金鑰儲存在 ICSF 中，且您的安裝架構已建立對 ICSF 中所儲存之金鑰的存取控制，請使用下列指令，確定您的 CHINIT 使用者 ID 具有 CSFKEYS 類別中設定檔的讀取權:

```
PERMIT IRR.DIGTCERT. userid.* CLASS(CSFKEYS) ID( userid ) ACCESS(READ)
```

其中 *userid* 是通道起始程式位址空間的使用者 ID。

使用「整合加密服務機能 (ICSF)」

如果未使用 TLS，當植入密碼保護演算法來模糊化流經用戶端通道的密碼時，通道起始程式可以使用 ICSF 來產生亂數。

如需進一步資訊，請參閱 第 221 頁的『使用「整合加密服務機能 (ICSF)」』

z/OS 當憑證或金鑰儲存庫的變更在 *z/OS* 上生效時

當通道起始程式啟動或儲存庫重新整理時，變更會生效。

具體而言，金鑰環中的憑證及金鑰儲存庫屬性的變更會在下列任一情況下生效:

- 啟動或重新啟動通道起始程式時。
- 當發出 REFRESH SECURITY TYPE (SSL) 指令來重新整理金鑰儲存庫的內容時。

z/OS 在 *z/OS* 上建立自簽個人憑證

使用此程序來建立自簽個人憑證。

1. 使用下列指令產生憑證及公開和私密金鑰組:

```
RACDCERT ID(userid2) GENCERT
SUBJECTSDN(CN('common-name')
           T('title')
           OU('organizational-unit')
```

```
           O('organization')
           L('locality')
           SP('state-or-province')
           C('country'))
WITHLABEL('label-name')
```

2. 使用下列指令將憑證連接至金鑰環:

```
RACDCERT ID(userid1)
CONNECT(ID(userid2) LABEL('label-name') RING(ring-name) USAGE(PERSONAL))
```

其中:

- *userid1* 是通道起始程式位址空間的使用者 ID 或共用金鑰環的擁有者。
- *userid2* 是與憑證相關聯的使用者 ID，且必須是通道起始程式位址空間的使用者 ID。

  userid1 和 *userid2* 可以是相同的 ID。
- *ring-name* 是您在 第 267 頁的『在 z/OS 上設定金鑰儲存庫』中提供的金鑰環名稱。
- *label-name* 必須是 IBM MQ **CERTLABL** 屬性的值 (如果有設定的話)，或是附加佇列管理程式名稱的預設 `ibmWebSphereMQ`。 如需詳細資料，請參閱 數位憑證標籤 。

z/OS 在 *z/OS* 上要求個人憑證

使用 RACF 來申請個人憑證。

若要申請個人憑證，請使用 RACF，如下所示:

1. 建立自簽個人憑證，例如 第 269 頁的『在 z/OS 上建立自簽個人憑證』。 此憑證為要求提供「識別名稱」的屬性值。
2. 使用下列指令，建立寫入資料集的 PKCS #10 Base64-encoded 憑證要求:

   ```
   RACDCERT ID(userid2) GENREQ(LABEL(' label_name ')) DSN(' output_data_set_name ')
   ```

   其中

   - *userid2* 是與憑證相關聯的使用者 ID，必須是通道起始程式位址空間的使用者 ID
   - *label_name* 是建立自簽憑證時使用的標籤

   如需詳細資料，請參閱 第 22 頁的『數位憑證標籤，瞭解需求』。
3. 將資料集傳送至憑證管理中心 (CA)，以要求新的個人憑證。
4. 當憑證管理中心將已簽章的憑證傳回給您時，請使用原始標籤將憑證新增回 RACF 資料庫，如 第 271 頁的『將個人憑證新增至 z/OS 上的金鑰儲存庫』中所述。

z/OS 建立 *RACF* 簽署的個人憑證

RACF 可以作為憑證管理中心，並發出其自己的 CA 憑證。

本節使用術語 簽章者憑證 來表示 RACF 發出的 CA 憑證。

在執行下列程序之前，簽章者憑證的私密金鑰必須位於 RACF 資料庫中:

1. 使用下列指令，利用 RACF 資料庫中包含的簽章者憑證來產生 RACF 所簽署的個人憑證:

```
RACDCERT ID(userid2) GENCERT
SUBJECTSDN(CN('common-name')
           T('title')
           OU('organizational-unit')
           O('organization')
           L('locality')
           SP('state-or-province')
           C('country'))
WITHLABEL('label-name')
SIGNWITH(CERTAUTH LABEL('signer-label'))
```

2. 使用下列指令將憑證連接至金鑰環:

```
RACDCERT ID(userid1)
CONNECT(ID(userid2) LABEL('label-name') RING(ring-name) USAGE(PERSONAL))
```

其中:

- *userid1* 是通道起始程式位址空間的使用者 ID 或共用金鑰環的擁有者。
- *userid2* 是與憑證相關聯的使用者 ID，且必須是通道起始程式位址空間的使用者 ID。

  userid1 和 *userid2* 可以是相同的 ID。
- *ring-name* 是您在 第 267 頁的『在 z/OS 上設定金鑰儲存庫』中提供的金鑰環名稱。
- *label-name* 必須是 IBM MQ **CERTLABL** 屬性的值 (如果有設定的話)，或是附加佇列管理程式或佇列共用群組名稱的預設 ibmWebSphereMQ。 如需詳細資料，請參閱 數位憑證標籤。
- *signer-label* 是您自己的簽章者憑證的標籤。

z/OS 將個人憑證新增至 *z/OS* 上的金鑰儲存庫

使用此程序將個人憑證新增或匯入至金鑰環。

在憑證管理中心傳送新的個人憑證給您之後，請使用下列程序將它新增至金鑰環:

1. 使用下列指令，將憑證新增至 RACF 資料庫:

   ```
   RACDCERT ID( userid2 ) ADD( input-data-set-name ) WITHLABEL(' label-name ')
   ```
2. 使用下列指令將憑證連接至金鑰環:

   ```
   RACDCERT ID( userid1 )
   CONNECT(ID( userid2 ) LABEL(' label-name ') RING( ring-name ) USAGE(PERSONAL))
   ```

其中:

- *userid1* 是通道起始程式位址空間的使用者 ID 或共用金鑰環的擁有者。
- *userid2* 是與憑證相關聯的使用者 ID，且必須是通道起始程式位址空間的使用者 ID。
- *ring-name* 是您在 第 267 頁的『在 z/OS 上設定金鑰儲存庫』中提供的金鑰環名稱。
- *input-data-set-name* 是包含 CA 簽章憑證的資料集名稱。 資料集必須已編目，且不能是 PDS 或 PDS 的成員。 RACDCERT 預期的記錄格式 (RECFM) 是 VB。 RACDCERT 會動態配置及開啟資料集，並將其中的憑證讀取為二進位資料。
- *label-name* 是您建立原始要求時使用的標籤名稱。 它必須是 IBM MQ **CERTLABL** 屬性的值 (如果已設定的話)，或預設 ibmWebSphereMQ 並附加佇列管理程式或佇列共用群組的名稱。 如需詳細資料，請參閱 數位憑證標籤。

z/OS 從 *z/OS* 上的金鑰儲存庫匯出個人憑證

使用 RACDCERT 指令匯出憑證。

在您要從中匯出憑證的系統上，使用下列指令:

```
RACDCERT ID(userid2) EXPORT(LABEL('label-name'))
DSN(output-data-set-name) FORMAT(CERTB64)
```

其中:

- *userid2* 是用來將憑證新增至金鑰環的使用者 ID。
- *label-name* 是您要擷取之憑證的標籤。
- *output-data-set-name* 是憑證放置所在的資料集。
- CERTB64 是採用 Base64 格式的 DER 編碼 X.509 憑證。 您可以選擇替代格式，例如:

  CERTDER
  DER 編碼二進位格式的 X.509 憑證

PKCS12B64
Base64 格式的 PKCS #12 憑證

PKCS12DER
二進位格式的 PKCS #12 憑證

z/OS 從 *z/OS* 上的金鑰儲存庫中刪除個人憑證

使用 RACDCERT 指令刪除個人憑證。

在刪除個人憑證之前，您可能想要儲存其副本。 若要在刪除資料集之前將個人憑證複製到資料集，請遵循 第 271 頁的『從 z/OS 上的金鑰儲存庫匯出個人憑證』中的程序。 然後使用下列指令來刪除您的個人憑證:

```
RACDCERT ID( userid2 ) DELETE(LABEL(' label-name '))
```

其中:

- *userid2* 是用來將憑證新增至金鑰環的使用者 ID。
- *label-name* 是您要刪除的憑證名稱。

z/OS 在 *z/OS* 上重新命名金鑰儲存庫中的個人憑證

使用 RACDCERT 指令重新命名憑證。

如果您不想要找到具有特定標籤的憑證，但不想刪除該憑證，則可以使用下列指令暫時重新命名該憑證:

```
RACDCERT ID( userid2 ) LABEL(' label-name ') NEWLABEL(' new-label-name ')
```

其中:

- *userid2* 是用來將憑證新增至金鑰環的使用者 ID。
- *label-name* 是您要重新命名的憑證名稱。
- *new-label-name* 是憑證的新名稱。

這在測試 TLS 用戶端鑑別時非常有用。

z/OS 在 *z/OS* 上建立使用者 *ID* 與數位憑證的關聯

IBM MQ 可以使用與 RACF 憑證相關聯的使用者 ID 作為通道使用者 ID。 將使用者 ID 與憑證建立關聯，方法是在該使用者 ID 下安裝憑證，或使用「憑證名稱過濾器」。

本主題中說明的方法是將使用者 ID 與使用通道鑑別記錄的數位憑證相關聯的獨立式平台方法的替代方案。 如需通道鑑別記錄的相關資訊，請參閱 第 40 頁的『通道鑑別記錄』。

當 TLS 通道一端的實體從遠端連線接收憑證時，該實體會詢問 RACF 是否有與該憑證相關聯的使用者 ID。 實體會使用該使用者 ID 作為通道使用者 ID。 如果沒有與憑證相關聯的使用者 ID，則實體會使用通道起始程式執行時所使用的使用者 ID。

使用下列其中一種方式，將使用者 ID 與憑證相關聯:

- 使用您要與該憑證相關聯的使用者 ID，將該憑證安裝至 RACF 資料庫，如 第 271 頁的『將個人憑證新增至 z/OS 上的金鑰儲存庫』中所述。
- 使用「憑證名稱過濾器 (CNF)」，將憑證主旨或發證者的「識別名稱」對映至使用者 ID，如 第 272 頁的『在 z/OS 上設定憑證名稱過濾器』中所述。

z/OS 在 *z/OS* 上設定憑證名稱過濾器

請利用 RACDCERT 指令來定義憑證名稱過濾器 (CNF)，它會將「識別名稱」對映至使用者 ID。

請執行下列步驟來設定 CNF。

1. 使用下列指令啟用 CNF 函數。 您需要類別 DIGTNMAP 的更新權限才能執行此動作。

   ```
   SETROPTS CLASSACT(DIGTNMAP) RACLIST(DIGTNMAP)
   ```

2. 定義 CNF。 例如:

```
RACDCERT ID(USER1) MAP WITHLABEL('filter1') TRUST
SDNFILTER('O=IBM.C=UK') IDNFILTER('O=ExampleCA.L=Internet')
```

其中 USER1 是在下列情況下要使用的使用者 ID:

- 主體的 DN 具有 IBM 組織及 UK 國家/地區。
- 發證者的 DN 具有 ExampleCA 的「組織」及 Internet 的「地區」。

3. 重新整理 CNF 對映:

```
SETROPTS RACLIST(DIGTNMAP) REFRESH
```

註:

1. 如果實際憑證儲存在 RACF 資料庫中，則安裝它時所使用的使用者 ID 優先於與任何 CNF 相關聯的使用者 ID。 如果憑證未儲存在 RACF 資料庫中，則會使用與最明確相符 CNF 相關聯的使用者 ID。 主體 DN 的相符項會被視為比發證者 DN 的相符項更具體。
2. 除非您重新整理 CNF 對映，否則不會套用 CNF 的變更。
3. 只有在 DN 過濾器與 DN 的 最低有效部分 相同時， DN 才會符合 CNF 中的 DN 過濾器。 DN 的最低有效部分包含通常列在 DN 最右邊，但出現在憑證開頭的屬性。

   例如，考量 SDNFILTER 'O=IBM.C=UK'。 'CN=QM1.O=IBM.C=UK' 的主體 DN 符合該過濾器，但 'CN=QM1.O=IBM.L=Hursley.C=UK' 的主體 DN 不符合該過濾器。

   部分憑證的最低有效部分可能包含不符合 DN 過濾器的欄位。 請考慮在 DEFINE CHANNEL 指令上的 SSLPEER 型樣中指定 DN 型樣，以排除這些憑證。
4. 如果將最符合的 CNF 定義為 RACF NOTRUST ，則實體會使用通道起始程式執行時所使用的使用者 ID。
5. RACF 使用 '.' 字元作為分隔字元。 IBM MQ 使用逗點或分號。

您可以定義 CNF ，以確保實體永不將通道使用者 ID 設為預設值，即通道起始程式執行時所使用的使用者 ID。 針對與實體相關聯之金鑰環中的每一個 CA 憑證，定義具有完全符合該 CA 憑證之主體 DN 的 IDNFILTER 的 CNF。 這可確保實體可能使用的所有憑證至少符合其中一個 CNF。 這是因為所有此類憑證都必須連接至與實體相關聯的金鑰環，或必須由憑證連接至與實體相關聯的金鑰環的 CA 發出。

如需您用來操作 CNF 之指令的相關資訊，請參閱 *SecureWay Security Server RACF Security Administrator 's Guide* 。

z/OS 在 *z/OS* 上的 *QMA* 上定義傳送端通道及傳輸佇列

使用 **DEFINE CHANNEL** 和 **DEFINE QLOCAL** 指令來設定必要的物件。

程序

在 QMA 上，發出類似下列範例的指令:

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QMB.MACH.COM) XMITQ(QMB)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256) DESCR('Sender channel using TLS from QMA to QMB')

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

結果

傳送端通道 TO.QMB 及傳輸佇列 QMB。

z/OS 在 *z/OS* 上的 *QMB* 上定義接收端通道

使用 **DEFINE CHANNEL** 指令來設定必要的物件。

程序

在 QMB 上，發出類似下列範例的指令:

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using TLS to QMB')
```

結果
接收端通道 TO.QMB。

z/OS 在 *z/OS* 上啟動 *QMA* 上的傳送端通道
必要的話，請啟動接聽器程式並重新整理安全。 然後使用 **START CHANNEL** 指令啟動通道。

程序
1. 選擇性的: 如果您尚未這樣做，請在 QMB 上啟動接聽器程式。
   接聽器程式會接聽送入的網路要求，並在需要時啟動接收端通道。 如需如何啟動接聽器的相關資訊，請參閱 啟動通道接聽器。
2. 選擇性的: 如果先前已執行任何 SSL/TLS 通道，請發出指令 REFRESH SECURITY TYPE(SSL)。
   這可確保對金鑰儲存庫所做的所有變更都可供使用。
3. 使用指令 START CHANNEL(TO.QMB) 在 QMA 上啟動通道。

結果
傳送端通道已啟動。

z/OS 在 *z/OS* 上交換自簽憑證
交換您先前擷取的憑證。 如果您使用 FTP，請使用正確的格式。

程序
將 QM1 憑證的 CA 部分傳送至 QM2 系統，反之亦然，例如，透過 FTP。

如果您使用 FTP 來傳送憑證，則必須以正確的格式來執行。

以 *binary* 格式傳送下列憑證類型:

- DER 編碼二進位 X.509
- PKCS #7 (CA 憑證)
- PKCS #12 (個人憑證)

以 ASCII 格式傳送下列憑證類型:

- PEM (隱私權-加強郵件)
- Base64 編碼 X.509

z/OS 在 *z/OS* 的 *QM1* 上定義傳送端通道及傳輸佇列
使用 **DEFINE CHANNEL** 和 **DEFINE QLOCAL** 指令來設定必要的物件。

程序
在 QM1 上，發出類似下列範例的指令:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QM1.MACH.COM) XMITQ(QM2)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA) DESCR('Sender channel using TLS from QM1 to QM2')

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

通道每一端的 CipherSpecs 必須相同。

如果您想要通道使用 TLS，則只有 SSLCIPH 參數是必要的。 如需 SSLCIPH 參數允許值的相關資訊，請參閱 第 33 頁的『IBM MQ 中的 CipherSpecs 和 CipherSuites』。

結果
傳送端通道 QM1.TO.QM2 及傳輸佇列 QM2。

z/OS 在 *z/OS* 上的 *QM2* 上定義接收端通道
使用 **DEFINE CHANNEL** 指令來設定必要的物件。

程序

在 QM2 上，發出類似下列範例的指令:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using TLS from QM1 to QM2')
```

通道必須與您在 第 274 頁的『在 z/OS 的 QM1 上定義傳送端通道及傳輸佇列』中定義的傳送端通道同名，並使用相同的 CipherSpec。

z/OS 在 *z/OS* 上的 *QM1* 上啟動傳送端通道
必要的話，請啟動接聽器程式並重新整理安全。 然後使用 **START CHANNEL** 指令啟動通道。

程序

1. 選擇性的: 如果您尚未這樣做，請在 QM2 上啟動接聽器程式。
   接聽器程式會接聽送入的網路要求，並在需要時啟動接收端通道。 如需如何啟動接聽器的相關資訊，請參閱 啟動通道接聽器
2. 選擇性的: 如果先前已執行任何 SSL/TLS 通道，請發出指令 REFRESH SECURITY TYPE (SSL)。
   這可確保對金鑰儲存庫所做的所有變更都可供使用。
3. 在 QM1 上，使用指令 `START CHANNEL(QM1.TO.QM2)` 啟動通道。

結果
傳送端通道已啟動。

z/OS 在 *z/OS* 上重新整理 *SSL* 或 *TLS* 環境
使用 **REFRESH SECURITY** 指令重新整理佇列管理程式 QMA 上的 TLS 環境。

程序

在 QMA 上，輸入下列指令:

```
REFRESH SECURITY TYPE(SSL)
```

這可確保對金鑰儲存庫所做的所有變更都可供使用。

z/OS 在 *z/OS* 的接收端通道上容許匿名連線
使用 **ALTER CHANNEL** 指令，將 SSL 或 TLS 用戶端鑑別設為選用。

程序

在 QMB 上，輸入下列指令:

```
ALTER CHANNEL(TO.QMB) CHLTYPE(RCVR) SSLCAUTH(OPTIONAL)
```

z/OS 在 *z/OS* 上的 *QM1* 上啟動傳送端通道
必要的話，請啟動通道起始程式，啟動接聽器程式，並重新整理安全。 然後使用 **START CHANNEL** 指令啟動通道。

程序

1. 選擇性的: 如果您尚未這麼做，請啟動通道起始程式。
2. 選擇性的: 如果您尚未這樣做，請在 QM2 上啟動接聽器程式。

   接聽器程式會接聽送入的網路要求，並在需要時啟動接收端通道。 如需如何啟動接聽器的相關資訊，請參閱 啟動通道接聽器
3. 選擇性的: 如果通道起始程式已在執行中或先前已執行任何 SSL/TLS 通道，請發出指令 REFRESH SECURITY TYPE (SSL)。

   這可確保對金鑰儲存庫所做的所有變更都可供使用。
4. 在 QM1 上，使用指令 `START CHANNEL(QM1.TO.QM2)` 啟動通道。

結果

傳送端通道已啟動。

z/OS 在 *z/OS* 上啟動 *QMA* 上的傳送端通道

必要的話，請啟動通道起始程式，啟動接聽器程式，並重新整理安全。 然後使用 **START CHANNEL** 指令啟動通道。

程序

1. 選擇性的: 如果您尚未這麼做，請啟動通道起始程式。
2. 選擇性的: 如果您尚未這樣做，請在 QMB 上啟動接聽器程式。

   接聽器程式會接聽送入的網路要求，並在需要時啟動接收端通道。 如需如何啟動接聽器的相關資訊，請參閱 啟動通道接聽器。
3. 選擇性的: 如果通道起始程式已在執行中，或先前已執行任何 SSL/TLS 通道，請發出指令 REFRESH SECURITY TYPE(SSL)。

   這可確保對金鑰儲存庫所做的所有變更都可供使用。
4. 使用指令 `START CHANNEL(TO.QMB)` 在 QMA 上啟動通道。

結果

傳送端通道已啟動。

z/OS 在 *z/OS* 上修改橢圓曲線索引鍵長度

如何修改 GSK_CLIENT_ECURVE_LIST 環境變數，以將用戶端指定的橢圓曲線或受支援群組清單設為由一或多個 4 字元值組成的字串 (依使用喜好設定順序)。

重要: 在使用 TLS 1.2 或 TLS1.0 協商連接時，您必須套用 z/OS APAR OA61783 中的修復程序，以允許作業系統使某些橢圓曲線生效。

您可以使用 CEEOPTS DD 陳述式，在通道起始程式啟動 JCL 中設定此 TLS 環境變數:

```
CEEOPTS DD DSN=<dataset-name>,DISP=SHR
```

在上面參照的資料集中，指定您要使用的清單，例如:

```
ENVAR("GSK_CLIENT_ECURVE_LIST=002300240025")
```

重要: 請勿將此 CEEOPTS 陳述式與串流中資料搭配使用，因為這會防止針對使用該陳述式的所有 TLS 作業設定環境變數。

請確定您參照循序資料集或分割的資料集成員，以便在使用大於 1 的 SSLTASKS 值時能夠運作。

您也可以使用 GSK_CLIENT_ECURVE_LIST 的伺服器模擬對等項目，即 GSK_SERVER_ALLOWED_KEX_ECURVES。 如需相關資訊，請參閱 限制金鑰交換橢圓曲線 。

此外，請參閱 密碼組合定義 中的表 5 ，以取得有效 4 個字元橢圓曲線及受支援群組規格的清單。

預設規格為 00210023002400250019。 如果已啟用 TLS V1.3，則 0029 (x25519) 會附加至預設清單的結尾。

識別及鑑別使用者

您可以使用 X.509 憑證、MQCSP 結構或數種類型的使用者結束程式來識別及鑑別使用者。

使用 X.509 憑證

您可以使用 x.509 憑證搭配 **CHLAUTH** 指令和 **SSLPEER** 參數來識別及鑑別使用者。 **SSLPEER** 參數指定過濾器，用於與通道另一端同層級佇列管理程式或用戶端中憑證的「主體識別名稱」進行比較。

如需使用 **CHLAUTH** 指令及 **SSLPEER** 參數的相關資訊，請參閱 SET CHLAUTH。

使用 MQCSP 結構

您可以在 MQCONNX 呼叫中指定 MQCSP 連線安全參數結構; 此結構包含使用者 ID 及密碼。 必要的話，您可以在安全結束程式中變更 MQCSP。

註: 物件權限管理程式 (OAM) 不使用密碼。 不過， OAM 會對使用者 ID 執行一些有限的工作，這可能被視為一種瑣碎的鑑別形式。 如果您在應用程式中使用那些參數，則這些檢查會停止您採用另一個使用者 ID。

警告: 在某些情況下，用戶端應用程式的 MQCSP 結構中的密碼將透過網路以純文字傳送。 若要確保用戶端應用程式密碼受到適當保護，請參閱 第 25 頁的『MQCSP 密碼保護』。

在安全結束程式中實作識別及鑑別

安全結束程式的主要目的是在通道的每一端啟用 MCA 來鑑別其夥伴。 在訊息通道的每一端，以及在 MQI 通道的伺服器端， MCA 通常會代表它所連接的佇列管理程式執行動作。 在 MQI 通道的用戶端端， MCA 通常會代表 IBM MQ 用戶端應用程式的使用者執行動作。 在此狀況下，實際上會在兩個佇列管理程式之間進行交互鑑別，或在佇列管理程式與 IBM MQ MQI client 應用程式的使用者之間進行交互鑑別。

提供的安全結束程式 (SSPI 通道結束程式) 說明如何透過交換由授信鑑別伺服器 (例如 Kerberos) 產生然後檢查的鑑別記號，來實作交互鑑別。 如需詳細資料，請參閱第 128 頁的『Windows 上的 SSPI 通道結束程式』。

也可以使用「公開金鑰基礎架構 (PKI)」技術來實作交互鑑別。 每一個安全結束程式都會產生一些隨機資料，使用它所代表的佇列管理程式或使用者的私密金鑰來簽署它，並在安全訊息中將簽署的資料傳送給它的夥伴。 夥伴安全結束程式會使用佇列管理程式或使用者的公開金鑰來檢查數位簽章，以執行鑑別。 在交換數位簽章之前，如果有多個演算法可供使用，安全結束程式可能需要同意產生訊息摘要的演算法。

當安全結束程式將已簽署的資料傳送至其夥伴時，它也需要傳送一些方法來識別它所代表的佇列管理程式或使用者。 這可能是「識別名稱」，甚至是數位憑證。 如果傳送數位憑證，夥伴安全結束程式可以透過主要 CA 憑證的憑證鏈來驗證憑證。 這可確保用來檢查數位簽章之公開金鑰的所有權。

夥伴安全結束程式只有在能夠存取包含憑證鏈中其餘憑證的金鑰儲存庫時，才能驗證數位憑證。 如果未傳送佇列管理程式或使用者的數位憑證，則必須在夥伴安全結束程式具有存取權的金鑰儲存庫中提供。 除非夥伴安全結束程式可以找到簽章者的公開金鑰，否則無法檢查數位簽章。

「傳輸層安全 (TLS)」使用 PKI 技術，如剛才說明的技術。 如需 TLS 如何執行鑑別的相關資訊，請參閱 第 13 頁的『傳輸層安全 (TLS) 概念』。

如果無法使用授信鑑別伺服器或 PKI 支援，則可以使用其他技術。 一般技術 (可在安全結束程式中實作) 使用對稱金鑰演算法。

其中一個安全結束程式 (結束程式 A) 會產生亂數，並以安全訊息將它傳送至其夥伴安全結束程式 (結束程式 B)。 結束程式 B 會使用只有兩個安全結束程式已知的金鑰副本來加密數字。 結束程式 B 會使用結束程式 B 已產生的第二個亂數，將加密號碼傳送至安全訊息中的結束程式 A。 結束程式 A 會驗證第一個亂數是否已正確加密，使用其金鑰副本來加密第二個亂數，並將已加密的數字傳送至安全訊息中的結束程式 B。 然後，結束程式 B 會驗證第二個亂數是否已正確加密。 在此交換期間，如果任一安全結束程式不滿意其他安全結束程式的確實性，則可以指示 MCA 關閉通道。

此技術的優點是在交換期間不會透過通訊連線傳送金鑰或密碼。 缺點是它無法提供如何以安全方式配送共用金鑰的問題解決方案。 第 390 頁的『在使用者結束程式中實作機密性』中說明此問題的一個解決方案。 當兩個 LU 連結以形成階段作業時，在 SNA 中使用類似的技術來進行兩個 LU 的交互鑑別。 該技術在 第 98 頁的『階段作業層次鑑別』中有說明。

所有先前用於交互鑑別的技術都可以調整為提供單向鑑別。

在訊息結束程式中實作識別及鑑別

當應用程式將訊息放入佇列時，訊息描述子中的 *UserIdentifier* 欄位會包含與應用程式相關聯的使用者 ID。 不過，沒有可用來鑑別使用者 ID 的資料。 此資料可以由通道傳送端的訊息結束程式新增，並由通道接收端的訊息結束程式檢查。 例如，鑑別資料可以是加密密碼或數位簽章。

如果在應用程式層次實作此服務，則它可能更有效。 基本需求是接收訊息之應用程式的使用者能夠識別及鑑別傳送訊息之應用程式的使用者。 因此，我們自然會考慮在應用層面推行這項服務。 如需相關資訊，請參閱第 281 頁的『API 結束程式和 API 交互結束程式中的身分對映』。

在 API 結束程式和 API 交互結束程式中實作識別和鑑別

在個別訊息的層次上，識別及鑑別是涉及兩個使用者的服務，即訊息的傳送端及接收端。 基本需求是接收訊息之應用程式的使用者能夠識別及鑑別傳送訊息之應用程式的使用者。 請注意，需求是單向 (而非雙向) 鑑別。

視實作方式而定，使用者及其應用程式可能需要與服務互動，甚至需要與服務互動。 此外，何時及如何使用服務可能取決於使用者及其應用程式所在的位置，以及應用程式本身的本質。 因此，考慮在應用程式層次而非鏈結層次實作服務是很自然的。

如果您考慮在鏈結層次實作此服務，則可能需要解決如下所示的問題:

- 在訊息通道上，如何只將服務套用至那些需要它的訊息?
- 如果這是需求，您如何讓使用者及其應用程式與服務互動?
- 在多躍點狀況下，在傳送訊息至目的地的途中，會透過多個訊息通道傳送訊息，您在何處呼叫服務的元件?

以下是一些範例，說明如何在應用程式層次實作識別及鑑別服務。 術語 *API* 結束程式 表示 API 結束程式或 API 交互結束程式。

- 當應用程式將訊息放入佇列時， API 結束程式可以從授信鑑別伺服器 (例如 Kerberos) 獲得鑑別記號。 API 結束程式可以將此記號新增至訊息中的應用程式資料。 當接收端應用程式擷取訊息時，第二個 API 結束程式可以檢查記號，要求鑑別伺服器鑑別傳送端。
- 當應用程式將訊息放入佇列時， API 結束程式可以將下列項目附加至訊息中的應用程式資料:
  - 傳送端的數位憑證
  - 傳送者的數位簽章

  如果用於產生訊息摘要的不同演算法可供使用，則 API 結束程式可以包括它所使用的演算法名稱。

  當接收端應用程式擷取訊息時，第二個 API 結束程式可以執行下列檢查:

  - API 結束程式可以透過主要 CA 憑證的憑證鏈來驗證數位憑證。 若要這樣做， API 結束程式必須有權存取包含憑證鏈中其餘憑證的金鑰儲存庫。 此檢查可確保由「識別名稱」識別的傳送者是憑證中所包含公開金鑰的真正擁有者。
  - API 結束程式可以使用憑證中包含的公開金鑰來檢查數位簽章。 此檢查會鑑別寄件者。

  可以傳送寄件者的「識別名稱」，而不是整個數位憑證。 在此情況下，金鑰儲存庫必須包含傳送端的憑證，以便第二個 API 結束程式可以找到傳送端的公開金鑰。 另一種可能是傳送憑證鏈中的所有憑證。
- 當應用程式將訊息放入佇列時，訊息描述子中的 *UserIdentifier* 欄位會包含與應用程式相關聯的使用者 ID。 使用者 ID 可用來識別寄件者。 若要啟用鑑別， API 結束程式可以將部分資料 (例如加密密碼) 附加至訊息中的應用程式資料。 當接收端應用程式擷取訊息時，第二個 API 結束程式可以使用隨訊息一起傳送的資料來鑑別使用者 ID。

  對於源自受控制及授信環境的訊息，以及在無法使用授信鑑別伺服器或 PKI 支援的情況下，此技術可能被視為已足夠。

外掛鑑別方法 (PAM)

Linux AIX

PAM 現在在 UNIX and Linux 平台上很常見，並提供一般機制來隱藏服務的使用者鑑別詳細資料。

透過配置規則，可以將不同的鑑別規則用於不同的服務，而無需對服務本身進行任何變更。

如需進一步資訊，請參閱第 291 頁的『使用外掛鑑別方法 (PAM)』。

特許使用者

特許使用者是對 IBM MQ 具有完整管理權限的使用者。

除了下表列出的使用者之外，授與存取權時必須格外小心的某些物件及授權，以確保佇列管理程式的完整性及安全。 授與下列任何授權時，必須套用額外的審查:

- 對 SYSTEM 物件的任何授權
- 建立、變更及刪除物件的管理授權。

  z/OS 在 z/OS 上，此授權是指令安全及指令資源安全權限，可發出 DEFINE、ALTER 及 DELETE 指令。

  Multi 在所有其他平台上，這些授權是管理授權，例如 `+crt`、 `+chg` 及 `+dlt`。
- 用於清除佇列的管理授權。

  z/OS 在 z/OS 上，此授權是發出 CLEAR 指令的指令安全及指令資源安全權限。

  Multi 在所有其他平台上，此授權為 `+clr`。
- 停止通道、取消或確定訊息的管理授權。

  z/OS 在 z/OS 上，此授權是指令安全及指令資源安全權限，可發出 RESET CHANNEL、START CHANNEL 及 STOP CHANNEL 等指令。

  Multi 在所有其他平台上，這些授權為 `+ctrl` 及 `+ctrlx`。
- 替代使用者 MQI 授權，可讓應用程式提升授權檢查的專用權。

  z/OS 在 z/OS 上，此授權是授與替代使用者安全設定檔的任何權限。

  Multi 在所有其他平台上，此授權為 `+altusr`。
- 容許應用程式變更訊息安全環境定義的環境定義授權。

  z/OS 在 z/OS 上，此授權是授與環境定義安全設定檔的任何權限。

  Multi 在所有其他平台上，這些授權為 `+setall` 及 `+setid`。

作為一般主體，傳訊應用程式只應該獲得所需佇列或主題的基本 MQI 授權。 在非特許 MCAUSER 及某些其他特殊類型的應用程式 (例如無法傳送郵件的佇列處理程式) 下執行的 MCA 通道，可能需要通常未授與應用程式正確運作的其他授權。

表 67: 依平台列出特許使用者

平台	特許使用者
Windows 系統	• 系統 • mqm 群組的成員 • 「管理者」群組的成員
AIX and Linux 系統	• mqm 群組的成員

表 67: 依平台列出特許使用者 (繼續)	
平台	**特許使用者**
IBM i IBM i 系統	• 設定檔 qmqm 及 qmqmadm • qmqmadm 群組的所有成員 • 使用 *ALLOBJ 設定定義的任何使用者
z/OS	執行通道起始程式、佇列管理程式及進階訊息安全位址空間的使用者 ID。 這些使用者 ID 不會自動具有 IBM MQ 的完整管理權限，但會因為通常授與這些使用者 ID 的存取層次而被視為特許。

使用 MQCSP 結構來識別及鑑別使用者

您可以指定 MQCONNX 呼叫的 MQCSP 連線安全參數結構。

MQCSP 連線安全參數結構包含使用者 ID 及密碼，授權服務可用來識別及鑑別使用者。

您可以在安全結束程式中變更 MQCSP。

警告: 在某些情況下，用戶端應用程式的 MQCSP 結構中的密碼將透過網路以純文字傳送。 若要確保用戶端應用程式密碼受到適當保護，請參閱 第 25 頁的『MQCSP 密碼保護』。

MQCSP 與 AdoptCTX 設定之間的關係

除非未啟用連線鑑別特性，否則 IBM MQ 一律會鑑別透過 MQCSP 結構傳遞的認證。 順利鑑別認證之後，除非未啟用 ADOPTCTX，否則 IBM MQ 會嘗試採用使用者 ID 進行未來授權檢查。

IBM MQ 對使用者 ID 的長度有限制，可供使用者進行授權檢查。 這些限制詳述於 第 72 頁的『使用者 ID』。 根據其他配置選項，當採用透過 MQCSP 結構 IBM MQ 傳遞的使用者 ID 時，會有不同的行為:

- 使用 LDAP 連線鑑別時， IBM MQ 會從該使用者的使用者 LDAP 記錄中擷取 SHORTUSR 中設定的欄位值，並採用該使用者 ID。

  例如，如果 SHORTUSR 設為 `'CN'`，且 LDAP 記錄將使用者列為 `'CN=Test,SN=MQ,O=IBM,C=UK'`，則會使用使用者 ID `Test`。
- 使用 OS 連線鑑別或 PAM 鑑別時，如果 ADOPTCTX 為 YES，則會截斷透過 MQCSP 結構傳遞的使用者 ID，以便在採用作為連線環境定義時符合 IBM MQ 的 12 個字元使用者 ID 限制。

  如果啟用 **ChlAuthEarlyAdopt**，則會在鑑別使用者認證之後進行截斷。

  如果未啟用 **ChlAuthEarlyAdopt**，則在採用之前會發生截斷。 在 Windows 上，如果以 `user@domain` 格式提供使用者，則這表示當使用者少於 12 個字元時，截斷可能會導致網域規格無效。

  例如，如果透過 MQCSP 提供使用者 `` `ibmmq@windowsdomain` ``，則在此實務範例中，會將其截斷為 `` `ibmmq@window` ``。 這會導致下列錯誤:

  ```
  AMQ8074W: 授權失敗，因為 SID 'SID' 不符合實體 'ibmmq@window'
  ```

  在此基礎上，如果您透過 MQCSP，傳遞長度超過 12 個字元的使用者 ID (例如 `user@domain` 格式的 Windows 網域使用者 ID)，您應該在 qm.ini 檔中配置 **ChlAuthEarlyAdopt**=*Y*，以避免發生此錯誤。

  或者，在 CONNAUTH AUTHINFO 配置上使用 ADOPTCTX (NO)，並使用替代方法 (例如 CHLAUTH USERMAP 規則、安全結束程式或通道物件 MCAUSER 設定) 來設定通道的使用者 ID。

在安全結束程式中實作識別及鑑別

您可以使用安全結束程式來實作單向或交互鑑別。

安全結束程式的主要目的是在通道的每一端啟用 MCA 來鑑別其夥伴。 在訊息通道的每一端，以及在 MQI 通道的伺服器端， MCA 通常會代表它所連接的佇列管理程式執行動作。 在 MQI 通道的用戶端端， MCA 通常會代表 IBM MQ MQI client 應用程式的使用者執行動作。 在此狀況下，實際上會在兩個佇列管理程式之間進行交互鑑別，或在佇列管理程式與 IBM MQ MQI client 應用程式的使用者之間進行交互鑑別。

提供的安全結束程式 (SSPI 通道結束程式) 說明如何透過交換由授信鑑別伺服器 (例如 Kerberos) 產生然後檢查的鑑別記號，來實作交互鑑別。 如需詳細資料，請參閱第 128 頁的『Windows 上的 SSPI 通道結束程式』。

也可以使用「公開金鑰基礎架構 (PKI)」技術來實作交互鑑別。 每一個安全結束程式都會產生一些隨機資料，使用它所代表的佇列管理程式或使用者的私密金鑰來簽署它，並在安全訊息中將簽署的資料傳送給它的夥伴。 夥伴安全結束程式會使用佇列管理程式或使用者的公開金鑰來檢查數位簽章，以執行鑑別。 在交換數位簽章之前，如果有多個演算法可供使用，安全結束程式可能需要同意產生訊息摘要的演算法。

當安全結束程式將已簽署的資料傳送至其夥伴時，它也需要傳送一些方法來識別它所代表的佇列管理程式或使用者。 這可能是「識別名稱」，甚至是數位憑證。 如果傳送數位憑證，夥伴安全結束程式可以透過主要 CA 憑證的憑證鏈來驗證憑證。 這可確保用來檢查數位簽章之公開金鑰的所有權。

夥伴安全結束程式只有在能夠存取包含憑證鏈中其餘憑證的金鑰儲存庫時，才能驗證數位憑證。 如果未傳送佇列管理程式或使用者的數位憑證，則必須在夥伴安全結束程式具有存取權的金鑰儲存庫中提供。 除非夥伴安全結束程式可以找到簽章者的公開金鑰，否則無法檢查數位簽章。

「傳輸層安全 (TLS)」使用 PKI 技術，如剛才說明的技術。 如需 Secure Sockets Layer 如何執行鑑別的相關資訊，請參閱 第 13 頁的『傳輸層安全 (TLS) 概念』。

如果無法使用授信鑑別伺服器或 PKI 支援，則可以使用其他技術。 一般技術 (可在安全結束程式中實作) 使用對稱金鑰演算法。

其中一個安全結束程式 (結束程式 A) 會產生亂數，並以安全訊息將它傳送至其夥伴安全結束程式 (結束程式 B)。 結束程式 B 會使用只有兩個安全結束程式已知的金鑰副本來加密數字。 結束程式 B 會使用結束程式 B 已產生的第二個亂數，將加密號碼傳送至安全訊息中的結束程式 A。 結束程式 A 會驗證第一個亂數是否已正確加密，使用其金鑰副本來加密第二個亂數，並將已加密的數字傳送至安全訊息中的結束程式 B。 然後，結束程式 B 會驗證第二個亂數是否已正確加密。 在此交換期間，如果任一安全結束程式不滿意其他安全結束程式的確實性，則可以指示 MCA 關閉通道。

此技術的優點是在交換期間不會透過通訊連線傳送金鑰或密碼。 缺點是它無法提供如何以安全方式配送共用金鑰的問題解決方案。 第 390 頁的『在使用者結束程式中實作機密性』中說明此問題的一個解決方案。 當兩個 LU 連結以形成階段作業時，在 SNA 中使用類似的技術來進行兩個 LU 的交互鑑別。 該技術在 第 98 頁的『階段作業層次鑑別』中有說明。

所有先前用於交互鑑別的技術都可以調整為提供單向鑑別。

訊息結束程式中的身分對映

您可以使用訊息結束程式來處理資訊，以鑑別使用者 ID ，但最好是在應用程式層次實作鑑別。

當應用程式將訊息放入佇列時，訊息描述子中的 *UserIdentifier* 欄位會包含與應用程式相關聯的使用者 ID。 不過，沒有可用來鑑別使用者 ID 的資料。 此資料可以由通道傳送端的訊息結束程式新增，並由通道接收端的訊息結束程式檢查。 例如，鑑別資料可以是加密密碼或數位簽章。

如果在應用程式層次實作此服務，則它可能更有效。 基本需求是接收訊息之應用程式的使用者能夠識別及鑑別傳送訊息之應用程式的使用者。 因此，我們自然會考慮在應用層面推行這項服務。 如需相關資訊，請參閱第 281 頁的『API 結束程式和 API 交互結束程式中的身分對映』。

API 結束程式和 API 交互結束程式中的身分對映

接收訊息的應用程式必須能夠識別及鑑別傳送訊息之應用程式的使用者。 此服務通常最好在應用程式層次實作。 API 結束程式可以多種方式來實作服務。

在個別訊息的層次上，識別及鑑別是涉及兩個使用者的服務，即訊息的傳送端及接收端。 基本需求是接收訊息之應用程式的使用者能夠識別及鑑別傳送訊息之應用程式的使用者。 請注意，需求是單向 (而非雙向) 鑑別。

視實作方式而定，使用者及其應用程式可能需要與服務互動，甚至需要與服務互動。 此外，何時及如何使用服務可能取決於使用者及其應用程式所在的位置，以及應用程式本身的本質。 因此，考慮在應用程式層次而非鏈結層次實作服務是很自然的。

如果您考慮在鏈結層次實作此服務，則可能需要解決如下所示的問題:

- 在訊息通道上，如何只將服務套用至那些需要它的訊息?

- 如果這是需求，您如何讓使用者及其應用程式與服務互動?
- 在多躍點狀況下，在傳送訊息至目的地的途中，會透過多個訊息通道傳送訊息，您在何處呼叫服務的元件?

以下是一些範例，說明如何在應用程式層次實作識別及鑑別服務。 術語 *API* 結束程式 表示 API 結束程式或 API 交互結束程式。

- 當應用程式將訊息放入佇列時， API 結束程式可以從授信鑑別伺服器 (例如 Kerberos) 獲得鑑別記號。 API 結束程式可以將此記號新增至訊息中的應用程式資料。 當接收端應用程式擷取訊息時，第二個 API 結束程式可以檢查記號，要求鑑別伺服器鑑別傳送端。
- 當應用程式將訊息放入佇列時， API 結束程式可以將下列項目附加至訊息中的應用程式資料:
  - 傳送端的數位憑證
  - 傳送者的數位簽章

  如果用於產生訊息摘要的不同演算法可供使用，則 API 結束程式可以包括它所使用的演算法名稱。

  當接收端應用程式擷取訊息時，第二個 API 結束程式可以執行下列檢查:

  - API 結束程式可以透過主要 CA 憑證的憑證鏈來驗證數位憑證。 若要這樣做， API 結束程式必須有權存取包含憑證鏈中其餘憑證的金鑰儲存庫。 此檢查可確保由「識別名稱」識別的傳送者是憑證中所包含公開金鑰的真正擁有者。
  - API 結束程式可以使用憑證中包含的公開金鑰來檢查數位簽章。 此檢查會鑑別寄件者。

  可以傳送寄件者的「識別名稱」，而不是整個數位憑證。 在此情況下，金鑰儲存庫必須包含傳送端的憑證，以便第二個 API 結束程式可以找到傳送端的公開金鑰。 另一種可能是傳送憑證鏈中的所有憑證。
- 當應用程式將訊息放入佇列時，訊息描述子中的 *UserIdentifier* 欄位會包含與應用程式相關聯的使用者 ID。 使用者 ID 可用來識別寄件者。 若要啟用鑑別， API 結束程式可以將部分資料 (例如加密密碼) 附加至訊息中的應用程式資料。 當接收端應用程式擷取訊息時，第二個 API 結束程式可以使用隨訊息一起傳送的資料來鑑別使用者 ID。

  對於源自受控制及授信環境的訊息，以及在無法使用授信鑑別伺服器或 PKI 支援的情況下，此技術可能被視為已足夠。

使用已撤銷的憑證

「憑證管理中心」可以撤銷數位憑證。 視平台而定，您可以使用 OCSP 或 LDAP 伺服器上的 CRL 來檢查憑證的撤銷狀態。

在 TLS 信號交換期間，通訊夥伴會使用數位憑證彼此鑑別。 鑑別時可能會檢查收到的憑證是否仍可信任。 憑證管理中心 (CA) 基於各種原因撤銷憑證，包括:

- 擁有者已移至不同的組織
- 私密金鑰不再是秘密金鑰

CA 在「憑證撤銷清冊 (CRL)」中發佈撤銷的個人憑證。 已撤銷的 CA 憑證會發佈在「權限撤銷清單 (ARL)」中。

ALW 在 AIX, Linux, and Windows 平台上， IBM MQ SSL 支援會使用 OCSP (線上憑證狀態通訊協定) 或使用 LDAP (輕量型目錄存取通訊協定) 伺服器上的 CRL 及 ARL ，來檢查已撤銷的憑證。 OCSP 是較好的方法。

「IBM MQ classes for Java」及 IBM MQ classes for JMS 無法在用戶端通道定義表檔案中使用 OCSP 資訊。 不過，您可以如使用線上憑證通訊協定中所述，配置 OCSP。

z/OS IBM i 在 IBM i 及 z/OS 平台上， IBM MQ SSL 僅支援使用 LDAP 伺服器上的 CRL 及 ARL 來檢查已撤銷的憑證。

如需「憑證管理中心」的相關資訊，請參閱 第 9 頁的『數位憑證』。

OCSP/CRL 檢查

對遠端送入憑證執行線上憑證狀態通訊協定 (OCSP)/憑證撤銷清冊 (CRL) 檢查。 此程序會檢查從遠端系統的個人憑證到其主要憑證所涉及的整個鏈結。

使用 openSSL 來驗證 OCSP 驗證

如果您的企業使用 openSSL 來驗證 OCSP，然後您嘗試使用 IBM Global Security Kit (GSKit) TLS 連線，則會收到「不明」狀態警告。

這是因為 GSKit 會檢查鏈結中除主要憑證以外的所有憑證，以取得撤銷狀態。 GSKit 作業符合 RFC 5280，這在 GSKit 信任原則中說明。 GSKit 演算法會嘗試撤銷資訊的所有可用來源，如 RFC 5280 及 GSKit 信任原則中所述。

OCSP/CRL 檢查在 IBM MQ 中如何運作?

IBM MQ 支援兩種機制來控制在憑證延伸或 AUTHINFO 物件中定義的針對具名 OCSP 或 CRL 端點檢查憑證時的行為:

- qm.ini 檔案的 SSL 段落的 **OCSPCheckExtensions**、 **CDPCheckExtensions** 及 **OCSPAuthentication** 屬性，以及
- 使用佇列管理程式的 SSLCRLNL 參數以及 AUTHINFO OCSP 和 CRLLDAP 配置。 如需相關資訊，請參閱 ALTER AUTHINFO 及 ALTER QMGR 。

小心:

搭配 **AUTHTYPE(OCSP)** 的 ALTER AUTHINFO 指令不適用於在 IBM i 或 z/OS 佇列管理程式上使用。 不過，您可以在那些平台上指定它，以複製到用戶端通道定義表 (CCDT) 供用戶端使用。

OCSPCheckExtensions 及 **CDPCheckExtensions** SSL 段落屬性控制 IBM MQ 是否根據憑證的 AIA 延伸規格內詳述的 OCSP 或 CRL 伺服器來驗證憑證。

如果未啟用，則不會聯絡憑證延伸中的 OCSP 或 CRL 伺服器。

如果 OCSP 或 CRL 伺服器透過 AUTHINFO 物件詳細說明，並使用 SSLCRLNL **QMGR** 屬性進行參照，則在憑證撤銷處理期間， IBM MQ 會嘗試聯絡這些伺服器。

重要: 在 SSLCRLNL 名單中只能定義一個 OCSP AUTHINFO 物件。

如果:

OCSPCheckExtensions= NO 和 **CDPCheckExtensions**=*NO* 已設定，且
在 AUTHINFO 物件中未定義任何 OCSP 或 CRL 伺服器

不執行憑證撤銷檢查。

當驗證憑證的撤銷狀態時， IBM MQ 會依下列順序聯絡 OCSP 或 CRL 伺服器 (如果已啟用的話):

1. OCSP 伺服器詳述於 **AUTHTYPE(OCSP)** 物件中，並在 SSLCRLNL **QMGR** 屬性中參照。
2. 如果 **OCSPCheckExtensions**=*YES*，則 OCSP 伺服器詳述於憑證的 AIA 延伸中。
3. 如果 **CDPCheckExtensions** =*YES*，則 CRL 伺服器詳述於憑證的 **CRLDistributionPoints** 延伸規格中。
4. 在 **AUTHINFO(CRLLDAP)** 物件中詳述並在 SSLCRLNL **QMGR** 屬性中參照的任何 CRL 伺服器。

在驗證憑證時，如果某個步驟導致 OCSP 伺服器或 CRL 伺服器傳回對憑證查詢的明確 REVOKED 或 VALID 回應，則不會執行進一步檢查，且會使用所呈現憑證的狀態來決定是否信任它。

如果 OCSP 伺服器或 CRL 伺服器傳回 UNKNOWN 的結果，則會繼續處理，直到 OCSP 或 CRL 伺服器傳回最終結果或所有選項都用盡為止。

對於 OCSP 和 CRL 伺服器，憑證是否被視為已撤銷 (如果無法判斷其狀態) 的行為不同:

- 對於 CRL 伺服器，如果無法取得 CRL，則會將憑證視為 NOT_REVOKED
- 對於 OCSP 伺服器，如果無法從指名的 OCSP 伺服器取得撤銷狀態，則會透過 qm.ini 檔案的「SSL 段落」中的 **OCSPAuthentication** 屬性來控制行為。

  您可以將此屬性配置為封鎖連線、容許連線或容許具有警告訊息的連線。

必要的話，您可以在 qm.ini 及 mqclient.ini 檔的 SSL 段落中使用 **SSLHTTPProxyName**=*string* 屬性，以進行 OCSP 檢查。 該字串是 GSKit 要用於 OCSP 檢查之 HTTP Proxy 伺服器的主機名稱或網址。

從 IBM MQ 9.1.5 開始，您可以在 qm.ini 或 mqclient.ini 檔案的 SSL 段落中設定 **OCSPTimeout** 值，以設定在執行撤銷檢查時等待 OCSP 回應端的秒數。

ALW 撤銷的憑證及 OCSP

IBM MQ 決定要使用的「線上憑證狀態通訊協定 (OCSP)」回應者，並處理收到的回應。 您可能需要執行一些步驟，才能讓 OCSP 回應端成為有存取權的。

註: 此資訊僅適用於 AIX, Linux, and Windows 系統上的 IBM MQ 。

若要使用 OCSP 檢查數位憑證的撤銷狀態， IBM MQ 可以使用兩種方法來判定要聯絡哪個 OCSP 回應端:

- 使用要檢查之憑證中的 AuthorityInfoAccess (AIA) 憑證延伸。
- 使用鑑別資訊物件中指定的 URL，或用戶端應用程式指定的 URL。

鑑別資訊物件或用戶端應用程式指定的 URL，其優先權高於 AIA 憑證延伸中的 URL。

如果 OCSP 回應端的 URL 位於防火牆後面，請重新配置防火牆，以便可以存取 OCSP 回應端，或設定 OCSP Proxy 伺服器。 在 SSL 段落中使用 SSLHTTPProxyName 變數，指定 Proxy 伺服器的名稱。 在用戶端系統上，您也可以使用環境變數 MQSSLPROXY 來指定 Proxy 伺服器的名稱。 如需詳細資料，請參閱相關資訊。

如果您不在意 TLS 憑證是否已撤銷，可能是因為您是在測試環境中執行，則您可以在 SSL 段落中，將 OCSPCheckExtensions 設為 NO。 如果設定此變數，則會忽略任何 AIA 憑證延伸。 但是在正式作業環境中，無法接受此解決方案，在此種作業環境中，您可能並不希望讓提出撤銷憑證的使用者進行存取。

呼叫存取 OCSP 回應端，會導致下列三種結果之一:

良好
憑證有效。

已撤銷
憑證已撤銷。

不明
產生此結果的原因，可能是下列三種之一:

- IBM MQ 無法存取 OCSP 回應者。
- OCSP 回應者已傳送回應，但 IBM MQ 無法驗證回應的數位簽章。
- OCSP 回應端已傳送回應，指出沒有憑證的撤銷資料。

如果 IBM MQ 收到**不明**的 OCSP 結果，則其行為視 OCSPAuthentication 屬性的設定而定。 對於佇列管理程式，此屬性保留在下列其中一個位置:

- Linux AIX 在 AIX and Linux 上 qm.ini 檔案的 SSL 段落中。
- Windows 在 Windows 登錄中。

可以使用 IBM MQ Explorer 來設定此屬性。 對於用戶端，該屬性保留在用戶端配置檔的 SSL 段落中。

如果收到**不明**的結果，且 OCSPAuthentication 設為 REQUIRED（預設值），則 IBM MQ 會拒絕連線並發出類型為 AMQ9716 的錯誤訊息。 如果已啟用佇列管理程式 SSL 事件訊息，則會產生類型 MQRC_CHANNEL_SSL_ERROR 且 ReasonQualifier 設為 MQRQ_SSL_HANDSHAKE_ERROR 的 SSL 事件訊息。

如果收到**不明**的結果，且 OCSPAuthentication 設為 OPTIONAL，則 IBM MQ 容許 SSL 通道啟動，且不會產生警告或 SSL 事件訊息。

如果收到**不明**的結果，且 OCSPAuthentication 設為 WARN，則會啟動 SSL 通道，但 IBM MQ 會在錯誤日誌中發出類型為 AMQ9717 的警告訊息。 如果已啟用佇列管理程式 SSL 事件訊息，則會產生類型 MQRC_CHANNEL_SSL_WARNING 且 ReasonQualifier 設為 MQRQ_SSL_UNKNOWN_REVOCATION 的 SSL 事件訊息。

OCSP 回應的數位簽章

OCSP 回應端可以利用下列三種方法之一來簽署其回應。 您的回應端會通知您要使用哪一種方法。

- OCSP 回應可以使用 CA 憑證以數位方式進行簽署，該憑證即發出所要檢查之憑證的相同 CA 憑證。 在此情況下，您不需要設定任何其他憑證; 您已採取來建立 TLS 連線功能的步驟足以驗證 OCSP 回應。
- OCSP 回應可以使用另一個憑證以數位方式進行簽署，該憑證由發出所要檢查之憑證的相同憑證管理中心 (CA) 進行簽署。 在此情況下，簽署憑證會隨 OCSP 回應一起傳送。 從 OCSP 回應端傳出的憑證，必須將「延伸金鑰使用延伸」設為 `id-kp-OCSPSigning`，才會信任它有此用途。 因為 OCSP 回應會隨簽署它的憑證一起傳送 (該憑證是由已授信 TLS 連線功能的 CA 所簽署) ，所以不需要其他憑證設定。
- OCSP 回應可以使用另一個憑證以數位方式進行簽署，該憑證與所要檢查之憑證沒有直接關聯。 在此情況下，OCSP 回應會以 OCSP 回應端本身所發出的憑證進行簽署。 您必須將 OCSP 回應端憑證的副本新增至執行 OCSP 檢查之用戶端或佇列管理程式的金鑰資料庫。 請參閱 第 255 頁的『將 CA 憑證或自簽憑證的公用部分新增至 AIX, Linux, and Windows 上的金鑰儲存庫』。 新增 CA 憑證時，預設會將它新增為授信主要憑證，此為這個環境定義的必要設定。 如果未新增此憑證，則 IBM MQ 無法驗證 OCSP 回應上的數位簽章，且 OCSP 檢查會導致「不明」結果，這可能會導致 IBM MQ 關閉通道，視 OCSPAuthentication 的值而定。

Java 及 JMS 用戶端應用程式中的線上憑證狀態通訊協定 (OCSP)

由於 Java API 的限制，只有在對整個 Java 虛擬機器 (JVM) 處理程序啟用 OCSP 時， IBM MQ 才能對 TLS 安全 Socket 使用「線上憑證狀態通訊協定 (OCSP)」憑證撤銷檢查。 有兩種方式可以為 JVM 中的所有安全 Socket 啟用 OCSP:

- 編輯 JRE java.security 檔案，以包含顯示在表格 1 中的 OCSP 配置設定，並重新啟動應用程式。
- 使用 java.security.Security.setProperty() API ，受任何有效的 Java Security Manager 原則所規範。

您至少必須指定 ocsp.enable 和 ocsp.responderURL 值的其中一個。

內容名稱	說明
ocsp.enable	此內容的值為 `true` 或 `false`。 若為 `true`，在進行憑證撤銷檢查時會啟用 OCSP 檢查；若為 `false` 或是未設定，則會停用 OCSP 檢查。
ocsp.responderURL	此內容的值是識別 OCSP 回應端位置的 URL。 例如: `ocsp.responderURL=http://ocsp.example.net:80`。 依預設，OCSP 回應端位置是由要驗證的憑證隱含地判定。 當憑證中沒有「權限資訊存取」延伸（定義於 RFC 3280）時，或是需要置換之時，會使用此內容。
ocsp.responderCertSubjectName	此內容的值是 OCSP 回應端憑證的主體名稱。 例如: `ocsp.responderCertSubjectName="CN=OCSP Responder, O=XYZ Corp"`。 依預設，OCSP 回應端的憑證是要驗證之憑證發證者的憑證。 此內容可在預設值不適用時識別 OCSP 回應端的憑證。 它的值是一個字串識別名稱（定義於 RFC 2253），可識別在憑證路徑驗證期間所提供之憑證集裡的憑證。 當單獨使用主體名稱不足以唯一識別憑證時，必須改為同時使用 ocsp.responderCertIssuerName 及 ocsp.responderCertSerialNumber 內容。 設定此內容時，會忽略 ocsp.responderCertIssuerName 及 ocsp.responderCertSerialNumber 內容。
ocsp.responderCertIssuerName	此內容的值是 OCSP 回應端憑證的發證者名稱。 例如: `ocsp.responderCertIssuerName="CN=Enterprise CA, O=XYZ Corp"`。 依預設，OCSP 回應端的憑證是要驗證之憑證發證者的憑證。 此內容可在預設值不適用時識別 OCSP 回應端的憑證。 它的值是一個字串識別名稱（定義於 RFC 2253），可識別在憑證路徑驗證期間所提供之憑證集裡的憑證。 設定此內容時，必須同時也設定 ocsp.responderCertSerialNumber 內容。 設定 ocsp.responderCertSubjectName 內容時，會忽略此內容。
ocsp.responderCertSerialNumber	此內容的值是 OCSP 回應端憑證的序號。 例如: `ocsp.responderCertSerialNumber=2A:FF:00`。 依預設，OCSP 回應端的憑證是要驗證之憑證發證者的憑證。 此內容可在預設值不適用時識別 OCSP 回應端的憑證。 這個值是一個十六進位數字的

內容名稱	說明
	字串（必須有冒號或空格分隔字元），可識別在憑證路徑驗證期間所提供之憑證集裡的憑證。 設定此內容時，必須同時也設定 ocsp.responderCertIssuerName 內容。 設定 ocsp.responderCertSubjectName 內容時，會忽略此內容。

以此方式啟用 OCSP 之前，有許多考量事項:

- 設定 OCSP 配置會影響 JVM 處理程序中的所有安全 Socket。 在某些情況下，當 JVM 與使用 TLS 安全 Socket 的其他應用程式碼共用時，此配置可能會產生不良的副作用。 請確定所選擇的 OCSP 配置適合在相同 JVM 中執行的所有應用程式。
- 套用維護到您的 JRE 可能會改寫 java.security 檔案。 當您套用 Java 臨時修正程式和產品維護時，請小心避免改寫 java.security 檔案。 套用維護之後可能需要重新套用您的 java.security 變更。 因此，您可能會考慮改用 java.security.Security.setProperty() API 來設定 OCSP 配置。
- 啟用 OCSP 檢查唯有在同時啟用撤銷檢查時才有效果。 撤銷檢查是以 `PKIXParameters.setRevocationEnabled()` 方法啟用。
- 如果您使用 在原生攔截程式中啟用 OCSP 檢查中說明的 AMS Java 攔截程式，請小心避免使用與金鑰儲存庫配置檔中 AMS OCSP 配置衝突的 java.security OCSP 配置。

使用憑證撤銷清冊及權限撤銷清冊

IBM MQ 對 CRL 及 ARL 的支援會因平台而異。

每個平台上的 CRL 及 ARL 支援如下:

- 在 z/OS 上，系統 SSL 支援 Tivoli 公開金鑰基礎架構產品儲存在 LDAP 伺服器中的 CRL 及 ARL。
- 在其他平台上， CRL 及 ARL 支援符合 PKIX X.509 V2 CRL 設定檔建議。

IBM MQ 會維護在過去 12 小時內已存取的 CRL 及 ARL 的快取。

當佇列管理程式或 IBM MQ MQI client 收到憑證時，它會檢查 CRL 以確認該憑證仍然有效。 IBM MQ 會先移入快取 (如果有快取的話)。 如果 CRL 不在快取中， IBM MQ 會依照 LDAP CRL 伺服器位置在 *SSLCRLNL* 屬性指定的鑑別資訊物件名稱清單中的出現順序來詢問它們，直到 IBM MQ 找到可用的 CRL 為止。 如果未指定名稱清單，或以空白值指定，則不會檢查 CRL。

設定 *LDAP* 伺服器

配置「LDAP 目錄資訊樹狀結構」結構，以反映 CA 識別名稱的階層。 使用「LDAP 資料交換格式」檔案來執行此動作。

配置 LDAP 目錄資訊樹狀結構 (DIT) 結構，以使用對應於發出憑證及 CRL 之 CA 識別名稱的階層。 您可以使用使用「LDAP 資料交換格式 (LDIF)」的檔案來設定 DIT 結構。 您也可以使用 LDIF 檔案來更新目錄。

LDIF 檔案是 ASCII 文字檔，包含在 LDAP 目錄中定義物件所需的資訊。 LDIF 檔案包含一或多個項目，每一個項目都包含「識別名稱」、至少一個物件類別定義，以及選擇性地包含多個屬性定義。

`certificateRevocationList;binary` 屬性包含已撤銷使用者憑證的二進位格式清單。 `authorityRevocationList;binary` 屬性包含已撤銷的 CA 憑證二進位清單。 若要與 IBM MQ TLS 搭配使用，這些屬性的二進位資料必須符合 DER (明確編碼規則) 格式。 如需 LDIF 檔案的相關資訊，請參閱 LDAP 伺服器隨附的文件。

第 287 頁的圖 20 顯示範例 LDIF 檔案，您可以建立作為 LDAP 伺服器的輸入，以載入 CA1 所發出的 CRL 及 ARL，這是一個虛的「憑證管理中心」，具有識別名稱 "CN=CA1， OU=Test， O=IBM， C=GB"，由「測試」組織在 IBM 內設定。

```
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)
```

圖 *20:* 憑證管理中心的 *LDIF* 檔案範例。 這可能因實作不同而異。

第 287 頁的圖 21 顯示當您載入 第 287 頁的圖 20 中所顯示的範例 LDIF 檔案，以及 CA2 的類似檔案時，LDAP 伺服器所建立的 DIT 結構。CA2 是 PKI 組織所設定的虛構「憑證管理中心」，也是在 IBM 內。

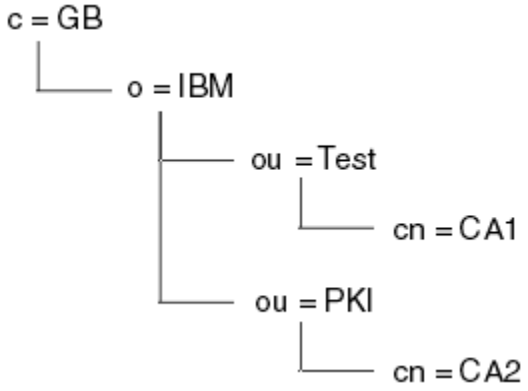


圖 *21: LDAP* 目錄資訊樹狀結構的範例

IBM MQ 會檢查 CRL 及 ARL。

註: 請確定 LDAP 伺服器的存取控制清單容許授權使用者讀取、搜尋及比較保留 CRL 及 ARL 的項目。 IBM MQ 會使用 AUTHINFO 物件的 LDAPUSER 及 LDAPPWD 內容來存取 LDAP 伺服器。

配置及更新 *LDAP* 伺服器
使用此程序來配置或更新 LDAP 伺服器。

1. 從「憑證管理中心」或「權限」取得 DER 格式的 CRL 及 ARL。
2. 使用文字編輯器或 LDAP 伺服器隨附的工具，建立一個以上 LDIF 檔案，其中包含 CA 的「識別名稱」及必要的物件類別定義。 將 DER 格式資料複製到 LDIF 檔案，作為 CRL 的 `certificateRevocationList;binary` 屬性及/或 ARL 的 `authorityRevocationList;binary` 屬性值。
3. 啟動 LDAP 伺服器。
4. 從您在步驟 第 287 頁的『2』建立的一或多個 LDIF 檔案中新增項目。

在配置 LDAP CRL 伺服器之後，請檢查它是否已正確設定。 首先，請嘗試使用通道上未撤銷的憑證，並檢查通道是否正確啟動。 然後使用已撤銷的憑證，並檢查通道是否無法啟動。

經常從「憑證管理中心」取得更新的 CRL。 請考慮每 12 小時在 LDAP 伺服器上執行一次。

使用佇列管理程式存取 *CRL* 及 *ARL*

佇列管理程式與一或多個鑑別資訊物件相關聯，這些鑑別資訊物件保留 LDAP CRL 伺服器的位址。

IBM i IBM i 上的 IBM MQ 與其他平台的行為不同。

請注意，在此區段中，「憑證撤銷清冊 (CRL)」的相關資訊也適用於「權限撤銷清冊 (ARL)」。

您可以向佇列管理程式提供鑑別資訊物件，每一個物件都保留 LDAP CRL 伺服器的位址，以告知佇列管理程式如何存取 CRL。 鑑別資訊物件保留在 *SSLCRLNL* 佇列管理程式屬性中指定的名單中。

在下列範例中，使用 MQSC 來指定參數:

1. 使用 DEFINE AUTHINFO MQSC 指令定義鑑別資訊物件，並將 AUTHTYPE 參數設為 CRLLDAP。

   IBM i 在 IBM i 上，您也可以使用 CRTMQMAUTI CL 指令。

   AUTHTYPE 參數的值 CRLLDAP 指出在 LDAP 伺服器上存取 CRL。 您建立的每一個類型為 CRLLDAP 的鑑別資訊物件都會保留 LDAP 伺服器的位址。 當您有多個鑑別資訊物件時，它們所指向的 LDAP 伺服器必須包含相同的資訊。 這可在一或多個 LDAP 伺服器失敗時提供服務的連續性。

   z/OS 此外，僅在 z/OS 上，必須使用相同的使用者 ID 和密碼來存取所有 LDAP 伺服器。 使用的使用者 ID 及密碼是在名單的第一個 AUTHINFO 物件中指定的使用者 ID 及密碼。

   在所有平台上，使用者 ID 和密碼會以未加密的方式傳送至 LDAP 伺服器。

2. 使用 DEFINE NAMELIST MQSC 指令，定義鑑別資訊物件名稱的名單。 z/OS 在 z/OS 上，確保 NLTYPE 名單屬性設為 AUTHINFO。
3. 使用 ALTER QMGR MQSC 指令，將名稱清單提供給佇列管理程式。 例如:

   ```
   ALTER QMGR SSLCRLNL(sslcrlnlname)
   ```

   其中 `sslcrlnlname` 是鑑別資訊物件的名單。

   此指令會設定稱為 *SSLCRLNL* 的佇列管理程式屬性。 此屬性的佇列管理程式起始值為空白。

IBM i 在 IBM i 上，您可以指定鑑別資訊物件，但佇列管理程式既不使用鑑別資訊物件，也不使用鑑別資訊物件的名稱清單。 只有使用 IBM i 佇列管理程式所產生之用戶端連線表格的 IBM MQ 用戶端，才會使用指定給該 IBM i 佇列管理程式的鑑別資訊。 IBM i 上的 *SSLCRLNL* 佇列管理程式屬性決定用戶端使用的鑑別資訊。 如需告知 IBM i 佇列管理程式如何存取 CRL 的相關資訊，請參閱 第 288 頁的『在 IBM i 上存取 CRL 及 ARL』 。

您可以在名單中新增最多 10 個替代 LDAP 伺服器的連線，以確保在一或多個 LDAP 伺服器失敗時服務的連續性。 請注意， LDAP 伺服器必須包含相同的資訊。

IBM i 在 *IBM i* 上存取 *CRL* 及 *ARL*

使用此程序來存取 IBM i 上的 CRL 或 ARL。

請注意，在此區段中，「憑證撤銷清冊 (CRL)」的相關資訊也適用於「權限撤銷清冊 (ARL)」。

請遵循下列步驟，在 IBM i 上設定特定憑證的 CRL 位置:

1. 存取 DCM 介面，如 第 230 頁的『存取 DCM』 中所述。
2. 在導覽畫面的 **管理 CRL 位置** 作業種類中，按一下 **新增 CRL 位置**。 「管理 CRL 位置」頁面會顯示在作業頁框中。
3. 在 **CRL 位置名稱** 欄位中，輸入 CRL 位置名稱，例如 `LDAP Server #1`
4. 在 **LDAP 伺服器** 欄位中，輸入 LDAP 伺服器名稱。
5. 在 **使用 Secure Sockets Layer (SSL)** 欄位中，如果您想要使用 TLS 連接至 LDAP 伺服器，請選取 **是** 。 否則，請選取 **否**。
6. 在 **埠號** 欄位中，輸入 LDAP 伺服器的埠號，例如 389。
7. 如果 LDAP 伺服器不容許匿名使用者查詢目錄，請在 **登入識別名稱** 欄位中鍵入伺服器的登入識別名稱。
8. 按一下**確定**。 DCM 會通知您它已建立 CRL 位置。
9. 在導覽畫面中，按一下 **選取憑證庫**。 作業頁框中會顯示「選取憑證庫」頁面。
10. 選取 **其他系統憑證庫** 勾選框，然後按一下 **繼續**。 即會顯示「憑證儲存庫及密碼」頁面。
11. 在 **憑證儲存庫路徑和檔名** 欄位中，鍵入您在 第 231 頁的『在 IBM i 上建立憑證儲存庫』 時設定的 IFS 路徑和檔名。

12. 在 **憑證庫密碼** 欄位中輸入密碼。 按一下**繼續**。 「現行憑證庫」頁面會顯示在作業頁框中。
13. 在導覽畫面的 **管理憑證** 作業種類中，按一下 **更新 CRL 位置指派**。 「CRL 位置指派」頁面會顯示在作業頁框中。
14. 選取您要指派 CRL 位置之 CA 憑證的圓鈕。 按一下 **更新 CRL 位置指派**。 「更新 CRL 位置指派」頁面會顯示在作業頁框中。
15. 選取您要指派給憑證之 CRL 位置的圓鈕。 按一下 **更新指派**。 DCM 會通知您它已更新指派。

請注意， DCM 可讓您依「憑證管理中心」指派不同的 LDAP 伺服器。

使用 *IBM MQ Explorer* 存取 *CRL* 及 *ARL*
您可以使用 IBM MQ Explorer 來告知佇列管理程式如何存取 CRL。

請注意，在此區段中，「憑證撤銷清冊 (CRL)」的相關資訊也適用於「權限撤銷清冊 (ARL)」。

請使用下列程序來設定與 CRL 的 LDAP 連線:

1. 請確定您已啟動佇列管理程式。
2. 用滑鼠右鍵按一下 **鑑別資訊** 資料夾，然後按一下 **新建-> 鑑別資訊**。 在開啟的內容表中:
   a. 在第一頁 **建立鑑別資訊**，輸入 CRL (LDAP) 物件的名稱。
   b. 在 **變更內容**的「 **一般** 」頁面上，選取連線類型。 您可以選擇性地輸入說明。
   c. 選取 **變更內容**的 **CRL (LDAP)** 頁面。
   d. 輸入 LDAP 伺服器名稱作為網路名稱或 IP 位址。
   e. 如果伺服器需要登入詳細資料，請提供使用者 ID 及密碼 (必要的話)。
   f. 按一下**確定**。
3. 用滑鼠右鍵按一下「名稱清單」 資料夾，然後按一下 **新建-> 名稱清單**。 在開啟的內容表中:
   a. 輸入名稱清單的名稱。
   b. 新增 CRL (LDAP) 物件的名稱 (從步驟 第 289 頁的『2.a』) 清單。
   c. 按一下**確定**。
4. 用滑鼠右鍵按一下佇列管理程式，選取 **內容**，然後選取 **SSL** 頁面:
   a. 選取 **根據憑證撤銷清冊檢查此佇列管理程式收到的憑證** 勾選框。
   b. 鍵入名稱清單的名稱 (來自步驟 第 289 頁的『3.a』) 在 **CRL 名單** 欄位中。

使用 *IBM MQ MQI client* 存取 *CRL* 及 *ARL*

您有三個選項可指定 LDAP 伺服器，這些伺服器保留供 IBM MQ MQI client 檢查的 CRL。

請注意，在此區段中，「憑證撤銷清冊 (CRL)」的相關資訊也適用於「權限撤銷清冊 (ARL)」。

指定 LDAP 伺服器的三種方式如下:

- 使用通道定義表
- 在 MQCONNX 呼叫中使用 SSL 配置選項結構 MQSCO
- 使用 Active Directory (在具有 Active Directory 支援的 Windows 系統上)

如需詳細資料，請參閱相關資訊。

您可以包括最多 10 個與替代 LDAP 伺服器的連線，以確保在一個以上 LDAP 伺服器失敗時服務的連續性。 請注意， LDAP 伺服器必須包含相同的資訊。

您無法從 Linux (zSeries 平台) 上執行的 IBM MQ MQI client 通道存取 LDAP CRL。

OCSP 回應者及保留 *CRL* 之 *LDAP* 伺服器的位置
在 IBM MQ MQI client 系統上，您可以指定 OCSP 回應端的位置，以及保留憑證撤銷清冊 (CRL) 之「輕量型目錄存取通訊協定 (LDAP)」伺服器的位置。

您可以用三種方式來指定這些位置，這裡以遞減優先順序來說明。

IBM i 若為 IBM i，請參閱 存取 IBM i 上的 CRL 及 ARL。

當 IBM MQ MQI client 應用程式發出 MQCONNX 呼叫時

您可以在 **MQCONNX** 呼叫上指定 OCSP 回應者或保留 CRL 的 LDAP 伺服器。

在 **MQCONNX** 呼叫上，連接選項結構 MQCNO 可以參照 SSL 配置選項結構 MQSCO。 接著， MQSCO 結構可以參照一或多個鑑別資訊記錄結構 MQAIR。 每一個 MQAIR 結構包含 IBM MQ MQI client 存取 OCSP 回應端或 LDAP 伺服器保留 CRL 所需的所有資訊。 例如， MQAIR 結構中的其中一個欄位是可以聯絡回應者的 URL。 如需 MQAIR 結構的相關資訊，請參閱 MQAIR-鑑別資訊記錄。

使用用戶端通道定義表 (ccdt) 來存取 OCSP 回應端或 LDAP 伺服器

因此， IBM MQ MQI client 可以存取保留 CRL 的 OCSP 回應端或 LDAP 伺服器，包括用戶端通道定義表中一個以上鑑別資訊物件的屬性。

在伺服器佇列管理程式上，您可以定義一或多個鑑別資訊物件。 鑑別物件的屬性包含存取 OCSP 回應者 (在支援 OCSP 的平台上) 或保留 CRL 的 LDAP 伺服器所需的所有資訊。 其中一個屬性指定 OCSP 回應端 URL，另一個屬性指定 LDAP 伺服器執行所在系統的主機位址或 IP 位址。

z/OS IBM i 具有 AUTHTYPE (OCSP) 的鑑別資訊物件不適用於 IBM i 或 z/OS 佇列管理程式，但可以在那些平台上指定它，以複製到用戶端通道定義表 (CCDT) 以供用戶端使用。

若要讓 IBM MQ MQI client 存取保留 CRL 的 OCSP 回應端或 LDAP 伺服器，可以在用戶端通道定義表中包含一或多個鑑別資訊物件的屬性。 您可以使用下列其中一種方式來併入此類屬性:

Multi

在伺服器平台上: AIX、 Linux、 IBM i 及 Windows

您可以定義名稱清單，其中包含一個以上鑑別資訊物件的名稱。 然後，您可以將佇列管理程式屬性 **SSLCRLNL** 設為此名單的名稱。

如果您使用 CRL，則可以配置多個 LDAP 伺服器以提供更高可用性。 目的是讓每一個 LDAP 伺服器保留相同的 CRL。 如果有一部 LDAP 伺服器在需要時無法使用，則 IBM MQ MQI client 可以嘗試存取另一部 LDAP 伺服器。

在這裡，名稱清單所識別的鑑別資訊物件屬性統稱為 憑證撤銷位置。 當您將佇列管理程式屬性 **SSLCRLNL** 設為名單名稱時，憑證撤銷位置會複製到與佇列管理程式相關聯的用戶端通道定義表中。 如果 CCDT 可以從用戶端系統作為共用檔案進行存取，或隨後將 CCDT 複製到用戶端系統，則該系統上的 IBM MQ MQI client 可以使用 CCDT 中的憑證撤銷位置，來存取包含 CRL 的 OCSP 回應端或 LDAP 伺服器。

如果稍後變更佇列管理程式的憑證撤銷位置，則變更會反映在與佇列管理程式相關聯的 CCDT 中。 如果佇列管理程式屬性 **SSLCRLNL** 設為空白，則會從 CCDT 中移除憑證撤銷位置。 這些變更不會反映在用戶端系統上表格的任何副本中。

如果您需要 MQI 通道的用戶端和伺服器端的憑證撤銷位置不同，且伺服器佇列管理程式是用來建立憑證撤銷位置的伺服器佇列管理程式，您可以執行下列動作:

1. 在伺服器佇列管理程式上，建立憑證撤銷位置以在用戶端系統上使用。
2. 將包含憑證撤銷位置的 CCDT 複製到用戶端系統。
3. 在伺服器佇列管理程式上，將憑證撤銷位置變更為 MQI 通道伺服器端所需要的位置。
4. 在用戶端機器上，您可以搭配使用 **runmqsc** 指令與 **-n** 參數。

Multi

在用戶端平台上: AIX、 Linux、 IBM i 及 Windows

您可以使用 runmqsc 指令搭配 CCDT 檔案中的 **-n** 參數及 **DEFINE AUTHINFO** 物件，在用戶端機器上建置 CCDT。 定義物件的順序是它們在檔案中的使用順序。 您在 **DEFINE AUTHINFO** 物件中可能使用的任何名稱都不會保留在檔案中。 當您在 CCDT 檔案中 **DISPLAY AUTHINFO** 物件時，只會使用位置號碼。

註: 如果您指定 **-n** 參數，則不得指定任何其他參數。

在 Windows 上使用 Active Directory

Windows

在 Windows 系統上，您可以使用 **setmqcrl** 控制指令，在 Active Directory 中發佈現行 CRL 資訊。

指令 **setmqcrl** 不會發佈 OCSP 資訊。

如需此指令及其語法的相關資訊，請參閱 setmqcrl。

使用 *IBM MQ classes for Java* 及 *IBM MQ classes for JMS* 存取 *CRL* 及 *ARL*

IBM MQ classes for Java 和 IBM MQ classes for JMS 存取 CRL 與其他平台不同。

如需使用 CRL 及 ARL 搭配 IBM MQ classes for Java 的相關資訊，請參閱 使用憑證撤銷清冊

如需使用 CRL 及 ARL 搭配 IBM MQ classes for JMS 的相關資訊，請參閱 SSLCERTSTORES 物件內容

操作鑑別資訊物件

您可以使用 MQSC 或 PCF 指令或 IBM MQ Explorer 來操作鑑別資訊物件。

下列 MQSC 指令會處理鑑別資訊物件:

- DEFINE AUTHINFO
- ALTER AUTHINFO
- DELETE AUTHINFO
- DISPLAY AUTHINFO

如需這些指令的完整說明，請參閱 MQSC 指令。

下列「可程式化指令格式 (PCF)」指令作用於鑑別資訊物件:

- 建立鑑別資訊
- 複製鑑別資訊
- 變更鑑別資訊
- 刪除鑑別資訊
- 查詢鑑別資訊
- 查詢鑑別資訊名稱

如需這些指令的完整說明，請參閱 可程式指令格式的定義。

在可使用它的平台上，您也可以使用 IBM MQ Explorer。

Linux AIX 使用外掛鑑別方法 (PAM)

您只能在 AIX and Linux 平台上使用 PAM。 一般 AIX 或 Linux 系統具有實作傳統鑑別機制的 PAM 模組; 不過，可能還有其他模組。 除了驗證密碼的基本作業之外，也可以呼叫 PAM 模組來執行其他規則。

配置檔定義每一個應用程式要使用的鑑別方法。 範例應用程式包括標準終端機登入、ftp 及 telnet。

PAM 的優點是應用程式不需要知道或關心使用者 ID 的實際鑑別方式。 只要應用程式可以提供正確形式的鑑別資料給 PAM，其背後的機制是透通的。

鑑別資料的形式取決於所使用的系統。 例如， IBM MQ 會透過參數取得密碼，例如 MQCONNX API 呼叫中使用的 MQCSP 結構。

重要: 在安裝 IBM MQ 8.0.0 Fix Pack 3 之前，您無法設定 **AUTHENMD** 屬性，然後使用 **-e CMDLEVEL=**層次 *802* (在 strmqm 指令上) 來設定所需的指令層次，以重新啟動佇列管理程式。

配置系統以使用 PAM

當呼叫 PAM 時， IBM MQ 所使用的服務名稱是 *ibmmq*。

請注意， IBM MQ 安裝會根據不同作業系統的已知預設值，嘗試維護預設 PAM 配置，以允許來自作業系統使用者的連線。

不過，您的系統管理者必須驗證 `/etc/pam.conf` 或 `/etc/pam.d/ibmmq` 檔案中定義的規則仍然適當。

授權存取物件

本節包含使用物件權限管理程式及通道跳出程式來控制物件存取權的相關資訊。

ALW 在 AIX, Linux, and Windows 系統上。 您可以使用物件權限管理程式 (OAM) 來控制對物件的存取權。 此主題集合包含使用 OAM 指令介面的相關資訊。

本節還包含一個核對清單，您可以使用該核對清單來決定要執行哪些作業，以將安全套用至所有平台上的系統，以及授與使用者管理 IBM MQ 及使用 IBM MQ 物件之權限的考量。

如果提供的安全機制不符合您的需求，您可以開發自己的通道結束程式。

判斷用於授權的使用者

存取資源的權限會授與使用者所屬的群組，或在特定模式下，直接授與與連線相關聯的使用者。 在連線處理程序期間，尤其是針對遠端 (用戶端) 連線，佇列管理程式的配置可以變更此身分。 此頁面列出 IBM MQ 的不同特性及其配置選項，這些可能影響連接應用程式的身分，以及這些特性生效的優先順序。

可修改採用哪個使用者的特性

可以設定哪些使用者應該獲得授權的不同特性如下:

應用程式主張的使用者
當 IBM MQ 啟動遠端連線時，執行處理程序的作業系統使用者會傳送至接收端佇列管理程式。 傳送此使用者以確保如果不存在修改使用者的進一步配置，則存在可用於授權檢查的使用者。

不建議使用此使用者作為授權的基礎，因為它容許連線主張其身分，而不需要任何伺服器端驗證。 這甚至可能包括管理使用者 ('mqm ')。

通道 MCAUSER 設定
透過網路連結連接的應用程式會使用 IBM MQ 通道定義來執行此動作。 通道定義支援 **MCAUSER** 屬性，可用來指定要用於授權的不同使用者，而不是連接應用程式所主張的使用者。

連線鑑別 ADOPTCTX
應用程式可以指定要傳送至佇列管理程式以進行鑑別的使用者和密碼。 這些認證是使用指定給「連線鑑別」特性的配置來鑑別。 「連線鑑別」的 **ADOPTCTX** 選項控制在順利驗證之後，是否應該使用使用者來進行授權。 如果設為 YES，則會採用提供鑑別的使用者來進行授權檢查。

通道鑑別記錄 MCAUSER
在連線處理期間，佇列管理程式會嘗試尋找符合連線的通道鑑別記錄。 如果通道鑑別記錄相符，且其 **USERSRC** 屬性值設為 MAP，則 IBM MQ 會將用於授權的使用者變更為 **MCAUSER** 屬性的值。

安全結束程式
安全結束程式是可在 IBM MQ 安全處理期間寫入及呼叫的自訂函數。 當呼叫此函數時，它會隨附 MQCD 結構的副本，其中包括與使用者將用於授權檢查的連線相關的數個欄位。 安全結束程式可以修改這些欄位，以變更將獲授權的使用者。

優先順序

下表顯示當 IBM MQ 選取要授權的使用者時， 第 292 頁的『可修改採用哪個使用者的特性』 中所說明的每一個安全特性的優先順序。 順序從最低到最高，即使用者在第一列的安全特性設定會被任何其他列置換。

表 *68:* 安全特性的優先順序

訂購	**特性**
1 (最低)	主張的應用程式 ID
2	通道定義 **MCAUSER** 屬性

表 *68:* 安全特性的優先順序 *(*繼續*)*	
訂購	**特性**
3	使用 **ADOPTCTX(YES)** 進行連線鑑別
4	使用 **USERSRC(MAP)** 的通道鑑別記錄
5 (最高)	安全結束程式

早期採用的影響

連線鑑別及通道鑑別記錄提供配置選項，可控制何時執行連線鑑別使用者採用。 此設定稱為早期採用。 如果已啟用早期採用，則會在處理通道鑑別記錄之前進行連線鑑別身分採用 (表示通道鑑別記錄會置換任何 **CONNAUTH** 採用)。

如果已停用，則會反轉順序-亦即，在 **CONNAUTH** 採用之前會先處理通道鑑別記錄。 在此狀況下，採用連線鑑別具有通道鑑別記錄的更高有效優先順序。

早期採用的預設值是 enabled。

ALW 在 AIX, Linux, and Windows 上使用 OAM 來控制對物件的存取權

物件權限管理程式 (OAM) 提供指令介面來授與及撤銷 IBM MQ 物件的權限。

您必須獲得適當授權，才能使用這些指令，如 第 334 頁的『在 AIX, Linux, and Windows 上管理 IBM MQ 的權限』中所述。 獲授權管理 IBM MQ 的使用者 ID 具有佇列管理程式的 超級使用者 權限，這表示您不需要授與他們進一步許可權來發出任何 MQI 要求或指令。

Linux AIX AIX and Linux 上的 OAM 使用者型許可權

從 IBM MQ 8.0，在 UNIX and Linux 系統上，物件權限管理程式 (OAM) 可以使用使用者型授權以及群組型授權。

在 IBM MQ 8.0 之前， UNIX and Linux 上的存取控制清單 (ACL) 僅基於群組。 從 IBM MQ 8.0 開始， ACL 同時以使用者 ID 及群組為基礎，而且您可以使用使用者型模型或群組型模型進行授權，方法是將 **SecurityPolicy** 屬性設為適當的值，如 配置可安裝的服務 及 在 AIX and Linux 上配置授權服務段落中所述。

IBM MQ 8.0 以及更新版本的行為變更

從 IBM MQ 8.0，使用使用者型原則執行時，部分指令會傳回與舊版產品不同的資訊:

- **dmpmqaut** 和 **dmpmqcfg** 指令會顯示使用者型記錄，以及 PCF 對等作業。
- IBM MQ Explorer 的 OAM 外掛程式會顯示使用者型記錄，並容許使用者型修改。
- OAM **Inquire** 函數會傳回顯示它具有使用者功能的結果。

在 qm.ini 檔案中啟用使用者型授權 (如 qm.ini 檔案的服務段落中所述) 時，在 **setmqaut** 指令上使用 **-p** 屬性不會將存取權授與相同主要群組中的所有使用者。

如果您開始採用使用者型授權，且有許多使用者，則 AUTH 佇列中儲存的記錄數可能比群組型模型的記錄數還要多，且授權處理程序可能需要比先前更長的時間，因為要驗證的記錄數更多。 預計這一增幅不會很大。 必要的話，您可以混合使用使用者和群組許可權。

移轉考量

如果您將現有佇列管理程式的模型從群組變更為使用者，則不會立即生效。 已進行的授權會繼續套用。 任何連接至佇列管理程式的使用者都會收到與之前相同的專用權: 其 ID 所屬所有群組的組合。 對使用者 ID 發出新的 **setmqaut** 指令時，它們會立即生效。

如果您使用使用者原則建立新的佇列管理程式，則此佇列管理程式僅對建立它的使用者具有許可權 (通常但不一定是 mqm 使用者 ID)。 也有自動授與 mqm 群組的許可權。 不過，如果您沒有 mqm 作為主要群組，則 mqm 群組不會併入起始授權集中。

如果您從使用者移至群組原則，則不會自動刪除使用者型授權。 不過，在許可權檢查期間不再使用它們。 在回復原則之前，請先儲存現行配置，變更原則，重新啟動佇列管理程式，然後重播 Script。 因為它現在是群組型佇列管理程式，所以會根據主要群組來儲存使用者 ID 規則。

相關概念

Object Authority Manager (OAM)

UNIX、 Linux 及 Windows 上的主體及群組

qm.ini 檔案的服務段落

相關參考

crtmqm (建立佇列管理程式) 指令

ALW 授與 AIX, Linux, and Windows 上 IBM MQ 物件的存取權

使用 **setmqaut** 控制指令、 **SET AUTHREC** MQSC 指令或 **MQCMD_SET_AUTH_REC** PCF 指令，為使用者及使用者群組提供 IBM MQ 物件的存取權。 請注意，在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

如需 **setmqaut** 控制指令及其語法的完整定義，請參閱 setmqaut。

如需 **SET AUTHREC** MQSC 指令及其語法的完整定義，請參閱 SET AUTHREC。

如需 **MQCMD_SET_AUTH_REC** PCF 指令及其語法的完整定義，請參閱 設定權限記錄。

佇列管理程式必須在執行中，才能使用這個指令。 當您變更主體的存取權時， OAM 會立即反映變更。

若要授與使用者對物件的存取權，您需要指定:

- 擁有您正在使用之物件的佇列管理程式名稱; 如果您未指定佇列管理程式的名稱，則會採用預設佇列管理程式。
- 物件的名稱和類型 (用來唯一識別物件)。 您將名稱指定為 設定檔 ; 這是物件的明確名稱，或通用名稱 (包括萬用字元)。 如需通用設定檔的詳細說明，以及在其中使用萬用字元，請參閱 第 295 頁的『在 AIX, Linux, and Windows 上使用 OAM 通用設定檔』。
- 套用權限的一或多個主體和群組名稱。

  如果使用者 ID 包含空格，當您使用這個指令時，請以引號括住它。 在 Windows 系統上，您可以使用網域名稱來限定使用者 ID。 如果實際使用者 ID 包含 at 符號 (@) ，請將它取代為 @ @ ， 以顯示它是使用者 ID 的一部分，而不是使用者 ID 與網域名稱之間的定界字元。
- 授權清單。 清單中的每一個項目都會指定要授與該物件 (或從中撤銷) 的存取權類型。 清單中的每一個授權都指定為關鍵字，並以加號 (+) 或減號 (-) 作為字首。 請使用加號來新增指定的授權，並使用減號來移除授權。 + 或-符號與關鍵字之間不得有空格。

  您可以在單一指令中指定任意數目的授權。 例如，允許使用者或群組將訊息放入佇列並瀏覽它們，但撤銷取得訊息的存取權的授權清單為:

```
+browse -get +put
```

使用 setmqaut 指令的範例

下列範例顯示如何使用 setmqaut 指令來授與及撤銷使用物件的許可權:

```
setmqaut -m saturn.queue.manager -t queue -n RED.LOCAL.QUEUE
         -g groupa +browse -get +put
```

在此範例中:

- saturn.queue.manager 是佇列管理程式名稱
- queue 是物件類型

- RED.LOCAL.QUEUE 是物件名稱
- groupa 是具有要變更之授權的群組 ID
- +browse -get +put 是指定佇列的授權清單
  - +browse 新增對佇列上瀏覽訊息的授權 (使用瀏覽選項發出 **MQGET**)
  - -get 會移除從佇列取得 (**MQGET**) 訊息的授權
  - +put 會新增佇列中放置 (**MQPUT**) 訊息的授權

下列指令會從主體 fvuser 以及群組 groupa 和 groupb 撤銷佇列 MyQueue 的放置權限。 在 AIX and Linux 系統上，此指令也會撤銷與 fvuser 相同的主要群組中所有主體的放置權限。

```
setmqaut -m saturn.queue.manager -t queue -n MyQueue -p fvuser
         -g groupa -g groupb -put
```

搭配使用 setmqaut 指令與不同的授權服務

如果您是使用自己的授權服務而非 OAM，則可以在 **setmqaut** 指令上指定此服務的名稱，以將指令導向此服務。 如果您同時有多個可安裝元件在執行中，則必須指定此參數; 如果您沒有執行，則會對授權服務的第一個可安裝元件進行更新。 依預設，這是提供的 OAM。

SET AUTHREC 的使用注意事項

要新增的授權清單和要移除的授權清單不能重疊。 例如，不能使用同一個指令新增顯示權限和移除顯示權限。 即使使用不同的選項來表示權限，也適用此規則。 例如，下列指令由於 DSP 權限與 ALLADM 權限重疊而失敗:

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(DSP) AUTHRMV(ALLADM)
```

此重疊行為的例外狀況是 ALL 權限。 下列指令會先新增 ALL 權限，然後移除 SETID 權限:

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(ALL) AUTHRMV(SETID)
```

下列指令會先移除 ALL 權限，然後新增 DSP 權限:

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(DSP) AUTHRMV(ALL)
```

無論在指令上提供這些權限的順序為何，都會先處理 ALL。

ALW 在 AIX, Linux, and Windows 上使用 OAM 通用設定檔

在單一作業中，使用 OAM 通用設定檔來設定使用者對許多物件的專用權; 而不必在建立時針對每一個個別物件發出個別 **setmqaut** 指令或 **SET AUTHREC** 指令。 請注意，在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

在 setmqaut 或 SET AUTHREC 指令中使用通用設定檔，可讓您針對符合該設定檔的所有物件設定通用權限。

這個主題集合更詳細地說明通用設定檔的用法。

在 OAM 設定檔中使用萬用字元

使設定檔成為通用的是在設定檔名稱中使用特殊字元 (萬用字元)。 例如，問號 (?) 萬用字元符合名稱中的任何單一字元。 因此，如果您指定 ABC.?EF，您提供給該設定檔的授權會套用至名稱為 ABC.DEF、ABC.CEF、 ABC.BEF 等的任何物件。

可用的萬用字元如下:

?
請使用問號 (?)，而不是任何單一字元。 例如， AB.?D 適用於物件 AB.CD、 AB.ED 和 AB.FD。

*****
使用星號 (*) 作為:

- 設定檔名稱中的 限定元，符合物件名稱中的任何一個限定元。 限定元為物件名稱的一部分，以句點區隔。 例如，在 ABC.DEF.GHI 中，限定元為 ABC、DEF 及 GHI。

  例如， ABC.*.JKL 會套用至物件 ABC.DEF.JKL 及 ABC.GHI.JKL。 (請注意，它 **不** 適用於 ABC.JKL ; * used in this context always indicates one qualifier.)
- 設定檔名稱中限定元內的字元，符合物件名稱中限定元內零個以上的字元。

  例如， ABC.DE*.JKL 適用於物件 ABC.DE.JKL、 ABC.DEF.JKL 和 ABC.DEGH.JKL。

******

在設定檔名稱中使用雙星號 (**) 一**次** :

- 符合所有物件名稱的整個設定檔名稱。 例如，如果您使用 -t prcs 來識別處理程序，然後使用 ** 作為設定檔名稱，則會變更所有處理程序的授權。
- 作為設定檔名稱中的開始、中間或結束限定元，以符合物件名稱中的零個以上限定元。 例如， **.ABC 會識別具有最終限定元 ABC 的所有物件。

您只能使用雙星號 ** 作為完整限定元:

```
**.DEF
  ABC.**
  A*.**
```

但不像

```
A**
```

否則，您會收到訊息 AMQ7226E: **設定檔名稱無效。**

註: 在 AIX and Linux 系統上使用萬用字元時，您 **必須** 以單引號括住設定檔名稱。

設定檔優先順序

當使用通用設定檔時，要瞭解的重要點是在決定要套用至所建立物件的權限時，提供設定檔的優先順序。 例如，假設您已發出下列指令:

```
setmqaut -n AB.* -t q +put -p fred
setmqaut -n AB.C* -t q +get -p fred
```

第一個會提供對主體 fred 的所有佇列的放置權限，這些佇列的名稱符合設定檔 AB. *; 第二個會提供取得權限給符合設定檔 AB.C*。

假設您現在建立名為 AB.CD。 根據萬用字元比對的規則， setmqaut 可以套用至該佇列。 所以它是有權力還是有權力?

若要尋找答案，您可以套用規則，每當多個設定檔可以套用至物件時， **只會套用最特定的**。 您套用此規則的方式是從左到右比較設定檔名稱。 無論它們有何不同，非一般字元比一般字元更具體。 因此，在此範例中，是佇列 AB.CD 具有 **get** 權限 (AB.C* 比 AB. *) 更具體。

當您比較一般字元時， 特定性 的順序如下:

1. ?
2. *
3. **

傾出設定檔設定

如需 **dmpmqaut** 控制指令及其語法的完整定義，請參閱 dmpmqaut。

如需 **DISPLAY AUTHREC** MQSC 指令及其語法的完整定義，請參閱 DISPLAY AUTHREC。

如需 **MQCMD_INQUIRE_AUTH_RECS** PCF 指令及其語法的完整定義，請參閱 查詢權限記錄。

下列範例顯示使用 **dmpmqaut** 控制指令來傾出通用設定檔的權限記錄:

1. 此範例會針對主體 user1，傾出其設定檔符合佇列 a.b.c 的所有權限記錄。

```
dmpmqaut -m qm1 -n a.b.c -t q -p user1
```

產生的傾出看起來如下:

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

註: 雖然 AIX and Linux 上的使用者可以對 **dmpmqaut** 指令使用 -p 選項，但在定義授權時必須改用 -g groupname 。

2. 此範例會傾出具有符合佇列 a.b.c 之設定檔的所有權限記錄。

```
dmpmqaut -m qmgr1 -n a.b.c -t q
```

產生的傾出看起來如下:

```
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. 此範例會傾出設定檔 a.b. * 的所有權限記錄， 類型為佇列的。

```
dmpmqaut -m qmgr1 -n a.b.* -t q
```

產生的傾出看起來如下:

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

4. 此範例會傾出佇列管理程式 qmX 的所有權限記錄。

```
dmpmqaut -m qmX
```

產生的傾出看起來如下:

```
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. 此範例會傾出佇列管理程式 qmX 的所有設定檔名稱及物件類型。

```
dmpmqaut -m qmX -l
```

產生的傾出看起來如下:

```
profile: q1, type: queue
profile: q*, type: queue
profile: name.*, type: namelist
profile: pr1, type: process
```

註: 僅適用於 IBM MQ for Windows , 所有顯示的主體都包括網域資訊, 例如:

```
profile:     a.b.*
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

ALW 在 *AIX, Linux, and Windows* 上的 *OAM* 設定檔中使用萬用字元

在物件權限管理程式 (OAM) 設定檔名稱中使用萬用字元, 使該設定檔適用於多個物件。

使設定檔成為通用的是在設定檔名稱中使用特殊字元 (萬用字元)。 例如, 問號 (?) 萬用字元符合名稱中的任何單一字元。 因此, 如果您指定 ABC.?EF, 您提供給該設定檔的授權會套用至名稱為 ABC.DEF、ABC.CEF、 ABC.BEF 等的任何物件。

可用的萬用字元如下:

?

請使用問號 (?), 而不是任何單一字元。 例如, AB.?D 適用於物件 AB.CD、 AB.ED 和 AB.FD。

*****

使用星號 (*) 作為:

- 設定檔名稱中的 限定元 , 符合物件名稱中的任何一個限定元。 限定元為物件名稱的一部分, 以句點區隔。 例如, 在 ABC.DEF.GHI 中, 限定元為 ABC、DEF 及 GHI。

  例如, ABC.*.JKL 會套用至物件 ABC.DEF.JKL 及 ABC.GHI.JKL。 (請注意, 它 **不** 適用於 ABC.JKL ; * used in this context always indicates one qualifier.)
- 設定檔名稱中限定元內的字元, 符合物件名稱中限定元內零個以上的字元。

  例如, ABC.DE*.JKL 適用於物件 ABC.DE.JKL、 ABC.DEF.JKL 和 ABC.DEGH.JKL。

******

在設定檔名稱中使用雙星號 (**) **一次** :

- 符合所有物件名稱的整個設定檔名稱。 例如, 如果您使用 -t prcs 來識別處理程序, 然後使用 ** 作為設定檔名稱, 則會變更所有處理程序的授權。
- 作為設定檔名稱中的開始、中間或結束限定元, 以符合物件名稱中的零個以上限定元。 例如, **.ABC 會識別具有最終限定元 ABC 的所有物件。

註: 在 AIX and Linux 系統上使用萬用字元時, 您 **必須** 以單引號括住設定檔名稱。

ALW *AIX, Linux, and Windows* 上的設定檔優先順序

多個通用設定檔可以套用至單一物件。 在這種情況下, 適用最具體的規則。

當使用通用設定檔時, 要瞭解的重要點是在決定要套用至所建立物件的權限時, 提供設定檔的優先順序。 例如, 假設您已發出下列指令:

```
setmqaut -n AB.* -t q +put -p fred
setmqaut -n AB.C* -t q +get -p fred
```

第一個會提供對主體 fred 的所有佇列的放置權限，這些佇列的名稱符合設定檔 AB. *; 第二個會提供取得權限給符合設定檔 AB.C*。

假設您現在建立名為 AB.CD。 根據萬用字元比對的規則， setmqaut 可以套用至該佇列。 所以它是有權力還是有權力?

若要尋找答案，您可以套用規則，每當多個設定檔可以套用至物件時， **只會套用最特定的**。 您套用此規則的方式是從左到右比較設定檔名稱。 無論它們有何不同，非一般字元比一般字元更具體。 因此，在此範例中，是佇列 AB.CD 具有 **get** 權限 (AB.C* 比 AB. *) 更具體。

當您比較一般字元時， 特定性 的順序如下:

1. ?
2. *
3. **

請參閱 SET AUTHREC ， 以取得使用此 MQSC 指令時的對等資訊。

ALW 在 *AIX, Linux, and Windows* 上傾出設定檔設定

使用 **dmpmqaut** 控制指令、 **DISPLAY AUTHREC** MQSC 指令或 **MQCMD_INQUIRE_AUTH_RECS** PCF 指令來傾出與指定設定檔相關聯的現行授權。 請注意，在 IBM MQ Appliance 上，您只能使用 **DISPLAY AUTHREC** 指令。

如需 **dmpmqaut** 控制指令及其語法的完整定義，請參閱 dmpmqaut。

如需 **DISPLAY AUTHREC** MQSC 指令及其語法的完整定義，請參閱 DISPLAY AUTHREC。

如需 **MQCMD_INQUIRE_AUTH_RECS** PCF 指令及其語法的完整定義，請參閱 查詢權限記錄。

下列範例顯示使用 **dmpmqaut** 控制指令來傾出通用設定檔的權限記錄:

1. 此範例會針對主體 user1，傾出其設定檔符合佇列 a.b.c 的所有權限記錄。

   ```
   dmpmqaut -m qm1 -n a.b.c -t q -p user1
   ```

   產生的傾出類似下列範例:

   ```
   profile:     a.b.*
   object type: queue
   entity:      user1
   type:        principal
   authority:   get, browse, put, inq
   ```

   註: AIX and Linux 使用者無法使用 `-p` 選項; 他們必須改用 `-g groupname` 。
2. 此範例會傾出具有符合佇列 a.b.c 之設定檔的所有權限記錄。

   ```
   dmpmqaut -m qmgr1 -n a.b.c -t q
   ```

   產生的傾出類似下列範例:

   ```
   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. 此範例會傾出設定檔 a.b. * 的所有權限記錄， 類型為佇列的。

```
dmpmqaut -m qmgr1 -n a.b.* -t q
```

產生的傾出類似下列範例:

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

4. 此範例會傾出佇列管理程式 qmX 的所有權限記錄。

```
dmpmqaut -m qmX
```

產生的傾出類似下列範例:

```
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. 此範例會傾出佇列管理程式 qmX 的所有設定檔名稱及物件類型。

```
dmpmqaut -m qmX -l
```

產生的傾出類似下列範例:

```
profile: q1, type: queue
profile: q*, type: queue
profile: name.*, type: namelist
profile: pr1, type: process
```

註: 僅適用於 IBM MQ for Windows，所有顯示的主體都包括網域資訊，例如:

```
profile:     a.b.*
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

ALW 在 AIX, Linux, and Windows 上顯示存取設定

使用 **dspmqaut** 控制指令、 **DISPLAY AUTHREC** MQSC 指令或 **MQCMD_INQUIRE_ENTITY_AUTH** PCF 指令來檢視特定主體或群組對特定物件的授權。 請注意，在 IBM MQ Appliance 上，您只能使用 **DISPLAY AUTHREC** 指令。

佇列管理程式必須在執行中，才能使用這個指令。 當您變更主體的存取權時， OAM 會立即反映這些變更。一次只能顯示一個群組或主體的授權。

如需 **dmpmqaut** 控制指令及其語法的完整定義，請參閱 dmpmqaut。

如需 **DISPLAY AUTHREC** MQSC 指令及其語法的完整定義，請參閱 DISPLAY AUTHREC。

如需 **MQCMD_INQUIRE_AUTH_RECS** PCF 指令及其語法的完整定義，請參閱 查詢權限記錄。

下列範例顯示使用 **dspmqaut** 控制指令，以顯示群組 GpAdmin 對佇列管理程式 QueueMan1 上名為 Annuities 之程序定義的授權。

```
dspmqaut -m QueueMan1 -t process -n Annuities -g GpAdmin
```

ALW 變更及撤銷 AIX, Linux, and Windows 上 IBM MQ 物件的存取權

若要變更使用者或群組對物件的存取層次，請使用 **setmqaut** 控制指令、 **DELETE AUTHREC** MQSC 指令或 **MQCMD_DELETE_AUTH_REC** PCF 指令。 MQ Appliance請注意，在 IBM MQ Appliance 上，您只能使用 **DELETE AUTHREC** 指令。

從群組中移除使用者的處理程序說明如下:

- Windows 第 121 頁的『在 Windows 上建立及管理群組』
- AIX 第 120 頁的『在 AIX 上建立及管理群組』
- Linux 第 121 頁的『在 Linux 上建立及管理群組』

.

建立 IBM MQ 物件的使用者 ID 會被授與對該物件的完整控制權限。 如果您從本端 mqm 群組 (或 Windows 系統上的 Administrators 群組) 移除此使用者 ID，則不會撤銷這些權限。 從 mqm 或 Administrators 群組中移除物件之後，請使用 **setmqaut** 控制指令或 **MQCMD_DELETE_AUTH_REC** PCF 指令來撤銷建立物件之使用者 ID 的物件存取權。

如需 setmqaut 控制指令及其語法的完整定義，請參閱 setmqaut。

如需 **DELETE AUTHREC** MQSC 指令及其語法的完整定義，請參閱 DELETE AUTHREC。

如需 **MQCMD_DELETE_AUTH_REC** PCF 指令及其語法的完整定義，請參閱 刪除權限記錄。

Windows 在 Windows 上，從 IBM MQ 8.0 中，您隨時可以使用 **setmqaut** 的 **-u** *SID* 參數來刪除對應於特定 Windows 使用者帳戶的 OAM 項目。

在 IBM MQ 8.0 之前，您必須先刪除對應於特定 Windows 使用者帳戶的 OAM 項目，然後再刪除使用者設定檔。 在移除使用者帳戶之後，無法移除 OAM 項目。

ALW 在 AIX, Linux, and Windows 系統上防止安全存取檢查

附註: 本主題說明不建議啟用的功能。 若要關閉安全檢查，您可以停用物件權限管理程式 (OAM)。 這可能適用於測試環境。 停用時，佇列管理程式將無法再執行授權或連線鑑別檢查。 仍然可以使用 TLS、通道鑑別記錄及安全結束程式。 停用或移除 OAM 之後，您無法將 OAM 新增至現有的佇列管理程式。

如果您決定不要執行安全檢查 (例如，在測試環境中)，您可以使用下列兩種方式之一來停用 OAM:

- 在建立佇列管理程式之前，請設定作業系統環境變數 MQSNAUT。

  如需設定 MQSNAUT 變數的含意以及如何在 AIX, Linux, and Windows 上設定 MQSNAUT 的相關資訊，請參閱 環境變數說明。
- 編輯佇列管理程式配置檔以移除服務。

警告: 當移除 OAM 時，無法將它放回現有的佇列管理程式。 這是因為 OAM 需要在物件建立時就位。 若要在移除 IBM MQ OAM 之後再次使用它，請重建佇列管理程式。

如果您在停用 OAM 時使用 **setmqaut** 或 **dspmqaut** 指令，請注意下列要點:

- OAM 不會驗證指定的主體或群組，這表示指令可以接受無效值。
- OAM 不會執行安全檢查，並指出所有主體和群組都已獲授權執行所有適用的物件作業。
- 不會驗證任何傳遞至 OAM 以進行鑑別檢查的認證。

相關工作
配置可安裝的服務
相關參考
UNIX、 Linux 及 Windows 的可安裝服務及元件
可安裝的服務參照資訊

授與對資源的必要存取權

請利用這個主題來決定要執行哪些作業，以將安全套用至 IBM MQ 系統。

關於這項作業

在此作業期間，您可以決定將適當的安全層次套用至 IBM MQ 安裝的元素所需的動作。 您所參照的每一項個別作業都會提供所有平台的逐步指示。

程序

1. 您是否需要將佇列管理程式的存取權限制為特定使用者?
   a) 否: 不採取進一步動作。
   b) 是: 請跳至下一個問題。
2. 這些使用者是否需要對佇列管理程式資源子集的局部管理存取權?
   a) 否: 請跳至下一個問題。
   b) 是: 請參閱 第 302 頁的『授與對佇列管理程式資源子集的局部管理存取權』。
3. 這些使用者是否需要佇列管理程式資源子集的完整管理存取權?
   a) 否: 請跳至下一個問題。
   b) 是: 請參閱 第 310 頁的『授與佇列管理程式資源子集的完整管理存取權』。
4. 這些使用者是否需要所有佇列管理程式資源的唯讀存取權?
   a) 否: 請跳至下一個問題。
   b) 是: 請參閱 第 315 頁的『授與佇列管理程式上所有資源的唯讀存取權』。
5. 這些使用者是否需要所有佇列管理程式資源的完整管理存取權?
   a) 否: 請跳至下一個問題。
   b) 是: 請參閱 第 316 頁的『授與佇列管理程式上所有資源的完整管理存取權』。
6. 您需要使用者應用程式來連接至佇列管理程式嗎?
   a) 否: 停用連線功能，如 第 317 頁的『移除佇列管理程式的連線功能』 中所述
   b) 是: 請參閱 第 318 頁的『容許使用者應用程式連接至佇列管理程式』。

z/OS Multi 授與對佇列管理程式資源子集的局部管理存取權

您需要將部分 (而非全部) 佇列管理程式資源的局部管理存取權提供給特定使用者。 使用此表格來決定您需要採取的動作。

表 *69:* 授與部分管理存取權給佇列管理程式資源的子集

使用者需要管理此類型的物件	**執行此動作**
佇列	授與對所需佇列的局部管理存取權，如 第 303 頁的『授與對部分佇列的有限管理存取權』 中所述
主題	授與對必要主題的局部管理存取權，如 第 304 頁的『授與部分主題的有限管理存取權』 中所述
通道	授與對必要通道的局部管理存取權，如 第 305 頁的『授與對部分通道的有限管理存取權』 中所述

表 69: 授與部分管理存取權給佇列管理程式資源的子集 (繼續)	
使用者需要管理此類型的物件	**執行此動作**
佇列管理程式	授與局部管理存取權給佇列管理程式，如 第 306 頁的『授與對佇列管理程式的有限管理存取權』 中所述
Processes	授與對必要處理程序的局部管理存取權，如 第 307 頁的『授與對部分處理程序的有限管理存取權』 中所述
名單	授與部分管理存取權給必要的名稱清單，如 第 308 頁的『授與部分名稱清單的有限管理存取權』 中所述
服務	授與對必要服務的局部管理存取權，如 第 309 頁的『授與部分服務的有限管理存取權』 中所述

授與對部分佇列的有限管理存取權

將佇列管理程式上某些佇列的局部管理存取權授與每一個具有商業需求的使用者群組。

關於這項作業

若要針對部分動作授與部分佇列的有限管理存取權，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- ALW
  若為 AIX, Linux, and Windows 系統，請發出下列指令:

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

- IBM i
  若為 IBM i，請發出下列指令:

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

- z/OS 對於 z/OS，請發出下列指令，以授與對指定佇列的存取權:

  ```
  RDEFINE MQADMIN QMgrName.QUEUE. ObjectProfile UACC(NONE)
  PERMIT QMgrName.QUEUE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  若要指定使用者可以在佇列上執行哪些 MQSC 指令，請針對每一個 MQSC 指令發出下列指令:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction. QType UACC(NONE)
  PERMIT QMgrName. ReqdAction. QType CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  若要允許使用者使用 DISPLAY QUEUE 指令，請發出下列指令:

  ```
  RDEFINE MQCMDS QMgrName.DISPLAY. QType UACC(NONE)
  PERMIT QMgrName.DISPLAY. QType CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  變數名稱具有下列意義:

  QMgrName
  佇列管理程式的名稱。

z/OS 在 z/OS 上，此值也可以是佇列共用群組的名稱。

ObjectProfile
要變更其授權的物件或通用設定檔名稱。

GroupName
要授與存取權的群組名稱。

ReqdAction
您容許群組採取的動作:

- ALW 在 AIX, Linux, and Windows 系統上，下列授權的任何組合: + chg、+ clr、+ dlt、+ dsp。 authorization + alladm 相當於 + chg + clr + dlt + dsp。
- IBM i 在 IBM i 上，下列授權的任何組合: *ADMCHG、*ADMCLR、*ADMDLT、*ADMDSP。 授權 *ALLADM 相當於所有這些個別授權。
- z/OS 在 z/OS 上，為 ALTER、CLEAR、DELETE 或 MOVE 其中一個值。

註: 對佇列授與 + crt 會間接使使用者或群組成為管理者。 請勿使用 + crt 權限來授與對部分佇列的有限管理存取權。

QTYPE

對於 DISPLAY 指令，為 QUEUE、QLOCAL、QALIAS、QMODEL、QREMOTE 或 QCLUSTER 其中一個值。

對於 *ReqdAction* 的其他值，為 QLOCAL、QALIAS、QMODEL 或 QREMOTE 值之一。

授與部分主題的有限管理存取權

將佇列管理程式上部分主題的局部管理存取權授與每一個具有商業需求的使用者群組。

關於這項作業

若要針對部分動作授與部分主題的有限管理存取權，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- ALW
  若為 AIX, Linux, and Windows 系統，請發出下列指令:

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

- IBM i
  若為 IBM i，請發出下列指令:

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

- z/OS 若為 z/OS，請發出下列指令:

```
RDEFINE MQADMIN QMgrName.TOPIC. ObjectProfile UACC(NONE)
PERMIT QMgrName.TOPIC. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

這些指令會授與對指定主題的存取權。 若要判定使用者可以對主題執行哪些 MQSC 指令，請針對每一個 MQSC 指令發出下列指令:

```
RDEFINE MQCMDS QMgrName. ReqdAction.TOPIC UACC(NONE)
PERMIT QMgrName. ReqdAction.TOPIC CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

若要允許使用者使用 DISPLAY Topic 指令，請發出下列指令:

```
RDEFINE MQCMDS QMgrName.DISPLAY.TOPIC UACC(NONE)
PERMIT QMgrName.DISPLAY.TOPIC CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

變數名稱具有下列意義:

QMgrName
佇列管理程式的名稱。

z/OS 在 z/OS 上，此值也可以是佇列共用群組的名稱。

ObjectProfile
要變更其授權的物件或通用設定檔名稱。

GroupName
要授與存取權的群組名稱。

ReqdAction
您容許群組採取的動作:

- ALW 在 AIX, Linux, and Windows 系統上，下列授權的任何組合: + chg、+ clr、+ crt、+ dlt、+ dsp。 + ctrl。 authorization + alladm 相當於 + chg + clr + dlt + dsp。
- IBM i 在 IBM i 上，下列授權的任何組合: *ADMCHG、*ADMCLR、*ADMCRT、*ADMDLT、*ADMDSP、*CTRL。 授權 *ALLADM 相當於所有這些個別授權。
- z/OS 在 z/OS 上，為 ALTER、CLEAR、DEFINE、DELETE 或 MOVE 其中一個值。

授與對部分通道的有限管理存取權

將佇列管理程式上某些通道的局部管理存取權授與每一個具有商業需求的使用者群組。

關於這項作業

若要針對部分動作授與部分通道的有限管理存取權，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- ALW
  在 AIX, Linux, and Windows 上:

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

- IBM i
  在 IBM i 上:

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

- z/OS 在 z/OS 上:

  ```
  RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
  PERMIT QMgrName.CHANNEL. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  這些指令會授與對指定通道的存取權。 若要判定使用者可以在通道上執行哪些 MQSC 指令，請針對每一個 MQSC 指令發出下列指令:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction.CHANNEL UACC(NONE)
  PERMIT QMgrName. ReqdAction.CHANNEL CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

若要允許使用者使用 DISPLAY CHANNEL 指令，請發出下列指令:

```
RDEFINE MQCMDS QMgrName.DISPLAY.CHANNEL UACC(NONE)
PERMIT QMgrName.DISPLAY.CHANNEL CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

變數名稱具有下列意義:

QMgrName
佇列管理程式的名稱。

z/OS 在 z/OS 上，此值也可以是佇列共用群組的名稱。

ObjectProfile
要變更其授權的物件或通用設定檔名稱。

GroupName
要授與存取權的群組名稱。

ReqdAction
您容許群組採取的動作:

- ALW 在 AIX, Linux, and Windows 上，下列授權的任何組合: + chg、+ clr、+ crt、+ dlt、+ dsp。 + ctrl ， + ctrlx。 authorization + alladm 相當於 + chg + clr + dlt + dsp。
- IBM i 在 IBM i 上，下列授權的任何組合: *ADMCHG、*ADMCLR、*ADMCRT、*ADMDLT、*ADMDSP、*CTRL、*CTRLx。 授權 *ALLADM 相當於所有這些個別授權。
- z/OS 在 z/OS 上，為 ALTER、CLEAR、DEFINE、DELETE 或 MOVE 其中一個值。

授與對佇列管理程式的有限管理存取權

將佇列管理程式的局部管理存取權授與每一個具有商業需求的使用者群組。

關於這項作業

如果要授與有限的管理存取權，以便在佇列管理程式上執行某些動作，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- ALW
  在 AIX, Linux, and Windows 上:

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

- IBM i
  在 IBM i 上:

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

- z/OS
  在 z/OS 上:

  若要判定您可以在佇列管理程式上執行哪些 MQSC 指令，請針對每一個 MQSC 指令發出下列指令:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction.QMGR UACC(NONE)
  PERMIT QMgrName. ReqdAction.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  若要允許使用者使用 DISPLAY QMGR 指令，請發出下列指令:

```
RDEFINE MQCMDS QMgrName.DISPLAY.QMGR UACC(NONE)
PERMIT QMgrName.DISPLAY.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

變數名稱具有下列意義:

QMgrName
佇列管理程式的名稱。

ObjectProfile
要變更其授權的物件或通用設定檔名稱。

GroupName
要授與存取權的群組名稱。

ReqdAction
您容許群組採取的動作:

- ALW 在 AIX, Linux, and Windows 上，下列授權的任何組合: + chg、+ clr、+ crt、+ dlt、+ dsp。 authorization + alladm 相當於 + chg + clr + dlt + dsp。

  雖然 + 集是 MQI 授權，且通常不會被視為管理，但在佇列管理程式上授與 + 集可能會間接導致完整管理權限。 請勿將 + 集授與一般使用者及應用程式。

- IBM i 在 IBM i 上，下列授權的任何組合: *ADMCHG、*ADMCLR、*ADMCRT、*ADMDLT、*ADMDSP。 授權 *ALLADM 相當於所有這些個別授權。

授與對部分處理程序的有限管理存取權

將佇列管理程式上某些處理程序的局部管理存取權授與具有商業需求的每一個使用者群組。

關於這項作業

若要針對某些動作授與部分處理程序的有限管理存取權，請針對您的作業系統使用適當的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- ALW
  在 AIX, Linux, and Windows 上:

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

- IBM i
  在 IBM i 上:

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

- z/OS 在 z/OS 上:

  ```
  RDEFINE MQADMIN QMgrName.PROCESS. ObjectProfile UACC(NONE)
  PERMIT QMgrName.PROCESS. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  這些指令會授與對指定通道的存取權。 若要判定使用者可以在通道上執行哪些 MQSC 指令，請針對每一個 MQSC 指令發出下列指令:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction.PROCESS UACC(NONE)
  PERMIT QMgrName. ReqdAction.PROCESS CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  若要允許使用者使用 DISPLAY PROCESS 指令，請發出下列指令:

```
RDEFINE MQCMDS QMgrName.DISPLAY.PROCESS UACC(NONE)
PERMIT QMgrName.DISPLAY.PROCESS CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

變數名稱具有下列意義:

QMgrName
佇列管理程式的名稱。

z/OS 在 z/OS 上，此值也可以是佇列共用群組的名稱。

ObjectProfile
要變更其授權的物件或通用設定檔名稱。

GroupName
要授與存取權的群組名稱。

ReqdAction
您容許群組採取的動作:

- ALW 在 AIX, Linux, and Windows 上，下列授權的任何組合: + chg、+ clr、+ crt、+ dlt、+ dsp。 authorization + alladm 相當於 + chg + clr + dlt + dsp。
- IBM i 在 IBM i 上，下列授權的任何組合: *ADMCHG、*ADMCLR、*ADMCRT、*ADMDLT、*ADMDSP。 授權 *ALLADM 相當於所有這些個別授權。
- z/OS 在 z/OS 上，為 ALTER、CLEAR、DEFINE、DELETE 或 MOVE 其中一個值。

授與部分名稱清單的有限管理存取權

將部分管理存取權授與佇列管理程式上的部分名稱清單，以及具有商業需求的每一個使用者群組。

關於這項作業

若要針對部分動作授與部分名稱清單的有限管理存取權，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- ALW
  在 AIX, Linux, and Windows 上:
  ```
  setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName ReqdAction
  ```
- IBM i
  在 IBM i 上:
  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*NMLIST) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```
- z/OS 在 z/OS 上:
  ```
  RDEFINE MQADMIN QMgrName.NAMELIST. ObjectProfile UACC(NONE)
  PERMIT QMgrName.NAMELIST. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```
  這些指令會授與指定名稱清單的存取權。 若要判定使用者可以在名單上執行哪些 MQSC 指令，請針對每一個 MQSC 指令發出下列指令:
  ```
  RDEFINE MQCMDS QMgrName. ReqdAction.NAMELIST UACC(NONE)
  PERMIT QMgrName. ReqdAction.NAMELIST CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```
  若要允許使用者使用 DISPLAY NAMELIST 指令，請發出下列指令:

```
RDEFINE MQCMDS QMgrName.DISPLAY.NAMELIST UACC(NONE)
PERMIT QMgrName.DISPLAY.NAMELIST CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

變數名稱具有下列意義:

QMgrName
佇列管理程式的名稱。

z/OS 在 z/OS 上，此值也可以是佇列共用群組的名稱。

ObjectProfile
要變更其授權的物件或通用設定檔名稱。

GroupName
要授與存取權的群組名稱。

ReqdAction
您容許群組採取的動作:

- ALW 在 AIX, Linux, and Windows 上，下列授權的任何組合: + chg、+ clr、+ crt、+ dlt、+ ctrl、+ ctrlx、+ dsp。 authorization + alladm 相當於 + chg + clr + dlt + dsp。
- IBM i 在 IBM i 上，下列授權的任何組合: *ADMCHG、*ADMCLR、*ADMCRT、*ADMDLT、*ADMDSP、*CTRL、*CTRLX。 授權 *ALLADM 相當於所有這些個別授權。
- z/OS 在 z/OS 上，為 ALTER、CLEAR、DEFINE、DELETE 或 MOVE 其中一個值。

授與部分服務的有限管理存取權

將佇列管理程式上部分服務的部分管理存取權授與具有商業需求的每一個使用者群組。

關於這項作業

若要針對部分動作授與部分服務的有限管理存取權，請針對您的作業系統使用適當的指令。 z/OS 請注意，服務物件不存在於 z/OS 上。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- ALW
  在 AIX, Linux, and Windows 上:

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

- 在 IBM i 上:

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

- z/OS 在 z/OS 上:

  這些指令會授與對指定服務的存取權。 若要判定使用者可以對服務執行哪些 MQSC 指令，請針對每一個 MQSC 指令發出下列指令:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction.SERVICE UACC(NONE)
  PERMIT QMgrName. ReqdAction.SERVICE CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  若要允許使用者使用 DISPLAY SERVICE 指令，請發出下列指令:

  ```
  RDEFINE MQCMDS QMgrName.DISPLAY.SERVICE UACC(NONE)
  PERMIT QMgrName.DISPLAY.SERVICE CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

變數名稱具有下列意義:

QMgrName
佇列管理程式的名稱。

ObjectProfile
要變更其授權的物件或通用設定檔名稱。

GroupName
要授與存取權的群組名稱。

ReqdAction
您容許群組採取的動作:

- ALW 在 AIX, Linux, and Windows 系統上，下列授權的任何組合: + chg、+ clr、+ crt、+ dlt、+ ctrl、+ ctrlx、+ dsp。 authorization + alladm 相當於 + chg + clr + dlt + dsp。
- IBM i 在 IBM i 上，下列授權的任何組合: *ADMCHG、*ADMCLR、*ADMCRT、*ADMDLT、*ADMDSP、*CTRL、*CTRLX。 授權 *ALLADM 相當於所有這些個別授權。

授與佇列管理程式資源子集的完整管理存取權

您需要為特定使用者提供部分但不是所有佇列管理程式資源的完整管理存取權。 請使用這些表格來決定您需要採取的動作。

表 *70:* 授與對佇列管理程式資源子集的完整管理存取權

使用者需要管理此類型的物件	**執行此動作**
佇列	授與所需佇列的完整管理存取權，如 第 310 頁的『授與部分佇列的完整管理存取權』 中所述
主題	授與必要主題的完整管理存取權，如 第 311 頁的『授與部分主題的完整管理存取權』 所述
通道	授與所需通道的完整管理存取權，如 第 312 頁的『授與部分通道的完整管理存取權』 中所述
佇列管理程式	授與佇列管理程式的完整管理存取權，如 第 312 頁的『授與佇列管理程式的完整管理存取權』 中所述
Processes	授與必要處理程序的完整管理存取權，如 第 313 頁的『授與部分處理程序的完整管理存取權』 中所述
名單	授與所需名稱清單的完整管理存取權，如 第 314 頁的『授與部分名稱清單的完整管理存取權』 中所述
服務	授與所需服務的完整管理存取權，如 第 314 頁的『授與部分服務的完整管理存取權』 所述

授與部分佇列的完整管理存取權

將佇列管理程式上某些佇列的完整管理存取權授與每一個具有商業需求的使用者群組。

關於這項作業

若要授與部分佇列的完整管理存取權，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- ALW
  在 AIX, Linux, and Windows 上:

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

- IBM i
  在 IBM i 上：

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

- z/OS
  在 z/OS 上：

  ```
  RDEFINE MQADMIN QMgrName.QUEUE. ObjectProfile UACC(NONE)
  PERMIT QMgrName.QUEUE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  變數名稱具有下列意義：

  QMgrName
  佇列管理程式的名稱。

  z/OS 在 z/OS 上，此值也可以是佇列共用群組的名稱。

  ObjectProfile
  要變更其授權的物件或通用設定檔名稱。

  GroupName
  要授與存取權的群組名稱。

授與部分主題的完整管理存取權

將佇列管理程式上部分主題的完整管理存取權授與每一個具有商業需求的使用者群組。

關於這項作業

若要針對部分動作授與部分主題的完整管理存取權，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註： MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- ALW
  在 AIX, Linux, and Windows 上：

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

- IBM i
  在 IBM i 上：

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

- z/OS
  在 z/OS 上：

  ```
  RDEFINE MQADMIN QMgrName.TOPIC. ObjectProfile UACC(NONE)
  PERMIT QMgrName.TOPIC. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  變數名稱具有下列意義：

  QMgrName
  佇列管理程式的名稱。

  z/OS 在 z/OS 上，此值也可以是佇列共用群組的名稱。

ObjectProfile
要變更其授權的物件或通用設定檔名稱。

GroupName
要授與存取權的群組名稱。

授與部分通道的完整管理存取權

將佇列管理程式上部分通道的完整管理存取權授與具有商業需求的每一個使用者群組。

關於這項作業

若要授與部分通道的完整管理存取權，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- ALW
  在 AIX, Linux, and Windows 上:

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

- IBM i
  在 IBM i 上:

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

- z/OS
  在 z/OS 上:

  ```
  RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
  PERMIT QMgrName.CHANNEL. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  變數名稱具有下列意義:

  QMgrName
  佇列管理程式的名稱。

  z/OS 在 z/OS 上，此值也可以是佇列共用群組的名稱。

  ObjectProfile
  要變更其授權的物件或通用設定檔名稱。

  GroupName
  要授與存取權的群組名稱。

授與佇列管理程式的完整管理存取權

將佇列管理程式的完整管理存取權授與每一個具有商業需求的使用者群組。

關於這項作業

若要授與佇列管理程式的完整管理存取權，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- ALW

在 AIX, Linux, and Windows 上：

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

- IBM i
  在 IBM i 上：

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

- z/OS
  在 z/OS 上：

```
RDEFINE MQADMIN QMgrName.QMGR UACC(NONE)
PERMIT QMgrName.QMGR CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

變數名稱具有下列意義：

QMgrName
佇列管理程式的名稱。

z/OS 在 z/OS 上，此值也可以是佇列共用群組的名稱。

ObjectProfile
要變更其授權的物件或通用設定檔名稱。

GroupName
要授與存取權的群組名稱。

授與部分處理程序的完整管理存取權

將佇列管理程式上部分處理程序的完整管理存取權授與具有商業需求的每一個使用者群組。

關於這項作業

若要授與部分處理程序的完整管理存取權，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註： MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- ALW
  在 AIX, Linux, and Windows 上：

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

- IBM i
  在 IBM i 上：

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

- z/OS
  在 z/OS 上：

```
RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
PERMIT QMgrName.PROCESS. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

變數名稱具有下列意義：

QMgrName
佇列管理程式的名稱。

z/OS 在 z/OS 上，此值也可以是佇列共用群組的名稱。

ObjectProfile
要變更其授權的物件或通用設定檔名稱。

GroupName
要授與存取權的群組名稱。

授與部分名稱清單的完整管理存取權

將佇列管理程式上某些名稱清單的完整管理存取權授與具有商業需求的每一個使用者群組。

關於這項作業

若要將完整管理存取權授與部分名稱清單，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- ALW
  在 AIX, Linux, and Windows 上:

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

- IBM i
  在 IBM i 上:

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

- z/OS
  在 z/OS 上:

```
RDEFINE MQADMIN QMgrName.NAMELIST. ObjectProfile UACC(NONE)
PERMIT QMgrName.NAMELIST. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  變數名稱具有下列意義:

  QMgrName
  佇列管理程式的名稱。

  z/OS 在 z/OS 上，此值也可以是佇列共用群組的名稱。

  ObjectProfile
  要變更其授權的物件或通用設定檔名稱。

  GroupName
  要授與存取權的群組名稱。

授與部分服務的完整管理存取權

將佇列管理程式上部分服務的完整管理存取權授與具有商業需求的每一個使用者群組。

關於這項作業

若要授與部分服務的完整管理存取權，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- ALW
  在 AIX, Linux, and Windows 上:

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

- IBM i
  在 IBM i 上:

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

- z/OS
  在 z/OS 上:

```
RDEFINE MQADMIN QMgrName.SERVICE. ObjectProfile UACC(NONE)
PERMIT QMgrName.SERVICE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  變數名稱具有下列意義:

  QMgrName
  佇列管理程式的名稱。

  z/OS 在 z/OS 上，此值也可以是佇列共用群組的名稱。

  ObjectProfile
  要變更其授權的物件或通用設定檔名稱。

  GroupName
  要授與存取權的群組名稱。

授與佇列管理程式上所有資源的唯讀存取權

將佇列管理程式上所有資源的唯讀存取權授與具有商業需求的每一個使用者或使用者群組。

關於這項作業

請使用「新增角色型權限」精靈或適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

變更任何授權詳細資料之後，請使用 REFRESH SECURITY 指令來執行安全重新整理。

程序

- 使用精靈:
  a) 在「 IBM MQ Explorer Navigator 」窗格中，用滑鼠右鍵按一下佇列管理程式，然後按一下 **物件權限 > 新增角色型權限**
  這時會開啟「新增角色型權限」精靈。

- ALW
  若為 AIX, Linux, and Windows 系統，請發出下列指令:

```
setmqaut -m QMgrName -n ** -t queue -g GroupName +browse +dsp
setmqaut -m QMgrName -n SYSTEM.ADMIN.COMMAND.QUEUE -t queue -g GroupName +dsp +inq +put
setmqaut -m QMgrName -n SYSTEM.MQEXPLORER.REPLY.MODEL -t queue -g GroupName +dsp +inq +get
+put
setmqaut -m QMgrName -n ** -t topic -g GroupName +dsp
setmqaut -m QMgrName -n ** -t channel -g GroupName +dsp
setmqaut -m QMgrName -n ** -t clntconn -g GroupName +dsp
setmqaut -m QMgrName -n ** -t authinfo -g GroupName +dsp
setmqaut -m QMgrName -n ** -t listener -g GroupName +dsp
```

```
setmqaut -m QMgrName -n ** -t namelist -g GroupName +dsp
setmqaut -m QMgrName -n ** -t process -g GroupName +dsp
setmqaut -m QMgrName -n ** -t service -g GroupName +dsp
setmqaut -m QMgrName -t qmgr -g GroupName +dsp +inq +connect
```

只有在您想要使用 IBM MQ Explorer 時，才需要 SYSTEM.ADMIN.COMMAND.QUEUE 和 SYSTEM.MQEXPLORER.REPLY.MODEL 的特定權限。

- IBM i

  若為 IBM i，請發出下列指令:

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*Q) USER('GroupName') AUT(*ADMDSP *BROWSE) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*TOPIC) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*CHL) USER('GroupName') AUT(*ADMDSP *INQ) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*CLTCN) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*AUTHINFO) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*LSR) USER('GroupName') AUT(*ADMDSP)MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*NMLIST) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*PRC) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*SVC) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ('object-name') OBJTYPE(*MQM) USER('GroupName') AUT(*ADMDSP *CONNECT *INQ)
MQMNAME('QMgrName')
```

- z/OS

  若為 z/OS，請發出下列指令:

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

變數名稱具有下列意義:

QMgrName
佇列管理程式的名稱。

z/OS 在 z/OS 上，此值也可以是佇列共用群組的名稱。

GroupName
要授與存取權的群組名稱。

授與佇列管理程式上所有資源的完整管理存取權

將佇列管理程式上所有資源的完整管理存取權授與具有商業需求的每一個使用者或使用者群組。

關於這項作業

您可以使用「新增角色型權限」精靈或適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

附註: ALW

1. 如果您使用 **runmqsc** 來管理佇列管理程式，而非 IBM MQ Explorer，則必須授與權限來取得、查詢及瀏覽 SYSTEM.MQSC.REPLY.QUEUE，而且您不需要授與 SYSTEM.MQEXPLORER.REPLY.MODEL 佇列。

2. 授與使用者對佇列管理程式上所有資源的存取權時，除非使用者具有 qm.ini 檔案的讀取權，否則使用者無法執行某些指令。 這是因為非 mqm 使用者能夠讀取 qm.ini 檔案的限制。

   除非您已授與該使用者對 qm.ini 檔案的讀取權，否則該使用者無法發出下列指令:

   - 定義配置為使用 TLS 的通道
   - 使用 qm.ini 中定義的自動配置插入變數來定義通道

程序

- 如果您使用精靈，請在「IBM MQ Explorer Navigator」窗格中，用滑鼠右鍵按一下佇列管理程式，然後按一下 **物件權限 > 新增角色型權限**。

  這時會開啟「新增角色型權限」精靈。

- Linux AIX

  若為 AIX and Linux 系統，請發出下列指令:

```
setmqaut -m QMgrName -n '**' -t queue -g GroupName +alladm +browse
setmqaut -m QMgrName -n @class -t queue -g GroupName +crt
setmqaut -m QMgrName -n SYSTEM.ADMIN.COMMAND.QUEUE -t queue -g GroupName +dsp +inq +put
setmqaut -m QMgrName -n SYSTEM.MQEXPLORER.REPLY.MODEL -t queue -g GroupName +dsp +inq +get +put
setmqaut -m QMgrName -n '**' -t topic -g GroupName +alladm
setmqaut -m QMgrName -n @class -t topic -g GroupName +crt
setmqaut -m QMgrName -n '**' -t channel -g GroupName +alladm
setmqaut -m QMgrName -n @class -t channel -g GroupName +crt
setmqaut -m QMgrName -n '**' -t clntconn -g GroupName +alladm
setmqaut -m QMgrName -n @class -t clntconn -g GroupName +crt
setmqaut -m QMgrName -n '**' -t authinfo -g GroupName +alladm
setmqaut -m QMgrName -n @class -t authinfo -g GroupName +crt
setmqaut -m QMgrName -n '**' -t listener -g GroupName +alladm
setmqaut -m QMgrName -n @class -t listener -g GroupName +crt
setmqaut -m QMgrName -n '**' -t namelist -g GroupName +alladm
setmqaut -m QMgrName -n @class -t namelist -g GroupName +crt
setmqaut -m QMgrName -n '**' -t process -g GroupName +alladm
setmqaut -m QMgrName -n @class -t process -g GroupName +crt
setmqaut -m QMgrName -n '**' -t service -g GroupName +alladm
setmqaut -m QMgrName -n @class -t service -g GroupName +crt
setmqaut -m QMgrName -t qmgr -g GroupName +alladm +connect
```

  如需 @class 的相關資訊，請參閱 **setmqaut**。

- Windows

  若為 Windows 系統，請發出與 AIX and Linux 系統相同的指令，但使用設定檔名稱 @CLASS 而非 @class。

- IBM i

  若為 IBM i，請發出下列指令:

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

- z/OS

  若為 z/OS，請發出下列指令:

```
RDEFINE MQADMIN QMgrName.*.** UACC(NONE)
PERMIT QMgrName.*.** CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  變數名稱具有下列意義:

  QMgrName
  佇列管理程式的名稱。

  z/OS 在 z/OS 上，此值也可以是佇列共用群組的名稱。

  GroupName
  要授與存取權的群組名稱。

移除佇列管理程式的連線功能

如果您不想要使用者應用程式連接至佇列管理程式，請移除其連接至佇列管理程式的權限。

關於這項作業

使用適合您作業系統的指令，取消所有使用者連接佇列管理程式的權限。

在 多平台上，您也可以使用 DELETE AUTHREC 指令。

註: 在 IBM MQ Appliance 上，您只能使用 **DELETE AUTHREC** 指令。

程序

- ALW

  若為 AIX, Linux, and Windows 系統，請發出下列指令:

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

- IBM i

  若為 IBM i，請發出下列指令:

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

- z/OS

  若為 z/OS，請發出下列指令:

  ```
  RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
  RDEFINE MQCONN QMgrName.CHIN UACC(NONE)
  RDEFINE MQCONN QMgrName.CICS UACC(NONE)
  RDEFINE MQCONN QMgrName.IMS UACC(NONE)
  ```

  請勿發出任何 PERMIT 指令。

  變數名稱具有下列意義:

  QMgrName
  佇列管理程式的名稱。

  z/OS 在 z/OS 上，此值也可以是佇列共用群組的名稱。

  GroupName
  要拒絕存取的群組名稱。

容許使用者應用程式連接至佇列管理程式

您想要容許使用者應用程式連接至佇列管理程式。 請使用本主題中的表格來決定要採取的動作。

首先，判定用戶端應用程式是否將連接至佇列管理程式。

如果將連接至佇列管理程式的應用程式都不是用戶端應用程式，請停用遠端存取，如 第 325 頁的『停用佇列管理程式的遠端存取』中所述。

如果將連接至佇列管理程式的一或多個應用程式是用戶端應用程式，請依照 第 319 頁的『保護佇列管理程式的遠端連線功能』中的說明來維護遠端連線功能安全。

在這兩種情況下，請依照 第 325 頁的『設定連線安全』 中的說明來設定連線安全

如果您想要控制每一個連接至佇列管理程式之使用者的資源存取權，請參閱下表。 如果第一個直欄中的陳述式為 true ，請採取第二個直欄中列出的動作。

陳述式	採取此動作
您具有使用佇列的應用程式	請參閱 第 326 頁的『控制使用者對佇列的存取權』
您具有使用主題的應用程式	請參閱 第 331 頁的『控制使用者對主題的存取權』。
您具有查詢佇列管理程式物件的應用程式	請參閱 第 332 頁的『授與查詢佇列管理程式的權限』。

陳述式	採取此動作
您具有使用程序物件的應用程式	請參閱 第 333 頁的『授與存取處理程序的權限』
您具有使用名稱清單的應用程式	請參閱 第 333 頁的『授與存取名稱清單的權限』

保護佇列管理程式的遠端連線功能

您可以使用 TLS、安全結束程式、通道鑑別記錄或這些方法的組合，來保護佇列管理程式的遠端連線功能。

關於這項作業

您可以使用用戶端工作站上的用戶端連線通道及伺服器上的伺服器連線通道，將用戶端連接至佇列管理程式。 使用下列其中一種方式來保護這類連線的安全。

程序

1. 搭配使用 TLS 與通道鑑別記錄:
   a) 使用 SSLPEERMAP 通道鑑別記錄將所有 DN 對映至 USERSRC (NOACCESS) ， 以防止任何「識別名稱 (DN)」開啟通道。
   b) 容許使用 SSLPEERMAP 通道鑑別記錄，將特定的 DN 或 DN 集對映至 USERSRC (CHANNEL) ， 以開啟通道。
2. 搭配使用 TLS 與安全結束程式:
   a) 將伺服器連線通道上的 MCAUSER 設為沒有專用權的使用者 ID。
   b) 根據 SSLPeerNamePtr 和 SSLPeerName 長度欄位傳遞至 MQCD 結構中結束程式的 TLS DN 值，撰寫安全結束程式以指派 MCAUSER 值。
3. 將 TLS 與固定通道定義值搭配使用:
   a) 將伺服器連線通道上的 SSLPEER 設為特定值或縮小值範圍。
   b) 將伺服器連線通道上的 MCAUSER 設定為通道應該用來執行的使用者 ID。
4. 在不使用 TLS 的通道上使用通道鑑別記錄:
   a) 使用 address (*) 和 USERSRC (NOACCESS) 的位址對映通道鑑別記錄，防止任何 IP 位址開啟通道。
   b) 容許特定 IP 位址開啟通道，方法是使用具有 USERSRC (CHANNEL) 的那些位址的位址對映通道鑑別記錄。
5. 使用安全結束程式:
   a) 撰寫安全結束程式，以根據您選擇的任何內容 (例如，原始 IP 位址) 來授權連線。
6. 您也可以使用具有安全結束程式的通道鑑別記錄，或使用這三種方法 (如果您的特定情況需要的話)。

封鎖特定 *IP* 位址

您可以使用通道鑑別記錄來防止特定通道接受來自 IP 位址的入埠連線，或防止整個佇列管理程式容許從 IP 位址存取。

開始之前

執行下列指令來啟用通道鑑別記錄:

```
ALTER QMGR CHLAUTH(ENABLED)
```

關於這項作業

若要禁止特定通道接受入埠連線，並確保只有在使用正確通道名稱時才接受連線，可以使用一種類型的規則來封鎖 IP 位址。 若要禁止 IP 位址存取整個佇列管理程式，您通常會使用防火牆來永久封鎖它。 不過，另一種類型的規則可讓您暫時封鎖一些位址，例如在您等待更新防火牆時。

程序

- 若要阻止 IP 位址使用特定通道，請使用 MQSC 指令 **SET CHLAUTH** 或 PCF 指令 **Set Channel Authentication Record** 來設定通道鑑別記錄。

```
SET CHLAUTH(generic-channel-name) TYPE(ADDRESSMAP) ADDRESS(generic-ip-address)
USERSRC(NOACCESS)
```

指令有三個部分:

SET CHLAUTH (*generic-channel-name*)
您可以使用指令的這個部分來控制是否要封鎖整個佇列管理程式、單一通道或通道範圍的連線。 您在這裡放置的內容會決定涵蓋哪些區域。

例如:

- SET CHLAUTH('*') -封鎖佇列管理程式上的每個通道，即整個佇列管理程式
- SET CHLAUTH ('SYSTEM. *')-封鎖以 SYSTEM 開頭的每個通道。
- SET CHLAUTH ('SYSTEM.DEF.SVRCONN')-封鎖通道 SYSTEM.DEF.SVRCONN

CHLAUTH 規則的類型
請使用指令的這個部分來指定指令類型，並決定您要提供單一位址或位址清單。

例如:

- TYPE(ADDRESSMAP) -如果您想要提供單一位址或萬用字元位址，請使用 ADDRESSMAP。 例如，ADDRESS('192.168.*') 會封鎖來自以 192.168 開頭之 IP 位址的任何連線。

  如需使用型樣過濾 IP 位址的相關資訊，請參閱 一般 IP 位址。
- TYPE(BLOCKADDR) -如果您想要提供要封鎖的位址清單，請使用 BLOCKADDR。

其他參數
這些參數視您在指令第二部分中使用的規則類型而定:

- 對於 TYPE(ADDRESSMAP)，您使用 ADDRESS
- 對於 TYPE(BLOCKADDR)，您使用 ADDRLIST

相關參考
SET CHLAUTH

如果佇列管理程式不在執行中，則暫時封鎖特定的 *IP* 位址
當佇列管理程式不在執行中且因此無法發出 MQSC 指令時，您可能想要封鎖特定 IP 位址或位址範圍。 您可以透過修改 blockaddr.ini 檔案，在異常情況下暫時封鎖 IP 位址。

關於這項作業

blockaddr.ini 檔案包含佇列管理程式使用的 BLOCKADDR 定義副本。 如果接聽器在佇列管理程式之前啟動，則接聽器會讀取此檔案。 在這些情況下，接聽器會使用您手動新增至 blockaddr.ini 檔案的任何值。

不過，請注意，當佇列管理程式啟動時，它會將一組 BLOCKADDR 定義寫入 blockaddr.ini 檔案，並改寫您可能已完成的任何手動編輯。 同樣地，每次您使用 **SET CHLAUTH** 指令新增或刪除 BLOCKADDR 定義時，都會更新 blockaddr.ini 檔案。 因此，只有在佇列管理程式執行時，您才能使用 **SET CHLAUTH** 指令對 BLOCKADDR 定義進行永久變更。

程序

1. 在文字編輯器中開啟 blockaddr.ini 檔案。
   該檔案位於佇列管理程式的資料目錄中。
2. 新增 IP 位址作為簡式關鍵字-值配對，其中關鍵字是 Addr。
   如需使用型樣過濾 IP 位址的相關資訊，請參閱 一般 IP 位址。
   例如:

```
Addr = 192.0.2.0
Addr = 192.0.*
Addr = 192.0.2.1-8
```

相關工作

第 319 頁的『封鎖特定 IP 位址』

您可以使用通道鑑別記錄來防止特定通道接受來自 IP 位址的入埠連線，或防止整個佇列管理程式容許從 IP 位址存取。

相關參考

SET CHLAUTH

封鎖特定使用者 *ID*

您可以指定使用者 ID (如果主張的話，則會導致通道結束)，以防止特定使用者使用通道。 透過設定通道鑑別記錄來執行此動作。

開始之前

請確定已啟用通道鑑別記錄，如下所示:

```
ALTER QMGR CHLAUTH(ENABLED)
```

程序

使用 MQSC 指令 **SET CHLAUTH** 或 PCF 指令 **Set Channel Authentication Record** 來設定通道鑑別記錄。 例如，您可以發出 MQSC 指令:

```
SET CHLAUTH(' generic-channel-name ') TYPE(BLOCKUSER) USERLIST(userID1, userID2)
```

generic-channel-name 是您要控制存取權的通道名稱，或包含星號 (*) 符號作為萬用字元且符合通道名稱的型樣。

在「TYPE(BLOCKUSER)」上提供的使用者清單僅適用於 SVRCONN 通道，不適用於佇列管理程式至佇列管理程式通道。

userID1 和 *userID2* 都是要防止使用通道的使用者 ID。 您也可以指定特殊值 *MQADMIN，以參照特許管理使用者。 如需特許使用者的相關資訊，請參閱 第 279 頁的『特許使用者』。 如需 *MQADMIN 的相關資訊，請參閱 SET CHLAUTH。

相關參考

SET CHLAUTH

將遠端佇列管理程式對映至 *MCAUSER* 使用者 *ID*

您可以根據通道所連接的佇列管理程式，使用通道鑑別記錄來設定通道的 MCAUSER 屬性。

開始之前

請確定已啟用通道鑑別記錄，如下所示:

```
ALTER QMGR CHLAUTH(ENABLED)
```

關於這項作業

您可以選擇性地限制套用規則的 IP 位址。

請注意，此技術不適用於伺服器連線通道。 如果您在下列指令中指定伺服器連線通道的名稱，則它沒有作用。

程序

- 使用 MQSC 指令 **SET CHLAUTH** 或 PCF 指令 **Set Channel Authentication Record** 來設定通道鑑別記錄。 例如，您可以發出 MQSC 指令:

```
SET CHLAUTH(' generic-channel-name ') TYPE (QMGRMAP) QMNAME(generic-partner-qmgr-name
) USERSRC(MAP) MCAUSER(user)
```

generic-channel-name 是您要控制存取權的通道名稱，或包含星號 (*) 符號作為萬用字元且符合通道名稱的型樣。

generic-partner-qmgr-name 是佇列管理程式的名稱，或包含星號 (*) 符號作為萬用字元且符合佇列管理程式名稱的型樣。

user 是用於來自指定佇列管理程式的所有連線的使用者 ID。

- 若要將此指令限制為特定 IP 位址，請包括 **ADDRESS** 參數，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 是您要控制存取權的通道名稱，或包含星號 (*) 符號作為萬用字元且符合通道名稱的型樣。

generic-ip-address 是單一位址，或包含星號 (*) 符號作為萬用字元或連字號 (-) 以指出符合位址的範圍的型樣。 如需一般 IP 位址的相關資訊，請參閱 一般 IP 位址。

相關參考

SET CHLAUTH

將用戶端使用者 *ID* 對映至 *MCAUSER* 使用者 *ID*

您可以根據從用戶端收到的使用者 ID，使用通道鑑別記錄來變更伺服器連線通道的 MCAUSER 屬性。

開始之前

請確定已啟用通道鑑別記錄，如下所示:

```
ALTER QMGR CHLAUTH(ENABLED)
```

關於這項作業

請注意，此技術僅適用於伺服器連線通道。 它對其他通道類型沒有影響。

程序

使用 MQSC 指令 **SET CHLAUTH** 或 PCF 指令 **Set Channel Authentication Record** 設定通道鑑別記錄。 例如，您可以發出 MQSC 指令:

```
SET CHLAUTH(' generic-channel-name ') TYPE (USERMAP) CLNTUSER(client-user-name) USERSRC(MAP)
MCAUSER(
user)
```

generic-channel-name 是您要控制存取權的通道名稱，或包含星號 (*) 符號作為萬用字元且符合通道名稱的型樣。

client-user-name 是與用戶端連線相關聯的使用者 ID，該值可以由用戶端應用程式主張、使用早期採用或透過通道結束程式設定的連線鑑別變更。

user 是要使用的使用者 ID，而不是用戶端使用者名稱。

相關參考

SET CHLAUTH

通道節的屬性 (ChlauthEarlyAdopt)

將 *SSL* 或 *TLS* 識別名稱對映至 *MCAUSER* 使用者 *ID*

您可以根據收到的「識別名稱 (DN)」，使用通道鑑別記錄來設定通道的 MCAUSER 屬性。

開始之前

請確定已啟用通道鑑別記錄，如下所示:

```
ALTER QMGR CHLAUTH(ENABLED)
```

程序

使用 MQSC 指令 **SET CHLAUTH** 或 PCF 指令 **Set Channel Authentication Record** 來設定通道鑑別記錄。 例如，您可以發出 MQSC 指令:

```
SET CHLAUTH('generic-channel-name') TYPE (SSLPEERMAP)
SSLPEER(generic-ssl-peer-name) SSLCERTI(generic-issuer-name)
USERSRC(MAP) MCAUSER(user)
```

generic-channel-name 是您要控制存取權的通道名稱，或包含星號 (*) 符號作為萬用字元且符合通道名稱的型樣。

generic-ssl-peer-name 是一個字串，遵循 SSLPEER 值的標準 IBM MQ 規則。 請參閱 SSLPEER 值的 IBM MQ 規則。

user 是要用於所有使用指定 DN 之連線的使用者 ID。

generic-issuer-name 是指要符合之憑證的「發證者 DN」。 此參數是選用的，但您應該使用它，以避免在使用多個憑證管理中心時，瘋狂地比對錯誤的憑證。

相關參考

SET CHLAUTH

封鎖從遠端佇列管理程式存取

您可以使用通道鑑別記錄來防止遠端佇列管理程式啟動通道。

開始之前

請確定已啟用通道鑑別記錄，如下所示:

```
ALTER QMGR CHLAUTH(ENABLED)
```

關於這項作業

請注意，此技術不適用於伺服器連線通道。 如果您在下列指令中指定伺服器連線通道的名稱，則它沒有作用。

程序

使用 MQSC 指令 **SET CHLAUTH** 或 PCF 指令 **Set Channel Authentication Record** 來設定通道鑑別記錄。 例如，您可以發出 MQSC 指令:

```
SET CHLAUTH(' generic-channel-name ') TYPE(QMGRMAP) QMNAME(' generic-partner-qmgr-name ')
USERSRC(NOACCESS)
```

generic-channel-name 是您要控制存取權的通道名稱，或包含星號 (*) 符號作為萬用字元且符合通道名稱的型樣。

generic-partner-qmgr-name 是佇列管理程式的名稱，或包含星號 (*) 符號作為萬用字元且符合佇列管理程式名稱的型樣。

相關參考

SET CHLAUTH

封鎖存取用戶端使用者 *ID*

您可以使用通道鑑別記錄來防止用戶端使用者 ID 建立通道連線。

開始之前

請確定已啟用通道鑑別記錄，如下所示:

```
ALTER QMGR CHLAUTH(ENABLED)
```

關於這項作業
請注意，此技術僅適用於伺服器連線通道。 它對其他通道類型沒有影響。

程序
使用 MQSC 指令 **SET CHLAUTH** 或 PCF 指令 **Set Channel Authentication Record** 來設定通道鑑別記錄。 例如，您可以發出 MQSC 指令:

```
SET CHLAUTH(' generic-channel-name ') TYPE(USERMAP) CLNTUSER(' client-user-name ')
USERSRC(NOACCESS)
```

generic-channel-name 是您要控制存取權的通道名稱，或包含星號 (*) 符號作為萬用字元且符合通道名稱的型樣。

client-user-name 是與用戶端連線相關聯的使用者 ID，該值可以由用戶端應用程式主張、使用早期採用或透過通道結束程式設定的連線鑑別變更。

相關參考

SET CHLAUTH

封鎖存取 *SSL* 或 *TLS* 識別名稱
您可以使用通道鑑別記錄來防止 TLS「識別名稱 (DN)」啟動通道。

開始之前
請確定已啟用通道鑑別記錄，如下所示:

```
ALTER QMGR CHLAUTH(ENABLED)
```

程序
使用 MQSC 指令 **SET CHLAUTH** 或 PCF 指令 **Set Channel Authentication Record** 來設定通道鑑別記錄。 例如，您可以發出 MQSC 指令:

```
SET CHLAUTH('generic-channel-name') TYPE(SSLPEERMAP)
SSLPEER('generic-ssl-peer-name') SSLCERTI(generic-issuer-name)
USERSRC(NOACCESS)
```

generic-channel-name 是您要控制存取權的通道名稱，或包含星號 (*) 符號作為萬用字元且符合通道名稱的型樣。

generic-ssl-peer-name 是一個字串，遵循 SSLPEER 值的標準 IBM MQ 規則。 請參閱 SSLPEER 值的 IBM MQ 規則。

generic-issuer-name 是指要符合之憑證的「發證者 DN」。 此參數是選用的，但您應該使用它，以避免在使用多個憑證管理中心時，瘋狂地比對錯誤的憑證。

相關參考

SET CHLAUTH

將 *IP* 位址對映至 *MCAUSER* 使用者 *ID*
您可以根據接收連線的 IP 位址，使用通道鑑別記錄來設定通道的 MCAUSER 屬性。

開始之前
請確定已啟用通道鑑別記錄，如下所示:

```
ALTER QMGR CHLAUTH(ENABLED)
```

程序
使用 MQSC 指令 **SET CHLAUTH** 或 PCF 指令 **Set Channel Authentication Record** 來設定通道鑑別記錄。 例如，您可以發出 MQSC 指令:

```
SET CHLAUTH(' generic-channel-name ') TYPE(ADDRESSMAP) ADDRESS(' generic-ip-address ')
USERSRC(MAP) MCAUSER(user)
```

generic-channel-name 是您要控制存取權的通道名稱，或包含星號 (*) 符號作為萬用字元且符合通道名稱的型樣。

user 是要用於所有使用指定 DN 之連線的使用者 ID。

generic-ip-address 是從中建立連線的位址，或包含星號 (*) 作為萬用字元或連字號 (-) 以指出符合位址的範圍的型樣。

相關參考

SET CHLAUTH

停用佇列管理程式的遠端存取

如果您不想要用戶端應用程式連接至佇列管理程式，請停用它的遠端存取。

關於這項作業

以下列其中一種方式阻止用戶端應用程式連接至佇列管理程式:

程序

- 使用 MQSC 指令 **DELETE CHANNEL** 刪除所有伺服器連線通道。
- 使用 MQSC 指令 **ALTER CHANNEL**，將通道的訊息通道代理程式使用者 ID (MCAUSER) 設為沒有存取權的使用者 ID。

設定連線安全

將連接至佇列管理程式的權限授與每一個使用者或具有商業需要的使用者群組。

關於這項作業

若要設定連線安全，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- ALW
  在 AIX, Linux, and Windows 上:
  ```
  setmqaut -m QMgrName -t qmgr -g GroupName +connect
  ```
- IBM i
  在 IBM i 上:
  ```
  GRTMQMAUT OBJ('QMgrName') OBJTYPE(*MQM) USER('GroupName') AUT(*CONNECT)
  ```
- z/OS
  在 z/OS 上:
  ```
  RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
  PERMIT QMgrName.BATCH CLASS(MQCONN) ID(GroupName) ACCESS(READ)
  RDEFINE MQCONN QMgrName.CICS UACC(NONE)
  PERMIT QMgrName.CICS CLASS(MQCONN) ID(GroupName) ACCESS(READ)
  RDEFINE MQCONN QMgrName.IMS UACC(NONE)
  PERMIT QMgrName.IMS CLASS(MQCONN) ID(GroupName) ACCESS(READ)
  RDEFINE MQCONN QMgrName.CHIN UACC(NONE)
  PERMIT QMgrName.CHIN CLASS(MQCONN) ID(GroupName) ACCESS(READ)
  ```

這些指令提供連接批次、 CICS、 IMS 及通道起始程式 (CHIN) 的權限。 如果您不使用特定類型的連線，請省略相關指令。

變數名稱具有下列意義:

QMgrName
佇列管理程式的名稱。 在 z/OS 上，此值也可以是佇列共用群組的名稱。

ObjectProfile
要變更其授權的物件或通用設定檔名稱。

GroupName
要授與存取權的群組名稱。

相關概念

第 168 頁的『通道起始程式的連線安全設定檔』
用於檢查來自通道起始程式之連線的設定檔由佇列管理程式或佇列共用群組名稱後面接著單字 *CHIN* 組成。 將連線設定檔的 READ 存取權提供給通道起始程式作業位址空間所使用的使用者 ID。

控制使用者對佇列的存取權

您想要控制應用程式對佇列的存取權。 請利用這個主題來決定要採取哪些動作。

針對第一個直欄中的每一個 true 陳述式，採取第二個直欄中指出的動作。

陳述式	動作
應用程式從佇列取得訊息	請參閱 第 326 頁的『授與從佇列取得訊息的權限』
應用程式集環境定義	請參閱 第 327 頁的『授與權限以設定環境定義』
應用程式傳遞環境定義	請參閱 第 328 頁的『授與權限以傳遞環境定義』
應用程式將訊息放置在叢集佇列上	請參閱 第 399 頁的『授權將訊息放置在遠端叢集佇列上』
應用程式將訊息放置在本端佇列上	請參閱 第 328 頁的『授與將訊息放入本端佇列的權限』
應用程式將訊息放置在模型佇列上	請參閱 第 329 頁的『授與將訊息放入模型佇列的權限』
應用程式將訊息放置在遠端佇列上	請參閱 第 330 頁的『授與將訊息放入遠端叢集佇列的權限』

授與從佇列取得訊息的權限
將從佇列或佇列集取得訊息的權限授與每一個具有商業需要的使用者群組。

關於這項作業

若要授與從部分佇列取得訊息的權限，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- 若為 AIX, Linux, and Windows 系統，請發出下列指令:

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

- 若為 IBM i，請發出下列指令:

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

- 若為 z/OS，請發出下列指令:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(UPDATE)
```

變數名稱具有下列意義:

QMgrName
佇列管理程式的名稱。 在 z/OS 上，此值也可以是佇列共用群組的名稱。

ObjectProfile
要變更其授權的物件或通用設定檔名稱。

GroupName
要授與存取權的群組名稱。

授與權限以設定環境定義
授與權限以將所放置訊息上的環境定義設定給每一個具有商業需求的使用者群組。

關於這項作業

若要授與在部分佇列上設定環境定義的權限，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- 若為 AIX, Linux, and Windows 系統，請發出下列其中一個指令:
  - 若要僅設定身分環境定義，請執行下列動作:

    ```
    setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +setid
    ```
  - 若要設定所有環境定義，請執行下列動作:

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

  註: 若要使用 `setid` 或 `setall` 權限，必須同時對適當的佇列物件及佇列管理程式物件授與授權。
- 若為 IBM i，請發出下列其中一個指令:
  - 若要僅設定身分環境定義，請執行下列動作:

    ```
    GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*SETID) MQMNAME('
    QMgrName ')
    ```
  - 若要設定所有環境定義，請執行下列動作:

    ```
    GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*SETALL) MQMNAME('
    QMgrName ')
    ```
- 若為 z/OS，請發出下列其中一組指令:
  - 若要僅設定身分環境定義，請執行下列動作:

    ```
    RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
    PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
    ```
  - 若要設定所有環境定義，請執行下列動作:

    ```
    RDEFINE MQQUEUE QMgrName. ObjectProfile UACC(NONE)
    PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(CONTROL)
    ```

  變數名稱具有下列意義:

  QMgrName
  佇列管理程式的名稱。 在 z/OS 上，此值也可以是佇列共用群組的名稱。

ObjectProfile
要變更其授權的物件或通用設定檔名稱。

GroupName
要授與存取權的群組名稱。

授與權限以傳遞環境定義
授與權限將環境定義從擷取的訊息傳遞至所放置的訊息，以及傳遞至具有商業需求的每一個使用者群組。

關於這項作業

若要授與在部分佇列上傳遞環境定義的權限，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- ALW
  若為 AIX, Linux, and Windows 系統，請發出下列其中一個指令:
  - 若要僅傳遞身分環境定義，請執行下列動作:

    ```
    setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +passid
    ```
  - 若要傳遞所有環境定義，請執行下列動作:

    ```
    setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +passall
    ```
- IBM i
  若為 IBM i，請發出下列其中一個指令:
  - 若要僅傳遞身分環境定義，請執行下列動作:

    ```
    GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PASSID) MQMNAME('
    QMgrName ')
    ```
  - 若要傳遞所有環境定義，請執行下列動作:

    ```
    GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PASSALL) MQMNAME('
    QMgrName ')
    ```
- z/OS
  對於 z/OS，請發出下列指令來傳遞身分環境定義或所有環境定義:

  ```
  RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
  ```

  變數名稱具有下列意義:

  QMgrName
  佇列管理程式的名稱。 在 z/OS 上，此值也可以是佇列共用群組的名稱。

  ObjectProfile
  要變更其授權的物件或通用設定檔名稱。

  GroupName
  要授與存取權的群組名稱。

授與將訊息放入本端佇列的權限
授與權限，以將訊息放入本端佇列或佇列集，授與具有商業需要的每一個使用者群組。

關於這項作業

若要授與將訊息放置到某些本端佇列的權限，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- 若為 AIX, Linux, and Windows 系統，請發出下列指令:

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

- 若為 IBM i，請發出下列指令:

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

- 若為 z/OS，請發出下列指令:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

  變數名稱具有下列意義:

  QMgrName
  佇列管理程式的名稱。 在 z/OS 上，此值也可以是佇列共用群組的名稱。

  ObjectProfile
  要變更其授權的物件或通用設定檔名稱。

  GroupName
  要授與存取權的群組名稱。

授與將訊息放入模型佇列的權限
將將訊息放置到模型佇列或模型佇列集的權限授與具有商業需求的每一個使用者群組。

關於這項作業

模型佇列用來建立動態佇列。 因此，您必須同時授與模型及動態佇列的權限。 若要授與這些權限，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- 若為 AIX, Linux, and Windows 系統，請發出下列指令:

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

- 若為 IBM i，請發出下列指令:

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

- 若為 z/OS，請發出下列指令:

```
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)
```

  變數名稱具有下列意義:

QMgrName
佇列管理程式的名稱。 在 z/OS 上，此值也可以是佇列共用群組的名稱。

ModelQueueName
動態佇列所根據的模型佇列名稱。

ObjectProfile
要變更其授權的動態佇列或通用設定檔名稱。

GroupName
要授與存取權的群組名稱。

授與將訊息放入遠端叢集佇列的權限
將將訊息放置到遠端叢集佇列或佇列集的權限授與具有商業需求的每一個使用者群組。

關於這項作業

若要將訊息放置在遠端叢集佇列上，您可以將它放置在遠端佇列的本端定義或完整遠端佇列上。 如果您使用遠端佇列的本端定義，則需要權限來放置至本端物件: 請參閱 第 328 頁的『授與將訊息放入本端佇列的權限』。 如果您使用完整的遠端佇列，則需要權限才能放入遠端佇列。 請使用適合您作業系統的指令來授與此權限。

預設行為是對 SYSTEM.CLUSTER.TRANSMIT.QUEUE 執行存取控制。 請注意，即使您使用多個傳輸佇列，也會套用此行為。

只有在您依照 安全段落 主題中的說明，將 qm.ini 檔中的 **ClusterQueueAccessControl** 屬性配置成 *RQMName*，並重新啟動佇列管理程式之後，這個主題中所說明的特定行為才適用。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- 若為 AIX, Linux, and Windows 系統，請發出下列指令:

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

  請注意，您只能對遠端叢集佇列使用 *rqmname* 物件。
- 若為 IBM i，請發出下列指令:

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

  請注意，您只能將 RMTMQMNAME 物件用於遠端叢集佇列。
- 若為 z/OS，請發出下列指令:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

  請注意，您只能對遠端叢集佇列使用遠端佇列管理程式 (或佇列共用群組) 的名稱。

  變數名稱具有下列意義:

  QMgrName
  佇列管理程式的名稱。 在 z/OS 上，此值也可以是佇列共用群組的名稱。

  ObjectProfile
  要變更其授權的遠端佇列管理程式或通用設定檔的名稱。

  GroupName
  要授與存取權的群組名稱。

控制使用者對主題的存取權

您需要控制應用程式對主題的存取權。 請利用這個主題來決定要採取哪些動作。

針對第一個直欄中的每一個 true 陳述式，採取第二個直欄中指出的動作。

表 *71:* 控制使用者對主題的存取權

陳述式	**動作**
應用程式會將訊息發佈至主題	請參閱 第 331 頁的『授與將訊息發佈至主題的權限』
應用程式訂閱主題	請參閱 第 331 頁的『授與訂閱主題的權限』

授與將訊息發佈至主題的權限

將訊息發佈至主題或主題集的權限授與具有商業需求的每一個使用者群組。

關於這項作業

若要授與將訊息發佈至部分主題的權限，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- 若為 AIX, Linux, and Windows 系統，請發出下列指令:

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

- 若為 IBM i，請發出下列指令:

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

- 若為 z/OS，請發出下列指令:

  ```
  RDEFINE MQTOPIC QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQTOPIC) ID(GroupName) ACCESS(UPDATE)
  ```

  變數名稱具有下列意義:

  QMgrName
  佇列管理程式的名稱。 在 z/OS 上，此值也可以是佇列共用群組的名稱。

  ObjectProfile
  要變更其授權的物件或通用設定檔名稱。

  GroupName
  要授與存取權的群組名稱。

授與訂閱主題的權限

將訂閱主題或一組主題的權限授與每一個具有商業需求的使用者群組。

關於這項作業

若要授與訂閱部分主題的權限，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- 若為 AIX, Linux, and Windows 系統，請發出下列指令:

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

- 若為 IBM i，請發出下列指令:

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

- 若為 z/OS，請發出下列指令:

```
RDEFINE MQTOPIC QMgrName.SUBSCRIBE.ObjectProfile UACC(NONE)
PERMIT QMgrName.SUBSCRIBE.ObjectProfile CLASS(MQTOPIC) ID(GroupName) ACCESS(UPDATE)
```

變數名稱具有下列意義:

QMgrName
佇列管理程式的名稱。 在 z/OS 上，此值也可以是佇列共用群組的名稱。

ObjectProfile
要變更其授權的物件或通用設定檔名稱。

GroupName
要授與存取權的群組名稱。

授與查詢佇列管理程式的權限

將查詢佇列管理程式的權限授與每一個具有商業需求的使用者群組。

關於這項作業

若要授與查詢佇列管理程式的權限，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- 若為 AIX, Linux, and Windows 系統，請發出下列指令:

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

- 若為 IBM i，請發出下列指令:

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

- 若為 z/OS，請發出下列指令:

```
RDEFINE MQCMDS QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

這些指令會授與指定佇列管理程式的存取權。 若要允許使用者使用 MQINQ 指令，請發出下列指令:

```
RDEFINE MQCMDS QMgrName.MQINQ.QMGR UACC(NONE)
PERMIT QMgrName.MQINQ.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

變數名稱具有下列意義:

QMgrName
佇列管理程式的名稱。 在 z/OS 上，此值也可以是佇列共用群組的名稱。

ObjectProfile
要變更其授權的物件或通用設定檔名稱。

GroupName
要授與存取權的群組名稱。

授與存取處理程序的權限

將存取程序或程序集的權限授與具有商業需求的每一個使用者群組。

關於這項作業

若要授與存取部分處理程序的權限，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- 若為 AIX, Linux, and Windows 系統，請發出下列指令:

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

- 若為 IBM i，請發出下列指令:

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

- 若為 z/OS，請發出下列指令:

```
RDEFINE MQPROC QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQPROC) ID(GroupName) ACCESS(READ)
```

變數名稱具有下列意義:

QMgrName
佇列管理程式的名稱。 在 z/OS 上，此值也可以是佇列共用群組的名稱。

ObjectProfile
要變更其授權的物件或通用設定檔名稱。

GroupName
要授與存取權的群組名稱。

授與存取名稱清單的權限

將存取名稱清單或一組名稱清單的權限授與每一個有商業需要的使用者群組。

關於這項作業

若要授與存取部分名稱清單的權限，請使用適合您作業系統的指令。

在 Multiplatforms 平台上，您也可以使用 SET AUTHREC 指令。

註: MQ Appliance 在 IBM MQ Appliance 上，您只能使用 **SET AUTHREC** 指令。

程序

- 若為 AIX, Linux, and Windows 系統，請發出下列指令:

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

- 若為 IBM i，請發出下列指令:

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

- 若為 z/OS，請發出下列指令:

```
RDEFINE MQNLIST
QMgrName.ObjectProfile UACC(NONE)
```

```
PERMIT QMgrName.ObjectProfile
CLASS(MQNLIST) ID(GroupName) ACCESS(READ)
```

變數名稱具有下列意義:

QMgrName
佇列管理程式的名稱。 在 z/OS 上，此值也可以是佇列共用群組的名稱。

ObjectProfile
要變更其授權的物件或通用設定檔名稱。

GroupName
要授與存取權的群組名稱。

ALW 在 AIX, Linux, and Windows 上管理 IBM MQ 的權限

IBM MQ 管理者可以使用所有 IBM MQ 指令，並將權限授與其他使用者。 當管理者向遠端佇列管理程式發出指令時，他們必須具有遠端佇列管理程式的必要權限。 進一步考量適用於 Windows 系統。

IBM MQ 管理者有權使用所有 IBM MQ 指令 (包括授與其他使用者 IBM MQ 權限的指令)。

若要成為 IBM MQ 管理者，您必須是稱為 **mqm** 群組之特殊群組的成員。

Windows 或者，僅在 Windows 上，如果本端帳戶是 Windows 系統上「管理者」群組的成員，則可以管理 IBM MQ 。

小心: 您可以使用管理者指令，將 Azure AD 使用者新增至 mqm 群組。 例如，使用指令 `net localgroup mqm AzureAD\<your userID> /add`。 然後執行 IBM MQ 管理指令或使用 IBM MQ Explorer。

安裝 IBM MQ 時會自動建立 **mqm** 群組。 您可以將進一步使用者新增至群組，以容許他們執行管理。 此群組的所有成員都有權存取所有資源。 只有從 **mqm** 群組中移除使用者並發出 **REFRESH SECURITY** 指令，才能撤銷此存取權。

管理者可以使用控制指令來管理 IBM MQ。 其中一個控制指令是 **setmqaut**，用來授與權限給其他使用者，讓他們能夠存取或控制 IBM MQ 資源。 用於管理權限記錄的 PCF 指令可供對佇列管理程式授與 dsp 及 chg 權限的非管理者使用。 如需使用 PCF 指令管理權限的相關資訊，請參閱 可程式指令格式。

管理者必須具有遠端佇列管理程式處理 MQSC 指令所需的權限。 IBM MQ Explorer 會發出 PCF 指令來執行管理作業。 管理者不需要其他權限，即可使用「IBM MQ Explorer」來管理本端系統上的佇列管理程式。 當使用「IBM MQ Explorer」來管理另一個系統上的佇列管理程式時，管理者必須具有遠端佇列管理程式處理 PCF 指令所需的權限。

小心: 從 IBM MQ 8.0 開始，您不需要是管理者，即可使用發出 IBM MQ Script (MQSC) 指令的控制指令 **runmqsc**。

以間接模式使用 **runmqsc** 將 MQSC 指令傳送至遠端佇列管理程式時，每一個 MQSC 指令都會封裝在 Escape PCF 指令內。

如需處理 PCF 及 MQSC 指令時授權檢查的相關資訊，請參閱下列主題:

- 對於在佇列管理程式、佇列、處理程序、名稱清單及鑑別資訊物件上運作的 PCF 指令，請參閱 使用 IBM MQ 物件的權限。 請參閱本節，以取得封裝在 Escape PCF 指令內的對等 MQSC 指令。
- 如需在通道、通道起始程式、接聽器及叢集上運作的 PCF 指令，請參閱 通道安全。
- 如需對權限記錄進行操作的 PCF 指令，請參閱 PCF 指令的權限檢查
- z/OS 如需 IBM MQ for z/OS 上的指令伺服器所處理的 MQSC 指令，請參閱 z/OS 上的指令安全及指令資源安全 。

此外，在 Windows 系統上， SYSTEM 帳戶對 IBM MQ 資源具有完整存取權。

在 AIX and Linux 平台上，也會建立特殊使用者 ID **mqm** ，僅供產品使用。 它絕不能供非特許使用者使用。 所有 IBM MQ 物件都由使用者 ID **mqm** 所擁有。

在 Windows 系統上， Administrators 群組的成員也可以管理任何佇列管理程式，就像 SYSTEM 帳戶一樣。 您也可以在網域控制站上建立網域 **mqm** 群組，其中包含網域內作用中的所有特許使用者 ID ，並將它新增至

本端 **mqm** 群組。 部分指令 (例如 **crtmqm**) 會操作 IBM MQ 物件的權限，因此需要權限才能使用這些物件 (如下列各節中所述)。 **mqm** 群組的成員有權使用所有物件，但如果您具有相同名稱的本端使用者及網域鑑別使用者，則在 Windows 系統上可能會拒絕權限。 這說明於第 337 頁的『AIX, Linux, and Windows 上的主體及群組』。

具有「使用者帳戶控制 (UAC)」特性的 Windows 版本會限制使用者可以在特定作業系統機能上執行的動作，即使他們是 Administrators 群組的成員也一樣。 如果您的使用者 ID 是在 Administrators 群組而非 **mqm** 群組中，則必須使用提升的命令提示字元來發出 IBM MQ admin 指令，例如 **crtmqm**，否則會產生錯誤 AMQ7077： **您未獲授權執行所要求的作業** 。 若要開啟提升的命令提示字元，請在命令提示字元的開始功能表項目或圖示上按一下滑鼠右鍵，然後選取 **以管理者身分執行**。

您不需要是 **mqm** 群組的成員，即可採取下列動作:

- 從應用程式發出 PCF 指令或 Escape PCF 指令內的 MQSC 指令，除非指令操作通道起始程式。 (這些指令在第 93 頁的『保護通道起始程式定義』中有說明)。
- 從應用程式發出 MQI 呼叫 (除非您想要在 MQCONNX 呼叫中使用捷徑連結)。
- 使用 crtmqcvx 指令來建立程式碼片段，以對資料類型結構執行資料轉換。
- 使用 dspmq 指令來顯示佇列管理程式。
- 使用 dspmqtrc 指令來顯示 IBM MQ 格式化追蹤輸出。

12 個字元的限制同時適用於群組及使用者 ID。

UNIX and Linux 平台通常會將使用者 ID 的長度限制為 12 個字元。 AIX 5.3 已提高此限制，但 IBM MQ 在所有 UNIX and Linux 平台上仍繼續遵守 12 個字元的限制。 如果您使用大於 12 個字元的使用者 ID， IBM MQ 會將它取代為值 UNKNOWN。 請勿定義值為 UNKNOWN 的使用者 ID。

ALW 在 AIX, Linux, and Windows 上管理 mqm 群組

mqm 群組中的使用者已獲授與 IBM MQ 的完整管理專用權。 因此，您不應在 mqm 群組中登記應用程式及一般使用者。 mqm 群組應該只包含 IBM MQ 管理者的帳戶。

這些作業說明如下:

- Windows 在 Windows 上建立及管理群組
- AIX 在 AIX 上建立及管理群組
- Linux 在 Linux 上建立及管理群組

Windows 如果網域控制站在 Windows 2000 或 Windows 2003 或更新版本上執行，則網域管理者可能必須設定特殊帳戶以供 IBM MQ 使用。 如需相關資訊，請參閱 使用 Prepare IBM MQ Wizard 來配置 IBM MQ 及 建立及設定 IBM MQ 的 Windows 網域帳戶。

ALW 在 AIX, Linux, and Windows 上使用 IBM MQ 物件的權限

所有物件都受到 IBM MQ 保護，且主體必須獲得適當的權限才能存取它們。 不同的主體需要對不同物件的不同存取權。

佇列管理程式、佇列、程序定義、名稱清單、通道、用戶端連線通道、接聽器、服務及鑑別資訊物件都可以從使用 MQI 呼叫或 PCF 指令的應用程式存取。 這些資源都受到 IBM MQ 保護，應用程式需要獲得許可權才能存取它們。 提出要求的實體可能是使用者、發出 MQI 呼叫的應用程式，或發出 PCF 指令的管理程式。 要求者的 ID 稱為 主體。

可以將相同物件的不同類型存取權授與不同的主體群組。 例如，對於特定佇列，可能容許一個群組同時執行放置及取得作業; 可能只容許另一個群組瀏覽佇列 (具有瀏覽選項的 MQGET)。 同樣地，部分群組可能具有佇列的放置及取得權限，但不容許變更佇列的屬性或刪除它。

部分作業特別敏感，且應該限制為特許使用者。 例如:

- 存取部分特殊佇列，例如傳輸佇列或指令佇列 SYSTEM.ADMIN.COMMAND.QUEUE
- 執行使用完整 MQI 環境定義選項的程式

- 建立及刪除應用程式佇列

物件的完整存取權會自動提供給建立物件的使用者 ID 及 mqm 群組的所有成員 (以及 Windows 系統上本端 Administrators 群組的成員)。

相關概念

第 334 頁的『在 AIX, Linux, and Windows 上管理 IBM MQ 的權限』

IBM MQ 管理者可以使用所有 IBM MQ 指令，並將權限授與其他使用者。 當管理者向遠端佇列管理程式發出指令時，他們必須具有遠端佇列管理程式的必要權限。 進一步考量適用於 Windows 系統。

ALW 在 AIX, Linux, and Windows 上進行安全檢查時

安全檢查通常是在連接至佇列管理程式、開啟或關閉物件，以及放置或取得訊息時進行。

對一般應用程式進行的安全檢查如下:

連接至佇列管理程式 (MQCONN 或 MQCONNX 呼叫)

這是應用程式第一次與特定佇列管理程式相關聯。 佇列管理程式會詢問作業環境，以探索與應用程式相關聯的使用者 ID。 然後， IBM MQ 會驗證使用者 ID 是否已獲授權連接至佇列管理程式，並保留使用者 ID 以供未來檢查。

使用者不需要登入 IBM MQ; IBM MQ 會假設使用者已登入基礎作業系統，且已經過該鑑別。

開啟物件 (MQOPEN 或 MQPUT1 呼叫)

透過開啟物件並對其發出指令來存取 IBM MQ 物件。 所有資源檢查都是在開啟物件時執行，而不是在實際存取物件時執行。 這表示 **MQOPEN** 要求必須指定所需的存取類型 (例如，使用者是否只想要瀏覽物件或執行更新，例如將訊息放入佇列)。

IBM MQ 會檢查 **MQOPEN** 要求中指定的資源。 對於別名或遠端佇列物件，所使用的授權是物件本身的授權，而不是別名或遠端佇列所解析的佇列。 這表示使用者不需要許可權即可存取它。 將建立佇列的權限限制為特許使用者。 如果您不這麼做，使用者只要建立別名，就可以略過一般存取控制。 如果明確使用佇列及佇列管理程式名稱來參照遠端佇列，則會檢查與遠端佇列管理程式相關聯的傳輸佇列。

動態佇列的權限是根據其衍生來源模型佇列的權限，但不一定相同。 這在附註 第 109 頁的『1』 中說明。

佇列管理程式用於存取權檢查的使用者 ID 是從連接至佇列管理程式之應用程式的作業環境中取得的使用者 ID。 適當授權的應用程式可以發出 **MQOPEN** 呼叫，並指定替代使用者 ID; 然後會對替代使用者 ID 進行存取控制檢查。 這不會變更與應用程式相關聯的使用者 ID ，只會用於存取控制檢查。

放置及取得訊息 (MQPUT 或 MQGET 呼叫)

不執行存取控制檢查。

關閉物件 (MQCLOSE)

除非 **MQCLOSE** 會導致刪除動態佇列，否則不會執行任何存取控制檢查。 在此情況下，會檢查使用者 ID 是否有權刪除佇列。

訂閱主題 (MQSUB)

當應用程式訂閱主題時，它會指定需要執行的作業類型。 它是建立新的訂閱、變更現存的訂閱，或回復現存的訂閱而不變更它。 對於每一種類型的作業，佇列管理程式會檢查與應用程式相關聯的使用者 ID 是否具有執行作業的權限。

當應用程式訂閱主題時，會針對主題樹狀結構中的主題物件執行權限檢查，這些主題物件位於應用程式訂閱的主題樹狀結構中的點或上方。 權限檢查可能涉及多個主題物件的檢查。

佇列管理程式用於權限檢查的使用者 ID 是應用程式連接至佇列管理程式時從作業系統取得的使用者 ID。

佇列管理程式會對訂閱者佇列執行權限檢查，但不會對受管理佇列執行權限檢查。

ALW IBM MQ on AIX, Linux, and Windows 如何實作存取控制

IBM MQ 使用基礎作業系統所提供的安全服務，並使用物件權限管理程式。 IBM MQ 提供指令來建立及維護存取控制清單。

稱為「授權服務介面」的存取控制介面是 IBM MQ 的一部分。 IBM MQ 提供存取控制管理程式 (符合「授權服務介面」) 的實作，稱為 物件權限管理程式 *(OAM)*。 除非您另行指定 (如 第 301 頁的『在 AIX, Linux, and Windows 系統上防止安全存取檢查』 所述)，否則系統會針對您建立的每一個佇列管理程式自動安裝並啟用此項。 OAM 可以由符合「授權服務介面」的任何使用者或供應商撰寫元件取代。

OAM 使用作業系統使用者和群組 ID 來利用基礎作業系統的安全特性。 只有在使用者具有正確權限時，才能存取 IBM MQ 物件。 第 293 頁的『在 AIX, Linux, and Windows 上使用 OAM 來控制對物件的存取權』 說明如何授與及撤銷此權限。

OAM 會針對它所控制的每一個資源維護存取控制清單 (ACL)。 授權資料儲存在稱為 SYSTEM.AUTH.DATA.QUEUE。 此佇列的存取權僅限於 mqm 群組中的使用者，此外，在 Windows 上，僅限於 Administrators 群組中的使用者，以及使用 SYSTEM ID 登入的使用者。 無法變更使用者對佇列的存取權。

IBM MQ 提供指令來建立及維護存取控制清單。 如需這些指令的相關資訊，請參閱第 293 頁的『在 AIX, Linux, and Windows 上使用 OAM 來控制對物件的存取權』。

IBM MQ 向 OAM 傳遞包含主體、資源名稱及存取類型的要求。 OAM 會根據它所維護的 ACL 來授與或拒絕存取權。 IBM MQ 遵循 OAM 的決策; 如果 OAM 無法做出決策，則 IBM MQ 不容許存取。

ALW 識別 AIX, Linux, and Windows 上的使用者 ID

物件權限管理程式會識別要求存取資源的主體。 作為主體的使用者 ID 會根據環境定義而有所不同。

物件權限管理程式 (OAM) 必須能夠識別要求存取特定資源的人員。 IBM MQ 使用術語 主體 來參照此 ID。 當應用程式第一次連接至佇列管理程式時，即會建立主體; 它是由佇列管理程式從與連接應用程式相關聯的使用者 ID 來決定。 (如果應用程式發出 XA 呼叫而未連接至佇列管理程式，則佇列管理程式會使用與發出 xa_open 呼叫之應用程式相關聯的使用者 ID 進行權限檢查。)

在 AIX and Linux 系統上，授權常式會檢查與應用程式相關聯的實際 (已登入) 使用者 ID 或有效使用者 ID。 所檢查的使用者 ID 可能相依於連結類型，如需詳細資料，請參閱 可安裝的服務。

IBM MQ 會在每一個訊息的訊息標頭 (MQMD 結構) 中傳送從系統收到的使用者 ID，以作為使用者的識別。 此 ID 是訊息環境定義資訊的一部分，並在 第 339 頁的『AIX, Linux, and Windows 上的環境定義權限』中說明。 除非應用程式已獲授權變更環境定義資訊，否則無法變更此資訊。

ALW *AIX, Linux, and Windows* 上的主體及群組

主體可以屬於群組。 透過將資源存取權授與群組而非個人，您可以減少所需的管理量。 「存取控制清單 (ACL)」同時以群組和使用者 ID 為基礎。

例如，您可以定義由想要執行特定應用程式的使用者組成的群組。 將其他使用者的使用者 ID 新增至適當的群組，即可授與他們對所有所需資源的存取權。

此定義及管理群組的程序針對特定平台進行說明:

- AIX 在 AIX 上建立及管理群組
- Linux 在 Linux 上建立及管理群組
- Windows 在 Windows 上建立及管理群組

主體可以屬於多個群組 (其群組集)。 它具有授與其群組集中每一個群組的所有權限的聚集。 會快取這些權限，因此除非您發出 MQSC 指令 **REFRESH SECURITY** (或其 PCF 對等項目)，否則在重新啟動佇列管理程式之前，不會辨識您對主體群組成員資格所做的任何變更。

Linux AIX AIX and Linux 系統

從 IBM MQ 8.0 開始，存取控制清單 (ACL) 同時以使用者 ID 和群組為基礎，您可以將 **SecurityPolicy** 屬性設為適當的值 (如 qm.ini 檔案的服務段落 及 在 AIX and Linux 上配置授權服務段落中所述)，來進行授權。

從 IBM MQ 8.0 開始，您可以使用 使用者型模型 進行授權，這可讓您同時使用使用者和群組。 不過，當您在 setmqaut 指令中指定使用者時，新的許可權僅適用於該使用者，而不適用於該使用者所屬的任何群組。 如需相關資訊，請參閱 UNIX 及 Linux 系統上的 OAM 使用者型許可權。

當您使用 群組型模型 進行授權時，使用者 ID 所屬的主要群組會包含在 ACL 中。 不包括個別使用者 ID，並將權限授與該群組的所有成員。 因此，請注意，您可以透過變更相同群組中另一個主體的權限，意外地變更主體的權限。

所有使用者名義上都會指派給預設使用者群組 nobody，依預設，不會授與此群組任何權限。 您可以變更 nobody 群組中的授權，將 IBM MQ 資源的存取權授與沒有特定授權的使用者。

V 9.2.1 從 IBM MQ 9.2.1，您可以使用 UserExternal**SecurityPolicy** 屬性的選項來建立非作業系統使用者名稱。 如果您建立非作業系統使用者名稱，則會將該使用者視為不屬於任何群組，但 nobody 群組除外。 有關此選項的更多信息，請參閱 crtmqm 和 qm.ini 文件的 Service 節.

請勿定義值為 UNKNOWN 的使用者 ID。 當使用者 ID 太長時，會使用值 UNKNOWN，因此任意使用者 ID 會使用 UNKNOWN 的存取權。

如需使用 LDAP 的相關資訊，請參閱 第 343 頁的『設定權限』。

使用者 ID 最多可以包含 12 個字元，群組名稱最多可以包含 12 個字元。

Windows Windows 系統

ACL 同時以使用者 ID 和群組為基礎。 檢查與 AIX and Linux 的檢查相同。 您可以在不同網域上具有相同使用者 ID 的不同使用者。 IBM MQ 允許依網域名稱來限定使用者 ID，以便可以為這些使用者提供不同層次的存取權。

群組名稱可以選擇性地包括以下列格式指定的網域名稱:

```
GroupName@domain domain_name\group_name
```

在下列兩種情況下， OAM 只會檢查廣域群組:

1. 佇列管理程式安全段落包括下列設定: GroupModel=GlobalGroups。 請參閱 保護安全。
2. 佇列管理程式正在使用替代安全存取群組。 請參閱 **crtmqm**。

使用者 ID 最多可以包含 20 個字元，網域名稱最多可以包含 15 個字元，群組名稱最多可以包含 64 個字元。

OAM 會先檢查本端安全資料庫，然後檢查主要網域的資料庫，最後檢查任何授信網域的資料庫。 OAM 會使用發現的第一個使用者 ID 進行檢查。 其中每一個使用者 ID 在特定電腦上可能具有不同的群組成員資格。

部分控制指令 (例如， **crtmqm**) 會使用物件權限管理程式 (OAM) 來變更對 IBM MQ 物件的權限。 OAM 會依照前述段落中給定的順序來搜尋安全資料庫，以判斷特定使用者 ID 的權限。 因此， OAM 所決定的權限可能會置換使用者 ID 是本端 mqm 群組成員的事實。 例如，如果您透過廣域群組，從具有本端 mqm 群組成員資格的網域控制站所鑑別的使用者 ID 發出 **crtmqm** 指令，則當系統具有本端 mqm 群組中沒有相同名稱的本端使用者時，指令會失敗。

如需在 Windows 上設定 **SecurityPolicy** 屬性的相關資訊，請參閱 可安裝的服務 及 在 Windows 上配置授權服務段落。

Windows *Windows 安全 ID (SID)*

Windows 上的 IBM MQ 使用可用的 SID。 如果授權要求未提供 Windows SID， IBM MQ 只會根據使用者名稱來識別使用者，但這可能會導致授與錯誤的權限。

在 Windows 系統上，安全 ID (SID) 用來補充使用者 ID。 SID 包含資訊，可識別在其中定義使用者的 Windows 安全帳戶管理員 (SAM) 資料庫上的完整使用者帳戶詳細資料。 在 IBM MQ for Windows 上建立訊息時， IBM MQ 會將 SID 儲存在訊息描述子中。 當 IBM MQ on Windows 執行授權檢查時，它會使用 SID 來查詢 SAM 資料庫的完整資訊。 (必須可存取在其中定義使用者的 SAM 資料庫，此查詢才會成功。)

依預設，如果授權要求未提供 Windows SID， IBM MQ 會單獨根據使用者名稱來識別使用者。 它透過依下列順序搜尋安全資料庫來執行此動作:

1. 本端安全資料庫
2. 主要網域的安全資料庫
3. 授信網域的安全資料庫

如果使用者名稱不是唯一的，則可能授與不正確的 IBM MQ 權限。 若要防止此問題，請在每一個授權要求中包含 SID; IBM MQ 會使用 SID 來建立使用者認證。

若要指定所有授權要求都必須包括 SID，請使用 **regedit**。 將 SecurityPolicy 設為 NTSIDsRequired。

ALW AIX, Linux, and Windows 上的替代使用者權限

您可以指定在存取 IBM MQ 物件時，使用者 ID 可以使用另一個使用者的權限。 這稱為 替代使用者權限，您可以在任何 IBM MQ 物件上使用它。

當伺服器接收來自程式的要求，且想要確保程式具有要求的必要權限時，替代使用者權限是必要的。 伺服器可能具有必要的權限，但它需要知道程式是否具有它所要求之動作的權限。

例如，假設以使用者 ID PAYSERV 執行的伺服器程式會從使用者 ID USER1 放置在佇列上的佇列中擷取要求訊息。 當伺服器程式取得要求訊息時，它會處理要求，並將回覆放回要求訊息所指定的回覆佇列中。 伺服器可以指定不同的使用者 ID (在此情況下為 USER1)，而不是使用自己的使用者 ID (PAYSERV) 來授權開啟回覆目的地佇列。 在此範例中，您可以使用替代使用者權限來控制是否容許 PAYSERV 在開啟回覆目的地佇列時指定 USER1 作為替代使用者 ID。

替代使用者 ID 指定在物件描述子的 **AlternateUserId** 欄位上。

Linux 解決 Linux 上的特定群組成員資格問題

部分系統透過正常系列的 **getgrent** 作業系統 API 呼叫緩慢地傳回群組資訊，而且如果您的企業有數千個群組要搜尋，並尋找 mqm 使用者所在的群組，則緩慢回應可能會導致內部佇列管理程式逾時。 為了規避此問題，有一個替代作業系統 API。

若要使用更快的替代 API，並從一個呼叫中傳回所有群組，請設定環境變數 MQS_GETGROUPLIST_API。

當授與連接存取權給使用者的次要群組並啟用 MQS_GETGROUPLIST_API 變數時，您可能收到 RC2035 錯誤，可紓解問題。

然後， IBM MQ 會使用 **getgrouplist** API，而非 **getgrent** API。

若要啟用 **getgrouplist**，請執行下列動作:

1. 停止佇列管理程式
2. 發出指令匯出 MQS_GETGROUPLIST_API=1
3. 重新啟動佇列管理程式

重試失敗的實務範例，如果您的問題已解決，您可以考慮修改使用者 mqm 的 `.bashrc / .profile` 檔來新增此環境變數，或將環境變數新增至您用來啟動佇列管理程式的 Script 中。

如果您的系統從多個儲存庫 (例如 NIS 或 LDAP) 合併作業系統的使用者或群組資訊，請確保群組或使用者 ID 在所有儲存庫 (包括本端儲存庫) 之間一致，因為這些儲存庫用來安裝及設定作業系統層次許可權。

ALW AIX, Linux, and Windows 上的環境定義權限

環境定義是適用於特定訊息的資訊，包含在訊息的訊息描述子 MQMD 中。 當發出 MQOPEN 或 MQPUT 呼叫時，應用程式可以指定環境定義資料。

環境定義資訊分為兩個區段:

身分區段
訊息來自誰。 它由 UserIdentifier、 AccountingToken 和 ApplIdentityData 欄位組成。

原始區段
訊息的來源，以及將訊息放入佇列的時間。 它由 PutApplType、 PutApplName、 PutDate、 PutTime 及 ApplOriginData 欄位組成。

當發出 MQOPEN 或 MQPUT 呼叫時，應用程式可以指定環境定義資料。 依預設，此資料可能由應用程式產生、從另一則訊息傳遞，或由佇列管理程式產生。 例如，伺服器程式可以使用環境定義資料來檢查要求端的身分，測試訊息是否來自以授權使用者 ID 執行的應用程式。

伺服器程式可以使用 UserIdentifier 來決定替代使用者的使用者 ID。 您可以使用環境定義授權來控制使用者是否可以在任何 MQOPEN 或 MQPUT1 呼叫上指定任何環境定義選項。

如需環境定義選項的相關資訊，請參閱 控制環境定義資訊 ; 如需環境定義相關訊息描述子欄位的說明，請參閱 MQMD 概觀 。

在安全結束程式中實作存取控制

您可以使用 MCAUserIdentifier 或物件權限管理程式，在安全結束程式中實作存取控制。

MCAUserIdentifier

現行通道的每一個實例都有相關聯的通道定義結構 MQCD。 MQCD 中欄位的起始值由 IBM MQ 管理者所建立的通道定義決定。 具體而言，其中一個欄位 *MCAUserIdentifier* 的起始值由 DEFINE CHANNEL 指令上的 MCAUSER 參數值決定，如果以另一種方式建立通道定義，則由相等於 MCAUSER 的值決定。

當 MCA 呼叫 MQCD 結構時，會將它傳遞給通道結束程式。 當 MCA 呼叫安全結束程式時，安全結束程式可以變更 *MCAUserIdentifier* 的值，以取代通道定義中指定的任何值。

Multi 在 多平台上，除非 *MCAUserIdentifier* 的值為空白，否則佇列管理程式會在 MCA 連接至佇列管理程式之後，使用 *MCAUserIdentifier* 的值作為使用者 ID ，以進行權限檢查。 如果 *MCAUserIdentifier* 的值為空白，則佇列管理程式會改用 MCA 的預設使用者 ID。 這適用於 RCVR、RQSTR、CLUSRCVR 及 SVRCONN 通道。 對於傳送 MCA ，一律使用預設使用者 ID 進行權限檢查，即使 *MCAUserIdentifier* 的值不是空白。

z/OS 在 z/OS 上，佇列管理程式可能會使用 *MCAUserIdentifier* 的值來進行權限檢查，前提是它不是空白。 對於接收 MCA 及伺服器連線 MCA ，佇列管理程式是否使用 *MCAUserIdentifier* 的值進行權限檢查取決於:

- 通道定義中 PUTAUT 參數的值
- 用於檢查的 RACF 設定檔
- RESLEVEL 設定檔之通道起始程式位址空間使用者 ID 的存取層次

對於傳送 MCA ，它取決於:

- 傳送端 MCA 是呼叫端還是回應端
- RESLEVEL 設定檔之通道起始程式位址空間使用者 ID 的存取層次

安全結束程式儲存在 *MCAUserIdentifier* 中的使用者 ID ，可以透過各種方式獲得。 這裡是一些範例:

- 假設 MQI 通道的用戶端端沒有安全結束程式，當用戶端應用程式發出 MQCONN 呼叫時，與 IBM MQ 用戶端應用程式相關聯的使用者 ID 會從用戶端連線 MCA 傳送至伺服器連線 MCA。伺服器連線 MCA 會將這個使用者 ID 儲存在通道定義結構 MQCD 的 *RemoteUserIdentifier* 欄位中。 如果此時 *MCAUserIdentifier* 的值為空白，則 MCA 會將相同的使用者 ID 儲存在 *MCAUserIdentifier* 中。 如果 MCA 未將使用者 ID 儲存在 *MCAUserIdentifier* 中，安全結束程式可以稍後將 *MCAUserIdentifier* 設為 *RemoteUserIdentifier* 值來執行此動作。

  如果來自用戶端系統的使用者 ID 正在進入新的安全網域，且在伺服器系統上無效，則安全結束程式可以將使用者 ID 替換為有效的使用者 ID ，並將替換的使用者 ID 儲存在 *MCAUserIdentifier* 中。
- 夥伴安全結束程式可以在安全訊息中傳送使用者 ID。

  在訊息通道上，傳送端 MCA 所呼叫的安全結束程式可以傳送傳送端 MCA 執行所用的使用者 ID。 然後接收 MCA 所呼叫的安全結束程式可以將使用者 ID 儲存在 *MCAUserIdentifier* 中。 同樣地，在 MQI 通道上，通道用戶端的安全結束程式可以傳送與 IBM MQ MQI client 應用程式相關聯的使用者 ID。 然後通道伺服器端的安全結束程式可以將使用者 ID 儲存在 *MCAUserIdentifier* 中。 如前一個範例所示，如果使用者 ID 在目標系統上無效，則安全結束程式可以將使用者 ID 替換為有效的使用者 ID ，並將替換的使用者 ID 儲存在 *MCAUserIdentifier* 中。

  如果接收數位憑證作為識別及鑑別服務的一部分，則安全結束程式可以將憑證中的「識別名稱」對映至目標系統上有效的使用者 ID。 然後，它可以將使用者 ID 儲存在 *MCAUserIdentifier* 中。

- 如果在通道上使用 TLS，則會將夥伴的「識別名稱 (DN)」傳遞至 MQCD 的 SSLPeerNamePtr 欄位中的結束程式，並將該憑證發證者的 DN 傳遞至 MQCXP 的 SSLRemCertIssNamePtr 欄位中的結束程式。

如需 *MCAUserIdentifier* 欄位、通道定義結構 MQCD 及通道結束程式參數結構 MQCXP 的相關資訊，請參閱通道結束程式呼叫及資料結構。 如需在 MQI 通道上從用戶端系統流動之使用者 ID 的相關資訊，請參閱 存取控制。

註: 在 IBM WebSphere MQ 7.1 版本之前建構的安全結束程式應用程式可能需要更新。 如需相關資訊，請參閱 通道安全結束程式。

IBM MQ 物件權限管理程式使用者鑑別

在 IBM MQ MQI client 連線上，安全結束程式可用來修改或建立物件權限管理程式 (OAM) 使用者鑑別中使用的 MQCSP 結構。 這在 傳訊通道的通道結束程式 中有說明。

在訊息結束程式中實作存取控制

您可能需要使用訊息結束程式，將一個使用者 ID 替換為另一個使用者 ID。

請考量將訊息傳送至伺服器應用程式的用戶端應用程式。 伺服器應用程式可以從訊息描述子中的 *UserIdentifier* 欄位擷取使用者 ID，如果它具有替代使用者權限，則當它代表用戶端存取 IBM MQ 資源時，會要求佇列管理程式使用此使用者 ID 進行權限檢查。

如果 PUTAUT 參數設為 CTX (或 z/OS 上的 ALTMCA) 在通道定義中，當 MCA 開啟目的地佇列時，會使用每一個送入訊息的 *UserIdentifier* 欄位中的使用者 ID 來進行權限檢查。

在某些情況下，產生報告訊息時，會使用導致報告之訊息的 *UserIdentifier* 欄位中使用者 ID 的權限來放置報告訊息。 尤其是「確認交付 (COD)」報告和到期報告一律具有此權限。

由於這些狀況，當訊息進入新的安全網域時，可能需要在 *UserIdentifier* 欄位中以一個使用者 ID 替代另一個使用者 ID。 這可以透過通道接收端的訊息結束程式來完成。 或者，您可以確定送入訊息的 *UserIdentifier* 欄位中的使用者 ID 已定義在新的安全網域中。

如果送入訊息包含傳送訊息之應用程式使用者的數位憑證，則訊息結束程式可以驗證憑證，並將憑證中的「識別名稱」對映至在接收系統上有效的使用者 ID。 然後，它可以將訊息描述子中的 *UserIdentifier* 欄位設為這個使用者 ID。

如果訊息結束程式必須變更送入訊息中 *UserIdentifier* 欄位的值，則訊息結束程式可能適合同時鑑別訊息的傳送者。 如需詳細資料，請參閱第 281 頁的『訊息結束程式中的身分對映』。

在 API 結束程式和 API 交互結束程式中實作存取控制

API 或 API 交互結束程式可以提供存取控制，以補充 IBM MQ 所提供的存取控制。 特別是，結束程式可以在訊息層次提供存取控制。 結束程式可確保應用程式只會將符合特定準則的訊息放置在佇列上，或從佇列中取得。

請考量下列範例:

- 訊息包含訂單的相關資訊。 當應用程式嘗試將訊息放入佇列時， API 或 API 交互結束程式可以檢查訂單的總值是否小於某些規定的限制。
- 從遠端佇列管理程式抵達目的地佇列的訊息。 當應用程式嘗試從佇列取得訊息時， API 或 API 交互結束程式可以檢查訊息傳送端是否已獲授權將訊息傳送至佇列。

V 9.2.3 Multi 串流佇列安全

串流佇列特性可讓管理者使用次要佇列來配置本端 (或模型) 佇列，每當將訊息放入原始佇列時，即會放置重複的訊息。 關於佇列串流權限，有兩個方面需要考量。

為串流重複訊息配置佇列的權限

如果您想要啟用將重複訊息從一個佇列串流至次要佇列的訊息串流，則必須具有這樣做的許可權。 您必須具備下列權限，才能配置佇列的 **STREAMQ** 屬性:

1. 變更其 **STREAMQ** 屬性之佇列的 CHG 權限
2. 您要放置重複訊息之佇列的 CHG 權限

在配置時，這兩項權限檢查的組合可確保只對原始佇列具有 CHG 權限的使用者，無法將訊息放置在他們沒有許可權的另一個佇列中。

開啟一或多個佇列及放置訊息的權限

當應用程式透過其 **STREAMQ** 屬性開啟已配置次要佇列的佇列時，會進行權限檢查，確定應用程式使用者對原始佇列具有 PUT 權限。

註: 不會對次要佇列上的應用程式使用者進行其他權限檢查，這與用於別名佇列的權限模型類似。

從原始佇列或次要佇列耗用訊息的應用程式只需要 GET 或 BROWSE 權限，即可使用它們所耗用的佇列。

不會在放置或取得時間進行其他授權檢查。

範例

下列範例顯示設定的正確權限，可讓使用者 admin 配置原始佇列 INQUIRIES.QUEUE，將其重複訊息串流至本端佇列 ANALYTICS.QUEUE，但防止 admin 將訊息複製到 PURCHASES.QUEUE:

```
SET AUTHREC PROFILE(INQUIRIES.QUEUE) PRINCIPAL('admin') AUTHADD(CHG)
SET AUTHREC PROFILE(ANALYTICS.QUEUE) PRINCIPAL('admin') AUTHADD(CHG)
SET AUTHREC PROFILE(PURCHASES.QUEUE) PRINCIPAL('admin') AUTHRMV(CHG)
```

然後，使用者 admin 能夠發出下列指令:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(ANALYTICS.QUEUE)
```

但如果相同的使用者發出下列指令:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(PURCHASES.QUEUE)
```

以配置 INQUIRIES.QUEUE ，將重複訊息放置到 PURCHASES.QUEUE，它們會收到下列錯誤:

錯誤 TBD

具有 INQUIRIES.QUEUE 已配置成將訊息複製到 ANALYTICS.QUEUE，下列權限記錄用於容許以使用者 appuser 身分執行的應用程式將訊息放置到 INQUIRIES.QUEUE，以及複製到 ANALYTICS.QUEUE:

```
SET AUTHREC PROFILE(INQUIRIES.QUEUE) PRINCIPAL('appuser') AUTHADD(PUT)
```

註: appuser 不需要 ANALYTICS.QUEUE。 佇列管理程式會將重複的訊息放入佇列。

相關概念
串流佇列

Multi LDAP 授權

您可以使用 LDAP 授權來移除本端使用者 ID 的需求。

受支援平台上 LDAP 授權的可用性

可在 Multiplatforms 上使用 LDAP 授權:

小心:

從 IBM MQ 9.0 通用版開始，不論全新或從舊版移轉，所有佇列管理程式都可以使用此功能。

LDAP 授權概觀

使用 LDAP 授權，處理授權配置的指令 (例如 **setmqaut** 和 **DISPLAY AUTHREC**) 可以處理「識別名稱」。先前，使用者是透過比較其認證與本端作業系統上使用者和群組的可用字元數上限來進行鑑別。

小心: 如果您已執行 **DEFINE AUTHINFO** 指令，則必須重新啟動佇列管理程式。 如果您未重新啟動佇列管理程式，則 setmqaut 指令不會傳回正確的結果。

如果使用者提供使用者 ID，而不是「識別名稱」，則會處理使用者 ID。 例如，當具有 PUTAUT (CTX) 的通道上有送入訊息時，使用者 ID 中的字元會對映至「LDAP 識別名稱」，並進行適當的授權檢查。

其他指令 (例如 **DISPLAY CONN**) 會繼續使用並顯示使用者 ID 的實際值，即使該使用者 ID 可能實際上並不存在於本端 OS 上。

Linux AIX 當 LDAP 授權就緒時，不論 qm.ini 檔中的 **SecurityPolicy** 屬性為何，佇列管理程式一律會使用 AIX and Linux 平台上的使用者安全模型。 因此，設定個別使用者的許可權只會影響該使用者，而不會影響屬於該使用者任何群組的其他任何人。

與 OS 模型一樣，使用者仍具有已指派給個人及使用者所屬所有群組 (如果有的話) 的結合權限。

例如，假設已在 LDAP 儲存庫中定義下列記錄。

- 在 **inetOrgPerson** 類別中:

```
dn="cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry"
        email=JohnDoe1@yourcompany.com [longer than 12 characters]
        shortu=jodoe
        Phone=1234567
```

- 在 **groupOfNames** 類別中:

```
dn="cn=Application Group A, ou=groups, o=yourcompany, c=yourcountry"
        longname=ApplicationGroupA [longer than 12 characters]
        members="cn=JaneDoe, ou=users, o=yourcompany, c=yourcountry",
                "cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry"
```

基於鑑別目的，必須已定義使用此 LDAP 伺服器的佇列管理程式，使其 **CONNAUTH** 值指向類型 IDPWLDAP 的 **AUTHINFO** 物件，且其相關名稱解析屬性可能設定如下:

```
USRFIELD(email) SHORTUSR(shortu)
BASEDNU(ou=users,o=yourcompany,c=yourcountry) CLASSUSR(inetOrgPerson)
```

在此鑑別配置下，應用程式可以使用下列其中一組值來完成 MQCNO 呼叫內使用的 CSPUserID 欄位:

```
" cn=JohnDoe ", " JohnDoe1@yourcompany.com ", " email=JohnDoe1@yourcompany.com "
```

or

```
" cn=JohnDoe, ou=users, o=ibm, c=uk ", " shortu=jodoe "
```

在任一情況下，系統都可以使用提供的值來鑑別 "" 的 OS 環境定義 jodoe"。

Multi 設定權限

如何使用短名稱或 **USRFIELD** 來設定授權。

第 342 頁的『LDAP 授權』中所述的使用多種格式的方法繼續延伸到授權命令中，並進一步擴展為 shortname 或 USRFIELD 可以以樸素的方式使用。

字串在命名使用者（主體）進行授權時指定 LDAP 記錄中的特定屬性。

重要: 該字串不能包含=字符，因為該字符不能在作業系統用戶 ID 中使用。

如果將主體名稱傳遞給 OAM 進行授權（可能是 shortname，則字串必須適合 12 個字元。 映射演算法首先嘗試使用 LDAP 查詢中的 SHORTUSR 屬性將其解析為 DN。

如果失敗並出現 UNKNOWN_ENTITY 錯誤，或者給定的字串不可能是 shortname，則將進一步嘗試使用 USRFIELD 屬性來建構 LDAP 查詢。

小心: 如果您已執行 DEFINE AUTHINFO 指令，則必須重新啟動佇列管理器。 如果不重新啟動佇列管理器， setmqaut 指令不會回傳正確的結果。

對於處理使用者授權，以下 setmqaut 指令設定都是等效的。

表 72: 使用者授權設定	
指令	**附註**
`setmqaut -m QM -t qmgr -p jodoe +connect`	這是一個扁平的、不合格的名稱，透過 SHORTUSR 解析。
`setmqaut -m QM -t qmgr -pJohnDoe1@yourcompany.com +connect`	也是一個扁平的、不合格的名稱，透過 USRFIELD 解析為同一實體。
`setmqaut -m QM -t qmgr -p email=JohnDoe1@yourcompany.com +connect`	使用命名屬性。
`setmqaut -m QM -t qmgr -p "phone=1234567" +connect`	使用另一個命名屬性，該屬性不必是 AUTHINFO 物件上配置的任何屬性。

您可以使用 SET AUTHREC MQSC 指令作為 **setmqaut** 指令的替代指令：

```
SET AUTHREC OBJTYPE(QMGR) PRINCIPAL('JohnDoe1@yourcompany.com') AUTHADD(connect)
```

或設定權限記錄 (MQCMD_SET_AUTH_REC) PCF 指令，其中 MQCACF_PRINCIPAL_ENTITY_NAMES 元素包含下列字串：

```
  "cn=JohnDoe,ou=users,o=yourcompany,c=yourcountry"
```

處理群組時， `shortname` 處理不會有任何歧義，因為不需要將任何形式的群組名稱放入 12 個字元中。 因此，群組中不存在與 SHORTUSR 屬性等效的屬性。

這表示第 344 頁的表 73 中所述的語法範例是有效的，假設您已使用擴充屬性配置了 AUTHINFO 對象，並設定為：

```
GRPFIELD(longname)
BASEDNG(ou=groups,o=yourcompany,c=yourcountry ) CLASSGRP(groupOfNames)
```

表 73: 群組授權設定	
指令	**附註**
`setmqaut -m QM -t qmgr -g ApplicationGroupA +connect`	使用 GRPFIELD 來解決
`setmqaut -m QM -t qmgr -g longname=ApplicationGroupA +connect`	命名單一屬性
`setmqaut -m QM -t qmgr -g "cn=Application Group A,ou=groups,o=yourcompany,c=yourcountry" +connect`	使用完整 DN

您可以使用 SET AUTHREC MQSC 指令作為前面 **setmqaut** 指令的替代指令：

```
SET AUTHREC OBJTYPE(QMGR) GROUP('ApplicationGroupA')
      AUTHADD(connect)
```

或設定權限記錄 (MQCMD_SET_AUTH_REC) PCF 命令，其中 MQCACF_GROUP_ENTITY_NAMES 元素包含下列字串：

```
  "ApplicationGroupA"
```

重要：

無論您使用哪種格式來引用名稱，無論是使用者還是群組，都必須能夠派生出唯一的 DN。

因此，例如，您不能擁有兩個都具有“ `shortu=jodoe` ”的不同記錄。

如果無法確定單一唯一 DN，OAM 將傳回 MQRC_UNKNOWN_ENTITY。

Multi 顯示授權

顯示使用者或群組授權的各種方法。

dspmqaut 指令

顯示使用者或群組可用的授權的最簡單方法是使用 dspmqaut 命令。

您可以使用任何語法變體的查詢來識別使用者或群組。 請注意，命令輸出以命令列上給定的格式重複標識。 輸出不報告完整解析的 DN。

例如：

```
dspmqaut -m QM -t qmgr -p johndoe
Entity johndoe has the following authorizations for object QM:
    connect
```

or

```
dspmqaut -m QM -t qmgr -p email=JohnDoe1@yourcompany.com
Entity email=JohnDoe1@yourcompany.com has the following authorizations for object QM:
    connect
```

dmpmqaut 和 dmpmqcfg 指令

dmpmqaut 指令及其 MQSC 或 PCF 等效項可以以任何受支援的格式指定主體或群組，例如第 343 頁的『設定權限』中所述的 **setmqaut** 表。 但是，與 **dspmqaut** 不同， **dmpmqaut** 命令始終報告完整的 DN。

```
dmpmqaut -m QM -t qmgr -p jodoe
------------------------------------
profile: self
object type:qmgr
entity:cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry
entity type: principal
authority: connect
```

同樣， dmpmqcfg 指令對所選記錄沒有任何過濾，始終以稍後可以重播的格式顯示完整 DN。

```
dmpmqcfg -m QM -x authrec
------------------------------------
SET AUTHREC PROFILE(SELF) +
    PRINCIPAL('cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry') +
    OBJTYPE(QMGR)
    AUTHADD(CONNECT)
```

Multi 使用 LDAP 授權時的其他考量

從 IBM MQ 9.0.0 使用 LDAP 授權時需要注意的「訊息佇列介面 (MQI)」及其他 MQSC 及 PCF 指令變更的簡要說明。

ADOPTCTX

應用程式不需要提供鑑別資訊，也不需要將 ADOPTCTX 屬性設為 YES。

如果應用程式未明確鑑別，或作用中 CONNAUTH 物件的 **ADOPTCTX** 設為 NO ，則會從作業系統使用者 ID 取得與應用程式相關聯的身分環境定義。

當需要套用授權時，會使用與 setmqaut 指令相同的規則，將該環境定義對映至 LDAP 身分。

MQI 呼叫的輸入參數

MQOPEN、 MQPUT1 及 MQSUB 具有容許指定替代使用者 ID 的結構。

如果使用這些欄位，則會使用與 **setmqaut**、 **dmpmqaut** 及 **dspmqaut** 指令上相同的規則，將 12 個字元的使用者 ID 對映至 DN。

MQPUT 和 MQPUT1 也容許適當授權的程式設定 MQMD UserIdentifier 欄位。 在 PUT 處理程序期間不會對此欄位的值進行輪詢，並且可以設為任何值。

不過，與平常一樣，在訊息處理的後續階段，例如在接收通道上定義 PUTAUT (CTX) 時，**UserIdentifier** 值可以用於授權。

此時，將使用該接收端佇列管理程式的配置 (可以是 LDAP 或 OS 型) 來檢查 ID 的授權。

MQI 呼叫的輸出參數

無論使用者 ID 在 MQI 結構中提供給程式，都是與連線相關聯的 12 個字元簡稱版本。

例如， API 結束程式的 **MQAXC.UserId** 值是從 LDAP 對映傳回的簡稱。

其他管理 MQSC 及 PCF 指令

以物件狀態 (例如 DISPLAY CONN USERID) 顯示使用者資訊的指令會傳回與環境定義相關聯的 12 個字元簡稱。 不會顯示完整 DN。

容許主張身分的指令 (例如通道的 CHLAUTH 對映規則或 MCAUSER 值) 可以採用高達為那些屬性定義的長度上限 (目前為 64 個字元) 的值。

語法沒有變更。 當該身分需要授權時，會使用與 **setmqaut**、 **dmpmqaut** 及 **dspmqaut** 指令相同的規則，在內部將它對映至 DN。

這表示通道定義上的 MCAUSER 值可能不會顯示為與 DISPLAY CHSTATUS 相同的字串，但它們會參照相同的身分。

例如:

```
DEFINE CHL(SV1) CHLTYPE(SVRCONN) MCAUSER('cn=JohnDoe')
DEFINE CHL(SV2) CHLTYPE(SVRCONN) MCAUSER('jodoe')
DEFINE CHL(SV3) CHLTYPE(SVRCONN) MCAUSER('JohnDoe1@yourcompany.com')
```

然後 DISPLAY CHSTATUS (*) ALL 會顯示所有連線的 SHORTUSR 值 *MCAUSER (jodoe)* 。

Multi 在 OS 與 LDAP 授權模型之間切換

如何在不同平台上的不同授權方法之間切換。

佇列管理程式的 CONNAUTH 屬性指向 AUTHINFO 物件。 當物件類型為 IDPWLDAP 時，會使用 LDAP 儲存庫進行鑑別。

您現在可以將授權方法套用至該相同物件，這可讓您繼續使用 OS 型授權，或使用 LDAP 授權

IBM i, AIX and Linux

IBM i Linux AIX

可以隨時在 OS 與 LDAP 模型之間切換佇列管理程式。 您可以使用 REFRESH SECURITY TYPE (CONNAUTH) 指令來變更配置並使該配置處於作用中。

比方說，如果這個物件已配置鑑別的連線資訊:

```
ALTER AUTHINFO(MYLDAP) AUTHTYPE(IDPWLDAP) +
        AUTHORMD(SEARCHGRP) +
        BASEDNG('ou=groups,o=ibm,c=uk') +
        <other attributes>
ALTER QMGR CONNAUTH(MYLDAP)
REFRESH SECURITY
```

Windows

Windows

如果權限配置變更涉及在 OS 與 LDAP 模型之間切換，則必須重新啟動佇列管理程式，變更才會生效。 否則，您可以使用 REFRESH SECURITY TYPE (CONNAUTH) 指令將變更設為作用中。

處理規則

從 OS 切換至 LDAP 授權時，任何已設定的現有 OS 權限規則都會變成非作用中及隱藏。

dmpmqaut 之類的指令不會顯示那些 OS 規則。 同樣地，從 LDAP 切換回 OS 時，任何已定義的 LDAP 授權都會變成非作用中及隱藏，還原原始 OS 規則。

如果您基於任何原因而想要使用 **dmpmqcfg** 指令來備份佇列管理程式的定義，則該備份只會包含在備份時生效的授權方法所定義的規則。

Multi LDAP 管理

每一個平台如何管理 LDAP 的概觀。

使用 LDAP 授權時，作業系統中 mqm 群組 (或對等項目) 的成員資格並不重要。 作為該群組的成員，只會控制是否可以處理某些指令行指令。

尤其是您必須在該群組中，才能發出 strmqm 和 endmqm 指令。

一旦佇列管理程式在執行中，現在就會限制完全特許帳戶。 除了發出 **strmqm** 指令之人員的使用者 ID 之外，屬於 OS mqm (或對等項目) 群組的其他使用者也不會取得特殊專用權。

其他使用者的授權是根據他們所屬的 LDAP 群組而定。 不容許在指令 (例如 **setmqaut**) 中不完整使用 mqm 群組名稱，以對映至任何 LDAP 群組。

AIX and Linux

Linux AIX

在佇列管理程式執行之後，唯一自動完全特許的帳戶是啟動佇列管理程式的實際使用者。

mqm ID 仍然存在，並用作 OS 資源 (例如檔案) 的擁有者，因為 mqm 是執行佇列管理程式的有效 ID。 不過，mqm 使用者將無法自動執行 OAM 所控制的管理作業。

Windows

Windows

在 Windows 上，自動完全特許帳戶是啟動佇列管理程式的 OS 使用者，以及執行核心佇列管理程式處理程序的使用者，例如 MUSR_MQADMIN (如果佇列管理程式是以 Windows 服務方式啟動)。

以 LDAP 授權模式執行時， Windows 的行為與 AIX and Linux 平台非常相似。 它處理 12 個字元的簡稱及完整 DN。

IBM i

IBM i

在 IBM i 上，自動特許帳戶是啟動佇列管理程式及 QMQM ID 的帳戶。

您需要這兩個 ID ，因為只有在啟動系統時才需要啟動佇列管理程式的使用者 ID。 執行之後，佇列管理程式處理程序僅具有 QMQM 權限。

提供 MQADMIN 專用權的範例 Script

Linux AIX

由於讓群組能夠在佇列管理程式上執行完整管理非常有用，因此在 AIX and Linux 平台上提供範例 Script 如下:

```
MQ_INSTALLATION_PATH/samp/bin/amqauthg.sh
```

此範例採用兩個參數:

- 佇列管理程式名稱
- LDAP 群組名稱

範例會處理 setmqaut 指令，並授與所有物件的完整權限。 這是「IBM MQ Explorer OAM 精靈」針對管理角色所產生的相同 Script。 例如，程式碼會啟動:

```
setmqaut -t q -m qmgr -n "**" +alladm -g
      groupname
```

訊息機密性

加密訊息可確保訊息內容保持機密。 視您的需求而定， IBM MQ 中有各種加密訊息的方法。

如果您需要應用程式層次、點對點傳訊基礎架構的端對端資料保護，您可以 使用 Advanced Message Security 來加密訊息，或 撰寫您自己的 API 結束程式或 API 交互結束程式。

最安全的解決方案是提供端對端加密，方法是將訊息從應用程式放置的點加密到消費端應用程式取得的點。 這可以使用 第 88 頁的『規劃 Advanced Message Security』 (AMS) 或撰寫您自己的 API 結束程式或跨 API 結束程式來完成; 如需相關資訊，請參閱 第 390 頁的『在使用者結束程式中實作機密性』 。

如果您只需要在透過網路傳輸訊息時加密訊息，則可以使用 TLS; 如需相關資訊，請參閱 第 20 頁的『IBM MQ 中的 TLS 安全通訊協定』 ，或者您可以撰寫自己的安全結束程式、訊息結束程式或傳送及接收結束程式來執行加密。

z/OS V 9.2.0 如果您需要在佇列管理程式上加密靜止的訊息，您可以在該佇列管理程式上使用 z/OS 資料集加密; 請參閱 在 IBM MQ for z/OS 上使用資料集加密的靜止資料的機密性 。 的文件以取得相關資訊。

相關工作

使用 TLS 連接兩個佇列管理程式

將用戶端安全連接至佇列管理程式

啟用 CipherSpecs

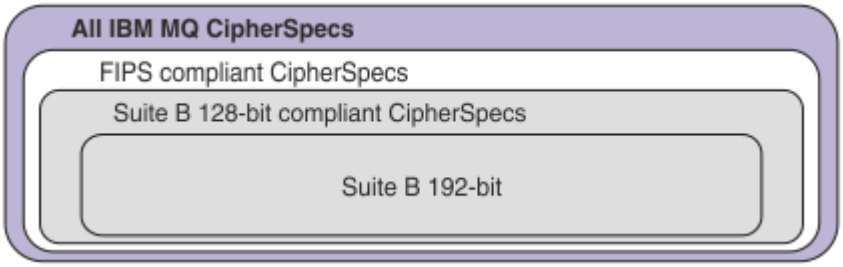
在 **DEFINE CHANNEL** 或 **ALTER CHANNEL** MQSC 指令中使用 **SSLCIPH** 參數來啟用 CipherSpec 。

註: 在 AIX, Linux, and Windows 上， IBM MQ 透過 IBM Crypto for C (ICC) 加密模組提供 FIPS 140-2 相符性。 此模組的憑證已移至「歷程」狀態。 客戶應該檢視 IBM Crypto for C (ICC) 憑證 ， 並注意 NIST 提供的任何建議。 目前正在進行取代 FIPS 140-3 模組，您可以在 處理程序清單中的 NIST CMVP 模組中搜尋它，以檢視其狀態。

您可以與 IBM MQ 搭配使用的部分 CipherSpecs 符合 FIPS 標準。 部分符合 FIPS 標準的 CipherSpecs 也符合「套組 B」標準，但其他例如 TLS_RSA_WITH_AES_256_CBC_SHA 則不符合。

所有 Suite B 相容 CipherSpecs 也符合 FIPS 標準。 所有 Suite B 相容 CipherSpecs 分為兩個群組: 128 位元 (例如， ECDHE_ECDSA_AES_128_GCM_SHA256) 和 192 位元 (例如， ECDHE_ECDSA_AES_256_GCM_SHA384) ,

下圖說明這些子集之間的關係:



V 9.2.0 V 9.2.0 從 IBM MQ 9.2.0 開始，產品在所有平台上都支援 TLS 1.3 安全通訊協定。

z/OS 在 IBM MQ for z/OS 上， TLS 1.3 僅在 z/OS 2.4 或更新版本上受支援。

第 349 頁的表 74 中列出您可以用於其中每一個平台的 CipherSpecs 。 有關使用這些 CipherSpecs,請參閱第 352 頁的『在 IBM MQ 中使用 TLS 1.3』和第 352 頁的『IBM MQ MQI client 及 TLS 1.3』 。

為了易於配置及未來移轉， IBM MQ 也提供一組別名 CipherSpecs。 移轉現有的安全配置以使用別名 CipherSpec ，表示您可以適應密碼新增及淘汰，而不需要在未來進行進一步的侵入性配置變更。 這些別名 CipherSpecs 會列在 第 349 頁的表 74 的「別名 CipherSpecs 」區段中。 有關遷移以使用別名 CipherSpec,請參閱遷移現有安全配置以使用別名 CipherSpec 。

V 9.2.0 您可以依照 第 352 頁的『在 IBM MQ 中啟用預設 CipherSpec 值』中的說明來配置預設 CipherSpecs 。 您也可以提供替代的 CipherSpecs 集， 這些 CipherSpec 可與上的通道搭配使用:

- Multi IBM MQ for Multiplatforms， 如 第 360 頁的『在 IBM MQ for Multiplatforms 上提供已訂購及已啟用 CipherSpecs 的自訂清單』中所述。
- z/OS IBM MQ for z/OS， 如 第 361 頁的『在 IBM MQ for z/OS 上提供已訂購及已啟用 CipherSpecs 的自訂清單』中所述。

第 361 頁的『已淘汰 CipherSpecs』中列出您可以重新啟用以在必要時與 IBM MQ 搭配使用的已淘汰 CipherSpecs 。 有關啟用已棄用的 CipherSpecs,請參閱第 364 頁的『在 IBM MQ for Multiplatforms 上啟用已淘汰的 CipherSpecs』或第 365 頁的『在 z/OS 上啟用已淘汰的 CipherSpecs』 。

可與 IBM MQ TLS 支援搭配使用的 CipherSpecs

下表列出您可以與 IBM MQ 佇列管理程式自動搭配使用的 CipherSpecs 。 當您要求個人憑證時， 要指定公開與私密金鑰組之金鑰大小。 TLS 握手期間使用的金鑰大小是儲存在憑證中的大小， 除非它由 CipherSpec, 如表中所述。

表 *74:* 可以與 *IBM MQ TLS* 支援一起使用的 *CipherSpec*

平台支援 第 351 頁的『1』	CipherSpec 名稱	十六進位碼	使用的通訊協定	MAC 演算法	加密演算法 (加密位元)	FIPS 第 351 頁的『2』	套組 B
別名 CipherSpec							
全部	ANY_TLS13_OR_HIGHER 第 351 頁的『3』 第 351 頁的『4』 第 351 頁的『5』	N/A	已協議	已協議	已協議	已協議	已協議
全部	ANY_TLS13 第 351 頁的『4』 第 351 頁的『5』 第 351 頁的『6』	N/A	TLS 1.3	已協議	已協議	已協議	已協議
全部	ANY_TLS12_OR_HIGHER 第 351 頁的『4』 第 351 頁的『5』 第 351 頁的『7』	N/A	已協議	已協議	已協議	已協議	已協議
全部	ANY_TLS12 第 351 頁的『8』	N/A	TLS 1.2	已協議	已協議	已協議	已協議

表 74: 可以與 *IBM MQ TLS* 支援一起使用的 *CipherSpec (*繼續*)*

平台支援 第 351 頁的『1』	CipherSpec 名稱	十六進位碼	使用的通訊協定	MAC 演算法	加密演算法 (加密位元)	FIPS 第 351 頁的『2』	套組 B
全部	ANY 第 351 頁的『9』	N/A	已協議	已協議	已協議	已協議	已協議
適用於 TLS 1.3 的 CipherSpec							
全部	TLS_AES_128_GCM_SHA256 第 351 頁的『4』	1301	TLS 1.3	GCM	AES-128，含 GCM (128)	是	否
全部	TLS_AES_256_GCM_SHA384 第 351 頁的『4』	1302	TLS 1.3	GCM	AES-256 (含 GCM (256))	是	否
全部	TLS_CHACHA20_POLY1305_SHA256 第 351 頁的『4』	1303	TLS 1.3	POLY1305	CHACHA20 (256)	否	否
ALW	TLS_AES_128_CCM_SHA256	1304	TLS 1.3	CBC-MAC	AES-128 (含 CTR (128))	是	否
ALW	TLS_AES_128_CCM_8_SHA256 第 351 頁的『11』	1305	TLS 1.3	CBC-MAC	AES-128 (含 CTR (128))	是	否
適用於 TLS 1.2 的 CipherSpec							
全部	TLS_RSA_WITH_AES_128_CBC_SHA256 第 351 頁的『10』	003C	TLS 1.2	SHA-256	AES (128)	是	否
全部	TLS_RSA_WITH_AES_256_CBC_SHA256 第 351 頁的『10』 第 351 頁的『12』	003D	TLS 1.2	SHA-256	AES (256)	是	否
全部	TLS_RSA_WITH_AES_128_GCM_SHA256 第 351 頁的『10』 第 351 頁的『13』	009C	TLS 1.2	SHA-256 和 AEAD GCM	AES (128)	是	否
全部	TLS_RSA_WITH_AES_256_GCM_SHA384 第 351 頁的『10』 第 351 頁的『12』 第 351 頁的『13』	009D	TLS 1.2	SHA-384 和 AEAD GCM	AES (256)	是	否
全部	ECDHE_ECDSA_AES_128_CBC_SHA256 第 351 頁的『10』	C023	TLS 1.2	SHA-256	AES (128)	是	否
全部	ECDHE_ECDSA_AES_256_CBC_SHA384 第 351 頁的『10』 第 351 頁的『12』	C024	TLS 1.2	SHA-384	AES (256)	是	否
全部	ECDHE_RSA_AES_128_CBC_SHA256 第 351 頁的『10』	C027	TLS 1.2	SHA-256	AES (128)	是	否
全部	ECDHE_RSA_AES_256_CBC_SHA384 第 351 頁的『10』 第 351 頁的『12』	C028	TLS 1.2	SHA-384	AES (256)	是	否
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256 第 351 頁的『12』 第 351 頁的『13』	C02B	TLS 1.2	SHA-256 和 AEAD GCM	AES (SHA384)	是	128 位元

表 74: 可以與 *IBM MQ TLS* 支援一起使用的 *CipherSpec (*繼續)

平台支援 第 351 頁的『1』	CipherSpec 名稱	十六進位碼	使用的通訊協定	MAC 演算法	加密演算法 (加密位元)	FIPS 第 351 頁的『2』	套組 B
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384 第 351 頁的『12』 第 351 頁的『13』	C02C	TLS 1.2	SHA-384 和 AEAD GCM	AES (SHA384)	是	192 位元
全部	ECDHE_RSA_AES_128_GCM_SHA256 第 351 頁的『13』	C02F	TLS 1.2	SHA-256 和 AEAD GCM	AES (128)	是	否
全部	ECDHE_RSA_AES_256_GCM_SHA384 第 351 頁的『12』 第 351 頁的『13』	C030	TLS 1.2	AEAD AES-128 GCM	AES (SHA384)	是	否

附註:

1. 如需每個平台圖示所涵蓋的平台清單，請參閱產品說明文件中的版次及平台圖示。
2. 指定 CipherSpec 是否在 FIPS 認證的平台上經過 FIPS 認證。 如需 FIPS 的說明，請參閱聯邦資訊存取安全標準 (FIPS)。
3. ALW ANY_TLS13_OR_HIGHER 別名 CipherSpec 協議遠端將容許但僅使用 TLS 1.3 或更高版本通訊協定進行連線的最高階安全。
4. z/OS 若要使用 TLS 1.3 或 ANY CipherSpec ，則在 IBM MQ for z/OS 上，作業系統必須是 z/OS 2.4 或更新版本。
5. IBM i 若要使用 TLS 1.3 或 ANY CipherSpec，在 IBM i 基礎作業系統版本上，必須支援 TLS 1.3。 請參閱 TLSv1.3 的系統 TLS 支援 以取得相關資訊。
6. ALW ANY_TLS13 別名 CipherSpec 表示使用 TLS 1.3 通訊協定的可接受 CipherSpec 子集，如下表中針對每個平台列出的 CipherSpec。
7. ALW ANY_TLS12_OR_HIGHER 別名 CipherSpec 協議遠端將容許但僅使用 TLS 1.2 或更高版本通訊協定進行連線的最高階安全。
8. ANY_TLS12 CipherSpec 表示使用 TLS 1.2 通訊協定的可接受 CipherSpec 子集，如下表中針對每個平台列出的 CipherSpec。
9. ALW ANY 別名 CipherSpec 協議遠端將容許的最高階安全。
10. IBM i 在「系統值 QSSLCSLCTL」設為 *OPSSYS 的 IBM i 7.4 系統上，未啟用這些 CipherSpecs。
11. ALW 這些 CipherSpec 使用 8 個八位元組完整性檢查值 (ICV)，而不是 16 個八位元組 ICV。
12. 除非適當的未限定原則檔套用至「瀏覽器」所使用的 JRE，否則無法使用這個 CipherSpec 來保護從 IBM MQ Explorer 到佇列管理程式的連線。
13. Windows Linux 遵循 GSKit 的建議， TLS 1.2 GCM CipherSpecs 有一項限制，即在使用相同階段作業金鑰傳送 24.5 TLS 記錄之後，連線會終止，並顯示訊息 AMQ9288E。 不論使用的 FIPS 模式為何，這項 GCM 限制都在作用中。

    若要防止發生此錯誤，請避免使用 TLS 1.2 GCM 密碼、啟用秘密金鑰重設，或在設定環境變數 GSK_ENFORCE_GCM_RESTRICTION=GSK_FALSE 的情況下啟動 IBM MQ 佇列管理程式或用戶端。 對於 GSKit 程式庫，您必須在連線兩端設定此環境變數，並將它套用至用戶端至佇列管理程式連線及佇列管理程式至佇列管理程式連線。 請注意，此設定會影響未受管理的 .NET 用戶端，但不會影響 Java 或受管理 .NET 用戶端。 如需相關資訊，請參閱 AES-GCM 密碼限制。

    此限制不適用於 IBM MQ for z/OS。

在 IBM MQ 中使用 TLS 1.3

從 IBM MQ 9.2.0 開始，產品在所有平台上都支援 TLS 1.3 。 在 IBM MQ 9.2.0 之前，在 AIX, Linux, and Windows for Continuous Delivery 上已從 IBM MQ 9.1.4 提供 TLS 1.3 支援。

依預設，在 IBM MQ 9.2.0 或更新版本建立的佇列管理程式支援 TLS 1.3 。 從舊版 IBM MQ 移轉的佇列管理程式需要啟用 TLS 1.3 。 您可以透過設定 **AllowTLSV13**=TRUE 內容，在已移轉的佇列管理程式上啟用 TLS 1.3 :

- Multi 若為 IBM MQ for Multiplatforms 佇列管理程式，請編輯 qm.ini 檔案，並在 SSL 段落 (鏈結至) 下新增 **AllowTLSV13**=TRUE 內容

```
SSL:
  AllowTLSV13=TRUE
```

- z/OS 若為 IBM MQ for z/OS 佇列管理程式，請編輯佇列管理程式啟動 JCL 中指定的 QMINI 資料集 ，並在 TransportSecurity 段落下新增 **AllowTLSV13**=TRUE 內容

```
TransportSecurity:
    AllowTLSV13=TRUE
```

啟用 TLS 1.3 後，根據 TLS 1.3 規範，任何與弱 CipherSpec,無論是否在 IBM MQ 中啟用，都會被拒絕。 TLS 1.3 認為弱的 CipherSpecs 是符合下列一或多個準則的 CipherSpecs :

- 使用 SSL 3.0 通訊協定。
- 使用 RC4 或 RC2 作為「加密」演算法。
- 具有等於或小於 112 的加密金鑰大小 (位元)。

在 已淘汰的 CipherSpecs 表格 1 中，會以附註 [3] 標示這些限制。

如果您需要繼續使用此類 CipherSpecs,則必須停用 TLS 1.3 模式:

- ALW 編輯佇列管理程式的 qm.ini 檔案，並將 **AllowTLSV13** 內容的設定變更為:

```
SSL:
  AllowTLSV13=FALSE
```

- z/OS V 9.2.0 V 9.2.0 編輯佇列管理程式的 QMINI 資料集 ，並將 **AllowTLSV13** 內容的設定變更為:

```
TransportSecurity:
    AllowTLSV13=FALSE
```

IBM MQ MQI client 及 TLS 1.3

ALW V 9.2.0

使用 IBM MQ MQI client 時，除非在應用程式所使用 mqclient.ini 檔案的 SSL 段落中明確指定 **AllowTLSV13** 的值，否則會推斷該值。

- 如果啟用任何弱 CipherSpecs ，則 **AllowTLSV13** 會設為 FALSE 且無法使用 TLS 1.3 CipherSpecs 。
- 否則， **AllowTLSV13** 會設為 TRUE ，並且可以使用新的 TLS 1.3 CipherSpecs 及別名 CipherSpecs 。

在 IBM MQ 中啟用預設 CipherSpec 值

在新 IBM MQ 佇列管理程式的預設配置中， IBM MQ 提供對 TLS 1.2 及 TLS 1.3 通訊協定的支援，以及使用 CipherSpecs 的各種加密演算法。 基於相容性目的， IBM MQ 也可以配置成使用 SSL 3.0 和 TLS 1.0 通訊協定，以及一些已知很弱或容易受到安全漏洞影響的加密演算法。 在預設配置中啟用的 CipherSpecs 清單可能會透過套用維護來變更。

可以使用下列控制項來配置 IBM MQ 以限制或允許使用 CipherSpecs :

- 僅允許符合 FIPS 140-2 標準的 CipherSpecs 使用 SSLFIPS。

- ALW 僅允許使用 SUITEB 的 NSA Suite B 相容 CipherSpecs 。
- Multi 允許使用 **AllowedCipherSpecs** 的 CipherSpecs 自訂清單。
- ALW 使用 **AMQ_ALLOWED_CIPHERS** 環境變數允許自訂 CipherSpecs 清單。
- ALW 允許使用 **AllowWeakCipher** 或 **AMQ_SSL_WEAK_CIPHER_ENABLE** 環境變數來使用已淘汰的 CipherSpecs 。
- z/OS 允許在 CHINIT JCL 中使用 DD 陳述式來使用已淘汰的 CipherSpecs 。

註: 如果您使用 **AllowedCipherSpecs** 或 **AMQ_ALLOWED_CIPHERS** 來指定 CipherSpecs 的自訂清單，則會置換任何已淘汰 CipherSpecs 的啟用。 請注意，當搭配使用 NSA Suite B 或 FIPS 140-2 限制與自訂 CipherSpec 清單時，您必須確保自訂清單僅包含 Suite B 或 FIPS 140-2 設定允許的 CipherSpecs 。

相關概念

第 37 頁的『IBM MQ 中的數位憑證及 CipherSpec 相容性』
本主題提供如何透過概述 CipherSpecs 與 IBM MQ 中數位憑證之間的關係，為安全原則選擇適當的 CipherSpecs 及數位憑證的相關資訊。

第 16 頁的『CipherSpecs 和 CipherSuites』
加密安全通訊協定必須同意安全連線所使用的演算法。 CipherSpecs 和 CipherSuites 定義演算法的特定組合。

第 35 頁的『為套組 B 配置 IBM MQ』
IBM MQ 可以配置為符合 AIX, Linux, and Windows 平台上的 NSA Suite B 標準。

第 27 頁的『聯邦資訊存取安全標準 (FIPS)』
本主題介紹 US National Institute of Standards and Technology 的 Federal Information Processing Standards (FIPS) Cryptomodule Validation Program ， 以及可在 TLS 通道上使用的加密函數。

相關工作

移轉現有安全配置以使用別名 CipherSpe

相關參考

定義通道

ALTER CHANNEL

變更、複製及建立通道

ALW AES-GCM 密碼限制

用於 TLS 加密法時對 AESGCM 密碼強制的限制手冊。 這些限制是由 IETF 及 NIST 組織所強制執行，並且要求在使用 AES-GCM 密碼時，不得使用相同的階段作業金鑰來安全地傳送超過 2 筆 $^{24.5}$ TLS 記錄。

如需這些限制的相關資訊，請參閱 RFC 9325 小節 4.4 金鑰使用限制 及 RFC 8446 小節 5.5。

IBM MQ 不會直接實作加密功能。 相反地，會使用數個不同的加密程式庫來提供 TLS 及 Advanced Message Security 功能。 在 Windows、 Linux 及 AIX 作業系統上， IBM MQ 使用的加密程式庫是 IBM Global Security Kit (GSKit)。 對於應用程式， C 及未受管理的 .NET 程式庫會使用 GSKit 進行加密功能。 GSKit 實作 AES-GCM 加密演算法包括標準群組指定的限制。 此外，依預設會啟用這些限制。 因此，當使用 AES-GCM 密碼時，如果使用相同階段作業金鑰傳輸超過 2 筆 $^{24.5}$ TLS 記錄，則會終止 IBM MQ TLS 通訊。

註: 此限制不存在於 IBM i、 IBM Z 或 IBM MQ for HPE NonStop 平台或 Java/JMS 受管理 .NET 應用程式上，因為使用不同的加密程式庫，且這些程式庫未實作相同的限制。

如果 IBM MQ 通道保持執行的時間足夠長，而使用相同的階段作業金鑰傳輸超過 2 筆 $^{24.5}$ TLS 記錄，則基礎加密程式庫會終止連線。 這會導致通道終止，並產生 AMQ9288E 錯誤訊息。 以這種方式終止其通訊的應用程式會從執行中的任何 IBM MQ 作業收到 MQRC_CONNECTION_BROKEN 回覆碼。

可以在通訊的任一端執行連線終止，但只能在使用 GSKit 進行加密功能的端頭上執行。

減輕限制的建議

如何防止或處理由於此限制而終止的通訊的部分選項如下:

使用可重新連接的用戶端

應用程式可以配置成在連線失敗時自動嘗試重新連線。 這包括由於 GCM 限制而終止的連線。 配置為重新連線時，用戶端應用程式會在任何失敗點自動還原，且會還原任何開啟物件的控點。 這會在不回到應用程式碼的情況下完成。

如需相關資訊，請參閱 自動用戶端重新連線。

設定秘密金鑰重設值

IBM MQ 可以配置為在透過通道傳送可配置的位元組數之後，要求階段作業金鑰重設。 達到此限制時，IBM MQ 會要求加密層執行階段作業金鑰重設，以產生新的階段作業金鑰。

請務必注意，指定的值是傳送的位元組數，與 IBM MQ 所傳送訊息的大小相關。 此限制是針對傳送的 TLS 記錄數目。 訊息位元組與 TLS 記錄之間沒有直接對映，因為 TLS 記錄可以根據網路的「最大傳輸單位 (MTU)」來傳送位元組數上限。 所傳送的任何大於此值的訊息都會作為多個 TLS 記錄進行傳輸。 MTU 值會因網路而異。 此外，還有其他原因可能需要在傳輸 IBM MQ 訊息資料之外傳送 TLS 記錄，例如 IBM MQ 活動訊號檢查、TLS 警示、其他 IBM MQ 通訊協定訊息。 這些額外 TLS 記錄會計入 TLS 記錄數目上限，但不會計入 IBM MQ 秘密金鑰重設值中。

使用秘密金鑰重設定期重設階段作業金鑰可以防止通道因 AES-GCM 限制而終止。

如需相關資訊，請參閱 重設 SSL 和 TLS 秘密金鑰。

V 9.2.0 **使用 TLS 1.3 密碼規格**

使用 TLS 1.3 通訊協定時， AES-GCM 限制仍然存在， TLS 1.3 通訊協定支援自動執行階段作業金鑰重設，而不需要岔斷 TLS 通訊。 這可讓 GSKit 在必要時管理重設階段作業金鑰，而 IBM MQ 不需要要求重設秘密金鑰。

如需相關資訊，請參閱 第 348 頁的『啟用 CipherSpecs』中的 在 IBM MQ 中使用 TLS 1.3 。

停用 AES-GCM 限制

必要的話，可以設定環境變數 **GSK_ENFORCE_GCM_RESTRICTION=GSK_FALSE** 來停用 AES-GCM 限制，以停用此限制。 這樣做可容許使用相同的階段作業金鑰來傳送任意數目的 TLS 記錄。 如果選擇此降低，則必須在使用 GSKit 進行安全通訊的通訊的每一端設定環境變數。

警告: 不建議使用此選項，因為在傳送超過 2 筆 $^{24.5}$ TLS 記錄之後，攻擊者可能會對已傳送的記錄執行分析，以判斷使用中的階段作業金鑰。 一旦確定階段作業金鑰，使用該階段作業金鑰的所有現有及未來通訊都會受損。

V 9.2.0 V 9.2.0 TLS 信號交換中的 CipherSpec 順序

在多個可能的 CipherSpecs,例如使用 ANY* CipherSpecs 之一時，將使用 CipherSpecs 的順序。

在 TLS 信號交換期間，用戶端和伺服器會依其喜好設定來交換它們所支援的 CipherSpecs 和通訊協定。 會選擇雙方設定優先順序的一般 CipherSpec ，並用於 TLS 通訊。 在選擇 CipherSpec 通訊協定時，也會考量版本，例如，如果伺服器在 TLS 1.3 CipherSpecs 之前列出 TLS 1.2 CipherSpecs ，則只要用戶端可以支援 TLS 1.3 並具有可使用的一般 TLS 1.3 CipherSpec ，則它仍會設定 TLS 1.3 的優先順序。

從 IBM MQ 9.2.0 開始，當針對 TLS 配置 IBM MQ 時，它會將 CipherSpecs 設為下表所示的順序 (從最偏好到最不偏好的順序)。

註: 如果未透過 **AllowedCipherSpecs** 屬性啟用 CipherSpec ，則不會配置在 TLS 信號交換期間使用它。

如果未指定 **AllowedCipherSpecs** 屬性，則會使用下表指出的已啟用密碼預設清單。

表 *75: CipherSpecs* 訂單，從 *IBM MQ 9.2.0*

平台	CipherSpec	通訊協定	十六進位碼	依預設啟用
全部	TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	是
全部	TLS_AES_256_GCM_SHA384	TLS 1.3	1302	是

表 75: CipherSpecs 訂單，從 IBM MQ 9.2.0 (繼續)				
平台	**CipherSpec**	**通訊協定**	**十六進位碼**	**依預設啟用**
全部	TLS_AES_128_GCM_SHA256	TLS 1.3	1301	是
ALW	TLS_AES_128_CCM_SHA256	TLS 1.3	1304	是
ALW	TLS_AES_128_CCM_8_SHA256	TLS 1.3	1305	是
全部	TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	是
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	C02C	是
全部	ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	是
全部	TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	是
全部	ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	是
全部	ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	是
全部	TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	是
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	C02B	是
全部	ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	是
全部	TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	是
全部	ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	是
全部	ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	是
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	C008	否
Multi	ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	C012	否

表 75: CipherSpecs 訂單，從 IBM MQ 9.2.0 (繼續)				
平台	CipherSpec	通訊協定	十六進位碼	依預設啟用
ALW	TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	0005	否
ALW	ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	C007	否
Multi	ECDHE_RSA_RC4_128_SHA256	TLS 1.2	C011	否
全部	TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	否
ALW	ECDHE_ECDSA_NULL_SHA256	TLS 1.2	C006	否
Multi	ECDHE_RSA_NULL_SHA256	TLS 1.2	C010	否
ALW	TLS_RSA_WITH_NULL_NULL	TLS 1.2	0000	否
ALW z/OS	TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	否
ALW z/OS	TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	否
IBM i	AES_SHA_US	TLS 1.0	002E	否
全部	TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	否
全部	TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	否
IBM i	TLS_RSA_WITH_RC4_128_MD5	TLS 1.0	0004	否
全部	TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	否
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5	TLS 1.0	0003	否
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5	TLS 1.0	0006	否
IBM i	TLS_RSA_WITH_NULL_SHA	TLS 1.0	0002	否
IBM i	TLS_RSA_WITH_NULL_MD5	TLS 1.0	0001	否

表 75: CipherSpecs 訂單，從 IBM MQ 9.2.0 (繼續)				
平台	**CipherSpec**	**通訊協定**	**十六進位碼**	**依預設啟用**
全部	TRIPLE_DES_SHA_US	SSL 第 3 版	000A	否
全部	RC4_SHA_US	SSL 第 3 版	0005	否
全部	RC4_MD5_US	SSL 第 3 版	0004	否
全部	DES_SHA_EXPORT	SSL 第 3 版	0009	否
全部	RC4_MD5_EXPORT	SSL 第 3 版	0003	否
全部	RC2_MD5_EXPORT	SSL 第 3 版	0006	否
全部	NULL_SHA	SSL 第 3 版	0002	否
全部	NULL_MD5	SSL 第 3 版	0001	否
ALW	FIPS_WITH_3DES_EDE_CBC_SHA	SSL 第 3 版	FEFF	否
ALW	RC4_56_SHA_EXPORT1024	SSL 第 3 版	0064	否
ALW	DES_SHA_EXPORT1024	SSL 第 3 版	0062	否
ALW	FIPS_WITH_DES_CBC_SHA	SSL 第 3 版	FEFE	否

此清單是透過使用 z/OS 上 IBM MQ 所使用的加密程式庫所提供的預設清單來排序通訊協定而建構的，而且在 z/OS 及分散式平台之間是一致的。

變更順序

如果需要不同的順序，則可以使用具有下列規則之 IBM MQ for Multiplatforms z/OS，或 IBM MQ for z/OS 上的 TransportSecurity 段落， 上 SSL 段落的 **AllowedCipherSpecs** 屬性來提供 CipherSpecs 的新順序:

- 不論它們在清單中的位置為何，一律會使用較高的通訊協定版本。
- 任何已停用的 CipherSpecs 都會重新啟用 (如果在清單中提供的話)。
- TLS 伺服器的清單順序比 TLS 用戶端具有更高的優先順序。
- 啟用 TLS 1.3 時，不支援某些 CipherSpecs 。

例如，在 IBM MQ for Multiplatforms 上，如果在佇列管理程式上配置下列項目:

```
SSL:
AllowedCipherSpecs=TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_AES_128_GCM_SHA256,TLS_AES_256_GCM_SHA384
,TLS_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_CBC_SHA
```

z/OS 及在 IBM MQ for z/OS 上，如果已在佇列管理程式上配置下列項目:

```
TransportSecurity:
AllowedCipherSpecs=TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_AES_128_GCM_SHA256,TLS_AES_256_GCM_SHA384
,TLS_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_CBC_SHA
```

然後:

- 使用 ANY_TLS12 連接的用戶端可能會使用 TLS 1.2 CipherSpec TLS_RSA_WITH_AES_128_GCM_SHA256。

- 使用 ANY_TLS12_OR_HIGHER 連接的用戶端可能會使用 TLS 1.3 CipherSpec TLS_AES_128_GCM_SHA256 (假設用戶端支援 TLS 1.3)。
- 使用 TLS 1.0 CipherSpec TLS_RSA_WITH_AES_256_CBC_SHA 連接的用戶端將會使用該 CipherSpec。

舊版 IBM MQ

在 IBM MQ 9.2.0 之前，使用 CipherSpecs 的下列順序:

表 *76: CipherSpecs* 在 *IBM MQ 9.2.0* 之前訂購

平台	CipherSpec	通訊協定	依預設啟用
ALW z/OS	TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	否
IBM i	AES_SHA_US	TLS 1.0	否
ALW z/OS	TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	否
全部	RC4_SHA_US	SSL 第 3 版	否
全部	TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	否
全部	RC4_MD5_US	SSL 第 3 版	否
IBM i	TLS_RSA_WITH_RC4_128_MD5	TLS 1.0	否
全部	TRIPLE_DES_SHA_US	SSL 第 3 版	否
全部	TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	否
ALW	DES_SHA_EXPORT1024	SSL 第 3 版	否
全部	RC4_56_SHA_EXPORT1024	SSL 第 3 版	否
全部	RC4_MD5_EXPORT	SSL 第 3 版	否
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5	TLS 1.0	否
全部	RC2_MD5_EXPORT	SSL 第 3 版	否
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5	TLS 1.0	否
全部	DES_SHA_EXPORT	SSL 第 3 版	否
全部	TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	否
全部	NULL_SHA	SSL 第 3 版	否
IBM i	TLS_RSA_WITH_NULL_SHA	TLS 1.0	否

表 76: CipherSpecs 在 IBM MQ 9.2.0 之前訂購 (繼續)			
平台	**CipherSpec**	**通訊協定**	**依預設啟用**
全部	NULL_MD5	SSL 第 3 版	否
IBM i	TLS_RSA_WITH_NULL_MD5	TLS 1.0	否
ALW	FIPS_WITH_DES_CBC_SHA	SSL 第 3 版	否
ALW	FIPS_WITH_3DES_EDE_CBC_SHA	SSL 第 3 版	否
全部	TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	是
全部	TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	是
全部	TLS_RSA_WITH_NULL_SHA256	TLS 1.2	否
全部	TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	是
全部	TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	是
ALW	ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	否
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	否
Multi	ECDHE_RSA_RC4_128_SHA256	TLS 1.2	否
Multi	ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	否
全部	ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	是
全部	ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	是
全部	ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	是
全部	ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	是
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	是
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	是
全部	ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	是
全部	ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	是

表 76: CipherSpecs 在 IBM MQ 9.2.0 之前訂購 (繼續)			
平台	**CipherSpec**	**通訊協定**	**依預設啟用**
Multi	ECDHE_RSA_NULL_SHA256	TLS 1.2	否
ALW	ECDHE_ECDSA_NULL_SHA256	TLS 1.2	否
ALW	TLS_RSA_WITH_NULL_NULL	TLS 1.2	否
ALW	TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	否
Multi	TLS_AES_128_GCM_SHA256	TLS 1.3	是
Multi	TLS_AES_256_GCM_SHA384	TLS 1.3	是
Multi	TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	是
ALW	TLS_AES_128_CCM_SHA256	TLS 1.3	是
ALW	TLS_AES_128_CCM_8_SHA256	TLS 1.3	是

重要: 截至 2020 年 7 月 23rd，以下 AllowedCipherSpecs 屬性僅啟用目前預設啟用的 CipherSpecs 。 但是，您應該使用目前資料驗證下列 AllowedCipherSpecs 屬性啟用的 CipherSpecs ，以確保自該日期以來已棄用的 CipherSpecs 不會無意中重新啟用。

如果您需要返回此 CipherSpecs,可以使用下列 **AllowedCipherSpecs** SSL/TransportSecurity 節屬性值來執行此操作:

```
AllowedCipherSpecs=TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_256_CBC_SHA256,TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_RSA_WITH_AES_256_GCM_SHA384,ECDHE_ECDSA_AES_128_CBC_SHA256,ECDHE_ECDSA_AES_256_CBC_SHA384,ECDHE_RSA_AES_128_CBC_SHA256,ECDHE_RSA_AES_256_CBC_SHA384,ECDHE_ECDSA_AES_128_GCM_SHA256,ECDHE_ECDSA_AES_256_GCM_SHA384,ECDHE_RSA_AES_128_GCM_SHA256,ECDHE_RSA_AES_256_GCM_SHA384
```

在 IBM MQ for Multiplatforms 上提供已訂購及已啟用 CipherSpecs 的自訂清單

Multi

您可以使用 ALW **AMQ_ALLOWED_CIPHERS** 環境變數 或 `.ini` 檔案的 **AllowedCipherSpecs** SSL 段落屬性，以您的喜好設定順序提供已啟用的替代 CipherSpecs 集，以與 IBM MQ 通道搭配使用。 基於下列任一原因，您可能想要使用此設定:

- 限制 IBM MQ 接聽器接受送入通道啟動要求，除非它們使用其中一個具名 CipherSpecs。
- 變更 TLS 信號交換中所使用 CipherSpecs 的優先順序。

此功能可用來控制 ANY* CipherSpecs 中包含的 CipherSpecs 。

AMQ_ALLOWED_CIPHERS 環境變數或 **AllowedCipherSpecs** SSL 段落屬性接受:

- 單一 CipherSpec 名稱。
- 要重新啟用的 CipherSpec 名稱清單 (以逗點區隔)。
- ALL 的特殊值，代表所有 CipherSpecs。

註: 您不應啟用 **ALL** CipherSpecs,因為這將啟用 SSL 3.0 和 TLS 1.0 協定以及大量弱加密演算法。

如果已配置此設定，它會置換預設 CipherSpec 清單，並導致 IBM MQ 忽略低保護性密碼淘汰設定 (請參閱下面的說明):

- IBM MQ 接聽器只接受使用其中一個具名 CipherSpecs 的 SSL/TLS 提案。
- IBM MQ 通道僅容許空白 SSLCIPH 值，或其中一個指定的 CipherSpecs。
- **runmqsc** 標籤完成 SSLCIPH 值會將完成值限制為其中一個名稱 CipherSpecs。

例如，如果您只想要容許定義/變更通道，且接聽器接受 ECDHE_RSA_AES_128_GCM_SHA256 或 ECDHE_ECDSA_AES_256_GCM_SHA384 ，則可以在 qm.ini 檔案中設定下列:

```
SSL:
AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256, ECDHE_ECDSA_AES_256_GCM_SHA384
```

此外，此清單中的 CipherSpecs 將用來決定在 TLS 信號交換期間所使用 CipherSpecs 的優先順序。 例如，如果您指定 TLS_RSA_WITH_AES_128_CBC_SHA256， TLS_RSA_WITH_AES_256_CBC_SHA256 列表，則在握手期間，如果客戶端連線指定這兩個 CipherSpecs,即使用 ANY_TLS12 連接的客戶端），則可能會選擇 TLS_RSA_WITH_AES_128_CBC_SHA256 CipherSpec 是 TLS_RSA_WITH_AES_128_CBC_SHA256 CipherSpec 是 TLS_RSA_WITH_AES_256_CBC_SHA256

請注意，使用 java.security 檔案設定可以限制 AMQP 或 MQTT 通道所使用的密碼。

在 IBM MQ for z/OS 上提供已訂購及已啟用 CipherSpecs 的自訂清單

z/OS

您可以使用 QMINI 資料集的 **AllowedCipherSpecs** TransportSecurity 段落屬性，提供已啟用且依喜好設定順序與 IBM MQ 通道搭配使用的替代 CipherSpecs 集。 基於下列任一原因，您可能想要執行此動作:

- 限制 IBM MQ 接聽器接受送入通道啟動要求，除非它們使用其中一個具名 CipherSpecs。
- 變更 TLS 信號交換中所使用 CipherSpecs 的優先順序。

您可以使用此功能來控制 ANY* CipherSpecs 中包含的 CipherSpecs 。 **AllowedCipherSpecs** 屬性接受:

- 單一 CipherSpec 名稱。
- 要重新啟用的 CipherSpec 名稱清單 (以逗點區隔)。
- ALL 的特殊值，代表所有 CipherSpecs。

  註: 您不應啟用 **ALL** CipherSpecs,因為這將啟用 SSL 3.0 和 TLS 1.0 協定以及大量弱加密演算法。 如果您配置此設定，它會置換預設 CipherSpec 清單，並導致 IBM MQ 忽略低保護性密碼淘汰設定; 請參閱 第 365 頁的『在 z/OS 上啟用已淘汰的 CipherSpecs』。

IBM MQ 接聽器只接受使用其中一個具名 CipherSpecs 及 IBM MQ 通道的 SSLCIPH 值，或其中一個具名 CipherSpecs 的 SSL/TLS 提案。

例如，如果您只想要容許定義/變更通道，且接聽器接受 ECDHE_RSA_AES_128_GCM_SHA256 或 ECDHE_RSA_AES_256_GCM_SHA384 ，則可以設定下列:

```
TransportSecurity:
AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256,
                   ECDHE_RSA_AES_256_GCM_SHA384
```

此外，此清單中的 CipherSpecs 用來決定在 TLS 信號交換期間所使用 CipherSpecs 的優先順序。 例如，如果您指定 TLS_RSA_WITH_AES_128_CBC_SHA256 、 TLS_RSA_WITH_AES_256_CBC_SHA256 列表，則在握手期間 CipherSpec 如果客戶端 CipherSpec 同時指定這兩個 TLS_RSA_WITH_AES_256_CBC_SHA256CipherSpecs,則可能會選擇 TLS_RSA_WITH_AES_128_CBC_SHA256 ，即使用 ANY_TLS12 連接的客戶端。

已淘汰 CipherSpecs

必要的話，您可以與 IBM MQ 搭配使用的已淘汰 CipherSpecs 清單。

註: 在 AIX, Linux, and Windows 上， IBM MQ 透過 IBM Crypto for C (ICC) 加密模組提供 FIPS 140-2 相符性。 此模組的憑證已移至「歷程」狀態。 客戶應該檢視 IBM Crypto for C (ICC) 憑證 ， 並注意 NIST 提供的

任何建議。 目前正在進行取代 FIPS 140-3 模組，您可以在 處理程序清單中的 NIST CMVP 模組中搜尋它，以檢視其狀態。

有關啟用已棄用的 CipherSpecs,請參閱第 364 頁的『在 IBM MQ for Multiplatforms 上啟用已淘汰的 CipherSpecs』 或第 365 頁的『在 z/OS 上啟用已淘汰的 CipherSpecs』 。

下表列出您可以與 IBM MQ TLS 支援搭配使用的已淘汰 CipherSpecs 。

表 77: 可以重新啟用已淘汰 *CipherSpec* 與 *IBM MQ* 搭配使用

平台支援 第 364 頁的『1』	CipherSpec 名稱	十六進位碼	使用的通訊協定	資料完整性	加密演算法 (加密位元)	FIPS 第 364 頁的『2』	套組 B	淘汰時更新
適用於 SSL 3.0 的 CipherSpec								
IBM i	AES_SHA_US 第 364 頁的『3』	002F	SSL 3.0	SHA-1	AES (128)	否	否	9.0.0.0
全部	DES_SHA_EXPORT 第 364 頁的『3』 第 364 頁的『4』 第 364 頁的『5』	0009	SSL 3.0	SHA-1	DES (56)	否	否	9.0.0.0
ALW	DES_SHA_EXPORT1024 第 364 頁的『3』 第 364 頁的『6』	0062	SSL 3.0	SHA-1	DES (56)	否	否	9.0.0.0
ALW	FIPS_WITH_DES_CBC_SHA 第 364 頁的『3』	FEFE	SSL 3.0	SHA-1	DES (56)	否 第 364 頁的『7』	否	9.0.0.0
ALW	FIPS_WITH_3DES_EDE_CBC_SHA 第 364 頁的『3』	FEFF	SSL 3.0	SHA-1	3DES (168)	否 第 364 頁的『8』	否	9.0.0.1 及 9.0.1
全部	NULL_MD5 第 364 頁的『3』	0001	SSL 3.0	MD5	無	否	否	9.0.0.1
全部	NULL_SHA 第 364 頁的『3』	0002	SSL 3.0	SHA-1	無	否	否	9.0.0.1
全部	RC2_MD5_EXPORT 第 364 頁的『3』 第 364 頁的『4』 第 364 頁的『5』	0006	SSL 3.0	MD5	RC2 (40)	否	否	9.0.0.0
全部	RC4_MD5_EXPORT 第 364 頁的『4』 第 364 頁的『3』	0003	SSL 3.0	MD5	RC4 (40)	否	否	9.0.0.0
全部	RC4_MD5_US 第 364 頁的『3』	0004	SSL 3.0	MD5	RC4 (128)	否	否	9.0.0.0
全部	RC4_SHA_US 第 364 頁的『3』 第 364 頁的『5』	0005	SSL 3.0	SHA-1	RC4 (128)	否	否	9.0.0.0
ALW	RC4_56_SHA_EXPORT1024 第 364 頁的『3』 第 364 頁的『6』	0064	SSL 3.0	SHA-1	RC4 (56)	否	否	9.0.0.0
全部	TRIPLE_DES_SHA_US 第 364 頁的『3』 第 364 頁的『5』	000A	SSL 3.0	SHA-1	3DES (168)	否	否	9.0.0.1 及 9.0.1
適用於 TLS 1.0 的 CipherSpec								
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5 第 364 頁的『3』	0006	TLS 1.0	MD5	RC2 (40)	否	否	9.0.0.0
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5 第 364 頁的『3』 第 364 頁的『4』	0003	TLS 1.0	MD5	RC4 (40)	否	否	9.0.0.0

表 77: 可以重新啟用已淘汰 *CipherSpec* 與 *IBM MQ* 搭配使用 (繼續)								
平台支援 第 364 頁的『1』	**CipherSpec 名稱**	**十六進位碼**	**使用的通訊協定**	**資料完整性**	**加密演算法 (加密位元)**	**FIPS** 第 364 頁的『2』	**套組 B**	**淘汰時更新**
全部	TLS_RSA_WITH_DES_CBC_SHA 第 364 頁的『3』	0009	TLS 1.0	SHA-1	DES (56)	否 第 364 頁的『9』	否	9.0.0.0
IBM i	TLS_RSA_WITH_NULL_MD5 第 364 頁的『3』	0001	TLS 1.0	MD5	無	否	否	9.0.0.1
IBM i	TLS_RSA_WITH_NULL_SHA 第 364 頁的『3』	0002	TLS 1.0	SHA-1	無	否	否	9.0.0.1
IBM i	TLS_RSA_WITH_RC4_128_MD5 第 364 頁的『3』	0004	TLS 1.0	MD5	RC4 (128)	否	否	9.0.0.0
ALW z/OS	TLS_RSA_WITH_AES_128_CBC_SHA 第 364 頁的『10』	002F	TLS 1.0	SHA-1	AES (128)	是	否	9.0.5
ALW z/OS	TLS_RSA_WITH_AES_256_CBC_SHA 第 364 頁的『6』 第 364 頁的『10』	0035	TLS 1.0	SHA-1	AES (256)	是	否	9.0.5
全部	TLS_RSA_WITH_3DES_EDE_CBC_SHA	000A	TLS 1.0	SHA-1	3DES (168)	是	否	9.0.0.1 及 9.0.1
適用於 TLS 1.2 的 CipherSpec								
ALW	ECDHE_ECDSA_NULL_SHA256 第 364 頁的『3』	C006	TLS 1.2	SHA-1	無	否	否	9.0.0.1
ALW	ECDHE_ECDSA_RC4_128_SHA256 第 364 頁的『3』	C007	TLS 1.2	SHA-1	RC4 (128)	否	否	9.0.0.0
ALW IBM i	ECDHE_RSA_NULL_SHA256 第 364 頁的『3』	C010	TLS 1.2	SHA-1	無	否	否	9.0.0.1
ALW IBM i	ECDHE_RSA_RC4_128_SHA256 第 364 頁的『3』	C011	TLS 1.2	SHA-1	RC4 (128)	否	否	9.0.0.0
ALW	TLS_RSA_WITH_NULL_NULL 第 364 頁的『3』	0000	TLS 1.2	無	無	否	否	9.0.0.1
全部	TLS_RSA_WITH_NULL_SHA256 第 364 頁的『3』	003B	TLS 1.2	SHA-256	無	否	否	9.0.0.1
ALW	TLS_RSA_WITH_RC4_128_SHA256 第 364 頁的『3』	0005	TLS 1.2	SHA-1	RC4 (128)	否	否	9.0.0.0
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	C0008	TLS 1.2	SHA-1	3DES (168)	是	否	9.0.0.1 及 9.0.1
ALW IBM i	ECDHE_RSA_3DES_EDE_CBC_SHA256	C012	TLS 1.2	SHA-1	3DES (168)	是	否	9.0.0.1 及 9.0.1

表 77: 可以重新啟用已淘汰 *CipherSpec* 與 *IBM MQ* 搭配使用 (繼續)

平台支援 第 364 頁的『1』	CipherSpec 名稱	十六進位碼	使用的通訊協定	資料完整性	加密演算法 (加密位元)	FIPS 第 364 頁的『2』	套組 B	淘汰時更新

附註:

1. 如需每個平台圖示所涵蓋的平台清單，請參閱產品說明文件中的版次及平台圖示。
2. 指定 CipherSpec 是否在 FIPS 認證的平台上經過 FIPS 認證。 如需 FIPS 的說明，請參閱聯邦資訊存取安全標準 (FIPS)。
3. ALW 當啟用 TLS 1.3 時，會停用這些 CipherSpec（透過 `qm.ini` 中的 AllowTLSV13 內容）。

   z/OS 在 IBM MQ for z/OS 9.2.0 或更新版本中建立的佇列管理程式預設會啟用 TLS 1.3，這會停用這些 CipherSpec。 如果需要，可以關閉 TLS V1.3，以啟用這些 CipherSpec。 方法是將 **AllowTLSV13**=*FALSE* 新增至佇列管理程式 JCL 中 QMINI 資料集的 TransportSecurity 段落。 依預設，從舊版移轉至 IBM MQ for z/OS 9.2.0 的佇列管理程式未啟用 TLS 1.3，因此已啟用這些 CipherSpecs。
4. 信號交換金鑰大小上限是 512 位元。 如果在 SSL 信號交換期間交換的兩個憑證中有一個金鑰大小超出 512 位元，則在信號交換期間會產生一個臨時的 512 位元金鑰以供使用。
5. 這些 CipherSpec 不再受 IBM MQ classes for Java 或 IBM MQ classes for JMS 支援。 如需相關資訊，請參閱 IBM MQ classes for Java 中的 SSL/TLS CipherSpecs 及 CipherSuites 或 IBM MQ classes for JMS 中的 SSL/TLS CipherSpecs 及 CipherSuites。
6. 信號交換金鑰大小是 1024 位元。
7. 在 2007 年 5 月 19 日之前，這個 CipherSpec 已經過 FIPS 140-2 認證。 名稱 `FIPS_WITH_DES_CBC_SHA` 是歷程，反映此 CipherSpec 先前（但不再）符合 FIPS 標準的事實。 這個 CipherSpec 已淘汰，不建議使用它。
8. 名稱 `FIPS_WITH_3DES_EDE_CBC_SHA` 是歷程，反映此 CipherSpec 先前（但不再）符合 FIPS 標準的事實。 這個 CipherSpec 的用法已淘汰。
9. 在 2007 年 5 月 19 日之前，這個 CipherSpec 已經過 FIPS 140-2 認證。
10. z/OS 僅重新啟用這些 CipherSpec，就不需要使用 CSQXWEAK DD 陳述式。

在 IBM MQ for Multiplatforms 上啟用已淘汰的 CipherSpecs

Multi

依預設，不容許您在通道定義上指定已淘汰的 CipherSpec。 如果您嘗試在 IBM MQ for Multiplatforms 上指定已淘汰的 CipherSpec，則會收到訊息 AMQ8242: SSLCIPH 定義錯誤，而且 PCF 會傳回 `MQRCCF_SSL_CIPHER_SPEC_ERROR`。

您無法使用已淘汰的 CipherSpec 來啟動通道。 如果您嘗試使用已棄用的 CipherSpec,系統將會向客戶端傳回 `MQCC_FAILED (2)`以及 `MQRC_SSL_INITIALIZATION_ERROR (2393)`的 **Reason** 。

您可以透過設定環境變數 **AMQ_SSL_WEAK_CIPHER_ENABLE**，在伺服器的執行時期重新啟用一個以上已淘汰的 CipherSpecs 來定義通道。

AMQ_SSL_WEAK_CIPHER_ENABLE 環境變數接受:

- 單一 CipherSpec 名稱，或
- 要重新啟用之 CipherSpec 名稱的逗點區隔清單，或
- `ALL` 的特殊值，代表所有 CipherSpecs。

小心: 雖然 `ALL` 是有效選項，但您 **應該** 在企業需要的特定狀況下使用它，因為重新啟用 `ALL` CipherSpecs 會啟用 SSL 3.0 和 TLS 1.0 通訊協定，以及大量低保護性加密演算法。

例如，如果您要重新啟用 `ECDHE_RSA_RC4_128_SHA256`，請設定下列環境變數:

```
export AMQ_SSL_WEAK_CIPHER_ENABLE=ECDHE_RSA_RC4_128_SHA256
```

或者，透過設定下列指令來變更 qm.ini 檔案中的 SSL 段落:

```
SSL:
  AllowTLSV1=Y
  AllowWeakCipherSpec=ECDHE_RSA_RC4_128_SHA256
```

在 z/OS 上啟用已淘汰的 CipherSpecs

z/OS

依預設，不容許您在通道定義上指定已淘汰的 CipherSpec 。 如果您嘗試在 z/OS 上指定已淘汰的 CipherSpec ，則會收到訊息 CSQM102E、訊息 CSQX616E 或 CSQX674E。

如果您收到任何這些訊息，且您的企業需要重新啟用使用弱 CipherSpecs，請遵循本節中列出的指示。

小心: 在下列指示中，若要讓虛擬定義 (DD) 陳述式生效， SSLTASKS 必須是非零值。 如果這需要變更 SSLTASKS ， 您必須重新啟動通道起始程式。

在 IBM MQ for z/OS 上，控制弱或壞的 CipherSpecs 現行方法如下:

- 如果您想要重新啟用弱 CipherSpecs,可以透過將名為 CSQXWEAK 的虛擬資料定義 (DD) 語句新增至通道啟動器 JCL 來實現。 如果自行指定，則這只會啟用與 TLS 1.2 通訊協定相關聯的弱 CipherSpecs ; 例如:

  ```
  //CSQXWEAK DD DUMMY
  ```

  註: 並非所有已淘汰的 CipherSpecs 都需要使用這個 DD 陳述式，請參閱上表中的附註 10。
- 如果您想要重新啟用 SSLv3 CipherSpecs,您也可以透過在通道啟動器 JCL 新增一條名為 CSQXSSL3 的虛擬 DD 語句來實現。 所有 SSLv3 CipherSpecs 都視為 **弱**，因此您也必須指定 CSQXWEAK:

  ```
  //CSQXSSL3 DD DUMMY
  ```
- 如果您想要重新啟用已棄用的 TLS V1 CipherSpecs,可以透過在頻道啟動器 JCL 新增名為 TLS10ON 的虛擬 DD 語句（開啟 TLS V1.0 ）來實現。 如果自行指定，則會啟用與 TLS 1.0 通訊協定相關聯的強 CipherSpecs :

  ```
  //TLS10ON DD DUMMY
  ```

  如果與 CSQXWEAK 一起指定，則也會啟用與 TLS 1.0 相關聯的 **弱** CipherSpecs 。
- 如果您想要明確關閉已棄用的 TLS V1 CipherSpecs,可以透過在通道啟動器 JCL 新增一個名為 TLS10OFF 的虛擬 DD 語句（關閉 TLS V1.0 ）來實現；例如:

  ```
  //TLS10OFF DD DUMMY
  ```

如果您只想使用 **System SSL** 預設密碼規格清單中列出的密碼規格來與接聽器協議，您需要在 CHINIT JCL 中定義下列 DD 陳述式:

```
JCL: //GSKDCIPS DD DUMMY
```

重要: V 9.2.0 V 9.2.0 對於 IBM MQ for z/OS 9.2.0 以及更新版本，在通道起始程式啟動期間顯示訊息時，會將先前列出的 DD 卡和 **AllowTLSV13** 值納入考量，以指出哪些通訊協定已啟用，哪些未啟用。 因此，即使指定先前列出的其中一個 DD 卡，也可能表示由於這些設定的組合，無法以另一個通訊協定來啟用特定通訊協定。 例如，如果啟用 TLS 1.3 ，則不容許通訊協定 SSL 3.0 。

如果資料定義變更不合適，可以使用替代機制強制重新啟用弱 CipherSpecs,和 SSLv3 支援。 如需進一步資訊，請聯絡 IBM 服務中心。

相關概念

第 37 頁的『IBM MQ 中的數位憑證及 CipherSpec 相容性』

本主題提供如何透過概述 CipherSpecs 與 IBM MQ 中數位憑證之間的關係，為安全原則選擇適當的 CipherSpecs 及數位憑證的相關資訊。

相關參考
定義通道
ALTER CHANNEL

別名 CipherSpec 設定之間的關係

本資訊說明用戶端和伺服器配置中別名 CipherSpecs 不同組合的預期行為。 在這裡，客戶端是指發起通訊的實體，例如客戶端應用程式或佇列管理程式傳送端通道，而伺服器是指從客戶端接收通訊的實體，例如伺服器連線通道或接收端通道。

通訊協定下限與固定通訊協定 CipherSpecs

V 9.2.0

IBM MQ 支援兩種不同類型的 CipherSpecs:

最小通訊協定
最小通訊協定 CipherSpecs 是未設定上限的通訊協定，例如 ANY、 ANY_TLS12_OR_HIGHER。

固定通訊協定
固定通訊協定 CipherSpecs 是識別特定通訊協定 (例如 ANY_TLS12 及 ANY_TLS13) 或特定演算法 (例如 ECDHE_ECDSA_3DES_EDE_CBC_SHA256)。

從 IBM MQ 9.2.0 開始，所有平台都支援最低及固定通訊協定 CipherSpecs 。

為了在維護安全的同時盡可能簡化配置，建議在通道兩端使用 **最低通訊協定** CipherSpecs 。 當雙方都支援新版本而不需要變更任一方的配置時，這可讓您的通訊自動支援並使用更高的 TLS 通訊協定版本。

在起始端使用 **最低通訊協定** CipherSpec ，但在接收端使用 **固定通訊協定** CipherSpec 可能會導致拒絕連線，且

- Multi 正在發出訊息 AMQ9631 及 AMQ9641 。
- z/OS V 9.2.0 V 9.2.0 發出訊息 CSQX631E 及 CSQX641E 。

下表顯示不同別名 CipherSpec 設定與預期結果之間的關係。 第 366 頁的表 78 顯示在用戶端及/或伺服器上未啟用 TLS 1.3 時的預期行為。 第 367 頁的表 79 顯示同時在用戶端和伺服器上啟用 TLS 1.3 時的預期行為。 在這兩種情況下，用戶端的 CipherSpecs 會顯示在表格的 Y 軸中，伺服器的 CipherSpecs 會顯示在表格的 X 軸中。

註: 在下列表格中，標示為 可能失敗 的資料格指出當您指定 **最小通訊協定** CipherSpec (用於連線的某個部分) 及特定 (**固定通訊協定**) CipherSpec (用於另一個組件) 時可能發生衝突。

例如，假設用戶端和伺服器設定為使用 ANY CipherSpec，且伺服器通道設定為使用特定的 CipherSpec:

- 如果用戶端和伺服器支援最強的 CipherSpec 符合通道上配置的特定 CipherSpec ，則 TLS 信號交換會順利解決。
- 不過，如果用戶端和伺服器都支援更強的 CipherSpec ，則 TLS 信號交換會解析為使用它，即使它不符合通道上指定的 CipherSpec 也是如此，且 TLS 信號交換會失敗。

表 *78:* 在用戶端及/或伺服器上未啟用 *TLS 1.3* 時的預期行為

	伺服器			
用戶端	**特定 TLS 1.2 CipherSpec**	**ANY**	**ANY_ TLS12**	**ANY_TLS12_ 更高**
特定 TLS 1.2 CipherSpec	連接次數	連接次數	連接次數	連接次數
任何	可能失敗	連接次數	連接次數	連接次數
ANY_ TLS12	可能失敗	連接次數	連接次數	連接次數

表 78: 在用戶端及/或伺服器上未啟用 TLS 1.3 時的預期行為 (繼續)				
	伺服器			
用戶端	**特定 TLS 1.2 CipherSpec**	**ANY**	**ANY_ TLS12**	**ANY_TLS12_ 更高**
ANY_TLS12_ 較高值	可能失敗	連接次數	連接次數	連接次數

表 79: 同時在用戶端及伺服器上啟用 TLS 1.3 時的預期行為							
	伺服器						
用戶端	**特定 TLS 1.2 CipherSpec**	**特定 TLS 1.3 CipherSpec**	**ANY**	**ANY_TLS 12**	**ANY_TLS 13**	**ANY_TLS12_ 或更高**	**ANY_TLS13_ 或更高**
特定 TLS 1.2 CipherSpec	連接次數	**失敗**	連接次數	連接次數	**失敗**	連接次數	**失敗**
特定 TLS 1.3 CipherSpec	**失敗**	連接次數	連接次數	**失敗**	連接次數	連接次數	連接次數
任何	**失敗**	可能失敗	連接次數	**失敗**	連接次數	連接次數	連接次數
ANY_TLS12	可能失敗	**失敗**	連接次數	連接次數	**失敗**	連接次數	**失敗**
ANY_TLS13	**失敗**	可能失敗	連接次數	**失敗**	連接次數	連接次數	連接次數
ANY_TLS12_ OR_SUPERR	**失敗**	可能失敗	連接次數	**失敗**	連接次數	連接次數	連接次數
ANY_TLS13_ OR_SUPERR	**失敗**	可能失敗	連接次數	**失敗**	連接次數	連接次數	連接次數

相關概念

第 37 頁的『IBM MQ 中的數位憑證及 CipherSpec 相容性』
本主題提供如何透過概述 CipherSpecs 與 IBM MQ 中數位憑證之間的關係，為安全原則選擇適當的 CipherSpecs 及數位憑證的相關資訊。

第 16 頁的『CipherSpecs 和 CipherSuites』
加密安全通訊協定必須同意安全連線所使用的演算法。 CipherSpecs 和 CipherSuites 定義演算法的特定組合。

第 348 頁的『啟用 CipherSpecs』
在 **DEFINE CHANNEL** 或 **ALTER CHANNEL** MQSC 指令中使用 **SSLCIPH** 參數來啟用 CipherSpec 。

相關工作

移轉現有的安全配置以使用 ANY_TLS12_OR_HIGHER CipherSpec

使用 IBM MQ Explorer 取得 CipherSpecs 的相關資訊

您可以使用 IBM MQ Explorer 來顯示 CipherSpecs 的說明。

使用下列程序來取得 第 348 頁的『啟用 CipherSpecs』中 CipherSpecs 的相關資訊:

1. 開啟 IBM MQ Explorer 並展開**佇列管理程式**資料夾。
2. 請確定您已啟動佇列管理程式。
3. 選取您要使用的佇列管理程式，然後按一下 **通道**。
4. 用滑鼠右鍵按一下您要使用的通道，然後選取 **內容**。

5. 選取 **SSL** 內容頁面。
6. 從清單中選取您要使用的 CipherSpec 。 說明會顯示在清單下方的視窗中。

z/OS IBM i 用於指定 CipherSpecs 的替代方案

對於作業系統提供 TLS 支援的那些平台，您的系統可能支援 第 348 頁的『啟用 CipherSpecs』中未包含的新 CipherSpecs 。

您可以使用 SSLCIPH 參數來指定新的 CipherSpec ，但您提供的值取決於您的平台。 在所有情況下，規格都必須對應於 TLS CipherSpec ，它既有效又受系統執行的 TLS 版本支援。

註: 本節不適用於 AIX, Linux, and Windows 系統，因為 CipherSpecs 隨附於 IBM MQ 產品，因此新的 CipherSpecs 在出貨之後不會變成可用。

IBM i IBM i

代表十六進位值的兩個字元字串。

如需允許值的相關資訊，請參閱 設定安全階段作業的字元資訊的「使用注意事項」一節中的第三點。

小心: 您不應在 **SSLCIPH** 中指定十六進位密碼值，因為從將使用的密碼值中不清楚，且無法決定要使用的通訊協定選項。 使用十六進位密碼值可能會導致 CipherSpec 不符錯誤。

您可以使用 **CHGMQMCHL** 或 **CRTMQMCHL** 指令來指定值，例如:

```
CRTMQMCHL CHLNAME(' channel name ') SSLCIPH(' hexadecimal value ')
```

您也可以使用 **ALTER QMGR** MQSC 指令來設定 **SSLCIPH** 參數。

z/OS z/OS

代表十六進位值的四個字元字串。 十六進位碼對應於 TLS 通訊協定中定義的值。

如需相關資訊，請參閱 密碼組合定義 ，其中列出所有支援的 TLS 1.0、TLS 1.2 及 TLS 1.3 密碼規格，格式為 4 位數十六進位碼。

註: 為了使用弱 CipherSpec,或屬於已棄用協定的 CipherSpec ，例如 SSL V3.0 或 TLS 1.0 ，您必須在通道啟動器啟動 JCL 中指定相關的 DD 卡。 如需相關資訊，請參閱 第 361 頁的『已淘汰 CipherSpecs』。

IBM MQ 叢集的考量

使用 IBM MQ 叢集最安全的方式是在 第 348 頁的『啟用 CipherSpecs』中使用 CipherSpec 名稱。 如果您使用替代規格，請注意該規格在其他平台上可能無效。 如需相關資訊，請參閱第 402 頁的『SSL/TLS 和叢集』。

指定 IBM MQ MQI client 的 CipherSpec

您有三個選項可指定 IBM MQ MQI client 的 CipherSpec 。

這些選項如下:

- 使用通道定義表
- 在 MQCONNX 呼叫中使用 MQCD 結構 (位於 MQCD_VERSION_7 或更高版本) 中的 SSLCipherSpec 欄位。
- 使用 Active Directory (在具有 Active Directory 支援的 Windows 系統上)

使用 IBM MQ classes for Java 和 IBM MQ classes for JMS 指定 CipherSuite

IBM MQ classes for Java 和 IBM MQ classes for JMS 指定不同於其他平台的 CipherSuites 。

如需使用 IBM MQ classes for Java 指定 CipherSuite 的相關資訊，請參閱 Java 的傳輸層安全 (TLS) 支援

如需使用 IBM MQ classes for JMS 來指定 CipherSuite 的相關資訊，請參閱 搭配使用傳輸層安全 (TLS) 與 IBM MQ classes for JMS

指定 IBM MQ.NET 的 CipherSpec

對於 IBM MQ.NET，您可以使用 MQEnvironment 類別或使用連線內容雜湊表中的 MQC.SSL_CIPHER_SPEC_PROPERTY 來指定 CipherSpec。

如需為 .NET 未受管理用戶端指定 CipherSpec 的相關資訊，請參閱 對未受管理 .NET 用戶端啟用 TLS

如需為 .NET 受管理用戶端指定 CipherSpec 的相關資訊，請參閱 受管理 .NET 用戶端的 CipherSpec 支援

z/OS 搭配使用 AT-TLS 與 IBM MQ for z/OS

「應用程式透通傳輸層安全 (AT-TLS)」提供 z/OS 應用程式的 TLS 支援，而不需要那些應用程式實作 TLS 支援，甚至知道正在使用 TLS。 AT-TLS 僅在 z/OS 上可用。

AT-TLS 可以與 IBM MQ for z/OS 的所有版本搭配使用。

在搭配使用 AT-TLS 與 IBM MQ for z/OS 之前，請確定您瞭解涉及的 第 372 頁的『限制』。

若要使用 應用程式透通傳輸層安全，您可以定義包含一組規則的原則陳述式， z/OS Communications Server 會使用這些規則來決定哪些 TCP/IP 連線已透通啟用 TLS。

IBM MQ for z/OS 具有自己的 TLS 實作，這需要通道具有使用受支援 CipherSpec 配置的 SSLCIPH 參數。

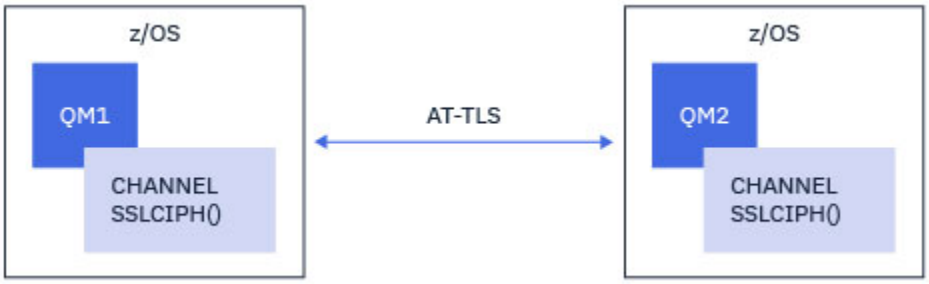
決定在通道上啟用 TLS 時， IBM MQ 管理者可以決定使用 AT-TLS 或 IBM MQ TLS。 此決策通常是根據 AT-TLS 是用於其他中介軟體，還是基於效能影響。 如需 AT-TLS 與 IBM MQ TLS 效能的基本比較，請參閱 MP16: Capacity Planning and Tuning for IBM MQ for z/OS。

實務練習

在下列情況下，支援搭配使用 AT-TLS 與 IBM MQ：

情境 1

在兩個 IBM MQ for z/OS 佇列管理程式之間，通道兩端都使用 AT-TLS。 亦即，兩個通道都不指定 SSLCIPH 屬性。 此方法可以與任何訊息通道搭配使用。



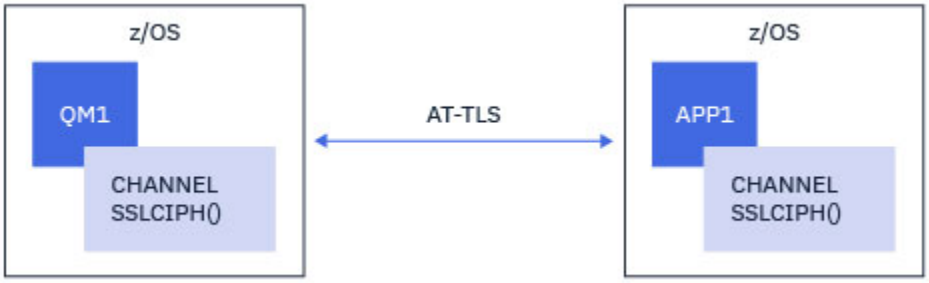
此實務範例的實作包含定義兩個 AT-TLS 原則，通道的每一端各一個。 這些原則與搭配 Scenario 3 或 Scenario 4 使用的原則相同。

例如，如果通道從使用名為 CipherSpec 的單一通道變更為使用 AT-TLS，則出埠通道將使用 第 373 頁的『使用名為 CipherSpec 的單一 IBM MQ for Multiplatforms 佇列管理程式在出埠通道上配置 AT-TLS』 中的原則，而入埠通道將使用 第 381 頁的『使用單一名稱 CipherSpec 在 IBM MQ for Multiplatforms 佇列管理程式的入埠通道上配置 AT-TLS』 中的原則。

如果通道從使用別名 CipherSpec 變更為使用 AT-TLS，則出埠通道將使用 第 377 頁的『使用別名 CipherSpecs 在 IBM MQ for Multiplatforms 佇列管理程式的出埠通道上配置 AT-TLS』 中的原則， 入埠通道將使用 第 385 頁的『使用別名 CipherSpec 在 IBM MQ for Multiplatforms 佇列管理程式的入埠通道上配置 AT-TLS』 中的原則。

實務範例 2

在 IBM MQ for z/OS 佇列管理程式與在 z/OS 上執行的 IBM MQ Java 用戶端應用程式之間，通道兩端都使用 AT-TLS。 也就是說，伺服器連線通道和用戶端連線通道都未指定 SSLCIPH 屬性。



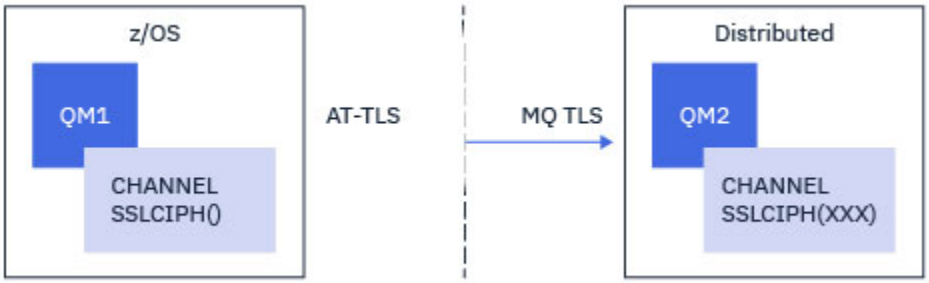
此實務範例的實作包含定義兩個 AT-TLS 原則，通道的每一端各一個。 這些原則與搭配 Scenario 3 或 Scenario 4 使用的原則相同。

例如，如果通道從使用名為 CipherSpec 的單一通道變更為使用 AT-TLS，則用戶端連線通道將使用 第 373 頁的『使用名為 CipherSpec 的單一 IBM MQ for Multiplatforms 佇列管理程式在出埠通道上配置 AT-TLS』 中的原則，而伺服器連線通道將使用 第 381 頁的『使用單一名稱 CipherSpec 在 IBM MQ for Multiplatforms 佇列管理程式的入埠通道上配置 AT-TLS』 中的原則。

如果通道從使用別名 CipherSpec 變更為使用 AT-TLS，則用戶端連線通道將使用 第 377 頁的『使用別名 CipherSpecs 在 IBM MQ for Multiplatforms 佇列管理程式的出埠通道上配置 AT-TLS』 中的原則，伺服器連線通道將使用 第 385 頁的『使用別名 CipherSpec 在 IBM MQ for Multiplatforms 佇列管理程式的入埠通道上配置 AT-TLS』 中的原則。

實務範例 3

在 IBM MQ for z/OS 佇列管理程式與執行於 IBM MQ for Multiplatforms 上的佇列管理程式之間 (其中 IBM MQ for z/OS 佇列管理程式會使用 AT-TLS，而 IBM MQ for Multiplatforms 佇列管理程式會使用 IBM MQ TLS)，方法是指定 SSLCIPH 屬性與單一名稱 CipherSpec。 這適用於叢集傳送端和叢集接收端以外的所有訊息通道類型。

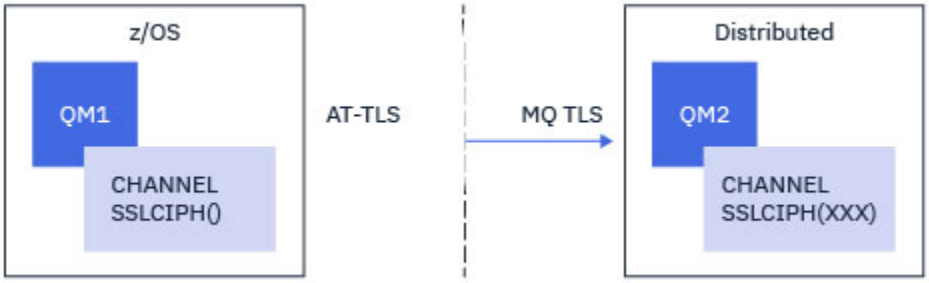


如需從 IBM MQ for z/OS 佇列管理程式至 IBM MQ for Multiplatforms 佇列管理程式之出埠通道的範例 AT-TLS 配置，請參閱 第 373 頁的『使用名為 CipherSpec 的單一 IBM MQ for Multiplatforms 佇列管理程式在出埠通道上配置 AT-TLS』；如需從 IBM MQ for Multiplatforms 佇列管理程式至 IBM MQ for z/OS 佇列管理程式之入埠通道的範例 AT-TLS 配置，請參閱 第 381 頁的『使用單一名稱 CipherSpec 在 IBM MQ for Multiplatforms 佇列管理程式的入埠通道上配置 AT-TLS』。

當兩個佇列管理程式都位於 z/OS 上，但右側的佇列管理程式尚未配置為使用 AT-TLS 時，可以使用相同的 AT-TLS 配置。

實務範例 4

在 IBM MQ for z/OS 佇列管理程式與在 IBM MQ for Multiplatforms 上執行的佇列管理程式之間，透過指定別名為 CipherSpec 的 SSLCIPH 屬性，其中 IBM MQ for z/OS 佇列管理程式會使用 AT-TLS，而 IBM MQ for Multiplatforms 佇列管理程式會使用 IBM MQ TLS。 這適用於叢集傳送端和叢集接收端以外的所有訊息通道類型。

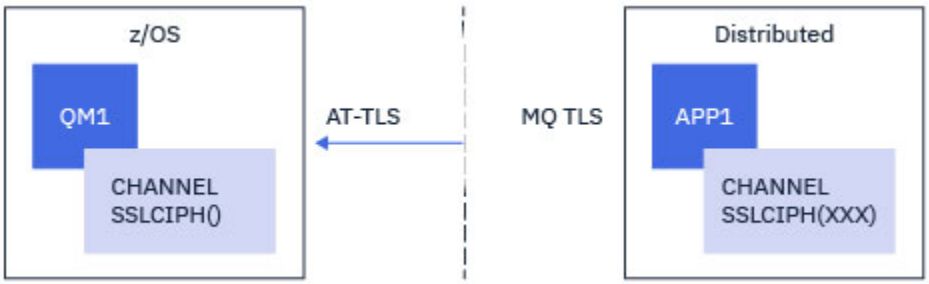


See 第 377 頁的『使用別名 CipherSpecs 在 IBM MQ for Multiplatforms 佇列管理程式的出埠通道上配置 AT-TLS』 for an example AT-TLS configuration for outbound channels from the IBM MQ for z/OS queue manager to the IBM MQ for Multiplatforms queue manager, and 第 385 頁的『使用別名 CipherSpec 在 IBM MQ for Multiplatforms 佇列管理程式的入埠通道上配置 AT-TLS』, and 第 385 頁的『使用別名 CipherSpec 在 IBM MQ for Multiplatforms 佇列管理程式的入埠通道上配置 AT-TLS』 for an example AT-TLS configuration for inbound channels from the IBM MQ for Multiplatforms queue manager to the IBM MQ for z/OS queue manager.

當兩個佇列管理程式都位於 z/OS 上，但右側的佇列管理程式尚未配置為使用 AT-TLS 時，可以使用相同的 AT-TLS 配置。

實務範例 5

在 IBM MQ for z/OS 佇列管理程式與在 IBM MQ for Multiplatforms 上執行的用戶端應用程式之間，其中 IBM MQ for z/OS 佇列管理程式會使用 AT-TLS，而用戶端應用程式會使用 IBM MQ TLS，方法是指定 SSLCIPH 屬性與單一名稱 CipherSpec。

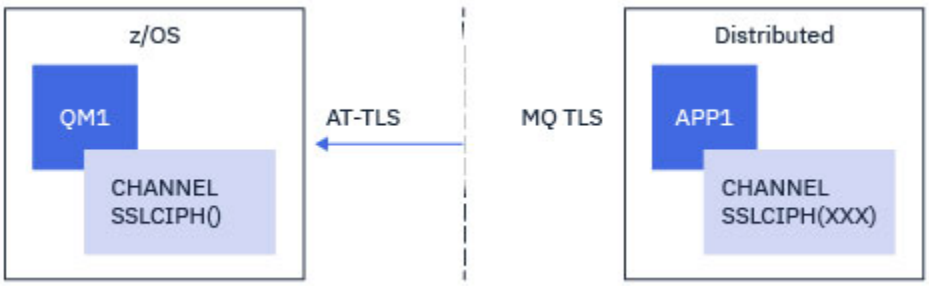


此實務範例需要符合入埠訊息通道所使用之相同需求的單一 AT-TLS 原則; 請參閱 第 381 頁的『使用單一名稱 CipherSpec 在 IBM MQ for Multiplatforms 佇列管理程式的入埠通道上配置 AT-TLS』。

當用戶端應用程式是 Java 應用程式時，也可以使用相同的 AT-TLS 配置，而且也在 z/OS 上執行，但尚未配置成使用 AT-TLS。

實務範例 6

在 IBM MQ for z/OS 佇列管理程式與在 IBM MQ for Multiplatforms 上執行的用戶端應用程式之間， IBM MQ for z/OS 佇列管理程式會使用 AT-TLS，而用戶端應用程式會使用 IBM MQ TLS，方法是指定別名為 CipherSpec 的 SSLCIPH 屬性。



此實務範例需要符合入埠訊息通道所使用之相同需求的單一 AT-TLS 原則; 請參閱 第 385 頁的『使用別名 CipherSpec 在 IBM MQ for Multiplatforms 佇列管理程式的入埠通道上配置 AT-TLS』。

當用戶端應用程式是 Java 應用程式時，也可以使用相同的 AT-TLS 配置，而且也在 z/OS 上執行，但尚未配置成使用 AT-TLS。

限制

IBM MQ for z/OS 不知道 AT-TLS，因此之前的實務範例有一些適用的限制:

- 與 IBM MQ TLS 組合的 AT-TLS 無法使用叢集傳送端和叢集接收端通道。
- IBM MQ for z/OS 佇列管理程式不知道他們正在使用 AT-TLS，且不會從其友機佇列管理程式或用戶端接收任何憑證資訊。 因此，下列屬性不會影響使用 AT-TLS 之通道的 z/OS 端:
  - SSLCAUTH 及 SSLPEER 通道屬性
  - SSLRKEYC 佇列管理程式屬性
  - CHLAUTH 規則的 SSLPEERMAP 屬性
- 使用 TLS 秘密金鑰重新協議需要通道兩端都使用 IBM MQ TLS。 因此，如果使用 AT-TLS 連接至 IBM MQ for z/OS 佇列管理程式，則 IBM MQ for Multiplatforms 佇列管理程式或用戶端不應該啟用 TLS 秘密金鑰重新協議。

  若要停用佇列管理程式的 TLS 秘密金鑰重新協議，請將佇列管理程式 SSLRKEYC 參數設為 `0`。 若為用戶端，請視用戶端類型而定，將相關參數設為 `0`。 如需如何執行此動作的詳細資料，請參閱 第 389 頁的『重設 SSL 和 TLS 秘密金鑰』。

AT-TLS 配置陳述式

AT-TLS 是使用一組陳述式來配置。 在本主題所記載的實務範例中使用的實務範例如下:

TTLSRule
指定一組準則，用來比對 TCP/IP 連線與 TLS 配置。 這又會參照其他陳述式類型。

TTLSGroupAction
指定是否啟用參照 `TTLSRule` 。

TTLSEnvironmentAction
指定參照 `TTLSRule` 的詳細配置，並參照一些其他陳述式。

TTLSKeyringParms
參照 AT-TLS 要使用的金鑰環。

TTLSCipherParms
定義要使用的密碼組合。

TTLSEnvironmentAdvancedParms
定義啟用哪些 TLS 或 SSL 通訊協定。

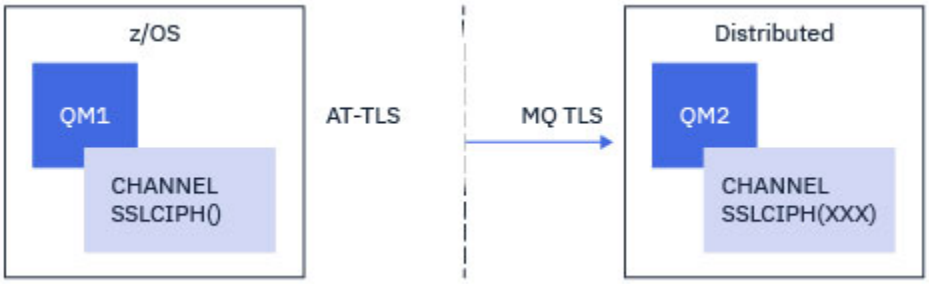
小心: 這裡未記載具有 AT-TLS 的其他 AT-TLS 原則陳述式，視需要可與 IBM MQ 搭配使用。 不過，IBM MQ 僅已使用本主題中說明的原則進行測試。

使用名為 *CipherSpec* 的單一 *IBM MQ for Multiplatforms* 佇列管理程式在出埠通道上配置 *AT-TLS*

如何在從 IBM MQ for z/OS 佇列管理程式至 IBM MQ for Multiplatforms 佇列管理程式的出埠通道上設定 AT-TLS。 在此情況下， z/OS 佇列管理程式上的通道是未設定 SSLCIPH 屬性的傳送端通道，而非 z/OS 佇列管理程式上的通道是 SSLCIPH 屬性設為單一 CipherSpec 的接收端通道。

如需使用別名 CipherSpec 的範例，請參閱 第 377 頁的『使用別名 CipherSpecs 在 IBM MQ for Multiplatforms 佇列管理程式的出埠通道上配置 AT-TLS』 。

在此範例中，將調整使用 ANY_TLS13 別名 CipherSpec 的現有傳送端-接收端通道配對，以便傳送端通道使用 AT-TLS 而非 IBM MQ TLS。



在此範例中，將調整使用 TLS 1.3 TLS_AES_256_GCM_SHA384 CipherSpec 的現有傳送端-接收端通道配對，以便傳送端通道使用 AT-TLS 而非 IBM MQ TLS。

對配置進行次要調整，即可使用其他 TLS 通訊協定及 CipherSpecs 。 除了叢集傳送端和叢集接收端通道之外，其他訊息通道類型可以在不變更 AT-TLS 配置的情況下使用。

小心: TLS 1.3 只能在 z/OS 2.4 版或更新版本上使用。

程序

步驟 1: 停止通道

步驟 2: 建立並套用 AT-TLS 原則

您需要針對此實務範例建立下列 AT-TLS 陳述式:

1. TTLSRule 陳述式，將通道起始程式位址空間的出埠連線與目標接收端通道的 IP 位址和埠號相符。 這些值應該符合傳送端通道的 CONNAME 中使用的資訊。 在這裡，已併入進一步過濾，以符合特定的通道起始程式工作名稱。

```
TTLSRule                        CSQ1-TO-REMOTE
{
  LocalAddr                     ALL
  RemoteAddr                    123.456.78.9
  RemotePortRange               1414
  Jobname                       CSQ1CHIN
  Direction                     OUTBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-OUTBOUND-ENVIRONMENT-ACTION
}
```

   前述規則會針對從 CSQ1CHIN 工作到埠 1414 上 IP 位址 123.456.78.9 的連線進行比對。

   如需其他進階過濾選項的說明，請參閱 TTLSRule。

2. 啟用規則的 TTLSGroupAction 陳述式。 TTLSRule 會使用 **TTLSGroupActionRef** 內容來參照 TTLSGroupAction 。

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. **TTLSEnvironmentActionRef** 內容與 TTLSRule 相關聯的 TTLSEnvironmentAction 陳述式。TTLSEnvironmentAction 會配置 TLS 環境，並指定要使用的金鑰環。

```
TTLSEnvironmentAction               CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                     CLIENT
  TTLSKeyringParmsRef               CSQ1-KEYRING
  TTLSCipherParmsRef                CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef   CSQ1-ENVIRONMENT-ADVANCED
}
```

4. 透過 **TTLSKeyringParmsRef** 內容與 TTLSEnvironmentAction 相關聯的 TTLSKeyringParms 陳述式，並定義 AT-TLS 使用的金鑰環。

   金鑰環應該包含遠端非 z/OS 佇列管理程式所信任的憑證。 此金鑰環的定義方式與通道起始程式所使用的金鑰環相同; 請參閱 第 215 頁的『配置 z/OS 系統以使用 TLS』。

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. **TTLSCipherParmsRef** 內容與 TTLSEnvironmentAction 相關聯的 TTLSCipherParms 陳述式。

   此陳述式必須包含單一密碼組合名稱，其必須與目標接收端通道上使用的 IBM MQ CipherSpec 名稱相等。

   註: AT-TLS 密碼組合名稱不一定符合 IBM MQ CipherSpec 名稱。 不過，若要尋找符合 IBM MQ CipherSpec 名稱的 AT-TLS 密碼組合名稱，可以在下表中尋找 IBM MQ CipherSpec 名稱，並使用 TTLSCipherParms 陳述式主題中「表 2」的展開字元直欄來交互參照十六進位代碼直欄。

表 *80: z/OS* 上的 *CipherSpecs*，來自 *IBM MQ for z/OS 9.2.0*

CipherSpec	**通訊協定**	**十六進位碼**	**依預設啟用**
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	是
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	是
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	是
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	是
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	是
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	是
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	是
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	是

表 80: z/OS 上的 CipherSpecs，來自 IBM MQ for z/OS 9.2.0 (繼續)			
CipherSpec	**通訊協定**	**十六進位碼**	**依預設啟用**
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	是
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	是
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	是
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	是
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	是
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	否
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	否
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	否
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	否
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	否
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	否
TRIPLE_DES_SHA_US	SSL 第 3 版	000A	否
RC4_SHA_US	SSL 第 3 版	0005	否
RC4_MD5_US	SSL 第 3 版	0004	否
DES_SHA_EXPORT	SSL 第 3 版	0009	N
RC4_MD5_EXPORT	SSL 第 3 版	0003	否
RC2_MD5_EXPORT	SSL 第 3 版	0006	否
NULL_SHA	SSL 第 3 版	0002	否
NULL_MD5	SSL 第 3 版	0001	否

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_AES_256_GCM_SHA384
}
```

6. TTLSEnvironmentAdvancedParms 陳述式由 **TTLSEnvironmentAdvancedParmsRef** 內容與 TTLSEnvironmentAction 相關聯。

   此陳述式可用來指定啟用哪些 SSL 及 TLS 通訊協定。 使用 IBM MQ 時，您應該只啟用符合 TTLSCipherParms 陳述式所用密碼組合名稱的單一通訊協定。

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

完整的陳述式集如下，應該套用至原則代理程式:

```
TTLSRule                       CSQ1-TO-REMOTE
{
  LocalAddr                    ALL
  RemoteAddr                   123.456.78.9
  RemotePortRange              1414
  Jobname                      CSQ1CHIN
  Direction                    OUTBOUND
  TTLSGroupActionRef           CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef     CSQ1-OUTBOUND-ENVIRONMENT-ACTION
}

TTLSGroupAction              CSQ1-GROUP-ACTION
{
  TTLSEnabled                ON
}

TTLSEnvironmentAction              CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                    CLIENT
  TTLSKeyringParmsRef              CSQ1-KEYRING
  TTLSCipherParmsRef               CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms             CSQ1-KEYRING
{
  Keyring                    MQCHIN/CSQ1RING
}

TTLSCipherParms              CSQ1-CIPHERPARM
{
  V3CipherSuites             TLS_AES_256_GCM_SHA384
}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

步驟 3: 從 z/OS 通道移除 SSLCIPH

使用下列指令，從 z/OS 通道移除 CipherSpec :

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

步驟 4: 啟動通道

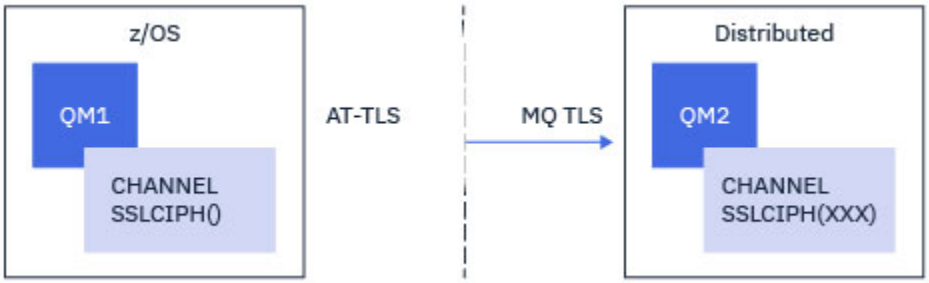
通道啟動之後，它將使用 AT-TLS 與 IBM MQ TLS 的組合。

小心: 之前的 AT-TLS 陳述式只是最小配置。 這裡未記載具有 AT-TLS 的其他 AT-TLS 原則陳述式 ， 視需要可與 IBM MQ 搭配使用。 不過， IBM MQ 僅已使用說明的原則進行測試。

使用別名 *CipherSpecs* 在 *IBM MQ for Multiplatforms* 佇列管理程式的出埠通道上配置 *AT-TLS*

如何在從 IBM MQ for z/OS 佇列管理程式至 IBM MQ for Multiplatforms 佇列管理程式的出埠通道上設定 AT-TLS。 在此情況下， z/OS 佇列管理程式上的通道是未設定 SSLCIPH 屬性的傳送端通道，而非 z/OS 佇列管理程式上的通道是將 SSLCIPH 屬性設為別名 CipherSpec 的接收端通道。

在此範例中，將調整使用 ANY_TLS13 別名 CipherSpec 的現有傳送端-接收端通道配對，以便傳送端通道使用 AT-TLS 而非 IBM MQ TLS。



對配置進行次要調整，即可使用其他 TLS 通訊協定及 CiperSpecs 。 除了叢集傳送端和叢集接收端通道之外，其他訊息通道類型可以在不變更 AT-TLS 配置的情況下使用。

小心: TLS 1.3 只能在 z/OS 2.4 版或更新版本上使用。

程序

步驟 1: 停止通道

步驟 2: 建立並套用 AT-TLS 原則

您需要針對此實務範例建立下列 AT-TLS 陳述式:

1. TTLSRule 陳述式，將通道起始程式位址空間的出埠連線與目標接收端通道的 IP 位址和埠號相符。 這些值應該符合傳送端通道的 CONNAME 中使用的資訊。 在這裡，已併入進一步過濾，以符合特定的通道起始程式工作名稱。

```
TTLSRule                        CSQ1-TO-REMOTE
{
  LocalAddr                     ALL
  RemoteAddr                    123.456.78.9
  RemotePortRange               1414
  Jobname                       CSQ1CHIN
  Direction                     OUTBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-OUTBOUND-ENVIRONMENT-ACTION
}
```

   前述規則會針對從 CSQ1CHIN 工作到埠 1414 上 IP 位址 123.456.78.9 的連線進行比對。

   如需其他進階過濾選項的說明，請參閱 TTLSRule。

2. 啟用規則的 TTLSGroupAction 陳述式。 TTLSRule 會使用 **TTLSGroupActionRef** 內容來參照 TTLSGroupAction 。

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. **TTLSEnvironmentActionRef** 內容與 TTLSRule 相關聯的 TTLSEnvironmentAction 陳述式。 TTLSEnvironmentAction 會配置 TLS 環境，並指定要使用的金鑰環。

```
TTLSEnvironmentAction                CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                      CLIENT
  TTLSKeyringParmsRef                CSQ1-KEYRING
  TTLSCipherParmsRef                 CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef    CSQ1-ENVIRONMENT-ADVANCED
}
```

4. 透過 **TTLSKeyringParmsRef** 內容與 TTLSEnvironmentAction 相關聯的 TTLSKeyringParms 陳述式，並定義 AT-TLS 使用的金鑰環。

   金鑰環應該包含遠端非 z/OS 佇列管理程式所信任的憑證。 此金鑰環的定義方式與通道起始程式所使用的金鑰環相同; 請參閱 第 215 頁的『配置 z/OS 系統以使用 TLS』。

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. **TTLSCipherParmsRef** 內容與 TTLSEnvironmentAction 相關聯的 TTLSCipherParms 陳述式。

   此陳述式必須包含一個以上密碼組合名稱，其中至少一個應該與目標接收端通道上所用別名 CipherSpec 所隱含的 CipherSpecs 集相容。

   註: AT-TLS 密碼組合名稱不一定符合 IBM MQ CipherSpec 名稱。 不過，若要尋找符合 IBM MQ CipherSpec 名稱的 AT-TLS 密碼組合名稱，可以在下表中找到 IBM MQ CipherSpec 名稱，並使用 TTLSCipherParms 主題中「表 2」的展開字元直欄來交互參照十六進位代碼直欄。

表 *81: z/OS* 上的 *CipherSpecs*，來自 *IBM MQ for z/OS 9.2.0*

CipherSpec	**通訊協定**	**十六進位碼**	**依預設啟用**
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	是
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	是
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	是
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	是
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	是
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	是
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	是
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	是
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	是
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	是

表 81: z/OS 上的 CipherSpecs，來自 IBM MQ for z/OS 9.2.0 (繼續)			
CipherSpec	**通訊協定**	**十六進位碼**	**依預設啟用**
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	是
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	是
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	是
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	否
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	否
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	否
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	否
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	否
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	否
TRIPLE_DES_SHA_US	SSL 第 3 版	000A	否
RC4_SHA_US	SSL 第 3 版	0005	否
RC4_MD5_US	SSL 第 3 版	0004	否
DES_SHA_EXPORT	SSL 第 3 版	0009	N
RC4_MD5_EXPORT	SSL 第 3 版	0003	否
RC2_MD5_EXPORT	SSL 第 3 版	0006	否
NULL_SHA	SSL 第 3 版	0002	否
NULL_MD5	SSL 第 3 版	0001	否

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256
}
```

小心: 如果佇列管理程式及 AT-TLS 原則都支援 TLS 1.3，則只有包含至少一個 TLS 1.3 CipherSpec 的別名 CipherSpecs 才容許啟動通道。 例如，使用 ANY_TLS12 會導致通道無法啟動，即使 TTLSCipherParms 包含 TLS 1.2 CipherSpecs，但使用 ANY_TLS12_OR_HIGHER 或 ANY_TLS13 則容許通道啟動。 如需說明，請參閱 第 366 頁的『別名 CipherSpec 設定之間的關係』。

6. TTLSEnvironmentAdvancedParms 陳述式由 **TTLSEnvironmentAdvancedParmsRef** 內容與 TTLSEnvironmentAction 相關聯。

   此陳述式可用來指定已啟用哪些 SSL 及 TLS 通訊協定，且應該與 TTLSCipherParms 陳述式中的密碼組合一致。

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

完整的陳述式集如下，應該套用至原則代理程式:

```
TTLSRule                      CSQ1-TO-REMOTE
{
  LocalAddr                   ALL
  RemoteAddr                  123.456.78.9
  RemotePortRange             1414
  Jobname                     CSQ1CHIN
  Direction                   OUTBOUND
  TTLSGroupActionRef          CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef    CSQ1-OUTBOUND-ENVIRONMENT-ACTION
}

TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}

TTLSEnvironmentAction             CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                   CLIENT
  TTLSKeyringParmsRef             CSQ1-KEYRING
  TTLSCipherParmsRef              CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}

TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256

}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

步驟 3: 從 z/OS 通道移除 SSLCIPH

使用下列指令，從 z/OS 通道移除 CipherSpec :

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

步驟 4: 啟動通道

通道啟動之後，它將使用 AT-TLS 與 IBM MQ TLS 的組合。

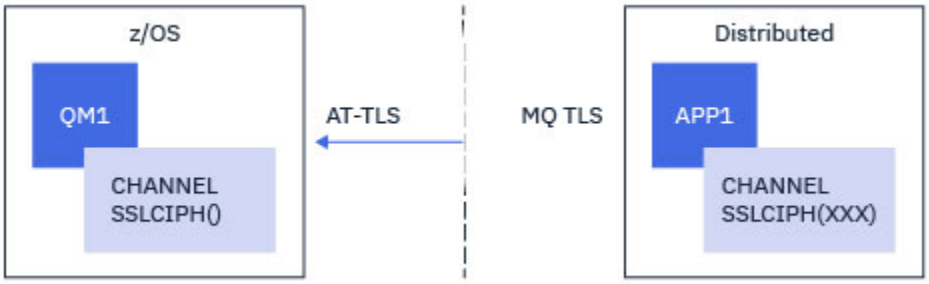
小心: 之前的 AT-TLS 陳述式只是最小配置。 這裡未記載具有 AT-TLS 的其他 AT-TLS 原則陳述式 ，視需要可與 IBM MQ 搭配使用。 不過， IBM MQ 僅已使用說明的原則進行測試。

使用單一名稱 *CipherSpec* 在 *IBM MQ for Multiplatforms* 佇列管理程式的入埠通道上配置 *AT-TLS*

如何在從 IBM MQ for Multiplatforms 佇列管理程式至 IBM MQ for z/OS 佇列管理程式的入埠通道上設定 AT-TLS。 在此情況下， z/OS 佇列管理程式上的通道是未設定 SSLCIPH 屬性的接收端通道，而非 z/OS 佇列管理程式上的通道是 SSLCIPH 屬性設為單一 CipherSpec 的傳送端通道。

如需使用別名 CipherSpec 的範例，請參閱 第 385 頁的『使用別名 CipherSpec 在 IBM MQ for Multiplatforms 佇列管理程式的入埠通道上配置 AT-TLS』 。

在此範例中，將調整使用 TLS 1.3 TLS_AES_256_GCM_SHA384 CipherSpec 的現有傳送端-接收端通道配對，以便接收端通道使用 AT-TLS 而非 IBM MQ TLS。



對配置進行次要調整，即可使用其他 TLS 通訊協定及 CipherSpecs 。 除了叢集傳送端和叢集接收端通道之外，其他訊息通道類型可以在不變更 AT-TLS 配置的情況下使用。

小心: TLS 1.3 只能在 z/OS 2.4 版或更新版本上使用。

程序

步驟 1: 停止通道

步驟 2: 建立並套用 AT-TLS 原則

您需要針對此實務範例建立下列 AT-TLS 陳述式:

1. TTLSRule 陳述式，用來比對從傳送端通道 IP 位址到通道起始程式位址空間的入埠連線。 在這裡，已併入進一步過濾，以符合特定的通道起始程式工作名稱。

```
TTLSRule                        REMOTE-TO-CSQ1
{
  LocalAddr                     ALL
  LocalPortRange                1414
  RemoteAddr                    123.456.78.9
  Jobname                       CSQ1CHIN
  Direction                     INBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-INBOUND-ENVIRONMENT-ACTION
}
```

   前述規則會比對從遠端 IP 位址 123.456.78.9 進入本端埠 1414 上 CSQ1CHIN 工作的連線。

   如需其他進階過濾選項的說明，請參閱 TTLSRule。

2. 啟用規則的 TTLSGroupAction 陳述式。 TTLSRule 會使用 **TTLSGroupActionRef** 內容來參照 TTLSGroupAction 。

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. TTLSEnvironmentAction 陳述式透過 **TTLSEnvironmentActionRef** 內容與 TTLSRule 相關聯。 TTLSEnvironmentAction 會配置 TLS 環境，並指定要使用的金鑰環。

```
TTLSEnvironmentAction                  CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                        SERVER
  TTLSKeyringParmsRef                  CSQ1-KEYRING
  TTLSCipherParmsRef                   CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef      CSQ1-ENVIRONMENT-ADVANCED
}
```

AT-TLS 提供提供交互鑑別的功能，相當於使用 SSLCAUTH 通道屬性。 這是透過針對入埠 TTLSEnvironmentAction 陳述式具有 **HandshakeRole** 值為 *ServerWithClientAuth* 的 TTLSEnvironmentAction 陳述式來完成。

4. TTLSKeyringParms 陳述式透過 **TTLSKeyringParmsRef** 內容與 TTLSEnvironmentAction 相關聯，並定義 AT-TLS 使用的金鑰環。

   金鑰環應該包含遠端非 z/OS 佇列管理程式所信任的憑證。 此金鑰環的定義方式與通道起始程式所使用的金鑰環相同; 請參閱 第 215 頁的『配置 z/OS 系統以使用 TLS』。

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. **TTLSCipherParmsRef** 內容與 TTLSEnvironmentAction 相關聯的 TTLSCipherParms 陳述式。

   此陳述式必須包含單一密碼組合名稱，其必須與遠端傳送端通道上使用的 IBM MQ CipherSpec 名稱相等。

   註: AT-TLS 密碼組合名稱不一定符合 IBM MQ CipherSpec 名稱。 不過，若要尋找符合 IBM MQ CipherSpec 名稱的 AT-TLS 密碼組合名稱，可以在下表中尋找 IBM MQ CipherSpec 名稱，並使用 TTLSCipherParms 陳述式主題中「表 2」的展開字元直欄來交互參照十六進位代碼直欄。

表 *82: z/OS* 上的 *CipherSpecs* ，來自 *IBM MQ for z/OS 9.2.0*

CipherSpec	**通訊協定**	**十六進位碼**	**依預設啟用**
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	是
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	是
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	是
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	是
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	是
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	是
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	是
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	是
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	是

表 82: z/OS 上的 CipherSpecs，來自 IBM MQ for z/OS 9.2.0 (繼續)			
CipherSpec	**通訊協定**	**十六進位碼**	**依預設啟用**
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	是
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	是
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	是
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	是
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	否
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	否
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	否
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	否
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	否
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	否
TRIPLE_DES_SHA_US	SSL 第 3 版	000A	否
RC4_SHA_US	SSL 第 3 版	0005	否
RC4_MD5_US	SSL 第 3 版	0004	否
DES_SHA_EXPORT	SSL 第 3 版	0009	N
RC4_MD5_EXPORT	SSL 第 3 版	0003	否
RC2_MD5_EXPORT	SSL 第 3 版	0006	否
NULL_SHA	SSL 第 3 版	0002	否
NULL_MD5	SSL 第 3 版	0001	否

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_AES_256_GCM_SHA384
}
```

6. TTLSEnvironmentAdvancedParms 陳述式由 **TTLSEnvironmentAdvancedParmsRef** 內容與 TTLSEnvironmentAction 相關聯。

   此陳述式可用來指定啟用哪些 SSL 及 TLS 通訊協定。 使用 IBM MQ 時，您應該只啟用符合 TTLSCipherParms 陳述式所用密碼組合名稱的單一通訊協定。

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

完整的陳述式集如下，應該套用至原則代理程式:

```
TTLSRule                       REMOTE-TO-CSQ1
{
  LocalAddr                    ALL
  LocalPortRange               1414
  RemoteAddr                   123.456.78.9
  Jobname                      CSQ1CHIN
  Direction                    INBOUND
  TTLSGroupActionRef           CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef     CSQ1-INBOUND-ENVIRONMENT-ACTION
}

TTLSGroupAction              CSQ1-GROUP-ACTION
{
  TTLSEnabled                ON
}

TTLSEnvironmentAction              CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                    SERVER
  TTLSKeyringParmsRef              CSQ1-KEYRING
  TTLSCipherParmsRef               CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms             CSQ1-KEYRING
{
  Keyring                    MQCHIN/CSQ1RING
}

TTLSCipherParms              CSQ1-CIPHERPARM
{
  V3CipherSuites             TLS_AES_256_GCM_SHA384
}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

步驟 3: 從 z/OS 通道移除 SSLCIPH

使用下列指令，從 z/OS 通道移除 CipherSpec :

```
ALTER CHANNEL(channel-name) CHLTYPE(RCVR) SSLCIPH(' ')
```

步驟 4: 啟動通道

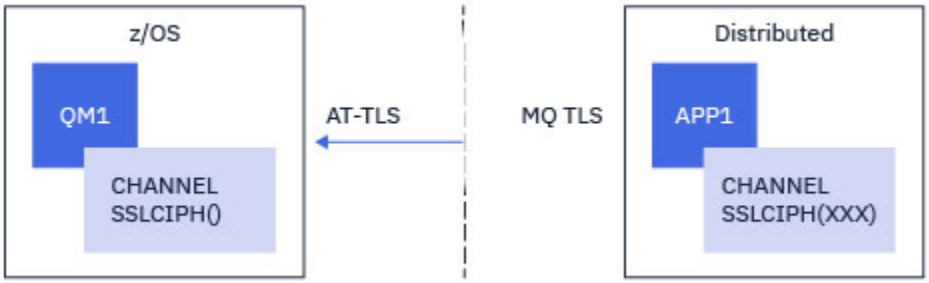
通道啟動之後，它將使用 AT-TLS 與 IBM MQ TLS 的組合。

小心: 之前的 AT-TLS 陳述式只是最小配置。 這裡未記載具有 AT-TLS 的其他 AT-TLS 原則陳述式 ， 視需要可與 IBM MQ 搭配使用。 不過， IBM MQ 僅已使用說明的原則進行測試。

使用別名 *CipherSpec* 在 *IBM MQ for Multiplatforms* 佇列管理程式的入埠通道上配置 *AT-TLS*

如何在從 IBM MQ for Multiplatforms 佇列管理程式至 IBM MQ for z/OS 佇列管理程式的入埠通道上設定 AT-TLS。 在此情況下， z/OS 佇列管理程式上的通道是未設定 SSLCIPH 屬性的接收端通道，而非 z/OS 佇列管理程式上的通道是 SSLCIPH 屬性設為別名 CipherSpec 的傳送端通道。

在此範例中，將調整使用任何 TLS 1.3 CipherSpec 的現有傳送端-接收端通道配對，以便接收端通道使用 AT-TLS 而非 IBM MQ TLS。



對配置進行次要調整，即可使用其他 TLS 通訊協定及 CiperSpecs 。 除了叢集傳送端和叢集接收端通道之外，其他訊息通道類型可以在不變更 AT-TLS 配置的情況下使用。

小心: TLS 1.3 只能在 z/OS 2.4 版或更新版本上使用。

程序

步驟 1: 停止通道

步驟 2: 建立並套用 AT-TLS 原則

您需要針對此實務範例建立下列 AT-TLS 陳述式:

1. TTLSRule 陳述式，用來比對從傳送端通道 IP 位址到通道起始程式位址空間的入埠連線。 在這裡，已併入進一步過濾，以符合特定的通道起始程式工作名稱。

```
TTLSRule                        REMOTE-TO-CSQ1
{
  LocalAddr                     ALL
  LocalPortRange                1414
  RemoteAddr                    123.456.78.9
  Jobname                       CSQ1CHIN
  Direction                     INBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-INBOUND-ENVIRONMENT-ACTION
}
```

   前述規則會比對從遠端 IP 位址 123.456.78.9 進入本端埠 1414 上 CSQ1CHIN 工作的連線。

   如需其他進階過濾選項的說明，請參閱 TTLSRule。

2. 啟用規則的 TTLSGroupAction 陳述式。 TTLSRule 會使用 **TTLSGroupActionRef** 內容來參照 TTLSGroupAction 。

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. TTLSEnvironmentAction 陳述式透過 **TTLSEnvironmentActionRef** 內容與 TTLSRule 相關聯。 TTLSEnvironmentAction 會配置 TLS 環境，並指定要使用的金鑰環。

```
TTLSEnvironmentAction                 CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                       SERVER
  TTLSKeyringParmsRef                 CSQ1-KEYRING
  TTLSCipherParmsRef                  CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef     CSQ1-ENVIRONMENT-ADVANCED
}
```

AT-TLS 提供提供交互鑑別的功能，相當於使用 SSLCAUTH 通道屬性。 這是透過針對入埠 TTLSEnvironmentAction 陳述式具有 **HandshakeRole** 值為 *ServerWithClientAuth* 的 TTLSEnvironmentAction 陳述式來完成。

4. TTLSKeyringParms 陳述式透過 **TTLSKeyringParmsRef** 內容與 TTLSEnvironmentAction 相關聯，並定義 AT-TLS 使用的金鑰環。

   金鑰環應該包含遠端非 z/OS 佇列管理程式所信任的憑證。 此金鑰環的定義方式與通道起始程式所使用的金鑰環相同; 請參閱 第 215 頁的『配置 z/OS 系統以使用 TLS』。

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. **TTLSCipherParmsRef** 內容與 TTLSEnvironmentAction 相關聯的 TTLSCipherParms 陳述式。

   此陳述式必須至少包含一個密碼組合名稱，該名稱包含在遠端傳送端通道上設定的別名 CipherSpec 中。

   註: AT-TLS 密碼組合名稱不一定符合 IBM MQ CipherSpec 名稱。 不過，若要尋找符合 IBM MQ CipherSpec 名稱的 AT-TLS 密碼組合名稱，可以在下表中尋找 IBM MQ CipherSpec 名稱，並使用 TTLSCipherParms 陳述式主題中「表 2」的展開字元直欄來交互參照十六進位代碼直欄。

表 *83: z/OS* 上的 *CipherSpecs*，來自 *IBM MQ for z/OS 9.2.0*

CipherSpec	**通訊協定**	**十六進位碼**	**依預設啟用**
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	是
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	是
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	是
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	是
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	是
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	是
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	是
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	是
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	是
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	是

表 83: z/OS 上的 CipherSpecs，來自 IBM MQ for z/OS 9.2.0 (繼續)			
CipherSpec	**通訊協定**	**十六進位碼**	**依預設啟用**
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	是
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	是
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	是
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	否
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	否
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	否
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	否
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	否
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	否
TRIPLE_DES_SHA_US	SSL 第 3 版	000A	否
RC4_SHA_US	SSL 第 3 版	0005	否
RC4_MD5_US	SSL 第 3 版	0004	否
DES_SHA_EXPORT	SSL 第 3 版	0009	N
RC4_MD5_EXPORT	SSL 第 3 版	0003	否
RC2_MD5_EXPORT	SSL 第 3 版	0006	否
NULL_SHA	SSL 第 3 版	0002	否
NULL_MD5	SSL 第 3 版	0001	否

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256
}
```

小心: 如果佇列管理程式及 AT-TLS 原則都支援 TLS 1.3，則只有包含至少一個 TLS 1.3 CipherSpec 的別名 CipherSpecs 才容許啟動通道。 例如，使用 ANY_TLS12 會導致通道無法啟動，即使 TTLSCipherParms 包含 TLS 1.2 CipherSpecs，但使用 ANY_TLS12_OR_HIGHER 或 ANY_TLS13 則容許通道啟動。 如需說明，請參閱 第 366 頁的『別名 CipherSpec 設定之間的關係』。

6. TTLSEnvironmentAdvancedParms 陳述式由 **TTLSEnvironmentAdvancedParmsRef** 內容與 TTLSEnvironmentAction 相關聯。

   此陳述式可用來指定已啟用哪些 SSL 及 TLS 通訊協定，且應該與 TTLSCipherParms 陳述式中的密碼組合一致。

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

完整的陳述式集如下，應該套用至原則代理程式:

```
TTLSRule                      REMOTE-TO-CSQ1
{
  LocalAddr                   ALL
  LocalPortRange              1414
  RemoteAddr                  123.456.78.9
  Jobname                     CSQ1CHIN
  Direction                   INBOUND
  TTLSGroupActionRef          CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef    CSQ1-INBOUND-ENVIRONMENT-ACTION
}

TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}

TTLSEnvironmentAction             CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                   SERVER
  TTLSKeyringParmsRef             CSQ1-KEYRING
  TTLSCipherParmsRef              CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}

TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256

}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

步驟 3: 從 z/OS 通道移除 SSLCIPH

使用下列指令，從 z/OS 通道移除 CipherSpec :

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

步驟 4: 啟動通道

通道啟動之後，它將使用 AT-TLS 與 IBM MQ TLS 的組合。

小心: 之前的 AT-TLS 陳述式只是最小配置。 這裡未記載具有 AT-TLS 的其他 AT-TLS 原則陳述式 ，視需要可與 IBM MQ 搭配使用。 不過， IBM MQ 僅已使用說明的原則進行測試。

重設 SSL 和 TLS 秘密金鑰

IBM MQ 支援重設佇列管理程式及用戶端上的秘密金鑰。

當通道中已傳送指定數目的已加密資料位元組時，會重設秘密金鑰。 如果已啟用通道活動訊號，則在通道活動訊號之後傳送或接收資料之前會重設秘密金鑰。

金鑰重設值一律由 IBM MQ 通道的起始端設定。

佇列管理程式

若為佇列管理程式，請搭配使用指令 **ALTER QMGR** 與參數 **SSLRKEYC** ，以設定金鑰重新協議期間使用的值。

在 IBM i 上，搭配使用 **CHGMQM** 與 **SSLRSTCNT** 參數。

MQI 用戶端

依預設， MQI 用戶端不會重新協議秘密金鑰。 您可以使用三種方式之一，讓 MQI 用戶端重新協議金鑰。 在下列清單中，方法依優先順序顯示。 如果您指定多個值，則會使用最高優先順序值。

1. 透過在 MQCONNX 呼叫中使用 MQSCO 結構中的 KeyResetCount 字段
2. 使用環境變數 MQSSLRESET
3. 透過在 MQI 用戶端設定檔中設定 SSLKeyResetCount 屬性

這些變數可以設為 0 到 999 999 999 範圍內的整數，代表重新協議 TLS 秘密金鑰之前在 TLS 交談內傳送及接收的未加密位元組數。 指定值 0 表示永不重新協議 TLS 秘密金鑰。 如果您指定範圍在 1 位元組到 32 KB 之間的 TLS 秘密金鑰重設計數，則 TLS 通道將使用 32 KB 的秘密金鑰重設計數。 這是為了避免對小型 TLS 秘密金鑰重設值進行過多的金鑰重設。

如果指定大於零的值，且通道已啟用通道活動訊號，則在通道活動訊號之後傳送或接收訊息資料之前，也會重新協議秘密金鑰。

每次成功重新協議之後重設下一個秘密金鑰重新協議之前的位元組計數。

有關 MQSCO 結構的完整詳細信息，請參閱 KeyResetCount (MQLONG) 。 如需 MQSSLRESET 的完整資料，請參閱 MQSSLRESET。 如需在用戶端配置檔中使用 TLS 的相關資訊，請參閱 用戶端配置檔的 SSL 段落。

Java

對於 IBM MQ classes for Java，應用程式可以使用下列其中一種方式來重設秘密金鑰:

- 透過設定 MQEnvironment 類別中的 sslResetCount 欄位。
- 透過在 Hashtable 物件中設定環境內容 MQC.SSL_RESET_COUNT_PROPERTY 。 然後應用程式會將雜湊表指派給 MQEnvironment 類別中的 `properties` 欄位，或將雜湊表傳遞給其建構子上的 MQQueueManager 物件。

如果應用程式使用多個這些方式，則會套用一般優先順序規則。 請參閱 類別 com.ibm.mq.MQEnvironment ，以取得優先順序規則。

sslResetCount 欄位或環境屬性 MQC.SSL_RESET_COUNT_PROPERTY 的值表示在重新協商金鑰之前 IBM MQ classes for Java 用戶端程式碼發送和接收的位元組總數。 傳送的位元組數是加密之前的數目，而接收的位元組數是解密之後的數目。 位元組數也包括 IBM MQ classes for Java 用戶端所傳送及接收的控制資訊。

如果重設計數為零 (預設值) ，則永不重新協議秘密金鑰。 如果未指定 CipherSuite ，則會忽略重設計數。

JMS

對於 IBM MQ classes for JMS， SSLRESETCOUNT 內容代表在重新協議用於加密的秘密金鑰之前，連線所傳送及接收的位元組總數。 傳送的位元組數是加密之前的數目，而接收的位元組數是解密之後的數目。 位元組數也包括 IBM MQ classes for JMS 所傳送及接收的控制資訊。 比方說，如果要配置 ConnectionFactory

物件，以用來透過啟用 TLS 之 MQI 通道建立連線，且具有在 4 MB 資料傳送之後重新協議的秘密金鑰，請向 JMSAdmin 發出下列指令:

```
ALTER CF(my.cf) SSLRESETCOUNT(4194304)
```

如果 SSLRESETCOUNT 的值為零 (這是預設值)，則永不重新協議秘密金鑰。 如果未設定 SSLCIPHERSUITE，則會忽略 SSLRESETCOUNT 內容。

.NET

對於.NET 非託管用戶端，整數屬性 SSLKeyResetCount 指示在重新協商金鑰之前在 TLS 會話中傳送和接收的未加密位元組數。

如需在 IBM MQ classes for .NET 中使用物件內容的相關資訊，請參閱 取得及設定屬性值。

對於 .NET 受管理用戶端， SSLStream 類別不支援秘密金鑰重設/重新協議。 但是，為了與其他 IBM MQ 用戶端保持一致， IBM MQ 管理的.NET 用戶端允許應用程式設定 SSLKeyResetCount 。 如需相關資訊，請參閱 秘密金鑰重設或重新協議。

XMS .NET

若為 XMS .NET 未受管理的用戶端，請參閱 與 IBM MQ 佇列管理程式的安全連線。

相關參考

ALTER QMGR
DISPLAY QMGR
變更訊息佇列管理程式 (CHGMQM)
顯示訊息佇列管理程式 (DSPMQM)

在使用者結束程式中實作機密性

在安全結束程式中實作機密性

安全結束程式可以在機密性服務中扮演一個角色，方法是產生並配送對稱金鑰，以加密及解密通道上流動的資料。 執行此動作的一般技術使用 PKI 技術。

一個安全結束程式會產生隨機資料值，使用佇列管理程式的公開金鑰或夥伴安全結束程式所代表的使用者來加密它，並將已加密資料傳送至安全訊息中的夥伴。 夥伴安全結束程式會使用它所代表的佇列管理程式或使用者的私密金鑰來解密隨機資料值。 現在，每一個安全結束程式都可以使用隨機資料值，透過使用兩者都已知的演算法來獨立衍生對稱金鑰。 或者，他們可以使用隨機資料值作為索引鍵。

如果此時第一個安全結束程式尚未鑑別其夥伴，則夥伴所傳送的下一個安全訊息可以包含以對稱金鑰加密的期望值。 第一個安全結束程式現在可以透過檢查夥伴安全結束程式是否能夠正確地加密期望值，來鑑別其夥伴。

如果有多個演算法可供使用，安全結束程式也可以利用這個機會來同意加密及解密通道上流動的資料的演算法。

在訊息結束程式中實作機密性

通道傳送端的訊息結束程式可以加密訊息中的應用程式資料，通道接收端的另一個訊息結束程式可以解密資料。 基於效能原因，通常會使用對稱金鑰演算法來達到此目的。 如需如何產生及配送對稱金鑰的相關資訊，請參閱 第 390 頁的『在使用者結束程式中實作機密性』。

訊息中的標頭 (例如傳輸佇列標頭 MQXQH，包含內嵌的訊息描述子) 不得由訊息結束程式加密。 這是因為在傳送端呼叫訊息結束程式之後，或在接收端呼叫訊息結束程式之前，會進行訊息標頭的資料轉換。 如果標頭已加密，則資料轉換會失敗，且通道會停止。

在傳送和接收結束程式中實作機密性

傳送及接收結束程式可用來加密及解密通道上流動的資料。 由於下列原因，它們比提供此服務的訊息結束程式更適合:

- 在訊息通道上，訊息標頭可以加密，也可以加密訊息中的應用程式資料。
- 傳送及接收結束程式可以在 MQI 通道及訊息通道上使用。 MQI 呼叫上的參數可能包含在 MQI 通道上流動時需要保護的機密應用程式資料。 因此，您可以在這兩種通道上使用相同的傳送及接收結束程式。

在 API 結束程式和 API 交互結束程式中實作機密性

當傳送端應用程式放置訊息時，訊息中的應用程式資料可以由 API 或 API 交互結束程式加密，當接收端應用程式擷取訊息時，由第二個結束程式解密。 基於效能原因，通常會使用對稱金鑰演算法來達到此目的。 不過，在應用程式層次，許多使用者可能彼此傳送訊息，問題是如何確定只有訊息的預期接收端能夠解密訊息。 其中一個解決方案是針對每對相互傳送訊息的使用者使用不同的對稱金鑰。 但此解決方案可能難以管理且耗時，尤其是當使用者屬於不同組織時。 解決此問題的標準方法稱為 數位封裝 ，並使用 PKI 技術。

當應用程式將訊息放入佇列時， API 或 API 交互結束程式會產生隨機對稱金鑰，並使用該金鑰來加密訊息中的應用程式資料。 結束程式會使用預期接收端的公開金鑰來加密對稱金鑰。 然後，它會將訊息中的應用程式資料取代為已加密的應用程式資料及已加密的對稱金鑰。 這樣，只有預期的接收者才能解密對稱金鑰，從而解密應用程式資料。 如果已加密訊息有多個可能的預期接收端，則結束程式可以加密每一個預期接收端的對稱金鑰副本。

如果可以使用不同的演算法來加密及解密應用程式資料，則結束程式可以包括它所使用的演算法名稱。

z/OS V 9.2.0 IBM MQ for z/OS 上靜態資料的機密性 (具有資料集加密)

IBM MQ for z/OS 可以透過將資料寫入作用中日誌資料集、保存日誌資料集、頁集、 引導資料集 (BSDS) 及共用訊息資料集 (SMDS)，來強化客戶及配置資料。

z/OS 提供有效且以原則為基礎的資料集加密。 IBM MQ for z/OS 支援下列項目的 z/OS 資料集加密:

- 作用中日誌資料集; 請參閱附註 第 391 頁的『1』
- 保存日誌資料集; 請參閱附註 第 391 頁的『2』
- 頁集; 請參閱附註 第 391 頁的『1』
- BSDS; 請參閱附註 第 391 頁的『2』
- CSQINP* 資料集; 請參閱附註 第 391 頁的『2』
- V 9.2.0 SMDS; 請參閱附註 第 391 頁的『1』

這會提供個別 z/OS 佇列管理程式上靜態資料的機密性。

附註:

1. V 9.2.0 從 IBM MQ for z/OS 9.2.0 開始， z/OS 作用中日誌的資料集加密。 頁集，且 SMDS 受支援。
2. 所有 IBM MQ for z/OS 版本都支援保存日誌、BSDS 及 CSQINP* 資料集的資料集加密。
3. IBM MQ Advanced Message Security 提供替代機制來保護靜態資料。 此外， AMS 也會保護記憶體及進行中的資料

如需 z/OS 資料集加密的相關資訊，請參閱 使用 z/OS 資料集加密加強功能 。

z/OS 資料集加密的配置不受 IBM MQ for z/OS 控制。 建立資料集時，加密設定會生效。

這表示必須先重建任何現有的資料集，才能使用新的資料集加密原則。

IBM MQ for z/OS 可以使用已加密及未加密資料集的混合執行，但標準配置會加密所有使用的資料集，或不加密任何使用的資料集。

z/OS V 9.2.0 加密 IBM MQ for z/OS 資料集的步驟概觀

如何加密 IBM MQ for z/OS 資料集。

開始之前

您必須確定已在企業中正確配置 z/OS 資料集加密。 如果您在佇列共用群組中設定資料集加密，則必須配置 z/OS 資料集加密以進行資料共用。

註: z/OS 加密資料集必須是延伸格式資料集。

程序

1. 在 RACF 中設定加密金鑰和 `key-label`， 以用來加密資料集。
2. 在 RACF CSFKEYS 類別中建立 `key-label` 的設定檔。
3. 將 READ 存取權授與佇列管理程式的使用者 ID， 以及任何其他需要存取已加密資料的使用者 ID。
   這可能包括用來對資料集執行列印公用程式的使用者 ID。 例如，執行 CSQUTIL SCQUESCON 的使用者需要解密相關頁面集。
4. 將加密 `key-label` 與資料集名稱相關聯。
   您可以針對資料集名稱或高階限定元，使用 SMS 資料類別或 RACF DFP 區段來執行此動作。
   您也可以在配置資料集時，將 `key-label` 與資料集相關聯。
5. 使用 IDCAMS ALTER 重新命名任何現有的資料集。
6. 請使用適當的屬性重新配置資料集。
7. 使用 IDCAMS REPRO， 將已重新命名資料集的內容複製到新的資料集。
   透過將資料複製到資料集的動作來加密資料。
8. 針對需要加密的任何其他資料集，重複步驟 第 392 頁的『4』 至 第 392 頁的『6』 。

z/OS V 9.2.0 如何加密佇列管理程式作用中日誌的範例

下列主題引導您完成在現有作用中日誌上啟用資料集加密的程序。

註: 其他資料集的處理程序與作用中日誌的處理程序類似。

在此範例中:

- 佇列管理程式 CSQ1 在使用者 QMCSQ1 下執行，且具有作用中日誌資料集 CSQ1.LOGS.LOGCOPY1.DS001， CSQ1.LOGS.LOGCOPY1.DS002 等
- 軟硬體環境能夠使用 z/OS 資料集加密
- RACF 用來作為 SAF
- 佇列管理程式已停止

請依下列順序執行程序:

1. 第 392 頁的『配置佇列管理程式的資料集加密金鑰』
2. 第 393 頁的『配置日誌資料集的資料集加密』

z/OS V 9.2.0 配置佇列管理程式的資料集加密金鑰

如何配置佇列管理程式的資料集加密金鑰。

關於這項作業

此作業是 第 393 頁的『配置日誌資料集的資料集加密』的必備項目。

程序

1. 使用 z/OS 金鑰產生器公用程式 (KGUP)來設定具有標籤 (例如， CSQ1DSKY) 的 AES-256 位元加密 DATA 金鑰。
2. 發出下列指令，為 CSQ1DSKY 加密金鑰定義 RACF CSFKEYS 設定檔:

```
RDEFINE CSFKEYS CSQ1DSKY UACC(NONE)
```

3. 透過發出下列指令，配置設定檔的 ICSF 區段，以容許將金鑰用作受保護金鑰:

```
RALTER CSFKEYS CSQ1DSKY ICSF(SYMCPACFWRAP(YES) SYMCPACFRET(YES))
```

4. 發出下列指令，授與 QMCSQ1 對設定檔的 READ 存取權，以容許佇列管理程式使用加密金鑰:

```
PERMIT CSQ1DSKY CLASS(CSFKEYS) ID(QMCSQ1) ACCESS(READ)
```

   提供相同的存取權給任何需要讀取或寫入已加密資料集的管理使用者。
5. 發出下列指令來重新整理 CSFKEYS 類別。

```
SETROPTS RACLIST(CSFKEYS) REFRESH
```

下一步

配置資料集的資料集加密，如 第 393 頁的『配置日誌資料集的資料集加密』 中所述

z/OS V 9.2.0 配置日誌資料集的資料集加密

如何在日誌資料集上配置加密。

開始之前

請確定您已閱讀:

加密 IBM MQ for z/OS 資料集的步驟概觀，並執行中的程序
第 392 頁的『配置佇列管理程式的資料集加密金鑰』

關於這項作業

此方法使用 RACF 通用設定檔的 DFP 區段，以便您可以對符合該設定檔的所有新資料集使用加密金鑰。

或者，您可以配置及使用 SMS 資料類別，或在配置資料集時直接指定金鑰標籤。

如先前所述，在此範例中，佇列管理程式 CSQ1 在使用者 QMCSQ1 下執行，並具有作用中日誌資料集 CSQ1.LOGS.LOGCOPY1.DS001， CSQ1.LOGS.LOGCOPY1.DS002 等。

程序

1. 發出下列指令來建立通用設定檔 (如果不存在的話):

```
ADDSD 'CSQ1.LOGS.*' UACC(NONE)
```

2. 發出下列指令，以允許佇列管理程式使用者變更設定檔的存取權:

```
PERMIT 'CSQ1.LOGS.*' ID(QMCSQ1) ACCESS(ALTER)
```

   此外，允許任何管理使用者所需的適當存取權。
3. 發出下列指令，以新增具有加密金鑰標籤的 DFP 區段:

```
ALTDSD 'CSQ1.LOGS.*' DFP(RESOWNER(QMCSQ1) DATAKEY(CSQ1DSKY))
```

   註: 您必須使用 配置佇列管理程式的資料集加密金鑰中所使用的相同加密金鑰。
4. 發出下列指令來重新整理通用資料集設定檔:

```
SETROPTS GENERIC(DATASET) REFRESH
```

5. 將每一個日誌資料集重新命名為備份，然後使用 IDCAMS 重建並還原資料。 下列 JCL 片段會轉換 CSQ1.LOGS.LOGCOPY1.DS001:

   a) 將資料集重新命名為備份

```
//RENAME   EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* RENAME DATASET TO BACKUP                                    */
  /*-------------------------------------------------------------*/
  ALTER 'CSQ1.LOGS.LOGCOPY1.DS001'                  -
        NEWNAME('CSQ1.BAK.LOGS.LOGCOPY1.DS001')
```

   b) 重新定義資料集。

   新資料集將根據 RACF 設定檔進行加密。

   註: 將 + + EXTDCLASS + + 取代為您要用於資料集的延伸格式資料類別名稱。

```
//REDEFINE EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* REDEFINE THE DATASET                                        */
  /*-------------------------------------------------------------*/
  DEFINE CLUSTER                                    -
         (NAME(CSQ1.LOGS.LOGCOPY1.DS001)            -
         LINEAR                                     -
         SHAREOPTIONS(2 3)                          -
         MODEL(CSQ1.BAK.LOGS.LOGCOPY1.DS001)        -
         DATACLAS(++EXTDCLASS++))
```

   c) 將資料從備份複製到重建的資料集。

   此步驟會加密資料:

```
//RESTORE  EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* RESTORE DATA INTO ENCRYPTED LOG                             */
  /*-------------------------------------------------------------*/
  REPRO INDATASET(CSQ1.BAK.LOGS.LOGCOPY1.DS001)     -
        OUTDATASET(CSQ1.LOGS.LOGCOPY1.DS001)
```

下一步

針對所有作用中日誌資料集重複步驟 第 394 頁的『5』 。

只需要單一加密金鑰，所有資料集都可以與相同的金鑰標籤相關聯。

重新啟動佇列管理程式 CSQ1。 使用 DISPLAY LOG 指令的輸出來驗證日誌資料集是否已加密。

z/OS V 9.2.0 佇列共用群組中 z/OS 資料集加密的考量

佇列共用群組 (QSG) 中的每一個佇列管理程式都必須能夠讀取 QSG 中每一個其他佇列管理程式的日誌、BSDS 及共用訊息資料集 (SMDS) 。

這表示 QSG 成員可以在其上執行的每一個系統都必須符合 z/OS 資料集加密的需求，而且用來保護 QSG 中每一個佇列管理程式的資料集的所有金鑰標籤及加密金鑰都必須在每一個系統上可用。

IBM MQ for z/OS 9.1.4 之前的佇列管理程式無法存取已加密的作用中日誌資料集。

IBM MQ for z/OS 9.1.5 之前的佇列管理程式無法存取已加密的 SMDS。

在使用 z/OS 資料集加密之前，您應該將 QSG 中的所有佇列管理程式至少移轉至 IBM MQ for z/OS 9.1.5。

如果 QSG 中的佇列管理程式以任何已加密的作用中日誌資料集啟動，且 QSG 中的任何其他佇列管理程式已啟動，但並非前次以支援已加密作用中日誌的 IBM MQ for z/OS 版本啟動，則具有已加密作用中日誌的佇列管理程式會異常終止，異常終止碼為 5C6-00F50033。

您可以透過下列方式，將 QSG 轉換為使用已加密的作用中日誌及 SMDS，而無需完全中斷:

1. 依序將每一個佇列管理程式至少移轉至 IBM MQ for z/OS 9.1.5。
2. 依序將每一個佇列管理程式的作用中日誌轉換成已加密資料集。 這需要先關閉再重新啟動佇列管理程式。

   同時，也可能針對已加密資料集啟用頁面集及保存日誌，但這不會影響 QSG 移轉。

   第 392 頁的『如何加密佇列管理程式作用中日誌的範例』 中說明轉換每一個資料集的程序
3. 依序將 SMDS 轉換為每一個個別 CF 結構的已加密資料集:
   a. 發出 RESET SMDS (*) ACCESS (DISABLED) CFSTRUCT (STRUCTURE-NAME) 指令，以暫停佇列管理程式對 SMDS 的存取權。

      請注意，在此期間，與 SMDS 相關聯的共用佇列上的資料暫時無法使用。
   b. 使用 第 392 頁的『如何加密佇列管理程式作用中日誌的範例』中說明的程序，將組成 SMDS 的每一個資料集轉換為已加密資料集。
   c. 發出 RESET SMDS (*) ACCESS (ENABLED) CFSTRUCT (structure-name) 指令，以回復 SMDS 的佇列管理程式存取權。

小心: 您應該在轉換日誌之前完全關閉佇列管理程式，而且在轉換期間可能無法進行連結機能結構回復，因為作用中日誌資料集將暫時無法使用。

z/OS V 9.2.0 使用 z/OS 資料集加密時的舊版移轉考量

當反向移轉具有一或多個加密資料集的佇列管理程式時，您需要考量下列事項。

下列 IBM MQ for z/OS 資料集支援 z/OS 資料集加密:

- 作用中日誌資料集
- 保存日誌資料集
- 頁集
- BSDS
- SMDS
- CSQINP* 資料集

BSDS、保存日誌或 CSINP* 資料集沒有舊版移轉考量。

不過，有一些考量

- SMDS
- 頁集及
- 作用中日誌

資料集，因為在 IBM MQ for z/OS 9.1.0 及更早的長期支援版本中不支援搭配使用這些資料集與 z/OS 資料集加密。

在舊版移轉之前，需要移除 SMDS、 頁集及作用中日誌資料集的所有加密原則，並將資料解密。 此程序在第 395 頁的『從資料集移除資料集加密』中說明。

小心: 如果要向後移轉的佇列管理程式是佇列共用群組 (QSG) 的一部分，請先閱讀 第 397 頁的『佇列共用群組考量』區段。

從資料集移除資料集加密

此範例說明如何從日誌資料集 CSQ1.LOGS.LOGCOPY1.DS001。 您可以對 V 9.2.0 SMDS 及 頁集使用對等處理程序。

此範例假設:

- RACF 是 SAF

- 已停止使用資料集的佇列管理程式
- 加密金鑰標籤已與通用 RACF 設定檔 CSQ1.LOGS.*

執行下列程序:

1. 將資料從資料集複製到備份資料集。
   a. 定義與加密金鑰標籤無關的備份資料集。

      註: 將 + + EXTDCLASS + + 取代為您要用於資料集的延伸格式資料類別名稱。

      ```
      //DEFINE EXEC PGM=IDCAMS,REGION=0M
      //SYSPRINT DD SYSOUT=*
      //SYSIN    DD *
        /*-------------------------------------------------------------*/
        /* DEFINE UNENCRYPTED DATA SET                                 */
        /*-------------------------------------------------------------*/
        DEFINE CLUSTER                                  -
               (NAME(CSQ1.BAK.LOGS.LOGCOPY1.DS001)      -
               LINEAR                                   -
               SHAREOPTIONS(2 3)                        -
               MODEL(CSQ1.LOGS.LOGCOPY1.DS001)          -
               DATACLAS(++EXTDCLASS++))
      /*
      ```

   b. 將原始資料集中的資料複製到備份。 此步驟會解密資料。

      ```
      //COPY     EXEC PGM=IDCAMS,REGION=0M
      //SYSPRINT DD SYSOUT=*
      //SYSIN    DD *
        /*-------------------------------------------------------------*/
        /* COPY DATA INTO UNENCRYPTED DATA SET                         */
        /*-------------------------------------------------------------*/
        REPRO INDATASET(CSQ1.LOGS.LOGCOPY1.DS001)          -
              OUTDATASET(CSQ1.BAK.LOGS.LOGCOPY1.DS001)
      /*
      ```

   c. 刪除原始資料集

      ```
      //DELETE   EXEC PGM=IDCAMS,REGION=0M
      //SYSPRINT DD SYSOUT=*
      //SYSIN    DD *
        /*-------------------------------------------------------------*/
        /* DELETE ORIGINAL                                             */
        /*-------------------------------------------------------------*/
        DELETE ('CSQ1.LOGS.LOGCOPY1.DS001')
      /*
      ```

   d. 將備份重新命名為原始資料集名稱。 資料仍未加密

      ```
      //RENAME   EXEC PGM=IDCAMS,REGION=0M
      //SYSPRINT DD SYSOUT=*
      //SYSIN    DD *
        /*-------------------------------------------------------------*/
        /* RENAME UNENCRYPTED DATA SET                                 */
        /*-------------------------------------------------------------*/
        ALTER CSQ1.BAK.LOGS.LOGCOPY1.DS001'                -
               NEWNAME('CSQ1.LOGS.LOGCOPY1.DS001)
        ALTER 'CSQ1.BAK.LOGS.LOGCOPY1.DS001.*'              -
               NEWNAME('CSQ1.LOGS.LOGCOPY1.DS001.*')
      /*
      ```

2. 選擇性地針對透過 CSQ1.LOGS.* 通用設定檔。
3. 選擇性地，如果所有資料集都與 CSQ1.LOGS.* 已解密通用設定檔，請發出下列指令來移除與通用設定檔相關聯的 DATAKEY

   ```
   ALTDSD 'CSQ1.LOGS.*' DFP(RESOWNER(QMCSQ1) DATAKEY(CSQ1DSKY))
   ```

4. 發出下列指令來重新整理通用資料集設定檔:

   ```
   SETROPTS GENERIC(DATASET) REFRESH
   ```

5. 重新啟動佇列管理程式。

6. 如果不再需要加密金鑰，請刪除它，並從 CSFKEYS 類別中刪除其相關聯的 RACF 設定檔。

佇列共用群組考量

如果屬於佇列共用群組的佇列管理程式將向後移轉至不支援資料集加密的 IBM MQ for z/OS 版本，則 QSG 中所有佇列管理程式的所有作用中日誌資料集 及 SMDS 都需要移除其資料集加密原則，並將其資料解密。

不論 QSG 的單一成員是向後移轉，還是 QSG 的所有成員，這都適用。

您可以透過下列方式，在沒有完整 QSG 中斷的情況下，達到移除加密原則及解密資料的目標:

1. 依序關閉 QSG 中的每一個佇列管理程式，移除加密原則並使用 第 395 頁的『從資料集移除資料集加密』 中說明的處理程序來解密其作用中日誌中的資料。

   如果要向後移轉佇列管理程式，此時也應該將其頁集解密。 然後重新啟動佇列管理程式。

2. V 9.2.0 依序移除每一個個別 CF 結構的加密原則及解密 SMDS 資料:

   a. 發出指令

   ```
   RESET SMDS(*) ACCESS(DISABLED) CFSTRUCT(structure-name)
   ```

   以暫停佇列管理程式對 SMDS 的存取權。 在此期間，與 SMDS 相關聯的共用佇列上的資料將暫時無法使用。

   b. 針對組成 SMDS 的每一個資料集，遵循 第 395 頁的『從資料集移除資料集加密』 中的處理程序。

   c. 發出指令

   ```
   RESET SMDS(*) ACCESS(ENABLED) CFSTRUCT(structure-name)
   ```

   以回復佇列管理程式對 SMDS 的存取權。

搭配使用 z/OS 資料集加密與不支援它的佇列管理程式

如果您意外將佇列管理程式向後移轉至不支援資料集加密的 IBM MQ for z/OS 版本，並忘記移除加密原則並解密在佇列管理程式嘗試存取資料集時收到錯誤的資料。

此錯誤視資料集類型而定，並顯示在下表中。

註: 如果發生其中一個以上錯誤，您需要遵循 第 395 頁的『從資料集移除資料集加密』 中針對受影響資料集所說明的處理程序。 可以在不變更 IBM MQ for z/OS 版本的情況下執行這些動作。

資料集	如果佇列管理程式不支援 z/OS 資料集加密，則會發生錯誤
頁集 0	在佇列管理程式啟動時異常終止 5C6-00C91400
頁集 1-99	MQRC 2193 存取頁集時發生「頁集錯誤」，例如，在 MQPUT 上
作用中日誌	在佇列管理程式啟動時異常終止 5C6-00E80084
V 9.2.0 SMDS	訊息 IEC161I-122 記載「資料集具有 KEYLABEL，但使用者未指定應用程式可以處理加密」。 SMDS 標示 AVAIL (ERROR)。

訊息的資料完整性

若要維護資料完整性，您可以使用各種類型的使用者結束程式，為您的訊息提供訊息摘要或數位簽章。

資料完整性

在訊息中實作資料完整性

當您使用 TLS 時，您選擇的 CipherSpec 會決定企業中的資料完整性層次。 如果您使用 IBM MQ Advanced Message Service (AMS)，則可以指定唯一訊息的完整性。

在訊息結束程式中實作資料完整性

訊息可以由通道傳送端的訊息結束程式進行數位簽署。 然後，通道接收端的訊息出口可以檢查數位簽章，以偵測訊息是否已刻意修改。

可以使用訊息摘要而非數位簽章來提供部分保護。 訊息摘要可能有效防止隨意或任意竄改，但不會阻止更知情的個人變更或取代訊息，並為其產生全新摘要。 如果用來產生訊息摘要的演算法是眾所周知的演算法，則尤其如此。

在傳送及接收結束程式中實作資料完整性

在訊息通道上，訊息結束程式更適合提供此服務，因為訊息結束程式可以存取整個訊息。 在 MQI 通道上， MQI 呼叫的參數可能包含需要保護的應用程式資料，且只有傳送及接收結束程式才能提供此保護。

在 API 結束程式或 API 交互結束程式中實作資料完整性

當傳送端應用程式放置訊息時，訊息可以由 API 或 API 交互結束程式進行數位簽署。 然後，當接收端應用程式擷取訊息時，第二個結束程式可以檢查數位簽章，以偵測訊息是否已刻意修改。

可以使用訊息摘要而非數位簽章來提供部分保護。 訊息摘要可能有效防止隨意或任意竄改，但不會阻止更知情的個人變更或取代訊息，並為其產生全新摘要。 如果用來產生訊息摘要的演算法是眾所周知的演算法，則尤其如此。

進一步資訊

如需確保資料完整性的相關資訊，請參閱 第 348 頁的『啟用 CipherSpecs』 一節。

相關工作

使用 TLS 連接兩個佇列管理程式

將用戶端安全連接至佇列管理程式

審核

您可以使用事件訊息來檢查安全侵入或嘗試侵入。 您也可以使用 IBM MQ Explorer 來檢查系統的安全。

若要偵測嘗試執行未獲授權的動作 (例如連接至佇列管理程式或將訊息放置在佇列上) ，請檢查佇列管理程式所產生的事件訊息，特別是權限事件訊息。 如需佇列管理程式事件訊息的相關資訊，請參閱 佇列管理程式事件，以及一般事件監視的相關資訊，請參閱 事件監視。

保持叢集安全

授權或防止佇列管理程式加入叢集或在叢集佇列上放置訊息。 強制佇列管理程式離開叢集。 在配置叢集的 TLS 時，請考量一些其他考量。

停止傳送訊息的未獲授權佇列管理程式

防止未獲授權的佇列管理程式使用通道安全結束程式將訊息傳送至佇列管理程式。

開始之前

叢集作業不會影響安全結束程式運作的方式。 您可以使用在分散式佇列環境中的相同方式來限制對佇列管理程式的存取。

關於這項作業

防止選取的佇列管理程式將訊息傳送至佇列管理程式:

程序

1. 在 CLUSRCVR 通道定義上定義通道安全跳出程式。
2. 撰寫程式，以鑑別嘗試在叢集接收端通道上傳送訊息的佇列管理程式，並在未獲授權時拒絕它們存取。

下一步

通道安全跳出程式在 MCA 起始及終止時呼叫。

停止在佇列上放置訊息的未獲授權佇列管理程式

使用叢集接收端通道上的通道放置權限屬性，可停止未獲授權的佇列管理程式將訊息放置在佇列上。 使用 z/OS 上的 RACF 或其他平台上的 OAM 來檢查訊息中的使用者 ID ，以授權遠端佇列管理程式。

關於這項作業

使用平台的安全機能，以及 IBM MQ 中的存取控制機制，來控制佇列的存取權。

程序

1. 若要防止某些佇列管理程式將訊息放置在佇列上，請使用平台上可用的安全機能。

   例如：

   - RACF 或 IBM MQ for z/OS 上的其他外部安全管理程式
   - 其他平台上的物件權限管理程式 (OAM)。
2. 在 CLUSRCVR 通道定義上使用放置權限 PUTAUT 屬性。

   PUTAUT 屬性可讓您指定要使用哪些使用者 ID 來建立將訊息放入佇列的權限。

   PUTAUT 屬性上的選項如下：

   DEF
   使用預設使用者 ID。 在 z/OS 上，檢查可能涉及使用從網路收到的使用者 ID 以及衍生自 MCAUSER 的使用者 ID。

   CTX
   在與訊息相關聯的環境定義資訊中使用使用者 ID。 在 z/OS 上，檢查可能涉及使用從網路收到的使用者 ID 及/或衍生自 MCAUSER 的使用者 ID。 如果鏈結受到信任和鑑別，請使用這個選項。

   ONLYMCA (僅限 z/OS)
   至於 DEF ，則不會使用從網路收到的任何使用者 ID。 如果鏈結不受信任，請使用此選項。 您只容許對它執行一組特定的動作，這些動作是針對 MCAUSER 所定義。

   ALTMCA (僅限 z/OS)
   對於 CTX，但不會使用從網路收到的任何使用者 ID。

授權將訊息放置在遠端叢集佇列上

在 z/OS 上，設定使用 RACF 放入叢集佇列的授權。 在其他平台上，授權存取以連接至佇列管理程式，以及放置至那些佇列管理程式上的佇列。

關於這項作業

預設行為是對 SYSTEM.CLUSTER.TRANSMIT.QUEUE 執行存取控制。 請注意，即使您使用多個傳輸佇列，也會套用此行為。

只有在您依照 安全段落 主題中的說明，將 qm.ini 檔中的 **ClusterQueueAccessControl** 屬性配置成 *RQMName*，並重新啟動佇列管理程式之後，這個主題中所說明的特定行為才適用。

程序

- 若為 z/OS，請發出下列指令：

```
RDEFINE MQQUEUE QMgrName.QUEUE. QueueName UACC(NONE)
PERMIT QMgrName.QUEUE. QueueName CLASS(MQADMIN) ID(GroupName) ACCESS(UPDATE)
```

- 若為 AIX, Linux, and Windows 系統，請發出下列指令：

```
setmqaut -m QMgrName -t qmgr -g GroupName +connect
setmqaut -m QMgrName -t queue -n QueueName -g GroupName -all +put
```

- 若為 IBM i，請發出下列指令:

```
GRTMQMAUT OBJ(' QMgrName ') OBJTYPE(*MQM) USER(GroupName) AUT(*CONNECT)
GRTMQMAUT OBJ(' QueueName ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
```

使用者只能將訊息放置到指定的叢集佇列，而不能放置其他叢集佇列。

變數名稱具有下列意義:

QMgrName
佇列管理程式的名稱。 在 z/OS 上，此值也可以是佇列共用群組的名稱。

GroupName
要授與存取權的群組名稱。

QueueName
要變更其授權的佇列或通用設定檔名稱。

下一步

如果您在叢集佇列上放置訊息時指定回覆目的地佇列，則消費端應用程式必須具有傳送回覆的權限。 遵循第 330 頁的『授與將訊息放入遠端叢集佇列的權限』中的指示來設定此權限。

相關概念
qm.ini 中的安全段落

防止佇列管理程式加入叢集

如果惡意佇列管理程式加入叢集，則很難阻止它接收您不想要它接收的訊息。

程序

如果您想要確保只有特定授權佇列管理程式加入叢集，您可以選擇三種技術:

- 使用通道鑑別記錄，您可以根據遠端 IP 位址、遠端佇列管理程式名稱或遠端系統提供的 TLS 識別名稱來封鎖叢集通道連線。
- 撰寫結束程式以防止未獲授權的佇列管理程式寫入 SYSTEM.CLUSTER.COMMAND.QUEUE。 請勿限制存取 SYSTEM.CLUSTER.COMMAND.QUEUE ，使任何佇列管理程式都無法寫入其中，否則您會阻止任何佇列管理程式加入叢集。
- CLUSRCVR 通道定義上的安全結束程式。

叢集通道上的安全結束程式

在叢集通道上使用安全結束程式時的額外考量。

關於這項作業

第一次啟動叢集傳送端通道時，它會使用系統管理者手動定義的屬性。 當通道停止並重新啟動時，它會從對應的叢集接收端通道定義中挑選屬性。 原始叢集傳送端通道定義會改寫為新屬性，包括 SecurityExit 屬性。

程序

1. 您必須同時在通道的叢集傳送端和叢集接收端定義安全結束程式。

   即使從叢集接收端定義傳送安全結束程式名稱，也必須使用安全結束程式信號交換來建立起始連線。
2. 在安全結束程式中驗證 MQCXP 結構中的 PartnerName 。

   只有在友機佇列管理程式已獲授權時，結束程式才必須容許通道啟動
3. 將叢集接收端定義上的安全結束程式設計成接收端起始。

4. 如果您將它設計為起始傳送端，則未獲授權且沒有安全結束程式的佇列管理程式可以加入叢集，因為不會執行安全檢查。

   在停止並重新啟動通道之後，才能從叢集接收端定義傳送 SCYEXIT 名稱，並進行完整安全檢查。
5. 如果要檢視目前使用中的叢集傳送端通道定義，請使用下列指令:

   ```
   DISPLAY CLUSQMGR( queue manager ) ALL
   ```

   此指令會顯示已從叢集接收端定義傳送的屬性。
6. 若要檢視原始定義，請使用下列指令:

   ```
   DISPLAY CHANNEL( channel name ) ALL
   ```
7. 如果佇列管理程式位於不同的平台上，您可能需要在叢集傳送端佇列管理程式上定義通道自動定義結束程式 CHADEXIT。

   使用通道自動定義結束程式，將 SecurityExit 屬性設為適合目標平台的格式。
8. 部署並配置 security-exit。

   z/OS

   安全結束程式載入模組必須位於通道起始程式位址空間程序的 CSQXLIB DD 陳述式中指定的資料集中。

   AIX, Linux, and Windows 系統

   - 安全結束程式動態鏈結程式庫必須位於通道定義的 SCYEXIT 屬性中指定的路徑。
   - 通道自動定義結束程式動態鏈結程式庫必須位於佇列管理程式定義的 CHADEXIT 屬性中指定的路徑。

強制不要的佇列管理程式離開叢集

在完整儲存庫佇列管理程式上發出 RESET CLUSTER 指令，以強制不要的佇列管理程式離開叢集。

關於這項作業

您可以強制不要的佇列管理程式離開叢集。 例如，如果佇列管理程式已刪除，但其叢集接收端通道仍定義給叢集。 您可能想要清理。

只有完整儲存庫佇列管理程式才有權從叢集中退出佇列管理程式。

註: 雖然使用 RESET CLUSTER 指令會強制從叢集中移除佇列管理程式，但單獨使用 RESET CLUSTER 並不會阻止佇列管理程式稍後重新加入叢集。 若要確保佇列管理程式不會重新結合叢集，請遵循 第 400 頁的『防止佇列管理程式加入叢集』中詳述的步驟。

請遵循此程序 OSLO 從叢集 NORWAY 退出佇列管理程式:

程序

1. 在完整儲存庫佇列管理程式上，發出下列指令:

   ```
   RESET CLUSTER(NORWAY) QMNAME(OSLO) ACTION(FORCEREMOVE)
   ```
2. 替代方案是在指令中使用 QMID 而非 QMNAME :

   ```
   RESET CLUSTER(NORWAY) QMID(qmid) ACTION(FORCEREMOVE)
   ```

   註: QMID 是一個字串，因此 qmid 的值應該以單引號括住，例如 QMID('FR01_2019-07-15_14.42.42')。

結果

強制移除的佇列管理程式不會變更; 其本端叢集定義會顯示它位於叢集中。 所有其他佇列管理程式中的定義不會顯示在叢集中。

防止佇列管理程式接收訊息

您可以使用結束程式來防止叢集佇列管理程式接收未獲授權接收的訊息。

關於這項作業

很難停止屬於叢集成員的佇列管理程式來定義佇列。 惡意佇列管理程式會加入叢集，並定義其自己的叢集其中一個佇列實例。 現在，它可以接收未獲授權接收的訊息。 若要防止佇列管理程式接收訊息，請使用程序中提供的下列其中一個選項。

程序

- 每一個叢集傳送端通道上的通道結束程式。 跳出程式會使用連線名稱來判斷目的地佇列管理程式是否適合傳送訊息。
- 叢集工作量結束程式，使用目的地記錄來判斷目的地佇列及佇列管理程式是否適合傳送訊息。

SSL/TLS 和叢集

為叢集配置 TLS 時，請注意 CLUSRCVR 通道定義會以自動定義的 CLUSSDR 通道形式延伸到其他佇列管理程式。 如果 CLUSRCVR 通道使用 TLS，您必須在使用該通道進行通訊的所有佇列管理程式上配置 TLS。

如需 TLS 的相關資訊，請參閱 第 20 頁的『IBM MQ 中的 TLS 安全通訊協定』。 這裡的建議通常適用於叢集通道，但您可能想要對下列項目提供一些特殊考量:

在 IBM MQ 叢集中，特定 CLUSRCVR 通道定義會經常延伸到許多其他佇列管理程式，並在其中轉換成自動定義的 CLUSSDR。 隨後會使用自動定義的 CLUSSDR 來啟動 CLUSRCVR 的通道。 如果針對 TLS 連線功能配置 CLUSRCVR，則下列考量適用:

- 所有想要與此 CLUSRCVR 通訊的佇列管理程式都必須具有 TLS 支援的存取權。 此 TLS 供應必須支援通道的 CipherSpec。
- 自動定義的叢集傳送端通道所傳送至的不同佇列管理程式將各有不同的相關聯識別名稱。 如果要在 CLUSRCVR 上使用識別名稱同層級檢查，則必須設定它，以便能夠順利比對所有可接收的識別名稱。

  例如，假設所有將管理叢集傳送端通道 (將連接至特定 CLUSRCVR) 的佇列管理程式都有相關聯的憑證。 讓我們也假設所有這些憑證中的識別名稱都將國家定義為 UK，組織定義為 IBM，組織單位定義為 IBM MQ Development，並且都具有格式為 DEVT.QMnnn 的通用名稱，其中 nnn 是數值。

  在此情況下， CLUSRCVR 上的 SSLPEER 值 `C=UK, O=IBM, OU=IBM MQ Development, CN=DEVT.QM*` 將容許所有必要的叢集傳送端通道順利連接，但會防止不想要的叢集傳送端通道連接。
- 如果使用自訂 CipherSpec 字串，請注意並非在所有平台上都容許自訂字串格式。 例如， CipherSpec 字串 RC4_SHA_US 在 IBM i 上具有值 05，但在 AIX, Linux, and Windows 系統上不是有效的規格。 因此，如果在 CLUSRCVR 上使用自訂 SSLCIPH 參數，則所有產生的自動定義叢集傳送端通道都應該位於基礎 TLS 支援實作此 CipherSpec 且可以使用自訂值來指定它的平台上。 如果您無法為 SSLCIPH 參數選取可在整個叢集中瞭解的值，則需要通道自動定義結束程式才能將它變更為所使用平台將瞭解的內容。 可能的話，請使用文字 CipherSpec 字串 (例如 TLS_RSA_WITH_AES_128_CBC_SHA)。

SSLCRLNL 參數適用於個別佇列管理程式，且不會延伸到叢集內的其他佇列管理程式。

將叢集佇列管理程式及通道升級至 SSL/TLS

一次升級一個叢集通道，變更 CLUSSDR 通道之前的所有 CLUSRCVR 通道。

開始之前

請考量下列考量，因為這些可能會影響您為叢集選擇的 CipherSpec:

- 部分 CipherSpecs 無法在所有平台上使用。 請小心選擇叢集中所有佇列管理程式支援的 CipherSpec。

- 部分 CipherSpecs 在現行 IBM MQ 版本中可能是新的，在舊版中可能不受支援。 包含在不同 MQ 版次上執行之佇列管理程式的叢集，只能使用每一個版次所支援的 CipherSpecs 。

  若要在叢集內使用新的 CipherSpec ，您必須先將所有叢集佇列管理程式移轉至現行版本。
- 部分 CipherSpecs 需要使用特定類型的數位憑證，特別是使用「橢圓曲線加密法」的憑證。

  小心: 在您要結合成叢集一部分的佇列管理程式上，無法混合使用橢圓曲線簽署憑證和 RSA 簽署憑證。

  叢集中的佇列管理程式必須全部使用 RSA 簽署憑證，或全部使用 EC 簽署憑證，而不是兩者混合使用。

  如需相關資訊，請參閱 第 37 頁的『IBM MQ 中的數位憑證及 CipherSpec 相容性』。

將叢集中的所有佇列管理程式升級至 IBM MQ V8 或更高版本 (如果它們尚未處於這些層次)。 配送憑證及金鑰，以便 TLS 從其中每一個憑證及金鑰運作。

如果您想要升級至或使用任何別名 CipherSpecs (ANY_TLS13、 ANY_TLS13_OR_HIGHER、 ANY_TLS12、ANY_TLS12_OR_HIGHER 等)，則必須將叢集中的所有 IBM MQ for Multiplatforms 佇列管理程式升級至 IBM MQ 9.1.4 或更高版本 V 9.2.0，並將叢集中的所有 IBM MQ for z/OS 佇列管理程式升級至 IBM MQ for z/OS 9.2.0 或更高版本。

關於這項作業

變更 CLUSSDR 通道之前的 CLUSRCVR 通道。

程序

1. 按您喜歡的任何順序將 CLUSRCVR 通道切換至 TLS ， 一次變更一個 CLUSRCVR ， 並容許變更在變更下一個之前流經叢集。

   重要: 請確定在現行通道的變更已分散到整個叢集之前，不要變更反向路徑。
2. 選擇性的: 將所有手動 CLUSSDR 通道切換至 TLS。

   除非您搭配使用 REFRESH CLUSTER 指令與 REPOS(YES) 選項，否則這不會對叢集的作業產生任何影響。

   註: 若為大型叢集，在叢集進行中時使用 **REFRESH CLUSTER** 指令可能會對叢集造成干擾，之後每隔 27 天，叢集物件會自動將狀態更新傳送至所有感興趣的佇列管理程式。 請參閱在大型叢集中重新整理可能影響叢集的效能及可用性。
3. 請使用 DISPLAY CLUSQMGR 指令來確保新的安全配置已在整個叢集中延伸。
4. 重新啟動通道以使用 TLS，並執行 REFRESH SECURITY TYPE(SSL)。

相關概念

第 348 頁的『啟用 CipherSpecs』
在 **DEFINE CHANNEL** 或 **ALTER CHANNEL** MQSC 指令中使用 **SSLCIPH** 參數來啟用 CipherSpec 。

第 37 頁的『IBM MQ 中的數位憑證及 CipherSpec 相容性』
本主題提供如何透過概述 CipherSpecs 與 IBM MQ 中數位憑證之間的關係，為安全原則選擇適當的 CipherSpecs 及數位憑證的相關資訊。

相關資訊

叢集作業：使用 REFRESH CLUSTER 最佳作法

在叢集佇列管理程式及通道上停用 SSL/TLS

若要關閉 TLS ， 請將 SSLCIPH 參數設為 ' '。 在叢集通道上個別停用 TLS ， 在叢集傳送端通道之前變更所有叢集接收端通道。

關於這項作業

一次變更一個叢集接收端通道，並容許變更在變更下一個之前流經叢集。

重要: 在現行通道的變更已分散到整個叢集之前，請確定您不會變更反向路徑。

程序

1. 將 SSLCIPH 參數的值設為 ' '(單引號中的空字串) IBM i，或 IBM i 上的 *NONE。
   您可以按想要的任何順序關閉叢集接收端通道上的 TLS。

   請注意，變更會在您保持 TLS 作用中的通道上以相反方向流動。
2. 使用指令 **DISPLAY CLUSQMGR(*)** ALL，檢查新值是否反映在所有其他佇列管理程式中。
3. 在所有手動叢集傳送端通道上關閉 TLS。
   除非您搭配使用 **REFRESH CLUSTER** 指令與 REPOS（YES）選項，否則對叢集作業沒有任何影響。

   對於大型叢集，在叢集進行中時使用 **REFRESH CLUSTER** 指令可能會對叢集造成干擾，然後在叢集物件自動將狀態更新傳送至所有相關佇列管理程式時，會定期再次造成干擾。 如需相關資訊，請參閱 在大型叢集中重新整理可能會影響叢集的效能及可用性。
4. 停止並重新啟動叢集傳送端通道。

發佈/訂閱安全

發佈/訂閱中所涉及的元件及互動，說明為下列更詳細的說明及範例的簡介。

發佈及訂閱主題涉及許多元件。 它們之間的部分安全關係在 第 404 頁的圖 22 中說明，並在下列範例中說明。

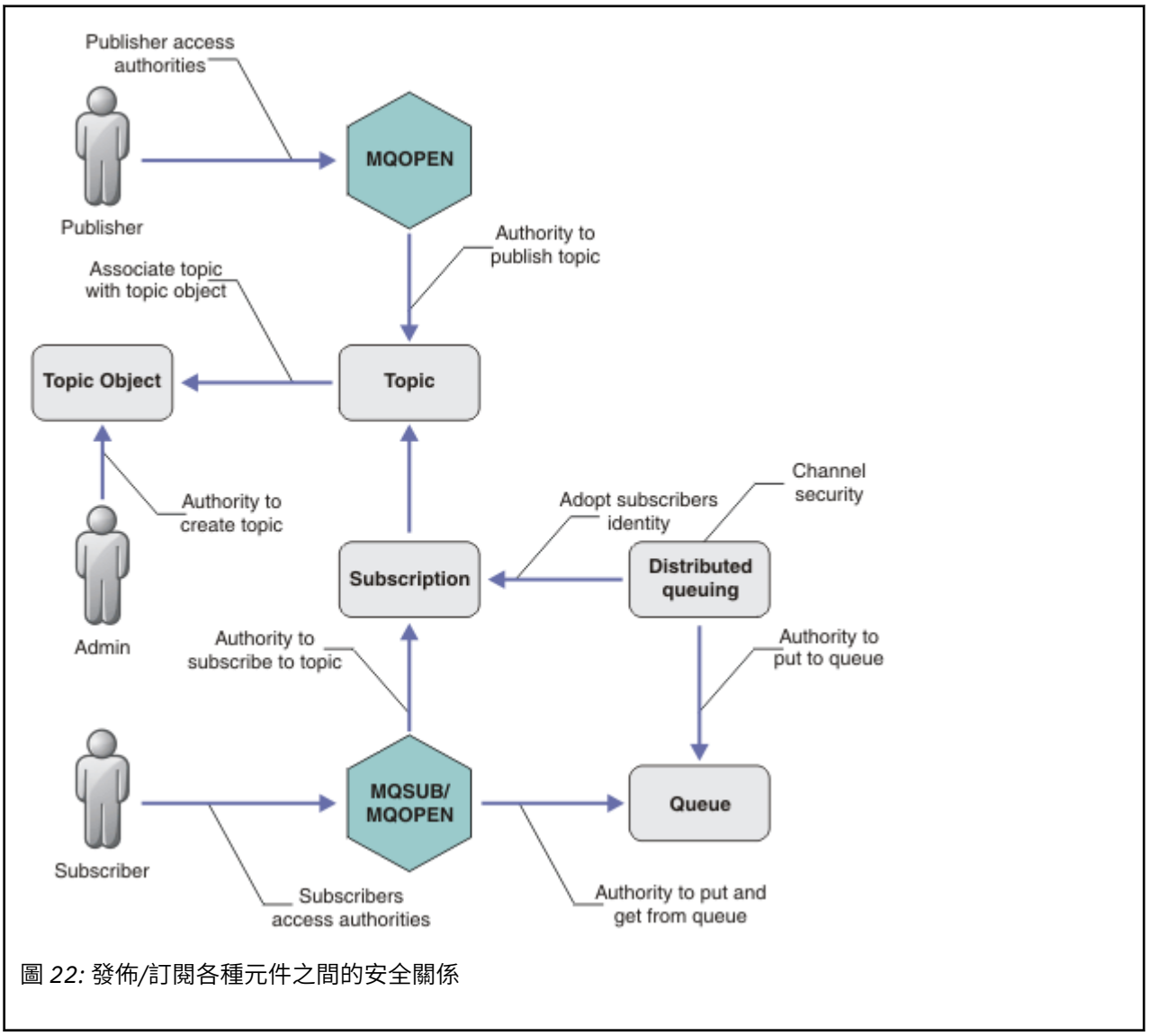


圖 22: 發佈/訂閱各種元件之間的安全關係

主題

主題由主題字串識別，且通常組織成樹狀結構，請參閱 主題樹狀結構。 您需要建立主題與主題物件的關聯，以控制對主題的存取權。 第 406 頁的『主題安全模型』 說明如何使用主題物件來保護主題的安全。

管理主題物件

您可以使用指令 **setmqaut** 搭配管理主題物件清單，來控制對主題具有存取權的人員，以及對主題具有存取權的用途。 請參閱範例 第 410 頁的『授與使用者訂閱主題的存取權』 和 第 416 頁的『授與使用者發佈至主題的存取權』。 z/OS 如需控制 z/OS 上主題物件的存取權，請參閱 主題安全的設定檔。

訂閱

訂閱一或多個主題，方法是建立訂閱來提供主題字串 (可包含萬用字元)，以符合發佈的主題字串。 如需進一步詳細資料，請參閱:

使用主題物件訂閱

第 407 頁的『使用主題物件名稱訂閱』

使用主題訂閱

第 408 頁的『使用主題節點不存在的主題字串來訂閱』

使用含萬用字元的主題來訂閱

第 408 頁的『使用包含萬用字元的主題字串來訂閱』

訂閱包含訂閱者身分及要放置發佈資訊之目的地佇列身分的相關資訊。 它還包含如何將發佈放置在目的地佇列上的相關資訊。

除了定義哪些訂閱者有權訂閱特定主題外，您還可以限制訂閱由個別訂閱者使用。 您也可以控制當發佈資訊放置到目的地佇列時，佇列管理程式會使用哪些訂閱者的相關資訊。 請參閱 第 421 頁的『訂閱安全』。

佇列

目的地佇列是要保護的重要佇列。 它是訂閱者的本端，且會將符合訂閱的發佈放置在其上。 您需要從兩個視景考量對目的地佇列的存取權:

1. 將發佈放入目的地佇列。
2. 正在從目的地佇列中取得發佈。

佇列管理程式會使用訂閱者所提供的身分，將發佈資訊放入目的地佇列。 訂閱者或已委派取得發佈作業的程式會從佇列中移除訊息。 請參閱 第 408 頁的『目的地佇列的權限』。

沒有主題物件別名，但您可以使用別名佇列作為主題物件的別名。 如果您這樣做，以及檢查使用發佈或訂閱主題的權限，佇列管理程式會檢查使用佇列的權限。

第 422 頁的『佇列管理程式之間的發佈/訂閱安全』

使用本端身分和授權，在本端佇列管理程式上檢查您發佈或訂閱主題的許可權。 授權不取決於是否定義主題，也不取決於定義主題的位置。 因此，當使用叢集主題時，您需要對叢集中的每個佇列管理程式執行主題授權。

註: 主題的安全模型與佇列的安全模型不同。 您可以在本端定義每個叢集佇列的佇列別名，以達到佇列的相同結果。

佇列管理程式會交換叢集中的訂閱。 在大部分 IBM MQ 叢集配置中，通道會配置 PUTAUT=DEF，以使用通道處理程序的權限將訊息放置在目標佇列上。 您可以修改通道配置來使用 PUTAUT=CTX，以要求訂閱使用者有權將訂閱延伸到叢集中的另一個佇列管理程式。

第 422 頁的『佇列管理程式之間的發佈/訂閱安全』 說明如何變更通道定義，以控制容許誰將訂閱延伸到叢集中的其他伺服器。

授權

您可以將授權套用至主題物件，就像佇列及其他物件一樣。 有三個您只能套用至主題的授權作業: **發佈**、 **訂閱**及 **回復** 。 如需詳細資料，請參閱 指定不同物件類型的權限。

函數呼叫

在發佈和訂閱程式中，例如在佇列程式中，當開啟、建立、變更或刪除物件時，會進行授權檢查。 當進行 MQPUT 或 MQGET MQI 呼叫來放置及取得發佈資訊時，不會進行檢查。

若要發佈主題，請對主題執行 MQOPEN，這會執行授權檢查。 使用 MQPUT 指令將訊息發佈至主題控點，這不會執行授權檢查。

如果要訂閱主題，通常您會執行 MQSUB 指令來建立或回復訂閱，以及開啟目的地佇列來接收發佈資訊。或者，執行個別 MQOPEN 以開啟目的地佇列，然後執行 MQSUB 以建立或回復訂閱。

不論您使用哪一個呼叫，佇列管理程式都會檢查您是否可以訂閱主題，並從目的地佇列取得產生的發佈資訊。 如果目的地佇列未受管理，也會進行授權檢查，讓佇列管理程式能夠將發佈放置在目的地佇列上。 它會使用從相符訂閱採用的身分。 假設佇列管理程式一律能夠將發佈放置在受管理目的地佇列上。

角色

使用者參與執行發佈/訂閱應用程式的四個角色:

1. 發佈者
2. 訂閱者
3. 主題管理者
4. IBM MQ 管理者-群組成員 mqm

使用對應於發佈、訂閱及主題管理角色的適當授權來定義群組。 然後，您可以將主體指派給這些群組，授權它們執行特定的發佈和訂閱作業。

此外，您還需要將管理作業授權延伸至負責移動發佈及訂閱之佇列及通道的管理者。

主題安全模型

只有已定義的主題物件可以具有相關聯的安全屬性。 如需主題物件的說明，請參閱 管理主題物件。 安全屬性指定是否允許指定的使用者 ID 或安全群組對每一個主題物件執行訂閱或發佈作業。

安全屬性與主題樹狀結構中的適當管理節點相關聯。 在訂閱或發佈作業期間對特定使用者 ID 進行權限檢查時，所授與的權限基於相關聯主題樹狀結構節點的安全屬性。

安全屬性是存取控制清單，指出特定作業系統使用者 ID 或安全群組對主題物件具有的權限。

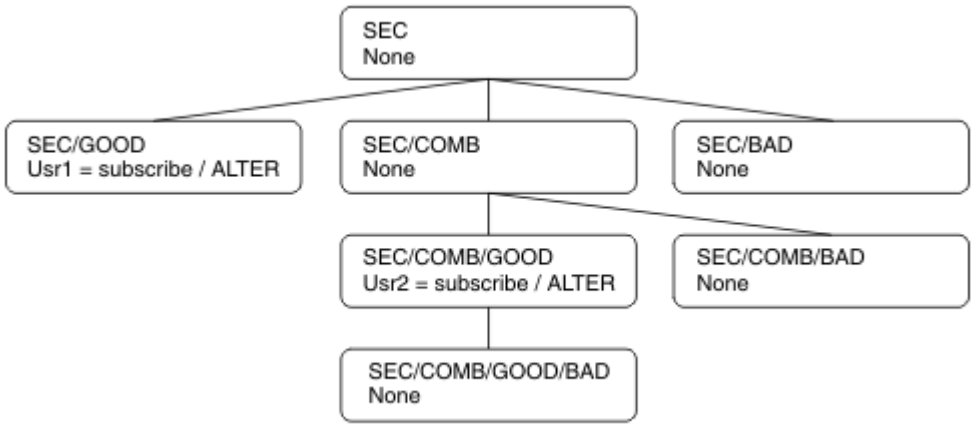
請考量下列範例，其中已使用所顯示的安全屬性或權限來定義主題物件:

表 *84:* 主題物件權限範例

主題名稱	主題字串	權限-非 z/OS	z/OS 權限
SECROOT	SEC	無	無
SECGOOD	SEC/GOOD	usr1+subscribe	ALTER HLQ.SUBSCRIBE.SECGOOD
SECBAD	SEC/BAD	無	無 HLQ.SUBSCRIBE.SECBAD
SECCOMB	SEC/COMB	無	無 HLQ.SUBSCRIBE.SECCOMB
SECCOMBB	SEC/COMB/ GOOD/BAD	無	無 HLQ.SUBSCRIBE.SECCOMBB
SECCOMBG	SEC/COMB/GOOD	usr2+subscribe	ALTER HLQ.SUBSCRIBE.SECCOMBG

表 *84:* 主題物件權限範例 *(*繼續*)*			
主題名稱	**主題字串**	**權限-非 z/OS**	**z/OS 權限**
SECCOMBN	SEC/COMB/BAD	無	無 HLQ.SUBSCRIBE.SECCOMBN

每一個節點上具有相關聯安全屬性的主題樹狀結構可以如下所示:



下列範例提供下列授權:

- 在樹狀結構 /SEC 的根節點上，沒有任何使用者在該節點上具有權限。
- usr1 已獲授與物件的訂閱權限 /SEC/GOOD
- usr2 已獲授與物件的訂閱權限 /SEC/COMB/GOOD

使用主題物件名稱訂閱

透過指定 MQCHAR48 名稱來訂閱主題物件時，會找到主題樹狀結構中對應的節點。 如果與節點相關聯的安全屬性指出使用者有權訂閱，則會授與存取權。

如果未授與使用者存取權，樹狀結構中的母節點會決定使用者是否有權在母節點層次訂閱。 如果是這樣，則會授與存取權。 如果沒有，則會考量該節點的母項。 遞迴會繼續進行，直到找到將訂閱權限授與使用者的節點為止。 當在未授與權限的情況下考量根節點時，遞迴會停止。 在後一種情況下，拒絕存取。

簡言之，如果路徑中的任何節點授與權限來訂閱該使用者或應用程式，則容許訂閱者在該節點或主題樹狀結構中該節點下方的任何位置進行訂閱。

範例中的根節點是 SEC。

如果存取控制清單指出使用者 ID 本身具有權限，或使用者 ID 所屬的作業系統安全群組具有權限，則會授與使用者訂閱權限。

例如:

- 如果 usr1 嘗試使用主題字串 SEC/GOOD 來訂閱，則容許訂閱，因為使用者 ID 有權存取與該主題相關聯的節點。 不過，如果 usr1 嘗試使用主題字串來訂閱 SEC/COMB/GOOD，則不容許訂閱，因為使用者 ID 無法存取與其相關聯的節點。
- 如果 usr2 嘗試訂閱，則會使用主題字串 SEC/COMB/GOOD 來容許訂閱，因為使用者 ID 有權存取與主題相關聯的節點。 不過，如果 usr2 嘗試訂閱 SEC/GOOD，則不容許訂閱，因為使用者 ID 沒有其相關聯節點的存取權。
- 如果 usr2 嘗試使用 SEC/COMB/GOOD/BAD 的主題字串進行訂閱，則容許訂閱，因為使用者 ID 有權存取上層節點 SEC/COMB/GOOD。
- 如果 usr1 或 usr2 嘗試使用主題字串 /SEC/COMB/BAD 來訂閱，則不容許它們訂閱，因為它們沒有與它相關聯的主題節點或該主題的上層節點的存取權。

指定不存在之主題物件名稱的訂閱作業會導致 MQRC_UNKNOWN_OBJECT_NAME 錯誤。

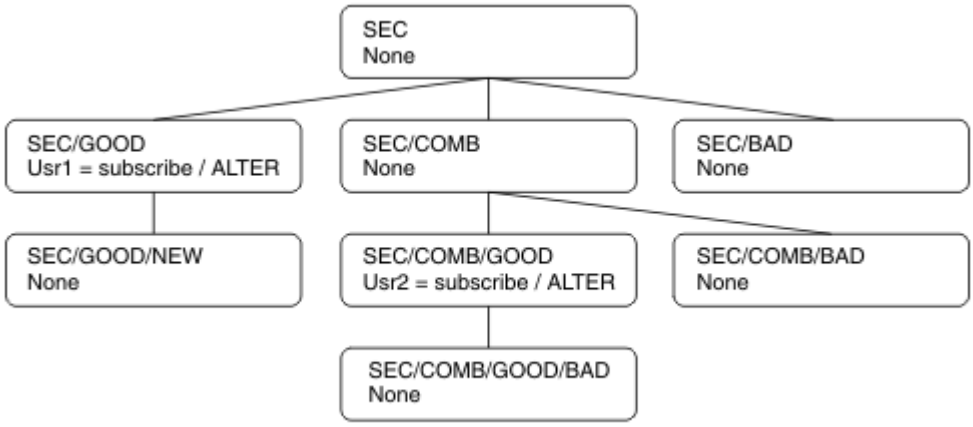
使用主題節點所在的主題字串來訂閱

此行為與以 MQCHAR48 物件名稱指定主題時的行為相同。

使用主題節點不存在的主題字串來訂閱

請考量應用程式訂閱的情況，指定代表主題樹狀結構中目前不存在之主題節點的主題字串。 會依照前一節的概述來執行權限檢查。 檢查從主題字串所代表的上層節點開始。 如果已授與權限，則會在主題樹狀結構中建立代表主題字串的新節點。

例如， usr1 嘗試訂閱主題 SEC/GOOD/NEW。 因為 usr1 具有母節點 SEC/GOOD 的存取權，所以授與權限。 樹狀結構中會建立新的主題節點，如下圖所示。 新的主題節點不是主題物件，它沒有任何直接相關聯的安全屬性; 這些屬性繼承自其母項。



使用包含萬用字元的主題字串來訂閱

請考量使用包含萬用字元的主題字串來訂閱。 對主題樹狀結構中與主題字串完整部分相符的節點執行權限檢查。

因此，如果應用程式訂閱 SEC/COMB/GOOD/*，則會執行權限檢查，如主題樹狀結構中節點 SEC/COMB/GOOD 的前兩節中所概述。

同樣地，如果應用程式需要訂閱 SEC/COMB/*/GOOD，則會在節點 SEC/COMB 上執行權限檢查。

目的地佇列的權限

訂閱主題時，其中一個參數是已開啟供輸出接收發佈的佇列控點 hobj 。

如果未指定 hobj ，但為空白，則會在下列條件適用時建立受管理佇列:

- 已指定 MQSO_MANAGED 選項。
- 訂閱不存在。
- 已指定建立。

如果 hobj 為空白，且您正在變更或回復現有的訂閱，則先前提供的目的地佇列可以是受管理或未受管理。

提出 MQSUB 要求的應用程式或使用者必須具有將訊息放入其提供之目的地佇列的權限; 實際上具有將已發佈訊息放入該佇列的權限。 權限檢查遵循佇列安全檢查的現有規則。

安全檢查包括替代使用者 ID 及環境定義安全檢查 (必要時)。 若要能夠設定任何身分環境定義欄位，您必須指定 MQSO_SET_IDENTITY_CONTEXT 選項以及 MQSO_CREATE 或 MQSO_ALTER 選項。 您無法在 MQSO_RESUME 要求上設定任何身分環境定義欄位。

如果目的地是受管理佇列，則不會對受管理目的地執行安全檢查。 如果容許您訂閱主題，則會假設您可以使用受管理目的地。

使用主題節點所在的主題名稱或主題字串進行發佈

用於發佈的安全模型與用於訂閱的安全模型相同，但萬用字元除外。 出版品不包含萬用字元; 因此沒有主題字串包含要考量的萬用字元的情況。

發佈和訂閱的權限是不同的。 使用者或群組可以具有執行一個動作的權限，而不需要能夠執行另一個動作。

透過指定 MQCHAR48 名稱或主題字串來發佈至主題物件時，會找到主題樹狀結構中的對應節點。 如果與主題節點相關聯的安全屬性指出使用者有權發佈，則會授與存取權。

如果未授與存取權，樹狀結構中的母節點會決定使用者是否有權在該層次發佈。 如果是這樣，則會授與存取權。 否則，遞迴會繼續進行，直到找到將發佈權限授與使用者的節點為止。 當在未授與權限的情況下考量根節點時，遞迴會停止。 在後一種情況下，拒絕存取。

簡言之，如果路徑中的任何節點授與權限來發佈給該使用者或應用程式，則允許發佈者在該節點或主題樹狀結構中該節點下方的任何位置發佈。

使用主題名稱或主題字串進行發佈，其中主題節點不存在

與訂閱作業一樣，當應用程式發佈時，指定代表主題樹狀結構中目前不存在的主題節點的主題字串時，會從主題字串所代表節點的母項開始執行權限檢查。 如果已授與權限，則會在主題樹狀結構中建立代表主題字串的新節點。

使用解析為主題物件的別名佇列來發佈

如果您使用解析為主題物件的別名佇列來發佈，則會在別名佇列及其解析成的基礎主題上進行安全檢查。

別名佇列上的安全檢查會驗證使用者是否有權將訊息放置在該別名佇列上，而主題上的安全檢查會驗證使用者是否可以發佈至該主題。 當別名佇列解析為另一個佇列時，不會對基礎佇列進行檢查。 主題和佇列的權限檢查執行方式不同。

關閉訂閱

如果您未在此控點下建立訂閱，當您使用 MQCO_REMOVE_SUB 選項來關閉訂閱時，會有額外的安全檢查。

會執行安全檢查，以確保您具有正確的權限來執行此動作，因為此動作會導致移除訂閱。 如果與主題節點相關聯的安全屬性指出使用者具有權限，則會授與存取權。 如果沒有，則會考量樹狀結構中的母節點，以判斷使用者是否有權關閉訂閱。 遞迴會繼續進行，直到授與權限或達到根節點為止。

定義、變更及刪除訂閱

當以管理方式而非使用 MQSUB API 要求來建立訂閱時，不會執行任何訂閱安全檢查。 管理者已透過指令獲得此權限。

會執行安全檢查，以確保發佈可以放置在與訂閱相關聯的目的地佇列上。 檢查的執行方式與 MQSUB 要求相同。

用於這些安全檢查的使用者 ID 取決於發出的指令。 如果指定 **SUBUSER** 參數，則會影響執行檢查的方式，如 第 409 頁的表 85 所示:

表 *85:* 用於指令安全檢查的使用者 *ID*

指令	已指定 **SUBUSER** 且空白	已指定且已完成 **SUBUSER**	未指定 **SUBUSER**
	使用管理者 ID		使用 LIKE 訂閱中的使用者 ID

表 *85:* 用於指令安全檢查的使用者 *ID (*繼續*)*			
指令	**已指定 SUBUSER 且空白**	**已指定且已完成 SUBUSER**	**未指定 SUBUSER**
	使用管理者 ID		使用來自.DEFAULT.SU SYSTEMB 訂閱-如果空白，請使用管理者 ID
	使用管理者 ID		使用現有訂閱中的使用者 ID

使用 DELETE SUB 指令刪除訂閱時唯一執行的安全檢查是指令安全檢查。

發佈/訂閱安全設定範例

本節說明在主題上設定存取控制的實務範例，可讓您視需要套用安全控制。

授與使用者訂閱主題的存取權

本主題是作業清單中的第一個主題，可告訴您如何由多個使用者授與對主題的存取權。

關於這項作業

此作業假設不存在任何管理主題物件，也沒有為取用或發佈定義任何設定檔。 應用程式正在建立新的訂閱，而不是回復現有的訂閱，並僅使用主題字串來執行此動作。

應用程式可以透過提供主題物件或主題字串，或兩者的組合來進行訂閱。 不論應用程式選取何種方式，其效果都是在主題樹狀結構中的某個點進行訂閱。 如果主題樹狀結構中的這個點是由管理主題物件代表，則會根據該主題物件的名稱來檢查安全設定檔。

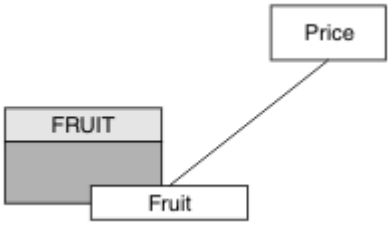


圖 *23:* 主題物件存取範例

表 *86:* 主題物件存取權範例		
主題	**需要訂閱存取權**	**Topic 物件**
計價	無使用者	無
價格/水果	USER1	水果

定義新的主題物件，如下所示:

程序

1. 發出 MQSC 指令 `DEF TOPIC(FRUIT) TOPICSTR('Price/Fruit')`。
2. 授與存取權，如下所示:

- z/OS **z/OS** :

授與使用者對 hlq.SUBSCRIBE.FRUIT 設定檔的存取權，以授與 USER1 存取權來訂閱主題 "Price/Fruit"。 使用下列 RACF 指令來執行此動作:

```
RDEFINE MXTOPIC hlq.SUBSCRIBE.FRUIT UACC(NONE)
PERMIT hlq.SUBSCRIBE.FRUIT CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
```

- 其他平台:

  授與使用者對 FRUIT 物件的存取權，以授與 USER1 存取權來訂閱主題 "Price/Fruit"。 請使用平台的授權指令來執行此動作:

  AIX, Linux, and Windows 系統

  ```
  setmqaut -t topic -n FRUIT -p USER1 +sub
  ```

  IBM i

  ```
  GRTMQAUT OBJ(FRUIT) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
  ```

結果

當 USER1 嘗試訂閱主題 "Price/Fruit" 時，結果是成功。

當 USER2 嘗試訂閱主題 "Price/Fruit" 時，結果會失敗，並出現 MQRC_NOT_AUTHORIZED 訊息，以及:

- 在 z/OS 上，在主控台上看到下列訊息，這些訊息會透過已嘗試的主題樹狀結構顯示完整安全路徑:

  ```
  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.FRUIT ...

  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
  ```

- 在其他平台上，下列授權事件:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit"
  ```

- 在 IBMi 上，發生下列授權事件:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit"
  ```

請注意，這是您看到的內容的圖解; 並非所有欄位。

授與使用者存取權，以訂閱樹狀結構內更深層的主題

本主題是作業清單中的第二個主題，可告訴您如何由多個使用者授與對主題的存取權。

開始之前

本主題使用 第 410 頁的『授與使用者訂閱主題的存取權』中說明的設定。

關於這項作業

如果管理主題物件未代表應用程式進行訂閱的主題樹狀結構中的點，請向上移動樹狀結構，直到找到最接近的上層管理主題物件為止。 會根據該主題物件的名稱來檢查安全設定檔。

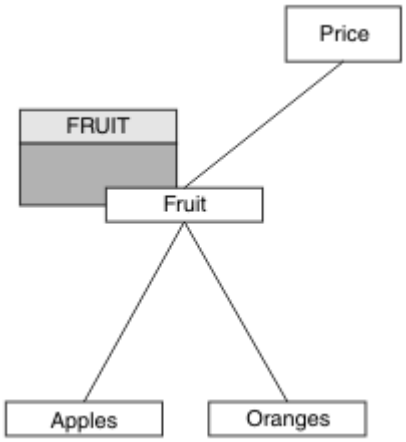


圖 *24:* 在主題樹狀結構內授與主題存取權的範例

表 *87:* 範例主題及主題物件的存取需求		
主題	**需要訂閱存取權**	**Topic 物件**
計價	無使用者	無
價格/水果	USER1	水果
價格/水果/蘋果	USER1	
價格/水果/場	USER1	

在前一項作業中， USER1 獲授與訂閱主題 "Price/Fruit" 的存取權，方法是授與它對 z/OS 上 hlq.SUBSCRIBE.FRUIT 設定檔的存取權，以及訂閱其他平台上 FRUIT 設定檔的存取權。 此單一設定檔也會授與 USER1 存取權，以訂閱 "Price/Fruit/Apples"、 "Price/Fruit/Oranges" 及 "Price/Fruit/#"。

當 USER1 嘗試訂閱主題 "Price/Fruit/Apples" 時，結果是成功。

當 USER2 嘗試訂閱主題 "Price/Fruit/Apples" 時，結果會失敗，並出現 MQRC_NOT_AUTHORIZED 訊息，以及:

- 在 z/OS 上，在主控台上看到下列訊息，這些訊息會透過已嘗試的主題樹狀結構顯示完整安全路徑:

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- 在其他平台上，下列授權事件:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Apples"
```

請注意下列項目:

- 您在 z/OS 上收到的訊息與在前一個作業中收到的訊息相同，因為相同的主題物件及設定檔正在控制存取權。
- 您在其他平台上收到的事件訊息與前一個作業中收到的事件訊息類似，但實際主題字串不同。

授與另一個使用者存取權，以便只訂閱樹狀結構中更深層的主題

本主題是作業清單中的第三個主題，可告訴您如何授與多個使用者訂閱主題的存取權。

開始之前

本主題使用 第 411 頁的『授與使用者存取權，以訂閱樹狀結構內更深層的主題』中說明的設定。

關於這項作業

在前一個作業中， USER2 已拒絕存取主題 "Price/Fruit/Apples"。 本主題告訴您如何授與對該主題的存取權，但不授與對任何其他主題的存取權。

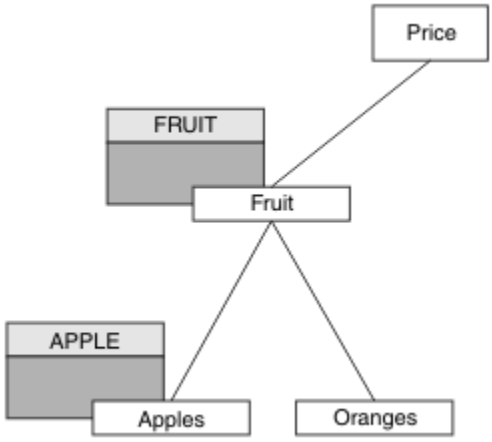


圖 *25:* 授與主題樹狀結構內特定主題的存取權

表 *88:* 範例主題及主題物件的存取需求

主題	需要訂閱存取權	Topic 物件
計價	無使用者	無
價格/水果	USER1	水果
價格/水果/蘋果	USER1 和 USER2	蘋果
價格/水果/場	USER1	

定義新的主題物件，如下所示:

程序

1. 發出 MQSC 指令 DEF TOPIC(APPLE) TOPICSTR('Price/Fruit/Apples')。
2. 授與存取權，如下所示:

   - z/OS **z/OS** :

     在前一個作業中，透過授與使用者對 hlq.SUBSCRIBE.FRUIT 設定檔的存取權， USER1 已獲授與訂閱主題 "Price/Fruit/Apples" 的存取權。

     此單一設定檔也已授與 USER1 存取權來訂閱 "Price/Fruit/Oranges" "Price/Fruit/#"，即使新增主題物件及其相關聯的設定檔，此存取權仍會保留。

     授與使用者對 hlq.SUBSCRIBE.APPLE 設定檔的存取權，以授與 USER2 存取權來訂閱主題 "Price/Fruit/Apples"。 使用下列 RACF 指令來執行此動作:

     ```
     RDEFINE MXTOPIC hlq.SUBSCRIBE.APPLE UACC(NONE)
     PERMIT hlq.SUBSCRIBE.FRUIT APPLE(MXTOPIC) ID(USER2) ACCESS(ALTER)
     ```

   - 其他平台:

     在前一個作業中，透過授與使用者對 FRUIT 設定檔的訂閱存取權， USER1 已獲授與訂閱主題 "Price/Fruit/Apples" 的存取權。

這個單一設定檔也已授與 USER1 存取權來訂閱 "Price/Fruit/Oranges" 和 "Price/Fruit/#"，即使新增主題物件及其相關聯的設定檔，這項存取權仍會保留。

授與使用者對 APPLE 設定檔的訂閱存取權，以授與 USER2 存取權來訂閱主題 "Price/Fruit/Apples"。 請使用平台的授權指令來執行此動作:

AIX, Linux, and Windows 系統

```
setmqaut -t topic -n APPLE -p USER2 +sub
```

IBM i

```
GRTMQAUT OBJ(APPLE) OBJTYPE(*TOPIC) USER(USER2) AUT(*SUB)
```

結果

在 z/OS 上，當 USER1 嘗試訂閱主題 "Price/Fruit/Apples" 時， hlq.SUBSCRIBE.APPLE 設定檔上的第一個安全檢查失敗，但在向上移動樹狀結構時， hlq.SUBSCRIBE.FRUIT 設定檔容許 USER1 訂閱，因此訂閱成功，且沒有回覆碼傳送至 MQSUB 呼叫。 不過，第一次檢查會產生 RACF ICH 訊息:

```
ICH408I USER(USER1   ) ...
  hlq.SUBSCRIBE.APPLE ...
```

當 USER2 嘗試訂閱主題 "Price/Fruit/Apples" 時，結果會成功，因為安全檢查在第一個設定檔上通過。

當 USER2 嘗試訂閱主題 "Price/Fruit/Oranges" 時，結果會失敗，並出現 MQRC_NOT_AUTHORIZED 訊息，以及:

- 在 z/OS 上，在主控台上看到下列訊息，這些訊息會透過已嘗試的主題樹狀結構顯示完整安全路徑:

  ```
  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.FRUIT ...

  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
  ```

- 在 AIX, Linux, and Windows 平台上，發生下列授權事件:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit/Oranges"
  ```

- 在 IBMi 上，發生下列授權事件:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit/Oranges"
  ```

此設定的缺點是在 z/OS 上，您會在主控台上收到其他 ICH 訊息。 如果您以不同方式保護主題樹狀結構的安全，則可以避免此情況。

變更存取控制以避免其他訊息

本主題是作業清單中的第四個主題，告訴您如何授與多個使用者訂閱主題的存取權，以及避免 z/OS 上的其他 RACF ICH408I 訊息。

開始之前

本主題加強 第 412 頁的『授與另一個使用者存取權，以便只訂閱樹狀結構中更深層的主題』 中說明的設定，以避免其他錯誤訊息。

關於這項作業

本主題告訴您如何授與樹狀結構中更深層的主題存取權，以及在沒有使用者需要時，如何移除樹狀結構中較低層次的主題存取權。

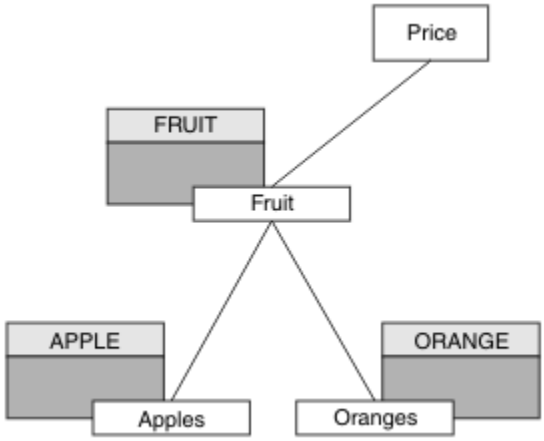


圖 *26:* 授與存取控制以避免其他訊息的範例。

定義新的主題物件，如下所示:

程序

1. 發出 MQSC 指令 `DEF TOPIC(ORANGE) TOPICSTR('Price/Fruit/Oranges')`。
2. 授與存取權，如下所示:

- z/OS **z/OS** :

  定義新的設定檔，並新增該設定檔及現有設定檔的存取權。 使用下列 RACF 指令來執行此動作:

  ```
  RDEFINE MXTOPIC hlq.SUBSCRIBE.ORANGE UACC(NONE)
  PERMIT hlq.SUBSCRIBE.ORANGE CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
  PERMIT hlq.SUBSCRIBE.APPLE CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
  ```

- 其他平台:

  使用平台的授權指令來設定同等存取權:

  ALW **AIX, Linux, and Windows 系統**

  ```
  setmqaut -t topic -n ORANGE -p USER1 +sub
  setmqaut -t topic -n APPLE -p USER1 +sub
  ```

  IBM i **IBM i**

  ```
  GRTMQAUT OBJ(ORANGE) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
  GRTMQAUT OBJ(APPLE) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
  ```

結果

在 z/OS 上，當 USER1 嘗試訂閱主題 `"Price/Fruit/Apples"` 時， `hlq.SUBSCRIBE.APPLE` 設定檔上的第一個安全檢查成功。

同樣地，當 USER2 嘗試訂閱主題 "Price/Fruit/Apples" 時，結果會成功，因為安全檢查在第一個設定檔上通過。

當 USER2 嘗試訂閱主題 "Price/Fruit/Oranges" 時，結果會失敗，並出現 MQRC_NOT_AUTHORIZED 訊息，以及:

- z/OS 在 z/OS 上，在主控台上看到下列訊息，這些訊息會透過已嘗試的主題樹狀結構顯示完整安全路徑:

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.ORANGE ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- ALW 在其他平台上，下列授權事件:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      ORANGE, FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

- IBM i 在 IBMi 上，發生下列授權事件:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      ORANGE, FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

授與使用者發佈至主題的存取權

本主題是作業清單中的第一個主題，可告訴您如何授與多個使用者對發佈主題的存取權。

關於這項作業

這項作業假設主題樹狀結構右側沒有管理主題物件，也沒有定義任何設定檔來發佈。 所使用的假設是發佈者僅使用主題字串。

應用程式可以透過提供主題物件、主題字串或兩者的組合來發佈至主題。 不論應用程式選取的方式，效果都是在主題樹狀結構中的某個點發佈。 如果主題樹狀結構中的這個點是由管理主題物件代表，則會根據該主題物件的名稱來檢查安全設定檔。 例如:

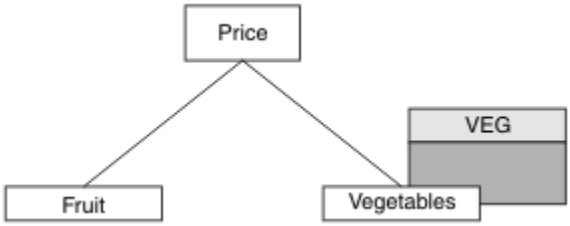


圖 *27:* 授與主題的發佈存取權

表 *89:* 發佈存取權需求範例

主題	需要發佈存取權	Topic 物件
計價	無使用者	無
價格/蔬菜	USER1	VEG

定義新的主題物件，如下所示:

程序

1. 發出 MQSC 指令 DEF TOPIC(VEG) TOPICSTR('Price/Vegetables')。
2. 授與存取權，如下所示:

   - z/OS **z/OS** :

     授與使用者對 hlq.PUBLISH.VEG 設定檔的存取權，以授與 USER1 存取權來發佈至主題 "Price/Vegetables"。 使用下列 RACF 指令來執行此動作:

     ```
     RDEFINE MXTOPIC hlq.PUBLISH.VEG UACC(NONE)
     PERMIT hlq.PUBLISH.VEG CLASS(MXTOPIC) ID(USER1) ACCESS(UPDATE)
     ```

   - 其他平台:

     授與使用者對 VEG 設定檔的存取權，以授與 USER1 存取權來發佈至主題 "Price/Vegetables"。 請使用平台的授權指令來執行此動作:

     ALW **AIX, Linux, and Windows 系統**

     ```
     setmqaut -t topic -n VEG -p USER1 +pub
     ```

     IBM i **IBM i**

     ```
     GRTMQAUT OBJ(VEG) OBJTYPE(*TOPIC) USER(USER1) AUT(*PUB)
     ```

結果

當 USER1 嘗試發佈至主題 "Price/Vegetables" 時，結果為成功; 亦即， MQOPEN 呼叫成功。

當 USER2 嘗試發佈至主題 "Price/Vegetables" 時， MQOPEN 呼叫會失敗，並出現 MQRC_NOT_AUTHORIZED 訊息:

- z/OS 在 z/OS 上，在主控台上看到下列訊息，這些訊息會透過已嘗試的主題樹狀結構顯示完整安全路徑:

  ```
  ICH408I USER(USER2   ) ...
    hlq.PUBLISH.VEG ...

  ICH408I USER(USER2   ) ...
    hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
  ```

- ALW 在其他平台上，下列授權事件:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
  TopicString           "Price/Vegetables"
  ```

- IBM i 在 IBMi 上，發生下列授權事件:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
  TopicString           "Price/Vegetables"
  ```

請注意，這是您看到的內容的圖解; 並非所有欄位。

授與使用者存取權以發佈至樹狀結構中更深層的主題

本主題是作業清單中的第二個主題，可告訴您如何授與多個使用者發佈至主題的存取權。

開始之前

本主題使用 第 416 頁的『授與使用者發佈至主題的存取權』中說明的設定。

關於這項作業

如果應用程式發佈所在主題樹狀結構中的點不是由管理主題物件所代表，請向上移動樹狀結構，直到找到最接近的上層管理主題物件為止。 會根據該主題物件的名稱來檢查安全設定檔。

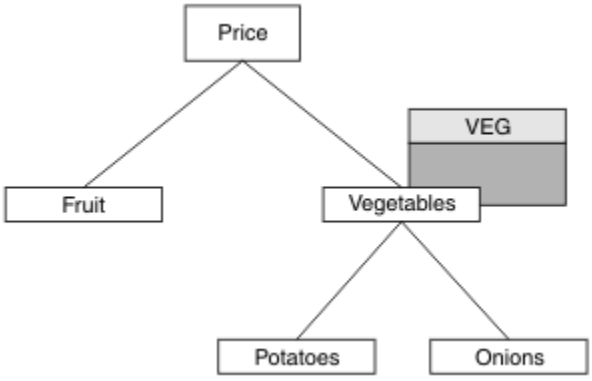


圖 *28:* 授與主題樹狀結構內主題的發佈存取權

表 *90:* 發佈存取權需求範例

主題	需要訂閱存取權	Topic 物件
計價	無使用者	無
價格/蔬菜	USER1	VEG
價格/蔬菜/馬鈴薯	USER1	
價格/蔬菜/洋蔥	USER1	

在前一項作業中， USER1 已獲授與發佈主題 "Price/Vegetables/Potatoes" 的存取權，方法是授與它對 z/OS 上 hlq.PUBLISH.VEG 設定檔的存取權，或對其他平台上 VEG 設定檔的發佈存取權。 此單一設定檔也會授與 USER1 在 "Price/Vegetables/Onions"上發佈的存取權。

當 USER1 嘗試在主題 "Price/Vegetables/Potatoes" 發佈時，結果為成功; 即 MQOPEN 呼叫成功。

當 USER2 嘗試訂閱主題 "Price/Vegetables/Potatoes" 時，結果為失敗; 亦即， MQOPEN 呼叫失敗，並出現 MQRC_NOT_AUTHORIZED 訊息，以及:

- 在 z/OS 上，在主控台上看到下列訊息，這些訊息會透過已嘗試的主題樹狀結構顯示完整安全路徑:

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.VEG ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- 在其他平台上，下列授權事件:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables/Potatoes"
```

請注意下列項目:

- 您在 z/OS 上收到的訊息與在前一個作業中收到的訊息相同，因為相同的主題物件及設定檔正在控制存取權。
- 您在其他平台上收到的事件訊息與前一個作業中收到的事件訊息類似，但實際主題字串不同。

授與發佈和訂閱的存取權

本主題是作業清單中的最後一個，可告訴您如何授與多個使用者發佈及訂閱主題的存取權。

開始之前

本主題使用 第 417 頁的『授與使用者存取權以發佈至樹狀結構中更深層的主題』中說明的設定。

關於這項作業

在前一項作業中， USER1 已獲授與訂閱主題 "Price/Fruit"的存取權。 本主題告訴您如何授與該使用者發佈至該主題的存取權。

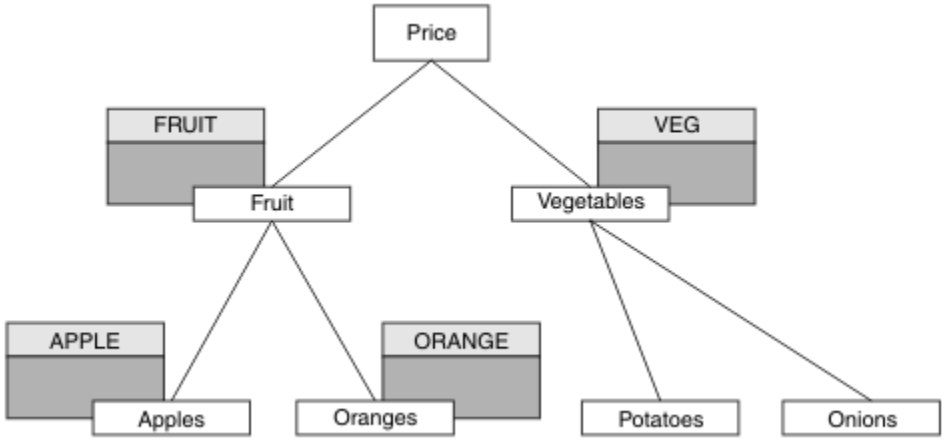


圖 *29:* 授與發佈及訂閱的存取權

表 *91:* 發佈和訂閱存取權需求範例

主題	需要訂閱存取權	需要發佈存取權	Topic 物件
計價	無使用者	無使用者	無
價格/水果	USER1	USER1	水果
價格/水果/蘋果	USER1 和 USER2		蘋果
價格/水果/場	USER1		橙色

程序

授與存取權，如下所示:

- z/OS **z/OS** :

  在較早的作業中，透過授與使用者對 hlq.SUBSCRIBE.FRUIT 設定檔的存取權， USER1 已獲授與訂閱主題 "Price/Fruit" 的存取權。

  若要發佈至 "Price/Fruit" 主題，請授與 USER1 對 hlq.PUBLISH.FRUIT 設定檔的存取權。 使用下列 RACF 指令來執行此動作:

  ```
  RDEFINE MXTOPIC hlq.PUBLISH.FRUIT UACC(NONE)
  PERMIT hlq.PUBLISH.FRUIT CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
  ```

- 其他平台:

  授與使用者對 FRUIT 設定檔的發佈存取權，以授與 USER1 存取權來發佈至主題 "Price/Fruit" 。 請使用平台的授權指令來執行此動作:

ALW AIX, Linux, and Windows 系統

```
setmqaut -t topic -n FRUIT -p USER1 +pub
```

IBM i IBM i

```
GRTMQAUT OBJ(FRUIT) OBJTYPE(*TOPIC) USER(USER1) AUTH(*PUB)
```

結果

在 z/OS 上，當 USER1 嘗試發佈至主題 "Price/Fruit" 時， MQOPEN 呼叫上的安全檢查會通過。

當 USER2 嘗試在主題 "Price/Fruit" 發佈時，結果會失敗並顯示 MQRC_NOT_AUTHORIZED 訊息，以及:

- z/OS 在 z/OS 上，在主控台上看到下列訊息，這些訊息會透過已嘗試的主題樹狀結構顯示完整安全路徑:

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- ALW 在 AIX, Linux, and Windows 平台上，發生下列授權事件:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

- IBM i 在 IBM i 上，發生下列授權事件:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

遵循這些作業的完整集，提供 USER1 及 USER2 下列存取權，以發佈及訂閱列出的主題:

表 *92:* 安全範例所產生存取權的完整清單

主題	需要訂閱存取權	需要發佈存取權	Topic 物件
計價	無使用者	無使用者	無
價格/水果	USER1	USER1	水果
價格/水果/蘋果	USER1 和 USER2		蘋果
價格/水果/場	USER1		橙色
價格/蔬菜		USER1	VEG
價格/蔬菜/馬鈴薯			
價格/蔬菜/洋蔥			

如果您在主題樹狀結構內的不同層次有不同的安全存取需求，仔細規劃可確保您不會在 z/OS 主控台日誌上收到額外的安全警告。 在樹狀結構內的正確層次設定安全，可避免誤導安全訊息。

訂閱安全

MQSO_ALTERNATE_USER_AUTHORITY

AlternateUserID 欄位包含用來驗證此 MQSUB 呼叫的使用者 ID。 只有在此 AlternateUserID 獲授權以指定的存取選項訂閱主題時，不論執行應用程式的使用者 ID 是否獲授權訂閱主題，呼叫才會成功。

MQSO_SET_IDENTITY_CONTEXT

訂閱是要使用 PubAccounting 記號和 PubApplIdentityData 欄位中提供的帳戶記號和應用程式身分資料。

如果指定此選項，則會執行相同的授權檢查，如同使用 MQOPEN 呼叫搭配 MQOO_SET_IDENTITY_CONTEXT 來存取目的地佇列一樣，但也使用 MQSO_MANAGED 選項時除外，在此情況下，目的地佇列上沒有授權檢查。

如果未指定此選項，則傳送至此訂閱者的發佈具有與其相關聯的預設環境定義資訊，如下所示:

表 *93:* 預設發佈環境定義資訊

MQMD 中的欄位	**使用的值**
UserIdentifier	發佈時與訂閱相關聯的使用者 ID (請參閱 DISPLAY SBSTATUS 上的 SUBUSER 欄位)。
AccountingToken	可能的話，從環境判定; 否則設為 MQACT_NONE。
ApplIdentityData	設為空白。

此選項僅適用於 MQSO_CREATE 及 MQSO_ALTER。 如果與 MQSO_RESUME 一起使用，則會忽略 PubAccounting 記號和 PubApplIdentityData 欄位，因此此選項沒有作用。

如果未使用先前訂閱已提供身分環境定義資訊的這個選項來變更訂閱，則會針對已變更的訂閱產生預設環境定義資訊。

如果訂閱容許不同的使用者 ID 與選項 MQSO_ANY_USERID 搭配使用，則會由不同的使用者 ID 回復，並為現在擁有訂閱的新使用者 ID 產生預設身分環境定義，且會遞送包含新身分環境定義的任何後續發佈。

AlternateSecurityId

這是隨 AlternateUserID 傳遞至授權服務以容許執行適當授權檢查的安全 ID。 只有在指定 MQSO_ALTERNATE_USER_AUTHORITY 且 AlternateUserID 欄位不是完全空白時，才會使用 AlternateSecurityID ，直到第一個空值字元或欄位結尾。

MQSO_ANY_USERID 訂閱選項

指定 MQSO_ANY_USERID 時，訂閱者的身分不受限於單一使用者 ID。 這可讓任何使用者在具有適當權限時變更或回復訂閱。 一次只能有單一使用者具有訂閱。 嘗試回復使用另一個應用程式目前正在使用的訂閱將導致呼叫失敗，並產生 MQRC_SUBSCRIPTION_IN_USE。

若要將此選項新增至現有訂閱， MQSUB 呼叫 (使用 MQSO_ALTER) 必須來自與原始訂閱相同的使用者 ID。

如果 MQSUB 呼叫參照已設定 MQSO_ANY_USERID 的現有訂閱，且使用者 ID 與原始訂閱不同，則只有在新使用者 ID 有權訂閱主題時，呼叫才會成功。 順利完成之後，此訂閱者的未來發佈會以發佈中設定的新使用者 ID 放置在訂閱者的佇列中。

MQSO_FIXED_USERID

當指定 MQSO_FIXED_USERID 時，只能由單一擁有使用者 ID 變更或回復訂閱。 此使用者 ID 是最後一個變更設定此選項之訂閱的使用者 ID ，因此移除 MQSO_ANY_USERID 選項，或者如果未發生任何變更，則它是建立訂閱的使用者 ID。

如果 MQSUB 動詞參照已設定 MQSO_ANY_USERID 的現有訂閱，並變更訂閱 (使用 MQSO_ALTER) 以使用選項 MQSO_FIXED_USERID，則訂閱的使用者 ID 現在會固定在這個新的使用者 ID。 只有在新使用者 ID 具有訂閱主題的權限時，呼叫才會成功。

如果記錄為擁有訂閱的使用者 ID 以外的使用者 ID 嘗試回復或變更 MQSO_FIXED_USERID 訂閱，則呼叫會因 MQRC_IDENTITY_MISMATCH 而失敗。 可以使用 DISPLAY SBSTATUS 指令來檢視訂閱的擁有使用者 ID。

如果未指定 MQSO_ANY_USERID 或 MQSO_FIXED_USERID，則預設值為 MQSO_FIXED_USERID。

佇列管理程式之間的發佈/訂閱安全

會使用一般通道安全規則，將發佈/訂閱內部訊息 (例如 Proxy 訂閱及發佈) 放置到發佈/訂閱系統佇列。 本主題中的資訊和圖表強調顯示遞送這些訊息所涉及的各種程序和使用者 ID。

本端存取控制

發佈和訂閱主題的存取權由 發佈/訂閱安全中說明的本端安全定義和規則控管。 在 z/OS 上，不需要任何本端主題物件即可建立存取控制。 其他平台上的存取控制也不需要本端主題。 管理者可以選擇將存取控制套用至叢集主題物件，而不論它們是否存在於叢集中。

系統管理者負責其本端系統上的存取控制。 他們必須信任階層或叢集群體的其他成員的管理者，以負責其存取控制原則。 因為存取控制是針對每一個個別機器所定義，如果需要精細層次控制，則很可能是負擔。 可能不需要強制任何存取控制，或可能在主題樹狀結構中的高階物件上定義存取控制。 可以針對主題名稱空間的每一個子細目定義精細層次存取控制。

建立 Proxy 訂閱

一般通道鑑別方法會確認組織將其佇列管理程式連接至佇列管理程式的信任。 如果也容許該授信組織執行分散式發佈/訂閱，則會執行權限檢查。 當通道將訊息放入分散式發佈/訂閱佇列時，即會進行檢查。 例如，如果將訊息放入 SYSTEM.INTER.QMGR.CONTROL 佇列。 佇列權限檢查的使用者 ID 取決於接收端通道的 PUTAUT 值。 例如，通道 MCAUSER 的使用者 ID (訊息環境定義)，視值和平台而定。 如需通道安全的相關資訊，請參閱 通道安全。

使用遠端佇列管理程式上分散式發佈/訂閱代理程式的使用者 ID 進行 Proxy 訂閱。 例如， 第 422 頁的圖 30 中的 QM2 。 然後會輕鬆授與使用者對本端主題物件設定檔的存取權，因為該使用者 ID 已定義在系統中，因此沒有網域衝突。

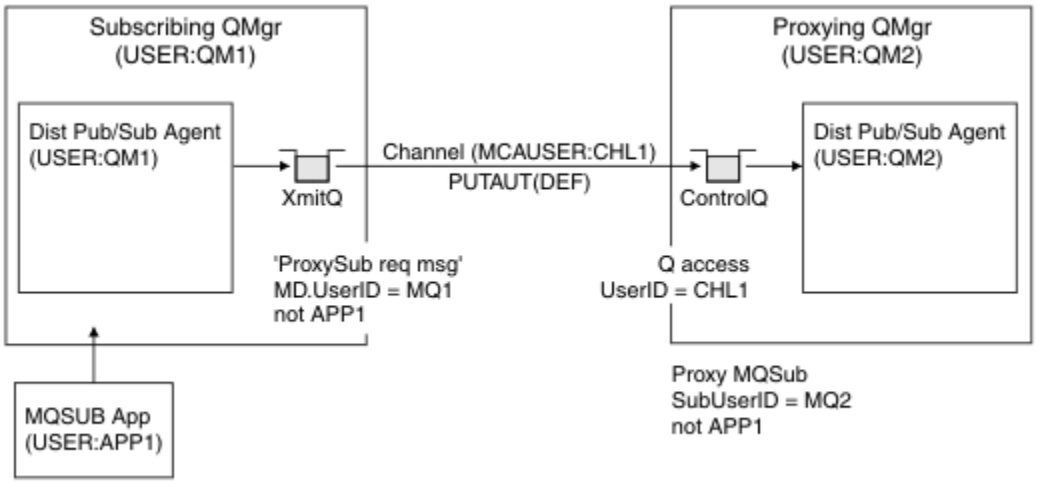


圖 *30: Proxy* 訂閱安全，建立訂閱

傳回遠端發佈

在發佈佇列管理程式上建立發佈時，會為任何 Proxy 訂閱建立發佈副本。 所複製發佈的環境定義包含進行訂閱之使用者 ID 的環境定義; 第 423 頁的圖 31 中的 QM2 。 建立 Proxy 訂閱時使用的目的地佇列是遠端佇列，因此發佈訊息會解析至傳輸佇列。

一般通道鑑別方法會確認信任組織將其佇列管理程式 QM2 連接至另一個佇列管理程式 QM1。 如果接著容許該授信組織執行分散式發佈/訂閱，當通道將發佈訊息放入分散式發佈/訂閱發佈佇列 SYSTEM.INTER.QMGR.PUBS 時，會執行權限檢查。 佇列權限檢查的使用者 ID 取決於接收端通道的 PUTAUT 值 (例如，通道的使用者 ID、 MCAUSER、訊息環境定義及其他，視值及平台而定)。 如需通道安全的相關資訊，請參閱 通道安全。

當發佈訊息到達訂閱佇列管理程式時，會在該佇列管理程式的權限下對主題執行另一個 MQPUT ，且含有訊息的環境定義會取代為每一個本端訂閱者的環境定義，因為每一個本端訂閱者都有訊息。

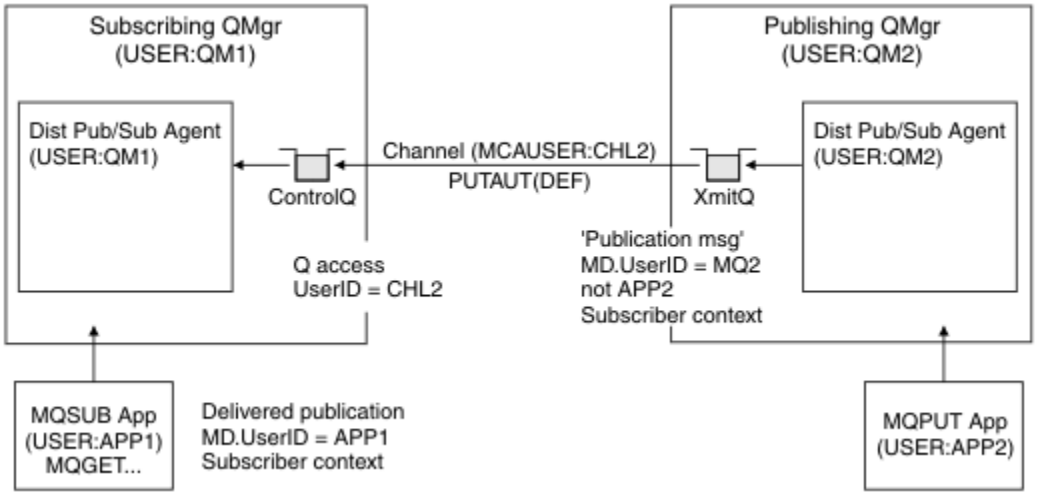


圖 *31: Proxy* 訂閱安全，轉遞發佈

在安全考量不多的系統上，分散式發佈/訂閱處理程序可能以 mqm 群組中的使用者 ID 執行，通道上的 MCAUSER 參數為空白 (預設值) ，並視需要將訊息遞送至各種系統佇列。 未受保護的系統可讓您輕鬆設定概念證明，以示範分散式發佈/訂閱。

在更認真考量安全的系統上，這些內部訊息會受到與任何通過通道的訊息相同的安全控制。

如果通道設定為非空白 MCAUSER 及指定必須勾選 MCAUSER 的 PUTAUT 值，則必須將 SYSTEM.INTER.QMGR.* 佇列的存取權授與有問題的 MCAUSER 。 如果有多個不同的遠端佇列管理程式，且通道在不同的 MCAUSER ID 下執行，則需要授與所有這些使用者 ID 對 SYSTEM.INTER.QMGR.* 佇列的存取權。 例如，在單一佇列管理程式上配置多個階層式連線時，可能會出現以不同 MCAUSER ID 執行的通道。

如果通道設定了 PUTAUT 值，指定使用訊息的環境定義，則會根據內部訊息內的使用者 ID 來檢查對 SYSTEM.INTER.QMGR.* 佇列的存取權。 因為所有這些訊息都與來自傳送內部訊息或發佈訊息之佇列管理程式的分散式發佈/訂閱代理程式使用者 ID 一起放置 (請參閱 第 423 頁的圖 31) ，所以如果您想要以此方式設定分散式發佈/訂閱安全，則授與存取各種系統佇列的使用者 ID 集不會太大 (每個遠端佇列管理程式一個)。 它仍然有通道環境定義安全一律會有的所有相同問題; 不同使用者 ID 網域的問題，以及訊息中的使用者 ID 可能未定義在接收系統上的事實。 不過，如果需要的話，這是完全可以接受的執行方式。

z/OS 系統佇列安全 提供安全設定分散式發佈/訂閱環境所需的佇列及存取權清單。 如果由於安全違規而無法放置任何內部訊息或發佈資訊，則通道會以正常方式將訊息寫入日誌，並可根據正常通道錯誤處理將訊息傳送至無法傳送郵件的佇列。

為了分散式發佈/訂閱的目的，使用一般通道安全來執行所有佇列間管理程式傳訊。

如需在主題層次限制發佈和 Proxy 訂閱的相關資訊，請參閱 發佈/訂閱安全。

將預設使用者 ID 與佇列管理程式階層搭配使用

如果您具有在不同平台上執行且使用預設使用者 ID 的佇列管理程式階層，請注意，這些預設使用者 ID 在平台之間不同，且在目標平台上可能不明。 因此，在一個平台上執行的佇列管理程式會拒絕從其他平台上的佇列管理程式收到的訊息，原因碼為 MQRC_NOT_AUTHORIZED。

為了避免拒絕訊息，至少需要將下列權限新增至其他平台上使用的預設使用者 ID:

- SYSTEM.BROKER。 佇列
- * SYSTEM.BROKER 上的 PUB *SUB 權限。 主題
- SYSTEM.BROKER.CONTROL.QUEUE 佇列。

具有佇列管理程式階層的預設使用者 ID 如下:

平台	預設使用者 ID
Windows	mqm
AIX and Linux 系統	mqm
IBM i	QMQM
z/OS	通道起始程式位址空間使用者 ID

針對 z/OS、 AIX, Linux, and Windows 平台上的佇列管理程式，如果以階層式方式連接至 IBM i 上的佇列管理程式，請建立並授與存取權給 'qmqm' 使用者 ID。

對於 IBM i 及 z/OS 平台上的佇列管理程式，如果以階層方式連接至 AIX, Linux, and Windows 上的佇列管理程式，則會建立並授與存取權給 'mqm' 使用者 ID。

針對「多平台」上的「佇列管理程式」，建立並授與使用者對 z/OS 通道起始程式位址空間使用者 ID 的存取權 (如果以階層方式連接至 z/OS 上的佇列管理程式)。

使用者 ID 可以區分大小寫。 原始佇列管理程式 (如果在 多平台上) 會強制使用者 ID 全部大寫。 接收端佇列管理程式 (如果在 AIX, Linux, and Windows 上) 會強制使用者 ID 全部為小寫。 因此，在 AIX and Linux 系統上建立的所有使用者 ID 都必須以小寫形式建立。 如果已安裝訊息結束程式，則不會強制使用者 ID 使用大寫或小寫。 請小心瞭解訊息結束程式如何處理使用者 ID。

若要避免使用者 ID 轉換的潛在問題，請執行下列動作:

- 在 AIX, Linux, and Windows 系統上，請確保以小寫形式指定使用者 ID。
- 在 IBM i 和 z/OS 上，請確定以大寫指定使用者 ID。

IBM MQ Console 和 REST API 安全

透過編輯 mqwebuser.xml 檔案中的 mqweb 伺服器配置來配置 IBM MQ Console 和 REST API 的安全。

關於這項作業

您可以透過檢查 mqweb 伺服器的日誌檔，來追蹤使用者動作並審核 IBM MQ Console 及 REST API 的使用。

IBM MQ Console 及 REST API 的使用者可以使用下列方式進行鑑別:

- 基本登錄
- LDAP 登錄
- 本端 OS 登錄
- z/OS 上的 SAF
- WebSphere Liberty 支援的任何其他登錄類型

角色可以指派給 IBM MQ Console 使用者，以及指派給 REST API 使用者，以決定他們獲授與 IBM MQ 物件的存取層次。 例如，若要執行傳訊，使用者必須獲指派 MQWebUser 角色。 如需可用角色的相關資訊，請參閱 第 434 頁的『IBM MQ Console 和 REST API 上的角色』。

指派角色給使用者之後，有許多方法可用來鑑別使用者。 使用 IBM MQ Console，使用者可以使用使用者名稱及密碼登入，也可以使用用戶端憑證鑑別。 使用 REST API，使用者可以使用基本 HTTP 鑑別、記號型鑑別或用戶端憑證鑑別。

程序

1. 定義使用者登錄以鑑別使用者，並指派角色給每一個使用者或群組，以授權使用者和群組使用 IBM MQ Console 或 REST API。 如需相關資訊，請參閱： 第 425 頁的『配置使用者和角色』
2. 選擇 IBM MQ Console 的使用者如何向 mqweb 伺服器進行鑑別。 您不必對所有使用者使用相同的方法:
   - 讓使用者使用記號鑑別進行鑑別。 在此情況下，使用者會在 IBM MQ Console 登入畫面中輸入使用者 ID 和密碼。 會產生 LTPA 記號，可讓使用者保持登入狀態並獲得設定時間量的授權。 不需要進一步配置即可使用此鑑別選項，但您可以選擇性地配置 LTPA 記號的到期時間。 如需相關資訊，請參閱 配置 LTPA 記號期限間隔。
   - 讓使用者使用用戶端憑證進行鑑別。 在此情況下，使用者不會使用使用者 ID 或密碼來登入 IBM MQ Console，而是改用用戶端憑證。 如需相關資訊，請參閱 第 438 頁的『搭配使用用戶端憑證鑑別與 REST API 及 IBM MQ Console』。
3. 選擇 REST API 的使用者如何向 mqweb 伺服器進行鑑別。 您不必對所有使用者使用相同的方法:
   - 讓使用者使用 HTTP 基本鑑別進行鑑別。 在此情況下，使用者名稱及密碼會編碼，但不會加密，並隨每一個 REST API 要求一起傳送，以鑑別及授權該要求的使用者。 為了使此鑑別安全，您必須使用安全連線。 也就是說，您必須使用 HTTP。 如需相關資訊，請參閱 第 441 頁的『搭配使用 HTTP 基本鑑別與 REST API』。
   - 讓使用者使用記號鑑別進行鑑別。 在此情況下，使用者會使用 HTTP POST 方法，將使用者 ID 及密碼提供給 REST API `login` 資源。 會產生 LTPA 記號，可讓使用者保持登入狀態並獲得設定時間量的授權。 如需相關資訊，請參閱第 442 頁的『搭配 REST API 使用記號型鑑別』。

     為了使此鑑別安全，您必須使用安全連線。 也就是說，您必須使用 HTTP。 不過，如果您已啟用 HTTP 連線，則可以容許將針對 HTTP 連線發出的 LTPA 記號用於 HTTP 連線。 如需相關資訊，請參閱 配置 LTPA 記號。
   - 讓使用者使用用戶端憑證進行鑑別。 在此情況下，使用者不會使用使用者 ID 或密碼來登入 REST API，而是改用用戶端憑證。 如需相關資訊，請參閱 第 438 頁的『搭配使用用戶端憑證鑑別與 REST API 及 IBM MQ Console』。
4. 選擇性的: 配置 REST API 的「跨原點資源共用」。

   依預設，當 Script 不是來自與 REST API 相同的原點時， Web 瀏覽器不容許 Script (例如 JavaScript) 呼叫 REST API 。 也就是說，未啟用跨原點要求。 您可以配置「跨原點資源共用 (CORS)」，以容許來自指定 URL 的跨原點要求。 如需相關資訊，請參閱第 445 頁的『配置 REST API 的 CORS』。
5. 選擇性的: 配置 IBM MQ Console 和 REST API 的主機標頭驗證。

   您可以配置主機標頭驗證，並建立主機名稱和埠的允許清單，以確保 IBM MQ Console 和 REST API 只會處理包含特定主機標頭的要求。 如需相關資訊，請參閱第 445 頁的『配置 IBM MQ Console 和 REST API 的主機標頭驗證』。

配置使用者和角色

若要使用 IBM MQ Console 或 REST API，使用者需要針對定義給 mqweb 伺服器的使用者登錄進行鑑別。

關於這項作業

已鑑別使用者必須是其中一個群組的成員，以授權存取 IBM MQ Console 及 REST API 的功能。 依預設，使用者登錄不包含任何使用者; 需要透過編輯 `mqwebuser.xml` 檔案來新增這些使用者。

當您配置使用者和群組時，請先配置使用者登錄來鑑別使用者和群組。 此使用者登錄在 IBM MQ Console 與 REST API 之間共用。 當您為使用者和群組配置角色時，您可以控制使用者和群組是否具有 IBM MQ Console、 REST API 或兩者的存取權。

在配置使用者登錄之後，您可以配置使用者和群組的角色，以授與他們授權。 有數個角色可用，包括使用 REST API for Managed File Transfer 的特定角色。 每一個角色會授與不同層次的存取權。 如需相關資訊，請參閱第 434 頁的『IBM MQ Console 和 REST API 上的角色』。

mqweb 伺服器提供了一些範例 XML 檔案，以簡化使用者和群組的配置。 熟悉在 WebSphere Liberty (WLP) 中配置安全的使用者可能偏好不使用範例。 除了這裡所記載的授權功能之外， WLP 還提供其他授權功能。

程序

- 使用 `basic_registry.xml` 檔案，以基本登錄來配置使用者和群組。

  登錄中的使用者名稱和密碼用來鑑別及授權 IBM MQ Console 和 REST API 的使用者。

  如果要使用 `basic_registry.xml` 範例檔來配置基本登錄，請參閱第 427 頁的『配置 IBM MQ Console 和 REST API 的基本登錄』。
- 使用 `ldap_registry.xml` 檔案，以 LDAP 登錄來配置使用者和群組。

  LDAP 登錄中的使用者名稱和密碼用來鑑別及授權使用 IBM MQ Console 和 REST API。

  如果要使用 `ldap_registry.xml` 範例檔來配置 LDAP 登錄，請參閱第 430 頁的『配置 IBM MQ Console 和 REST API 的 LDAP 登錄』。
- ALW

  使用 `local_os_registry.xml` 檔案，以本端作業系統登錄來配置使用者和群組。

  作業系統登錄中的使用者名稱和密碼用來鑑別及授權 IBM MQ Console 和 REST API 的使用者。

  若要使用 `local_os_registry.xml` 範例檔來配置本端 OS 登錄，請參閱第 429 頁的『配置 IBM MQ Console 和 REST API 的本端 OS 登錄』。
- z/OS

  使用 `zos_saf_registry.xml` 檔，在 z/OS 上使用「系統授權機能 (SAF)」介面來配置使用者和群組。

  RACF 或其他安全產品會使用設定檔來授與使用者和群組對角色的存取權。 RACF 資料庫中的使用者名稱和密碼用來鑑別及授權 IBM MQ Console 和 REST API 的使用者。

  如果要使用 `zos_saf_registry.xml` 範例檔來配置 SAF 介面，請參閱第 431 頁的『配置 IBM MQ Console 和 REST API 的 SAF 登錄』。
- 停用安全，包括使用 `no_security.xml` 檔案來存取 IBM MQ Console 或 REST API 的能力。

下一步

選擇使用者鑑別的方式:

IBM MQ Console 鑑別選項

- 讓使用者使用記號鑑別進行鑑別。 在此情況下，使用者會在 IBM MQ Console 登入畫面中輸入使用者 ID 和密碼。 會產生 LTPA 記號，可讓使用者保持登入狀態並獲得設定時間量的授權。 不需要進一步配置即可使用此鑑別選項，但您可以選擇性地配置 LTPA 記號的期限間隔。 如需相關資訊，請參閱 配置 LTPA 記號期限間隔。
- 讓使用者使用用戶端憑證進行鑑別。 在此情況下，使用者不會使用使用者 ID 或密碼來登入 IBM MQ Console，而是改用用戶端憑證。 如需相關資訊，請參閱第 438 頁的『搭配使用用戶端憑證鑑別與 REST API 及 IBM MQ Console』。

REST API 鑑別選項

- 讓使用者使用 HTTP 基本鑑別進行鑑別。 在此情況下，使用者名稱及密碼會編碼，但不會加密，並隨每一個 REST API 要求一起傳送，以鑑別及授權該要求的使用者。 為了使此鑑別安全，您必須使用安全連線。 也就是說，您必須使用 HTTP。 如需相關資訊，請參閱第 441 頁的『搭配使用 HTTP 基本鑑別與 REST API』。
- 讓使用者使用記號鑑別進行鑑別。 在此情況下，使用者會使用 HTTP POST 方法，將使用者 ID 及密碼提供給 REST API `login` 資源。 會產生 LTPA 記號，可讓使用者保持登入狀態並獲得設定時間量的授

權。 如需相關資訊，請參閱第 442 頁的『搭配 REST API 使用記號型鑑別』。 您可以配置 LTPA 記號的期限間隔。 如需相關資訊，請參閱 配置 LTPA 記號。

- 讓使用者使用用戶端憑證進行鑑別。 在此情況下，使用者不會使用使用者 ID 或密碼來登入 REST API，而是改用用戶端憑證。 如需相關資訊，請參閱 第 438 頁的『搭配使用用戶端憑證鑑別與 REST API 及 IBM MQ Console』。

配置 IBM MQ Console 和 REST API 的基本登錄

您可以在 mqwebuser.xml 檔內配置基本登錄。 xml 檔中的使用者名稱、密碼和角色用來鑑別及授權 IBM MQ Console 和 REST API 的使用者。

開始之前

- 當您在基本登錄內配置使用者時，必須為每一個使用者指派一個角色。 每一個角色提供不同層次的專用權來存取 IBM MQ Console 和 REST API，並決定在嘗試容許的作業時所使用的安全環境定義。 在配置基本登錄之前，您需要瞭解這些角色。 如需每一個角色的相關資訊，請參閱 第 434 頁的『IBM MQ Console 和 REST API 上的角色』。
- 若要完成此作業，您必須是具有足夠專用權來編輯 mqwebuser.xml 檔案的使用者:
  - z/OS 在 z/OS 上，您必須具有 mqwebuser.xml 檔案的寫入權。
  - Multi 在所有其他作業系統上，必須是 特許使用者。

程序

1. 從下列其中一個路徑複製範例 XML 檔 basic_registry.xml :
   - ALW 在 AIX, Linux, and Windows 上: *MQ_INSTALLATION_PATH* /web/mq/samp/configuration
   - z/OS 在 z/OS 上: PathPrefix /web/mq/samp/configuration

     其中 PathPrefix 是 IBM MQ for z/OS UNIX System Services Components 安裝路徑。
2. 將範例檔放在適當的目錄中:
   - ALW

     在 AIX, Linux, and Windows 上: *MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb
   - z/OS

     在 z/OS 上: *WLP_user_directory*/servers/mqweb

     其中 *WLP_user_directory* 是在執行 **crtmqweb** Script 以建立 mqweb 伺服器定義時指定的目錄。
3. 選擇性的: 如果您在 mqwebuser.xml 中變更任何配置設定，請將它們複製到範例檔。
4. 刪除現有的 mqwebuser.xml 檔案，並將範例檔案重新命名為 mqwebuser.xml。
5. 編輯新的 mqwebuser.xml 檔案，以在 **basicRegistry** 標籤內新增使用者和群組。

   請注意，任何具有 MQWebUser 角色的使用者都只能執行授與使用者 ID 在佇列管理程式上執行的作業。 因此，登錄中定義的使用者 ID 在安裝 IBM MQ 的系統上必須具有相同的使用者 ID。 這些使用者 ID 必須在相同案例中，否則使用者 ID 之間的對映可能會失敗。

   如需配置基本使用者登錄的相關資訊，請參閱 WebSphere Liberty 說明文件中的 配置 Liberty 的基本使用者登錄 。
6. 透過編輯 mqwebuser.xml 檔案，將角色指派給使用者和群組:

   有數個角色可授權使用者和群組使用 IBM MQ Console 及 REST API。 每一個角色會授與不同層次的存取權。 如需相關資訊，請參閱第 434 頁的『IBM MQ Console 和 REST API 上的角色』。

- 若要指派角色並授與對 IBM MQ Console 的存取權，請在 **<enterpriseApplication id="com.ibm.mq.console">** 標籤內適當的 **security-role** 標籤之間新增使用者和群組。
- 若要指派角色並授與對 REST API 的存取權，請在 **<enterpriseApplication id="com.ibm.mq.rest">** 標籤內適當的 **security-role** 標籤之間新增使用者和群組。

如需 **security-role** 標籤內使用者和群組資訊格式的說明，請參閱 範例。

7. 如果您在 mqwebuser.xml 中為使用者提供密碼，則應該使用 WebSphere Liberty 提供的 **securityUtility encoding** 指令來編碼這些密碼，使它們更安全。 如需相關資訊，請參閱 WebSphere Liberty 產品說明文件中的 Liberty:securityUtility 指令 。

範例

在下列範例中，群組 MQWebAdminGroup 獲授與角色為 MQWebAdmin 之 IBM MQ Console 的存取權。 使用者 reader 獲授與角色 MQWebAdminRO 的存取權，而使用者 guest 獲授與角色 MQWebUser 的存取權:

```
<enterpriseApplication id="com.ibm.mq.console">
        <application-bnd>
            <security-role name="MQWebAdmin">
                <group name="MQWebAdminGroup" realm="defaultRealm"/>
            </security-role>
            <security-role name="MQWebAdminRO">
                <user name="reader" realm="defaultRealm"/>
            </security-role>
            <security-role name="MQWebUser">
                <user name="guest" realm="defaultRealm"/>
            </security-role>
        </application-bnd>
    </enterpriseApplication>
```

在下列範例中，使用者 reader 和 guest 獲授與 IBM MQ Console 的存取權。 使用者 user 獲授與 REST API 的存取權，而 MQAdmin 群組內的任何使用者則獲授與 IBM MQ Console 及 REST API 的存取權。 mftadmin 使用者已獲授與 REST API for MFT 的存取權:

```
<enterpriseApplication id="com.ibm.mq.console">
    <application-bnd>
        <security-role name="MQWebAdmin">
            <group name="MQAdmin" realm="defaultRealm"/>
        </security-role>
        <security-role name="MQWebAdminRO">
            <user name="reader" realm="defaultRealm"/>
        </security-role>
        <security-role name="MQWebUser">
            <user name="guest" realm="defaultRealm"/>
        </security-role>
    </application-bnd>
</enterpriseApplication>

<enterpriseApplication id="com.ibm.mq.rest">
    <application-bnd>
        <security-role name="MQWebAdmin">
            <group name="MQAdmin" realm="defaultRealm"/>
        </security-role>
        <security-role name="MQWebUser">
            <user name="user" realm="defaultRealm"/>
        </security-role>
        <security-role name="MFTWebAdmin">
            <user name="mftadmin" realm="defaultRealm"/>
        </security-role>
     </application-bnd>
</enterpriseApplication>
```

下一步

選擇使用者鑑別的方式:

IBM MQ Console 鑑別選項

- 讓使用者使用記號鑑別進行鑑別。 在此情況下，使用者會在 IBM MQ Console 登入畫面中輸入使用者 ID 和密碼。 會產生 LTPA 記號，可讓使用者保持登入狀態並獲得設定時間量的授權。 不需要進一步配

置即可使用此鑑別選項，但您可以選擇性地配置 LTPA 記號的期限間隔。 如需相關資訊，請參閱 配置 LTPA 記號期限間隔。

- 讓使用者使用用戶端憑證進行鑑別。 在此情況下，使用者不會使用使用者 ID 或密碼來登入 IBM MQ Console，而是改用用戶端憑證。 如需相關資訊，請參閱 第 438 頁的『搭配使用用戶端憑證鑑別與 REST API 及 IBM MQ Console』。

REST API 鑑別選項

- 讓使用者使用 HTTP 基本鑑別進行鑑別。 在此情況下，使用者名稱及密碼會編碼，但不會加密，並隨每一個 REST API 要求一起傳送，以鑑別及授權該要求的使用者。 為了使此鑑別安全，您必須使用安全連線。 也就是說，您必須使用 HTTP。 如需相關資訊，請參閱 第 441 頁的『搭配使用 HTTP 基本鑑別與 REST API』。
- 讓使用者使用記號鑑別進行鑑別。 在此情況下，使用者會使用 HTTP POST 方法，將使用者 ID 及密碼提供給 REST API `login` 資源。 會產生 LTPA 記號，可讓使用者保持登入狀態並獲得設定時間量的授權。 如需相關資訊，請參閱第 442 頁的『搭配 REST API 使用記號型鑑別』。 您可以配置 LTPA 記號的期限間隔。 如需相關資訊，請參閱 配置 LTPA 記號。
- 讓使用者使用用戶端憑證進行鑑別。 在此情況下，使用者不會使用使用者 ID 或密碼來登入 REST API，而是改用用戶端憑證。 如需相關資訊，請參閱 第 438 頁的『搭配使用用戶端憑證鑑別與 REST API 及 IBM MQ Console』。

ALW 配置 IBM MQ Console 和 REST API 的本端 OS 登錄

您可以在 `mqwebuser.xml` 檔內配置本端作業系統登錄。 本端作業系統上的使用者名稱和密碼用來鑑別及授權 IBM MQ Console 和 REST API 的使用者。

開始之前

- 對於具有本端 OS 鑑別特性的用戶端憑證鑑別，使用者身分是來自用戶端憑證識別名稱 (DN) 的通用名稱 (CN)。 如果使用者身分作為作業系統使用者不存在，則用戶端憑證登入將失敗並撤回至密碼型鑑別。
- 若要完成此作業，您必須是 特許使用者。

關於這項作業

使用本端作業系統登錄，使用者和群組會自動獲指派角色:

- 任何屬於 'mqm' 群組或 IBM i 上 'QMQMADM' 群組的使用者都會被授與 MQWebAdmin 及 MFTWebAdmin 角色。
- 所有其他使用者都會獲授與 MQWebUser 角色。

如需這些角色的相關資訊，請參閱 第 434 頁的『IBM MQ Console 和 REST API 上的角色』。

本端作業系統登錄只能在 AIX, Linux, and Windows 上使用。 透過配置 SAF 登錄，在 z/OS 上提供對等功能。 如需相關資訊，請參閱第 431 頁的『配置 IBM MQ Console 和 REST API 的 SAF 登錄』。

程序

1. 從下列路徑複製範例 XML 檔 `local_os_registry.xml` :
   MQ_INSTALLATION_PATH`/web/mq/samp/configuration`
2. 將範例檔放在下列目錄中:
   MQ_DATA_PATH`/web/installations/`*installationName*`/servers/mqweb`
3. 選擇性的: 如果您在 `mqwebuser.xml` 中變更任何配置設定，請將它們複製到範例檔。
4. 刪除現有的 `mqwebuser.xml` 檔案，並將範例檔案重新命名為 `mqwebuser.xml`。

下一步

選擇使用者鑑別的方式:

IBM MQ Console 鑑別選項

- 讓使用者使用記號鑑別進行鑑別。 在此情況下，使用者會在 IBM MQ Console 登入畫面中輸入使用者 ID 和密碼。 會產生 LTPA 記號，可讓使用者保持登入狀態並獲得設定時間量的授權。 不需要進一步配置即可使用此鑑別選項，但您可以選擇性地配置 LTPA 記號的期限間隔。 如需相關資訊，請參閱 配置 LTPA 記號期限間隔。
- 讓使用者使用用戶端憑證進行鑑別。 在此情況下，使用者不會使用使用者 ID 或密碼來登入 IBM MQ Console，而是改用用戶端憑證。 如需相關資訊，請參閱 第 438 頁的『搭配使用用戶端憑證鑑別與 REST API 及 IBM MQ Console』。

REST API 鑑別選項

- 讓使用者使用 HTTP 基本鑑別進行鑑別。 在此情況下，使用者名稱及密碼會編碼，但不會加密，並隨每一個 REST API 要求一起傳送，以鑑別及授權該要求的使用者。 為了使此鑑別安全，您必須使用安全連線。 也就是說，您必須使用 HTTP。 如需相關資訊，請參閱 第 441 頁的『搭配使用 HTTP 基本鑑別與 REST API』。
- 讓使用者使用記號鑑別進行鑑別。 在此情況下，使用者會使用 HTTP POST 方法，將使用者 ID 及密碼提供給 REST API `login` 資源。 會產生 LTPA 記號，可讓使用者保持登入狀態並獲得設定時間量的授權。 如需相關資訊，請參閱第 442 頁的『搭配 REST API 使用記號型鑑別』。 您可以配置 LTPA 記號的期限間隔。 如需相關資訊，請參閱 配置 LTPA 記號。
- 讓使用者使用用戶端憑證進行鑑別。 在此情況下，使用者不會使用使用者 ID 或密碼來登入 REST API，而是改用用戶端憑證。 如需相關資訊，請參閱 第 438 頁的『搭配使用用戶端憑證鑑別與 REST API 及 IBM MQ Console』。

配置 IBM MQ Console 和 REST API 的 LDAP 登錄

您可以在 `mqwebuser.xml` 檔內配置 LDAP 登錄。 LDAP 登錄中的使用者名稱和密碼用來鑑別及授權 IBM MQ Console 和 REST API 的使用者。

開始之前

- 當您配置 LDAP 登錄時，必須為每一個使用者指派一個角色。 每一個角色提供不同層次的專用權來存取 IBM MQ Console 和 REST API，並決定在嘗試容許的作業時所使用的安全環境定義。 在配置登錄之前，您需要先瞭解這些角色。 如需每一個角色的相關資訊，請參閱 第 434 頁的『IBM MQ Console 和 REST API 上的角色』。

  請注意，任何具有 `MQWebUser` 角色的使用者都只能執行授與使用者 ID 在佇列管理程式上執行的作業。 因此， LDAP 伺服器上定義的使用者 ID 在安裝 IBM MQ 的系統上必須具有相同的使用者 ID。 這些使用者 ID 必須在相同案例中，否則使用者 ID 之間的對映可能會失敗。
- 若要完成此作業，您必須是具有足夠專用權來編輯 `mqwebuser.xml` 檔案的使用者:
  - z/OS 在 z/OS 上，您必須具有 `mqwebuser.xml` 檔案的寫入權。
  - Multi 在所有其他作業系統上，必須是 特許使用者。

程序

1. 從下列其中一個路徑複製範例 XML 檔 `ldap_registry.xml` :
   - ALW 在 AIX, Linux, and Windows 上: *MQ_INSTALLATION_PATH* `/web/mq/samp/configuration`
   - z/OS 在 z/OS 上: `PathPrefix /web/mq/samp/configuration`

     其中 `PathPrefix` 是 IBM MQ for z/OS UNIX System Services Components 安裝路徑。
2. 將範例檔放在適當的目錄中:
   - ALW

在 AIX, Linux, and Windows 上: *MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb

- z/OS

  在 z/OS 上: *WLP_user_directory*/servers/mqweb

  其中 *WLP_user_directory* 是在執行 **crtmqweb** Script 以建立 mqweb 伺服器定義時指定的目錄。

3. 選擇性的: 如果您在 mqwebuser.xml 中變更任何配置設定，請將它們複製到範例檔。
4. 刪除現有的 mqwebuser.xml 檔案，並將範例檔案重新命名為 mqwebuser.xml。
5. 編輯新的 mqwebuser.xml 檔，以變更 **ldapRegistry** 和 **idsLdapFilterProperties** 標籤內的 LDAP 登錄設定。

   如需配置 LDAP 登錄的相關資訊，請參閱 WebSphere Liberty 說明文件中的 在 Liberty 中配置 LDAP 使用者登錄 。
6. 透過編輯 mqwebuser.xml 檔案，將角色指派給使用者和群組:

   有數個角色可授權使用者和群組使用 IBM MQ Console 及 REST API。 每一個角色會授與不同層次的存取權。 如需相關資訊，請參閱第 434 頁的『IBM MQ Console 和 REST API 上的角色』。

   - 若要指派角色並授與對 IBM MQ Console 的存取權，請在 **<enterpriseApplication id="com.ibm.mq.console">** 標籤內適當的 **security-role** 標籤之間新增使用者和群組。
   - 若要指派角色並授與對 REST API 的存取權，請在 **<enterpriseApplication id="com.ibm.mq.rest">** 標籤內適當的 **security-role** 標籤之間新增使用者和群組。

下一步

選擇使用者鑑別的方式:

IBM MQ Console 鑑別選項

- 讓使用者使用記號鑑別進行鑑別。 在此情況下，使用者會在 IBM MQ Console 登入畫面中輸入使用者 ID 和密碼。 會產生 LTPA 記號，可讓使用者保持登入狀態並獲得設定時間量的授權。 不需要進一步配置即可使用此鑑別選項，但您可以選擇性地配置 LTPA 記號的期限間隔。 如需相關資訊，請參閱 配置 LTPA 記號期限間隔。
- 讓使用者使用用戶端憑證進行鑑別。 在此情況下，使用者不會使用使用者 ID 或密碼來登入 IBM MQ Console，而是改用用戶端憑證。 如需相關資訊，請參閱 第 438 頁的『搭配使用用戶端憑證鑑別與 REST API 及 IBM MQ Console』。

REST API 鑑別選項

- 讓使用者使用 HTTP 基本鑑別進行鑑別。 在此情況下，使用者名稱及密碼會編碼，但不會加密，並隨每一個 REST API 要求一起傳送，以鑑別及授權該要求的使用者。 為了使此鑑別安全，您必須使用安全連線。 也就是說，您必須使用 HTTP。 如需相關資訊，請參閱 第 441 頁的『搭配使用 HTTP 基本鑑別與 REST API』。
- 讓使用者使用記號鑑別進行鑑別。 在此情況下，使用者會使用 HTTP POST 方法，將使用者 ID 及密碼提供給 REST API login 資源。 會產生 LTPA 記號，可讓使用者保持登入狀態並獲得設定時間量的授權。 如需相關資訊，請參閱第 442 頁的『搭配 REST API 使用記號型鑑別』。 您可以配置 LTPA 記號的期限間隔。 如需相關資訊，請參閱 配置 LTPA 記號。
- 讓使用者使用用戶端憑證進行鑑別。 在此情況下，使用者不會使用使用者 ID 或密碼來登入 REST API，而是改用用戶端憑證。 如需相關資訊，請參閱 第 438 頁的『搭配使用用戶端憑證鑑別與 REST API 及 IBM MQ Console』。

z/OS 配置 IBM MQ Console 和 REST API 的 SAF 登錄

「系統授權機能 (SAF)」介面可讓 mqweb 伺服器呼叫外部安全管理程式來進行鑑別及授權檢查。 然後，使用者可以使用 z/OS 使用者 ID 及密碼登入 IBM MQ Console 及 REST API 。

開始之前

- 當您配置 SAF 登錄時，必須指派角色給使用者。 每一個角色提供不同層次的專用權來存取 IBM MQ Console 和 REST API，並決定在嘗試容許的作業時所使用的安全環境定義。 在配置登錄之前，您需要先瞭解這些角色。 如需每一個角色的相關資訊，請參閱 第 434 頁的『IBM MQ Console 和 REST API 上的角色』。
- 您需要執行 WebSphere Liberty Angel Process，才能使用 SAF 的授權介面。 如需相關資訊，請參閱 在 Liberty for z/OS 上啟用 z/OS 授權服務。
- 若要完成此作業，您必須具有 `mqwebuser.xml` 檔案的寫入權，以及定義安全管理程式設定檔的權限。

註: V9.2.0.25 從 IBM MQ 9.2.0 Fix Pack 25，已更新範例配置檔 `zos_saf_registry.xml`，以移除重複的 safAuthorization 項目。

此更新可修正下列問題: 當 z/OS 上的 MQ Console 升級至 WebSphere Liberty Profile 22.0.0.12 或更新版本的層次時，即會發生 ICH408I 錯誤: 從 IBM MQ 9.2.0 CSU 8。 不支援具有多個 safAuthorization 陳述式，當不在 EBJROLE 類別中 MQWebAdmin 或 MQWebAdminRO 角色的使用者嘗試透過 MQ Console 存取 z/OS 佇列管理程式時，可能會導致 ICH408I 錯誤。

racRouteLog 的預設值是 NONE，指定嘗試記載的存取權類型。 如果您需要其他報告或記錄來進行安全審核，請參閱 SAF 授權 (safAuthorization)，以取得相關資訊。

關於這項作業

SAF 介面可讓 mqweb 伺服器呼叫外部安全管理程式來進行 IBM MQ Console 和 REST API 的鑑別和授權檢查。

程序

1. 遵循 在 Liberty for z/OS 上啟用 z/OS 授權服務 中的步驟，以授與您 mqweb 伺服器存取權來使用 z/OS 授權服務。

   用於啟動 Angel Process 的範例 JCL 位於 USS_ROOT/web/templates/zos/procs/bbgzangl.jcl 中，其中 USS_ROOT 是 z/OS UNIX System Services (z/OS UNIX) 中安裝 z/OS UNIX 元件的路徑。

   在 bbgzangl.jcl 中，將 SET ROOT 陳述式變更為指向 USS_ROOT/web，例如 /usr/lpp/mqm/V9R2M0/web。

   如需停止和啟動 Angel Process 的進一步資訊，請參閱 在 z/OS。
2. 遵循 Liberty: 設定系統授權機能 (SAF) 未經鑑別的使用者 中的步驟，來建立 Liberty 所需的未經鑑別使用者。
3. 從下列路徑複製 `zos_saf_registry.xml` 檔案: `PathPrefix /web/mq/samp/configuration`，其中 `PathPrefix` 是 z/OS UNIX Components 安裝路徑。
4. 將範例檔放在 *WLP_user_directory*/servers/mqweb 目錄中，其中 *WLP_user_directory* 是在執行 **crtmqweb** Script 以建立 mqweb 伺服器定義時指定的目錄。
5. 選擇性的: 如果您先前已在 `mqwebuser.xml` 中變更任何配置設定，請將它們複製到範例檔。
6. 刪除現有的 `mqwebuser.xml` 檔案，並將範例檔案重新命名為 `mqwebuser.xml`。
7. 自訂 `mqwebuser.xml` 中的 **safCredentials** 元素。
   a. 將 **profilePrefix** 設為您的 Liberty 伺服器唯一的名稱。 如果您在單一系統上執行多個 mqweb 伺服器，則需要為每一部伺服器選擇不同的名稱; 例如 MQWEB920 及 MQWEB915。
   b. 將 **unauthenticatedUser** 設為步驟 第 432 頁的『2』中所建立未經鑑別使用者的名稱。
8. 將 mqweb 伺服器 APPLID 定義為 RACF。

   APPLID 資源名稱是您在步驟 第 432 頁的『7』的 **profilePrefix** 屬性中指定的值。 下列範例在 RACF 中定義 mqweb 伺服器 APPLID:

   ```
   RDEFINE APPL profilePrefix UACC(NONE)
   ```
9. 授與所有使用者或群組對 APPL 類別中 mqweb 伺服器 APPLID 的 MQ Console 或 REST API READ 存取權進行鑑別。

您也必須對步驟 第 432 頁的『2』中定義的未經鑑別使用者執行此動作。 下列範例授與使用者對 RACF 中 mqweb 伺服器 APPLID 的 READ 存取權:

```
PERMIT profilePrefix CLASS(APPL) ACCESS(READ) ID(userID)
```

10. 使用 **SETROPTS** RACF 指令來重新整理儲存體內 RACLISTed APPL 類別設定檔:

    ```
    SETROPTS RACLIST(APPL) REFRESH
    ```

11. 在 EJBROLE 類別中定義必要的設定檔，讓使用者能夠存取 MQ Console 和 REST API 中的角色。

    下列範例定義 RACF 中的設定檔，其中 **profilePrefix** 是在步驟 第 432 頁的『7』中指定給 **profilePrefix** 屬性的值。

```
RDEFINE EJBROLE profilePrefix.com.ibm.mq.console.MQWebAdmin UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.console.MQWebAdminRO UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.console.MQWebUser UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MQWebAdmin UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MQWebAdminRO UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MQWebUser UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MFTWebAdmin UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MFTWebAdminRO UACC(NONE)
```

12. 授與使用者對 MQ Console 及 REST API 中角色的存取權。

    如果要這麼做，請授與使用者或群組對步驟 第 433 頁的『11』所建立之 EBJROLE 類別中的一或多個設定檔的 READ 存取權。 如需角色的相關資訊，請參閱 第 434 頁的『IBM MQ Console 和 REST API 上的角色』。

    下列範例提供使用者對 RACF 中 REST API 的 MQWebAdmin 角色的存取權，其中 **profilePrefix** 是在步驟 第 432 頁的『7』中指定給 **profilePrefix** 屬性的值。

```
PERMIT profilePrefix.com.ibm.mq.rest.MQWebAdmin CLASS(EJBROLE) ACCESS(READ) ID(userID)
```

結果

您已設定 IBM MQ Console 和 REST API 的 SAF 鑑別。

下一步

選擇使用者鑑別的方式:

IBM MQ Console 鑑別選項

- 讓使用者使用記號鑑別進行鑑別。 在此情況下，使用者會在 IBM MQ Console 登入畫面中輸入使用者 ID 和密碼。 會產生 LTPA 記號，可讓使用者保持登入狀態並獲得設定時間量的授權。 不需要進一步配置即可使用此鑑別選項，但您可以選擇性地配置 LTPA 記號的期限間隔。 如需相關資訊，請參閱 配置 LTPA 記號期限間隔。
- 讓使用者使用用戶端憑證進行鑑別。 在此情況下，使用者不會使用使用者 ID 或密碼來登入 IBM MQ Console，而是改用用戶端憑證。 如需相關資訊，請參閱 第 438 頁的『搭配使用用戶端憑證鑑別與 REST API 及 IBM MQ Console』。

REST API 鑑別選項

- 讓使用者使用 HTTP 基本鑑別進行鑑別。 在此情況下，使用者名稱及密碼會編碼，但不會加密，並隨每一個 REST API 要求一起傳送，以鑑別及授權該要求的使用者。 為了使此鑑別安全，您必須使用安全連線。 也就是說，您必須使用 HTTP。 如需相關資訊，請參閱 第 441 頁的『搭配使用 HTTP 基本鑑別與 REST API』。
- 讓使用者使用記號鑑別進行鑑別。 在此情況下，使用者會使用 HTTP POST 方法，將使用者 ID 及密碼提供給 REST API `login` 資源。 會產生 LTPA 記號，可讓使用者保持登入狀態並獲得設定時間量的授權。 如需相關資訊，請參閱第 442 頁的『搭配 REST API 使用記號型鑑別』。 您可以配置 LTPA 記號的期限間隔。 如需相關資訊，請參閱 配置 LTPA 記號。
- 讓使用者使用用戶端憑證進行鑑別。 在此情況下，使用者不會使用使用者 ID 或密碼來登入 REST API，而是改用用戶端憑證。 如需相關資訊，請參閱 第 438 頁的『搭配使用用戶端憑證鑑別與 REST API 及 IBM MQ Console』。

IBM MQ Console 和 REST API 上的角色

當您授權使用者和群組使用 IBM MQ Console 或 REST API 時，必須指派下列其中一個可用角色給使用者和群組: **MQWebAdmin**、 **MQWebAdminRO**、 **MQWebUser**、 **MFTWebAdmin** 及 **MFTWebAdminRO**。 每一個角色提供不同層次的專用權來存取 IBM MQ Console 和 REST API，並決定在嘗試容許的作業時所使用的安全環境定義。

註: 除了 **MQWebUser** 角色之外，使用者 ID 不區分大小寫。 如需此角色的特定需求，請參閱 第 434 頁的『MQWebUser』 。

MQWebAdmin
獲指派此角色的使用者或群組可以執行所有管理作業，並在用來啟動 mqweb 伺服器之作業系統使用者 ID 的安全環境定義下運作。

具有此角色的使用者或群組無權存取下列 REST 服務:

- MFT 的 REST API 。 若要使用這些服務，使用者或群組也必須獲指派 **MFTWebAdmin** 或 **MFTWebAdminRO** 角色。
- messaging REST API。 若要使用 messaging REST API，使用者必須獲指派 **MQWebUser** 角色。

MQWebAdminRO
此角色提供 IBM MQ Console 或 REST API 的唯讀存取權。 獲指派此角色的使用者或群組可以執行下列作業:

- 顯示及查詢 IBM MQ 物件 (例如佇列及通道) 的作業。
- 瀏覽佇列上的訊息。

獲指派此角色的使用者或群組會在用來啟動 mqweb 伺服器之作業系統使用者 ID 的安全環境定義下運作。

具有此角色的使用者或群組無權存取下列 REST 服務:

- MFT 的 REST API 。 若要使用這些服務，使用者或群組也必須獲指派 **MFTWebAdmin** 或 **MFTWebAdminRO** 角色。
- messaging REST API。 若要使用 messaging REST API，使用者必須獲指派 **MQWebUser** 角色。

MQWebUser
獲指派此角色的使用者或群組可以執行授與使用者 ID 在佇列管理程式上執行的任何作業。 例如:

- 在 IBM MQ 物件 (例如通道) 上啟動和停止作業。
- 定義及設定 IBM MQ 物件 (例如佇列及通道) 的作業。
- 顯示及查詢 IBM MQ 物件 (例如佇列及通道) 的作業。
- 使用 messaging REST API 來放置及取得訊息。

獲指派此角色的使用者或群組會在主體的安全環境定義下運作，且只能執行授與使用者 ID 在佇列管理程式上執行的作業。

因此，在 IBM MQ 內必須為 mqweb 使用者登錄中定義的使用者或群組提供權限，該使用者才能執行任何作業。 透過使用此角色，您可以精細控制哪些使用者在使用 IBM MQ Console 及 REST API 時具有特定 IBM MQ 資源的存取權類型。

註:

- 獲指派此角色的使用者 ID 長度上限為 12 個字元。
- 在 mqweb 使用者登錄及 IBM MQ 系統上，使用者 ID 的大小寫必須相同。 如果使用者 ID 的大小寫不同，則使用者可能由 IBM MQ Console 及 REST API 鑑別，但未獲授權使用 IBM MQ 資源。

MFTWebAdmin
獲指派此角色的使用者或群組可以執行所有 MFT REST 作業，並在用來啟動 mqweb 伺服器之作業系統使用者 ID 的安全環境定義下運作。

具有此角色的使用者或群組無權存取任何 IBM MQ REST API 服務。 若要使用這些服務，還必須為使用者或群組指派 **MQWebAdmin**、 **MQWebAdminRO** 或 **MQWebUser** 角色。

MFTWebAdminRO

此角色提供對 REST API for MFT 的唯讀存取權。 獲指派此角色的使用者或群組可以執行唯讀作業 (GET 要求)，例如清單傳送及清單代理程式。

獲指派此角色的使用者或群組會在用來啟動 mqweb 伺服器之作業系統使用者 ID 的安全環境定義下運作。

具有此角色的使用者或群組無權存取任何 IBM MQ REST API 服務。 若要使用這些服務，還必須為使用者或群組指派 **MQWebAdmin**、 **MQWebAdminRO** 或 **MQWebUser** 角色。

如需配置使用者和群組以使用這些角色的相關資訊，請參閱 第 425 頁的『配置使用者和角色』。

重疊角色

使用者或群組可以獲指派多個角色。 當使用者在此狀況下執行作業時，會使用適用於該作業的最高專用權角色。 例如，如果具有角色 **MQWebAdminRO** 及 **MQWebUser** 的使用者執行查詢佇列作業，則會使用 **MQWebAdminRO** 角色，並在啟動 Web 伺服器之系統使用者 ID 的環境定義下嘗試該作業。 如果相同使用者執行定義作業，則會使用 **MQWebUser** 角色，並在主體的環境定義下嘗試該作業。

ALW 將 IBM MQ Console 提供的憑證變更為您的瀏覽器

您可以配置 IBM MQ Console，以呈現您自己的 CA 簽章憑證來進行鑑別。 這樣做會移除 Web 瀏覽器在存取 IBM MQ Console 主控台時所呈現的自簽憑證警告

開始之前

配置要授權使用 IBM MQ Console 的使用者、群組及角色。 如需相關資訊，請參閱 第 425 頁的『配置使用者和角色』。

關於這項作業

主控台安全是由 IBM MQ 安裝所使用的 IBM WebSphere Application Server Liberty 所提供。

若要變更此伺服器提供給瀏覽器的憑證，您需要:

1. 將您要呈現的憑證新增至 Web 伺服器金鑰儲存庫。
2. 標示憑證。
3. 編輯 `mqwebuser.xml` 檔案，以關閉預設安全配置。
4. 在 `mqwebuser.xml` 檔中開啟您自己的安全配置，並指定您要呈現的憑證。

此程序假設您:

- 使用 AIX, Linux, and Windows 系統。
- 特許使用者。

附註:

- 下列範例會使用在 Linux 機器上發出的指令 (即 **ls**，而不是 Windows 機器上使用的 **dir**) 來建立並使用自簽憑證。
- 這會顯示概念，但不會移除瀏覽器警告。
- 若要移除瀏覽器警告，您必須提供 CA 簽章憑證。

程序

1. 如果 Liberty 伺服器正在執行中，請在指令行上輸入 **endmqweb** 指令來停止伺服器。
2. 將憑證新增至 Liberty 應用程式伺服器使用的金鑰儲存庫，以便它可以尋找憑證並將其呈現給 Web 瀏覽器。

   a) 發出下列指令來移至金鑰儲存庫位置，並列出輸出:

```
cd /var/mqm/web/installations/Installation1/servers/mqweb/resources/security
ls
```

例如，您會看到下列輸出，其中顯示名稱為 key.jks 的金鑰儲存庫:

```
/var/mqm/web/installations/Installation1/servers/mqweb/resources/security$
ls key.jks ltpa.keys
```

b) 建立自簽憑證:

若要建立自簽憑證 (基於教育目的，並以 password 密碼新增至 key.jks)，請發出下列指令:

```
runmqckm -cert -create -db key.jks -pw password -dn
"cn=QueueManager,o=IBM,c=UK" -label myowncertificate
```

-dn 旗標可讓您指定憑證上顯示的值。

c) 發出下列指令，驗證您已順利新增憑證:

```
runmqckm -cert -list -db key.jks -pw password
```

例如，您會看到下列輸出，其中顯示已新增憑證及其標籤，以及伺服器目前使用標示為 default 的憑證:

```
/var/mqm/web/installations/Installation1/servers/mqweb/resources/security
$ runmqckm -cert -list -db key.jks -pw password
Certificates in database /var/mqm/web/installations/Installation1/servers/mqweb/resources/
security/key.jks
   default
   myown certificate
```

3. 編輯 mqwebuser.xml 檔案，使伺服器提供新的憑證。

a) 移至 mqwebuser.xml 檔案的位置，然後在您選擇的文字編輯器中開啟它以進行編輯，在此情況下為 *nano*

```
cd /var/mqm/web/installations/Installation1/servers/mqweb
nano mqwebuser.xml
```

b) 關閉預設安全配置。

註銷下列這一行，方法是將 <!-- 新增至程式碼行開頭，並將 --> 新增至程式碼行結尾:

```
<!--
<sslDefault sslRef="mqDefaultSSLConfig"/>
-->
```

c) 啟用並指定您自己的配置。

若要執行此動作，請執行下列程序:

i) 透過從程式碼區塊的開頭移除 <!-- ，並從程式碼區塊的結尾移除 --> ，將下列程式碼行解除註解。

```
<!--
<keyStore id="defaultKeyStore" location="key.jks" type="JKS" password="password"/>
<keyStore id="defaultTrustStore" location="trust.jks" type="JKS" password="password"/>
<ssl id="thisSSLConfig" clientAuthenticationSupported="true" keyStoreRef="defaultKeyStore"
serverKeyAlias="default" trustStoreRef="defaultTrustStore" sslProtocol="TLSv1.2"/>
<sslDefault sslRef="thisSSLConfig"/>
-->
```

ii) **請勿變更程式碼區塊的第一行** ，因為這一行指定主控台用來儲存其個人憑證的金鑰儲存庫。

iii) **註銷程式碼區塊的第二行**，因為這一行指定主控台將在其中尋找用戶端憑證的信任儲存庫。 當您使用記號鑑別時，您尚未建立信任儲存庫，將程式碼行保留在中，會在主控台啟動時造成錯誤。

iv) 在程式碼區塊的第三行中 **將 serverKeyAlias= "default" 變更為 serverKeyAlias= "myowncertificate"** ，並讓其他所有項目保持相同。

v) **不要變更程式碼區塊的最後一行**，因為這會告知伺服器使用您剛才指定的配置。

程式碼區塊現在看起來如下:

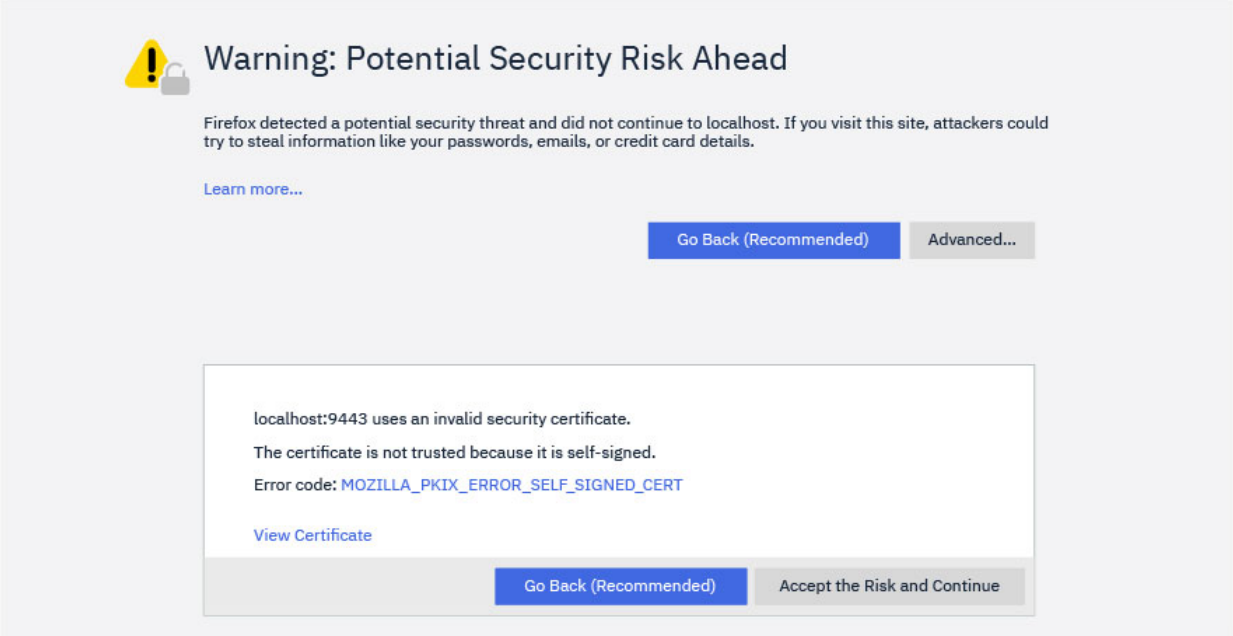
```
<keyStore id="defaultKeyStore" location="key.jks" type="JKS" password="password"/>
<!-- Commenting out the defaultTrustStore as otherwise we get errors (viewable in the messages.log file
in the logs folder) j
<keyStore id="defaultTrustStore" location="trust.jks" type="JKS" password="password"/>
-->
<ssl id="thisSSLConfig" clientAuthenticationSupported="true" keyStoreRef="defaultKeyStore"
serverKeyAlias="myowncertificate" trustStoreRef="defaultTrustStore" sslProtocol="TLSv1.2"/>
<sslDefault sslRef="thisSSLConfig"/>
```

4. 使用 **strmqweb** 指令重新啟動 Web 伺服器。

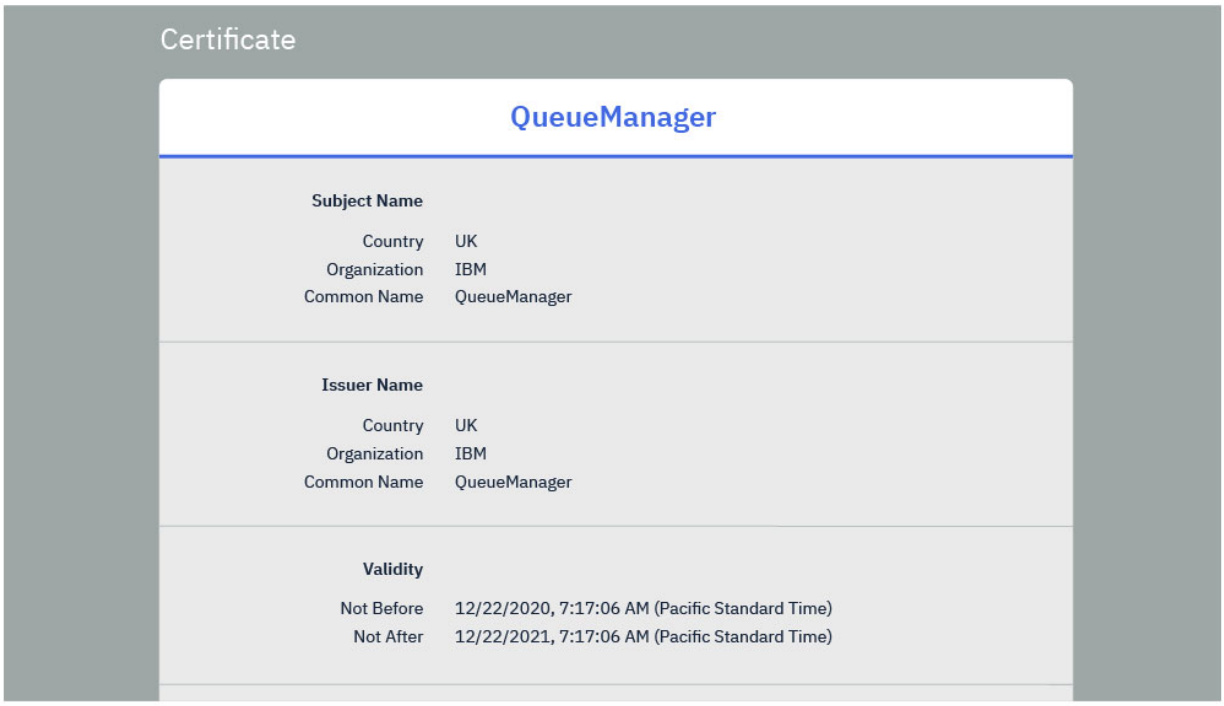
結果

當 Web 伺服器啟動時，瀏覽至 IBM MQ Console 並重新整理。 如果您使用您所建立的自簽憑證，並使用步驟 第 435 頁的『2』 及 第 436 頁的『3』中之前文字所說明的程序，則會看到安全性警告。

請注意，此警告的格式視您使用的瀏覽器而定。



如果您按一下 **檢視憑證**，當您在步驟 第 436 頁的『2.b』中建立憑證時，您會看到它具有您在 **-dn** 旗標上提供的詳細資料。



不過，如果您使用 CA 簽章憑證，瀏覽器會信任您發出下列指令所新增的憑證:

```
runmqckm -cert -add -db key.jks -pw password -label myCACertificate
```

其中 `myCACertificate` 是含有您的 CA 憑證之檔案的檔案路徑，您會直接進入登入頁面。

小心: 如果您使用 CA 簽章憑證，且該 CA 憑證是憑證鏈的一部分，則必須從主要 CA 憑證開始，將所有憑證新增至該憑證鏈。 如需相關資訊，請參閱 第 255 頁的『將 CA 憑證或自簽憑證的公用部分新增至 AIX, Linux, and Windows 上的金鑰儲存庫』。

ALW 搭配使用用戶端憑證鑑別與 REST API 及 IBM MQ Console

您可以將用戶端憑證對映至主體，以鑑別 IBM MQ Console 及 REST API 使用者。

開始之前

- 將使用者、群組和角色配置成獲授權使用 IBM MQ Console 和 REST API。 如需相關資訊，請參閱第 425 頁的『配置使用者和角色』。
- 當您使用 REST API 時，可以在 `login` 資源上使用 HTTP GET 方法來查詢現行使用者的認證，並提供用戶端憑證來鑑別要求。 此要求會傳回使用者名稱及獲指派使用者之角色的相關資訊。 如需相關資訊，請參閱 GET `/login`。
- 當您將用戶端憑證對映至主體以鑑別使用者時，會使用用戶端憑證的識別名稱來比對所配置使用者登錄中的使用者:
  - 對於基本登錄，「通用名稱 (CN)」符合使用者。 例如， `CN=Fred, O=IBM, C=GB` 符合使用者名稱 `Fred`。
  - 若為 LDAP 登錄，依預設會比對 LDAP 的完整識別名稱。 您可以設定過濾器和對映來自訂比對。 如需相關資訊，請參閱 WebSphere Liberty 說明文件中的 Liberty :LDAP 憑證對映模式 。

關於這項作業

當使用者使用用戶端憑證進行鑑別時，會使用憑證來取代使用者名稱及密碼。 對於 REST API，用戶端憑證隨每一個 REST 要求一起提供，以鑑別使用者。 對於 IBM MQ Console，當使用者使用憑證登入時，無法將使用者登出。

此程序假設下列資訊:

- `mqwebuser.xml` 檔案基於下列其中一個範例:
  - `basic_registry.xml`
  - `local_os_registry.xml`
  - `ldap_registry.xml`
- 您正在使用 AIX, Linux, and Windows 系統。
- 您是 特許使用者。

若要使用 z/OS 上的 RACF 金鑰環來配置用戶端憑證鑑別，請遵循 第 448 頁的『在 z/OS 上為 REST API 和 IBM MQ Console 配置 TLS』中的程序。

註: 下列程序概述搭配使用用戶端憑證與 IBM MQ Console 及 REST API 所需的步驟。 為了方便開發人員，這些步驟詳細說明如何建立及使用自簽憑證。 不過，對於正式作業，請使用從憑證管理中心取得的憑證。

程序

1. 在指令行上輸入 **strmqweb** 指令，以啟動 mqweb 伺服器。
2. 建立用戶端憑證:
   a) 建立 PKCS#12 金鑰儲存庫:
      i) 在指令行上輸入 **strmqikm** 指令，以開啟 IBM Key Management 工具。
      ii) 從 IBM Key Management 工具中的 **金鑰資料庫檔** 功能表，按一下 **新建**。
      iii) 從 **金鑰資料庫類型** 清單中選取 **PKCS12** 。
      iv) 選取儲存金鑰儲存庫的位置，並在 **檔名** 欄位中輸入適當的名稱。 例如： `user.p12`
      v) 提示時設定密碼。
   b) 透過建立自簽憑證或從憑證管理中心取得憑證來建立憑證:
      - 建立自簽憑證:
        i) 按一下 **新建自簽**。
        ii) 在 **金鑰標籤** 欄位中輸入 `user` 。
        iii) 如果您使用基本使用者登錄，請在 **通用名稱** 欄位中輸入使用者登錄中的使用者名稱。 例如，`mqadmin`。 若為 LDAP 使用者登錄，請確定憑證的識別名稱符合 LDAP 登錄中的識別名稱。
        iv) 按一下**確定**。
      - 從憑證管理中心取得憑證。 CA 憑證必須在識別名稱 (DN) 欄位的通用名稱 (CN) 內包括適當的使用者名稱:
        i) 要求新憑證。 從 **建立** 功能表中，按一下 **新建憑證申請**。
        ii) 在 **金鑰標籤** 欄位中，輸入憑證標籤。
        iii) 如果您使用基本使用者登錄，請在 **通用名稱** 欄位中，輸入憑證適用之使用者的使用者名稱。

        如果您使用本端 OS 登錄， **通用名稱** 欄位必須符合本端 OS 使用者 ID。

        若為 LDAP 使用者登錄，請確定憑證的識別名稱符合 LDAP 登錄中的識別名稱。
        iv) 輸入或選取其餘欄位的值 (視適用情況而定)。
        v) 選擇儲存憑證申請的位置，以及憑證申請的檔名，然後按一下 **確定**。
        vi) 將憑證申請檔案傳送至憑證管理中心 (CA)。

vii) 當您從 CA 取得憑證時，請在指令行上輸入 **strmqikm** 指令，以開啟 IBM Key Management 工具。

viii) 從 IBM Key Management 工具中的 **金鑰資料庫檔** 功能表，按一下 **開啟**。

ix) 選取保留用戶端憑證的 PKCS#12 金鑰儲存庫。 例如：`user.p12`

x) 按一下 **接收**，選取適當的憑證，然後按一下 **確定**。

3. 擷取用戶端憑證的公用部分:
   a) 在指令行上輸入 **strmqikm** 指令，以開啟 IBM Key Management 工具。
   b) 從 IBM Key Management 工具中的 **金鑰資料庫檔** 功能表，按一下 **開啟**。
   c) 選取保留用戶端憑證的 PKCS#12 金鑰儲存庫。 例如：`user.p12`
   d) 從 IBM Key Management 工具的憑證清單中選取用戶端憑證。
   e) 按一下 **擷取憑證**。
   f) 選取儲存憑證的位置，並在 **憑證檔名** 欄位中輸入適當的檔名。 例如，`user.arm`。
4. 將用戶端憑證的公用部分匯入 mqweb 伺服器信任金鑰儲存庫作為簽章者憑證，以便伺服器可以驗證用戶端憑證:
   a) 建立供 mqweb 伺服器使用的 `trust.jks` 金鑰儲存庫 (如果尚未存在的話):
      i) 從 IBM Key Management 工具中的 **金鑰資料庫檔** 功能表，按一下 **新建**。
      ii) 從**金鑰資料庫類型**清單中選取 **JKS**。
      iii) 按一下 **瀏覽** ，並導覽至: `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb/resources/security`。

      此目錄應該已包含 `key.jks` 檔案。 如果 `trust.jks` 檔案已存在，請開啟現有的檔案，而不是改寫它。

      iv) 在 **檔名** 欄位中輸入 `trust.jks` 。
      v) 提示時設定密碼。
   b) 從下拉功能表中，選取 **簽章者憑證**。
   c) 按一下**新增** 。
   d) 選取適當的 arm 檔案，然後按一下 **確定**。 例如，選取 `user.arm`。
   e) 輸入憑證的標籤。
5. 變更 mqweb 伺服器金鑰儲存庫的密碼:
   a) 從**金鑰資料庫檔**功能表，按一下**開啟**。
   b) 從**金鑰資料庫類型**清單中選取 **JKS**。
   c) 按一下 **瀏覽** ，並導覽至 *MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb/resources/security
   d) 選取 `key.jks` 金鑰儲存庫，然後按一下 **開啟**。
   e) 提示時輸入密碼。 預設密碼為 `password`。
   f) 從 **金鑰資料庫檔** 功能表中，按一下 **變更密碼**。
   g) 輸入金鑰儲存庫的新密碼。
6. 在 `mqwebuser.xml` 檔中啟用用戶端憑證鑑別:

   可以在下列路徑上找到 `mqwebuser.xml` 檔案: *MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb

   a) 解除註解 `mqwebuser.xml` 檔案中啟用用戶端憑證鑑別的區段。 該區段包含下列文字:

```
<keyStore id="defaultKeyStore" location="key.jks" type="JKS" password="password"/>
    <keyStore id="defaultTrustStore" location="trust.jks" type="JKS" password="password"/>
    <ssl id="thisSSLConfig" clientAuthenticationSupported="true"
keyStoreRef="defaultKeyStore"
          trustStoreRef="defaultTrustStore" sslProtocol="TLSv1.2"
serverKeyAlias="default"/>
    <sslDefault sslRef="thisSSLConfig"/>
```

b) 檢查 **serverKeyAlias** 值是否符合伺服器憑證的名稱。 如果您使用預設伺服器憑證，則值是正確的。

c) 將 defaultKeyStore 的 **password** 值變更為 key.jks 金鑰儲存庫的密碼編碼版本:

i) 從 *MQ_INSTALLATION_PATH*/web/bin 目錄，在指令行輸入下列指令:

```
securityUtility encode password
```

ii) 將此指令的輸出放在 defaultKeyStore 的 **password** 欄位中。

d) 變更 defaultTrustStore 的 **password** 值，以符合 trust.jks 金鑰儲存庫的密碼:

i) 從 *MQ_INSTALLATION_PATH*/web/bin 目錄，在指令行輸入下列指令:

```
securityUtility encode password
```

ii) 將此指令的輸出放在 defaultTrustStore 的 **password** 欄位中。

e) 從 mqwebuser.xml 檔案中移除或註銷下列行:

```
<sslDefault sslRef="mqDefaultSSLConfig"/>
```

7. 在指令行上輸入 **endmqweb** 指令，以停止 mqweb 伺服器。
8. 在指令行上輸入 **strmqweb** 指令，以啟動 mqweb 伺服器。
9. 使用用戶端憑證來鑑別:

- 若要搭配使用用戶端憑證與 IBM MQ Console，請將用戶端憑證安裝至用來存取 IBM MQ Console 的 Web 瀏覽器。 例如，安裝用戶端憑證 user.p12 作為個人憑證。
- 若要搭配使用用戶端憑證與 REST API，請為用戶端憑證提供每一個 REST 要求。 當您使用 HTTP POST、PATCH 或 DELETE 方法時，必須提供用戶端憑證的額外鑑別，以防止偽造跨網站要求攻擊。 也就是說，額外鑑別用來確認認證擁有者正在使用用來鑑別要求的認證。

  此額外鑑別由 ibm-mq-rest-csrf-token HTTP 標頭提供。 將 ibm-mq-csrf-token 標頭的值設為包括空白在內的任何值，然後提交要求。

範例

重要: 在此範例中，並非所有 cURL 實作都支援自簽憑證，因此您必須使用執行的 cURL 實作。

下列 cURL 範例顯示如何在佇列管理程式 QM1 上建立具有用戶端憑證鑑別的新佇列 Q1。 此 cURL 指令的確切配置取決於建置 cURL 所針對的程式庫。 此範例以 Windows 系統為基礎，具有針對 OpenSSL 建置的 cURL 。

- 搭配使用 HTTP POST 方法與佇列資源，以用戶端憑證進行鑑別，並以任意值包括 ibm-mq-rest-csrf-token HTTP 標頭。 此值可以是任何值，包括空白。 --cert-type 旗標指定憑證是 PKCS#12 憑證。 --cert 旗標指定憑證的位置，後面接著冒號、:，然後是憑證的密碼:

```
curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X POST -
-cert-type P12 --cert c:\user.p12:password
-H "ibm-mq-rest-csrf-token: value"
-H "Content-Type: application/json" --data "{\"name\":\"Q1\"}"
```

搭配使用 HTTP 基本鑑別與 REST API

REST API 的使用者可以透過在 HTTP 標頭內提供其使用者 ID 和密碼來進行鑑別。 若要搭配使用此鑑別方法與 HTTP 方法 (例如 POST、PATCH 及 DELETE) ，也必須提供 ibm-mq-rest-csrf-token HTTP 標頭，以及使用者 ID 和密碼。

開始之前

- 配置要授權使用 REST API 的使用者、群組及角色。 如需相關資訊，請參閱 第 425 頁的『配置使用者和角色』。

- 請確定已啟用 HTTP 基本鑑別。 檢查 mqwebuser.xml 檔案中是否存在下列 XML，且未註銷。 此 XML 必須在 <featureManager> 標籤內:

```
<feature>basicAuthenticationMQ-1.0</feature>
```

z/OS 在 z/OS 上，您必須是具有 mqwebuser.xml 寫入權的使用者，才能編輯此檔案。

Multi 在所有其他作業系統上，您必須是 特許使用者，才能編輯 mqwebuser.xml 檔案。

- 當您傳送 REST 要求時，請確定您使用安全連線。 由於使用者名稱和密碼組合已編碼，但未加密，當您對 REST API 使用 HTTP 基本鑑別時，必須使用安全連線 (HTTPS)。
- 您可以在 login 資源上使用 HTTP GET 方法來查詢現行使用者的認證，並提供基本鑑別資訊來鑑別要求。 此要求會傳回使用者名稱及獲指派使用者之角色的相關資訊。 如需相關資訊，請參閱 GET /login。

程序

1. 以冒號和密碼來連結使用者名稱。 請注意，使用者名稱區分大小寫。
   例如，使用者名稱 admin 及密碼 admin 會變成下列字串:

```
admin:admin
```

2. 以 base64 編碼來編碼這個使用者名稱和密碼字串。
3. 將這個已編碼的使用者名稱和密碼包含在 HTTP Authorization: Basic 標頭中。
   例如，使用已編碼的使用者名稱 admin，以及密碼 admin，會建立下列標頭:

```
Authorization: Basic YWRtaW46YWRtaW4=
```

4. 當您使用 HTTP POST、PATCH 或 DELETE 方法時，必須提供額外鑑別，以及使用者名稱和密碼。
   此額外鑑別由 ibm-mq-rest-csrf-token HTTP 標頭提供。 ibm-mq-rest-csrf-token HTTP 標頭必須存在於要求中，但其值可以是任何值 (包括空白)。
5. 將 REST 要求提交至具有適當標頭的 IBM MQ 。

範例

下列範例顯示如何在 Windows 系統上，在佇列管理程式 QM1 上建立具有基本鑑別的新佇列 Q1。 該範例使用 cURL:

- 搭配使用 HTTP POST 方法與佇列資源，以基本鑑別進行鑑別，並以任意值包含 ibm-mq-rest-csrf-token HTTP 標頭。 此值可以是任何值，包括空白:

```
curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X POST
-u mqadmin:mqadmin
-H "ibm-mq-rest-csrf-token: value"
-H "Content-Type: application/json" --data "{\"name\":\"Q1\"}"
```

搭配 REST API 使用記號型鑑別

REST API 的使用者可以透過使用 HTTP POST 方法向 REST API login 資源提供使用者 ID 和密碼來進行鑑別。 會產生 LTPA 記號，可讓使用者鑑別未來的要求。 這個 LTPA 記號具有字首 LtpaToken2。 使用者可以使用 HTTP DELETE 方法登出，並且可以使用 HTTP GET 方法查詢現行使用者的登入資訊。

開始之前

- 配置要授權使用 REST API 的使用者、群組及角色。 如需相關資訊，請參閱 第 425 頁的『配置使用者和角色』。
- 依預設，包含 LTPA 記號的 Cookie 名稱會以 LtpaToken2 開頭，並包含可在 mqweb 伺服器重新啟動時變更的字尾。 此隨機化 Cookie 名稱容許多個 mqweb 伺服器在相同系統上執行。 不過，如果您想要 Cookie 名稱保持一致值，您可以使用 **setmqweb** 指令來指定 Cookie 所擁有的名稱。 如需相關資訊，請參閱 配置 LTPA 記號。

- 依預設， LTPA 記號 Cookie 會在 120 分鐘之後到期。 您可以使用 **setmqweb** 指令來配置 LTPA 記號 Cookie 的到期時間。 如需相關資訊，請參閱 配置 LTPA 記號。
- 當您傳送 REST 要求時，請確定您使用安全連線。 在 `login` 資源上使用 HTTP POST 方法時，不會加密隨要求一起傳送的使用者名稱及密碼組合。 因此，當您對 REST API 使用記號型鑑別時，必須使用安全連線 (HTTPS)。 依預設，您無法搭配使用 HTTP 與 LTPA 記號鑑別。 您可以將 **secureLTPA** 設為 `False`，讓 LTPA 記號可供不安全的 HTTP 連線使用。 如需相關資訊，請參閱 配置 LTPA 記號。
- 您可以在 `login` 資源上使用 HTTP GET 方法，並提供 LTPA 記號來鑑別要求，以查詢現行使用者的認證。 此要求會傳回使用者名稱及獲指派使用者之角色的相關資訊。 如需相關資訊，請參閱 GET /login。

程序

1. 登入使用者:
   a) 在 `login` 資源上使用 HTTP POST 方法:

   ```
   https://host:port/ibmmq/rest/v1/login
   ```

   在 JSON 要求的內文中包含使用者名稱和密碼，格式如下:

   ```
   {
       "username" : name,
       "password" : password
   }
   ```

   b) 將要求所傳回的 LTPA 記號儲存在本端 Cookie 儲存庫中。 依預設，這個 LTPA 記號的字首為 `LtpaToken2`。
2. 使用儲存的 LTPA 記號作為每一個要求的 Cookie 來鑑別 REST 要求。

   對於使用 HTTP PUT、PATCH 或 DELETE 方法的要求，請包含 `ibm-mq-rest-csrf-token` 標頭。 此標頭的值可以是任何值，包括空白。
3. 登出使用者:
   a) 在 `login` 資源上使用 HTTP DELETE 方法:

   ```
   https://host:9443/ibmmq/rest/v1/login
   ```

   您必須提供 LTPA 記號作為 Cookie 來鑑別要求，並包含 `ibm-mq-rest-csrf-token` 標頭。 此標頭的值可以是任何值，包括空白

   b) 處理指示以從本端 Cookie 儲存庫刪除 LTPA 記號。

   註: 如果未處理指示，且 LTPA 記號仍留在本端 Cookie 儲存庫中，則 LTPA 記號可用來鑑別未來的 REST 要求。 也就是說，在階段作業結束之後，當使用者嘗試向 LTPA 記號進行鑑別時，會建立使用現有記號的新階段作業。

範例

下列 cURL 範例顯示如何在 Windows 系統上，在佇列管理程式 QM1 上使用記號型鑑別建立新佇列 Q1:

- 登入並將字首為 `LtpaToken2` 的 LTPA 記號新增至本端 Cookie 儲存庫。 使用者名稱和密碼資訊包含在 JSON 主體中。 `-c` 旗標指定要在其中儲存記號的檔案位置:

  ```
  curl -k https://localhost:9443/ibmmq/rest/v1/login -X POST
  -H "Content-Type: application/json" --data
  "{\"username\":\"mqadmin\",\"password\":\"mqadmin\"}"
  -c c:\cookiejar.txt
  ```

- 建立佇列。 搭配使用 HTTP POST 方法與佇列資源，並以 LTPA 記號進行鑑別。 使用 `-b` 旗標從 `cookiejar.txt` 檔中擷取字首為 `LtpaToken2` 的 LTPA 記號。 CSRF 保護是由 `ibm-mq-rest-csrf-token` HTTP 標頭的存在所提供:

  ```
  curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X POST -b
  c:\cookiejar.txt -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json"
  --data "{\"name\":\"Q1\"}"
  ```

- 登出並從本端 Cookie 儲存庫刪除 LTPA 記號。 LTPA 記號是使用 `-b` 旗標從 `cookiejar.txt` 檔擷取。 CSRF 保護是由 `ibm-mq-rest-csrf-token` HTTP 標頭的存在所提供。 `cookiejar.txt` 檔的位置由 `-c` 旗標指定，以便從檔案中刪除 LTPA 記號:

```
curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X DELETE
-H "ibm-mq-rest-csrf-token: value" -b c:\cookiejar.txt
-c c:\cookiejar.txt
```

相關參考

POST /login

GET /login

刪除 /login

V 9.2.0 在 IFrame 中內嵌 IBM MQ Console

The HTML <iframe> element can be used to embed one web page into another using an Inline Frame (IFrame). 基於安全理由，依預設無法將 IBM MQ Console 內嵌至 IFrame。 不過，您可以在 mqweb 伺服器上使用 **mqConsoleFrameAncestors** 配置內容來啟用 IFrame。

關於這項作業

mqweb 伺服器維護可使用 IFrame 內嵌 IBM MQ Console 之網頁原點的允許清單。 原點是 URL 架構、網域及埠的組合，例如 `https://example.com:1234`。

您可以在 mqweb 伺服器上使用 **mqConsoleFrameAncestors** 配置內容來指定清單中的項目。

依預設， **mqConsoleFrameAncestors** 為空白，表示 IBM MQ Console 無法內嵌在 IFrame 中。

程序

輸入下列指令，以指定可將 IBM MQ Console 內嵌在 IFrame 中的網頁原點清單:

```
setmqweb properties -k mqConsoleFrameAncestors -v allowedOrigins
```

其中 *allowedOrigins* 是以逗點區隔的原點清單。 每一個來源都應該包含:

- 主機名稱或 IP 位址
- 選用的 URL 架構
- 選用埠號

請注意，主機名稱可以萬用字元 (*) 開頭，埠號也可以使用萬用字元 (*)。
範例原點為:

```
https://example.com:1234
```

這可讓 `https://example.com:1234` 所提供的任何網頁將 IBM MQ Console 內嵌在 IFrame 中。

```
https://*.example.com:*
```

這可讓主機名稱以 `example.com` 結尾且使用任何埠的任何 HTTP 網頁將 IBM MQ Console 內嵌在 IFrame 中。

範例

下列範例容許從 `https://site2.example.com:1234` 或 `https://site2.example.com:1235` 所提供的網頁，將 IBM MQ Console 內嵌在 IFrame 中:

```
setmqweb properties -k mqConsoleFrameAncestors -v
https://site2.example.com:1234,https://site2.example.com:1235
```

配置 REST API 的 CORS

依預設，當 Script 不是來自與 REST API 相同的原點時， Web 瀏覽器不容許 Script (例如 JavaScript) 呼叫 REST API 。 也就是說，未啟用跨原點要求。 您可以配置「跨原點資源共用 (CORS)」，以容許來自指定原點的跨原點要求。

關於這項作業

您可以透過 Web 瀏覽器來存取 REST API ，例如透過 Script。 由於這些要求是從不同的原點到 REST API，Web 瀏覽器會拒絕要求，因為它是跨原點要求。 如果網域、埠或架構不同，則原點不同。

例如，如果您有一個在 `http://localhost:1999/` 管理的 Script ，當您在 `https://localhost:9443/`管理的網站上發出 HTTP GET 時，即會發出跨原點要求。 此要求是跨原點要求，因為埠號與架構 (HTTP) 不同。

您可以透過配置 CORS 並指定容許存取 REST API 的原點，來啟用跨原點要求。

如需 CORS 的相關資訊，請參閱 https://www.w3.org/TR/cors/ 和 https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS。

程序

1. 輸入下列指令，以檢視現行配置:

   ```
   dspmqweb properties -a
   ```

   `mqRestCorsAllowedOrigins` 項目指定容許的原點。 `mqRestCorsMaxAgeInSeconds` 項目指定 Web 瀏覽器可以快取任何 CORS 飛行前檢查結果的時間 (以秒為單位)。
2. 輸入下列指令，以指定容許存取 REST API 的原點:

   ```
   setmqweb properties -k mqRestCorsAllowedOrigins -v allowedOrigins
   ```

   其中 *allowedOrigins* 指定您要容許跨原點要求的來源。 您可以使用以雙引號 "*" 括住的星號，以容許所有跨原點要求。 您可以在以逗點區隔的清單中輸入多個原點，並以雙引號括住。 若要不容許跨原點要求，請輸入空引號作為 *allowedOrigins* 的值。
3. 輸入下列指令，以指定您要容許 Web 瀏覽器快取任何 CORS 飛行前檢查結果的時間 (秒):

   ```
   setmqweb properties -k mqRestCorsMaxAgeInSeconds -v time
   ```

範例

下列範例顯示針對 `http://localhost:9883`、 `https://localhost:1999` 及 `https://localhost:9663` 啟用的跨原點要求。 任何 CORS 預先檢查的快取結果經歷時間上限設為 90 秒:

```
setmqweb properties -k mqRestCorsAllowedOrigins -v "http://localhost:9883,https://
localhost:1999,https://localhost:9663"
setmqweb properties -k mqRestCorsMaxAgeInSeconds -v 90
```

配置 IBM MQ Console 和 REST API 的主機標頭驗證

您可以將 mqweb 伺服器配置為限制對 IBM MQ Console 和 REST API 的存取權，以便只處理與指定容許清單相符的主機標頭所傳送的要求。 如果使用不在允許清單上的主機標頭值，則會傳回錯誤。

關於這項作業

mqweb 伺服器使用虛擬主機來定義可接受主機標頭的允許清單。 如需虛擬主機的相關資訊，請參閱 WebSphere Liberty 說明文件: https://www.ibm.com/docs/SSEQTP_liberty/com.ibm.websphere.wlp.doc/ae/cwlp_virtual_hosts.html

若要完成此作業，您必須是具有足夠專用權來編輯 `mqwebuser.xml` 檔案的使用者:

- z/OS 在 z/OS 上，您必須具有 `mqwebuser.xml` 檔案的寫入權。

- Multi 在所有其他作業系統上，必須是 特許使用者。

程序

1. 開啟 mqwebuser.xml 檔案。 此檔案位於下列其中一個位置:
   - ALW

     在 AIX, Linux, and Windows 上: *MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb
   - z/OS

     在 z/OS 上: *WLP_user_directory*/servers/mqweb

     其中 *WLP_user_directory* 是在執行 **crtmqweb** Script 以建立 mqweb 伺服器定義時指定的目錄。
2. 在 mqwebuser.xml 檔案中新增或解除註解下列程式碼:

```
<virtualHost allowFromEndpointRef="defaultHttpEndpoint" id="default_host">
        <hostAlias>localhost:9080</hostAlias>
</virtualHost>
```

3. 編輯 **<hostAlias>** 欄位，插入您要容許的主機名稱和埠組合。

   此組合可能是您在 mqweb 伺服器配置中使用的主機名稱及埠名稱。 例如，如果您使用預設配置 localhost:9443，則可能想要在 **<hostAlias>** 欄位中使用 localhost:9443 。

   必要的話，您可以在 **<virtualHost>** 標籤內新增多個 **<hostAlias>** 欄位，以容許更多主機名稱及埠組合。 例如，容許使用 HTTP 埠的主機標頭，以及使用 HTTP 埠的主機標頭。

審核

透過啟用佇列管理程式指令及配置事件，可以產生在 IBM MQ Console 及 REST API 中執行之作業的審核記錄，而在 AIX, Linux, and Windows 上，重要狀態變更會記錄在 mqweb 伺服器的日誌檔中。

重大狀態變更

ALW

在 AIX, Linux, and Windows 上， IBM MQ Console 會將重大狀態變更記錄為 mqweb 伺服器日誌中的訊息。每一則訊息都指出要求作業的已鑑別主體名稱。

重要狀態變更 (例如建立、啟動、結束或刪除佇列管理程式的時間) 會以 [AUDIT] 記載層次記載在 mqweb 伺服器 messages.log 及 console.log 檔案中。 每一個日誌項目都指出要求作業的已鑑別主體名稱。

messages.log 和 console.log 檔案位於下列位置:

- ALW 在 AIX, Linux, and Windows 上: *MQ_DATA_PATH*\web\installations*installationName*\servers\mqweb\logs

如需配置 mqweb 伺服器記載層次的相關資訊，請參閱 配置記載。

指令及配置事件

您可以選擇性地在佇列管理程式上啟用指令及配置事件，以提供大部分 IBM MQ Console 及 REST API 活動的相關資訊。 例如，建立通道及查詢佇列會產生指令及配置事件。 如需啟用指令及配置事件的相關資訊，請參閱 控制配置、指令及日誌程式事件。

對於這些指令及配置事件訊息， MQIACF_EVENT_ORIGIN 欄位會設為 MQEVO_REST ，且 MQCACF_EVENT_APPL_IDENTITY 欄位會報告已鑑別主體名稱的前 32 個字元。 如果使用者具有 **MQWebAdmin** 或 **MQWebAdminRO** 角色，則 MQCACF_EVENT_USER_ID 欄位會報告 mqweb 伺服器使用者 ID ，而不是發出指令之主體的使用者名稱。 不過，如果使用者具有 **MQWebUser** 角色，MQCACF_EVENT_USER_ID 會報告發出指令之主體的使用者名稱。

相關概念

第 398 頁的『審核』

您可以使用事件訊息來檢查安全侵入或嘗試侵入。 您也可以使用 IBM MQ Explorer 來檢查系統的安全。

z/OS 上 IBM MQ Console 和 REST API 的安全考量

IBM MQ Console 和 REST API 具有安全特性，可控制使用者是否可以發出、顯示或變更指令。 然後會將指令傳遞至佇列管理程式，並使用佇列管理程式安全來控制是否容許使用者向該特定佇列管理程式發出指令。

程序

1. 請確保 mqweb 伺服器已啟動作業使用者 ID 具有適當的權限，可以發出特定 PCF 指令並存取特定佇列。 如需相關資訊，請參閱 第 447 頁的『mqweb 伺服器啟動作業使用者 ID 所需的權限』。
2. 確保任何獲授與 `MQWebUser` 角色的使用者都具有適當的權限。

   指派給 `MQWebUser` 角色的 IBM MQ Console 及 REST API 使用者在主體的安全環境定義下運作。 這些使用者 ID 只能執行使用者 ID 被授與在佇列管理程式上執行的作業，且需要被授與與 mqweb 伺服器位址空間相同的系統佇列存取權。

   mqweb 伺服器啟動型作業使用者 ID 必須獲得指派給 `MQWebUser` 角色之所有使用者的替代使用者存取權。

   如需為具有 `MQWebUser` 角色的使用者授與適當權限的相關資訊，請參閱 第 448 頁的『存取使用 MQ Console 或 REST API 所需的 IBM MQ 資源』。
3. 選擇性的: 配置 IBM MQ Console 和 REST API 的 TLS。 如需相關資訊，請參閱第 448 頁的『在 z/OS 上為 REST API 和 IBM MQ Console 配置 TLS』。

z/OS mqweb 伺服器啟動作業使用者 ID 所需的權限

在 z/OS 上， mqweb 伺服器啟動型作業使用者 ID 需要特定權限才能發出 PCF 指令及存取系統資源。

mqweb 伺服器已啟動作業使用者 ID 需要:

- 能夠使用 z/OS UNIX System Services 的 z/OS UNIX 使用者 ID (UID)。
- 存取 IBM MQ 安裝中的 `hlq.SCSQAUTH` 及 `hlq.SCSQANL*` 資料集。
- z/OS UNIX System Services 中 IBM MQ 安裝檔案的讀取權。
- 對 **crtmqweb** Script 所建立 Liberty 使用者目錄的讀取及寫入權。
- 連接佇列管理程式的權限。 授與 mqweb 伺服器啟動型作業使用者 ID 對 MQCONN 類別中 `hlq.BATCH` 設定檔的 *READ* 存取權。
- 發出 IBM MQ 指令及存取特定佇列的權限。 這些詳細資料在 第 193 頁的『IBM MQ Console -必要的指令安全設定檔』、 第 174 頁的『系統佇列安全』和 第 183 頁的『環境定義安全的設定檔』中有說明。
- 訂閱 `SYSTEM.FTE` 主題的權限，以便將 REST API 用於 MFT。 授與 mqweb 伺服器已啟動作業使用者 ID *ALTER* 對 MXTOPIC 類別中 `hlq.SUBSCRIBE.SYSTEM.FTE` 設定檔的存取權。
- 如果您要配置 SAF 登錄，請存取各種安全設定檔。 如需相關資訊，請參閱 第 431 頁的『配置 IBM MQ Console 和 REST API 的 SAF 登錄』。

連線鑑別

如果佇列管理程式已配置為需要所有批次應用程式都提供有效的使用者 ID 及密碼，則您必須透過設定 CHKLOCL (REQUIRED) ， 將 MQCONN 類別中 `hlq.BATCH` 設定檔的 *UPDATE* 存取權提供給 mqweb 伺服器已啟動作業使用者 ID。

此權限會導致 mqweb 伺服器已啟動作業使用者 ID 的連線鑑別以 CHKLOCL (OPTIONAL) 模式運作。

如果您尚未將佇列管理程式配置成要求所有批次應用程式都提供有效的使用者 ID 和密碼，則只要將 MQCONN 類別中 `hlq.BATCH` 設定檔的 *READ* 存取權提供給啟動 mqweb 伺服器作業的使用者 ID 即可。

如需 CHCKLOCL 的相關資訊，請參閱 第 166 頁的『在本端連結的應用程式上使用 CHCKLOCL』。

存取使用 MQ Console 或 REST API 所需的 IBM MQ 資源

在 MQ Console 或 REST API 中由 MQWebUser 角色中的使用者執行的作業會在使用者的安全環境定義下進行。

關於這項作業

如需 MQ Console 及 REST API 中角色的相關資訊，請參閱 第 434 頁的『IBM MQ Console 和 REST API 上的角色』 。

使用下列程序，以 MQWebUser 角色授與使用者對使用 MQ Console 或 REST API 所需的佇列管理程式資源的存取權。

程序

1. 授與 mqweb server started task 使用者 ID 替代使用者對 MQWebUser 角色中每一個使用者 ID 的存取權。

   在使用者將透過 MQ Console 或 REST API 管理的每個佇列管理程式上執行此動作。

   您可以使用下列範例 RACF 指令，將 mqweb server started task 使用者 ID 替代使用者存取權授與 MQWebUser 角色中的使用者:

   ```
   RDEFINE MQADMIN hlq.ALTERNATE.USER.userId UACC(NONE)
   PERMIT hlq.ALTERNATE.USER.userId CLASS(MQADMIN) ACCESS(UPDATE) ID(mqwebUserId)
   SETROPTS RACLIST(MQADMIN) REFRESH
   ```

   其中:

   hlq
   是設定檔字首，可以是佇列管理程式名稱或佇列共用群組名稱

   userId
   使用者是否具有 MQWebUser 角色

   mqwebUserId
   是 mqweb server started task 使用者 ID

   註: 如果您使用大小寫混合的安全，請使用 MXADMIN 類別而非 MQADMIN 類別。
2. 授權 MQWebUser 角色中的每一位使用者存取使用 MQ Console 和 REST API 所需的系統佇列。

   若要這麼做，請針對這兩個 SYSTEM.ADMIN.COMMAND.QUEUE 和 SYSTEM.REST.REPLY.QUEUE，視是否使用混合大小寫安全而定，為每一個使用者提供 MQQUEUE 或 MXQUEUE 類別的 UPDATE 存取權。

   您需要在使用者將透過「REST API」管理的每個佇列管理程式上執行此動作，包括透過 administrative REST API 閘道管理的遠端佇列管理程式。
3. 若要容許具有 MQWebUser 角色的使用者管理遠端佇列管理程式，請授與使用者對 MQQUEUE 或 MXQUEUE 類別中設定檔的 UPDATE 存取權，以保護用來將指令傳送至遠端佇列管理程式的傳輸佇列。請注意，您需要授與使用者對閘道佇列管理程式的 UPDATE 存取權。

   在遠端佇列管理程式上，授與存取權給相同的使用者，以便將用來傳送指令回應訊息回到閘道佇列管理程式的傳輸佇列。
4. 授與 MQWebUser 角色中的使用者對執行 MQ Console 和 REST API 所支援的作業所需的任何其他資源的存取權。

   需要存取下列項目:

   - 在 REST API 中執行作業，在個別 REST API 資源 的 安全需求 小節中有說明。
   - 第 193 頁的『IBM MQ Console -必要的指令安全設定檔』 中說明了 MQ Console 的發出指令

z/OS 在 z/OS 上為 REST API 和 IBM MQ Console 配置 TLS

在 z/OS 上，您可以將 mqweb 伺服器配置為使用 RACF 金鑰環來儲存憑證，以使用 TLS 及用戶端憑證鑑別進行安全連線。

開始之前

您必須是具備 `mqwebuser.xml` 檔寫入權的使用者，且具備使用 SAF 金鑰環的權限，才能完成這個程序。

關於這項作業

預設 mqweb 伺服器配置會將 Java 金鑰儲存庫用於伺服器及授信憑證。 在 z/OS 上，您可以將 mqweb 伺服器配置為使用 RACF 金鑰環，而非 Java 金鑰儲存庫。 伺服器也可以配置成容許使用者使用用戶端憑證進行鑑別。

如需在 Liberty 中使用 RACF 金鑰環的相關資訊，請參閱 Liberty: 金鑰儲存庫 。

遵循此程序，將 mqweb 伺服器配置為使用 RACF 金鑰環，並選擇性地配置用戶端憑證鑑別。 此程序說明建立及使用以您自己的憑證管理中心 (CA) 憑證簽署之憑證的必要步驟。 對於正式作業，您可能偏好使用從外部憑證管理中心取得的憑證。

程序

1. 建立憑證管理中心 (CA) 憑證，以用來簽署伺服器憑證。 例如，輸入下列 RACF 指令:

```
RACDCERT GENCERT -
  CERTAUTH -
  SUBJECTSDN(CN('mqweb Certification Authority') -
    O('IBM') -
    OU('MQ')) -
  SIZE(2048) -
  WITHLABEL('mqwebCertauth')
```

2. 輸入下列指令，以建立伺服器憑證 (使用步驟 1 中所建立的 CA 憑證簽署):

```
RACDCERT ID(mqwebUserId) GENCERT -
  SUBJECTSDN(CN('hostname') -
   O('IBM') -
   OU('MQ')) -
   SIZE(2048) -
   SIGNWITH (CERTAUTH LABEL('mqwebCertauth')) -
  WITHLABEL('mqwebServerCert')
```

   其中 *mqwebUserId* 是 mqweb 伺服器已啟動作業使用者 ID ，而 *hostname* 是 mqweb 伺服器的主機名稱。

3. 輸入下列指令，將 CA 憑證和伺服器憑證連接至 SAF 金鑰環:

```
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebCertauth') CERTAUTH)
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebServerCert'))
```

   其中 *mqwebUserId* 是 mqweb 伺服器啟動的作業使用者 ID ，而 *keyring* 是您要使用的金鑰環名稱。

4. 輸入下列指令，將 CA 憑證匯出至 CER 檔:

```
RACDCERT CERTAUTH EXPORT(LABEL('mqwebCertauth')) -
        DSN('hlq.CERT.MQWEBCA') -
        FORMAT(CERTDER) -
        PASSWORD('password')
```

5. 以二進位格式將匯出的 CA 憑證以 FTP 傳送至工作站，並將它作為憑證管理中心憑證匯入至瀏覽器。
6. 選擇性的: 如果您想要配置用戶端憑證鑑別，請建立並匯出用戶端憑證。

   a) 建立憑證管理中心 (CA) 憑證，以用來簽署用戶端憑證。 例如，輸入下列 RACF 指令:

```
RACDCERT GENCERT -
  CERTAUTH -
  SUBJECTSDN(CN('mqweb User CA') -
   O('IBM') -
   OU('MQ')) -
   SIZE(2048) -
  WITHLABEL('mqwebUserCertauth')
```

   b) 輸入下列指令，將 CA 憑證連接至 SAF 金鑰環:

```
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebUserCertauth') CERTAUTH)
```

其中 *mqwebUserId* 是 mqweb 伺服器啟動的作業使用者 ID，而 *keyring* 是您要使用的金鑰環名稱。

c) 建立以 CA 憑證簽署的用戶端憑證。 例如，輸入下列指令：

```
RACDCERT ID(clientUserId) GENCERT -
  SUBJECTSDN(CN('clientUserId') -
     O('IBM') -
     OU('MQ')) -
   SIZE(2048) -
   SIGNWITH (CERTAUTH LABEL('mqwebUserCertauth')) -
  WITHLABEL('userCertLabel')
```

其中 *clientUserId* 是使用者名稱。

用來將憑證對映至主體的方法取決於所配置的使用者登錄類型：

- 如果您使用基本登錄，憑證中的「通用名稱」欄位會比對登錄中的使用者。
- 如果您使用 SAF 登錄，且憑證位於 RACF 資料庫中，則會使用在建立憑證時以 **ID** 參數指定的憑證擁有者。
- 如果您使用 LDAP 登錄，憑證中的完整識別名稱會比對 LDAP 登錄。

d) 輸入下列指令，將用戶端憑證匯出至 PKCS #12 檔案：

```
RACDCERT ID(mqwebUserId) EXPORT(LABEL('userCertLabel')) -
         PASSWORD('password') DSN('hlq.USER.CERT')
```

e) 以二進位格式將匯出的憑證 FTP 至工作站。 若要搭配使用用戶端憑證與 IBM MQ Console，請將它匯入用來存取 IBM MQ Console 的 Web 瀏覽器中作為個人憑證。

7. 編輯檔案 *WLP_user_directory*/servers/mqweb/mqwebuser.xml，其中 *WLP_user_directory* 是在執行 **crtmqweb** Script 以建立 mqweb 伺服器定義時指定的目錄。

進行下列變更，以將 mqweb 伺服器配置為使用 RACF 金鑰環：

a) 移除或註銷下列行：

```
<sslDefault sslRef="mqDefaultSSLConfig"/>
```

b) 新增下列陳述式：

```
<keyStore id="defaultKeyStore" filebased="false"
          location="safkeyring://mqwebUserId/keyring"
          password="password" readOnly="true" type="JCERACFKS" />
<ssl id="thisSSLConfig" keyStoreRef="defaultKeyStore" sslProtocol="TLSv1.2"
          serverKeyAlias="mqwebServerCert" clientAuthenticationSupported="true" />
<sslDefault sslRef="thisSSLConfig"/>
```

其中：

- *mqwebUserID* 是 mqweb 伺服器已啟動作業使用者 ID。
- *keyring* 是 RACF 金鑰環的名稱。
- *mqwebServer* 憑證 是 mqweb 伺服器憑證的標籤。

附註: keyStore password 的值會被忽略。

8. 透過停止並重新啟動 mqweb 伺服器已啟動作業來重新啟動 mqweb 伺服器。
9. 選擇性的: 使用用戶端憑證來鑑別：

- 若要搭配使用用戶端憑證與 IBM MQ Console，請在已安裝用戶端憑證的 Web 瀏覽器中輸入 MQ Console 的 URL。
- 若要搭配使用用戶端憑證與 REST API，請為用戶端憑證提供每一個 REST 要求。

附註:

a. 如果您只使用憑證向 IBM MQ Console 進行鑑別，瀏覽器可能會顯示憑證清單供您選取。

b. 如果您想要使用不同的憑證，則可能需要關閉並重新啟動瀏覽器。

c. 如果您使用不在 RACF 資料庫中的用戶端憑證，則可以使用 RACF 憑證名稱過濾，將憑證屬性對映至使用者 ID。 例如：

```
RACDCERT ID(DEPT3USR) MAP SDNFILTER(OU=DEPT1.C=US)
```

將具有主題識別名稱 (包含 OU=DEPT1 及 C=US) 的憑證對映至使用者 ID DEPT3USR。

結果

您已為 IBM MQ Console 和 REST API 設定 TLS 介面。

ALW 在 AIX, Linux, and Windows 上管理金鑰和憑證

在 AIX, Linux, and Windows 上，使用 **runmqckm** 和 **runmqakm** 指令來管理金鑰、憑證和憑證申請。

runmqckm 指令提供與 **iKeyman** 類似的功能， 而 **runmqakm** 指令提供與 **gskitcapicmd** 類似的功能。 在使用 **runmqckm** 或 **runmqakm** 之前，請執行 **setmqenv** 指令，確定已正確配置系統環境變數。

runmqckm 指令需要安裝 IBM MQ JRE 元件。 如果未安裝此元件，您可以改用 **runmqakm** 指令。

如果您需要以符合 FIPS 標準的方式管理 TLS 憑證，請使用 **runmqakm** 指令而非 **runmqckm** 指令。 這是因為 **runmqakm** 指令支援更強的加密。

使用 **runmqckm** 和 **runmqakm** 指令來執行下列動作：

- 建立 IBM MQ 需要的 CMS 金鑰資料庫檔類型
- 建立憑證申請
- 匯入個人憑證
- 匯入 CA 憑證
- 管理自簽憑證

相關資訊

金鑰工具

ALW AIX, Linux, and Windows 上的 runmqckm 及 runmqakm 指令

本節根據指令的物件來說明 **runmqckm** 和 **runmqakm** 指令。

兩個指令之間的主要差異如下：

- **runmqckm**
  - 提供類似於 **iKeycmd** 的功能
  - 支援 JKS 和 JCEKS 金鑰儲存庫檔案格式
- **runmqakm**
  - 提供類似於 **gskitcapicmd** 的功能
  - 支援建立具有「橢圓曲線」公開金鑰的憑證和憑證申請，而 **runmqckm** 指令不支援
  - 支援透過 **-strong** 參數對金鑰儲存庫檔進行比 **runmqckm** 指令更強的加密
  - 已認證為符合 FIPS 140-2 標準，且可以使用 **-fips** 參數配置為以符合 FIPS 標準的方式運作

小心: runmqckm 指令需要安裝 IBM MQ Java runtime environment (JRE) 特性。

每一個指令至少指定一個 物件。 PKCS #11 裝置作業的指令可能會指定其他物件。 金鑰資料庫、憑證及憑證申請物件的指令也會指定 動作。 物件可以是下列其中一項：

-keydb
適用於金鑰資料庫的動作

-憑證
適用於憑證的動作

-certreq
適用於憑證申請的動作

-救命
顯示說明

-version
顯示版本資訊

下列子主題說明您可以對金鑰資料庫、憑證及憑證申請物件採取的動作; 如需這些指令的選項說明，請參閱第 464 頁的『AIX, Linux, and Windows 上的 runmqckm 及 runmqakm 選項』。

ALW 僅在 AIX, Linux, and Windows 上用於 CMS 金鑰資料庫的指令

您可以使用 **runmqckm** 和 **runmqakm** 指令來管理 CMS 金鑰資料庫的金鑰和憑證。

-keydb -changepw
變更 CMS 金鑰資料庫的密碼:

使用 **runmqckm** 指令:

```
-keydb -changepw -db filename -pw password -new_pw new_password
-stash
```

使用 **runmqakm** 指令:

```
-keydb -changepw -db filename -pw password -new_pw new_password
-stash -fips -strong
```

-keydb -create
建立 CMS 金鑰資料庫:

使用 **runmqckm** 指令:

```
-keydb -create -db filename -pw password -type cms -expire days
-stash
```

使用 **runmqakm** 指令:

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

-keydb -stashpw
將 CMS 金鑰資料庫的密碼隱藏在檔案中:

使用 **runmqckm** 指令:

```
-keydb -stashpw -db filename -pw password
```

使用 **runmqakm** 指令:

```
-keydb -stashpw -db filename -pw password -fips
```

-cert -getdefault

註: IBM MQ 8.0 不支援預設憑證。 您應該使用憑證標籤配置，如 第 22 頁的『數位憑證標籤，瞭解需求』中所述。

取得預設個人憑證:

使用 **runmqckm** 指令:

```
-cert -getdefault -db filename -pw password
```

使用 **runmqakm** 指令:

```
-cert -getdefault -db filename -pw password -fips
```

-cert-modify
修改憑證。

註: 目前，唯一可以修改的欄位是「憑證信任」欄位。

使用 **runmqckm** 指令:

```
-cert -modify -db filename -pw password -label label
-trust enable|disable
```

使用 **runmqakm** 指令:

```
-cert -modify -db filename -pw password -label label
-trust enable|disable -fips
```

-cert -setdefault

註: IBM MQ 8.0 或更新版本不支援預設憑證。 您應該使用憑證標籤配置，如 第 22 頁的『數位憑證標籤，瞭解需求』中所述。

ALW AIX, Linux, and Windows 上 CMS 或 PKCS #12 金鑰資料庫的指令

使用 **runmqckm** 和 **runmqakm** 指令來管理 CMS 金鑰資料庫或 PKCS #12 金鑰資料庫的金鑰和憑證。

註: IBM MQ 不支援 SHA-3 或 SHA-5 演算法。 您可以使用數位簽章演算法名稱 SHA384WithRSA 及 SHA512WithRSA ，因為這兩個演算法都是 SHA-2 系列的成員。

數位簽章演算法名稱 SHA3WithRSA 和 SHA5WithRSA 已淘汰，因為它們分別是 SHA384WithRSA 和 SHA512WithRSA 縮寫形式。

-keydb -changepw
變更金鑰資料庫的密碼:

使用 **runmqckm** 指令:

```
-keydb -changepw -db filename -pw password -new_pw new_password -expire days
```

使用 **runmqakm** 指令:

```
-keydb -changepw -db filename -pw password -new_pw new_password -expire days
-fips -strong
```

-keydb -convert
對於 **runmqckm** 指令，將金鑰資料庫從一種格式轉換為另一種格式:

```
-keydb -convert -db filename -pw password
-old_format cms | pkcs12 -new_format cms
```

使用 **runmqakm** 指令，將舊版 CMS 金鑰資料庫轉換為新版 CMS 金鑰資料庫:

```
-keydb -convert -db filename -pw password
-new_db filename -new_pw password -strong -fips
```

-keydb -create
建立金鑰資料庫:

使用 **runmqckm** 指令:

```
-keydb -create -db filename -pw password -type cms
| pkcs12
```

使用 **runmqakm** 指令:

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

-keydb -delete
刪除金鑰資料庫:

使用任一指令:

```
-keydb -delete -db filename -pw password
```

-keydb -list
列出目前支援的金鑰資料庫類型:

使用 **runmqckm** 指令:

```
-keydb -list
```

使用 **runmqakm** 指令:

```
-keydb -list -fips
```

-cert -add
將憑證從檔案新增至金鑰資料庫:

使用 **runmqckm** 指令:

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

使用 **runmqakm** 指令:

```
-cert -add -db filename -pw password -label label -file filename
-format ascii | binary -fips
```

-cert -create
建立自簽憑證:

使用 **runmqckm** 指令:

```
-cert -create -db filename -pw password -label label
-dn distinguished_name -size 1024 | 512 -x509version 3 | 1 | 2
-expire days -sig_alg MD2_WITH_RSA | MD2WithRSA |
MD5_WITH_RSA | MD5WithRSA |
SHA1WithDSA | SHA1WithRSA |
SHA256_WITH_RSA | SHA256WithRSA |
SHA2WithRSA | SHA384_WITH_RSA |
SHA384WithRSA | SHA512_WITH_RSA |
SHA512WithRSA | SHA_WITH_DSA |
SHA_WITH_RSA | SHAWithDSA |
SHAWithRSA
```

使用 **runmqakm** 指令:

```
-cert -create -db filename -pw password -label label
-dn distinguished_name -size 2048 | 1024 | 512 -x509version 3 | 1 | 2
-expire days -fips -sig_alg md5 |
MD5_WITH_RSA | 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 |
EC_ecdsa_with_SHA512
```

-cert -delete
刪除憑證:

使用 **runmqckm** 指令:

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

使用 **runmqakm** 指令:

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

-cert -details
列出特定憑證的詳細資訊:

使用 **runmqckm** 指令:

```
-cert -details -db filename -pw password -label label
```

使用 **runmqakm** 指令:

```
-cert -details -db filename -pw password -label label -fips
```

-cert -export
將個人憑證及其相關聯的私密金鑰從金鑰資料庫匯出至 PKCS #12 檔案，或匯出至另一個金鑰資料庫:

使用 **runmqckm** 指令:

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

使用 **runmqakm** 指令:

```
-cert -export -db filename -pw password -label label -type cms | pkcs12
-target filename -target_pw password -target_type  cms | pkcs12
-encryption strong | weak -fips
```

-cert -extract
從金鑰資料庫擷取憑證:

使用 **runmqckm** 指令:

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

使用 **runmqakm** 指令:

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

-cert -import
從金鑰資料庫匯入個人憑證:

使用 **runmqckm** 指令:

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

使用 **runmqakm** 指令:

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

對於這兩個指令:

- 需要 -label 選項，並指定要從來源金鑰資料庫匯入之憑證的標籤。
- 此外，您可以使用 -new_label 選項。 這可讓匯入的憑證在目標金鑰資料庫中獲得與來源資料庫中的標籤不同的標籤。

-cert -list
列出金鑰資料庫中的所有憑證:

使用 **runmqckm** 指令:

```
-cert -list all | personal | CA -db filename -pw password
```

使用 **runmqakm** 指令:

```
-cert -list all | personal | CA -db filename -pw password -fips
```

-cert -receive
從檔案接收憑證:

使用 **runmqckm** 指令:

```
-cert -receive -file filename -db filename -pw password
-format ascii | binary -default_cert yes | no
```

使用 **runmqakm** 指令:

```
-cert -receive -file filename -db filename -pw password
-format ascii | binary -default_cert yes | no -fips
```

-cert -sign
簽署憑證:

使用 **runmqckm** 指令:

```
-cert -sign -db filename -file filename -pw password
-label label -target filename -format ascii | binary -expire days
-sig_alg MD2_WITH_RSA | MD2WithRSA | MD5_WITH_RSA |
MD5WithRSA | SHA1WithDSA | SHA1WithRSA |
SHA256_WITH_RSA | SHA256WithRSA |
SHA2WithRSA | SHA384_WITH_RSA |
SHA384WithRSA | SHA512_WITH_RSA |
SHA512WithRSA | SHA_WITH_DSA |
SHA_WITH_RSA | SHAWithDSA |
SHAWithRSA
```

使用 **runmqakm** 指令:

```
-cert -sign -db filename -file filename -pw password
-label label -target filename -format ascii | binary -expire days -fips
-sig_alg md5 | MD5_WITH_RSA | 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 |
EC_ecdsa_with_SHA512
```

-certreq -create

建立憑證申請:

使用 **runmqckm** 指令:

```
-certreq -create -db filename -pw password -label label -dn distinguished_name
-size 1024 | 512 -file filename
-sig_alg MD2_WITH_RSA | MD2WithRSA |
MD5_WITH_RSA | MD5WithRSA |
SHA1WithDSA | SHA1WithRSA |
SHA256_WITH_RSA | SHA256WithRSA |
SHA2WithRSA | SHA384_WITH_RSA |
SHA384WithRSA | SHA512_WITH_RSA |
SHA512WithRSA | SHA_WITH_DSA |
SHA_WITH_RSA | SHAWithDSA |
SHAWithRSA
```

使用 **runmqakm** 指令:

```
-certreq -create -db filename -pw password -label label -dn distinguished_name
-size 2048 | 1024 | 512 -file filename -fips
-sig_alg md5 | MD5_WITH_RSA | 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 |
EC_ecdsa_with_SHA512
```

-certreq -delete

刪除憑證申請:

使用 **runmqckm** 指令:

```
-certreq -delete -db filename -pw password -label  label
```

使用 **runmqakm** 指令:

```
-certreq -delete -db filename -pw password -label  label -fips
```

-certreq -details

列出特定憑證申請的詳細資訊:

使用 **runmqckm** 指令:

```
-certreq -details -db filename -pw password -label  label
```

使用 **runmqakm** 指令:

```
-certreq -details -db filename -pw password -label  label -fips
```

列出憑證申請的詳細資訊，並顯示完整憑證申請:

使用 **runmqckm** 指令:

```
-certreq -details -showOID -db filename -pw password -label label
```

使用 **runmqakm** 指令:

```
-certreq -details -showOID -db filename -pw password -label label -fips
```

-certreq -extract
將憑證申請資料庫中的憑證申請擷取至檔案:

對於 **runmqckm** 指令:

```
-certreq -extract -db filename -pw password -label label -target filename
```

使用 **runmqakm** 指令:

```
-certreq -extract -db filename -pw password -label label -target filename -fips
```

-certreq -list
列出憑證申請資料庫中的所有憑證申請:

使用 **runmqckm** 指令:

```
-certreq -list -db filename -pw password
```

使用 **runmqakm** 指令:

```
-certreq -list -db filename -pw password -fips
```

-certreq -recreate
重建憑證申請:

使用 **runmqckm** 指令:

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

使用 **runmqakm** 指令:

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

ALW AIX, Linux, and Windows 上用於加密裝置作業的指令

您可以使用 **runmqckm** (iKeycmd) 和 **runmqakm** 指令來管理加密裝置作業的金鑰和憑證。

註: IBM MQ 不支援 SHA-3 或 SHA-5 演算法。 您可以使用數位簽章演算法名稱 SHA384WithRSA 及 SHA512WithRSA ，因為這兩個演算法都是 SHA-2 系列的成員。

數位簽章演算法名稱 SHA3WithRSA 和 SHA5WithRSA 已淘汰，因為它們分別是 SHA384WithRSA 和 SHA512WithRSA 縮寫形式。

-keydb -changepw
變更加密裝置的密碼:

使用 **runmqckm** 指令:

```
-keydb -changepw -crypto module_name -tokenlabel token_label
-pw password -new_pw new_password
```

如果使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 **runmqckm** 及 **strmqikm** 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 僅 Windows 及 Linux x86 32 位元平台例外，因為在這些平台，**strmqikm** 和 **runmqckm** 程式都是 32 位元的。

使用 **runmqakm** 指令:

```
-keydb -changepw -db filename -crypto module_name -tokenlabel  token_label
-pw password -new_pw new_password -fips -strong
```

-keydb -list
列出目前支援的金鑰資料庫類型:

使用 **runmqckm** 指令:

```
-keydb -list
```

如果使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 **runmqckm** 及 **strmqikm** 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 僅 Windows 及 Linux x86 32 位元平台例外，因為在這些平台，**strmqikm** 和 **runmqckm** 程式都是 32 位元的。

使用 **runmqakm** 指令:

```
-keydb -list -fips
```

-cert -add

將憑證從檔案新增至加密裝置:

使用 **runmqckm** 指令:

```
-cert -add -crypto module_name -tokenlabel token_label -pw password
-label label -file filename -format ascii | binary
```

如果使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 **runmqckm** 及 **strmqikm** 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 僅 Windows 及 Linux x86 32 位元平台例外，因為在這些平台，**strmqikm** 和 **runmqckm** 程式都是 32 位元的。

使用 **runmqakm** 指令:

```
-cert -add -crypto module_name -tokenlabel token_label -pw password
-label label -file filename -format ascii | binary -fips
```

-cert -create

在加密裝置上建立自簽憑證:

使用 **runmqckm** 指令:

```
-cert -create -crypto module_name -tokenlabel token_label
-pw password -label label -dn distinguished_name
-size 1024 | 512 -x509version 3 | 1 | 2
-default_cert no | yes -expire days
-sig_alg MD2_WITH_RSA | MD2WithRSA |
MD5_WITH_RSA | MD5WithRSA |
SHA1WithDSA | SHA1WithRSA |
SHA256_WITH_RSA | SHA256WithRSA |
SHA2WithRSA | SHA384_WITH_RSA |
SHA384WithRSA | SHA512_WITH_RSA |
SHA512WithRSA | SHA_WITH_DSA |
SHA_WITH_RSA | SHAWithDSA |
SHAWithRSA
```

如果使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 **runmqckm** 及 **strmqikm** 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 僅 Windows 及 Linux x86 32 位元平台例外，因為在這些平台，**strmqikm** 和 **runmqckm** 程式都是 32 位元的。

使用 **runmqakm** 指令:

```
-cert -create -crypto module_name -tokenlabel token_label
-pw password -label label -dn distinguished_name
-size 2048 | 1024 | 512 -x509version 3 | 1 | 2
-default_cert no | yes -expire days
-fips -sig_alg md5 | MD5_WITH_RSA | 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 |
EC_ecdsa_with_SHA512
```

-cert -delete
刪除加密裝置上的憑證:

使用 **runmqckm** 指令:

```
-cert -delete -crypto module_name -tokenlabel token_label -pw password -label label
```

如果使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 **runmqckm** 及 **strmqikm** 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 僅 Windows 及 Linux x86 32 位元平台例外，因為在這些平台，**strmqikm** 和 **runmqckm** 程式都是 32 位元的。

使用 **runmqakm** 指令:

```
-cert -delete -crypto module_name -tokenlabel token_label -pw password -label label -fips
```

-cert -details
列出加密裝置上特定憑證的詳細資訊:

使用 **runmqckm** 指令:

```
-cert -details -crypto module_name -tokenlabel token_label
-pw password -label label
```

如果使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 **runmqckm** 及 **strmqikm** 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 僅 Windows 及 Linux x86 32 位元平台例外，因為在這些平台，**strmqikm** 和 **runmqckm** 程式都是 32 位元的。

使用 **runmqakm** 指令:

```
-cert -details -crypto module_name -tokenlabel token_label
-pw password -label label -fips
```

列出詳細資訊，並顯示加密裝置上特定憑證的完整憑證:

使用 **runmqckm** 指令:

```
-cert -details -showOID -crypto module_name -tokenlabel  token_label
-pw password -label label
```

如果使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 **runmqckm** 及 **strmqikm** 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 僅 Windows 及 Linux x86 32 位元平台例外，因為在這些平台，**strmqikm** 和 **runmqckm** 程式都是 32 位元的。

使用 **runmqakm** 指令:

```
-cert -details -showOID -crypto module_name -tokenlabel  token_label
-pw password -label label -fips
```

-cert -extract
從金鑰資料庫擷取憑證:

使用 **runmqckm** 指令:

```
-cert -extract -crypto module_name -tokenlabel token_label -pw password
-label label -target filename -format ascii | binary
```

如果使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 **runmqckm** 及 **strmqikm** 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 僅 Windows 及 Linux x86 32 位元平台例外，因為在這些平台，**strmqikm** 和 **runmqckm** 程式都是 32 位元的。

使用 **runmqakm** 指令:

```
-cert -extract -crypto module_name -tokenlabel token_label -pw password
-label label -target filename -format ascii | binary -fips
```

-cert -import

將憑證匯入至具有次要金鑰資料庫支援的加密裝置:

使用 **runmqckm** 指令:

```
-cert -import -db filename -pw password -label label -type cms
-crypto module_name -tokenlabel token_label -pw  password
-secondaryDB filename -secondaryDBpw password
```

如果使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 **runmqckm** 及 **strmqikm** 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 僅 Windows 及 Linux x86 32 位元平台例外，因為在這些平台，**strmqikm** 和 **runmqckm** 程式都是 32 位元的。

使用 **runmqakm** 指令:

```
-cert -import -db filename -pw password -label label -type cms
-crypto module_name -tokenlabel token_label -pw  password
-secondaryDB filename -secondaryDBpw password -fips
```

將 PKCS #12 憑證匯入至具有次要金鑰資料庫支援的加密裝置:

使用 **runmqckm** 指令:

```
-cert -import -file filename -pw password -type pkcs12
-crypto module_name -tokenlabel token_label -pw  password
-secondaryDB filename -secondaryDBpw password
```

如果使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 **runmqckm** 及 **strmqikm** 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 僅 Windows 及 Linux x86 32 位元平台例外，因為在這些平台，**strmqikm** 和 **runmqckm** 程式都是 32 位元的。

使用 **runmqakm** 指令:

```
-cert -import -file filename -pw password -type pkcs12
-crypto module_name -tokenlabel token_label -pw  password
-secondaryDB filename -secondaryDBpw password -fips
```

-cert -list

列出加密裝置上的所有憑證:

使用 **runmqckm** 指令:

```
-cert -list all | personal | CA -crypto module_name
-tokenlabel token_label -pw  password
```

如果使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 **runmqckm** 及 **strmqikm** 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 僅 Windows 及 Linux x86 32 位元平台例外，因為在這些平台，**strmqikm** 和 **runmqckm** 程式都是 32 位元的。

使用 **runmqakm** 指令:

```
-cert -list all | personal | CA -crypto module_name
-tokenlabel token_label -pw  password -fips
```

-cert -receive
從檔案接收憑證至具有次要金鑰資料庫支援的加密裝置:

使用 **runmqckm** 指令:

```
-cert -receive -file filename -crypto module_name -tokenlabel  token_label
-pw password -default_cert yes | no -secondaryDB filename
-secondaryDBpw password -format  ascii | binary
```

如果使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 **runmqckm** 及 **strmqikm** 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 僅 Windows 及 Linux x86 32 位元平台例外，因為在這些平台，**strmqikm** 和 **runmqckm** 程式都是 32 位元的。

使用 **runmqakm** 指令:

```
-cert -receive -file filename -crypto module_name -tokenlabel  token_label
-pw password -default_cert yes | no -secondaryDB filename
-secondaryDBpw password -format  ascii | binary -fips
```

-certreq -create
在加密裝置上建立憑證申請:

使用 **runmqckm** 指令:

```
-certreq -create -crypto module_name -tokenlabel token_label
-pw password -label label -dn distinguished_name
-size 1024 | 512 -file filename
-sig_alg MD2_WITH_RSA | MD2WithRSA | MD5_WITH_RSA |
MD5WithRSA | SHA1WithDSA | SHA1WithRSA |
SHA256_WITH_RSA | SHA256WithRSA |
SHA2WithRSA | SHA384_WITH_RSA |
SHA384WithRSA | SHA512_WITH_RSA |
SHA512WithRSA | SHA_WITH_DSA |
SHA_WITH_RSA | SHAWithDSA |
SHAWithRSA
```

如果使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 **runmqckm** 及 **strmqikm** 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 僅 Windows 及 Linux x86 32 位元平台例外，因為在這些平台，**strmqikm** 和 **runmqckm** 程式都是 32 位元的。

使用 **runmqakm** 指令:

```
-certreq -create -crypto module_name -tokenlabel token_label
-pw password -label label -dn distinguished_name
-size 2048 | 1024 | 512 -file filename -fips
-sig_alg md5 | MD5_WITH_RSA | SHA_WITH_DSA |
SHA_WITH_RA | 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 |
EC_ecdsa_with_SHA512
```

-certreq -delete
從加密裝置刪除憑證申請:

使用 **runmqckm** 指令:

```
-certreq -delete -crypto module_name -tokenlabel token_label
-pw password -label label
```

如果使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 **runmqckm** 及 **strmqikm** 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 僅 Windows 及 Linux x86 32 位元平台例外，因為在這些平台，**strmqikm** 和 **runmqckm** 程式都是 32 位元的。

使用 **runmqakm** 指令:

```
-certreq -delete -crypto module_name -tokenlabel token_label
-pw password -label label -fips
```

-certreq -details
列出加密裝置上特定憑證申請的詳細資訊:

使用 **runmqckm** 指令:

```
-certreq -details -crypto module_name -tokenlabel token_label
-pw password -label label
```

如果使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 **runmqckm** 及 **strmqikm** 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 僅 Windows 及 Linux x86 32 位元平台例外，因為在這些平台，**strmqikm** 和 **runmqckm** 程式都是 32 位元的。

使用 **runmqakm** 指令:

```
-certreq -details -crypto module_name -tokenlabel token_label
-pw password -label label -fips
```

列出憑證申請的詳細資訊，並在加密裝置上顯示完整憑證申請:

使用 **runmqckm** 指令:

```
-certreq -details -showOID -crypto module_name -tokenlabel  token_label
-pw password -label label
```

如果使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 **runmqckm** 及 **strmqikm** 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 僅 Windows 及 Linux x86 32 位元平台例外，因為在這些平台，**strmqikm** 和 **runmqckm** 程式都是 32 位元的。

使用 **runmqakm** 指令:

```
-certreq -details -showOID -crypto module_name -tokenlabel  token_label
-pw password -label label -fips
```

-certreq -extract
從加密裝置上的憑證申請資料庫中擷取憑證申請至檔案:

使用 **runmqckm** 指令:

```
-certreq -extract -crypto module_name -tokenlabel token_label
-pw password -label label -target filename
```

如果使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 **runmqckm** 及 **strmqikm** 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 僅 Windows 及 Linux x86 32 位元平台例外，因為在這些平台，**strmqikm** 和 **runmqckm** 程式都是 32 位元的。

使用 **runmqakm** 指令:

```
-certreq -extract -crypto module_name -tokenlabel token_label
-pw password -label label -target filename -fips
```

-certreq -list
列出加密裝置上憑證申請資料庫中的所有憑證申請:

使用 **runmqckm** 指令:

```
-certreq -list -crypto module_name -tokenlabel token_label
-pw password
```

如果使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 **runmqckm** 及 **strmqikm** 是 64 位元程式。 PKCS #11 支援所需要的外部模組將載入到 64 位元程序中，因此您必須安裝 64 位元 PKCS #11 程式庫來管理加密硬體。 僅 Windows 及 Linux x86 32 位元平台例外，因為在這些平台，**strmqikm** 和 **runmqckm** 程式都是 32 位元的。

使用 **runmqakm** 指令:

```
-certreq -list -crypto module_name -tokenlabel token_label
-pw password -fips
```

ALW AIX, Linux, and Windows 上的 runmqckm 及 runmqakm 選項

您可以使用 **runmqckm** 及 **runmqakm** 指令行選項來管理金鑰、憑證及憑證申請。 **runmqckm** 提供類似於 **iKeycmd** 的功能，而 **runmqakm** 提供類似於 **gskitcapicmd** 的功能。

註: IBM MQ 不支援 SHA-3 或 SHA-5 演算法。 您可以使用數位簽章演算法名稱 SHA384WithRSA 及 SHA512WithRSA ，因為這兩個演算法都是 SHA-2 系列的成員。

數位簽章演算法名稱 SHA3WithRSA 和 SHA5WithRSA 已淘汰，因為它們分別是 SHA384WithRSA 和 SHA512WithRSA 縮寫形式。

選項的意義視指令中指定的物件和動作而定。

表 *94:* 可與 ***runmqckm*** 和 ***runmqakm*** 搭配使用的選項

參數	**說明**
-create	用來建立金鑰資料庫的選項。
-crypto	用來管理 PKCS #11 加密裝置的模組名稱。 如果您在內容檔中指定模組名稱，則 **-crypto** 之後的值是選用的。 如果您使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意， **runmqckm** 和 **strmqikm** 是使用 IBM MQ 安裝所提供的 Java 虛擬機器 (JVM) 來執行。 PKCS #11 支援所需的外部模組將載入 JVM 程序中，因此您必須安裝 PKCS #11 程式庫，以管理符合 JVM 位元的加密硬體，且必須將此程式庫指定為 **runmqckm** 或 **strmqikm**。
-db	金鑰資料庫的完整路徑名稱。
-default_cert	將憑證設為預設憑證。 值可以是 yes 或 no。 預設值為 NO。
-dn	X.500 識別名稱。 該值是以雙引號括住的字串，例如 "CN=John Smith,O=IBM,OU=Test,C=GB"。 請注意，只需要 O 和 C 屬性。 指定通用名稱 (CN) 是選用的。
-encryption	憑證匯出指令中使用的加密強度。 值可以是 **強** 或 **弱**。 預設值是 strong。
-expire	憑證或資料庫密碼的有效期限 (天)。 憑證密碼的預設值為 365 天。 資料庫密碼沒有預設時間: 請使用 **-expire** 參數來明確設定資料庫密碼有效期限。
-file	憑證或憑證申請的檔名。
-fips	指定以 FIPS 模式執行指令。 處於 FIPS 模式時， IBM Crypto for C (ICC) 元件會使用已驗證 FIPS 140-2 的演算法。 如果 ICC 元件未在 FIPS 模式下起始設定，則 **runmqakm** 指令會失敗。
-format	憑證的格式。 該值可以是 ascii (代表 Base64_encoded ASCII) 或 binary (代表二進位 DER 資料)。 預設值為 ascii。

表 94: 可與 ***runmqckm*** 和 ***runmqakm*** 搭配使用的選項 (繼續)	
參數	**說明**
-label	附加至憑證或憑證申請的標籤。 如果憑證是用來識別 IBM MQ 用戶端應用程式或佇列管理程式的個人憑證，則標籤必須對應於 IBM MQ 憑證標籤 (CERTLABL) 設定，如需相關資訊，請參閱 第 22 頁的『數位憑證標籤，瞭解需求』。
-new_format	金鑰資料庫的新格式。
-new_label	在憑證匯入指令上使用時，此選項容許使用與來源金鑰資料庫中的標籤不同的標籤來匯入憑證。 如果憑證是用來識別 IBM MQ 用戶端應用程式或佇列管理程式的個人憑證，則標籤必須對應於 IBM MQ 憑證標籤 (CERTLABL) 設定，如需相關資訊，請參閱 第 22 頁的『數位憑證標籤，瞭解需求』。
-new_pw	新的資料庫密碼。
-old_format	金鑰資料庫的舊格式。
-pw	金鑰資料庫或 PKCS #12 檔案的密碼。
-secondaryDB	PKCS #11 裝置作業的次要金鑰資料庫名稱。
-secondaryDBpw	PKCS #11 裝置作業的次要金鑰資料庫密碼。
-showOID	顯示完整憑證或憑證申請。
-sig_alg	在建立憑證申請、自簽憑證或簽署憑證期間使用的雜湊演算法。 此雜湊演算法用來建立與新建憑證或憑證申請相關聯的簽章。 對於 **runmqckm** SHA384_WITH_RSA 該值可以是 MD2_WITH_RSA、 MD2WithRSA、 MD5_WITH_RSA、 MD5WithRSA、 SHA1WithDSA、 SHA1WithECDSA SHA3WithECDSA SHA1WithRSA、 SHA2/ECDSA、 SHA224WithECDSA SHA256WithECDSA SHA2WithECDSA、 SHA3/ECDSA、 SHA256_WITH_RSA SHA256WithRSA、 SHA384WithECDSA、 SHA384WithRSA、 SHA5/ECDSA、 SHA512_WITH_RSA、 SHA512WithECDSA、 SHA512WithRSA、 SHA5WithECDSA、 SHA_WITH_DSA、 SHA_WITH_RSA、 SHAWithDSA、 SHAWithRSA 對於 **runmqakm**，該值可以是 md5、 MD5_WITH_RSA、 MD5WithRSA、 sha1、 SHA_WITH_RSA、 SHAWithRSA SHA384_WITH_RSA SHA1WithRSA sha512 sha224、 SHA224_WITH_RSA、 SHA224WithRSA、 25656 sha384 sha256 SHA256_WITH_RSA、 SHA256WithRSA、 SHA3_256WithRSA SHA3_384WithRSA SHA384WithRSA sha3_384 sha3_256 , SHA512_WITH_RSA , SHA512WithRSA , sha3_512 , SHA3_512WithRSA , RSASSAPSS , RSASSAPSSPSS , SHA224_WITH_RSASSAPSS , SHA256_WITH_RSASSAPSS SHA3_512WithRSASSAPSS SHA512_WITH_RSASSAPSS SHA224WithRSASSAPSS SHA256WithRSASSAPSS SHA512_WITH_RSASSAPSS SHA384WithRSASSAPSS , SHA384_WITH_RSASSAPSS , SHA256WithDSA , 212W745 SHA1WithDSA SHA3_384WithRSASSAPSS SHA512WithRSASSAPSS SHA3_256WithRSASSAPSS , SHA1WithECDSA , EC_ecdsa_with_SHA1 , SHA224WithECDSA , EC_ecdsa_with_SHA224 , SHA256WithECDSA、 EC_ecdsa_with_SHA256、 SHA384WithECDSA、 EC_ecdsa_with_SHA384、 SHA512WithECDSA、 SHA3_512WithECDSA、 SHA3_256WithECDSA、 EC_ecdsa_with_SHA512、 SHA3_384WithECDSA 3834DS 234DS 234DS 4534DS 4534DS 4530000 DH、 Kyber ,**雙鋰**, SHA256WithDilithium 帶雙鋰 , SHA384WithDilithium 雙鋰 , SHA512WithDilithium

表 94: 可與 ***runmqckm*** 和 ***runmqakm*** 搭配使用的選項 (繼續)	
參數	**說明**
-size	金鑰大小。 對於 **runmqckm**，此值可以是 512、1024 或 2048。 預設值為 1024 位元。 對於 **runmqakm**，值取決於簽章演算法: • 對於 RSA 簽章演算法 (如果未指定 **-sig_alg**，則使用預設演算法)，值可以是 512、1024、2048 或 4096。 如果啟用 **-fips** 參數，則不允許 512 位元的 RSA 金鑰大小。 預設 RSA 金鑰大小為 2048 位元。 • 對於「橢圓曲線」演算法，該值可以是 256、384 或 512。 預設「橢圓曲線」金鑰大小取決於簽章演算法。 若為 SHA256，它是 256; 若為 SHA384，它是 384; 若為 SHA512，它是 512。
-stash	將金鑰資料庫密碼隱藏至檔案。 僅適用於 CMS 和 PKCS12 類型的資料庫。 **註: -stash** 在 **-keydb** -create 指令上有效，可讓 **runmqckm**/**runmqakm** 建立包含密碼的隱藏檔。 發出指令 `$ runmqakm -help` 只會列出高階說明參數。
-stashed	指出金鑰資料庫或 PKCS #12 檔案的密碼位於隱藏檔中。 **註: -stashed** 選項適用於 **-keydb** -create 指令以外的呼叫。 如果未指定此選項，則必須使用 **-pw** 來提供密碼。 此外，只有在您指示您正在執行的動作類型時，才會出現顯示 **-stashed** 的詳細說明。
-target	目的地檔案或資料庫。
-target_pw	如果 **-target** 指定金鑰資料庫，則為金鑰資料庫的密碼。
-target_type	**-target** 運算元指定的資料庫類型。 請參閱 **-type** 參數，以取得允許的值。
-tokenLabel	PKCS #11 加密裝置的標籤。
-trust	CA 憑證的信任狀態。 值可以是 `enable` 或 `disable`。 預設值為 `enable`。
-type	資料庫的類型。 此值可以是下列任一值: • `cms`，適用於 CMS 金鑰資料庫 • `pkcs12` 代表 PKCS #12 檔案。
-x509version	要建立的 X.509 憑證版本。 值可以是 1、2 或 3。 預設是 3。

-rfc3339	使用此參數，可針對 `runmqakm -cert -details` 指令以 RFC 3339 格式輸出日期，格式如下:
	`Not Before : 2015-08-26T08:53:37Z`
	`Not After : 2016-08-26T08:53:37Z`
	請注意， **-rfc3339** 參數必須出現在指令中其他參數之後:
	`runmqakm -cert -details -db exampleDB -stashed -label certficateLabel -rfc3339`

ALW AIX, Linux, and Windows 上的 runmqakm 錯誤碼

runmqakm 發出的數值錯誤碼及其意義的表格。

錯誤碼	錯誤訊息
0	成功
1	發生不明錯誤
2	發生 ASN.1 編碼/解碼錯誤。
3	起始設定 ASN.1 編碼器/解碼器時發生錯誤。
4	發生 ASN.1 編碼/解碼錯誤，因為索引超出範圍或選用欄位不存在。
5	發生資料庫錯誤。
6	開啟資料庫檔案時發生錯誤，請檢查檔案是否存在及許可權。
7	重新開啟資料庫檔案時發生錯誤。
8	資料庫建立失敗。
9	資料庫已存在。
10	刪除資料庫檔案時發生錯誤。
11	無法開啟資料庫。
12	讀取資料庫檔案時發生錯誤。
13	將資料寫入資料庫檔案時發生錯誤。
14	發生資料庫驗證錯誤。
15	發現無效的資料庫版本。
16	發現無效的資料庫密碼。
17	發現無效的資料庫檔案類型。
18	指定的資料庫已毀損。
19	提供的密碼無效，或金鑰資料庫已遭竄改或毀損。
20	發生資料庫金鑰項目完整性錯誤。
21	資料庫中已存在重複的憑證。
22	資料庫中已存在重複的索引鍵 (記錄 ID)。
23	金鑰資料庫中已存在具有相同標籤的憑證。
24	資料庫中已存在重複的金鑰 (簽章)。
25 GB	資料庫中已存在重複的金鑰 (未簽署憑證)。
26	資料庫中已存在重複的索引鍵 (發證者和序號)。
27	資料庫中已存在重複的金鑰 (主體公開金鑰資訊)。
28	資料庫中已存在重複的索引鍵 (未簽署 CRL)。
29	資料庫中已使用標籤。
30	發生密碼加密錯誤。
31	發生 LDAP 相關錯誤。 (此程式不支援 LDAP)
32	發生加密錯誤。
33	發生加密/解密錯誤。

錯誤碼	錯誤訊息
34	發現無效的加密演算法。
成功的專案比其他專案多出 35	簽署資料時發生錯誤。
36	驗證資料時發生錯誤。
37	計算資料摘要時發生錯誤。
38	發現無效的加密參數。
39	發現不受支援的加密演算法。
40	指定的輸入大小大於支援的模數大小。
41	找到不受支援的模數大小。
42	發生資料庫驗證錯誤。
43	金鑰項目驗證失敗。
44	存在重複的延伸欄位。
45	金鑰的版本錯誤。
46	必要的副檔名欄位不存在。
47	有效期間不包括今天或不在其發行者的有效期間內
48	有效期間不包括今天或不在其發行者的有效期間內。
49	驗證私密金鑰用法延伸時發生錯誤。
50	找不到金鑰的發證者。
51	遺漏必要的憑證延伸。
52	發現無效的基本限制延伸。
53	金鑰簽章驗證失敗。
54	金鑰的根金鑰不受信任。
55	已撤銷金鑰。
56	驗證權限金鑰 ID 延伸時發生錯誤。
57	驗證私密金鑰用法延伸時發生錯誤。
58	驗證主體替代副檔名時發生錯誤。
59	驗證發證者替代副檔名時發生錯誤。
60	驗證金鑰用法延伸時發生錯誤。
61	找到不明重要延伸。
62	驗證金鑰組項目時發生錯誤。
63	驗證 CRL 時發生錯誤。
64	發生互斥錯誤。
65	找到無效的參數。
66	發現空值參數或記憶體配置錯誤。
67	數字或大小太大或太小。
68	舊密碼無效。

錯誤碼	錯誤訊息
69	新密碼無效。
70	密碼已過期。
71	發生執行緒相關錯誤。
72	建立執行緒時發生錯誤。
73	執行緒等待結束時發生錯誤。
74	發生 I/O 錯誤。
75	載入 CMS 時發生錯誤。
76	發生加密法硬體相關錯誤。
77	未順利呼叫檔案庫起始設定常式。
78	內部資料庫控點表格已毀損。
79	記憶體配置有錯。
80	找到無法辨識的選項。
81	取得時間資訊時發生錯誤。
82	發生互斥建立錯誤。
83	開啟訊息型錄時發生錯誤。
84	開啟錯誤訊息型錄時發生錯誤
85 個	找到空值檔名。
86	開啟檔案時發生錯誤，請檢查檔案是否存在及許可權。
87	開啟要讀取的檔案時發生錯誤。
88	開啟要寫入的檔案時發生錯誤。
89	沒有這類檔案。
90	無法開啟檔案，因為其許可權設定。
91	將資料寫入檔案時發生錯誤。
92	刪除檔案時發生錯誤。
93	找到無效的 Base64-encoded 資料。
94	找到無效的 Base64 訊息類型。
95	使用 Base64 編碼規則編碼資料時發生錯誤。
96	解碼 Base64-encoded 資料時發生錯誤。
97	取得識別名稱標籤時發生錯誤。
98	必要的通用名稱欄位是空的。
99	必要的國家或地區名稱欄位是空的。
100	找到無效的資料庫控點。
101	金鑰資料庫不存在。
102	要求金鑰組資料庫不存在。
103	密碼檔不存在。

錯誤碼	錯誤訊息
104	新密碼與舊密碼相同。
105	在金鑰資料庫中找不到金鑰。
106	找不到要求金鑰。
107	找不到授信 CA。
108	找不到憑證的要求金鑰。
109	金鑰資料庫中沒有私密金鑰。
110	金鑰資料庫中沒有預設金鑰。
111	金鑰記錄中沒有私密金鑰。
112	金鑰記錄中沒有憑證。
113	沒有 CRL 項目。
114	發現無效的金鑰資料庫檔名。
115	發現無法辨識的私密金鑰類型。
116	發現無效的識別名稱輸入。
117	找不到具有指定金鑰標籤的金鑰項目。
118	金鑰標籤清單已毀損。
119	輸入資料不是有效的 PKCS12 資料。
120	密碼無效，或 PKCS12 資料已毀損或以更新版本的 PKCS12 建立
121	找到無法辨識的金鑰匯出類型。
122	找到不受支援的密碼型加密演算法。
123	將金鑰環檔轉換成 CMS 金鑰資料庫時發生錯誤。
124	將 CMS 金鑰資料庫轉換成金鑰環檔案時發生錯誤。
125	建立憑證申請的憑證時發生錯誤。
126	無法建置完整的發證者鏈。
127	找到無效的 WEBDB 資料。
128	沒有要寫入金鑰環檔案的資料。
129	您輸入的天數超出允許的有效期限。
130	密碼太短; 它必須至少包含 {0} 個字元。
131	密碼必須至少包含一個數字。
132	密碼中的所有字元都是英文字母或數值字元。
133	指定了無法辨識或不受支援的簽章演算法。
134	發現無效的資料庫類型。
135	另一個 PKCS#11 裝置正在使用指定的次要金鑰資料庫。
136	未指定次要金鑰資料庫。
137	標籤不存在於 PKCS#11 裝置上。

錯誤碼	錯誤訊息
138	存取 PKCS#11 裝置所需的密碼。
139	存取 PKCS#11 裝置不需要密碼。
140	無法載入加密程式庫。
141	此作業不支援 PKCS#11 。
142	PKCS#11 裝置上的作業失敗。
143	LDAP 使用者不是有效的使用者。 (此程式不支援 LDAP)
144	LDAP 使用者不是有效的使用者。 (此程式不支援 LDAP)
145	LDAP 查詢失敗。 (此程式不支援 LDAP)
146	找到無效的憑證鏈。
147	主要憑證不受信任。
148	發現已撤銷的憑證。
149	加密物件函數失敗。
150	沒有可用的憑證撤銷清冊資料來源。
151	沒有可用的加密記號。
152	無法使用 FIPS 模式。
153	與 FIPS 模式設定發生衝突。
154	輸入的密碼不符合所需強度下限。
200	程式起始設定期間發生失敗。
201	傳遞至 runmqakm 程式的引數記號化失敗。
202	指令中識別的物件不是可辨識的物件。
203	所傳遞的動作不是已知的 -keydb 動作。
204	所傳遞的動作不是已知的 -cert 動作。
205	所傳遞的動作不是已知的 -certreq 動作。
206	所要求的指令遺漏標籤。
207	以 -version 標籤傳遞的值不是可辨識的值。
208	以 -size 標籤傳遞的值不是可辨識的值。
209	以 -dn 標籤傳入的值格式不正確。
210	以 -format 標籤傳入的值不是可辨識的值。
211	開啟檔案時發生相關錯誤。
212	此階段不支援 PKCS12 。
213	您嘗試變更其密碼的加密記號未受密碼保護。
214	此階段不支援 PKCS12 。
215	輸入的密碼不符合所需強度下限。
216	無法使用 FIPS 模式。

錯誤碼	錯誤訊息
217	當到期日超出容許的範圍時，您所輸入的天數。
218	密碼強度未達到最低需求。
219	在所要求的金鑰資料庫中找不到預設憑證。
220	發現無效的信任狀態。
221	發現不受支援的簽章演算法。 在此階段僅支援 MD5 和 SHA1 。
222	該特定作業不支援 PCKS11 。
223	所傳遞的動作不是已知的隨機動作。
224	不容許小於零的長度。
225	使用 -strong 標籤時，密碼長度下限為 14 個字元。
226	使用 -strong 標籤時，密碼長度上限為 300 個字元。
227	處於 FIPS 模式時不支援 MD5 演算法。
228	-cert -list 指令不支援站台標籤。 新增此屬性是為了舊版相容性及潛在的未來加強功能。
229	無法辨識與 -ca 標籤相關聯的值。 值必須是 'true' 或 'false'。
230	以 -type 標籤傳入的值無效。
231	以 -expire 標籤傳入的值低於容許的範圍。
232	不支援使用或要求的加密演算法。
233	目標已存在。

V 9.2.0 V 9.2.0 保護 IBM MQ 元件配置檔中的密碼

為了使用 IBM MQ 的特定特性，可能必須將密碼直接提供給 IBM MQ ，或在該特性所讀取的配置檔內提供。從 IBM MQ 9.2.0 開始，會實作新的密碼保護系統，以容許保護這些配置檔內的密碼。

您應該保護配置檔中的密碼。 下列清單說明用於每一個元件的一般術語:

起始金鑰
您提供用於加密程序的加密金鑰。

小心: 初始金鑰包含敏感資訊。 為了提高安全性，請將它存放在使用者的主目錄中，並使用適當的權限限制存取。

對於列出的每一個元件，在保護或讀取該元件的配置檔中儲存的密碼時，您可以提供包含要使用之加密金鑰的起始金鑰檔。

檔案必須包含至少一個字元的單行。

加密金鑰的長度沒有限制或需求，不過，您的金鑰檔應該至少包含 16 個字元。 例如，您的檔案可能包含下列內容:

```
Th1sIs@n3Ncypt|onK$y
```

此外，您提供的起始金鑰檔應該:

- 包含唯一加密金鑰
- 使用作業系統許可權來適當保護。

預設起始金鑰
加密資料時未提供起始金鑰時所使用的預設加密金鑰。 不過，您 **不應** 使用預設起始金鑰。

純文字字串
已加密的字串，通常是密碼

已編碼密碼
包含已加密密碼的字串，其格式為 IBM MQ 所瞭解的格式。

重要: 您為了與一個元件搭配使用而產生的已編碼密碼字串無法複製到另一個元件的配置檔以供使用。 必須使用元件特定的公用程式來保護每一個元件的每一個密碼。

下列各節列出如何保護支援密碼保護之 IBM MQ 的每一個元件密碼的詳細資料:

- Advanced Message Security
- 第 474 頁的『Managed File Transfer』
- 第 474 頁的『IBM MQ Internet Pass-Thru』
- Deprecated 第 475 頁的『IBM MQ Bridge to blockchain』
- Deprecated 第 475 頁的『IBM MQ Bridge to Salesforce』
- V 9.2.3 第 476 頁的『使用加密硬體的 IBM MQ 用戶端』

Advanced Message Security

Advanced Message Security (AMS) Java 用戶端需要存取包含私密金鑰的金鑰儲存庫，才能保護訊息。

V 9.2.2 Advanced Message Security (AMS) 配置為執行 MCA 截取的 MQI 用戶端或佇列管理程式可能需要存取 PKCS#11 加密硬體或 PEM 檔案，這些檔案包含保護訊息的私密金鑰。

若要存取 這些， a 密碼 ， 必須在稱為 `keystore.conf` 的 AMS 配置檔中提供。 使用 **runamscred** 指令來保護 `keystore.conf` 檔案中包含的機密性資訊。 例如:

```
runamscred -f <keystore configuration file>
```

runamscred 指令使用 **-f** 旗標來保護指定檔案內的機密參數。

V 9.2.2 已將兩個 **runamscred** 程式新增至 IBM MQ 安裝:

- 位於 `<IBM MQ installation root>/bin` 中的 MQI **runamscred** 程式
- 位於 `<IBM MQ installation root>/java/bin` 中的 Java **runamscred** 程式

小心:

1. V 9.2.2 為了確保相容性，請使用 Java **runamscred** 程式來保護要與 Java AMS 用戶端搭配使用的配置檔，以及使用 MQI **runamscred** 程式來保護要與 MQI AMS 用戶端搭配使用的配置檔。
2. 在執行 **runamscred** 之後，您應該驗證所有必要的機密性資訊都受到保護。
3. 您可以將受保護檔案正常提供給已啟用 AMS 的應用程式。

如果要置換或提供要在 AMS 應用程式執行時期使用的起始金鑰檔，或當使用 **runamscred** 來保護金鑰儲存庫配置檔時，請使用下列四種機制之一。 依優先順序，這些是:

1. **-sf** 參數 (僅限 **runamscred**)
2. MQS_AMSCRED_KEYFILE 環境變數
3. 配置檔中的 **amscred.keyfile** 參數
4. 如果未指定上述任何選項，則為預設起始金鑰檔。

小心: V 9.2.2 您不應使用預設起始金鑰。

在 IBM MQ 9.2 之前，已使用不同的密碼保護系統來保護 AMS Java 配置檔中的密碼。

依預設， **runamscred** 程式會使用新系統來保護密碼。 這表示新的配置檔與舊版 AMS Java 不相容。 若要使用舊密碼保護系統來保護配置檔，請使用 **-sp** *0* 旗標。

Managed File Transfer

Managed File Transfer (MFT) 會將存取佇列管理程式或其他資源所需的認證儲存在數個 XML 內容檔中:

- `MQMFTCredentials.xml` -用於連接至代理程式、協調及指令佇列管理程式的認證，以及用於連接至金鑰儲存庫以進行安全通訊的密碼。
- `ProtocolBridgeCredentials.xml` -用於連接至通訊協定伺服器的認證，例如 FTP/SFTP/FTPS。
- `ConnectDirectCredentials.xml` - Connect:Direct 代理程式連接至 Connect:Direct 節點的認證。

如需相關資訊，請參閱 第 478 頁的『加密 MFT 中儲存的認證』 。

若要保護儲存在這些檔案中的機密性資訊，請使用 **-f** 旗標，在指定的檔案內使用 fteObfuscate 指令。 例如:

```
fteObfuscate -f <File to protect>
```

若要提供起始金鑰檔以在保護 MFT 配置期間使用，請使用 **-sf** 旗標:

```
fteObfuscate -f <File to protect> -sf <initial key file>
```

如果您未提供起始金鑰，則會使用預設金鑰來保護機密性資訊，雖然您不應使用此選項。

小心:

1. 在執行 **fteObfuscate** 之後，您應該驗證所有必要的機密性資訊都受到保護。
2. 您可以正常提供受保護的檔案給 MFT。

在執行時期，透過下列三種機制提供要使用的起始金鑰檔。 依優先順序排列如下:

1. 使用 Java 系統內容。
   - V 9.2.0.15 在 IBM MQ 9.2.0 Fix Pack 15 之前，此 Java 系統內容的名稱在產品型號中拼錯為 `com.ibm.wqmfte.cred.keyfile`。 從 IBM MQ 9.2.0 Fix Pack 15 開始，內容名稱的拼字會更正為 `com.ibm.wmqfte.cred.keyfile`。 當 Managed File Transfer 檢查使用者是否指定包含要用於加密及解密認證之起始金鑰的檔案時，會使用這兩個版本的 Java 系統內容。 這可讓您使用內容名稱的正確拼字，同時維護舊版與舊拼錯名稱的相容性。 請注意，如果同時設定兩個 Java 系統內容，則會使用正確拼寫內容 `com.ibm.wmqfte.cred.keyfile` 的值。
   - 在 IBM MQ 9.2.0 Fix Pack 15 之前，請使用內容 `com.ibm.wqmfte.cred.keyfile`。
2. 在代理程式、日誌程式、指令及協調內容檔中。
3. 在 `installation.properties` 檔案中。

在 IBM MQ 9.2 之前，已使用不同的認證保護系統來保護 MFT 配置檔中的認證。

依預設， **fteObfuscate** 會使用新系統來保護認證; 這表示配置檔與舊版 MFT 不相容。

若要使用舊認證保護系統來保護配置檔，請使用 **-sp** *0* 旗標。

IBM MQ Internet Pass-Thru

IBM MQ Internet Pass-Thru (MQIPT) 配置檔可以包含用來存取各種資源的密碼，以及 MQIPT 管理密碼。

您可以使用 MQIPT 隨附的 **mqiptPW** 指令來保護這些密碼。

```
mqiptPW
```

若要使用特定的起始金鑰來保護密碼，請提供 **-sf** 旗標:

```
mqiptPW -sf <intial key file>
```

如需相關資訊，請參閱 指定密碼加密金鑰 。

如果您未提供起始金鑰，則會使用預設金鑰來保護機密性資訊，雖然您不應使用此選項。

mqiptPW 會提示您安全地輸入要保護的密碼，並傳回需要複製到 MQIPT 配置檔的字串。

在執行時期，透過下列四種機制提供要使用的起始金鑰檔。 依優先順序排列如下:

1. 啟動 MQIPT 時透過 **-sf** 參數。
2. 在 MQS_MQIPTCRED_KEYFILE 環境變數中。
3. 在 **com.ibm.mq.ipt.cred.keyfile** Java 內容中。
4. 在 MQIPT 起始目錄中名為 `mqipt_cred.key` 的檔案中，這是包含 MQIPT 配置及日誌檔的目錄，以及其他目錄。

在 IBM MQ 9.2 之前，已使用不同的認證保護系統來保護 MQIPT 配置檔中的認證。

依預設， **mqiptPW** 會使用新系統來保護認證; 這表示配置檔與舊版 MQIPT 不相容。

若要使用舊的認證保護系統來保護金鑰儲存庫密碼，請使用 IBM MQ 9.2 之前版本中支援的 **mqiptPW** 指令語法。

IBM MQ Bridge to blockchain

Deprecated

Bridge to blockchain 配置儲存在可使用 **runmqbcb** 指令產生的檔案中。 執行此指令時，系統會要求您安全地提供要使用的密碼及起始金鑰檔的位置。

若要置換在執行時期或配置模式期間要使用的起始金鑰檔，請使用 **-sf** 旗標。 例如，若要產生具有特定起始金鑰檔的配置，請執行下列動作:

```
runmqbcb -o <output file> -sf <initial key file>
```

或者，在執行時期使用特定的起始金鑰檔:

```
runmqbcb -f <config file> -sf <initial key file>
```

在 IBM MQ 9.2 之前，已使用不同的認證保護系統來保護 Bridge to blockchain 配置檔中的認證。

依預設， **runmqbcb** 會使用新系統來保護認證; 這表示配置檔與舊版 Bridge to blockchain 不相容。

若要使用舊認證保護系統來保護配置檔，請使用 **-sp** *0* 旗標。

重要:

- Deprecated 從 2022 年 11 月 22 日開始的所有版本都已淘汰 IBM MQ Bridge to blockchain (請參閱 美國公告信 222-341)。
- V 9.2.0.21 Removed 若為 Long Term Support，會在 IBM MQ 9.2.0 CSU 21 中移除 IBM MQ Bridge to blockchain 。

IBM MQ Bridge to Salesforce

Deprecated

Bridge to Salesforce 配置儲存在可使用 **runmqsfb** 指令產生的檔案中。 執行此指令時，系統會要求您安全地提供要使用的密碼及起始金鑰檔的位置。

若要置換在執行時期或配置模式期間要使用的起始金鑰檔，請使用 **-sf** 旗標。 例如，若要產生具有特定起始金鑰檔的配置，請執行下列動作:

```
runmqsfb -o <output file> -sf <initial key file>
```

或者，在執行時期使用特定的起始金鑰檔:

```
runmqsfb -f <config file> -sf <initial key file>
```

在 IBM MQ 9.2 之前，已使用不同的認證保護系統來保護 Bridge to Salesforce 配置檔中的認證。

依預設， **runmqfsb** 會使用新系統來保護認證; 這表示配置檔與舊版 Bridge to Salesforce 不相容。

若要使用舊認證保護系統來保護配置檔，請使用 **-sp** *0* 旗標。

重要: 從 2022 年 11 月 22 日開始的所有版本都已淘汰 IBM MQ Bridge to Salesforce (請參閱 美國公告信 222-341)。

使用加密硬體的 IBM MQ 用戶端

V 9.2.3

您可以配置 IBM MQ 用戶端以使用 PKCS #11 加密硬體來儲存 TLS 通訊中使用的私密金鑰及憑證。 為了存取 PKCS #11 裝置，您必須在提供給 IBM MQ client 的配置字串中提供密碼。

重要: 透過 MQSCO 提供的密碼。**SSLCryptoHardware** 結構字串或佇列管理程式 **SSLCRYP** 屬性無法使用此機制來保護。

您可以使用 **runp11cred** 指令來保護此密碼，該指令位於 IBM MQ 安裝根目錄的 bin 資料夾中。

runp11cred 指令會提示您安全輸入要保護的密碼，並傳回需要複製到加密硬體配置字串的字串。

例如，如果您的 GSK_PKCS11 為:

```
GSK_PKCS11=/usr/lib/pkcs11/
PKCS11_API.so;tokenlabel;Passw0rd;SYMMETRIC_CIPHER_ON
```

然後，當系統提示時，輸入 **Passw0rd**。 **runp11cred** 會傳回類似於下列內容的字串:

```
<P11>!2!0TyDxrRaS6JUsjON9zfK6S4wEHmSNF0/ZsOdCaTD2dc=!MdpCoxGnFqPtZ1dTLQ58kg==
```

以粗體複製字串，以取代 GSK_PKCS11 字串中的 **Passw0rd** :

```
GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;<P11>!2!
0TyDxrRaS6JUsjON9zfK6S4wEHm SNF0/ZsOdCaTD2dc=!
MdpCoxGnFqPtZ1dTLQ58kg==;SYMMETRIC_CIPHER_ON
```

若要使用特定的起始金鑰來保護密碼，請使用下列其中一種機制。 依優先順序，這些是:

1. **-sf** 參數 (僅限 **runp11cred** 指令)
2. MQS_SSLCRYP_KEYFILE 環境變數
3. **SSLCryptoHardwareKeyFile** SSL 段落屬性 (僅限 IBM MQ client)
4. 如果未指定上述任何選項，則為預設起始金鑰檔。

小心: 您不應使用預設起始金鑰。

資料庫鑑別的保護詳細資料

如果您使用使用者名稱及密碼鑑別來連接至資料庫管理程式，則可以將它們儲存在 MQ XA 認證儲存庫中，以避免將密碼以純文字儲存在 qm.ini 檔案中。

更新資源管理程式的 XAOpenString

若要使用認證儲存庫，您必須修改 qm.ini 檔案中的 XAOpenString 。 字串用來連接至資料庫管理程式。 您可以指定可更換欄位，以識別在 XAOpenString 字串內替換使用者名稱和密碼的位置。

- +USER+ 欄位會取代為儲存在 XACredentials 儲存庫中的使用者名稱值。
- +PASSWORD+ 欄位會取代為儲存在 XACredentials 儲存庫中的密碼值。

下列範例顯示如何修改 XAOpenString ，以使用認證檔來連接至資料庫。

連接至 Db2 資料庫

```
XAResourceManager:
  Name=mydb2
```

```
  SwitchFile=db2swit
  XAOpenString=db=mydbname,uid=+USER+,pwd=+PASSWORD+,toc=t
  ThreadOfControl=THREAD
```

連接至 Oracle 資料庫

```
XAResourceManager:
  Name=myoracle
  SwitchFile=oraswit
  XAOpenString=Oracle_XA+Acc=P/+USER+/+PASSWORD++SesTm=35
            +LogDir=/tmp+threads=true
  ThreadOfControl=THREAD
```

使用資料庫至 MQ XA 認證儲存庫的認證

使用可更換的認證字串更新 qm.ini 檔案之後，您必須使用 **setmqxacred** 指令，將使用者名稱及密碼新增至 MQ 認證儲存庫。 您也可以使用 **setmqxacred** 來修改現有認證、刪除認證或列出認證。 下列範例提供一些一般使用案例:

新增認證

下列指令會安全地儲存資源 mqdb2 之佇列管理程式 QM1 的使用者名稱及密碼。

```
setmqxacred -m QM1 -x mydb2 -u user1 -p Password2
```

更新認證

若要更新用來連接至資料庫的使用者名稱及密碼，請使用新的使用者名稱及密碼重新發出 **setmqxacred** 指令:

```
setmqxacred -m QM1 -x mydb2 -u user3 -p Password4
```

您必須重新啟動佇列管理程式，變更才會生效。

刪除認證

下列指令會刪除認證:

```
setmqxacred -m QM1 -x mydb2 -d
```

列出認證

下列指令列出認證:

```
setmqxacred -m QM1 -l
```

相關參考

setmqxacred

保護 Managed File Transfer

Managed File Transfer 在剛安裝且未經修改時具有一種安全層次，適合在受保護的環境下進行測試或評估用途。 但是，在正式作業環境中，您必須考量適當地控制誰可以啟動檔案傳送、誰可以讀取及寫入傳送的檔案，以及如何保護檔案的完整性。

相關工作

限制 MFT 特定資源的群組權限

管理 MFT 特定資源的權限

第 531 頁的『將 Advanced Message Security 與 Managed File Transfer 搭配使用』

此實務範例說明如何配置 Advanced Message Security ， 以針對透過 Managed File Transfer 傳送的資料提供訊息隱私權。

相關參考

MFT 存取檔案系統的權限

commandPath MFT 內容

發佈 MFT 代理程式日誌及狀態訊息的權限

V 9.2.0 V 9.2.0 加密 MFT 中儲存的認證

Managed File Transfer (MFT) 需要數個使用者 ID 和認證 (儲存在兩個 XML 檔案中)，您可以使用 **fteObfuscate** 指令來模糊化這些使用者 ID 和認證。從 IBM MQ 9.2.0 開始，此指令提供對儲存認證的加強保護。

認證檔案

MQMFTCredentials.xml
此檔案包含用於連接至代理程式及協調與指令佇列管理程式的使用者 ID 和認證。存取金鑰儲存庫以與佇列管理程式建立安全連線的認證也會儲存在相同檔案中。

如需定義 MQMFTCredentials.xml 檔案位置之內容值的詳細資料，請參閱 第 480 頁的『MFT 及 IBM MQ 連線鑑別』。

ProtocolBridgeCredentials.xml
此檔案包含用於連接至通訊協定伺服器的使用者 ID 及認證。

使用 fteObfuscate 指令加密認證

從 IBM MQ 9.2.0 開始，**fteObfuscate** 指令接受下列參數:

- **credentialsFileName**，這是必要項目
- **protection mode**、**credentialsKeyFile** 和 **outputFileName**，所有這些都是選用項目

如需參數的詳細資料，請參閱 **fteObfuscate**。

如果您未指定保護模式或認證金鑰檔，則指令會使用預設保護模式，並使用最新演算法，但使用固定金鑰來加密認證。

如果您指定保護模式 0，但未指定認證金鑰檔，則指令會如舊版產品一樣運作。您會在主控台上收到警告訊息，指出使用已淘汰的保護。

如果您指定保護模式 0，並指定認證金鑰檔，則會在主控台上收到錯誤輸出，指出在使用保護模式 0 時指定金鑰檔無效。

如果您指定 1 的保護模式，且未指定認證金鑰檔，則指令會使用最新演算法，但使用固定金鑰來加密認證。

如果您指定 1 的保護模式，並指定認證金鑰檔，則指令會使用最新演算法來加密認證。

如果您指定 1 的保護模式，或未指定保護模式，並指定不存在的認證金鑰檔，則會在主控台上輸出錯誤，指出該檔案不存在。

如果您指定 1 的保護模式，或未指定保護模式，並指定無法讀取的認證金鑰檔，則會在主控台上輸出錯誤，指出無法讀取檔案。

V 9.2.4 如果您指定 2 的保護模式，但未指定認證金鑰檔，則指令會使用保護模式 2，以使用最新演算法來加密認證，並使用要加密的固定金鑰來加密認證。

V 9.2.4 如果您指定 2 的保護模式，並指定認證金鑰檔，則指令會使用保護模式 2，以使用最新演算法來加密認證，並使用使用者指定的金鑰來加密。

V 9.2.4 如果您指定 2 的保護模式，或未指定保護模式，並指定不存在的認證金鑰檔，則會在主控台上輸出錯誤，指出該檔案不存在。

V 9.2.4 如果您指定 2 的保護模式，或未指定保護模式，並指定無法讀取的認證金鑰檔，則會在主控台上輸出錯誤，指出無法讀取檔案。

解密認證

您可以在不同位置指定起始金鑰檔的路徑。為了解密使用預設金鑰以外的起始金鑰所加密的認證，必須以下列其中一種方式將包含起始金鑰的檔案名稱提供給 MFT，其優先順序如下:

1. 使用 Java 虛擬機器 (JVM) 內容 `com.ibm.wqmfte.cred.keyfile`，例如:

```
-Dcom.ibm.wmqfte.cred.keyfile=/usr/hime/credkeyfile.key
```

2. 透過在代理程式、指令、協調或日誌程式內容檔中設定內容。 下表顯示內容檔的名稱，以及需要在其中設定的內容:

內容檔	內容名稱
agent.properties	agentCredentialsKeyFile
command.properties	commandCredentialsKeyFile
coordination.properties	coordinationCredentialsKeyFile
logger.properties	loggerCredentialsKeyFile

3. 在 installation.properties 檔案中。

   您可以將 **commonCredentialsKeyFile** 內容新增至現有的一般 `installation.properties` 檔案，讓代理程式、日誌程式及指令可以使用相同的內容，而不是在個別內容檔中新增內容。

您可能已在多個位置中定義各種 **CredentialsKeyFile** 內容，因此認證金鑰檔的路徑用於:

- 代理程式及日誌程式會記載至該代理程式或日誌程式的 `output0.log` 檔案。
- 指令會顯示在主控台上。

Java 系統內容 **com.ibm.wqmfte.cred.keyfile** 會置換所有其他內容。 如果未設定系統內容，代理程式會查看 `agent.properties` 檔案，後面接著起始金鑰檔的 `installation.properties` 檔案。

如果仍找不到起始金鑰檔，且您已將 **fteObfuscate** 指令上的保護模式設為 `1`，則代理程式會在 `output0.log` 檔案中記載錯誤訊息

如果您已在 **fteObfuscate** 指令上將保護模式設為 `0`，則會記載一則警告訊息，指出已淘汰。

日誌程式及指令遵循相同的步驟來尋找起始金鑰檔。

通訊協定橋接器及 Connect:Direct 橋接器

「通訊協定橋接器」使用內容檔 `ProtocolBridgeProperties.xml` 來連接至 FTP、SFTP 及 FTPS 伺服器。 此內容檔包含連接至這些伺服器所需的連線屬性。

如果您在 `ProtocolBridgeProperties.xml` 檔案中修改 **credentialsFile** 或 **credentialsKeyFile** 屬性的值，則需要重新啟動橋接器代理程式。

其中一個屬性是 **credentialsFile**，該值包含 XML 檔案的路徑，該檔案包含連接至這些伺服器所需的 UID 或 PWD 或金鑰。 該屬性的預設值為 *ProtocolBridgeCredentials.xml* ，且該檔案位於您的主目錄，就像 `MQMFTCredentials.xml` 檔案一樣。

```
<tns:credentialsFile path="$HOME/ProtocolBridgeCredentials.xml" />
```

就像 *MQMFTCredentails.xml*, 您可以使用 **fteObfuscate** 指令加密 *ProtocolBridgeCredentials.xml*。 基於解密目的，您可以使用其他元素 **credentialsKeyFile** 來指定認證金鑰檔的必要路徑，如下列文字所示。路徑可以包含環境變數。

```
<tns:credentialsKeyFile path="$HOME/CredKey.key"/>
```

註: 在 `installation.properties` 中或透過系統內容 **com.ibm.wqmfte.cred.keyfile** 指定 **agentCredentialsKeyFile** 代理程式內容 **commonCredentialsKeyFile** 內容的值，不會對指定給 **credentialsKeyFile** 屬性的值產生任何影響。

同樣地， Connect:Direct Bridge 使用 *ConnectDirectNodeProperties.xml* 連線到 Connect:Direct 伺服器。 XML 檔案包含必要的連線資訊，以及定義認證 XML 檔案路徑的屬性。 此認證 XML 檔案包含 UID 或 PWD ，以及連接至 Connect:Direct 伺服器所需的其他資訊。

```
<tns:credentialsFile path="$HOME/ ConnectDirectCredentials.xml" />
```

就像 *ProtocolBridgeCredentials.xml* 檔案一樣，您可以使用 **fteObfuscate** 指令加密 *ConnectDirectCredentials.xml*。 基於解密目的，您可以使用其他元素 **credentialsKeyFile** 來指定認證金鑰檔的必要路徑，如下列文字所示。 路徑可以包含環境變數。

```
<tns:credentialsKeyFile path="$HOME/CredKey.key”/>
```

註: 在 `installation.properties` 中或透過系統內容 **com.ibm.wqmfte.cred.keyfile** 指定 **agentCredentialsKeyFile** 代理程式內容 **commonCredentialsKeyFile** 內容的值，不會對指定給 **credentialsKeyFile** 屬性的值產生任何影響。

您可以指定 **credentialsKeyFile** 元素，而不在 *ProtocolBridgeProperties.xml* 檔案中指定 **credentialsFile** 元素。

如果不指定 **credentialsFile** 元素，協議橋接器代理會使用預設憑證檔案 *ProtocolBridgeCredentials.xml* ，並使用 **credentialsKeyFile** 屬性中指定的金鑰檔案的值來解密憑證檔案。

同樣地，您可以指定 **credentialsKeyFile** 元件，而不在 *ConnectDirectNodeProperties.xml* 檔案中指定 **credentialsFile** 元件。

如果不指定 **credentialsFile** 元素， Connect:Direct 网桥将使用默认的凭据文件 *ConnectDirectCredentials.xml* ，并使用 **credentialsKeyFile** 属性中指定的密钥文件的值来解密凭据文件。

在 z/OS 上使用資料集中的索引鍵

z/OS

在 z/OS 上，您可以指定 **MQMFTCredentials** ，並使用 PDSE 提供認證金鑰檔。 請參閱第 482 頁的『在 z/OS 上配置 MQMFTCredentials.xml』。

相關參考

MFT installation.properties 檔
各個 MFT 指令對應連接的佇列管理程式
MFT 認證檔案格式
fteObfuscate (加密機密資料)

MFT 及 IBM MQ 連線鑑別

連線鑑別可讓佇列管理程式配置成使用提供的使用者 ID 和密碼來鑑別應用程式。 如果相關聯的佇列管理程式已啟用安全，且需要認證詳細資料 (使用者 ID 和密碼) ，則必須先啟用連線鑑別功能，才能順利建立與佇列管理程式的連線。 連線鑑別可以在相容模式或 MQCSP 鑑別模式中執行。

提供認證詳細資料的方法

許多 Managed File Transfer 指令支援下列方法來提供認證詳細資料:

指令行引數所提供的詳細資料。
可以使用 **-mquserid** 和 **-mqpassword** 參數來指定認證詳細資料。 如果未提供 **-mqpassword** ，則會要求使用者提供未顯示輸入的密碼。

從認證檔提供的詳細資料：MQMFTCredentials.xml。
可以在 `MQMFTCredentials.xml` 檔案中將認證詳細資料預先定義為明碼或模糊文字。

如需在 IBM MQ for Multiplatforms 上設定 MQMFTCredentials.xml 檔案的相關資訊，請參閱 第 481 頁的『在 Multiplatforms 上配置 MQMFTCredentials.xml』。

如需在 IBM MQ for z/OS 上設定 MQMFTCredentials.xml 檔案的相關資訊，請參閱 第 482 頁的『在 z/OS 上配置 MQMFTCredentials.xml』。

優先順序

判定認證詳細資料的優先順序如下:

1. 指令行引數。
2. 依相關聯佇列管理程式及執行指令之使用者的 MQMFTCredentials.xml 索引。
3. 依相關聯佇列管理程式的 MQMFTCredentials.xml 索引。
4. 未提供認證詳細資料以容許與舊版 IBM MQ 或 IBM WebSphere MQ 相容的預設舊版相容模式

附註:

- **fteStartAgent** 及 **fteStartLogger** 指令不支援指令行引數 **-mquserid** 或 **-mqpassword**，因此認證詳細資料只能使用 MQMFTCredentials.xml 檔案予以指定。
- z/OS

  在 z/OS 上，即使使用者的密碼使用小寫字母，密碼也必須為大寫。 例如，若 使用者的密碼是 "password"，便須輸入為 "PASSWORD"。

相關參考

各個 MFT 指令對應連接的佇列管理程式

MFT 認證檔案格式

在 Multiplatforms 上配置 MQMFTCredentials.xml

如果在已啟用安全的情況下配置 Managed File Transfer (MFT)，則連線鑑別需要所有與佇列管理程式連接的 MFT 指令提供使用者 ID 和密碼認證。 同樣地，當連接至資料庫時，可能需要 MFT 日誌程式來指定使用者 ID 和密碼。 此認證資訊可以儲存在 MFT 認證檔中。

關於這項作業

MQMFTCredentials.xml 檔案中的元素必須符合 MQMFTCredentials.xsd 綱目。 如需 MQMFTCredentials.xml 格式的相關資訊，請參閱 MFT 認證檔案格式。

您可以在 MQ_INSTALLATION_PATH/mqft/samples/credentials 目錄中找到範例認證檔。

您的協調佇列管理程式、指令佇列管理程式、每個代理程式和每個日誌程式，可分別有一個 MFT 認證檔。 或者，您也可以有一個檔案，供拓蹼中的所有項目使用。

MFT 認證檔的預設位置如下:

Linux AIX **AIX and Linux**

$HOME

Windows **Windows**

%USERPROFILE% 或 %HOMEDRIVE%%HOMEPATH%

如果認證檔儲存在不同位置，則您可以使用下列內容來指定指令應該尋找它的位置:

表 *95: :* 這些內容定義各種指令的 *MQMFTCredentials.xml* 檔案位置。

指令類型	內容檔	內容名稱
連接至協調佇列管理程式的指令	coordination.properties	coordinationQMgrAuthenticationCredentialsFile
連接至指令佇列管理程式的指令	connection.properties	connectionQMgrAuthenticationCredentialsFile
連接至代理程式處理程序的指令	agent.properties	agentQMgrAuthenticationCredentialsFile
連接至日誌程式處理程序的指令	logger.properties	loggerQMgrAuthenticationCredentialsFile

表 96: : 定義代理程式及日誌程式處理程序的 *MQMFTCredentials.xml* 檔案位置的內容。		
指令類型	**內容檔**	**內容名稱**
MFT 代理程式	agent.properties	agentQMgrAuthenticationCredentialsFile
MFT 日誌程式	logger.properties	loggerQMgrAuthenticationCredentialsFile

如需哪些指令及處理程序連接至哪個佇列管理程式的詳細資料，請參閱 哪些 MFT 指令及處理程序連接至哪個佇列管理程式。

V 9.2.0 V 9.2.0 您可以將 **commonCredentialsKeyFile** 內容新增至現有的一般 `installation.properties` 檔案，讓代理程式、日誌程式及指令可以使用相同的內容，而不是在個別內容檔中新增內容。

重要: 由於此檔案包含敏感的使用者 ID 和密碼資訊，因此必須透過設定適當的權限來保護憑證檔案，以防止未經授權的存取:

1. Linux AIX **AIX and Linux**

   ```
   chown <agent owner userid>
   chmod 600
   ```

2. Windows **Windows**
   請確定未啟用繼承，然後移除所有使用者 ID，但執行代理程式或日誌程式且將使用認證檔的使用者 ID 除外。

在的 IBM MQ Explorer Managed File Transfer 外掛程式中，用來連接至 MFT 協調佇列管理程式的認證詳細資料取決於配置類型:

廣域（本端磁碟上的配置）
廣域配置會使用協調和指令內容中指定的認證檔。

本端（於「IBM MQ Explorer」內定義）：
本端配置會使用「IBM MQ Explorer」中相關聯佇列管理程式的連線詳細資料內容。

相關工作
第 484 頁的『啟用 MFT 的連線鑑別』
使用協調佇列管理程式或指令佇列管理程式連接之 IBM MQ Explorer MFT 外掛程式的連線鑑別，以及使用協調佇列管理程式或指令佇列管理程式連接之 Managed File Transfer 代理程式的連線鑑別，可以在相容模式或 MQCSP 鑑別模式下執行。

相關參考
MFT 認證檔案格式
fteObfuscate：加密機密資料

z/OS 在 z/OS 上配置 MQMFTCredentials.xml

如果在已啟用安全的情況下配置 Managed File Transfer (MFT)，則連線鑑別需要所有 MFT 代理程式以及連接至佇列管理程式的指令，才能提供使用者 ID 和密碼認證。

同樣地，當連接至資料庫時，可能需要 MFT 日誌程式來指定使用者 ID 和密碼。

此認證資訊可以儲存在 MFT 認證檔中。 請注意，認證檔是選用的，不過，在自訂環境之前更容易定義您需要的一或多個檔案。

此外，如果您有認證檔，則會收到較少的警告訊息。 警告訊息會通知您 MFT 認為佇列管理程式安全已關閉，因此您不應提供鑑別詳細資料。

您可以在 `MQ_INSTALLATION_PATH/mqft/samples/credentials` 目錄中找到範例認證檔。

以下是 `MQMFTCredentials.xml` 檔的範例:

```
<?xml version="1.0" encoding="IBM-1047"?>
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MFTCredentials"
```

```
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/MFTCredentials MFTCredentials.xsd">
  <tns:qmgr name="MQPH" user="ADMIN" mqUserId="JOHNDOEH" mqPassword="cXXXX" />
  <tns:qmgr name="MQPI" user="ADMIN" mqUserId="JOHNDOEI" mqPassword="yXXXX" />
  <tns:qmgr name="MQPH" mqUserId="NONEH" mqPassword="yXXXX" />
  <tns:qmgr name="MQPI" mqUserId="NONEI" mqPassword="yXXXX" />
</tns:mqmftCredentials>
```

使用者 ID 為 ADMIN 的工作在需要連接到佇列管理程式 MQPH 時，將會傳遞使用者 ID *JOHNDOEH*，並使用密碼 *cXXXX*。

如果工作由任何其他使用者 ID 執行，並連接 MQPH，該工作將會傳遞使用者 ID *NONEH* 和密碼 *yXXXX*。

`MQMFTCredentials.xml` 檔的預設位置是使用者在 z/OS UNIX System Services (USS) 上的起始目錄。 也可以將檔案儲存在 USS 上的不同位置，或儲存在分割資料集內的成員中。

如果認證檔儲存在不同位置，則您可以使用下列內容來指定指令應該尋找它的位置:

表 *97: :* 這些內容定義各種指令的 *MQMFTCredentials.xml* 檔案位置。

指令類型	**內容檔**	**內容名稱**
連接至協調佇列管理程式的指令	coordination.properties	coordinationQMgrAuthenticationCredentialsFile
連接至指令佇列管理程式的指令	connection.properties	connectionQMgrAuthenticationCredentialsFile
連接至代理程式處理程序的指令	agent.properties	agentQMgrAuthenticationCredentialsFile
連接至日誌程式處理程序的指令	logger.properties	loggerQMgrAuthenticationCredentialsFile

表 *98: :* 定義代理程式及日誌程式處理程序的 *MQMFTCredentials.xml* 檔案位置的內容。

指令類型	**內容檔**	**內容名稱**
MFT 代理程式	agent.properties	agentQMgrAuthenticationCredentialsFile
MFT 日誌程式	logger.properties	loggerQMgrAuthenticationCredentialsFile

如需哪些指令及處理程序連接至哪個佇列管理程式的詳細資料，請參閱 哪些 MFT 指令及處理程序連接至哪個佇列管理程式。

若要在分割的資料集內建立認證檔，請執行下列步驟:

- 使用格式 VB 和邏輯記錄長度 (Lrecl) 200 來建立 PDSE。
- 在資料集內建立成員、記下資料集和成員，然後將下列程式碼新增至成員:

```
<?xml version="1.0" encoding="IBM-1047"?>
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MQMFTCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/MFTCredentials MQMFTCredentials.xsd">
  <!--credentials information goes here-->
</tns:mqmftCredentials>
```

您可以使用安全產品 (例如 RACF) 來保護認證檔，但執行 Managed File Transfer 指令及管理代理程式和日誌程式處理程序的使用者 ID 需要此檔案的讀取權。

您可以在成員 BFGCROBS 中使用 JCL，以遮蔽此檔案中的資訊。 這會取走檔案，並將 IBM MQ 使用者 ID 和密碼加密。 例如，成員 BFGCROBS 取走以下一行

```
<tns:qmgr name="MQPI" user="JOHNDOE2" mqUserId="JOHNDOE1" mqPassword="yXXXX" />
```

並建立

```
<tns:qmgr mqPasswordCipher="e977c61e9b9c363c" mqUserIdCipher="c394c5887867157c"
name="MQPI" user="JOHNDOE2"/>
```

如果您想要將使用者 ID 保留在 IBM MQ 使用者 ID 對映，您可以在檔案中新增註解。 例如

```
<!--      name="MQPI" user="ADMIN"    mqUserId="JOHNDOE1 -->
```

這些註解並不會因為遮蔽程序而變更。

請注意，內容只是遮蔽而已，並未嚴密加密。 您應限制哪些使用者 ID 可存取檔案。

相關工作

第 481 頁的『在 Multiplatforms 上配置 MQMFTCredentials.xml』

如果在已啟用安全的情況下配置 Managed File Transfer (MFT)，則連線鑑別需要所有與佇列管理程式連接的 MFT 指令提供使用者 ID 和密碼認證。 同樣地，當連接至資料庫時，可能需要 MFT 日誌程式來指定使用者 ID 和密碼。 此認證資訊可以儲存在 MFT 認證檔中。

啟用 MFT 的連線鑑別

使用協調佇列管理程式或指令佇列管理程式連接之 IBM MQ Explorer MFT 外掛程式的連線鑑別，以及使用協調佇列管理程式或指令佇列管理程式連接之 Managed File Transfer 代理程式的連線鑑別，可以在相容模式或 MQCSP 鑑別模式下執行。

關於這項作業

在 IBM MQ 9.2.0 之前，相容模式是連線鑑別的預設值。 不過，您可以停用預設相容模式，並啟用 MQCSP 鑑別模式。

V 9.2.0 從 IBM MQ 9.2.0 開始， MQCSP 鑑別模式是預設值。

對於使用 CLIENT 傳輸連接至佇列管理程式的 IBM MQ Explorer Managed File Transfer 外掛程式或 Managed File Transfer 代理程式的連線鑑別，只有 MQCSP 鑑別模式才支援長度超過 12 個字元的密碼。 如果您在使用相容模式授權時指定的密碼長度超過 12 個字元，則會發生錯誤，且代理程式不會向佇列管理程式進行鑑別。 請參閱 診斷訊息: BFGAG0001 - BFGAG9999 中的 BFGAG0187E 訊息。

程序

- 若要在 IBM MQ Explorer 中選取協調佇列管理程式或指令佇列管理程式的連線鑑別模式，請完成下列步驟:
  a) 選取要連接的佇列管理程式。
  b) 按一下滑鼠右鍵並從蹦現功能表中選取**連線詳細資料 -> 內容**。
  c) 按一下**使用者 ID** 標籤。
  d) 確定已選取您要使用之連線鑑別模式的勾選框:
  - V 9.1.0 從 IBM MQ 9.1.0 開始，依預設會取消選取 **使用者識別相容模式** 勾選框。 這表示如果選取 **啟用使用者識別** 勾選框，則 IBM MQ Explorer 會在連接至佇列管理程式時使用 MQCSP 鑑別。 如果 IBM MQ Explorer 需要使用相容模式而非 MQCSP 鑑別來連接至佇列管理程式，請確保同時選取 **啟用使用者識別** 及 **使用者識別相容模式** 勾選框。
  - 在 IBM MQ 9.1.0 之前，依預設會選取 **使用者識別相容模式** 勾選框。 這表示如果選取 **啟用使用者識別** 勾選框，則 IBM MQ Explorer 會在連接至佇列管理程式時使用相容模式。 如果 IBM MQ Explorer 需要使用 MQCSP 鑑別連接至佇列管理程式，請確定已選取 **啟用使用者識別** 勾選框，且未選取 **使用者識別相容模式** 勾選框。
- 若要使用 `MQMFTCredentials.xml` 檔案來啟用或停用 Managed File Transfer 代理程式的 MQCSP 鑑別模式，請將參數 **useMQCSPAuthentication** 新增至相關使用者的 `MQMFTCredentials.xml` 檔案。

  useMQCSPAuthentication 參數具有下列值:

  true
  MQCSP 鑑別模式是用來向佇列管理程式鑑別使用者。

V 9.2.0 從 IBM MQ 9.2.0 開始， `true` 是預設值。 如果未指定 **useMQCSPAuthentication** 參數，依預設會將它設為 `true` ，並使用 MQCSP 鑑別模式向佇列管理程式鑑別使用者。

false
相容模式是用來向佇列管理程式鑑別使用者。

在 IBM MQ 9.2.0 之前，如果未指定 **useMQCSPAuthentication** 參數，則依預設會設為 `false` ，並使用相容模式向佇列管理程式鑑別使用者。

下列範例說明如何在 `MQMFTCredentials.xml` 檔案中設定 **useMQCSPAuthentication** 參數：

```
<tns:qmgr name="CoordQueueMgr" user="ernest" mqUserId="ernest"
    mqPassword="AveryL0ngPassw0rd2135" useMQCSPAuthentication="true"/>
```

相關概念
第 25 頁的『MQCSP 密碼保護』
從 IBM MQ 8.0 開始，您可以使用 IBM MQ 功能來傳送受保護的 MQCSP 結構中包含的密碼，或使用 TLS 加密來加密的密碼。

相關參考
第 480 頁的『MFT 及 IBM MQ 連線鑑別』
連線鑑別可讓佇列管理程式配置成使用提供的使用者 ID 和密碼來鑑別應用程式。 如果相關聯的佇列管理程式已啟用安全，且需要認證詳細資料 (使用者 ID 和密碼) ，則必須先啟用連線鑑別功能，才能順利建立與佇列管理程式的連線。 連線鑑別可以在相容模式或 MQCSP 鑑別模式中執行。

MFT 認證檔案格式

MFT 沙盤推演

您可以限制代理程式在傳送過程中可存取的檔案系統區域。 代理程式受限的區域稱為沙盤推演。 您可以將限制套用至代理程式或要求傳送的使用者。

代理程式為通訊協定橋接器代理程式或 Connect:Direct 橋接器代理程式時，不支援沙盤推演。 對於必須傳送至 IBM MQ 佇列或從該處傳送的代理程式，您無法使用代理程式沙盤推演。

相關參考
第 485 頁的『使用 MFT 代理程式沙盤推演』
若要對 Managed File Transfer 增加其他安全等級，您可以限制代理程式可存取的檔案系統區域。

第 486 頁的『使用 MFT 使用者沙盤推演』
您可以根據要求傳送的 MQMD 使用者名稱，限制可來回傳送檔案的檔案系統區域。

使用 MFT 代理程式沙盤推演

若要對 Managed File Transfer 增加其他安全等級，您可以限制代理程式可存取的檔案系統區域。

對於傳送至 IBM MQ 佇列或從該處傳送的代理程式，您無法使用代理程式沙盤推演。 以沙盤推演限制 IBM MQ 佇列的存取，可改為透過使用者沙盤推演加以實作，建議所有沙盤推演需求皆利用此解決方案。 如需使用者沙盤推演的相關資訊，請參閱第 486 頁的『使用 MFT 使用者沙盤推演』。

若要啟用代理程式沙盤推演作業，請針對您想要限制的代理程式，將下列內容新增至 `agent.properties` 檔中：

```
sandboxRoot=[!]restricted_directory_nameseparator...separator[!]restricted_directory_name
```

其中：

- *restricted_directory_name* 是要允許或拒絕的目錄路徑。
- ！是選用項目，並指定拒絕 (排除) *restricted_directory_name* 的下列值。 如果未指定！，則 *restricted_directory_name* 是容許 (併入) 的路徑。
- *separator* 是平台專用分隔字元。

比方說，例如想要限制 AGENT1 只能存取 /tmp 目錄，但不允許存取子目錄 private，請在屬於 AGENT1 的 agent.properties 檔中設定如下內容：sandboxRoot=/tmp:!/tmp/private。

sandboxRoot 內容詳述於進階代理程式內容中。

通訊協定橋接器代理程式或 Connect:Direct 橋接器代理程式上，皆不支援代理程式及使用者沙盤推演。

在 AIX, Linux, and Windows 平台上使用沙盤推演

ALW 在 AIX, Linux, and Windows 平台上，沙盤推演作業會限制 Managed File Transfer Agent 可以讀取及寫入哪些目錄。 啟動沙盤推演作業時，Managed File Transfer Agent 可讀取及寫入指定為允許的目錄，以及所指定目錄包含的任何子目錄，除非子目錄在 sandboxRoot 中指定為拒絕。 Managed File Transfer 沙盤推演作業的優先順序不高於作業系統安全。 啟動 Managed File Transfer Agent 的使用者對任何目錄必須具有適當的作業系統層次存取權，才能夠讀取或寫入該目錄。 如果鏈結的目錄位於指定的 sandboxRoot 目錄（及子目錄）外，則不會遵循目錄的符號鏈結。

在 z/OS 上使用沙盤推演

z/OS 在 z/OS 上，沙盤推演作業會限制 Managed File Transfer Agent 可讀取及寫入的資料集名稱限定元。 啟動 Managed File Transfer Agent 的使用者對任何相關資料集必須具有正確的作業系統權限。 如果以雙引號括住 sandboxRoot 資料集名稱限定元的值，則值遵循一般 z/OS 慣例，且視為完整路徑。 如果省略雙引號，則 sandboxRoot 會以現行使用者 ID 為字首。 例如，如果您設定 sandboxRoot 內容如下：sandboxRoot=//test，則代理程式可存取下列資料集（採用標準 z/OS 表示法）//*username*.test.** 在執行時期，如果完整解析的資料集名稱的起始層次不符合 sandboxRoot，則會拒絕傳送要求。

在 IBM i 系統上使用沙盤推演

IBM i 若為 IBM i 系統的整合檔案系統中的檔案，沙盤推演作業會限制 Managed File Transfer Agent 可讀取及寫入的目錄。 啟動沙盤推演作業時，Managed File Transfer Agent 可讀取及寫入指定為允許的目錄，以及所指定目錄包含的任何子目錄，除非子目錄在 sandboxRoot 中指定為拒絕。 Managed File Transfer 沙盤推演作業的優先順序不高於作業系統安全。 啟動 Managed File Transfer Agent 的使用者對任何目錄必須具有適當的作業系統層次存取權，才能夠讀取或寫入該目錄。 如果鏈結的目錄位於指定的 sandboxRoot 目錄（及子目錄）外，則不會遵循目錄的符號鏈結。

相關參考

第 489 頁的『對萬用字元傳送進行其他檢查』
如果已使用使用者或代理程式沙盤推演來配置代理程式，以限制代理程式可以在其中來回傳送檔案的位置，則您可以指定對該代理程式的萬用字元傳送進行其他檢查。

第 485 頁的『使用 MFT 代理程式沙盤推演』
若要對 Managed File Transfer 增加其他安全等級，您可以限制代理程式可存取的檔案系統區域。

MFT agent.properties 檔案

使用 MFT 使用者沙盤推演

您可以根據要求傳送的 MQMD 使用者名稱，限制可來回傳送檔案的檔案系統區域。

當代理程式是通訊協定橋接器代理程式或 Connect:Direct 橋接器代理程式時，不支援使用者沙盤推演。

若要啟用使用者沙盤推演，請針對您想要限制的代理程式，將下列內容新增至 agent.properties 檔案中：

```
userSandboxes=true
```

有了這項內容並設為 true 時，代理程式就會使用 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/agents/*agent_name*/UserSandboxes.xml 檔案中的資訊，來決定要求傳送的使用者可以存取檔案系統的哪些部分。

UserSandboxes.xml XML 由包含零個以上 <sandbox> 元素的 <agent> 元素組成。 這些元素說明哪些規則適用於哪些使用者。 <sandbox> 元素的 user 屬性是一種型樣，用來比對要求的 MQMD 使用者。

代理程式會定期重新載入 UserSandboxes.xml 檔案，因此對該檔案所做的任何有效變更，都會影響代理程式的行為。 預設重新載入間隔是 30 秒。 透過指定 agent.properties 檔案中的代理程式內容 xmlConfigReloadInterval，即可變更此間隔。

如果指定 userPattern="regex" 屬性或值，則 user 屬性會解譯為 Java 正規表示式。 如需相關資訊，請參閱 MFT 使用的正規表示式。

如果您未指定 userPattern="regex" 屬性或值，則 user 屬性會解譯為具有下列萬用字元的型樣:

- 星號 (*)，代表零個以上字元
- 問號 (?) ，正好代表一個字元

會依照該檔案中列出 <sandbox> 元素的順序來執行比對。 只會使用第一個相符項，而忽略該檔案中所有後續可能的相符項。 如果該檔案中指定的 <sandbox> 元素全部都不符合與傳送要求訊息相關聯的 MQMD 使用者，則傳送時將無法存取檔案系統。 在 MQMD 使用者名稱與 user 屬性之間找到相符項之後，此相符項將識別 <sandbox> 元素內套用至傳送的一組規則。 這組規則用來決定傳送過程中可以讀取或寫入的檔案或資料集。

每一組規則可指定 <read> 元素來識別可讀取的檔案，以及指定 <write> 元素來識別可寫入的檔案。 如果您在一組規則中省略了 <read> 或 <write> 元素，即假設與該組規則相關聯的使用者不得執行任何讀取或寫入。

註: 在 UserSandboxes.xml 檔中，<read> 元素必須在 <write> 元素之前，<include> 元素必須在 <exclude> 元素之前。

每一個 <read> 或 <write> 元素包含一個以上型樣，用來決定沙盤推演中是否存在檔案以及是否可以進行傳送。 請使用 <include> 及 <exclude> 元素來指定這些型樣。 <include> 或 <exclude> 元素的 name 屬性指定要比對的型樣。 選用的 type 屬性指定名稱值是檔案或佇列型樣。 如果未指定 type 屬性，代理程式會將此型樣視為檔案或目錄路徑型樣。 例如:

```
<tns:read>
    <tns:include name="/home/user/**"/>
    <tns:include name="USER.**" type="queue"/>
    <tns:exclude name="/home/user/private/**"/>
</tns:read>
```

代理程式使用 <include> 及 <exclude> name 型樣，來決定是否可以讀取或寫入檔案、資料集或佇列。 如果標準檔案路徑、資料集或佇列名稱符合至少其中一個包括的型樣，而排除的型樣一個都不符合，則容許作業。 使用 <include> 及 <exclude> 元素的 name 屬性指定的型樣，會使用適用於代理程式執行所在平台的路徑分隔字元及使用慣例。 如果您指定相對檔案路徑，則解析的路徑將相對於代理程式的 transferRoot 內容。

指定佇列限制時，支援 QUEUE@QUEUEMANAGER 的語法，且規則如下:

- 如果項目中遺漏 at 字元 (@)，則將該型樣視為可以在任何佇列管理程式上存取的佇列名稱。 比方說，例如型樣是 name，則將它視為與 name@** 相同。
- 如果 at 字元 (@) 是項目中的第一個字元，則將該型樣視為佇列管理程式名稱，並可存取該佇列管理程式上的所有佇列。 比方說，例如型樣是 @name，則將它視為與 **@name 相同。

當您指定下列萬用字元作為 <include> 及 <exclude> 元素的 name 屬性的一部分時，它們具有特殊意義:

*****
在目錄名稱中，或在 資料集名稱或 佇列名稱的限定元中，單一星號符合零或多個字元。

?
問號符合目錄名稱中的正好一個字元，或符合 資料集名稱或 佇列名稱的限定元。

******
兩個星號字元符合零或多個目錄名稱，或 資料集名稱或 佇列名稱中零或多個限定元。 同時，結尾是路徑分隔字元的路徑有隱含的 "**" 新增至路徑結尾。 因此，/home/user/ 與 /home/user/** 相同。

例如:

- /**/test/** 符合其路徑中有 test 目錄的任何檔案

- /test/file? 符合 /test 目錄內以字串 file 開頭且後面接著任何單一字元的任何檔案
- c:\test*.txt 符合 c:\test 目錄內具有 .txt 副檔名的任何檔案
- c:\test***.txt 符合 'c:\test 目錄內任何檔案，或其中一個子目錄具有 .txt 副檔名
- z/OS //'TEST.*.DATA' 符合其第一個限定元為 TEST、有任何第二個限定元，且第三個限定元為 DATA 的任何資料集。
- *@QM1 符合佇列管理程式 QM1 中具有單一限定元的任何佇列。
- TEST.*.QUEUE@QM1 符合佇列管理程式 QM1 上其第一個限定元為 TEST、有任何第二個限定元，且第三個限定元為 QUEUE 的任何佇列。
- **@QM1 符合佇列管理程式 QM1 上的任何佇列。

符號鏈結

您必須完整解析 UserSandboxes.xml 檔案中檔案路徑所使用的任何符號鏈結，方法是在 <include> 及 <exclude> 元素中指定固定鏈結。 例如，如果您具有 /var 對映至 /SYSTEM/var 的符號鏈結，則必須將此路徑指定為 <tns:include name="/SYSTEM/var"/>，否則預期的傳送會因使用者沙盤推演安全錯誤而失敗。

範例

此範例顯示如何容許具有 MQMD 使用者名稱 guest 的使用者，透過將下列 <sandbox> 元素新增至 AGENT_JUPITER 配置目錄中的 UserSandboxes.xml 檔案，從代理程式 AGENT_JUPITER 執行所在系統上的 /home/user/public 目錄或其任何子目錄中傳送任何檔案:

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:userSandboxes
        xmlns:tns="http://wmqfte.ibm.com/UserSandboxes"
        xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://wmqfte.ibm.com/UserSandboxes UserSandboxes.xsd">
    <tns:agent>
        <tns:sandbox user="guest">
            <tns:read>
                <tns:include name="/home/user/public/**"/>
             </tns:read>
        </tns:sandbox>
    </tns:agent>
</tns:userSandboxes>
```

範例

此範例說明如何容許具有 MQMD 使用者名稱 account 且後面跟著單一數字的任何使用者（例如 account4）完成下列動作:

- 傳送來自 /home/account 目錄或其任何子目錄的任何檔案（不包括代理程式 AGENT_SATURN 執行所在系統上的 /home/account/private 目錄）
- 將任何檔案傳送至代理程式 AGENT_SATURN 執行所在系統上的 /home/account/output 目錄或其任何子目錄
- 從本端佇列管理程式上以字首 ACCOUNT. 開頭的佇列中讀取訊息，除非它以 ACCOUNT.PRIVATE. 開頭 (即在第二層具有 PRIVATE)。
- 將資料傳送至任何佇列管理程式上開頭為字首 ACCOUNT.OUTPUT. 的佇列上。

若要容許具有 MQMD 使用者名稱 account 的使用者完成這些動作，請將下列 <sandbox> 元素新增至 AGENT_SATURN 配置目錄中的 UserSandboxes.xml 檔案:

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:userSandboxes
            xmlns:tns="http://wmqfte.ibm.com/UserSandboxes"
            xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="http://wmqfte.ibm.com/UserSandboxes UserSandboxes.xsd">
    <tns:agent>
        <tns:sandbox user="account[0-9]" userPattern="regex">
            <tns:read>
                <tns:include name="/home/account/**"/>
```

```
                <tns:include name="ACCOUNT.**" type="queue"/>
                <tns:exclude name="ACCOUNT.PRIVATE.**" type="queue"/>
                <tns:exclude name="/home/account/private/**"/>
                                                        </tns:read>
            <tns:write>
                <tns:include name="/home/account/output/**"/>
                <tns:include name="ACCOUNT.OUTPUT.**" type="queue"/>
            </tns:write>
        </tns:sandbox>
    </tns:agent>
</tns:userSandboxes>
```

相關參考

第 489 頁的『對萬用字元傳送進行其他檢查』
如果已使用使用者或代理程式沙盤推演來配置代理程式，以限制代理程式可以在其中來回傳送檔案的位置，則您可以指定對該代理程式的萬用字元傳送進行其他檢查。

MFT agent.properties 檔案

對萬用字元傳送進行其他檢查

如果已使用使用者或代理程式沙盤推演來配置代理程式，以限制代理程式可以在其中來回傳送檔案的位置，則您可以指定對該代理程式的萬用字元傳送進行其他檢查。

additionalWildcardSandboxChecking 內容

若要對萬用字元傳送啟用其他檢查，請針對您想要檢查的代理程式，將下列內容新增至 agent.properties 檔案中：

```
additionalWildcardSandboxChecking=true
```

如果將此內容設為 true，而且代理程式所提出的傳送要求，嘗試讀取的位置與萬用字元相符，但位於定義給檔案的沙盤推演外部，則傳送會失敗。 如果在一個傳送要求中進行多個傳送，而且其中一個要求由於嘗試讀取沙盤推演外部的位置而失敗，則整個傳送都會失敗。 如果檢查失敗，則錯誤訊息會提供失敗原因。

如果 additionalWildcardSandboxChecking 內容已從代理程式的 agent.properties 檔案中省略或者設為 false，則不會對該代理程式的萬用字元傳送進行其他檢查。

萬用字元檢查的錯誤訊息

對所配置沙盤推演位置以外的位置發出萬用字元傳送要求時所報告的訊息如下。

如果傳送要求中的萬用字元檔案路徑位於所要求沙盤推演外部，則會出現下列訊息：

BFGSS0077E: 嘗試讀取檔案路徑 *path* 遭拒絕。
此檔案路徑位於限制性傳送沙盤推演外部。

如果多重傳送要求中的一項傳送包含一個萬用字元傳送要求，但其中的路徑位於限制性沙盤推演外部，則會出現下列訊息：

BFGSS0078E: Attempt to read file path: *path* has being ignored as another transfer
受管理傳送中的另一個傳送項目嘗試在限制性傳送沙盤推演外部進行讀取。

如果檔案位於限制性沙盤推演外部，則會出現下列訊息：

BFGSS0079E: 嘗試讀取檔案 *file path* 遭拒絕。
該檔案位於受限傳送沙盤推演外。

在多重傳送要求中，如果一個萬用字元傳送要求由於其他萬用字元傳送要求而遭忽略，則會出現下列訊息：

BFGSS0080E: 嘗試讀取檔案: *檔案路徑* 已被忽略，因為另一個傳送
受管理傳送中的另一個傳送項目嘗試在限制性傳送沙盤推演外部進行讀取。

如果單一檔案傳送不包含萬用字元，則當傳送涉及的檔案位於沙盤推演外部時，報告的訊息與舊版相同：

由於下列訊息而失敗：BFGIO0056E: 嘗試讀取檔案 "*FILE*" 遭拒絕。
該檔案位於受限傳送沙盤推演外。

相關參考

第 486 頁的『使用 MFT 使用者沙盤推演』
您可以根據要求傳送的 MQMD 使用者名稱，限制可來回傳送檔案的檔案系統區域。

第 485 頁的『使用 MFT 代理程式沙盤推演』
若要對 Managed File Transfer 增加其他安全等級，您可以限制代理程式可存取的檔案系統區域。

MFT agent.properties 檔案

配置 MFT 的 SSL 或 TLS 加密

您可以搭配使用 SSL 或 TLS 與 IBM MQ Managed File Transfer，以保護代理程式與其代理程式佇列管理程式、指令及其所連接的佇列管理程式，以及拓蹼內各種佇列管理程式至佇列管理程式連線之間的通訊安全。

開始之前

您可以使用 SSL 或 TLS 加密來加密流經 IBM MQ Managed File Transfer 拓蹼的訊息。 其中包括：

- 在代理程式與其代理程式佇列管理程式之間傳遞的訊息。
- 指令及其所連接之佇列管理程式的訊息。
- 在拓蹼內的代理程式佇列管理程式、指令佇列管理程式及協調佇列管理程式之間流動的內部訊息。

關於這項作業

如需搭配使用 SSL 與 IBM MQ 的一般資訊，請參閱 第 230 頁的『使用 SSL/TLS』。 就 IBM MQ 而言，Managed File Transfer 是標準 Java 用戶端應用程式。

遵循這些步驟對 Managed File Transfer 使用 SSL：

程序

1. 建立信任儲存庫檔案及（選擇性地）金鑰儲存庫檔（這些檔案可以是相同檔案）。 如果不需要用戶端鑑別（亦即，通道上的 SSLCAUTH=OPTIONAL），則不需要提供金鑰儲存庫。 您只需要信任儲存庫來鑑別佇列管理程式的憑證。

   用於為信任儲存庫和金鑰儲存庫建立憑證的金鑰演算法必須是 RSA，才能使用 IBM MQ。
2. 設定 IBM MQ 佇列管理程式來使用 SSL。

   如需利用「IBM MQ Explorer」設定佇列管理程式以使用 SSL 的相關資訊，請參閱在佇列管理程式上配置 SSL。
3. 將信任儲存庫檔案及金鑰儲存庫檔（如果有的話）儲存在適當位置。 建議位置是 *config_directory*/*coordination_qmgr*/agents/*agent_name* 目錄。
4. 在適當的 Managed File Transfer 內容檔中設定每一個已啟用 SSL 佇列管理程式所需的 SSL 內容。 每一組內容各自參照個別的佇列管理程式（代理程式、協調及指令），但一個佇列管理程式可能扮演其中兩個以上的角色。

   需要 **CipherSpec** 或 **CipherSuite** 內容其中一個，否則用戶端會嘗試不使用 SSL 而直接連接。 同時提供 **CipherSpec** 或 **CipherSuite** 內容，因為 IBM MQ 與 Java 之間的術語差異。 Managed File Transfer 會接受任一內容並執行必要的轉換，所以您不需要同時設定這兩個內容。 如果同時指定 **CipherSpec** 或 **CipherSuite** 內容，則優先採用 **CipherSpec**。

   PeerName 是選用內容。 此內容可設為您想要連接的佇列管理程式的「識別名稱」。 Managed File Transfer 會拒絕連線至「識別名稱」不符的不正確 SSL 伺服器。

   將 **SslTrustStore** 及 **SslKeyStore** 內容設為指向信任儲存庫及金鑰儲存庫檔的檔案名稱。 如果是針對已執行的代理程式設定這些內容，請停止並重新啟動代理程式，並以 SSL 模式重新連接。

   內容檔包含純文字密碼，請考慮設定適當的檔案系統權限。

   如需 SSL 內容的相關資訊，請參閱 第 491 頁的『MFT 的 SSL/TLS 內容』。
5. 如果代理程式佇列管理程式使用 SSL，則建立代理程式時您無法提供必要的詳細資料。 請使用下列步驟來建立代理程式：

   a) 使用 **fteCreateAgent** 指令建立代理程式。 您會收到警告，表示無法將代理程式存在的事實發佈至代理程式協調佇列管理程式。

b) 編輯前一個步驟所建立的 agent.properties 檔來新增 SSL 資訊。 代理程式順利啟動後，即再次嘗試發佈。

6. 如果在變更 agent.properties 檔案或 coordination.properties 檔案中的 SSL 內容時，正在執行「IBM MQ 探險家」的代理程式或實例，則必須重新啟動代理程式或 IBM MQ Explorer。

相關參考

MFT agent.properties 檔案

MFT 的 SSL/TLS 內容

部分 MFT 內容檔包括 SSL 及 TLS 內容。 您可以搭配使用 SSL 或 TLS 與 IBM MQ 及 Managed File Transfer，以防止代理程式與佇列管理程式之間的未獲授權連線，以及加密代理程式與佇列管理程式之間的訊息資料流量。

下列 MFT 內容檔包括 SSL 內容：

- MFT agent.properties 檔案的 SSL/TLS 內容
- MFT coordination.properties 檔案的 SSL/TLS 內容
- MFT command.properties 檔案的 SSL/TLS 內容
- MFT logger.properties 檔案的 SSL/TLS 內容

如需搭配使用 SSL 或 TLS 與 Managed File Transfer 的相關資訊，請參閱 配置 MFT 的 SSL 或 TLS 加密。

從 IBM WebSphere MQ 7.5 開始，您可以在代表檔案或目錄位置的部分 Managed File Transfer 內容中使用環境變數。 這可讓執行產品的組件時所使用的檔案或目錄，隨著環境變更（例如執行程序的使用者為何）而改變其所在位置。 如需相關資訊，請參閱 在 MFT 內容中使用環境變數。

在用戶端模式下使用通道鑑別連接至佇列管理程式

IBM WebSphere MQ 7.1 推出了通道鑑別記錄功能，可更精確地控制通道層次上的存取。 這項行為變更意味著新建立的 IBM WebSphere MQ 7.1 版或更新版本佇列管理程式，依預設會拒絕來自 Managed File Transfer 元件的用戶端連線。

如需通道鑑別的相關資訊，請參閱 第 40 頁的『通道鑑別記錄』。

如果 Managed File Transfer 所使用的 SVRCONN 的通道鑑別配置指定了非特許 MCAUSER ID，您必須授與佇列管理程式、佇列及主題的特定權限記錄，以讓 Managed File Transfer Agent 及指令正常運作。 請使用 MQSC 指令 SET CHLAUTH 或 PCF 指令設定通道鑑別記錄，來建立、修改或移除通道鑑別記錄。 對於您要連接至 IBM WebSphere MQ 7.1 或更新版本佇列管理程式的所有 Managed File Transfer 代理程式，您可以設定要用於所有代理程式的 MCAUSER ID，或為每一個代理程式設定個別的 MCAUSER ID。

請為每一個 MCAUSER ID 授與下列權限：

- 佇列管理程式所需的權限記錄：
  - 連接
  - setid
  - inq
- 佇列所需的權限記錄。

  對於所有代理程式特定佇列（即下列清單中以 *agent_name* 結尾的佇列名稱），您必須為要使用用戶端連線連接至 IBM WebSphere MQ 7.1 版或更新版本佇列管理程式的每個代理程式，建立這些佇列權限記錄。
  - put, get, dsp (SYSTEM.DEFAULT.MODEL.QUEUE)
  - put, get, setid, browse (SYSTEM.FTE.COMMAND.*agent_name*)
  - put, get (SYSTEM.FTE.DATA.*agent_name*)
  - put, get (SYSTEM.FTE.REPLY.*agent_name*)
  - put, get, inq, browse (SYSTEM.FTE.STATE.*agent_name*)
  - put, get, browse (SYSTEM.FTE.EVENT.*agent_name*)
  - put, get (SYSTEM.FTE)

- 主題所需的權限記錄：
  - sub, pub (SYSTEM.FTE)
- 檔案傳送所需的權限記錄。

  如果來源及目的地代理程式具有個別的 MCAUSER ID，請在來源及目的地代理程式佇列中建立權限記錄。

  例如，如果來源代理程式的 MCAUSER ID 為 **user1**，目的地代理程式 MCAUSER ID 為 **user2**，請為代理程式使用者設定下列權限：

代理程式使用者	佇列	所需的權限
user1	SYSTEM.FTE.DATA.*destination_agent_name*	放置
user1	SYSTEM.FTE.COMMAND.*destination_agent_name*	放置
user2	SYSTEM.FTE.REPLY.*source_agent_name*	放置
user2	SYSTEM.FTE.COMMAND.*source_agent_name*	放置

在 Connect:Direct 橋接器代理程式與 Connect:Direct 節點之間配置 SSL 或 TLS

透過建立金鑰儲存庫與信任儲存庫，以及設定 Connect:Direct 橋接器代理程式內容檔中的內容，即可將 Connect:Direct 橋接器代理程式與 Connect:Direct 節點，配置為透過 SSL 通訊協定彼此連接。

關於這項作業

下列步驟包含一些指示來取得透過憑證管理中心簽署的金鑰。 如果您不使用憑證管理中心，則可產生自簽憑證。 如需產生自簽憑證的相關資訊，請參閱 第 240 頁的『在 AIX, Linux, and Windows 上使用 SSL/TLS』。

下列步驟包含為 Connect:Direct 橋接器代理程式，建立新的金鑰儲存庫及信任儲存庫的指示。 如果 Connect:Direct 橋接器代理程式，已經具有用於安全連接至 IBM MQ 佇列管理程式的金鑰儲存庫及信任儲存庫，則可在安全連接至 Connect:Direct 節點時使用現有的金鑰儲存庫及信任儲存庫。 如需相關資訊，請參閱 第 490 頁的『配置 MFT 的 SSL 或 TLS 加密』。

程序

對於 Connect:Direct 節點，請完成下列步驟：

1. 產生用於 Connect:Direct 節點的金鑰及已簽章的憑證。

   您可以使用 IBM MQ 隨附的 IBM Key Management 工具來執行此動作。 如需相關資訊，請參閱 第 230 頁的『使用 SSL/TLS』。
2. 向憑證管理中心傳送簽署金鑰的要求。 您會收到回報的憑證。
3. 建立包含憑證管理中心公開金鑰的文字檔，例如 `/test/ssl/certs/CAcert`。
4. 在 Connect:Direct 節點上安裝 Secure+ Option。

   如果該節點已經存在，您可以透過下列方式來安裝 Secure+ Option：再次執行安裝程式，指定現有安裝的位置，然後選擇僅安裝 Secure+ Option。
5. 建立新的文字檔，例如 `/test/ssl/cd/keyCertFile/`*node_name*`.txt`。
6. 將您從憑證管理中心收到的憑證，以及位於 `/test/ssl/cd/privateKeys/`*node_name*`.key` 中的私密金鑰，複製到該文字檔中。

   `/test/ssl/cd/keyCertFile/`*node_name*`.txt` 的內容必須採用下列格式：

```
-----BEGIN CERTIFICATE-----
MIICnzCCAgigAwIBAgIBGjANBgkqhkiG9w0BAQUFADBeMQswCQYDVQQGEwJHQjES
MBAGA1UECBMJSGFtcHNoaXJlMRAwDgYDVQQHEwdIdXJzbGV5MQwwCgYDVQQKEwNJ
Qk0xDjAMBgNVBAsTBU1RSVBUMQswCQYDVQQDEwJDQTAeFw0xMTAzMDExNjIwNDZa
Fw0yMTAyMjYxNjIwNDZaMFAxCzAJBgNVBAYTAkdCMRIwEAYDVQQIEwlIYW1wc2hp
cmUxDDAKBgNVBAoTA0lCTTEOMAwGA1UECxMFTVFGVEUxDzANBgNVBAMTBmJpbmJh
ZzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAvgP1QIklU9ypSKD1XoODo1yk
EyMFXBOUpZRrDVxjoSEC0vtWNcJ199e+Vc4UpNybDyBu+NkDlMNofX4QxeQcLAFj
WnhakqCiQ+JIAD5AurhnrwChe0MV3kjA84GKH/rOSVqtl984mu/lDyS819XcfSSn
cOOMsK1KbneVSCIV2XECAwEAAaN7MHkwCQYDVR0TBAIwADAsBglghkgBhvhCAQ0E
HxYdT3BlblNTTCBHZW5lcmF0ZWQgQ2VydGlmaWNhdGUwHQYDVR0OBBYEFNXMIpSc
```

```
csBXUniW4A3UrZnCRsv3MB8GA1UdIwQYMBaAFDXY8rmj4lVz5+FVAoQb++cns+B4
MA0GCSqGSIb3DQEBBQUAA4GBAFc7klXa4pGKYgwchxKpE3ZF6FNwy4vBXS216/ja
8h/vl8+iv01OCL8t0ZOKSU95fyZLzOPKnCH7v+ItFSE3CIiEk9Dlz2U6WO9lICwn
l7PL72TdfaL3kabwHYVf17IVcuL+VZsZ3HjLggP2qHO9ZuJPspeT9+AxFVMLiaAb
8eHw
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,64A02DA15B6B6EF9

57kqxLOJ/gRUOIQ6hVK2YN13B4E1jAi1gSme0I5ZpEIG8CHXISKB7/0cke2FTqsV
lvI99QyCxsDWoMNt5fj51v7aPmVeS60bOm+UlGre8B/Ze18JVj2O4K2Uh72rDCXE
5e6eFxSdUM207sQDy20euBVELJtM2kOkL1ROdoQQSlU3XQNgJw/t3ZIx5hPXWEQT
rjRQO64BEhb+PzzxPF8uwzZ9IrUK9BJ/UUnqC6OdBR87IeA4pnJD1Jvb2ML7EN9Z
5Y+50hTKI8OGvBvWXO4fHyvIX5aslwhBoArXIS1AtNTrptPvoaP1zyIAeZ6OCVo/
SFo+A2UhmtEJeOJaZG2XZ3H495fAw/EHmjehzIACwukQ9nSIETgu4A1+CV64RJED
aYBCM8UjaAkbZDH5gn7+eBov0ssXAXWDyJBVhUOjXjvAj/e1h+kcSF1hax5D//AI
66nRMZzboSxNqkjcVd8wfDwP+bEjDzUaaarJTS7lIFeLLw7eJ8MNAkMGicDkycL0
EPBU9X5QnHKLKOfYHN/1WgUk8qt3UytFXXfzTXGF3EbsWbBupkT5e5+lYcX8OVZ6
sHFPNlHluCNy/riUcBy9iviVeodX8IomOchSyO5DKl8bwZNjYtUP+CtYHNFU5BaD
I+1uUOAeJ+wjQYKT1WaeIGZ3VxuNITJul8y5qDTXXfX7vxM5OoWXa6U5+AYuGUMg
/itPZmUmNrHjTk7ghT6i1IQOaBowXXKJBlMmq/6BQXN2IhkD9ys2qrvM1hdi5nAf
egmdiG50loLnBRqWbfR+DykpAhK4SaDi2F52Uxovw3Lhiw8dQP7lzQ==
-----END RSA PRIVATE KEY-----
```

7. 啟動 Secure+ Admin Tool。
   - 在 AIX and Linux 系統上，執行 **spadmin.sh** 指令。
   - 在 Windows 系統上，按一下**開始** > **程式集** > **Sterling Commerce Connect:Direct** > **CD Secure+ Admin Tool**

   CD Secure+ Admin Tool 即會啟動。
8. 在 CD Secure+ Admin Tool 中，按兩下 **.Local** 行以編輯主要 SSL 或 TLS 設定。
   a) 視您使用的通訊協定而定，選取**啟用 SSL 通訊協定**或**啟用 TLS 通訊協定**。
   b) 選取**停用置換**。
   c) 至少選取一個密碼組合。
   d) 如果您想要雙向鑑別，請將 **啟用用戶端鑑別** 的值變更為 Yes。
   e) 在**授信主要憑證**欄位中，輸入憑證管理中心的公開憑證檔路徑 (/test/ssl/certs/CAcert)。
   f) 在**金鑰憑證檔**欄位中，輸入您所建立的檔案路徑 (/test/ssl/cd/keyCertFile/*node_name*.txt)。
9. 按兩下 **.Client** 行以編輯主要 SSL 或 TLS 設定。
   a) 視您使用的通訊協定而定，選取**啟用 SSL 通訊協定**或**啟用 TLS 通訊協定**。
   b) 選取**停用置換**。

對於 Connect:Direct 橋接器代理程式，請執行下列步驟：

10. 建立信任儲存庫。 您可以透過先建立後刪除虛擬金鑰來執行此作業。
    您可以使用下列指令：

```
keytool -genkey -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

```
keytool -delete -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

11. 將憑證管理中心的公開憑證匯入信任儲存庫。
    您可以使用下列指令：

```
keytool -import -trustcacerts -alias myCA
        -file /test/ssl/certs/CAcert
        -keystore /test/ssl/fte/stores/truststore.jks
```

12. 編輯 Connect:Direct 橋接器代理程式內容檔。
    將下列行併入檔案中的任何位置：

```
cdNodeProtocol=protocol
cdNodeTruststore=/test/ssl/fte/stores/truststore.jks
cdNodeTruststorePassword=password
```

在此步驟的範例中，*protocol* 是您使用的通訊協定（SSL 或 TLS），*password* 是您在建立信任儲存庫時指定的密碼。

13. 如果要進行雙向鑑別，請為 Connect:Direct 橋接器代理程式建立金鑰及憑證。
    a) 建立金鑰儲存庫及金鑰。
    您可以使用下列指令：

```
keytool -genkey -keyalg RSA -alias agent_name
        -keystore /test/ssl/fte/stores/keystore.jks
        -storepass password -validity 365
```

    b) 產生簽署要求。
    您可以使用下列指令：

```
keytool -certreq -v -alias agent_name
        -keystore /test/ssl/fte/stores/keystore.jks -storepass password
        -file /test/ssl/fte/requests/agent_name.request
```

    c) 將您從之前步驟中收到的憑證匯入金鑰儲存庫。 此憑證必須採用 x.509 格式。
    您可以使用下列指令：

```
keytool -import -keystore /test/ssl/fte/stores/keystore.jks
        -storepass password -file certificate_file_path
```

    d) 編輯 Connect:Direct 橋接器代理程式內容檔。
    將下列行併入檔案中的任何位置：

```
cdNodeKeystore=/test/ssl/fte/stores/keystore.jks
cdNodeKeystorePassword=password
```

    在此步驟的範例中，*password* 是您在建立金鑰儲存庫時指定的密碼。

相關工作

配置 Connect:Direct 橋接器

ALW 保護 AMQP 用戶端安全

您可以使用一系列安全機制來保護來自 AMQP 用戶端的連線安全，並確保網路上的資料受到適當保護。 您可以在 MQ Light 應用程式中建置安全。 您也可以使用 IBM MQ 的現有安全特性與 AMQP 用戶端搭配使用，方法與這些特性用於其他應用程式的方式相同。

通道鑑別規則 (CHLAUTH)

您可以使用通道鑑別規則來限制與佇列管理程式的 TCP 連線。 AMQP 通道支援使用您為佇列管理程式配置的通道鑑別規則。 如果使用符合佇列管理程式上任何 AMQP 通道的設定檔來定義通道鑑別規則，則這些規則會套用至那些通道。 依預設，會在新的 IBM® MQ 佇列管理程式上啟用通道鑑別，因此您必須至少完成部分配置，才能使用 AMQP 通道。

如需如何配置通道鑑別規則以容許 AMQP 連線至佇列管理程式的相關資訊，請參閱 建立及使用 AMQP 通道。

連線鑑別 (CONNAUTH)

您可以使用連線鑑別來鑑別與佇列管理程式的連線。 AMQP 通道支援使用連線鑑別來控制從 AMQP 應用程式對佇列管理程式的存取權。

AMQP 通訊協定使用 SASL (簡易鑑別及安全層) 架構來指定如何鑑別連線。 有各種 SASL 機制，且 IBM MQ 支援兩種 SASL 機制 :ANONYMOUS 和 PLAIN。

如果是 ANONYMOUS，則不會將任何認證從用戶端傳遞至佇列管理程式以進行鑑別。 如果 CONNAUTH 屬性中指定的 MQ AUTHINFO 物件具有 REQUIRED 或 REQDADM (如果以管理使用者身分連接) 的 CHCKCLNT 值，則會拒絕連線。 如果 CHCKCLNT 的值是 NONE 或 OPTIONAL，則會接受連線。

若為 PLAIN，則會將使用者名稱及密碼從用戶端傳遞至佇列管理程式以進行鑑別。 如果 CONNAUTH 屬性中指定的 MQ AUTHINFO 物件具有 CHCKCLNT 值 NONE，則會拒絕連線。 如果 CHCKCLNT 的值是 OPTIONAL、REQUIRED 或 REQDADM (如果以管理使用者身分連接)，則佇列管理程式會檢查使用者名稱及密碼。 佇列管理程式會檢查作業系統 (如果 AUTHINFO 物件是 IDPWOS 類型) 或 LDAP 儲存庫 (如果 AUTHINFO 物件是 IDPWLDAP 類型)。

下表彙總此鑑別行為:

表 *99: SASL* 機制和連線鑑別的摘要

SASL 機制	**從用戶端傳遞至佇列管理程式的認證?**	**CHCKCLNT 值**
匿名	否	REQUIRED 或 REQDADM-拒絕連線 NONE 或 OPTIONAL-接受連線
PLAIN	是，使用者名稱和密碼	REQUIRED、REQDADM 或 OPTIONAL-佇列管理程式所檢查的使用者名稱和密碼 NONE-拒絕連線

如果您是使用 MQ Light 用戶端，則可以透過將認證包含在您所連接的 AMQP 位址中來指定認證，例如:

```
amqp://mwhitehead:mYp4ssw0rd@localhost:5672/sports/football
```

通道上的 MCAUSER 設定

AMQP 通道具有 MCAUSER 屬性，您可以用來設定對該通道的所有連線進行授權的 IBM MQ 使用者 ID。 從 AMQP 用戶端到該通道的所有連線都採用您已配置的 MCAUSER ID。 該使用者 ID 用於不同主題的傳訊授權。

建議您使用通道鑑別 (CHLAUTH) 來保護佇列管理程式的連線安全。 如果您使用通道鑑別，建議您將 MCAUSER 的值配置給非特許使用者。 這可確保如果 CHLAUTH 規則不符合通道連線，則連線未獲授權在佇列管理程式上執行任何傳訊。

註: Windows 在 Windows 上，在 IBM MQ 9.1.1 之前，最多只有 12 個字元的使用者 ID 才支援 MCAUSER 使用者 ID 設定。 V 9.2.0 從 IBM MQ 9.1.1 Continuous Delivery，以及從 IBM MQ 9.2.0 Long Term Support，移除 12 個字元的限制。

SSL/TLS 支援

AMQP 通道支援使用為佇列管理程式配置之金鑰儲存庫中的金鑰進行 SSL/TLS 加密。 SSL/TLS 加密的 AMQP 通道配置選項支援與其他類型 MQ 通道相同的選項; 您可以指定密碼規格，以及佇列管理程式是否需要來自 AMQP 用戶端連線的憑證。

透過使用佇列管理程式的 FIPS 屬性，您可以控制 SSL/TLS 密碼組合，可用來保護來自 AMQP 用戶端的連線安全。

如需如何為佇列管理程式設定金鑰儲存庫的相關資訊，請參閱 在 UNIX、 Linux 及 Windows 系統上使用 SSL 或 TLS。

如需如何為 AMQP 用戶端連線配置 SSL/TLS 支援的相關資訊，請參閱 建立及使用 AMQP 通道。

Java 鑑別和授權服務 (Java Authentication and Authorization Service, JAAS)

您可以選擇性地使用 JAAS 登入模組來配置 AMQP 通道，該模組可以檢查 AMQP 用戶端提供的使用者名稱及密碼。 請參閱 第 496 頁的『配置 AMQP 通道的 JAAS』。

相關工作

開發 AMQP 用戶端應用程式

建立及使用 AMQP 通道

ALW 限制 AMQP 用戶端接管

建立與現有 AMQP 用戶端連線具有相同用戶端 ID 的 AMQP 用戶端連線時，依預設會中斷現有用戶端連線。不過，您可以配置佇列管理程式來限制用戶端接管行為，以便只有在符合特定準則時才能進行接管。

例如，如果有不同團隊正在開發的 AMQP 應用程式，且它們碰巧使用相同的用戶端 ID，則中斷現有用戶端連線可能不適用。 若要解決此問題，您可以根據所使用的 AMQP 通道名稱、用戶端的 IP 位址及用戶端使用者 ID (啟用 SASL 鑑別時) 來限制用戶端接管。

使用佇列管理程式屬性 **AdoptNewMCA** 及 **AdoptNewMCACheck** 的設定，來指定用戶端接管限制的必要層次，如下表中所詳述:

表 *100:* 限制用戶端接管的 ***AdoptNewMCA*** 及 ***AdoptNewMCACheck*** 設定

AdoptNewMCA	**AdoptNewMCACheck**	**在容許用戶端接管之前檢查準則**
否或未定義	不適用	無。 對於已鑑別並傳遞所有 CHLAUTH 規則的所有用戶端連線，容許用戶端接管。
ALL (或 NO 以外的值)	QM 或未定義	無。 對於已鑑別並傳遞所有 CHLAUTH 規則的所有用戶端連線，容許用戶端接管。
ALL (或 NO 以外的值)	名稱	使用者 ID (啟用 SASL 時) 通道名稱
ALL (或 NO 以外的值)	ADDRESS	使用者 ID (啟用 SASL 時) IP 位址
ALL (或 NO 以外的值)	ALL	使用者 ID (啟用 SASL 時) 通道名稱 IP 位址

佇列管理程式屬性 **AdoptNewMCA** 及 **AdoptNewMCACheck** 是佇列管理程式配置的一部分，定義在 CHANNELS 段落中。 在 IBM MQ for Windows 及 IBM MQ for Linux x86-64 系統上，請使用 IBM MQ Explorer 來修改配置資訊。 在其他系統上，透過編輯 `qm.ini` 配置檔來修改資訊。 如需如何修改佇列管理程式通道資訊的相關資訊，請參閱 通道屬性。

相關工作

開發 AMQP 用戶端應用程式

建立及使用 AMQP 通道

ALW 配置 AMQP 通道的 JAAS

Java 鑑別和授權服務 (JAAS) 自訂模組可用來鑑別 AMQP 用戶端在連接時傳遞給 AMQP 通道的使用者名稱和密碼認證。

關於這項作業

如果您已在其他 Java 型系統中使用 JAAS 模組進行鑑別，且想要重複使用那些模組來鑑別 MQ 的 AMQP 連線，則您可能想要使用自訂 JAAS 模組。 或者，如果 MQ 中內建的鑑別特性不支援您要使用的鑑別機制，您可能想要撰寫自訂 JAAS 模組。

AMQP 通道的 JAAS 模組配置是在佇列管理程式層次完成。 這表示如果您配置 JAAS 模組來鑑別佇列管理程式的 AMQP 連線，該模組將套用至所有 AMQP 通道。 已呼叫 JAAS 模組的通道名稱會傳給模組，可讓您撰寫不同通道的不同 JAAS 登入行為。

也會傳遞其他資訊給 JAAS 模組:

- 嘗試鑑別之 AMQP 用戶端的用戶端 ID。
- AMQP 用戶端的網址。
- 呼叫 JAAS 模組的通道名稱。

程序

您可以完成下列步驟來配置 AMQP 通道的 JAAS 配置模組:

1. 定義包含一或多個 JAAS 模組配置段落的 `jaas.config` 檔案。 段落必須指定實作 JAAS `javax.security.auth.spi.LoginModule` 介面之 Java 類別的完整名稱。
   - 產品隨附預設 `jaas.config` 檔案，位於 *QM_data_directory*`/amqp/jaas.config` 中。
   - 名稱為 `MQXRConfig` 的預先配置段落已定義在預設 `jaas.config` 檔案中。
2. 指定用於 AMQP 通道的段落名稱。
   - Linux AIX 將內容新增至 `amqp_unix.properties` 檔案。
   - Windows 將內容新增至 `amqp_win.properties` 檔案。

   內容具有下列格式:

   ```
   com.ibm.mq.MQXR.JAASConfig=JAAS_stanza_name
   ```

   例如:

   ```
   com.ibm.mq.MQXR.JAASConfig=MQXRConfig
   ```
3. 將佇列管理程式環境配置成包含自訂模組的類別。 AMQP 服務必須具有 JAAS 配置段落中所配置 Java 類別的存取權。

   作法是將 JAAS 類別的路徑新增至 MQ `service.env` 檔。 編輯 MQ 配置目錄 (*MQ_config_directory*) 或佇列管理程式配置目錄 (*QM_config_directory*) 中的 `service.env` 檔，將 CLASSPATH 變數設為 JAAS 模組類別的位置。

下一步

mq_installation_directory`/amqp/samples` 目錄中的產品隨附範例 JAAS 登入模組。 不論用戶端用來連接的使用者名稱或密碼為何，範例 JAAS 登入模組都會鑑別所有用戶端連線。

您可以修改範例的原始碼並重新編譯它，以嘗試只鑑別具有特定密碼的特定使用者。 若要在 UNIX 系統上配置 AMQP 通道，以使用產品隨附的範例 JAAS 登入模組:

1. 編輯檔案 `/var/mqm/qmgrs/QMNAME/amqp/amqp_unix.properties`，並設定內容 `com.ibm.mq.MQXR.JAASConfig=MQXRConfig`。
2. 編輯 /var/mqm/service.env 檔案，並設定內容 CLASSPATH=*mq_installation_location*`/amqp/samples`

`jaas.config` 檔案已包含名為 `MQXRConfig` 的段落，該段落指定範例類別 `samples.JAASLoginModule` 作為登入模組類別。 在您嘗試範例模組之前，不需要對 `jaas.config` 進行任何變更。

相關工作

開發 AMQP 用戶端應用程式

建立及使用 AMQP 通道

Advanced Message Security

Advanced Message Security (AMS) 是 IBM MQ 的元件，可為流經 IBM MQ 網路的機密資料提供高階保護，同時不會影響一般應用程式。

Advanced Message Security 的概觀

IBM MQ 應用程式可以使用 Advanced Message Security，透過使用公開金鑰加密法模型，以不同層次的保護來傳送機密資料 (例如高價值金融交易及個人資訊)。

相關概念

第 540 頁的『訊息通道代理程式 (MCA) 截取及 AMS』

MCA 截取可讓在 IBM MQ 下執行的佇列管理程式選擇性地啟用要針對伺服器連線通道套用的原則。

相關參考

AMS 訊息中使用的 GSKit 回覆碼

Advanced Message Security 的特性及功能

Advanced Message Security 擴充 IBM MQ 安全服務，以提供訊息層次的資料簽署及加密。 展開的服務可保證在訊息資料最初放置在佇列上與擷取時之間，不會修改訊息資料。 此外， AMS 還會驗證訊息資料的傳送端是否已獲授權將已簽署的訊息放置在目標佇列上。

AMS 提供下列函數：

- 保護 IBM MQ 所處理的機密或高價值交易。
- 在接收端應用程式處理惡意或未獲授權的訊息之前，先偵測並移除它們。
- 驗證在從佇列到佇列的傳輸期間未修改訊息。
- 不僅在資料流經網路時，而且在將資料放入佇列時，也會保護資料。
- 保護 IBM MQ 的現有專有及客戶撰寫應用程式。
- z/OS V 9.2.0 從 IBM MQ 9.1.3 開始， IBM MQ for z/OS 可讓您選擇性地從流經網路的訊息中移除及新增 AMS 保護，或對這些訊息分別新增 AMS 保護。 這稱為 伺服器至伺服器訊息通道代理程式 *(MCA)* 截取。
- ALW 從 IBM MQ 9.1.4 和 IBM MQ 9.1.0 Fix Pack 4 開始，會將檢查新增至在客戶應用程式內執行的 IBM MQ 程式庫程式碼。 檢查會在其起始設定的早期執行，以讀取環境變數 *AMQ_AMS_FIPS_OFF* 的值，如果設為任何值，則會在該應用程式中以非 FIPS 模式執行 IBM Global Security Kit (GSKit) 程式碼。

AMS 提供的保護品質

Advanced Message Security、 `Integrity`、 `Privacy` 和 `Confidentiality` 有三種保護品質。

`Integrity` 保護由數位簽章提供，可確保訊息是誰建立的，且訊息未被變更或竄改。

`Privacy` 保護是由數位簽署和加密的組合所提供。 加密可確保只有預期的收件者或收件者可以檢視訊息資料。 即使未獲授權的收件者取得已加密訊息資料的副本，他們也無法檢視實際訊息資料本身。

`Confidentiality` 保護僅透過加密提供，並具有選用性的金鑰重複使用。

對效能的影響

AMS 使用對稱及非對稱加密常式的組合來提供數位簽署及加密。 與 CPU 密集的非對稱金鑰作業相比，對稱金鑰作業非常快速，這反過來會對使用 AMS 保護大量訊息的成本產生重大影響。

非對稱加密常式

例如，在放置已簽署的訊息時，會使用非對稱金鑰作業來簽署訊息雜湊。

取得已簽署訊息時，會使用進一步的非對稱金鑰作業來驗證已簽署的雜湊。

因此，每個訊息至少需要兩個非對稱金鑰作業來簽署及驗證訊息資料。

非對稱及對稱加密常式

在放置加密訊息時，會產生對稱金鑰，然後針對訊息的每個預期收件者使用非對稱金鑰作業進行加密。

然後會使用對稱金鑰來加密訊息資料。 取得加密訊息時，預期的收件者需要使用非對稱金鑰作業來探索用於訊息的對稱金鑰。

因此，所有三種保護品質都包含 CPU 密集非對稱金鑰作業的不同元素，這將顯著影響應用程式放置及取得訊息的可達到傳訊速率上限。

不過， `Confidentiality` 原則容許在一系列訊息上重複使用對稱金鑰。 透過重複使用對稱金鑰，可以使用 `Confidentiality` 原則來節省大量 CPU 成本。 這種作業模式會繼續使用 PKCS#7 格式來共用對稱加密金鑰。 不過，沒有數位簽章，這會刪除個別訊息的部分非對稱金鑰作業。 對於每一個收件者，仍然需要使用非對稱金鑰作業來加密對稱金鑰，但是對稱金鑰可以選擇性地在針對相同收件者的多個訊息上重複使用。 如果原則允許重複使用金鑰，則只有第一個訊息需要非對稱金鑰作業。 後續訊息只需要使用對稱金鑰作業。

重複使用金鑰

使用 `Confidentiality` 原則，您可以使用對稱金鑰重複使用方法，以大幅減少加密放置至相同佇列且預期用於相同收件者或收件者的許多訊息所涉及的成本。

例如，將 10 個已加密訊息放置到同一組收件者時，會產生對稱金鑰，然後針對第一則訊息使用非對稱金鑰作業來加密訊息的每一個預期收件者。

根據原則控制的限制，加密的對稱金鑰隨後可以由預期用於相同收件者的後續訊息重複使用。 若要容許後續訊息重複使用對稱金鑰，在將訊息放入佇列之後，應用程式必須保持佇列開啟。 MQPUT1 作業無法重複使用對稱金鑰。 取得加密訊息的應用程式可以套用相同的最佳化，因為應用程式可以偵測對稱金鑰何時未變更，並避免擷取對稱金鑰的費用。

在此範例中，透過重複使用相同的金鑰來放置及取得應用程式，可避免 90% 的非對稱金鑰作業。

如需如何使用金鑰重複使用的相關資訊，請參閱:

- MQSC 指令 SET POLICY
- 控制指令 setmqspl
- IBM i IBM i 指令 SETMQMSPL

AMS 中的主要概念

瞭解 Advanced Message Security 中的主要概念，以瞭解工具如何運作以及如何有效地管理它。

公開金鑰基礎架構和 *Advanced Message Security*

公開金鑰基礎架構 (PKI) 是支援使用公開金鑰加密法來取得安全通訊的設施、原則及服務系統。

沒有定義公開金鑰基礎架構元件的單一標準，但 PKI 通常涉及公開金鑰憑證的使用，並包含提供下列服務的憑證管理中心 (CA) 及其他註冊管理中心 (RA):

- 發出數位憑證
- 驗證數位憑證
- 撤銷數位憑證
- 配送憑證

在與已簽署或已加密訊息相關聯的憑證中， **識別名稱 (DN)** 欄位代表使用者和應用程式的身分。 Advanced Message Security 使用此身分來代表使用者或應用程式。 若要鑑別此身分，使用者或應用程式必須具有儲存憑證及相關聯私密金鑰之金鑰儲存庫的存取權。 每一個憑證都由金鑰儲存庫中的標籤代表。

相關概念

第 534 頁的『搭配使用金鑰儲存庫和憑證與 AMS』

為了向 IBM MQ 應用程式提供透通加密保護， Advanced Message Security 會使用金鑰儲存庫檔，其中儲存公開金鑰憑證和私密金鑰。 在 z/OS 上，會使用 SAF 金鑰環來取代金鑰儲存庫檔。

AMS 中的數位憑證

Advanced Message Security 會將使用者和應用程式與 X.509 標準數位憑證相關聯。 X.509 憑證通常由授信憑證管理中心 (CA) 簽署，且涉及用於加密及解密的私密及公開金鑰。

數位憑證透過將公開金鑰連結至其擁有者 (無論該擁有者是個人、佇列管理程式或某個其他實體)，來提供避免模擬的保護。 數位憑證也稱為公開金鑰憑證，因為當您使用非對稱金鑰架構時，它們可讓您保證公開金鑰的所有權。 此架構需要為應用程式產生公開金鑰及私密金鑰。 使用公開金鑰加密的資料只能使用對應的私密金鑰解密，而使用私密金鑰加密的資料只能使用對應的公開金鑰解密。 私密金鑰儲存在受密碼保護的金鑰資料庫檔中。 只有其擁有者才能存取用來解密使用對應公開金鑰加密之訊息的私密金鑰。

如果公開金鑰由其擁有者直接傳送至另一個實體，則可能會截取訊息，而公開金鑰會被另一個實體替代。 這被稱為 "中間人" 攻擊。 解決方案是透過具公信力第三者交換公開金鑰，向使用者提供強烈保證公開金鑰屬於您與之通訊的實體。 您不是直接傳送公開金鑰，而是要求具公信力第三者將它納入數位憑證中。 發出數位憑證的授信協力廠商稱為憑證管理中心 (CA)。

如需數位憑證的相關資訊，請參閱 數位憑證中的內容。

數位憑證包含實體的公開金鑰，並指出公開金鑰屬於該實體:

- 當憑證適用於個別實體時，它稱為 個人憑證 或 使用者憑證。
- 當憑證用於憑證管理中心時，該憑證稱為 *CA* 憑證 或 簽章者憑證。

註: Advanced Message Security 同時在 Java 和原生應用程式中支援自簽憑證

相關概念

第 7 頁的『加密法』

加密法是在可讀取文字 (稱為 純文字) 與無法讀取格式 (稱為 密文) 之間進行轉換的程序。

Multi 物件權限管理程式及 *AMS*

在 Multiplatforms 上，「物件權限管理程式 (OAM)」是隨 IBM MQ 產品提供的授權服務元件。

對 Advanced Message Security 實體的存取權是透過 IBM MQ 使用者群組及 OAM 來控制。 管理者可以視需要使用指令行介面來授與或撤銷授權。 不同使用者群組可以對相同物件具有不同類型的存取權。 例如，一個群組可以針對特定佇列執行 PUT 及 GET 作業，而另一個群組只能瀏覽佇列。 同樣地，部分群組可能具有佇列的 GET 及 PUT 權限，但不容許變更或刪除佇列。

透過 OAM，您可以控制:

- 透過「訊息佇列介面 (MQI)」存取 Advanced Message Security 物件。 當應用程式嘗試存取物件時， OAM 會檢查提出要求的使用者設定檔是否具有所要求作業的授權。 這表示佇列及佇列上的訊息可以受到保護，不會遭到未獲授權的存取。
- 使用 PCF 及 MQSC 指令的許可權。

相關概念

物件權限管理程式

訊息佇列介面概觀

Advanced Message Security 支援的技術

Advanced Message Security 依賴數個技術元件來提供安全基礎架構。

Advanced Message Security 支援下列 IBM MQ 應用程式設計介面 (API):

- 訊息佇列介面 (MQI)
- IBM MQ Java Message Service (JMS) 1.0.2 和 1.1。
- IBM MQ Java 的基礎類別
- 未受管理模式中 .Net 的 IBM MQ 類別

註: Advanced Message Security 支援 X.509 相容憑證管理中心。

AMS 的已知限制

有許多 IBM MQ 選項要么不受支持，要么對 Advanced Message Security (AMS) 有限制。

- 下列 IBM MQ 選項不受支援或有限制:

  發佈/訂閱
  發佈/訂閱傳訊模型透過點對點的其中一個主要好處是傳送及接收應用程式不需要彼此瞭解任何相關資訊，即可傳送及接收資料。 使用必須定義預期收件者或授權簽章者的 Advanced Message Security 原則會使此權益無效。 應用程式可以透過受原則保護的別名佇列定義發佈至主題，訂閱應用程式也可以從受原則保護的佇列取得訊息。 無法將原則直接指派給主題字串，原則只能指派給佇列定義。

  通道資料轉換
  Advanced Message Security 受保護訊息的受保護有效負載是使用二進位格式來傳輸，這可確保應用程式之間通道上的資料轉換不會使訊息摘要失效。 從原則受保護佇列擷取訊息的應用程式應該要求資料轉換，在順利驗證及未受保護的訊息之後，將會嘗試轉換受保護的有效負載。

  分送清單
  在保護應用程式將訊息放入配送清單時，可以使用 Advanced Message Security 原則，前提是清單中的每一個目的地佇列都已定義相同的原則。 當應用程式開啟配送清單時，如果識別出不一致的原則，則開啟作業會失敗，且會傳回安全錯誤給應用程式。

  應用程式訊息分段
  受原則保護訊息的大小將會增加，應用程式無法精確指定訊息的區段界限。

  在受管理模式中使用 IBM MQ classes for .NET 的應用程式 (用戶端連線)
  不支援在受管理模式 (用戶端連線) 中使用 IBM MQ classes for .NET 的應用程式。

  註: MCA 截取可用來容許不受支援的用戶端使用 AMS。

  受管理模式中 .NET (XMS) 應用程式的訊息服務用戶端
  不支援受管理模式中的 .NET (XMS) 應用程式的訊息服務用戶端。

  註: MCA 截取可用來容許不受支援的用戶端使用 AMS。

  IMS 橋接器處理的 IBM MQ 佇列
  不支援 IMS 橋接器所處理的 IBM MQ 佇列。

  註: AMS 在 CICS 橋接器佇列上受支援。 您應該對 CICS 橋接器佇列上的 MQPUT (encrypt) 及 MQGET (decrypt) 使用相同的使用者 ID。

  放置到等待中的 getter
  針對已定義 AMS 原則的佇列，不支援將 getter 應用程式放置在等待中的 getter。

  V 9.2.0 **伺服器至伺服器 MCA 截取**
  從 IBM MQ for z/OS 9.1.3 開始，只有傳送端、伺服器、接收端及要求端通道類型才支援伺服器至伺服器 MCA 截取。

- 使用者應該避免在單一金鑰儲存庫檔中放置多個具有相同「識別名稱」的憑證，因為未定義保護訊息時要使用的憑證。
- 如果 **WMQ_PROVIDER_VERSION** 內容設為 6 ，則 JMS 中不支援 AMS 。
- AMQP 或 MQTT 通道不支援 AMS 攔截程式。

z/OS V 9.2.0 訊息通道上的 Advanced Message Security 截取

在 z/OS 上， Advanced Message Security (AMS) 截取為傳送端、伺服器、接收端及要求端通道提供安全原則保護 (SPLPROT) 的其他選項，可讓您支援 AMS 並與不支援 AMS 的事業夥伴通訊。

以與銀行通訊的結算所為例， 圖 1 顯示在沒有 AMS 攔截的情況下，系統兩端都需要支援 AMS。

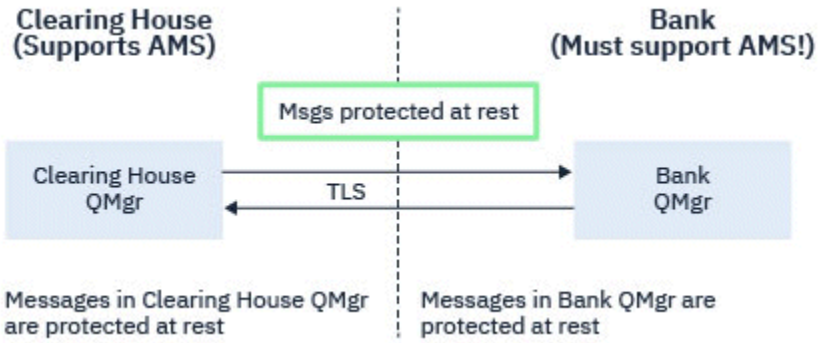


圖 *32:* 使用無 *AMS* 截取的 *AMS*

AMS 截取選項的主要好處是，如果您的企業已 AMS 配置，且並非所有事業夥伴都支援 AMS，則可以移除對出埠訊息的保護，並保護與不支援 AMS 的那些事業夥伴之間的通道上的入埠訊息。

使用結算所及銀行的範例，此實務範例顯示在 圖 2 中，其中在結算所、銀行及商業夥伴之間有訊息流程，其中部分機構具有 AMS，而其他機構則沒有。

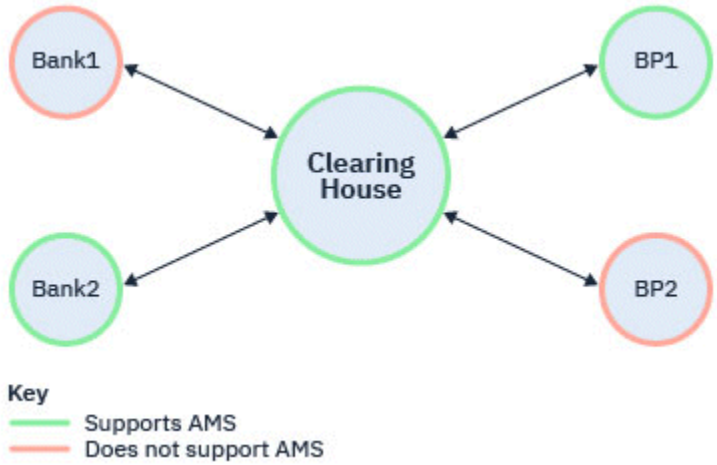


圖 *33:* 有些夥伴支援 *AMS* ，有些則不支援 *AMS*

通道通常已啟用 TLS。

不過，有些銀行和事業夥伴可能不支援 AMS，且需要能夠在所有銀行和事業夥伴之間交換訊息。 此實務範例顯示在 圖 3 中

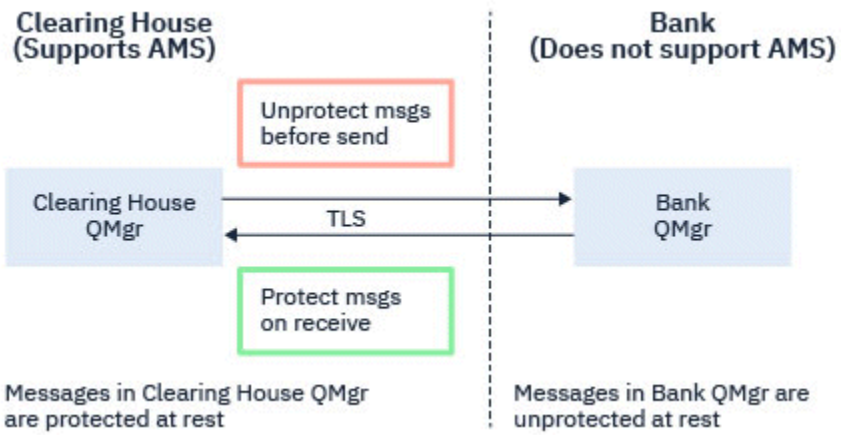


圖 *34:* 事業夥伴之間的訊息流程

相關工作

伺服器至伺服器訊息通道攔截範例配置

z/OS V 9.2.0 伺服器至伺服器訊息通道上的 *AMS* 截取

當傳送端類型訊息通道代理程式從傳輸佇列取得訊息，且接收端類型訊息通道代理程式將訊息放置到目標佇列時，伺服器至伺服器訊息通道截取提供一種方法來控制訊息是否應該套用任何適用的 Advanced Message Security (AMS) 原則。

當使用傳送端、伺服器、接收端和要求端類型的伺服器對伺服器訊息通道，與未啟用 AMS 的佇列管理程式進行通訊時，這可讓在佇列管理程式上啟用 AMS 保護。

也就是說，在傳送至非 AMS 啟用的佇列管理程式之前， AMS 啟用的佇列管理程式中的 AMS 受保護訊息可以不受保護，而從非 AMS 啟用的佇列管理程式接收到的未受保護訊息可以在 AMS 啟用的佇列管理程式上受到適用 AMS 原則的保護。

配置伺服器至伺服器訊息通道截取

在通道類型為傳送端、伺服器、接收端或要求端的通道上，使用 SPLPROT 屬性來配置伺服器至伺服器訊息通道截取。 用來配置行為的可用選項取決於指定的通道類型:

PASSTHRU

訊息通道代理程式為此通道傳送或接收的透通、未變更及其他任何訊息。

此值適用於通道類型 (**CHLTYPE**) 為 SDR、SVR、RCVR 或 RQSTR 的通道，並且是預設值。

REMOVE

從訊息通道代理程式擷取自傳輸佇列的訊息中移除任何 AMS 保護，並將訊息傳送至友機。

在訊息通道代理程式從傳輸佇列取得訊息時，如果已為傳輸佇列定義 AMS 原則，則會套用該原則，以在跨越通道傳送訊息之前從訊息中移除任何 AMS 保護。 如果沒有為傳輸佇列定義 AMS 原則，則會依現狀傳送訊息。

此值僅適用於通道類型為 SDR 或 SVR 的通道。

ASPOLICY

根據為目標佇列定義的原則，將 AMS 保護套用至入埠訊息，然後再將其放入目標佇列。

在訊息通道代理程式接收入埠訊息時，如果已為目標佇列定義 AMS 原則，則會將 AMS 保護套用至訊息，然後再將訊息放入目標佇列。 如果沒有為目標佇列定義 AMS 原則，則會依現狀將訊息放入目標佇列。

此值僅適用於通道類型為 RCVR 或 RQSTR 的通道。

訊息通道截取的使用者 ID

與伺服器至伺服器訊息通道截取搭配使用的使用者 ID 需求，與現有已啟用 AMS 之應用程式的使用者 ID 需求相同。 對於執行中通道，傳送訊息通道代理程式會從傳輸佇列取得訊息，而接收訊息通道代理程式會將訊息放入目標佇列。 在伺服器至伺服器通道上設定的訊息通道代理程式使用者 ID (MCAUSER) 欄位，定義訊息通道代理程式用來執行放置及取得要求的使用者 ID。

利用伺服器至伺服器訊息通道截取，在取得及放置要求期間會執行 AMS 功能，如同其他已啟用 AMS 的應用程式一樣。 因此，訊息通道代理程式使用者 ID 的需求與 AMS 應用程式使用者 ID 的需求相同。

用來執行 put 和 get 的 MCAUSER 是可配置的，且取決於它是出埠還是入埠通道。 如需所選使用者 ID 如何對訊息通道代理程式執行動作的詳細資料，請參閱 MCAUSER 。 因此，執行通道起始程式所使用的使用者 ID ，是在伺服器至伺服器訊息通道截取期間所執行的 AMS 功能所使用的使用者 ID。 因此，這些使用者 ID 的需求與 AMS 應用程式使用者 ID 的需求相同。

使用具有 PUTAUT 配置之通道詳細的現有通道規則來執行鑑別。 如需相關資訊，請參閱 通道起始程式使用的使用者 ID 。

註: 伺服器至伺服器訊息通道截取未考量 PUTAUT 通道屬性的值。

訊息大小及 MAXMSGL

由於 AMS 保護，受保護訊息的訊息大小將大於原始訊息大小。

受保護的訊息大於未受保護的訊息。 因此，在佇列及通道上，可能需要變更 **MAXMSGL** 屬性的值，以考量受保護訊息的大小。

相關參考

伺服器至伺服器訊息通道攔截範例配置

AMS 的錯誤處理

IBM MQ Advanced Message Security 定義錯誤處理佇列，以管理包含錯誤或無法不受保護的訊息。

毀損的訊息會作為例外情況來處理。 如果收到的訊息不符合其佇列的安全需求，例如，如果訊息在應該加密時簽署，或解密或簽章驗證失敗，則會將訊息傳送至錯誤處理佇列。 由於下列原因，可能會將訊息傳送至錯誤處理佇列:

- 保護品質不符-收到的訊息與安全原則中的 QOP 定義之間存在保護品質 (QOP) 不符。
- 解密錯誤-無法解密訊息。
- PDMQ 標頭錯誤-無法存取 Advanced Message Security (AMS) 訊息標頭。
- 大小不符-解密之後的訊息長度不同於預期。
- 加密演算法強度不符-訊息加密演算法弱於必要。
- 不明錯誤-發生非預期的錯誤。

AMS 使用 SYSTEM.PROTECTION.ERROR.QUEUE 作為其錯誤處理佇列。 由 IBM MQ AMS 放置到 SYSTEM.PROTECTION.ERROR.QUEUE 之前有 MQDLH 標頭。

您的 IBM MQ 管理者也可以定義 SYSTEM.PROTECTION.ERROR.QUEUE 作為指向另一個佇列的別名佇列。

z/OS V 9.2.0 從 IBM MQ for z/OS 上的 IBM MQ 9.1.3，如果正在使用伺服器至伺服器的「訊息通道代理程式 (MCA)」截取:

- 如果基於上述其中一個原因， IBM MQ AMS 會將訊息從傳輸佇列移至錯誤處理佇列，傳送端 MCA 只會繼續處理傳輸佇列上的下一個可用訊息。
- 一般而言，現有的通道規則適用於:
  - 將訊息放入「無法傳送的郵件」佇列，以及
  - 放置到「無法傳送的郵件佇列」失敗時所採取的動作。

如需特定實務範例的進一步資訊，請參閱 第 504 頁的『z/OS 上 AMS 的未遞送訊息』 。

z/OS V 9.2.0 *z/OS 上 AMS 的未遞送訊息*

IBM MQ for z/OS 上與伺服器對伺服器「訊息通道代理程式」截取相關的特定實務範例。

從 IBM MQ for z/OS 上的 IBM MQ 9.1.3，如果正在使用伺服器至伺服器的「訊息通道代理程式 (MCA)」截取:

- 在取得及未受保護的訊息之後，如果傳送端 MCA 由於某些原因而無法遞送訊息，例如，因為訊息對通道而言太大，如果 USEDLQ 傳送端通道屬性設為 YES ，則傳送端 MCA 會將訊息移至本端「無法傳送的郵件佇列 (DLQ)」。

  如果 SYSTEM.DEAD.LETTER.QUEUE 正用作本端 DLQ ，訊息未受保護。

  註: IBM MQ AMS 不支援保護放入系統佇列的訊息。

  如果使用具名 DLQ 作為本端 DLQ ，則如果您已定義與具名 DLQ 同名的 IBM MQ AMS 原則，且未定義適當的原則，則訊息會受到保護。
- 如果由於某些原因而無法將訊息放入本端 DLQ ，則如果通道的 NPMSPEED 設為 NORMAL ，或訊息是持續訊息，則會取消現行訊息批次，且通道會進入 RETRY 狀態。 否則，會捨棄訊息，且傳送端 MCA 會繼續處理傳輸佇列上的下一個訊息。

- 假設安全原則對 SYSTEM.DEAD.LETTER.QUEUE，或 第 565 頁的『AMS 中的系統佇列保護』中列出的其他 SYSTEM 佇列 (如果是 SYSTEM.DEAD.LETTER.QUEUE 在使用中， MCA 放置在此佇列中的訊息會依現狀放置。 也就是說，如果訊息先前受到保護，則會將它們放置在受保護的位置; 否則，會將它們放置在不受保護的位置。

  如果佇列管理程式 DEADQ 屬性已設為替代 (非系統) 無法傳送郵件的佇列名稱，且同名的 AMS 原則不存在，則 MCA 放置到此佇列的訊息會依現狀放置。 也就是說，如果訊息先前受到保護，則會將它們放置在受保護的位置; 否則，會將它們放置在不受保護的位置。

  如果佇列管理程式 DEADQ 屬性已設為替代 (非系統) 無法傳送郵件的佇列名稱，且存在與 DLQ 同名的 AMS 原則，則會使用該原則來保護 MCA 放入此佇列的訊息。 如果訊息先前已受到保護，則不會再次受到保護; 這是為了避免雙重保護。 如果不存在同名的 AMS 原則，則會依現狀放置訊息。
- 如果 DLQ 的原則將 setmqspl 指令中的容忍選項設為 off (即 '**-t** O') ，則在訊息未受 AMS 保護且因此沒有 PDMQ 標頭時， DLQ 的放置會失敗。 如果訊息到達接收端時沒有 PDMQ 標頭，則會發生此情況。 亦即，訊息的原始輸出程式沒有目的地的原則，且接收端未設定 SPLPROT (ASPOLICY)。
- 如果為 DLQ 定義的 AMS 原則不允許通道起始程式用來執行保護訊息的使用者 ID ，則 MCA 可能無法將訊息放置到 DLQ。
- 接收端通道通常會將無法遞送的訊息放置在本端 DLQ 中，而傳送端通道通常會將因某些原因而無法處理的訊息放置在本端 DLQ 中，例如，訊息對佇列而言太大，或 MQXQH 標頭不正確等。
- DLQ 處理程式通常只會查看 DLQ 標頭 (DLH) ，而不是訊息有效負載本身。 因此，訊息有效負載可能受到保護的事實，並不會阻止處理程式判斷訊息放置在 DLQ 上的原因。
- 如果未定義 DLQ ，則通道:
  - 如果無法遞送持續訊息，則會異常結束 (並進入重試中狀態)。
  - 捨棄非持續性未遞送訊息，並繼續執行。

相關概念

第 504 頁的『AMS 的錯誤處理』

IBM MQ Advanced Message Security 定義錯誤處理佇列，以管理包含錯誤或無法不受保護的訊息。

AMS 的使用者實務範例

熟悉可能的實務範例，以瞭解您可以使用 Advanced Message Security 達成哪些商業目標。

Windows *Windows 平台上 AMS 的快速入門手冊*

使用本手冊來快速配置 Advanced Message Security ，以在 Windows 平台上提供訊息安全。 當您完成它時，您已建立金鑰資料庫來驗證使用者身分，以及定義佇列管理程式的簽署/加密原則。

開始之前

您應該至少已在系統上安裝下列特性:

- 伺服器
- 開發工具箱 (適用於範例程式)
- Advanced Message Security

如需詳細資料，請參閱 Windows 系統的 IBM MQ 特性 。

如需使用 **setmqenv** 指令來起始設定現行環境，以便作業系統可以找到並執行適當 IBM MQ 指令的相關資訊，請參閱 setmqenv (set IBM MQ environment)。

1. 建立佇列管理程式及佇列

關於這項作業

下列所有範例都使用名為 TEST.Q 的佇列，在應用程式之間傳遞訊息。 Advanced Message Security 在訊息透過標準 IBM MQ 介面進入 IBM MQ 基礎架構時，會使用攔截程式來簽署及加密訊息。 基本設定在 IBM MQ 中完成，並在下列步驟中配置。

您可以使用 IBM MQ Explorer，透過使用所有預設精靈設定來建立佇列管理程式 QM_VERIFY_AMS 及其本端佇列 TEST.Q，也可以使用在 C:\Program Files\IBM\MQ\bin 中找到的指令。 請記住，您必須是 mqm 使用者群組的成員，才能執行下列管理指令。

程序

1. 建立佇列管理程式

```
crtmqm QM_VERIFY_AMS
```

2. 啟動佇列管理程式

```
strmqm QM_VERIFY_AMS
```

3. 在 **runmqsc** 中針對佇列管理程式 QM_VERIFY_AMS 輸入下列指令，以建立稱為 TEST.Q 的佇列。

```
DEFINE QLOCAL(TEST.Q)
```

結果

如果程序已完成，則在 **runmqsc** 中輸入的指令將顯示 TEST.Q 的詳細資料:

```
DISPLAY Q(TEST.Q)
```

2. 建立及授權使用者

關於這項作業

在此範例中有兩個使用者: alice(傳送端) 和 bob(接收端)。 若要使用應用程式佇列，必須授與這些使用者使用它的權限。 此外，為了順利使用我們將定義這些使用者的保護原則，必須授與部分系統佇列的存取權。 如需 **setmqaut** 指令的相關資訊，請參閱 **setmqaut**。

程序

1. 建立這兩個使用者，並確保同時為這兩個使用者設定 HOMEPATH 和 HOMEDRIVE 。
2. 授權使用者連接至佇列管理程式及使用佇列

```
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. 您也應該容許這兩個使用者瀏覽系統原則佇列，並將訊息放置在錯誤佇列上。

```
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
```

小心: IBM MQ 可透過快取原則來最佳化效能，以便您無需在 SYSTEM.PROTECTION.POLICY.QUEUE 。

IBM MQ 不會快取所有可用的原則。 如果有大量原則， IBM MQ 會快取有限數目的原則。 因此，如果佇列管理程式定義的原則數目較少，則不需要提供瀏覽選項給 SYSTEM.PROTECTION.POLICY.QUEUE。

不過，如果定義了大量原則，或您使用舊用戶端，您應該提供此佇列的瀏覽權限。 SYSTEM.PROTECTION.ERROR.QUEUE 用來放置 AMS 程式碼所產生的錯誤訊息。 只有在您嘗試將錯誤訊息放入佇列時，才會檢查此佇列的放置權限。 當您嘗試從 AMS 受保護佇列中放置或取得訊息時，不會檢查您對佇列的放置權限。

結果

現在會建立使用者，並將必要的權限授與他們。

下一步

若要驗證步驟是否正確執行，請使用 amqsput 及 amqsget 範例，如 第 509 頁的『7. 測試設定』小節中所述。

3. 建立金鑰資料庫及憑證

關於這項作業

攔截程式需要傳送端使用者的公開金鑰才能加密訊息。 因此，必須建立對映至公開和私密金鑰之使用者身分的金鑰資料庫。 在實際系統中，使用者和應用程式分散在多部電腦上，每個使用者都有自己的專用金鑰儲存庫。 同樣地，在本手冊中，我們為 alice 和 bob 建立金鑰資料庫，並在它們之間共用使用者憑證。

註: 在本手冊中，我們使用以 C 撰寫並使用本端連結來連接的範例應用程式。 如果您打算使用用戶端連結來使用 Java 應用程式，您必須使用 **keytool** 指令來建立 JKS 金鑰儲存庫和憑證，這是 JRE 的一部分 (如需詳細資料，請參閱 第 524 頁的『AMS 與 Java 用戶端的快速入門手冊』)。 對於所有其他語言，以及對於使用本端連結的 Java 應用程式，本手冊中的步驟是正確的。

程序

1. 使用 IBM 金鑰管理 GUI (strmqikm.exe) 為使用者 alice 建立新的金鑰資料庫。

   ```
   Type:       CMS
   Filename:   alicekey.kdb
   Location:   C:/Documents and Settings/alice/AMS
   ```

   註:

   - 建議使用高保護性密碼來保護資料庫安全。
   - 確定已選取 **將密碼隱藏至檔案** 勾選框。
2. 將金鑰資料庫內容視圖變更為 **個人憑證**。
3. 選取 **新建自簽** ; 在此實務範例中使用自簽憑證。
4. 使用下列欄位來建立憑證，以識別用於加密的使用者 alice :

   ```
   Key label: Alice_Cert
   Common Name: alice
   Organisation: IBM
   Country: GB
   ```

   註:

   - 基於本手冊的目的，我們使用自簽憑證，無需使用「憑證管理中心」即可建立該憑證。 對於正式作業系統，建議不要使用自簽憑證，而是依賴「憑證管理中心」所簽署的憑證。
   - **Key label** 參數指定憑證的名稱，攔截程式會查閱該憑證以接收必要的資訊。
   - **Common Name** 及選用參數指定 **識別名稱** (DN) 的詳細資料，對每一個使用者而言必須是唯一的。
5. 針對使用者 bob 重複步驟 1-4

結果

這兩個使用者 alice 和 bob 現在各有一個自簽憑證。

4. 建立 *keystore.conf*

關於這項作業

您必須將 Advanced Message Security 攔截程式指向金鑰資料庫和憑證所在的目錄。 這是透過 keystore.conf 檔案來完成，該檔案以純文字形式保留該資訊。 每一位使用者在 .mqs 資料夾中必須各有一個 keystore.conf 檔。 必須同時對 alice 和 bob 執行此步驟。

keystore.conf 的內容必須是下列格式:

```
cms.keystore = dir/keystore_file
cms.certificate = certificate_label
```

範例

在此實務範例中， `keystore.conf` 的內容如下:

```
cms.keystore = C:/Documents and Settings/alice/AMS/alicekey
cms.certificate = Alice_Cert
```

註:

- 金鑰儲存庫檔的路徑不得提供副檔名。
- 憑證標籤可以包含空格，因此 "Alice_Cert" 和 "Alice_Cert" 例如， (結尾有一個空格) 會辨識為兩個不同憑證的標籤。 不過，為了避免混淆，最好不要在標籤名稱中使用空格。
- 金鑰儲存庫格式如下: CMS (加密訊息語法)、 JKS (Java 金鑰儲存庫) 和 JCEKS (Java 加密延伸金鑰儲存庫)。 如需相關資訊，請參閱第 535 頁的『AMS 的金鑰儲存庫配置檔 (keystore.conf) 的結構』。
- *%HOMEDRIVE%\%HOMEPATH%*\.mqs\keystore.conf (例如: C:\Documents and Settings\alice\.mqs\keystore.conf) 是 Advanced Message Security 在其中搜尋 `keystore.conf` 檔案的預設位置。 如需如何對 `keystore.conf` 使用非預設位置的相關資訊，請參閱 第 534 頁的『搭配使用金鑰儲存庫和憑證與 AMS』。
- 若要建立 `.mqs` 目錄，您必須使用命令提示字元。

5. 共用憑證

關於這項作業

在兩個金鑰資料庫之間共用憑證，讓每一個使用者都可以順利識別另一個。 作法是將每一個使用者的公用憑證擷取至檔案，然後將該檔案新增至另一個使用者的金鑰資料庫。

註: 請小心使用 *extract* 選項，而不是 *export* 選項。 擷取 會取得使用者的公開金鑰，而 匯出 會同時取得公開和私密金鑰。 錯誤地使用 *export* 將完全損害您的應用程式，因為會傳遞其私密金鑰。

程序

1. 將識別 `alice` 的憑證擷取至外部檔案:

   ```
   runmqakm -cert -extract -db "C:/Documents and Settings/alice/AMS/alicekey.kdb" -pw passw0rd
   -label Alice_Cert -target alice_public.arm
   ```

2. 將憑證新增至 `bob's` 金鑰儲存庫:

   ```
   runmqakm -cert -add -db "C:/Documents and Settings/bob/AMS/bobkey.kdb" -pw passw0rd -label
   Alice_Cert -file alice_public.arm
   ```

3. 針對 bob 重複步驟:

   ```
   runmqakm -cert -extract -db "C:/Documents and Settings/bob/AMS/bobkey.kdb" -pw passw0rd
   -label Bob_Cert -target bob_public.arm

   runmqakm -cert -add -db "C:/Documents and Settings/alice/AMS/alicekey.kdb" -pw passw0rd
   -label Bob_Cert -file bob_public.arm
   ```

結果

現在，這兩個使用者 `alice` 和 bob 能夠順利識別彼此已建立及共用自簽憑證。

下一步

使用 GUI 瀏覽憑證，或執行下列指令來印出其詳細資料，以驗證憑證是否在金鑰儲存庫中:

```
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. 定義佇列原則

關於這項作業

在建立佇列管理程式並準備截取訊息及存取加密金鑰的情況下，我們可以開始使用 `setmqspl` 指令在 `QM_VERIFY_AMS` 上定義保護原則。 如需此指令的相關資訊，請參閱 setmqspl 。 每一個原則名稱必須與要套用它的佇列名稱相同。

範例

這是針對 `TEST.Q` 佇列定義的原則範例。 在此範例中，訊息以 `SHA1` 演算法簽署，並以 `AES256` 演算法加密。 `alice` 是唯一有效的傳送端，而 `bob` 是此佇列上訊息的唯一接收端:

```
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"
```

註: DN 完全符合金鑰資料庫中個別使用者憑證中指定的那些 DN。

下一步

若要驗證您已定義的原則，請發出下列指令:

```
dspmqspl -m QM_VERIFY_AMS
```

若要將原則詳細資料列印為一組 `setmqspl` 指令，請使用 `-export` 旗標。 這容許儲存已定義的原則:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. 測試設定

關於這項作業

透過在不同使用者下執行不同的程式，您可以驗證應用程式是否已適當配置。

程序

1. 將使用者切換成以使用者 `alice` 身分執行

   用滑鼠右鍵按一下 `cmd.exe` ，然後選取 **執行身分 ...**。 當系統提示時，以使用者 `alice` 身分登入。
2. 當使用者 `alice` 使用範例應用程式放置訊息時:

   ```
   amqsput TEST.Q QM_VERIFY_AMS
   ```
3. 鍵入訊息文字，然後按 Enter 鍵。
4. 將使用者切換成以使用者 `bob` 身分執行

   用滑鼠右鍵按一下 `cmd.exe` 並選取 **執行身分 ...**，以開啟另一個視窗。 當系統提示時，以使用者 `bob` 身分登入。
5. 當使用者 `bob` 使用範例應用程式取得訊息時:

   ```
   amqsget TEST.Q QM_VERIFY_AMS
   ```

結果

如果已針對這兩個使用者適當地配置應用程式，則當 `bob` 執行取得應用程式時，會顯示使用者 `alice` 的訊息。

8. 測試加密

關於這項作業

若要驗證是否如預期般進行加密，請建立參照原始佇列 `TEST.Q` 的別名佇列。 此別名佇列將沒有安全原則，因此沒有使用者具有解密訊息的資訊，因此會顯示已加密資料。

程序

1. 針對佇列管理程式 QM_VERIFY_AMS 使用 **runmqsc** 指令，建立別名佇列。

```
DEFINE QALIAS(TEST.ALIAS) TARGET(TEST.Q)
```

2. 授與 bob 存取權以從別名佇列瀏覽

```
setmqaut -m QM_VERIFY_AMS -n TEST.ALIAS -t queue -p bob +browse
```

3. 以使用者 alice 身分，使用範例應用程式來放置另一則訊息，就像之前一樣:

```
amqsput TEST.Q QM_VERIFY_AMS
```

4. 以使用者 bob 身分，這次使用範例應用程式透過別名佇列來瀏覽訊息:

```
amqsbcg TEST.ALIAS QM_VERIFY_AMS
```

5. 以使用者 bob 身分，使用本端佇列中的範例應用程式來取得訊息:

```
amqsget TEST.Q QM_VERIFY_AMS
```

結果

amqsbcg 應用程式的輸出會顯示佇列上的已加密資料，證明訊息已加密。

Linux AIX *AIX and Linux 上 AMS 的快速入門手冊*

使用本手冊來快速配置 Advanced Message Security ，以在 AIX and Linux 上提供訊息安全。 當您完成它時，您已建立金鑰資料庫來驗證使用者身分，以及定義佇列管理程式的簽署/加密原則。

開始之前

您應該至少已在系統上安裝下列元件:

- 執行時期
- 伺服器
- 範例程式
- IBM Global Security Kit (GSKit)
- Advanced Message Security

請參閱下列主題，以取得每一個特定平台上的元件名稱:

- Linux Linux 系統的 IBM MQ 元件
- AIX AIX 系統的 IBM MQ 元件

1. 建立佇列管理程式及佇列

關於這項作業

下列所有範例都使用名為 TEST.Q 的佇列，在應用程式之間傳遞訊息。 Advanced Message Security 在訊息透過標準 IBM MQ 介面進入 IBM MQ 基礎架構時，會使用攔截程式來簽署及加密訊息。 基本設定在 IBM MQ 中完成，並在下列步驟中配置。

您可以使用「IBM MQ 探險家」，利用所有預設精靈設定來建立佇列管理程式 QM_VERIFY_AMS 及其本端佇列 TEST.Q，也可以使用在 *MQ_INSTALLATION_PATH*/bin 中找到的指令。 請記住，您必須是 mqm 使用者群組的成員，才能執行下列管理指令。

程序

1. 建立佇列管理程式

```
crtmqm QM_VERIFY_AMS
```

2. 啟動佇列管理程式

```
strmqm QM_VERIFY_AMS
```

3. 在 **runmqsc** 中針對佇列管理程式 QM_VERIFY_AMS 輸入下列指令，以建立稱為 TEST.Q 的佇列。

```
DEFINE QLOCAL(TEST.Q)
```

結果

如果程序順利完成，則在 **runmqsc** 中輸入的下列指令將顯示 TEST.Q 的詳細資料:

```
DISPLAY Q(TEST.Q)
```

2. 建立及授權使用者

關於這項作業

在此範例中有兩個使用者: alice(傳送端) 和 bob(接收端)。 若要使用應用程式佇列，必須授與這些使用者使用它的權限。 此外，為了順利使用我們將定義這些使用者的保護原則，必須授與部分系統佇列的存取權。 如需 **setmqaut** 指令的相關資訊，請參閱 **setmqaut**。

程序

1. 建立兩個使用者

```
useradd alice
useradd bob
```

2. 授權使用者連接至佇列管理程式及使用佇列

```
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. 您也應該容許這兩個使用者瀏覽系統原則佇列，並將訊息放置在錯誤佇列上。

```
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
```

小心: IBM MQ 可透過快取原則來最佳化效能，以便您無需在 SYSTEM.PROTECTION.POLICY.QUEUE 。

IBM MQ 不會快取所有可用的原則。 如果有大量原則， IBM MQ 會快取有限數目的原則。 因此，如果佇列管理程式定義的原則數目較少，則不需要提供瀏覽選項給 SYSTEM.PROTECTION.POLICY.QUEUE。

不過，如果定義了大量原則，或您使用舊用戶端，您應該提供此佇列的瀏覽權限。 SYSTEM.PROTECTION.ERROR.QUEUE 用來放置 AMS 程式碼所產生的錯誤訊息。 只有在您嘗試將錯誤訊息放入佇列時，才會檢查此佇列的放置權限。 當您嘗試從 AMS 受保護佇列中放置或取得訊息時，不會檢查您對佇列的放置權限。

結果

現在會建立使用者群組，並將必要的權限授與這些使用者群組。 如此一來，指派給那些群組的使用者也將有權連接至佇列管理程式，以及從佇列中放置及取得。

下一步

若要驗證步驟是否正確執行，請使用 amqsput 及 amqsget 範例，如 第 515 頁的『8. 測試加密』小節中所述。

3. 建立金鑰資料庫及憑證

關於這項作業

如果要加密訊息，攔截程式需要傳送使用者的私密金鑰，以及收件者的公開金鑰。 因此，必須建立對映至公開和私密金鑰之使用者身分的金鑰資料庫。 在實際系統中，使用者和應用程式分散在多部電腦上，每個使用者都有自己的專用金鑰儲存庫。 同樣地，在本手冊中，我們為 `alice` 和 `bob` 建立金鑰資料庫，並在它們之間共用使用者憑證。

註: 在本手冊中，我們使用以 C 撰寫並使用本端連結來連接的範例應用程式。 如果您打算使用用戶端連結來使用 Java 應用程式，您必須使用 **keytool** 指令來建立 JKS 金鑰儲存庫和憑證，這是 JRE 的一部分 (如需詳細資料，請參閱 第 524 頁的『AMS 與 Java 用戶端的快速入門手冊』)。 對於所有其他語言，以及對於使用本端連結的 Java 應用程式，本手冊中的步驟是正確的。

程序

1. 為使用者 `alice` 建立新的金鑰資料庫

   ```
   mkdir /home/alice/.mqs -p
   runmqakm -keydb -create -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -stash
   ```

   註:

   - 建議使用高保護性密碼來保護資料庫安全。
   - **stash** 參數會將密碼儲存在 `key.sth` 檔中，供攔截程式用來開啟資料庫。
2. 確定金鑰資料庫可讀取

   ```
   chmod +r /home/alice/.mqs/alicekey.kdb
   ```

3. 建立憑證，以識別要在加密中使用的使用者 `alice`

   ```
   runmqakm -cert -create -db /home/alice/.mqs/alicekey.kdb -pw passw0rd
   -label Alice_Cert -dn "cn=alice,O=IBM,c=GB" -default_cert yes
   ```

   註:

   - 基於本手冊的目的，我們使用自簽憑證，無需使用「憑證管理中心」即可建立該憑證。 對於正式作業系統，建議不要使用自簽憑證，而是依賴「憑證管理中心」所簽署的憑證。
   - **label** 參數指定憑證的名稱，攔截程式會查閱該憑證以接收必要的資訊。
   - **DN** 參數指定 **識別名稱** (DN) 的詳細資料，對於每一個使用者而言必須是唯一的。
4. 現在我們已建立金鑰資料庫，我們應該設定其所有權，並確保所有其他使用者都無法讀取它。

   ```
   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. 針對使用者 `bob` 重複步驟 1-4

結果

這兩個使用者 `alice` 和 `bob` 現在各有一個自簽憑證。

4. 建立 *keystore.conf*

關於這項作業

您必須將 Advanced Message Security 攔截程式指向金鑰資料庫和憑證所在的目錄。 這是透過 `keystore.conf` 檔案來完成，該檔案以純文字形式保留該資訊。 每一位使用者在 `.mqs` 資料夾中必須各有一個 `keystore.conf` 檔。 必須同時對 `alice` 和 `bob` 執行此步驟。

`keystore.conf` 的內容必須是下列格式:

```
cms.keystore = dir/keystore_file
cms.certificate = certificate_label
```

範例

在此實務範例中， keystore.conf 的內容如下:

```
cms.keystore = /home/alice/.mqs/alicekey
cms.certificate = Alice_Cert
```

註:

- 金鑰儲存庫檔的路徑不得提供副檔名。
- 金鑰儲存庫格式如下: CMS (加密訊息語法)、 JKS (Java 金鑰儲存庫) 和 JCEKS (Java 加密延伸金鑰儲存庫)。 如需相關資訊，請參閱第 535 頁的『AMS 的金鑰儲存庫配置檔 (keystore.conf) 的結構』。
- HOME/.mqs/keystore.conf 是 Advanced Message Security 在其中搜尋 keystore.conf 檔案的預設位置。 如需如何對 keystore.conf 使用非預設位置的相關資訊，請參閱 第 534 頁的『搭配使用金鑰儲存庫和憑證與 AMS』。

5. 共用憑證

關於這項作業

在兩個金鑰資料庫之間共用憑證，讓每一個使用者都可以順利識別另一個。 作法是將每一個使用者的公用憑證擷取至檔案，然後將該檔案新增至另一個使用者的金鑰資料庫。

註: 請小心使用 *extract* 選項，而不是 *export* 選項。 擷取 會取得使用者的公開金鑰，而 匯出 會同時取得公開和私密金鑰。 錯誤地使用 *export* 將完全損害您的應用程式，因為會傳遞其私密金鑰。

程序

1. 將識別 alice 的憑證擷取至外部檔案:

   ```
   runmqakm -cert -extract -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Alice_Cert
   -target alice_public.arm
   ```

2. 將憑證新增至 bob's 金鑰儲存庫:

   ```
   runmqakm -cert -add -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Alice_Cert -file
   alice_public.arm
   ```

3. 針對 bob 重複步驟:

   ```
   runmqakm -cert -extract -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Bob_Cert -target
   bob_public.arm
   ```

4. 將 bob 的憑證新增至 alice's 金鑰儲存庫:

   ```
   runmqakm -cert -add -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Bob_Cert -file
   bob_public.arm
   ```

結果

現在，這兩個使用者 alice 和 bob 能夠順利識別彼此已建立及共用自簽憑證。

下一步

執行下列指令來印出憑證的詳細資料，以驗證憑證是否位於金鑰儲存庫中:

```
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. 定義佇列原則

關於這項作業

在建立佇列管理程式並準備截取訊息及存取加密金鑰的情況下，我們可以開始使用 setmqspl 指令在 QM_VERIFY_AMS 上定義保護原則。 如需此指令的相關資訊，請參閱 setmqspl 。 每一個原則名稱必須與要套用它的佇列名稱相同。

範例

這是針對 TEST.Q 佇列定義的原則範例。 在此範例中，訊息由使用者 alice 使用 SHA1 演算法簽署，並使用 256 位元 AES 演算法加密。 alice 是唯一有效的傳送端，而 bob 是此佇列上訊息的唯一接收端:

```
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"
```

註: DN 完全符合金鑰資料庫中個別使用者憑證中指定的那些 DN。

下一步

若要驗證您已定義的原則，請發出下列指令:

```
dspmqspl -m QM_VERIFY_AMS
```

若要將原則詳細資料列印為一組 setmqspl 指令，請使用 -export 旗標。 這容許儲存已定義的原則:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. 測試設定

關於這項作業

透過在不同使用者下執行不同的程式，您可以驗證應用程式是否已適當配置。

程序

1. 切換至包含範例的目錄。 如果 MQ 安裝在非預設位置，則可能位於不同的位置。

   ```
   cd /opt/mqm/samp/bin
   ```

2. 將使用者切換成以使用者 alice 身分執行

   ```
   su alice
   ```

3. 以使用者 alice 身分，使用範例應用程式放置訊息:

   ```
   ./amqsput TEST.Q QM_VERIFY_AMS
   ```

4. 鍵入訊息文字，然後按 Enter 鍵。
5. 停止以使用者 alice 身分執行

   ```
   exit
   ```

6. 將使用者切換成以使用者 bob 身分執行

   ```
   su bob
   ```

7. 以使用者 bob 身分，使用範例應用程式取得訊息:

   ```
   ./amqsget TEST.Q QM_VERIFY_AMS
   ```

結果
如果已針對這兩個使用者適當地配置應用程式，則當 bob 執行取得應用程式時，會顯示使用者 alice 的訊息。

8. 測試加密

關於這項作業
若要驗證是否如預期般進行加密，請建立參照原始佇列 TEST.Q 的別名佇列。 此別名佇列將沒有安全原則，因此沒有使用者具有解密訊息的資訊，因此會顯示已加密資料。

程序
1. 針對佇列管理程式 QM_VERIFY_AMS 使用 **runmqsc** 指令，建立別名佇列。

```
DEFINE QALIAS(TEST.ALIAS) TARGET(TEST.Q)
```

2. 授與 bob 存取權以從別名佇列瀏覽

```
setmqaut -m QM_VERIFY_AMS -n TEST.ALIAS -t queue -p bob +browse
```

3. 以使用者 alice 身分，使用範例應用程式來放置另一則訊息，就像之前一樣:

```
./amqsput TEST.Q QM_VERIFY_AMS
```

4. 以使用者 bob 身分，這次使用範例應用程式透過別名佇列來瀏覽訊息:

```
./amqsbcg TEST.ALIAS QM_VERIFY_AMS
```

5. 以使用者 bob 身分，使用本端佇列中的範例應用程式來取得訊息:

```
./amqsget TEST.Q QM_VERIFY_AMS
```

結果
amqsbcg 應用程式的輸出將顯示佇列上的已加密資料，證明訊息已加密。

z/OS *z/OS 上的 AMS 配置範例*
本節提供 z/OS 上 Advanced Message Security 佇列作業實務範例的原則及憑證配置範例。

如需如何配置 Advanced Message Security 的詳細資料，請參閱 配置 Advanced Message Security for z/OS 。

範例涵蓋所需的 Advanced Message Security 原則，以及相對於使用者和金鑰環必須存在的數位憑證。 這些範例假設已遵循 授與使用者 Advanced Message Security 的資源許可權中所提供的指示，來設定實務範例中涉及的使用者。

V 9.2.0 此外，從 IBM MQ 9.1.3 開始，請參閱 伺服器至伺服器訊息通道攔截範例。

z/OS *z/OS 上 AMS 之受完整性保護訊息的本端佇列作業*
此範例詳細說明在放置及取得應用程式的本端佇列中來回傳送及擷取受完整性保護訊息所需的 Advanced Message Security 原則及憑證。

範例佇列管理程式及佇列如下:

```
BNK6        - Queue manager
FIN.XFER.Q7 - Local queue
```

使用下列使用者:

```
WMQBNK6  - AMS task user
TELLER5  - Sending user
FINADM2  - Recipient user
```

建立使用者憑證

在此範例中，只需要一個使用者憑證。 這是傳送端使用者簽署完整性保護訊息所需的憑證。 傳送使用者是 'TELLER5'。

也需要「憑證管理中心 (CA)」憑證。 CA 憑證是發出使用者憑證之憑證管理中心的憑證。 這可以是憑證鏈。 如果如此，則在 Advanced Message Security 作業使用者 (在此情況下為使用者 WMQBNK6) 的金鑰環中需要鏈中的所有憑證。

可以使用 RACF RACDCERT 指令來建立 CA 憑證。 此憑證用來發出使用者憑證。 例如：

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

這個 RACDCERT 指令會建立 CA 憑證，然後可用來發出使用者 'TELLER5' 的使用者憑證。 例如：

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

您的安裝將具有選擇或建立 CA 憑證的程序，以及發出憑證並將它們配送至相關系統的程序。

匯出及匯入這些憑證時， Advanced Message Security 需要：

- CA 憑證 (鏈結)。
- 使用者憑證及其私密金鑰。

如果您使用 RACF， RACDCERT EXPORT 指令可用來將憑證匯出至資料集， RACDCERT ADD 指令可用來從資料集匯入憑證。 如需這些及其他 RACDCERT 指令的相關資訊，請參閱 *z/OS: Security Server RACF Command Language Reference*。

在此情況下，在執行佇列管理程式 BNK6 的 z/OS 系統上需要憑證。

在執行 BNK6 的 z/OS 系統上匯入憑證時，使用者憑證需要 TRUST 屬性。 RACDCERT ALTER 指令可用來將 TRUST 屬性新增至憑證。 例如：

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

在此範例中，收件者使用者不需要憑證。

將憑證連接至相關金鑰環

當必要的憑證已建立或匯入，並設為授信時，它們必須連接至執行 BNK6 的 z/OS 系統上的適當使用者金鑰環。 如果要建立金鑰環，請使用 RACDCERT ADDRING 指令：

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

這會建立 Advanced Message Security 作業使用者 WMQBNK6 的金鑰環，以及傳送使用者 'TELLER5' 的金鑰環。 請注意，金鑰環名稱 drq.ams.keyring 是必要的，且名稱區分大小寫。

建立金鑰環之後，即可連接相關憑證：

```
RACDCERT ID(WMQBNK6) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))

RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

傳送端使用者憑證必須連接為 DEFAULT。 如果傳送使用者在其 drq.ams.keyring 中具有多個憑證，則會使用預設憑證進行簽署。

除非停止並重新啟動佇列管理程式，或使用 z/OS **MODIFY** 指令來重新整理 Advanced Message Security 憑證配置，否則 Advanced Message Security 無法辨識憑證的建立及修改。 例如：

```
F BNK6AMSM,REFRESH KEYRING
```

建立 Advanced Message Security 原則

在此範例中，會將受完整性保護的訊息放入佇列 FIN.XFER.Q7 由以使用者 'TELLER5' 身分執行的應用程式執行，並由以使用者 'FINADM2' 身分執行的應用程式從相同佇列中擷取，因此只需要一個 Advanced Message Security 原則。

Advanced Message Security 原則是使用 訊息安全原則公用程式 (CSQ0UTIL)中所記載的 CSQ0UTIL 公用程式來建立。

使用 CSQ0UTIL 公用程式來執行下列指令:

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

在此原則中，佇列管理程式識別為 BNK6。 原則名稱及相關聯的佇列是 FIN.XFER.Q7。 用來產生傳送者簽章的演算法是 MD5，傳送使用者的識別名稱 (DN) 是 'CN=Teller5,O=BCO,C=US'。

定義原則之後，請重新啟動 BNK6 佇列管理程式，或使用 z/OS **MODIFY** 指令來重新整理 Advanced Message Security 原則配置。 例如:

```
F BNK6AMSM,REFRESH POLICY
```

z/OS *z/OS 上 AMS 之受隱私權保護訊息的本端佇列作業*

此範例詳細說明在放置及取得應用程式的本端佇列中來回傳送及擷取受隱私權保護訊息所需的 Advanced Message Security 原則及憑證。 受隱私權保護的訊息會同時簽署及加密。

範例佇列管理程式及本端佇列如下:

```
BNK6        - Queue manager
FIN.XFER.Q8 - Local queue
```

使用下列使用者:

```
WMQBNK6  - AMS task user
TELLER5  - Sending user
FINADM2  - Recipient user
```

配置此實務範例的步驟如下:

建立使用者憑證

在此範例中，需要兩個使用者憑證。 這些是傳送端使用者簽署訊息所需的憑證，以及接收端使用者加密及解密訊息資料所需的憑證。 傳送使用者為 'TELLER5'，收件者使用者為 'FINADM2'。

也需要「憑證管理中心 (CA)」憑證。CA 憑證是發出使用者憑證之憑證管理中心的憑證。 這可以是憑證鏈。 如果如此，則在 Advanced Message Security 作業使用者 (在此情況下為使用者 WMQBNK6) 的金鑰環中需要鏈中的所有憑證。

可以使用 RACF RACDCERT 指令來建立 CA 憑證。 此憑證用來發出使用者憑證。 例如:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

這個 RACDCERT 指令會建立 CA 憑證，然後可用來發出使用者 'TELLER5' 和 'FINADM2' 的使用者憑證。 例如:

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

RACDCERT ID(FINADM2) GENCERT SUBJECTSDN(CN('FinAdm2') O('BCO') C('US'))
WITHLABEL('FinAdm2') SIGNWITH(CERTAUTH LABEL('BCOCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

您的安裝將具有選擇或建立 CA 憑證的程序，以及發出憑證並將它們配送至相關系統的程序。

匯出及匯入這些憑證時， Advanced Message Security 需要:

- CA 憑證 (鏈結)。
- 傳送端使用者憑證及其私密金鑰。
- 收件者使用者憑證及其私密金鑰。

如果您使用 RACF， RACDCERT EXPORT 指令可用來將憑證匯出至資料集， RACDCERT ADD 指令可用來從資料集匯入憑證。 有關這些命令和其他 RACDCERT 命令的詳細信息，請參閱 *z/OS*： 安全伺服器 *RACF* 命令語言參考中的 RACDCERT（管理 RACF 數字證書）。

在此情況下，在執行佇列管理程式 BNK6 的 z/OS 系統上需要憑證。

在執行 BNK6 的 z/OS 系統上匯入憑證時，使用者憑證需要 TRUST 屬性。 RACDCERT ALTER 指令可用來將 TRUST 屬性新增至憑證。 例如:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

將憑證連接至相關金鑰環

當必要的憑證已建立或匯入，並設為授信時，它們必須連接至執行 BNK6 的 z/OS 系統上的適當使用者金鑰環。 如果要建立金鑰環，請使用 RACDCERT ADDRING 指令:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
RACDCERT ID(FINADM2) ADDRING(drq.ams.keyring)
```

這會為 Advanced Message Security 作業使用者建立金鑰環，並為傳送端及接收端使用者建立金鑰環。 請注意，金鑰環名稱 drq.ams.keyring 是必要的，且名稱區分大小寫。

已建立金鑰環時，可以連接相關憑證。

```
RACDCERT ID(WMQBNK6) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))

RACDCERT ID(WMQBNK6) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) USAGE(SITE))

RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))

RACDCERT ID(FINADM2) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

傳送端和接收端使用者憑證必須連接為 DEFAULT。 如果任一使用者在其 drq.ams.keyring 中有多個憑證，則預設憑證用於簽署及解密目的。

收件者使用者的憑證也必須使用 USAGE (SITE) 連接至 Advanced Message Security 作業使用者的金鑰環。 這是因為在加密訊息資料時， Advanced Message Security 作業需要收件者的公開金鑰。 USAGE (SITE) 會阻止在金鑰環中存取私密金鑰。

除非停止並重新啟動佇列管理程式，或使用 z/OS **MODIFY** 指令來重新整理 Advanced Message Security 憑證配置，否則 Advanced Message Security 無法辨識憑證的建立及修改。 例如:

```
F BNK6AMSM,REFRESH KEYRING
```

建立 Advanced Message Security 原則

在此範例中，受隱私權保護的訊息會放入佇列 FIN.XFER.Q8 由以使用者 'TELLER5' 身分執行的應用程式執行，並由以使用者 'FINADM2' 身分執行的應用程式從相同佇列中擷取，因此只需要一個 Advanced Message Security 原則。

Advanced Message Security 原則是使用 訊息安全原則公用程式 (CSQ0UTIL)中所記載的 CSQ0UTIL 公用程式來建立。

使用 CSQ0UTIL 公用程式來執行下列指令:

```
setmqspl -m BNK6 -p FIN.XFER.Q8 -s SHA1 -e 3DES -a CN=Teller5,O=BCO,C=US -r
CN=FinAdm2,O=BCO,C=US
```

在此原則中，佇列管理程式識別為 BNK6。 原則名稱及相關聯的佇列是 FIN.XFER.Q8。 用於產生發送者簽署的演算法是 SHA1 ，發送用戶的專有名稱 (DN) 是“ CN=Teller5,O=BCO,C=US ”，接收用戶是'CN=FinAdm2,O=BCO,C=US' 。 用來加密訊息資料的演算法是 3DES。

定義原則之後，請重新啟動 BNK6 佇列管理程式，或使用 z/OS **MODIFY** 指令來重新整理 Advanced Message Security 原則配置。 例如:

```
F BNK6AMSM,REFRESH POLICY
```

z/OS *z/OS 上 AMS 之受完整性保護訊息的遠端佇列作業*

此範例詳細說明在兩個不同佇列管理程式所管理的佇列中來回傳送及擷取受完整性保護的訊息所需的 Advanced Message Security 原則及憑證。 兩個佇列管理程式可以在相同的 z/OS 系統上執行，或在不同的 z/OS 系統上執行，或者一個佇列管理程式可以在執行 Advanced Message Security 的分散式系統上執行。

佇列管理程式和佇列範例如下:

```
BNK6        - Sending queue manager
BNK7        - Recipient queue manager
FIN.XFER.Q7 - Remote queue on BNK6
FIN.RCPT.Q7 - Local queue on BNK7
```

附註: 在此範例中， BNK6 和 BNK7 是在不同 z/OS 系統上執行的佇列管理程式。

使用下列使用者:

```
WMQBNK6  - AMS task user on BNK6
WMQBNK7  - AMStask user on BNK7
TELLER5  - Sending user on BNK6
FINADM2  - Recipient user on BNK7
```

配置此實務範例的步驟如下:

建立使用者憑證

在此範例中，只需要一個使用者憑證。 這是傳送使用者簽署完整性保護訊息所需的憑證。 傳送使用者是'TELLER5'。

也需要「憑證管理中心 (CA)」憑證。CA 憑證是發出使用者憑證之憑證管理中心的憑證。 這可以是憑證鏈。 如果是這樣，則在 Advanced Message Security 作業使用者 (在此情況下為使用者 WMQBNK7) 的金鑰環中需要鏈中的所有憑證。

可以使用 RACF RACDCERT 指令來建立 CA 憑證。 此憑證用來發出使用者憑證。 例如:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

這個 RACDCERT 指令會建立 CA 憑證，然後可用來發出使用者 'TELLER5' 的使用者憑證。 例如:

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

您的安裝將具有選擇或建立 CA 憑證的程序，以及發出憑證並將它們配送至相關系統的程序。

匯出及匯入這些憑證時， Advanced Message Security 需要:

- CA 憑證 (鏈結)。
- 傳送端使用者憑證及其私密金鑰。

如果您使用 RACF， RACDCERT EXPORT 指令可用來將憑證匯出至資料集， RACDCERT ADD 指令可用來從資料集匯入憑證。 有關這些命令和其他 RACDCERT 命令的詳細信息，請參閱 *z/OS*： 安全伺服器 *RACF* 命令語言參考中的 RACDCERT（管理 RACF 數字證書）。

在此情況下，在執行佇列管理程式 BNK6 及 BNK7 的 z/OS 系統上需要憑證。

在此範例中，必須在執行 BNK6 的 z/OS 系統上匯入傳送端憑證，且必須在執行 BNK7 的 z/OS 系統上匯入 CA 憑證。 匯入憑證之後，使用者憑證需要 TRUST 屬性。 RACDCERT ALTER 指令可用來將 TRUST 屬性新增至憑證。 例如，在 BNK6:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

將憑證連接至相關金鑰環

當必要的憑證已建立或匯入，並設為受信任時，它們必須連接至執行 BNK6 及 BNK7 的 z/OS 系統上的適當使用者金鑰環。

若要建立金鑰環，請在 BNK6:

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

這會在 BNK6 上建立傳送使用者的金鑰環。 請注意，金鑰環名稱 drq.ams.keyring 是必要的，且名稱區分大小寫。

在 BNK7:

```
RACDCERT ID(WMQBNK7) ADDRING(drq.ams.keyring)
```

這會在 BNK7 上為 Advanced Message Security 作業使用者建立金鑰環。 BNK7 上的 'TELLER5' 不需要使用者金鑰環。

已建立金鑰環時，可以連接相關憑證。

在 BNK6:

```
RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

在 BNK7:

```
RACDCERT ID(WMQBNK7) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))
```

傳送端使用者憑證必須連接為 DEFAULT。 如果傳送使用者在其 drq.ams.keyring 中具有多個憑證，則會使用預設憑證進行簽署。

除非停止並重新啟動佇列管理程式，或使用 z/OS **MODIFY** 指令來重新整理 Advanced Message Security 憑證配置，否則 Advanced Message Security 無法辨識憑證的建立及修改。 例如：

在 BNK6:

```
F BNK6AMSM,REFRESH,KEYRING
```

在 BNK7:

```
F BNK7AMSM,REFRESH,KEYRING
```

建立 Advanced Message Security 原則

在此範例中，以使用者 'TELLER5' 身分執行的應用程式會將受完整性保護的訊息放置在 BNK6 上的遠端佇列 FIN.XFER.Q7 ，並由以使用者 'FINADM2' 身分執行的應用程式從 BNK7 上的本端佇列 FIN.RCPT.Q7 擷取，因此需要兩個 Advanced Message Security 原則。

Advanced Message Security 原則是使用 訊息安全原則公用程式 (CSQ0UTIL)中所記載的 CSQ0UTIL 公用程式來建立。

使用 CSQ0UTIL 公用程式來執行下列指令，以定義 BNK6:

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

在此原則中，佇列管理程式識別為 BNK6。 原則名稱及相關聯的佇列是 FIN.XFER.Q7。 用來產生傳送者簽章的演算法是 MD5，傳送使用者的識別名稱 (DN) 是 'CN=Teller5,O=BCO,C=US'。

此外，使用 CSQ0UTIL 公用程式來執行下列指令，以定義 BNK7:

```
setmqspl -m BNK7 -p FIN.RCPT.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

在此原則中，佇列管理程式識別為 BNK7。 原則名稱及相關聯的佇列是 FIN.RCPT.Q7。 傳送端簽章預期的演算法是 MD5，傳送端使用者的識別名稱 (DN) 預期為 'CN=Teller5,O=BCO,C=US'。

在定義這兩個原則之後，請重新啟動 BNK6 和 BNK7 佇列管理程式，或使用 z/OS **MODIFY** 指令來重新整理 Advanced Message Security 原則配置。 例如：

在 BNK6:

```
F BNK6AMSM,REFRESH,POLICY
```

在 BNK7:

```
F BNK7AMSM,REFRESH,POLICY
```

z/OS *z/OS 上 AMS 之受隱私權保護訊息的遠端佇列作業*

此範例詳細說明在兩個不同佇列管理程式所管理的佇列中來回傳送及擷取受隱私權保護訊息所需的 Advanced Message Security 原則及憑證。 兩個佇列管理程式可以在相同的 z/OS 系統上執行，或在不同的 z/OS 系統上執行，或者一個佇列管理程式可以在執行 Advanced Message Security 的分散式系統上執行。

佇列管理程式和佇列範例如下:

```
BNK6        - Sending queue manager
BNK7        - Recipient queue manager
FIN.XFER.Q7 - Remote queue on BNK6
FIN.RCPT.Q7 - Local queue on BNK7
```

附註: 在此範例中， BNK6 和 BNK7 是在相同名稱的不同 z/OS 系統上執行的佇列管理程式。

使用下列使用者:

```
WMQBNK6  - AMS task user on BNK6
WMQBNK7  - AMS task user on BNK7
TELLER5  - Sending user on BNK6
FINADM2  - Recipient user on BNK7
```

配置此實務範例的步驟如下:

建立使用者憑證

在此範例中，需要兩個使用者憑證。 這些是傳送端使用者簽署訊息所需的憑證，以及接收端使用者加密及解密訊息資料所需的憑證。 傳送使用者為 'TELLER5' ，收件者使用者為 'FINADM2'。

也需要「憑證管理中心 (CA)」憑證。CA 憑證是發出使用者憑證之憑證管理中心的憑證。 這可以是憑證鏈。 如果是這樣，則在 Advanced Message Security 作業使用者 (在此情況下為使用者 WMQBNK7) 的金鑰環中需要鏈中的所有憑證。

可以使用 RACF RACDCERT 指令來建立 CA 憑證。 此憑證用來發出使用者憑證。 例如:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

這個 RACDCERT 指令會建立 CA 憑證，然後可用來發出使用者 'TELLER5' 和 'FINADM2' 的使用者憑證。 例如：

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

RACDCERT ID(FINADM2) GENCERT SUBJECTSDN(CN('FinAdm2') O('BCO') C('US'))
WITHLABEL('FinAdm2') SIGNWITH(CERTAUTH LABEL('BCOCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

您的安裝將具有選擇或建立 CA 憑證的程序，以及發出憑證並將它們配送至相關系統的程序。

匯出及匯入這些憑證時， Advanced Message Security 需要：

- CA 憑證 (鏈結)。
- 傳送端使用者憑證及其私密金鑰。
- 收件者使用者憑證及其私密金鑰。

如果您使用 RACF， RACDCERT EXPORT 指令可用來將憑證匯出至資料集， RACDCERT ADD 指令可用來從資料集匯入憑證。

有關這些命令和其他 RACDCERT 命令的詳細信息，請參閱 *z/OS*： 安全伺服器 *RACF* 命令語言參考中的 RACDCERT（管理 RACF 數字證書） 。

在此情況下，在執行佇列管理程式 BNK6 及 BNK7 的 z/OS 系統上需要憑證。

在此範例中，必須在執行 BNK6 的 z/OS 系統上匯入傳送端和接收端憑證，並且必須在執行 BNK7 的 z/OS 系統上匯入 CA 和接收端憑證。 匯入憑證之後，使用者憑證需要 TRUST 屬性。 RACDCERT ALTER 指令可用來將 TRUST 屬性新增至憑證。 例如：

在 BNK6:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

在 BNK7:

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

將憑證連接至相關金鑰環

已建立或匯入必要憑證並設為受信任時，它們必須連接至執行 BNK6 及 BNK7 的 z/OS 系統上的適當使用者金鑰環。

如果要建立金鑰環，請使用 RACDCERT ADDRING 指令：

在 BNK6:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

這會在 BNK6 上為 Advanced Message Security 作業使用者建立金鑰環，並為傳送使用者建立金鑰環。 請注意，金鑰環名稱 drq.ams.keyring 是必要的，且名稱區分大小寫。

在 BNK7:

```
RACDCERT ID(WMQBNK7) ADDRING(drq.ams.keyring)
RACDCERT ID(FINADM2) ADDRING(drq.ams.keyring)
```

這會在 BNK7 上為 Advanced Message Security 作業使用者建立金鑰環，並為收件者使用者建立金鑰環。

已建立金鑰環時，可以連接相關憑證。

在 BNK6:

```
RACDCERT ID(WMQBNK6) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) USAGE(SITE))

RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

在 BNK7:

```
RACDCERT ID(WMQBNK7) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))

RACDCERT ID(FINADM2) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

傳送端和接收端使用者憑證必須連接為 DEFAULT。 如果任一使用者在其 drq.ams.keyring 中有多個憑證，則會使用預設憑證進行簽署及加密/解密。

在 BNK6 上，收件者使用者的憑證也必須使用 USAGE (SITE) 連接至 Advanced Message Security 作業使用者的金鑰環。 這是因為在加密訊息資料時， Advanced Message Security 作業需要收件者的公開金鑰。 USAGE (SITE) 會阻止在金鑰環中存取私密金鑰。

除非停止並重新啟動佇列管理程式，或使用 z/OS **MODIFY** 指令來重新整理 Advanced Message Security 憑證配置，否則 Advanced Message Security 無法辨識憑證的建立及修改。 例如：

在 BNK6:

```
F BNK6AMSM,REFRESH,KEYRING
```

在 BNK7:

```
F BNK7AMSM,REFRESH,KEYRING
```

建立 Advanced Message Security 原則

在此範例中，以使用者 'TELLER5' 身分執行的應用程式會將受隱私權保護訊息放入 BNK6 上的遠端佇列 FIN.XFER.Q7 ， 並以使用者 'FINADM2' 身分執行的應用程式會從 BNK7 上的本端佇列 FIN.RCPT.Q7 擷取訊息，因此需要兩個 Advanced Message Security 原則。

Advanced Message Security 原則是使用 訊息安全原則公用程式 (CSQ0UTIL)中所記載的 CSQ0UTIL 公用程式來建立。

使用 CSQ0UTIL 公用程式來執行下列指令，以定義 BNK6:

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s SHA1 -e 3DES -a CN=Teller5,O=BCO,C=US -r
CN=FinAdm2,O=BCO,C=US
```

在此原則中，佇列管理程式識別為 BNK6。 原則名稱及相關聯的佇列是 FIN.XFER.Q7。 用於產生發送者簽署的演算法是 SHA1 ， 發送用戶的專有名稱 (DN) 是“ CN=Teller5,O=BCO,C=US ”， 接收用戶是'CN=FinAdm2,O=BCO,C=US' 。 用來加密訊息資料的演算法是 3DES。

此外，使用 CSQ0UTIL 公用程式來執行下列指令，以定義 BNK7:

```
setmqspl -m BNK7 -p FIN.RCPT.Q7 -s SHA1 -e 3DES -a CN=Teller5,O=BCO,C=US -r
CN=FinAdm2,O=BCO,C=US
```

在此原則中，佇列管理程式識別為 BNK7。 原則名稱及相關聯的佇列是 FIN.RCPT.Q7。 發送方簽章所需的演算法為 SHA1 ， 發送者用戶的專有名稱 (DN) 預計為“ CN=Teller5,O=BCO,C=US ”， 接收者用戶為'CN=FinAdm2,O=BCO,C=US' 。 用來解密訊息資料的演算法是 3DES。

在定義兩個原則之後，請重新啟動 BNK6 及 BNK7 佇列管理程式，或使用 z/OS **MODIFY** 指令來重新整理 Advanced Message Security 原則配置。 例如：

在 BNK6:

```
F BNK6AMSM,REFRESH,POLICY
```

在 BNK7:

```
F BNK7AMSM,REFRESH,POLICY
```

AMS 與 *Java* 用戶端的快速入門手冊

使用本手冊來快速配置 Advanced Message Security，為使用用戶端連結連接的 Java 應用程式提供訊息安全。 當您完成它時，您已建立金鑰儲存庫來驗證使用者身分，以及定義佇列管理程式的簽署/加密原則。

開始之前

確保您已安裝適當的元件，如 **快速入門手冊** (Windows 或 AIX and Linux) 中所述。

1. 建立佇列管理程式及佇列

關於這項作業

下列所有範例都使用名為 TEST.Q 的佇列，在應用程式之間傳遞訊息。 Advanced Message Security 在訊息透過標準 IBM MQ 介面進入 IBM MQ 基礎架構時，會使用攔截程式來簽署及加密訊息。 基本設定在 IBM MQ 中完成，並在下列步驟中配置。

程序

1. 建立佇列管理程式

   ```
   crtmqm QM_VERIFY_AMS
   ```

2. 啟動佇列管理程式

   ```
   strmqm QM_VERIFY_AMS
   ```

3. 在 **runmqsc** 中針對佇列管理程式 QM_VERIFY_AMS 輸入下列指令，以建立並啟動接聽器

   ```
   DEFINE LISTENER(AMS.LSTR) TRPTYPE(TCP) PORT(1414) CONTROL(QMGR)
   START LISTENER(AMS.LSTR)
   ```

4. 在 **runmqsc** for queue manager QM_VERIFY_AMS 中輸入下列指令，以建立通道供應用程式透過來連接。

   ```
   DEFINE CHANNEL(AMS.SVRCONN) CHLTYPE(SVRCONN)
   ```

5. 在 **runmqsc** 中針對佇列管理程式 QM_VERIFY_AMS 輸入下列指令，以建立稱為 TEST.Q 的佇列。

   ```
   DEFINE QLOCAL(TEST.Q)
   ```

結果

如果程序順利完成，則在 **runmqsc** 中輸入的下列指令會顯示 TEST.Q 的詳細資料:

```
DISPLAY Q(TEST.Q)
```

2. 建立及授權使用者

關於這項作業

在此實務範例中有兩個使用者: alice(傳送端) 和 bob(接收端)。 若要使用應用程式佇列，必須授與這些使用者使用它的權限。 此外，若要順利使用此實務中定義的保護原則，必須授與這些使用者對部分系統佇列的存取權。 如需 **setmqaut** 指令的相關資訊，請參閱 **setmqaut**。

程序

1. 依照您平台的 **快速入門手冊** (Windows 或 AIX and Linux) 中的說明，建立這兩個使用者。
2. 授權使用者連接至佇列管理程式及使用佇列

```
setmqaut -m QM_VERIFY_AMS -t qmgr -p alice -p bob +connect +inq
setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p alice +put
setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p bob +get +inq +browse
```

3. 您也應該容許這兩個使用者瀏覽系統原則佇列，並將訊息放置在錯誤佇列上。

```
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
```

小心: IBM MQ 可透過快取原則來最佳化效能，以便您無需在 SYSTEM.PROTECTION.POLICY.QUEUE 。

IBM MQ 不會快取所有可用的原則。 如果有大量原則， IBM MQ 會快取有限數目的原則。 因此，如果佇列管理程式定義的原則數目較少，則不需要提供瀏覽選項給 SYSTEM.PROTECTION.POLICY.QUEUE。

不過，如果定義了大量原則，或您使用舊用戶端，您應該提供此佇列的瀏覽權限。 SYSTEM.PROTECTION.ERROR.QUEUE 用來放置 AMS 程式碼所產生的錯誤訊息。 只有在您嘗試將錯誤訊息放入佇列時，才會檢查此佇列的放置權限。 當您嘗試從 AMS 受保護佇列中放置或取得訊息時，不會檢查您對佇列的放置權限。

結果

現在會建立使用者，並將必要的權限授與他們。

下一步

若要驗證步驟是否正確執行，請使用 `JmsProducer` 及 `JmsConsumer` 範例，如 第 527 頁的『7. 測試設定』小節中所述。

3. 建立金鑰資料庫及憑證

關於這項作業

如果要將訊息加密至攔截程式，則需要傳送端使用者的公開金鑰。 因此，必須建立對映至公開和私密金鑰之使用者身分的金鑰資料庫。 在實際系統中，使用者和應用程式分散在多部電腦上，每個使用者都有自己的專用金鑰儲存庫。 同樣地，在本手冊中，我們為 `alice` 和 `bob` 建立金鑰資料庫，並在它們之間共用使用者憑證。

註: 在本手冊中，我們使用在使用用戶端連結連接的 Java 中撰寫的範例應用程式。 如果您計劃使用使用本端連結或 C 應用程式的 Java 應用程式，則必須使用 **runmqakm** 指令來建立 CMS 金鑰儲存庫及憑證。 這會顯示在 **快速入門手冊** (Windows 或 AIX and Linux) 中。

程序

1. 建立要在其中建立金鑰儲存庫的目錄，例如 `/home/alice/.mqs`。 您可能想要在 **快速入門手冊** (Windows 或 AIX and Linux) 針對您的平台所使用的相同目錄中建立它。

   註: 在下列步驟中，此目錄稱為 *keystore-dir*
2. 建立新的金鑰儲存庫和憑證，以識別要在加密中使用的使用者 `alice`

   註: **keytool** 指令是 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
```

   註:

   - 如果 *keystore-dir* 包含空格，則必須以引號括住金鑰儲存庫的完整名稱

- 建議使用高保護性密碼來保護金鑰儲存庫安全。
- 基於本手冊的目的，我們使用自簽憑證，無需使用「憑證管理中心」即可建立該憑證。 對於正式作業系統，建議不要使用自簽憑證，而是依賴「憑證管理中心」所簽署的憑證。
- **alias** 參數指定憑證的名稱，攔截程式會查閱該憑證以接收必要的資訊。
- **dname** 參數指定 **識別名稱** (DN) 的詳細資料，對於每一個使用者而言必須是唯一的。

3. 在 AIX and Linux 上，確保可讀取金鑰儲存庫

```
chmod +r keystore-dir/keystore.jks
```

4. 針對使用者 bob 重複 step1-4

結果

這兩個使用者 alice 和 bob 現在各有一個自簽憑證。

4. 建立 *keystore.conf*

關於這項作業

您必須將 Advanced Message Security 攔截程式指向金鑰資料庫和憑證所在的目錄。 這是透過 keystore.conf 檔案來完成，該檔案以純文字格式保留該資訊。 每一位使用者都必須有個別的 keystore.conf 檔。 應該同時針對 alice 和 bob 執行此步驟。

範例

在此實務範例中， keystore.conf for alice 的內容如下:

```
JKS.keystore = keystore-dir/keystore
JKS.certificate = Alice_Java_Cert
JKS.encrypted = no
JKS.keystore_pass = passw0rd
JKS.key_pass = passw0rd
JKS.provider = IBMJCE
```

在此實務範例中， keystore.conf for bob 的內容如下:

```
JKS.keystore = keystore-dir/keystore
JKS.certificate = Bob_Java_Cert
JKS.encrypted = no
JKS.keystore_pass = passw0rd
JKS.key_pass = passw0rd
JKS.provider = IBMJCE
```

註:

- 金鑰儲存庫檔的路徑不得提供副檔名。
- If you already have a keystore.conf file because you have followed the instructions in the Quick Start Guide (Windows or AIX and Linux), you can edit the existing file to add these lines.
- 如需相關資訊，請參閱 第 535 頁的『AMS 的金鑰儲存庫配置檔 (keystore.conf) 的結構』。

5. 共用憑證

關於這項作業

在兩個金鑰儲存庫之間共用憑證，以便每一個使用者都可以順利識別另一個金鑰儲存庫。 作法是擷取每一個使用者的憑證，並將它匯入至另一個使用者的金鑰儲存庫。

註: 不同的憑證工具會以不同方式使用術語 擷取 和 匯出 。 例如， IBM Global Security Kit (GSKit) **strmqikm** 指令 (ikeyman) 工具會區分您 擷取 憑證 (公開金鑰) 並 匯出 私密金鑰。 對於提供這兩個選項的工具而言，這項區別非常重要，因為錯誤地使用 匯出 將會透過傳遞其私密金鑰來完全損害您的應用程式。 由於區別非常重要， IBM MQ 文件力求一致地使用這些術語。 不過， Java keytool 提供一個稱為 *exportcert* 的指令行選項，只會擷取公開金鑰。 基於這些原因，下列程序是指使用 *exportcert* 選項來 擷取 憑證。

程序

1. 擷取識別 alice 的憑證。

```
keytool -exportcert -keystore alice-keystore-dir/keystore.jks -storepass passw0rd
-alias Alice_Java_Cert -file alice-keystore-dir/Alice_Java_Cert.cer
```

2. 將憑證識別 alice 匯入至 bob 將使用的金鑰儲存庫。 當系統提示時，表示您將信任此憑證。

```
keytool -importcert -file alice-keystore-dir/Alice_Java_Cert.cer -alias Alice_Java_Cert
-keystore bob-keystore-dir/keystore.jks -storepass passw0rd
```

3. 針對 bob 重複步驟

結果

現在，這兩個使用者 alice 和 bob 能夠順利識別彼此已建立及共用自簽憑證。

下一步

執行下列指令來印出憑證的詳細資料，以驗證憑證是否位於金鑰儲存庫中:

```
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. 定義佇列原則

關於這項作業

在建立佇列管理程式並準備截取訊息及存取加密金鑰的情況下，我們可以開始使用 setmqspl 指令在 QM_VERIFY_AMS 上定義保護原則。 如需此指令的相關資訊，請參閱 setmqspl 。 每一個原則名稱必須與要套用它的佇列名稱相同。

範例

這是在 TEST.Q 佇列上定義的原則範例，由使用者 alice 使用 SHA1 演算法簽署，並針對使用者 bob 使用 256 位元 AES 演算法進行加密:

```
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"
```

註: DN 完全符合金鑰資料庫中個別使用者憑證中指定的那些 DN。

下一步

若要驗證您已定義的原則，請發出下列指令:

```
dspmqspl -m QM_VERIFY_AMS
```

若要將原則詳細資料列印為一組 setmqspl 指令，請使用 -export 旗標。 這容許儲存已定義的原則:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. 測試設定

開始之前

請確定您使用的 Java 版本已安裝未限定的 JCE 原則檔。

註: IBM MQ 安裝中提供的 Java 版本已具有這些原則檔。 它可以在 *MQ_INSTALLATION_PATH*/java/bin 中找到。

關於這項作業

透過在不同使用者下執行不同的程式，您可以驗證應用程式是否已適當配置。 如需在不同使用者下執行程式的詳細資料，請參閱適用於您平台的 **快速入門手冊** (Windows 或 AIX)。

程序

1. 若要執行這些 JMS 範例應用程式，請使用平台的 CLASSPATH 設定，如 IBM MQ classes for JMS 使用的環境變數 中所示，以確保包含範例目錄。
2. 以使用者 `alice` 身分，使用範例應用程式來放置訊息，並以用戶端連接:

```
java JmsProducer -m QM_VERIFY_AMS -d TEST.Q -h localhost -p 1414 -l AMS.SVRCONN
```

3. 以使用者 bob 身分，使用範例應用程式取得訊息，並以用戶端身分進行連接:

```
java JmsConsumer -m QM_VERIFY_AMS -d TEST.Q -h localhost -p 1414 -l AMS.SVRCONN
```

結果

如果已針對這兩個使用者適當地配置應用程式，則當 bob 執行取得應用程式時，會顯示使用者 `alice` 的訊息。

在 *AMS* 上保護遠端佇列

若要完全保護遠端佇列，必須在遠端佇列及將訊息傳輸至其中的本端佇列上設定原則。

將訊息放入遠端佇列時， Advanced Message Security 會截取作業，並根據遠端佇列的原則集來處理訊息。 例如，對於加密原則，訊息在傳遞至 IBM MQ 以處理之前會先加密。 在 Advanced Message Security 處理放入遠端佇列的訊息之後， IBM MQ 會將它放入相關聯的傳輸佇列，並將它轉遞至目標佇列管理程式及目標佇列。

在本端佇列上執行 GET 作業時， Advanced Message Security 會嘗試根據本端佇列上的原則集來解碼訊息。 若要讓作業成功，用來解密訊息的原則必須與用來加密訊息的原則相同。 任何不相符都會導致拒絕訊息。

如果基於任何原因而無法同時設定這兩個原則，則會提供暫置實施支援。 原則可以在已開啟容錯旗標的本端佇列上設定，這指出當嘗試從佇列擷取訊息所涉及的訊息沒有安全原則集時，可以忽略與佇列相關聯的原則。 在此情況下， GET 會嘗試解密訊息，但會容許遞送未加密的訊息。 如此一來，在本端佇列受到保護 (及測試) 之後，就可以設定遠端佇列上的原則。

記住: 完成 Advanced Message Security 轉出之後，請移除容錯旗標。

相關參考

setmqspl (設定安全原則)

使用 *IBM Integration Bus* 透過 *AMS* 遞送受保護訊息

Advanced Message Security 可以保護已安裝 IBM Integration Bus 或 WebSphere Message Broker 8.0.0.1 (或更新版本) 的基礎架構中的訊息。 在 IBM Integration Bus 環境中套用安全之前，您應該先瞭解這兩個產品的本質。

關於這項作業

Advanced Message Security 提供訊息有效負載的端對端安全。 這表示只有指定為訊息有效寄件者及收件者的當事人才能產生或接收訊息。 這意味著為了保護流經 IBM Integration Bus 的訊息安全，您可以容許 IBM Integration Bus 在不知道訊息內容的情況下處理訊息 (實務範例 1) 或讓它成為授權使用者能夠接收及傳送訊息 (實務範例 2)。

實務範例 *1- Integration Bus* 無法查看訊息內容

開始之前

您應該將「 IBM Integration Bus 」連接至現有的佇列管理程式。 在後面的指令中，將 *QMgrName* 取代為這個現有的佇列管理程式名稱。

關於這項作業

在此實務範例中， Alice 將受保護訊息放入輸入佇列 QIN。 根據訊息內容 routeTo，訊息會遞送至 *bob* (QBOB) ，[1](QCECIL) 或預設 (QDEF) 佇列。 遞送是可能的，因為 Advanced Message Security 只會保護訊息有效負載，而不會保護其標頭和內容，這些標頭和內容仍未受保護，可供 IBM Integration Bus 讀取。 Advanced Message Security 僅由 *alice*、 *bob* 及 *cecil* 使用。 不需要針對 IBM Integration Bus 安裝或配置它。

IBM Integration Bus 會從未受保護的別名佇列接收受保護的訊息，以避免任何解密訊息的嘗試。 如果要直接使用受保護的佇列，則會將訊息放在無法解密的「無法傳送的郵件」佇列中。 訊息由 IBM Integration Bus 遞送，且到達目標佇列時未變更。 因此仍由原始作者簽署 (*bob* 和 *cecil* 都只接受 *alice* 所傳送的訊息) 並像之前一樣受到保護 (只有 *bob* 和 *cecil* 可以讀取它)。 IBM Integration Bus 會將遞送的訊息放置到未受保護的別名。 收件者會從受保護的輸出佇列中擷取訊息， AMS 會在其中透通地解密訊息。

程序

1. 依照 **快速入門手冊** (Windows 或 AIX) 中的說明，將 阿利切、 波布 和 塞西爾 配置成使用 Advanced Message Security 。

   請確定已完成下列步驟:

   - 建立及授權使用者
   - 建立金鑰資料庫及憑證
   - 正在建立 keystore.conf
2. 提供 *alice* 的憑證給 *bob* 和 *cecil*，以便在檢查訊息上的數位簽章時可以識別 *alice* 。

   作法是將識別 *alice* 的憑證擷取至外部檔案，然後將擷取的憑證新增至 *bob* 及 *cecil* 金鑰儲存庫。 請務必使用 **作業 5 中說明的方法**。 **快速入門手冊 中的 共用憑證** (Windows 或 AIX)。
3. 將 *bob* 和 *cecil* 的 憑證提供給 *alice*，以便 *alice* 可以傳送針對 *bob* 和 *cecil* 加密的訊息。

   請使用前一個步驟中指定的方法來執行此動作。
4. 在佇列管理程式上，定義稱為 QIN、 QBOB、 QCECIL 及 QDEF 的本端佇列。

   ```
   DEFINE QLOCAL(QIN)
   ```
5. 將 QIN 佇列的安全原則設定為合格配置。 對 QBOB、 QCECIL 和 QDEF 佇列使用相同的設定。

   ```
   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"
   ```

   此實務範例假設安全原則，其中 *alice* 是唯一授權寄件者，而 *bob* 及 *cecil* 是收件者。
6. 分別定義別名佇列 AIN、 ABOB 及 ACECIL 參照本端佇列 QIN、 QBOB 及 QCECIL 。

   ```
   DEFINE QALIAS(AIN) TARGET(QIN)
   ```
7. 請驗證前一個步驟中所指定別名的安全配置不存在; 否則請將其原則設為 NONE。

   ```
   dspmqspl -m QMgrName -p AIN
   ```
8. 在 IBM Integration Bus 中建立訊息流程，以根據訊息的 routeTo 內容，將到達 AIN 別名佇列的訊息遞送至 BOB、CECIL 或 DEF 節點。 若要執行此動作:

   a) 建立稱為 IN 的 MQInput 節點，並指派 AIN 別名作為其佇列名稱。

   b) 建立稱為 BOB、 CECIL 和 DEF 的 MQOutput 節點，並指派別名佇列 ABOB、 ACECIL 和 ADEF 作為其各自的佇列名稱。

   c) 建立路徑節點並將它稱為 TEST。

   d) 將 IN 節點連接至 TEST 節點的輸入端。

   e) 為 TEST 節點建立 bob 和 cecil 輸出端。

   f) 將 bob 輸出端連接至 BOB 節點。

[1] 塞西爾

g) 將 `cecil` 輸出端連接至 `CECIL` 節點。

h) 將 DEF 節點連接至預設輸出端。

i) 套用下列規則:

```
$Root/MQRFH2/usr/routeTo/text()="bob"
$Root/MQRFH2/usr/routeTo/text()="cecil"
```

9. 將訊息流程部署至 IBM Integration Bus 執行時期元件。
10. 以使用者 `Alice` 身分執行會放置訊息，其中也包含稱為 `routeTo` 且值為 `bob` 或 `cecil` 的訊息內容。執行範例應用程式 **amqsstm** 將容許您執行此動作。

```
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. 以使用者 *bob* 身分執行， `QBOB` 使用範例應用程式 **amqsget** 從佇列擷取訊息。

結果

當 *alice* 將訊息放置在 `QIN` 佇列上時，訊息會受到保護。 它由 IBM Integration Bus 從 `AIN` 別名佇列中以受保護的形式擷取。 IBM Integration Bus 決定將讀取 `routeTo` 內容的訊息遞送至何處，因為所有內容都未加密。 IBM Integration Bus 會將訊息放在適當的未受保護別名上，避免進一步保護。 當 *bob* 或 *cecil* 從佇列接收時，會解密訊息並驗證數位簽章。

實務範例 *2- Integration Bus* 可以查看訊息內容

關於這項作業

在此實務範例中，容許一組個人將訊息傳送至 IBM Integration Bus。 另一個群組獲授權接收 IBM Integration Bus 所建立的訊息。 無法竊聽雙方與 IBM Integration Bus 之間的傳輸。

請記住，只有在開啟佇列時， IBM Integration Bus 才會讀取保護原則和憑證，因此在對保護原則進行任何更新之後，您必須重新載入執行群組，變更才會生效。

```
mqsireload execution-group-name
```

如果將 IBM Integration Bus 視為容許讀取或簽署訊息有效負載的授權方，您必須為啟動 IBM Integration Bus 服務的使用者配置 Advanced Message Security 。 請注意，不一定是將訊息放入/取得佇列的使用者，或是建立及部署 IBM Integration Bus 應用程式的使用者。

程序

1. 配置 阿利切、 波布、 塞西爾 和 達韋 以及 IBM Integration Bus 服務使用者，以使用 Advanced Message Security ， 如 **快速入門手冊** (Windows 或 AIX) 中所述。

   請確定已完成下列步驟:

   - 建立及授權使用者
   - 建立金鑰資料庫及憑證
   - 正在建立 keystore.conf

2. 提供 *alice*、 *bob*、 *cecil* 及 *dave* 憑證給 IBM Integration Bus 服務使用者。

   作法是將識別 *alice*、 *bob*、 *cecil* 及 *dave* 的每一個憑證擷取至外部檔案，然後將擷取的憑證新增至 IBM Integration Bus 金鑰儲存庫。 請務必使用 **作業 5 中說明的方法**。 **快速入門手冊 中的 共用憑證** (Windows 或 AIX)。

3. 提供 IBM Integration Bus 服務使用者的憑證給 *alice*、 *bob*、 *cecil* 及 *dave*。

請使用前一個步驟中指定的方法來執行此動作。

註: *Alice* 和 *bob* 需要 IBM Integration Bus 服務使用者的憑證，才能正確加密訊息。 IBM Integration Bus 服務使用者需要 *alice* 及 *bob* 憑證，才能驗證訊息的作者。 IBM Integration Bus 服務使用者需要 *cecil* 及 *dave* 憑證來加密其訊息。 *cecil* 和 *dave* 需要 IBM Integration Bus 服務使用者的憑證來驗證訊息是否來自 IBM Integration Bus。

4. 定義名為 IN 的本端佇列，並定義安全原則，並將 *alice* 和 *bob* 指定為作者，並將 IBM Integration Bus 的服務使用者指定為收件者:

```
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. 定義名為 OUT 的本端佇列，並使用指定為作者之 IBM Integration Bus 的服務使用者，以及指定為收件者 *cecil* 及 *dave* 的安全原則:

```
setmqspl -m QMgrName -p OUT -s MD5 -a "CN=broker,O=IBM,C=GB" -e AES256
-r "CN=cecil,O=IBM,C=GB" -r "CN=dave,O=IBM,C=GB"
```

6. 在 IBM Integration Bus 中，建立含有 MQInput 和 MQOutput 節點的訊息流程。 將 MQInput 節點配置成使用 IN 佇列，並將 MQOutput 節點配置成使用 OUT 佇列。
7. 將訊息流程部署至 IBM Integration Bus 執行時期元件。
8. 以使用者 *alice* 或 *bob* 身分執行，會 IN 使用範例應用程式 **amqsput** 將訊息放置在佇列上。
9. 以使用者 *cecil* 或 *dave* 身分執行，使用範例應用程式 **amqsget** 從佇列 OUT 擷取訊息。

結果

alice 或 *bob* 傳送至輸入佇列 IN 的訊息已加密，僅容許 IBM Integration Bus 讀取。 IBM Integration Bus 只接受來自 *alice* 和 *bob* 的訊息，並拒絕任何其他訊息。 會適當地處理接受的訊息，然後使用 *cecil* 的 及 *dave* 的 金鑰來簽署及加密，然後再放入輸出佇列 OUT。 只有 *cecil* 和 *dave* 能夠讀取它， IBM Integration Bus 未簽署的訊息會被拒絕。

將 *Advanced Message Security* 與 *Managed File Transfer* 搭配使用

此實務範例說明如何配置 Advanced Message Security ， 以針對透過 Managed File Transfer 傳送的資料提供訊息隱私權。

開始之前

確保您已在 IBM MQ 安裝上安裝 Advanced Message Security 元件，該元件管理您要保護的 Managed File Transfer 所使用的佇列。

如果您的 Managed File Transfer 代理程式以連結模式連接，請確定您也已在其本端安裝上安裝 IBM Global Security Kit (GSKit) 元件。

關於這項作業

當兩個 Managed File Transfer 代理程式之間的資料傳送岔斷時，在用來管理傳送的基礎 IBM MQ 佇列上，機密資料可能仍未受保護。 此實務範例說明如何配置及使用 Advanced Message Security 來保護 Managed File Transfer 佇列上的此類資料。

在此實務範例中，我們認為簡式拓蹼包含一部具有兩個 Managed File Transfer 佇列及兩個代理程式 (AGENT1 和 AGENT2) 的機器，共用單一佇列管理程式，如實務範例 Managed File Transfer 實務範例中所述。 這兩個代理程式以相同方式 (以連結模式或用戶端模式) 連接。

1. 建立憑證

開始之前

此實務範例使用簡式模型，其中使用者 ftagent 在群組中 FTAGENTS 用來執行 Managed File Transfer Agent 處理程序。 如果您使用自己的使用者和群組名稱，請相應地變更指令。

關於這項作業

Advanced Message Security 使用公開金鑰加密法來簽署及/或加密受保護佇列上的訊息。

註:

- 如果您的 Managed File Transfer 代理程式以連結模式執行，則您用來建立 CMS (加密訊息語法) 金鑰儲存庫的指令詳述於適用於您平台的 **快速入門手冊** (Windows 或 AIX)。
- 如果 Managed File Transfer 代理程式以用戶端模式執行，則 第 524 頁的『AMS 與 Java 用戶端的快速入門手冊』中詳述您將需要建立 JKS (Java 金鑰儲存庫) 的指令。

程序

1. Create a self-signed certificate to identify the user `ftagent` as detailed in the appropriate Quick Start Guide.
   使用識別名稱 (DN)，如下所示:

   ```
   CN=ftagent, OU=MFT, O=<organisation>, L=<location>, ST=<state>, C=<country>
   ```

2. Create a `keystore.conf` file to identify the location of the keystore and the certificate within it as detailed in the appropriate Quick Start Guide.

2. 配置訊息保護

關於這項作業

您應該使用 **setmqspl** 指令，為 AGENT2 所使用的資料佇列定義安全原則。 在此實務範例中，會使用相同的使用者來啟動兩個代理程式，因此簽章者和接收端 DN 是相同的，且符合我們所產生的憑證。

程序

1. 使用 **fteStopAgent** 指令關閉 Managed File Transfer 代理程式，以準備進行保護。
2. 建立安全原則以保護 SYSTEM.FTE.DATA.AGENT2 佇列。

   ```
   setmqspl -m hubQM -p SYSTEM.FTE.DATA.AGENT2 -s SHA1 -a "CN=ftagent, OU=MFT,
   O=<organisation>, L=<location>, ST=<state>, C=<country>"
   -e AES128 -r "CN=ftagent, OU=MFT, O=<organisation>, L=<location>, ST=<state>, C=<country>"
   ```

3. 確保執行 Managed File Transfer Agent 處理程序的使用者有權瀏覽系統原則佇列，並將訊息放置在錯誤佇列上。

   ```
   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** 指令重新啟動 Managed File Transfer 代理程式。
5. 使用 **fteListAgents** 指令並驗證代理程式是否處於 READY 狀態，以確認代理程式已順利重新啟動。

結果

您現在可以從 AGENT1 提交傳送至 AGENT2，並在兩個代理程式之間安全地傳輸檔案內容。

Advanced Message Security 安裝概觀

在各種平台上安裝 Advanced Message Security 元件。

程序

- 在多平台上安裝 Advanced Message Security。
- 安裝 IBM MQ Advanced for z/OS。
- 安裝 IBM MQ Advanced for z/OS Value Unit Edition。

相關工作

解除安裝 Advanced Message Security

z/OS 上 AMS 的審核

Advanced Message Security (AMS) for z/OS 提供一種方法，可讓應用程式在受原則保護的佇列上選擇性地審核作業。 啟用時，會針對原則保護佇列上這些作業的成功和失敗產生 IBM 系統管理機能 (SMF) 審核記錄。 審核的作業包括 MQPUT、 MQPUT1 及 MQGET。

依預設會停用審核，不過，您可以透過在 AMS 位址空間的已配置 Language Environment ® _CEE_ENVFILE 檔案中配置 _AMS_SMF_TYPE 及 _AMS_SMF_AUDIT 來啟動審核。 如需相關資訊，請參閱 建立 Advanced Message Security 的程序。 _AMS_SMF_TYPE 變數用來指定 SMF 記錄類型，並且是介於 128 和 255 之間的數字。 SMF 記錄類型通常是 180，但並非必要。 透過指定值 0 來停用審核。 _AMS_SMF_AUDIT 變數會配置是否針對成功及/或失敗的作業建立審核記錄。 當 AMS 使用操作員指令作用中時，也可以動態變更審核選項。 如需相關資訊，請參閱 操作 Advanced Message Security。

SMF 記錄是使用子類型來定義，子類型 1 是一般審核事件。 SMF 記錄包含與正在處理之要求相關的所有資料。

SMF 記錄由目標程式庫 SCSQMACS 中提供的 CSQ0KSMF 巨集對映 (請注意巨集名稱中的零)。 如果您要撰寫 SMF 資料的資料縮減程式，您可以併入這個對映巨集，以協助開發及自訂 SMF 後處理常式。

在 Advanced Message Security for z/OS 所產生的 SMF 記錄中，資料會組織成區段。 記錄包含:

- 標準 SMF 標頭
- 由 Advanced Message Security 針對 z/OS 定義的標頭延伸
- 產品區段
- 資料區段

SMF 記錄的產品區段一律存在於 Advanced Message Security for z/OS 所產生的記錄中。 資料區段會根據子類型而有所不同。 目前會定義一個子類型，因此會使用單一資料區段。

SMF 說明在 z/OS System Management Facilities 手冊 (SA22-7630) 中。 系統 PARMLIB 資料集的 SMFPRMxx 成員中說明有效的記錄類型。 如需相關資訊，請參閱 SMF 說明文件。

Advanced Message Security 審核報告產生器 (CSQ0USMF)

Advanced Message Security for z/OS 提供稱為 CSQ0USMF 的審核報告產生器工具，在安裝 SCSQAUTH 程式庫中提供。 在安裝程式庫 SCSQPROC 中提供用來執行 CSQ0USMF 公用程式 (稱為 CSQ40RSM) 的範例 JCL。

在執行 CSQ0USMF 公用程式之前，必須將 SMF 類型 180 記錄從系統 SMF 資料集傾出至循序資料集。 舉例來說，這個 JCL 會從 SMF 資料集傾出 SMF 類型 180 記錄，並將它們傳送至目標資料集:

```
//IFAUDUMP EXEC PGM=IFASMFDP
//INDD1 DD DSN=SYSn.MANn.syst,DISP=SHR
//OUTDD1 DD DSN=your.target.dataset,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
INDD(INDD1,OPTIONS(DUMP))
OUTDD(OUTDD1,TYPE(180))
/*
```

您必須驗證安裝所使用的實際 SMF 資料集名稱。 所傾出記錄的目標資料集必須具有記錄格式 VBS，且記錄長度為 32760。

註: 如果正在使用 SMF 日誌串流，您必須使用程式 IFASMFDL，將日誌串流傾出至循序資料集。 如需所使用 JCL 的範例，請參閱 處理類型 116 SMF 記錄 。

然後，目標資料集可以用作 CSQ0USMF 公用程式的輸入，以產生 AMS 審核報告。 例如:

```
//STEP1 EXEC PGM=CSQ0USMF,
// PARM=('/ -SMFTYPE 180 -M qmgr')
//STEPLIB DD DSN=thlqual.SCSQANLE,DISP=SHR
//    DD DSN=thlqual.SCSQAUTH,DISP=SHR
//SMFIN DD DSN=your.target.dataset,DISP=SHR
//
```

CSQ0USMF 程式接受兩個選用參數，列在 第 534 頁的表 101 中:

表 101: CSQ0USMF 選用參數		
參數	**值**	**說明**
SMFTYPE	nnn	適用於審核報告的 SMF 記錄類型。 CSQ0USMF 程式在產生報告時只會使用符合 SMFTYPE 值的 SMF 記錄。 如果您未指定 SMFTYPE，則會使用預設值 180。
M	qmgr	適用於審核報告的 IBM MQ 佇列管理程式名稱。 如果不指定-M 參數，稽核報告將包括 SMFIN 資料集中表示的所有佇列管理器的所有稽核記錄。

搭配使用金鑰儲存庫和憑證與 AMS

為了向 IBM MQ 應用程式提供透通加密保護， Advanced Message Security 會使用金鑰儲存庫檔，其中儲存公開金鑰憑證和私密金鑰。 在 z/OS 上，會使用 SAF 金鑰環來取代金鑰儲存庫檔。

在 Advanced Message Security 中，使用者和應用程式是以公開金鑰基礎架構 (PKI) 身分來代表。 這種類型的身分用來簽署及加密訊息。 在與已簽署及加密訊息相關聯的憑證中，主體的 **識別名稱 (DN)** 欄位代表 PKI 身分。 若要讓使用者或應用程式加密其訊息，他們需要存取儲存憑證及相關聯私密和公開金鑰的金鑰儲存庫檔。

在 AIX, Linux, and Windows 上，金鑰儲存庫的位置在金鑰儲存庫配置檔中提供，依預設為 `keystore.conf`。 每一個 Advanced Message Security 使用者都必須具有指向金鑰儲存庫檔的金鑰儲存庫配置檔。 Advanced Message Security 接受下列格式的金鑰儲存庫檔: `.kdb`、 `.jceks`、 `.jks`。

`keystore.conf` 檔案的預設位置為:

- IBM i Linux AIX 在 IBM i 上， AIX and Linux: `$HOME/.mqs/keystore.conf`
- Windows 在 Windows 上: *%HOMEDRIVE%%HOMEPATH%*`\.mqs\keystore.conf`

  註: Windows 上的路徑可以且應該指定磁碟機代號 (如果有多個磁碟機代號可用的話)。

如果您使用指定的金鑰儲存庫檔名和位置，您應該使用下列指令

- 若為 Java: `java -D`*`MQS_KEYSTORE_CONF=path/filename app_name`*
- 若為 C 用戶端及伺服器:
  - 在 AIX and Linux 上: `export MQS_KEYSTORE_CONF=`*`path/filename`*
  - 在 Windows 上: `set MQS_KEYSTORE_CONF=`*`path\filename`*

保護 keystore.conf 檔案中的機密性資訊

V 9.2.0 V 9.2.0

為了存取金鑰儲存庫檔機密性資訊 (例如密碼)，您必須提供記號，讓 IBM MQ Advanced Message Security (AMS) 可以存取金鑰儲存庫，以及簽署和加密訊息。

您應該使用 AMS 隨附的 **runamscred** 指令，來保護金鑰儲存庫配置檔中包含的機密性資訊。 如需如何保護配置檔的詳細資料，請參閱 第 550 頁的『設定 配置檔 的 AMS 密碼保護』。

保護密碼時，您應該使用自訂高度加密金鑰。 為了在執行時期存取密碼，必須將此加密金鑰提供給 AMS。

提供加密金鑰檔的位置有兩種方法:

- `keystore.conf` 檔中的 **amscred.keyfile** 配置內容
- **MQS_AMSCRED_KEYFILE** 環境變數 (environment variable)

優先順序是 **MQS_AMSCRED_KEYFILE**，後面接著 **amscred.keyfile**，然後是預設金鑰。

如需相關資訊，請參閱 第 473 頁的『Advanced Message Security』。

相關概念

第 558 頁的『AMS 中的寄件者識別名稱』
傳送端識別名稱 (DN) 可識別獲授權將訊息放置在佇列上的使用者。 在將訊息放入佇列之前，傳送端會使用其憑證來簽署訊息。

第 559 頁的『AMS 中的收件者識別名稱』
收件者識別名稱 (DN) 可識別獲授權從佇列擷取訊息的使用者。

AMS 的金鑰儲存庫配置檔 (keystore.conf) 的結構

金鑰儲存庫配置檔 (`keystore.conf`) 指向 Advanced Message Security 適當金鑰儲存庫的位置。

下列每一個配置檔類型都有字首:

V 9.2.0 V 9.2.0 **AMSCRED**
與密碼保護系統相關的參數。

CMS
憑證管理系統，配置項目字首為: `cms.`

PKCS#11
公開金鑰加密法標準 #11，配置項目的字首為: `pkcs11.`

IBM i **PEM**
「隱私權加強型郵件」格式，配置項目的字首為: `pem.`

JKS
Java KeyStore,設定條目前綴為： `jks.`

JCEKS
Java 加密金鑰 KeyStore,設定條目前綴為： `jceks.`

z/OS MQ Adv. VUE **JCERACFKS**
Java 加密 RACF 密鑰環密鑰 KeyStore,配置條目前綴為： `jceracfks` 。

重要: 從 IBM MQ 9.0 開始，會忽略 `JCEKS.provider` 和 `JKS.provider` 值。 使用 Bouncy Castle 提供者，與使用中 JRE 所提供的任何 JCE/JCE 供應一起使用。 如需相關資訊，請參閱第 539 頁的『使用 AMS 支援非 IBM JRE』。

金鑰儲存庫的範例結構:

CMS

```
cms.keystore = /dir/keystore_file
cms.certificate = certificate_label
```

PKCS#11

```
pkcs11.library = dir\cryptoki.dll
pkcs11.certificate = certificatelabel
pkcs11.token = tokenlabel
pkcs11.token_pin = tokenpin
pkcs11.secondary_keystore = dir\signers
```
V 9.2.2 `pkcs11.encrypted = no`

IBM i PEM

```
pem.private = /dir/keystore_file_private_key
pem.public = /dir/keystore_file_public_keys
pem.password = password
```
V 9.2.2 `pem.encrypted = no`

Java JKS

```
jks.keystore = dir/Keystore
jks.certificate = certificate_label
jks.encrypted = no
jks.keystore_pass = password
jks.key_pass = password
```

Java JCEKS

```
jceks.keystore = dir/Keystore
jceks.certificate = certificate_label
jceks.encrypted = no
jceks.keystore_pass = password
jceks.key_pass = password
```

Java JCERACFKS

```
jceracfks.keystore = safkeyring://user/keyring
jceracfks.certificate = certificate_label
```

Java PKCS#11

```
pkcs11.library = dir\cryptoki.dll
pkcs11.certificate = certificatelabel
pkcs11.token = tokenlabel
pkcs11.token_pin = tokenpin
pkcs11.secondary_keystore = dir\signers
pkcs11.secondary_keystore_pass = password
pkcs11.encrypted = no
```

表 *102:* 每一種配置檔類型所需的參數摘要

參數	必要	配置檔類型				
		Java (PKCS#11、JKS、JCEKS 及 JCERACFKS)	**IBM i PEM**	**PKCS#11**	**CMS**	**AMSCRED**
keystore	✓	✓			✓	
IBM i private	✓		IBM i ✓			
IBM i public	✓		IBM i ✓			
IBM i password	✓		IBM i ✓			
library	✓	✓		✓		
certificate	✓	✓		✓	✓	
token	✓	✓		✓		
token_pin	✓	✓		✓		
secondary_keystore	✓	✓		✓		

表 102: 每一種配置檔類型所需的參數摘要 (繼續)

參數	必要	配置檔類型				
		Java (PKCS#11、JKS、JCEKS 及 JCERACFKS)	IBM i PEM	PKCS#11	CMS	AMSCRED
secondary_keystore_password	✓	✓				
encrypted		✓	IBM i V 9.2.2 ✓	V 9.2.2 ✓		
keystore_pass	✓	✓				
key_pass		✓				
provider		✓				
keyfile						✓

請注意，您可以使用 # 符號來新增註解。

配置檔參數定義如下:

keystore

僅限 CMS 和 Java 配置。

CMS、JKS 及 JCEKS 配置的金鑰儲存庫檔路徑。

z/OS MQ Adv. VUE JCERACFKS 配置的 RACF 金鑰環 URI。

重要:

- 金鑰儲存庫檔的路徑不得包含副檔名。
- z/OS MQ Adv. VUE RACF 金鑰環的 URI 必須是下列格式:

```
safkeyring://user/keyring
```

其中:

- *user* 是擁有金鑰環的使用者 ID
- *keyring* 是金鑰環名稱。

IBM i **private**

僅限 PEM 配置。

包含 PEM 格式之私密金鑰及憑證的檔案檔名。

IBM i **public**

僅限 PEM 配置。

包含 PEM 格式的授信公用憑證之檔案的檔名。

IBM i **password**

僅限 PEM 配置。

用來解密已加密私密金鑰的密碼。

V 9.2.2 您應該使用原生 AMS 密碼保護工具來保護此欄位; 請參閱 第 539 頁的『保護密碼』

library
僅限 PKCS#11 。

PKCS#11 程式庫的路徑名稱。

certificate
僅限 CMS、 PKCS#11 及 Java 配置。

憑證標籤。

token
僅限 PKCS#11 。

記號標籤。

token_pin
僅限 PKCS#11 。

用來解除鎖定記號的 PIN 碼。

V 9.2.0 V 9.2.0 僅適用於 Java 作業; 您應該使用 Java AMS 密碼保護工具來保護此欄位; 請參閱 第 539 頁的『保護密碼』。

V 9.2.2 僅適用於原生作業; 您應該使用原生 AMS 密碼保護工具來保護此欄位; 請參閱 第 539 頁的『保護密碼』。

secondary_keystore
僅限 PKCS#11 。

提供不含 .kdb 副檔名的 CMS 金鑰儲存庫的路徑名稱，其中包含 PKCS #11 記號上儲存的憑證所需的錨點憑證 (主要憑證)。 次要金鑰儲存庫也可以包含信任鏈中的中繼憑證，以及隱私權安全原則中定義的收件者憑證。 這個 CMS 金鑰儲存庫必須隨附一個隱藏檔，且必須位於與次要金鑰儲存庫相同的目錄中。

對於 Java 環境，需要 JKS 金鑰儲存庫，且您必須提供 **secondary_keystore_password**。

secondary_keystore_password
僅限 Java PKCS#11 。

透過 secondary_keystore 內容提供之 JKS 金鑰儲存庫的密碼。 您應該使用 Java AMS 密碼保護工具來保護此欄位; 請參閱 第 539 頁的『保護密碼』。

encrypted
V 9.2.0 V 9.2.0 僅限 Java 配置。

V 9.2.2 Java、 PKCS#11 及 IBM i PEM 配置。

密碼的狀態。

keystore_pass
僅限 Java 配置。

金鑰儲存庫檔的密碼。

V 9.2.0 V 9.2.0 僅適用於 Java 作業。 您應該使用 Java AMS 密碼保護工具來保護此欄位; 請參閱 第 539 頁的『保護密碼』。

key_pass
僅限 Java 配置。

使用者私密金鑰的密碼。

V 9.2.0 V 9.2.0 僅適用於 Java 作業; 您應該使用 Java AMS 密碼保護工具來保護此欄位; 請參閱 第 539 頁的『保護密碼』。

V 9.2.0 V 9.2.0 **keyfile**
提供在保護或解密此配置檔中包含的密碼時要使用的起始金鑰位置; 請參閱 第 539 頁的『保護密碼』

provider
僅限 Java 配置。

實作金鑰儲存庫憑證所需之加密演算法的 Java 安全提供者。

重要: 儲存在金鑰儲存庫中的資訊對於使用 IBM MQ 所傳送的資料安全流程至關重要。 當安全管理者將檔案許可權指派給這些檔案時，必須特別注意。

保護密碼

V 9.2.0 V 9.2.0

您應該保護 keystore.conf 檔案中包含的密碼及其他機密性資訊。 如需相關資訊，請參閱 **runamscred** 。

keystore.conf 檔案的範例:

V 9.2.0 V 9.2.0

```
# Native AMS application configuration
cms.keystore = c:\Documents and Settings\Alice\AliceKeystore
cms.certificate = AliceCert

# Java AMS application configuration
jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = no
jceks.keystore_pass = passw0rd
jceks.key_pass = passw0rd
```

相關工作

第 550 頁的『設定 配置檔 的 AMS 密碼保護』

將金鑰儲存庫及私密金鑰密碼儲存為純文字會造成安全風險，因此 Advanced Message Security 提供的工具可以使用使用者的金鑰來編碼這些密碼。

使用 AMS 支援非 IBM JRE

使用非 IBM JRE 執行時， IBM MQ classes for Java 及 IBM MQ classes for JMS 支援 Advanced Message Security 作業。

Advanced Message Security (AMS) 實作 加密訊息語法 (CMS)。 CMS 語法用來數位簽署、摘要、鑑別或加密任意訊息內容。

從 IBM MQ 9.0 開始， IBM MQ classes for Java 和 IBM MQ classes for JMS 中的 Advanced Message Security 支援會使用開放程式碼 Bouncy Castle 套件來支援 CMS。 這表示當使用非 IBM JRE 來執行時，這些類別可以支援 Advanced Message Security 作業。

在 IBM MQ 9.0 之前， Java 用戶端中的非 IBM JRE 不支援 Advanced Message Security 。 IBM MQ classes for Java 和 IBM MQ classes for JMS 中的 Advanced Message Security 支援取決於 Java Cryptography Extensions (JCE) 的 IBM 實作所特別提供的 CMS 支援。 由於此限制，只有在使用包含 Java JCE 提供者的 Java runtime environment (JRE) 時，才能使用此功能。

Bouncy Castle JAR 檔的位置和版本編號

IBM MQ classes for Java 和 IBM MQ classes for JMS 安裝套件包含支援非 IBM JRE 所需的 Bouncy Castle JAR 檔。

使用的 Bouncy Castle JAR 檔是下列檔案:

提供者 JAR 檔，這是 Bouncy Castle 作業的基礎。
這個 JAR 檔稱為 bcprov-jdk15on.jar。

"PKIX" JAR 檔，包含 Advanced Message Security 所使用的 CMS 作業支援。
這個 JAR 檔稱為 bcpkix-jdk15on.jar。

V 9.2.0.4 V 9.2.4 **"util" JAR 檔，包含其他 Bouncy Castle JAR 檔所使用的類別。**
這個 JAR 檔稱為 bcutil-jdk15on.jar。

相依關係

已使用 IBM JRE 及 Oracle JRE 來測試 IBM MQ 9.1 及更新版本的類別。 它們也可能在任何符合 J2SE-compliant 標準的 JRE 下順利執行。 不過，您應該注意下列相依關係:

- Advanced Message Security 配置沒有任何變更。
- Bouncy Castle 類別僅用於 CMS 作業。 所有其他安全相關作業 (例如金鑰儲存庫存取、資料的實際加密，以及簽章總和檢查的計算) 都使用 JRE 所提供的功能。

  重要: 因此，所使用的 JRE 必須包含 JCE 提供者實作。
- 如果要使用某些 高度 加密演算法，您可能需要安裝 JRE JCE 實作的 未限定 原則檔。

  如需詳細資料，請參閱 JRE 說明文件。
- 如果您已啟用 Java 安全:
  - 將 `java.security.SecurityPermissioninsertProvider.BC` 新增至應用程式，以便可以使用 Bouncy Castle 類別作為安全提供者。
  - 授與 `java.security.AllPermission` 至 Bouncy Castle JAR 檔，其為:

    V 9.2.0.4 V 9.2.4 *mq_install_dir*/java/lib/bcutil-jdk15on.jar
    mq_install_dir/java/lib/bcpkix-jdk15on.jar
    mq_install_dir/java/lib/bcprov-jdk15on.jar

相關概念

針對 IBM MQ for JMS 類別安裝的內容
針對 IBM MQ for Java 類別所安裝的項目

Multi 訊息通道代理程式 (MCA) 截取及 AMS

MCA 截取可讓在 IBM MQ 下執行的佇列管理程式選擇性地啟用要針對伺服器連線通道套用的原則。

MCA 截取可讓留在 AMS 外部的用戶端仍連接至佇列管理程式，並將其訊息加密及解密。

當無法在用戶端啟用 AMS 時， MCA 攔截旨在提供 AMS 功能。 請注意，使用 MCA 攔截及啟用 AMS 的用戶端，會導致雙重保護訊息，而在接收應用程式時可能會有問題。 如需相關資訊，請參閱第 542 頁的『在用戶端停用 Advanced Message Security』。

註: AMQP 或 MQTT 通道不支援 MCA 攔截程式。

金鑰儲存庫配置檔

依預設， MCA 攔截的金鑰儲存庫配置檔是 `keystore.conf`，且位於啟動佇列管理程式或接聽器之使用者的 HOME 目錄路徑中的 `.mqs` 目錄。 也可以使用 MQS_KEYSTORE_CONF 環境變數來配置金鑰儲存庫。 如需配置 AMS 金鑰儲存庫的相關資訊，請參閱 第 534 頁的『搭配使用金鑰儲存庫和憑證與 AMS』。

若要啟用 MCA 截取，您必須提供要在金鑰儲存庫配置檔中使用的通道名稱。 對於 MCA 截取，只能使用 cms 金鑰儲存庫類型。

如需設定 MCA 截取的範例，請參閱 第 541 頁的『AMS 的 MCA 攔截範例』。

小心: 您必須在選取的通道上完成用戶端鑑別及加密 (例如，使用 SSL 及 SSLPEER 或 CHLAUTH TYPE (SSLPEERMAP))，以確保只有授權用戶端才能連接及使用此功能。

IBM i

如果您的企業使用 IBM i，且您選取商業憑證管理中心 (CA) 來簽署您的憑證，則「數位 Certificate Manager」會以 PEM (隱私權-加強郵件) 格式建立憑證申請。 您必須將要求轉遞至您選擇的 CA。

若要這樣做，您必須使用下列指令，為 `channelname` 中指定的通道選取正確的憑證:

```
pem.certificate.channel.channelname
```

AMS 的 *MCA* 攔截範例

關於如何設定 AMS MCA 截取的範例作業。

開始之前

小心: 您必須在選取的通道上完成用戶端鑑別及加密 (例如，使用 SSL 及 SSLPEER 或 CHLAUTH TYPE (SSLPEERMAP))，以確保只有授權用戶端才能連接及使用此功能。

如果您的企業使用 IBM i，且您選取商業憑證管理中心 (CA) 來簽署您的憑證，則「數位 Certificate Manager」會以 PEM (隱私權-加強郵件) 格式建立憑證申請。 您必須將要求轉遞至您選擇的 CA。

關於這項作業

此作業會引導您完成設定系統以使用 MCA 截取，然後驗證設定的程序。

註: 在 IBM WebSphere MQ 7.5 之前， AMS 是一個附加程式產品，需要個別安裝，並配置攔截程式來保護應用程式。 從 IBM WebSphere MQ 7.5 開始，在 MQ 用戶端和伺服器執行時期環境中，會自動包含和動態啟用攔截程式。 在此 MCA 截取範例中，在通道的伺服器端提供攔截程式，並使用較舊的用戶端執行時期 (在步驟 12 中) 在通道中放置未受保護的訊息，以便可以看到它受到 MCA 攔截程式的保護。 如果此範例使用 IBM WebSphere MQ 7.5 或更新版本的用戶端，則會導致訊息受到兩次保護，因為 MQ 用戶端執行時期攔截程式和 MCA 攔截程式都會在訊息進入 MQ 時保護訊息。

小心: 將程式碼中的 `userID` 取代為您的使用者 ID。

程序

1. 使用下列指令來建立金鑰資料庫及憑證，以建立 Shell Script。
   此外，請變更 **INSTLOC** 和 **KEYSTORELOC**，或執行必要的指令。 請注意，您可能不需要為 bob 建立憑證。

```
INSTLOC=/opt/mq90
KEYSTORELOC=/home/testusr/ssl/ams1
mkdir -p $KEYSTORELOC
chmod -R 777 $KEYSTORELOC
chown -R mqm:mqm $KEYSTORELOC
export PATH=$PATH:$INSTLOC/gskit8/bin
echo "PATH = $PATH"
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$INSTLOC/gskit8/lib64

gsk8capicmd_64 -keydb -create -db $KEYSTORELOC/alicekey.kdb -pw passw0rd -stash
gsk8capicmd_64 -keydb -create -db $KEYSTORELOC/bobkey.kdb -pw passw0rd -stash
gsk8capicmd_64 -cert -create -db $KEYSTORELOC/alicekey.kdb -pw passw0rd
-label alice_cert -dn "cn=alice,O=IBM,c=IN" -default_cert yes
gsk8capicmd_64 -cert -create -db $KEYSTORELOC/bobkey.kdb -pw passw0rd
-label bob_cert  -dn "cn=bob,O=IBM,c=IN"  -default_cert yes
```

2. 在兩個金鑰資料庫之間共用憑證，讓每一個使用者都可以順利識別另一個。
   請務必使用 **作業 5 中說明的方法**。 **快速入門手冊 中的 共用憑證** (Windows 或 AIX and Linux)。
3. 使用下列配置建立 `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. 建立並啟動佇列管理程式 AMSQMGR1
5. 使用 埠 14567 和 控制項 QMGR 定義接聽器
6. 停用通道權限或設定通道權限的規則。
   如需相關資訊，請參閱 SET CHLAUTH。
7. 停止佇列管理程式。
8. 設定金鑰儲存庫:

```
export MQS_KEYSTORE_CONF=/home/userID/ssl/ams1/keystore.conf
```

9. 在相同的 Shell 上啟動佇列管理程式。
10. 設定安全原則並驗證:

```
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
```

如需相關資訊，請參閱 setmqspl 及 dspmqspl 。

11. 設定通道配置:

```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/127.0.0.1(14567)'
```

12. 從不會自動啟用 MCA 攔截程式的 MQ 用戶端 (例如 IBM WebSphere MQ 7.1 或更早版本的用戶端) 執行 **amqsputc** 。 放置下列兩則訊息:

```
/opt/mqm/samp/bin/amqsputc TESTQ TESTQMGR
```

13. 移除安全原則並驗證結果:

```
setmqspl -m AMSQMGR1 -p TESTQ -remove
dspmqspl -m AMSQMGR1
```

14. 從 IBM MQ 9.0 安裝架構瀏覽佇列:

```
/opt/mq90/samp/bin/amqsbcg TESTQ AMSQMGR1
```

瀏覽輸出會以加密格式顯示訊息。

15. 設定安全原則並驗證結果:

```
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. 從 IBM MQ 9.0 安裝執行 **amqsgetc** :

```
/opt/mqm/samp/bin/amqsgetc TESTQ TESTQMGR
```

相關工作

第 524 頁的『AMS 與 Java 用戶端的快速入門手冊』
使用本手冊來快速配置 Advanced Message Security ，為使用用戶端連結連接的 Java 應用程式提供訊息安全。 當您完成它時，您已建立金鑰儲存庫來驗證使用者身分，以及定義佇列管理程式的簽署/加密原則。

相關參考

第 500 頁的『AMS 的已知限制』
有許多 IBM MQ 選項要么不受支持，要么對 Advanced Message Security (AMS) 有限制。

在用戶端停用 Advanced Message Security

如果您使用 IBM WebSphere MQ 7.5 或更新版本用戶端從舊版產品連接至佇列管理程式，且報告 2085 (MQRC_UNKNOWN_OBJECT_NAME) 錯誤，則需要停用 IBM MQ Advanced Message Security (AMS)。

關於這項作業

從 IBM WebSphere MQ 7.5 開始，會自動在 IBM MQ 用戶端中啟用 IBM MQ Advanced Message Security (AMS) ，因此依預設，用戶端會嘗試檢查佇列管理程式中物件的安全原則。 不過，舊版產品 (例如 IBM WebSphere MQ 7.1) 上的伺服器未啟用 AMS ，這會導致報告 2085 (MQRC_UNKNOWN_OBJECT_NAME) 錯誤。

如果報告此錯誤，當您嘗試從舊版產品連接至佇列管理程式時，您可以停用 AMS ，如下所示:

- 若為 Java 用戶端，請使用下列任何方式:
  - 透過設定環境變數 AMQ_DISABLE_CLIENT_AMS。
  - 透過設定 Java 系統內容 com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS。

  - 透過使用 DisableClientAMS 內容，在 `mqclient.ini` 檔案中的 **Security** 段落下。
- 若為 C 用戶端，請使用下列其中一種方式:
  - 透過設定環境變數 MQS_DISABLE_ALL_INTERCEPT。
  - 透過使用 DisableClientAMS 內容，在 `mqclient.ini` 檔案中的 **Security** 段落下。

註: 在 IBM WebSphere MQ 7.5 中，您也可以使用環境變數 AMQ_DISABLE_CLIENT_AMS。 適用於 C 用戶端。 從 IBM MQ 8.0 開始，您無法再將 AMQ_DISABLE_CLIENT_AMS 環境變數用於 C 用戶端。 您需要改用 MQS_DISABLE_ALL_INTERCEPT 環境變數。

程序

- 若要在用戶端停用 AMS，請使用下列其中一個選項:

  AMQ_DISABLE_CLIENT_AMS 環境變數
  在下列情況下，您需要設定此變數:
  - 如果您使用 IBM Java 執行時期環境 (JRE) 以外的 Java 執行時期環境 (JRE)
  - 如果您使用 IBM WebSphere MQ 7.5，或更新版本 IBM MQ classes for JMS 或 IBM MQ classes for Java 用戶端。

  建立 AMQ_DISABLE_CLIENT_AMS 環境變數，並在應用程式執行所在的環境中將其設為 TRUE。 例如:

  ```
  export AMQ_DISABLE_CLIENT_AMS=TRUE
  ```

  Java 系統內容 com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS
  對於 IBM MQ classes for JMS 和 IBM MQ classes for Java 用戶端，您可以將 Java 應用程式的 Java 系統內容 com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS 設定為值 TRUE。

  例如，當呼叫 Java 指令時，您可以將 Java 系統內容設為 `-D` 選項:

  ```
  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，您可以在 JMS 配置檔 `jms.config` 內指定 Java 系統內容 (如果應用程式使用此檔案)。

  MQS_DISABLE_ALL_INTERCEPT 環境變數
  如果您將 IBM MQ 8.0 或更新版本與原生用戶端搭配使用，且需要在用戶端停用 AMS，則需要設定此變數。

  建立環境變數 MQS_DISABLE_ALL_INTERCEPT，並在用戶端執行所在的環境中將它設為 TRUE。 例如:

  ```
  export MQS_DISABLE_ALL_INTERCEPT =TRUE
  ```

  您只能對 C 用戶端使用 MQS_DISABLE_ALL_INTERCEPT 環境變數。 對於 Java 用戶端，您需要改用 AMQ_DISABLE_CLIENT_AMS 環境變數。

  `mqclient.ini` 檔案中的 DisableClientAMS 內容
  您可以將此選項用於 IBM MQ classes for JMS 及 IBM MQ classes for Java 用戶端，以及用於 C 用戶端。

  在 `mqclient.ini` 檔的 **Security** 段落下新增內容名稱 `DisableClientAMS`，如下列範例所示:

  ```
  Security:
  DisableClientAMS=Yes
  ```

  您也可以啟用 AMS，如下列範例所示:

  ```
  Security:
  DisableClientAMS=No
  ```

下一步

如需開啟 AMS 受保護佇列的問題相關資訊，請參閱 搭配使用 AMS 與 JMS 時開啟受保護佇列時發生問題。

相關概念

第 540 頁的『訊息通道代理程式 (MCA) 截取及 AMS』

MCA 截取可讓在 IBM MQ 下執行的佇列管理程式選擇性地啟用要針對伺服器連線通道套用的原則。

相關工作

使用配置檔來配置用戶端

相關參考

IBM MQ classes for JMS 配置檔

AMS 的憑證需求

憑證必須具有 RSA 公開金鑰，才能與 Advanced Message Security 搭配使用。

如需不同公開金鑰類型及其建立方式的相關資訊，請參閱 第 37 頁的『IBM MQ 中的數位憑證及 CipherSpec 相容性』。

金鑰用法延伸

金鑰用法延伸對憑證的使用方式有其他限制。

在 Advanced Message Security 中，必須根據 RFC 5280 規格來設定 X.509 v3 憑證的金鑰用法。

為了保護品質完整性，如果已設定憑證金鑰使用延伸，則該設定必須至少包含下列兩者之一:

- **nonRepudiation**
- **digitalSignature**

對於保護隱私權的品質，如果已設定憑證金鑰使用延伸，則該設定必須包括:

- **keyEncipherment**

對於保護機密性的品質，如果已設定憑證金鑰使用延伸，則該設定必須包括:

- **dataEncipherment**

延伸金鑰用法會進一步精簡金鑰用法延伸。 對於所有保護品質，如果已設定憑證延伸金鑰用法，則該集必須包括:

- **emailProtection**

相關概念

第 561 頁的『AMS 中的保護品質』

Advanced Message Security 資料保護原則暗示保護品質 (QOP)。

AMS 中的憑證驗證方法

您可以使用 Advanced Message Security 來偵測及拒絕已撤銷的憑證，以便使用不符合安全標準的憑證來保護佇列上的訊息。

AMS 可讓您使用「線上憑證狀態通訊協定 (OCSP)」或憑證撤銷清冊 (CRL) 來驗證憑證有效性。

AMS 可以配置為進行 OCSP 及/或 CRL 檢查。 如果同時啟用這兩種方法，基於效能考量， AMS 會先使用 OCSP 作為撤銷狀態。 在 OCSP 檢查之後，如果無法判斷憑證的撤銷狀態， AMS 會使用 CRL 檢查。

請注意，依預設會同時啟用 OCSP 和 CRL 檢查。

相關概念

第 545 頁的『AMS 中的線上憑證狀態通訊協定 (OCSP)』

「線上憑證狀態通訊協定 (OCSP)」會判斷憑證是否已撤銷，因此有助於判斷憑證是否可信任。 依預設會啟用 OCSP。

第 546 頁的『AMS 中的憑證撤銷清冊 (CRL)』

CRL 會保留憑證管理中心 (CA) 因各種原因 (例如，私密金鑰遺失或受損) 標示為不再受信任的憑證清單。

AMS 中的線上憑證狀態通訊協定 *(OCSP)*

「線上憑證狀態通訊協定 (OCSP)」會判斷憑證是否已撤銷，因此有助於判斷憑證是否可信任。 依預設會啟用 OCSP。

IBM i 系統不支援 OCSP。

在 *Advanced Message Security* 的原生攔截程式中啟用 *OCSP* 檢查
根據所使用憑證中的資訊，依預設會啟用 Advanced Message Security 中的「線上憑證狀態通訊協定 (OCSP)」檢查。

程序

將下列選項新增至金鑰儲存庫配置檔:

註: 所有 OCSP 段落都是選用的，且可以獨立指定。

選項	說明
ocsp.enable=off	如果要檢查的憑證具有「權限資訊存取 (AIA)」延伸，且 PKIX_AD_OCSP 存取方法包含「OCSP 回應端」所在的 URI，請啟用 OCSP 檢查。 可能的值: on 或 off。
ocsp.url=*responder_URL*	OCSP 回應端的 URL 位址。 如果省略此選項，則會停用非 AIA OCSP 檢查。
ocsp.http.proxy.host=*OCSP_proxy*	OCSP Proxy 伺服器的 URL 位址。 如果省略此選項，則 Proxy 不會用於非 AIA 線上憑證檢查。
ocsp.http.proxy.port=*port_number*	OCSP Proxy 伺服器的埠號。 如果省略此選項，則會使用預設埠 8080。
ocsp.nonce.generation=on/off	查詢 OCSP 時產生暫時性要求。 預設值是 off。
ocsp.nonce.check=on/off	收到 OCSP 的回應之後檢查暫時性要求。 預設值是 off。
ocsp.nonce.size=8	暫時性要求大小（以位元組為單位）。
ocsp.http.get=on/off	指定 HTTP GET 作為您的要求方法。 如果此選項設為 off，會使用 HTTP POST。 預設值為 off。
ocsp.max_response_size=20480	來自 OCSP 回應端的回應大小上限（以位元組為單位提供）。
ocsp.cache_size=100	啟用內部 OCSP 回應快取，並設定快取項目數的限制。
ocsp.timeout=30	伺服器回應的等待時間（秒），該時間過後，Advanced Message Security 便會逾時。
ocsp.unknown=ACCEPT	定義無法在逾時期間內呼叫到 OCSP 伺服器時的行為。 可能的值如下: • ACCEPT 容許憑證 • WARN 容許憑證並記載警告 • REJECT 防止使用憑證並記載錯誤

在 *AMS* 中啟用 *Java* 中的 *OCSP* 檢查
若要在 Advanced Message Security 中啟用 Java 的 OCSP 檢查，請修改 `java.security` 檔或金鑰儲存庫配置檔。

關於這項作業

在 Advanced Message Security 中啟用 OCSP 檢查有兩種方式:

使用 *java.security*
請檢查您的憑證是否包含「權限資訊存取 (AIA)」憑證延伸。

程序

1. 如果未設定 AIA 或您想要置換憑證，請使用下列內容編輯 `$JAVA_HOME/lib/security/java.security` 檔案:

   ```
   ocsp.responderURL=http://url.to.responder:port
   ocsp.responderCertSubjectName=CN=Example CA,O=IBM,C=US
   ```

   並使用下列行編輯 `$JAVA_HOME/lib/security/java.security` 檔案，以啟用 OCSP 檢查:

   ```
   ocsp.enable=true
   ```

2. 如果已設定 AIA，請編輯具有下列行的 `$JAVA_HOME/lib/security/java.security` 檔案，以啟用 OCSP 檢查:

   ```
   ocsp.enable=true
   ```

下一步

如果您使用 Java Security Manager，請將下列 Java 許可權新增至 `lib/security/java.policy`

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
```

使用 *keystore.conf*

程序

將下列屬性新增至配置檔:

```
ocsp.enable=true
```

重要: 在配置檔中設定這個屬性會置換 java.security 設定。

下一步

若要完成配置，請將下列 Java 許可權新增至 `lib/security/java.policy`:

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
permission java.security.SecurityPermission "setProperty.ocsp.enable";
```

AMS 中的憑證撤銷清冊 *(CRL)*

CRL 會保留憑證管理中心 (CA) 因各種原因 (例如，私密金鑰遺失或受損) 標示為不再受信任的憑證清單。

為了驗證憑證， Advanced Message Security 會建構一個憑證鏈，由簽章者的憑證和憑證管理中心 (CA) 的憑證鏈組成，直到信任錨點為止。 信任錨點是一個授信金鑰儲存庫檔，其中包含用來主張憑證信任的授信憑證或授信主要憑證。 AMS 會使用 PKIX 驗證演算法來驗證憑證路徑。 當建立並驗證鏈結時， AMS 會完成憑證驗證，其中包括根據現行日期來驗證鏈結中每一個憑證的發出及到期日，並檢查金鑰用法延伸是否存在於「終端實體」憑證中。 如果延伸附加至憑證， AMS 會驗證是否也設定 **digitalSignature** 或 **nonRepudiation** 。 如果沒有，則會報告並記載 `MQRC_SECURITY_ERROR` 。 接下來，視配置檔中指定的值而定， AMS 會從檔案或 LDAP 下載 CRL。 AMS 只支援以 DER 格式編碼的 CRL。 如果在金鑰儲存庫配置檔

中找不到 CRL 相關配置，則 AMS 不會執行 CRL 有效性檢查。 對於每一個 CA 憑證， AMS 會使用 CA 的「識別名稱」來查詢 LDAP 中的 CRL ，以尋找其 CRL。 LDAP 查詢中包含下列屬性:

```
certificateRevocationList,
certificateRevocationList;binary,
authorityRevocationList,
authorityRevocationList;binary
deltaRevocationList
deltaRevocationList;binary,
```

註: 只有在指定為分佈點時，才支援 `deltaRevocationList` 。

ULW 在原生攔截程式中啟用憑證驗證和憑證撤銷清冊支援

您必須修改金鑰儲存庫配置檔，讓 Advanced Message Security 可以從「輕量型目錄存取通訊協定 (LDAP)」伺服器下載 CLR。

開始之前

IBM i IBM i 上的 Advanced Message Security 不支援在原生攔截程式中啟用憑證驗證和憑證撤銷清冊支援。

z/OS 如需 Advanced Message Security 上的 z/OS ，請參閱 第 549 頁的『在 z/OS 上啟用憑證撤銷清冊 (CRL)』 。

程序

將下列選項新增至配置檔:

註: 所有 CRL 段落都是選用的，可以獨立指定。

選項	說明
`crl.ldap.host=`*`host_name`*	LDAP 伺服器主機名稱。
`crl.ldap.port=`*`port_number`*	LDAP 伺服器埠號。 在 UNIX, Linux, and Windows 系統上， 每個 AMS 原生攔截器只支援一個 LDAP CRL 伺服器。
`crl.cdp=off`	使用此選項來檢查或使用憑證中的 CRLDistributionPoints 延伸。
`crl.ldap.version=3`	LDAP 通訊協定版本號碼。 可能的值: 2 或 3。
`crl.ldap.user=cn=`*`username`*	登入 LDAP 伺服器。 如果未指定此值，則 LDAP 中的 CRL 屬性必須是全球可讀取的
`crl.ldap.pass=`*`password`*	LDAP 伺服器的密碼。
V 9.2.2 `crl.ldap.encrypted=`*`no/yes`*	`clr.ldap.pass` 是否已加密。 如需相關資訊，請參閱 保護 AMS 配置檔中的密碼 。
`crl.ldap.cache_lifetime=0`	LDAP 快取生命期限 (以秒為單位)。 可能的值: 0-86400。
`crl.ldap.cache_size=50`	LDAP 快取記憶體大小。 只有在 `crl.ldap.cache_lifetime` 值大於 0 時，才能指定此選項。
`crl.http.proxy.host=some.host.com`	用於 CDP CRL 擷取的 HTTP Proxy 伺服器埠。
`crl.http.proxy.port=8080`	HTTP Proxy 伺服器埠號。

選項	說明
crl.http.max_response_size=204800	CRL 的最大大小，以位元組為單位，可從 IBM Global Security Kit (GSKit) 接受的 HTTP 伺服器擷取。
crl.http.timeout=30	伺服器回應的等待時間 (以秒為單位) ，在此之後 AMS 會逾時。
crl.http.cache_size=0	HTTP 快取記憶體大小，位元組。
crl.unknown=ACCEPT	定義無法在逾時期間內呼叫到 CRL 伺服器時的行為。 可能的值如下: • ACCEPT 容許憑證 • WARN 容許憑證並記載警告 • REJECT 防止使用憑證並記載錯誤

在 *AMS* 的 *Java* 中啟用憑證撤銷清冊支援
若要在 Advanced Message Security (AMS) 中啟用 CRL 支援，您必須修改 keystore 配置檔案，允許 AMS 從輕量級目錄存取通訊協定 (LDAP) 伺服器下載 CRL，並配置 java.security 檔案。

程序

1. 將下列選項新增至配置檔:

標頭	說明
crl.ldap.host.*N*=*host_name*	LDAP 主機名稱，其中 *N* 是 1 到 9 的數字。 若要提供多個 CRL LDAP 伺服器，請設定多個 LDAP 主機名稱和連接埠號碼。
crl.ldap.port.*N*=*port_number*	LDAP 伺服器連接埠號碼，其中 *N* 為 1 到 9 的數字。 當 LDAP 連線失敗時，會使用多部 LDAP 主機來確保透通失效接手。 所有 LDAP 伺服器都應該是複本，並包含相同的資料。 AMS Java 拦截器成功连接到 LDAP 服务器时，不会尝试从其余可用服务器下载 CRL。 Java 不使用 和 值。 crl.ldap.user crl.ldaworldp.pass 當連接至 LDAP 伺服器時，它不會使用使用者和密碼。 因此，LDAP 中的 CRL 屬性必須是世界可讀的。
crl.cdp=on/off	使用此選項來檢查或使用憑證中的 CRLDistributionPoints 延伸。

2. 使用下列內容修改 JRE/lib/security/java.security 檔案:

內容名稱	說明
com.ibm.security.enableCRLDP	此內容採用下列值: true、 false。 如果設定為 true ，當進行憑證廢止檢查時，憑證廢止清冊會使用憑證廢止清冊分發點擴充欄位 URL 來定位。 如果設為 false 或未設定，則會停用使用 CRL 配送點延伸來檢查 CRL。

內容名稱	說明
ibm.security.certpath.ldap.cache.lifetime	此內容可用來將 LDAP CertStore 記憶體快取中的項目生命期限設為以秒為單位的值。 值為 0 時，會停用快取； -1 表示使用期限不受限制。 如果未設定，則預設生命期限為 30 秒。
com.ibm.security.enableAIAEXT	此內容採用下列值: true、 false。 如果設為 true，則會檢查在所建置憑證路徑的憑證內找到的任何「權限資訊存取」延伸，以判斷它們是否包含 LDAP URI。 對於找到的每一個 LDAP URI，會建立 LDAPCertStore 物件，並將其新增至 CertStores 集合，以用來尋找建置憑證路徑所需的其他憑證。 如果設為 false 或未設定，則不會建立其他 LDAPCertStore 物件。

z/OS 在 *z/OS* 上啟用憑證撤銷清冊 *(CRL)*

Advanced Message Security 支援憑證撤銷清冊 (CRL) 檢查用來保護資料訊息的數位憑證

關於這項作業

啟用後， Advanced Message Security 將在將訊息放置到受隱私權保護佇列時驗證收件者憑證，並在從受保護佇列擷取訊息時驗證傳送端憑證 (完整性或隱私權)。 在此情況下，驗證包括驗證相關憑證未登錄在相關 CRL 中。

Advanced Message Security 使用 IBM System SSL 服務來驗證傳送端和接收端憑證。 您可以在 z/OS Cryptographic Services System Secure Sockets Layer Programming 手冊中找到有關 System SSL 憑證驗證的詳細文件。

若要啟用 CRL 檢查，您可以在 AMS 位址空間的已啟動作業 JCL 中，透過 CRLFILE DD 指定 CRL 配置檔的位置。 *thlqual*.SCSQPROC (CSQ40CRL) 中提供可自訂的 CRL 配置檔範例。 此檔案中允許的設定如下:

表 *103: Advanced Message Security CRL* 配置變數

變數	有效值	說明
crl.ldap.host.*N*	*hostname-or-hostname: port*	管理發證者憑證 CRL 之 LDAP 伺服器的 ipaddr/hostname。 如果您沒有指定 LDAP 伺服器的連接埠號，則會使用 crl.ldap.port 指定的連接埠號。 您最多可以指定 10 個 CRL LDAP 伺服器主機名稱，如這裡所述。
crl.ldap.port	埠	LDAP 伺服器的 TCP/IP 埠號。
crl.ldap.user	*LDAP* 使用者	連接至 LDAP 伺服器時要使用的 LDAP 使用者名稱。
crl.ldap.pass	*LDAP* 密碼	與 crl.ldap.user 相關聯的 LDAP 密碼。

您可以指定多個 LDAP 伺服器主機名稱及埠，如下所示:

```
crl.ldap.host.1 = hostname -or hostname:port
crl.ldap.host.2 = hostname -or hostname:port
crl.ldap.host.3 = hostname -or hostname:port
```

您最多可以指定 10 個主機名稱。 如果您未指定 LDAP 伺服器的埠號，則會使用 crl.ldap.port 指定的埠號。 每一部 LDAP 伺服器必須使用相同的 crl.ldap.user/password 組合來進行存取。

指定 CRLFILE DD 時，會在起始設定 Advanced Message Security 位址空間期間載入配置，並啟用 CRL 檢查。 如果未指定 CRLFILE DD ，或 CRL 配置檔無法使用或無效，則會停用 CRL 檢查。

AMS 使用 IBM System SSL 憑證驗證服務來執行 CRL 檢查，如下所示:

表 *104: Advanced Message Security CRL* 檢查		
作業	**保護品質**	**已檢查憑證**
PUT	隱私權	收件者
GET	完整性/隱私權	傳送端

如果訊息作業失敗， CRL 檢查 Advanced Message Security 會執行下列動作:

表 *105: Advanced Message Security CRL* 檢查失敗行為	
作業	**CRL 檢查失敗**
PUT	訊息未放入目標佇列。 完成碼 MQCC_FAILED 及原因碼 MQRC_SECURITY_ERROR 會傳回給應用程式。
GET	訊息會從目標佇列中移除，並移至系統保護錯誤佇列。 完成碼 MQCC_FAILED 及原因碼 MQRC_SECURITY_ERROR 會傳回給應用程式。

AMS for z/OS 使用 IBM System SSL 服務來驗證憑證，其中包括 CRL 及信任檢查。

IBM MQ 使用安全設定，其中憑證驗證需要 LDAP 伺服器可連接，但不需要定義 CRL。

註: 管理者負責確保相關 LDAP 服務可用，並維護相關憑證管理中心的 CRL 項目。

V 9.2.0 V 9.2.0 設定 配置檔 的 AMS 密碼保護

將金鑰儲存庫及私密金鑰密碼儲存為純文字會造成安全風險，因此 Advanced Message Security 提供的工具可以使用使用者的金鑰來編碼這些密碼。

開始之前

`keystore.conf` 檔案擁有者必須確定只有檔案擁有者有權讀取及寫入檔案。 本主題中說明的密碼保護只是額外的保護措施。 此外，您應該在安全系統上執行此程序。

V 9.2.2 對於將要讀取配置檔的 AMS 用戶端類型，請確保使用正確的 **runamscred** 變式。 如果 AMS 用戶端是:

- Java 用戶端，您應該使用 Java **runamscred** 指令，其位於 `<IBM MQ installation root>/java/bin`
- MQI 用戶端，您應該使用位於 `<IBM MQ installation root>/bin` 中的 MQI **runmqascred** 指令。

程序

1. 編輯 `keystore.conf` 檔案，以併入所有必要資訊，包括需要保護的密碼。

```
jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = no
jceks.keystore_pass = passw0rd
jceks.key_pass = passw0rd
jceks.provider = IBMJCE
```

2. 將加密金鑰放置在保護 `keystore.conf` 檔案。

   V 9.2.2 此金鑰必須與稍後將由 AMS 用戶端使用的金鑰相同:

```
ThisIsAnExampleEncryptionKey
```

3. 執行 **runamscred** 指令，以保護提供加密金鑰檔的 keystore.conf 檔案。

```
runamscred -f <location of keystore.conf> -sf <location of encryption keyfile>
```

4. 驗證 keystore.conf 檔案已受保護且包含已加密密碼。

範例

下列範例顯示受保護 keystore.conf 檔案的外觀:

```
jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = yes
jceks.keystore_pass =
<AMS>1!62K/a4RinT+bks4RjFWx4A==!Vhi/RjIN2FH5qStUJ/0hsgKyn2IdMuhanemRRDrJq
HM=
jceks.key_pass =
<AMS>1!qmnxY++rsOUtZfDSgwcR1g==!VmWVREdVkNp1xYJstvuW64ph5vxxf7SPoqtsXxYh2
Tk=
jceks.provider = IBMJCE
```

相關資訊

runamscred: 保護 AMS 關鍵字

z/OS 在 z/OS 上搭配使用憑證與 AMS

關於這項作業

Advanced Message Security 實作三個層次的保護: 完整性、機密性和隱私權。

使用完整性原則，會使用發送端 (執行 MQPUT 的應用程式) 的私密金鑰來簽署訊息。 完整性提供訊息修改的偵測，但訊息文字本身未加密。

當使用機密性原則時，訊息會在放入佇列時加密。 訊息會使用相關 Advanced Message Security 原則中指定的對稱金鑰及演算法來加密。 對稱金鑰本身會以每一個收件者 (執行 MQGET 的應用程式) 的公開金鑰來加密。 公開金鑰與儲存在金鑰環中的憑證相關聯。

使用隱私權原則時，訊息會同時簽署及加密。

當執行 MQGET 的收件者應用程式將受隱私權保護的訊息移出佇列時，必須將訊息解密。 因為它是使用收件者的公開金鑰來加密，所以必須使用在金鑰環中找到的收件者私密金鑰來解密。

z/OS 搭配使用 *SAF* 金鑰環與 *z/OS* 上的 *AMS*

Advanced Message Security (AMS) 利用 z/OS SAF 金鑰環服務來定義和管理簽署和加密所需的憑證。 如果提供相同層次的支援，則可以使用功能相當於 RACF 的安全產品，而不使用 RACF 。

有效使用金鑰環可以減少管理憑證所需的管理。

在產生 (或匯入) 憑證之後，它必須連接至金鑰環，才能變成可存取。 相同的憑證可以連接至多個金鑰環。

Advanced Message Security 使用兩組金鑰環。 一組由產生或接收訊息的個別使用者 ID 所擁有的金鑰環組成。 每一個金鑰環都包含與擁有使用者 ID 的憑證相關聯的私密金鑰。 每一個憑證的私密金鑰用來簽署受完整性保護或受隱私權保護佇列的訊息。 它也用來在接收訊息時，從受保護隱私權或受機密性保護的佇列中解密訊息。

另一組是 AMS 位址空間使用者所擁有的單一金鑰環。 它包含驗證訊息發送端及收件者的憑證所需的簽署 CA 憑證鏈。

使用隱私權或機密性保護時， AMS 位址空間使用者所擁有的金鑰環也包含訊息收件者的憑證。 這些憑證中的公開金鑰用來加密將訊息放入受保護佇列時用來加密訊息資料的對稱金鑰。 擷取這些訊息時，會使用相關收件者的私密金鑰來解密對稱金鑰，然後使用對稱金鑰來解密訊息資料。

在搜尋憑證和私密金鑰時， Advanced Message Security 會使用金鑰環名稱 **drq.ams.keyring** 。 這是使用者和 AMS 位址空間金鑰環的情況。

如需憑證和金鑰環及其在資料保護中的角色的圖解和進一步說明，請參閱 憑證相關作業的摘要。

用於簽署的私密金鑰可以具有任何標籤，但必須連接作為預設憑證。 在 APAR PH44820 之前，用於解密的私密金鑰可以具有任何標籤，但必須連接作為預設憑證。 當套用 APAR PH44820 時，用於解密的私密金鑰可以具有任何標籤，且必須連接至金鑰環，但不再需要連接作為預設憑證。

數位憑證和金鑰環主要在 RACF 中使用 RACDCERT 指令來管理。

如需憑證、標籤及 RACDCERT 指令的相關資訊，請參閱 *z/OS: Security Server RACF Command Language Reference* 及 *z/OS: Security Server RACF Security Administrator 's Guide*。

z/OS 取代憑證

當更新或取代憑證時 (例如，當現有憑證接近其到期日時) ，不一定可以從已在受機密性或隱私權原則保護之佇列上的現有訊息中移除保護。

當憑證為:

- 以相同的私密金鑰更新，且重新發出的憑證已取代原始憑證
- 以新的私密金鑰重新建立索引，且 RACDCERT 輪替指令已刪除原始私密金鑰

在 APAR PH44820 之前，當新憑證連接至使用者的金鑰環作為預設憑證時，無法再解密使用舊憑證加密的訊息。 當套用 APAR PH44820 時，如果必要的憑證連接至使用者的金鑰環，則會將訊息解密; 不再需要連接作為預設值。 當連接新憑證時，這可讓已在佇列中的訊息順利解密。

下列範例顯示在套用 APAR PH44820 時如何根據現有憑證產生新憑證:

- 會根據現有憑證來建立新的憑證，並使用新的公開/私密金鑰組。
- 新憑證由發行機構簽署。
- 舊憑證的公開金鑰會從 AMS 位址空間的金鑰環中移除，並新增新憑證的公開金鑰。
- 除了舊憑證之外，還會將新的憑證和私密金鑰新增至使用者的金鑰環。

```
RACDCERT ID(user1) REKEY(LABEL('user1'))          -
         WITHLABEL('user1new')

RACDCERT GENREQ(LABEL('user1new')) ID(user1)      -
         DSN(output_data_set_name)

RACDCERT GENCERT(output_data_set_name) ID(user1)  -
         SIGNWITH(CERTAUTH LABEL('AMSCA'))

RACDCERT ID(user1) ALTER (LABEL('user1new'))      -
         TRUST

RACDCERT ID(WMQAMSD) REMOVE(ID(user1)             -
         LABEL('user1')                           -
         RING(drq.ams.keyring)  )

RACDCERT ID(WMQAMSD) CONNECT(ID(user1)            -
         LABEL('user1new') USAGE(SITE)            -
         RING(drq.ams.keyring)  )

RACDCERT ID(user1) CONNECT(ID(user1)              -
         LABEL('user1new') USAGE(PERSONAL)        -
         RING(drq.ams.keyring) DEFAULT )
```

如需憑證、標籤及 RACDCERT 指令的相關資訊，請參閱 *z/OS: Security Server RACF Command Language Reference* 及 *z/OS: Security Server RACF Security Administrator 's Guide*。

z/OS 在 *z/OS* 上授權存取 *AMS* 的 *RACDCERT* 指令

使用 RACDCERT 指令的授權是一項後置安裝作業，應該已由 z/OS 系統程式設計師完成。 這項作業涉及將相關許可權授與 Advanced Message Security 安全管理者。

作為摘要，需要這些指令才能容許存取 RACF RACDCERT 指令:

```
RDEFINE FACILITY IRR.DIGTCERT.* UACC(NONE)
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID( admin ) ACCESS(CONTROL)
SETROPTS RACLIST(FACILITY) REFRESH
```

在此範例中， *admin* 指定安全管理者的使用者 ID ，或您要使用 RACDCERT 指令的任何使用者。

z/OS 在 *z/OS* 上為 *AMS* 使用者建立憑證和金鑰環

本節記載使用 RACF 憑證管理中心 (CA) 為 Advanced Message Security (AMS) 的 z/OS 使用者建立必要的憑證和金鑰環所需的步驟。

解決在 z/OS 上使用 Advanced Message Security 時憑證的問題

如果金鑰儲存庫中的憑證和遺漏項目有問題，您可以啟用 GSKIT 追蹤。

在 AMS 啟動型作業程序中的 ENVARS DD 所參照的檔案中，新增:

```
GSK_TRACE_FILE=/u/... /gsktrace
GSK_TRACE=0xff
```

如需相關資訊，請參閱 環境變數 。

每次存取金鑰儲存庫時，都會將資料寫入 GSK_TRACE_FILE 中指定的追蹤檔。

若要格式化追蹤檔，請使用下列指令:

```
gsktrace inputtrace file > output_file
```

範例情節

傳送端應用程式和接收端應用程式的實務範例用來說明必要的步驟。

在下列範例中， user1 是訊息的發送端，而 user2 是收件者。 Advanced Message Security 位址空間的使用者 ID 是 WMQAMSD。

此處所示範例中的所有命令均由管理使用者 ID admin 從 ISPF 選項 6 發出。

z/OS 在 *z/OS* 上定義 *AMS* 的本端憑證管理中心憑證

如果您使用 RACF 作為 CA ，則必須建立憑證管理中心憑證 (如果您尚未這麼做的話)。 這裡顯示的指令會建立憑證管理中心 (或簽章者) 憑證。 此範例會建立一個稱為 AMSCA 的憑證，以在建立反映 Advanced Message Security 使用者及應用程式身分的後續憑證時使用。

可以修改此指令 (特別是 SUBJECTSDN) ，以反映安裝時使用的命名結構及慣例:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('AMSCA') O('ibm') C('us'))
KEYUSAGE(CERTSIGN) WITHLABEL('AMSCA')
```

註: 使用此本端憑證管理中心憑證簽署的憑證會在使用 RACDCERT LIST 指令列出時顯示 CN=AMSCA , O=ibm , C=us 的發證者。

z/OS 在 *z/OS* 上使用 *AMS* 的私密金鑰建立數位憑證

必須為每一個 Advanced Message Security 使用者產生具有私密金鑰的數位憑證。 在這裡顯示的範例中，RACDCERT 指令用來產生 user1 和 user2 的憑證，這些憑證是以標籤 AMSCA 所識別的本端 CA 憑證來簽署。

```
RACDCERT ID(user1) GENCERT SUBJECTSDN(CN('user1') O('ibm') C('us'))
WITHLABEL('user1') SIGNWITH(CERTAUTH LABEL('AMSCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

RACDCERT ID(user2) GENCERT SUBJECTSDN(CN('user2') O('ibm') C('us'))
WITHLABEL('user2') SIGNWITH(CERTAUTH LABEL('AMSCA'))
```

```
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

RACDCERT ID(user1) ALTER (LABEL('user1')) TRUST
RACDCERT ID(user2) ALTER (LABEL('user2')) TRUST
```

需要 RACDCERT ALTER 指令，才能將 TRUST 屬性新增至憑證。 第一次使用此程序建立憑證時，其有效日期範圍與簽署憑證不同。 因此， RACF 會將它標示為 NOTRUST ，這表示不會使用憑證。 請使用 RACDCERT ALTER 指令來設定 TRUST 屬性。

必須針對 Advanced Message Security 使用的憑證指定 KEYUSAGE 屬性 HANDSHAKE、DATAENCRYPT 及 DOCSIGN。

表 *106: RACDCERT KEYUSAGE* 值及指示器

KEYUSAGE 值	**指示器集**
信號交換	digitalSignature 和 keyEncipherment
DATAENCRYPT	dataEncipherment
Docsign	nonRepudiation
CERTSIGN	keyCertSign 和 cRLSign

z/OS 在 *z/OS* 上建立 *AMS* 的 *RACF* 金鑰環

這裡顯示的指令會為 RACF 定義的使用者 ID user1、 user2 及 Advanced Message Security 位址空間作業使用者 WMQAMSD 建立金鑰環。 金鑰環名稱由 Advanced Message Security 修正，且必須依所示進行編碼，不含引號。 此值區分大小寫。

```
RACDCERT ID(user1) ADDRING(drq.ams.keyring)
RACDCERT ID(user2) ADDRING(drq.ams.keyring)
RACDCERT ID(WMQAMSD) ADDRING(drq.ams.keyring)
```

z/OS 將憑證連接至 *z/OS* 上 *AMS* 的金鑰環

將使用者及 CA 憑證連接至金鑰環:

```
RACDCERT ID(WMQAMSD) CONNECT(CERTAUTH LABEL('AMSCA')
RING(drq.ams.keyring))
RACDCERT ID(user1) CONNECT(ID(user1) LABEL('user1')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
RACDCERT ID(user2) CONNECT(ID(user2) LABEL('user2')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
RACDCERT ID(WMQAMSD) CONNECT(ID(user2) LABEL('user2')
RING(drq.ams.keyring) USAGE(SITE))
```

在 APAR PH44820 之前，包含用於解密之私密金鑰的憑證必須連接至使用者的金鑰環作為預設憑證。 當套用 APAR PH44820 時，任何包含用於解密的私密金鑰或金鑰的憑證都必須連接至使用者的金鑰環，不過它們不再需要連接作為預設憑證。

RACDCERT USAGE (SITE) 屬性可防止在金鑰環中存取私密金鑰，而 RACDCERT USAGE (PERSONAL) 屬性則容許使用私密金鑰 (如果存在的話)。 User2 的憑證必須連接至 Advanced Message Security 位址空間金鑰環，因為在將訊息放入佇列時，需要其公開金鑰來加密訊息。 USAGE (SITE) 限制暴露 user2 的私密金鑰。

具有標籤 AMSCA 的 CERTAUTH 憑證必須連接至 Advanced Message Security 位址空間金鑰環，因為它是用來簽署訊息發送端 user1 的憑證。 它用來驗證 user1 的簽署憑證。

z/OS *z/OS* 上 *AMS* 的金鑰環驗證

在輸入所有指令之後，金鑰環應該如這裡所示:

```
RACDCERT ID(user1) LISTRING(drq.ams.keyring)
Digital ring information for user USER1:
Ring:>drq.ams.keyring<:

Certificate Label Name           Cert Owner   USAGE    DEFAULT
```

```
------------------------------- ------------ -------- -------
user1                           ID(USER1)    PERSONAL YES

RACDCERT ID(user2) LISTRING(drq.ams.keyring)
Digital ring information for user USER2:
Ring:>drq.ams.keyring<:

Certificate Label Name          Cert Owner   USAGE    DEFAULT
------------------------------- ------------ -------- -------
user2                           ID(USER2)    PERSONAL YES

RACDCERT ID(WMQAMSD) LISTRING(drq.ams.keyring)
Digital ring information for user WMQAMSD:
Ring:>drq.ams.keyring<:

Certificate Label Name          Cert Owner   USAGE    DEFAULT
------------------------------- ------------ -------- -------
AMSCA                           CERTAUTH     CERTAUTH NO
user2                           ID(USER2)    SITE     NO
```

列出個別憑證也會顯示環關聯。

```
RACDCERT ID(user2) LIST(label('user2'))
Digital certificate information for user USER2:

***
Label: user2
Certificate ID: 2QfH8Pny9/LzpKKFmfFA
Status: TRUST
Start Date: 2010/05/03 22:59:53
End Date:  2011/05/04 22:59:52
Serial Number:>15<:
Issuer's Name:>OU=AMSCA.O=ibm.C=us<:
Subject's Name:>CN=user2.O=ibm.C=us<:
Key Usage: HANDSHAKE, DATAENCRYPT, DOCSIGN
Private Key Type: Non-ICSF
Private Key Size: 1024
Ring Associations:
Ring Owner: USER2
Ring:>drq.ams.keyring<:
Ring Owner: WMQAMSD
Ring:>drq.ams.keyring<:
```

為了增進效能，會在位址空間的生命期限內快取與 AMS 位址空間相關聯的 drq.ams.keyring 內容。 該金鑰環的變更不會自動生效。 管理者可以透過下列任一方式重新整理快取:

- 停止並重新啟動佇列管理程式。
- 使用 z/OS MODIFY 指令:

```
F qmgrAMSM,REFRESH KEYRING
```

相關工作

操作 Advanced Message Security

z/OS *z/OS 上 AMS 的憑證相關作業摘要*

第 556 頁的圖 35 說明傳送及接收應用程式與相關憑證之間的關係。 所說明的實務範例涉及使用資料保護隱私權原則在兩個 z/OS 佇列管理程式之間進行遠端佇列作業。 在 第 556 頁的圖 35 中， "AMS" 表示 " Advanced Message Security"。

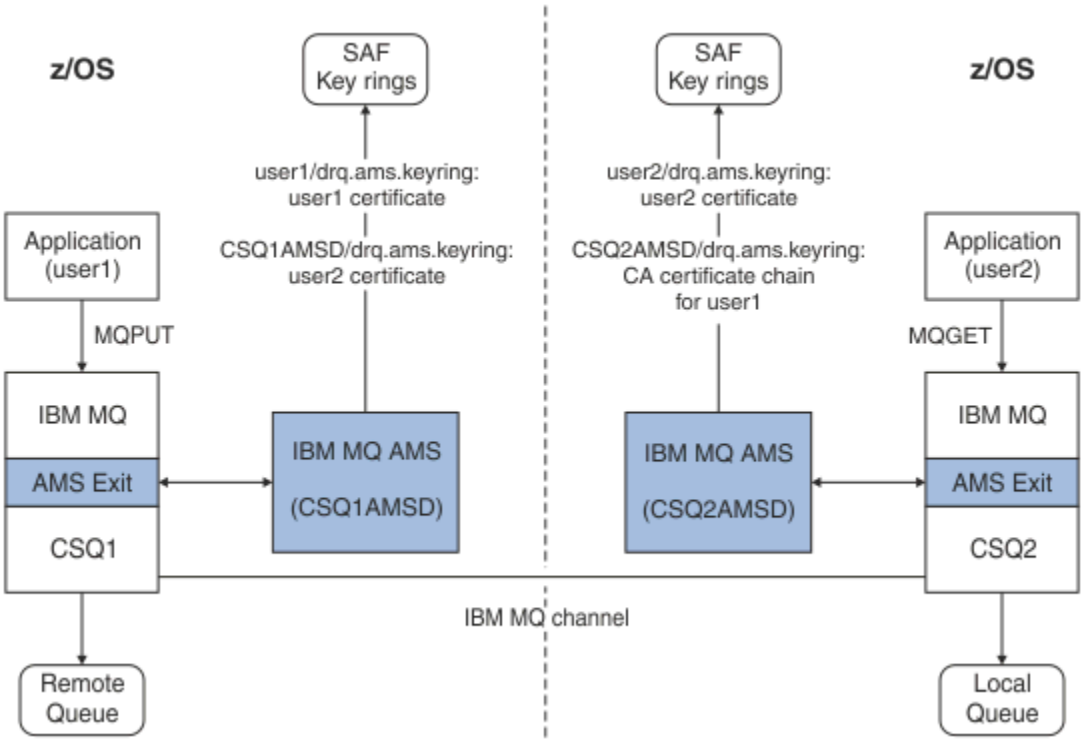


圖 *35:* 應用程式與憑證關係

在此圖中，以 'user1' 身分執行的應用程式會將訊息放到佇列管理程式 CSQ1 所管理的遠端佇列中，預期由以 'user2' 身分執行的應用程式從佇列管理程式 CSQ2 所管理的本端佇列中擷取。 此圖表假設 Advanced Message Security 隱私權原則，這表示訊息已簽署且已加密。

當放置發生時， Advanced Message Security 會截取訊息，並使用 user2 的憑證 (儲存在 AMS 位址空間使用者金鑰環中) 來加密用來加密訊息資料的對稱金鑰。

請注意， user2 的憑證已使用選項 USAGE (SITE) 連接至 AMS 位址空間使用者金鑰環。 這表示 AMS 位址空間使用者可以存取憑證和公開金鑰，但不能存取私密金鑰。

在接收端， Advanced Message Security 會截取 user2 所發出的 get ，並使用 user2 的憑證來解密對稱金鑰，以便它可以解密訊息資料。 然後，它會使用儲存在 AMS 位址空間使用者金鑰環中的 user1 憑證的 CA 憑證鏈來驗證 user1 的簽章。

在此情況下，如果具有完整性的資料保護原則，則不需要 user2 的憑證。

若要使用 Advanced Message Security 在具有隱私權或完整性之訊息保護原則的 IBM MQ 受保護佇列上移入訊息， Advanced Message Security 必須具有下列資料項目的存取權:

- 使用者將訊息移入佇列的 X.509 V2 或 V3 憑證及私密金鑰。
- 用來簽署所有訊息簽章者的數位憑證的憑證鏈。
- 如果資料保護原則是隱私權，則預期收件者的 X.509 V2 或 V3 憑證。 預期的收件者會列在與佇列相關聯的 Advanced Message Security 原則中。

對於在 z/OS 上執行的程序及應用程式， Advanced Message Security 必須在兩個位置具有憑證:

- 在與傳送端應用程式 (將受保護訊息移入佇列的應用程式) 或接收端應用程式 (如果使用隱私權) 的 RACF 身分相關聯的 SAF 管理金鑰環中。

  Advanced Message Security 所尋找的憑證是預設憑證，且必須包含私密金鑰。 Advanced Message Security 假設傳送應用程式的 z/OS 使用者身分。 也就是說，它會作為代理，因此它可以存取使用者的私密金鑰。
- 在與 AMS 位址空間使用者相關聯的 SAF 管理金鑰環中。

傳送受隱私權保護的訊息時，此金鑰環包含訊息收件者的公開金鑰憑證。 接收訊息時，它包含驗證訊息傳送端簽章所需的「憑證管理中心」憑證鏈。

先前顯示的範例已使用 RACF 作為本端 CA。 不過，您可以在安裝時使用另一個 PKI 提供者 (憑證管理中心)。 如果您打算使用另一個 PKI 產品，請記住，私密金鑰和憑證必須匯入至與 Advanced Message Security 所保護之 z/OS RACF 使用者 ID IBM MQ 訊息相關聯的金鑰環中。

您可以使用 RACF RACDCERT 指令作為產生憑證申請的機制，憑證申請可以匯出及傳送至您選擇要發出的 PKI 提供者。

以下是憑證相關步驟的摘要:

1. 要求建立 CA 憑證，其中 RACF 是本端 CA。 如果您使用另一個 PKI 提供者，請省略此步驟。
2. 產生 CA 所簽署的使用者憑證。
3. 建立使用者及 Advanced Message Security AMS 位址空間 ID 的金鑰環。
4. 使用預設屬性將使用者憑證連接至使用者金鑰環。
5. 使用用法 (網站) 屬性將收件者憑證連接至 Advanced Message Security AMS 位址空間使用者金鑰環 (只有最終將成為受隱私權保護訊息收件者的使用者憑證才需要此步驟)。
6. 將訊息傳送者的 CA 憑證鏈連接至 Advanced Message Security AMS 位址空間使用者金鑰環。 (只有將驗證傳送端簽章的 AMS 作業才需要此步驟。)

z/OS 配置 *AMS* 的非 *z/OS* 常駐 *PKI*

Advanced Message Security for z/OS，在保護-處理放置在 IBM MQ 佇列上或從佇列接收的訊息時，使用 X.509 V3 數位憑證。 Advanced Message Security 本身不會建立或管理這些憑證的生命週期; 該功能由公開金鑰基礎架構 (PKI) 提供。 本出版品中說明如何使用憑證的範例使用 z/OS Security Server RACF 來填寫憑證申請。

不論是否使用 z/OS 或非 z/OS 常駐 PKI， AMS for z/OS 只會使用 RACF 或其對等項目所管理的金鑰環。 這些金鑰環是以「安全授權機能 (SAF)」為基礎，並且是 AMS for z/OS 用來擷取憑證的儲存庫，這些憑證是放置在 IBM MQ 佇列上或從佇列接收之訊息的發送端和接收端。

對於源自 z/OS 且受完整性或加密原則保護的訊息，原始使用者 ID 的憑證和私密金鑰必須儲存在與訊息發送端的 z/OS 使用者 ID 相關聯的 SAF 管理金鑰環中。

RACF 包括將憑證及私密金鑰匯入至 RACF 管理的金鑰環的功能。 如需如何將憑證載入 RACF 受管理金鑰環的詳細資料及範例，請參閱 z/OS Security Server RACF 出版品。

如果您的安裝使用其中一個支援的 PKI 產品，請參閱產品隨附的出版品，以取得如何部署它的相關資訊。

管理 Advanced Message Security 安全原則

Advanced Message Security 使用安全原則來指定加密及簽章演算法，以加密及鑑別流經佇列的訊息。

AMS 的安全原則概觀

Advanced Message Security 安全原則是概念性物件，說明訊息加密及簽署的方式。

如需安全原則屬性的詳細資料，請參閱下列子主題:

相關概念

第 561 頁的『AMS 中的保護品質』
Advanced Message Security 資料保護原則暗示保護品質 (QOP)。

第 560 頁的『AMS 中的安全原則屬性』
您可以使用 Advanced Message Security 來選取特定的演算法或方法，以保護資料。

AMS 中的原則名稱

原則名稱是識別特定 Advanced Message Security 原則及其套用至的佇列的唯一名稱。

原則名稱必須與其套用的佇列名稱相同。 Advanced Message Security (AMS) 之間有一對一對映 原則及佇列。

透過建立與佇列同名的原則，您可以啟動該佇列的原則。 沒有相符原則名稱的佇列不受 AMS 保護。

原則的範圍與本端佇列管理程式及其佇列相關。 對於遠端佇列管理程式所管理的佇列，遠端佇列管理程式必須有自己的本端定義原則。

AMS 中的簽章演算法

簽章演算法指出簽署資料訊息時應使用的演算法。

有效的值包括：

- MD5
- SHA-1
- SHA-2 系列(F)：
  - SHA256
  - SHA384 (可接受的金鑰長度下限-768 位元)
  - SHA512 (可接受的金鑰長度下限-768 位元)

不指定簽章演算法或指定演算法 NONE 的原則，意味著不會簽署放置在與原則相關聯的佇列上的訊息。

註: 用於訊息放置及取得功能的保護品質必須相符。 如果佇列與佇列中的訊息之間有原則保護品質不符，則不會接受訊息並傳送至錯誤處理佇列。 此規則同時適用於本端及遠端佇列。

AMS 中的加密演算法

加密演算法指出在加密放置在與原則相關聯的佇列上的資料訊息時應該使用的演算法。

有效的值包括：

- RC2
- DES
- 3DES
- AES128
- AES256

不指定加密演算法或指定演算法 NONE 的原則表示不會加密放置在與原則相關聯的佇列上的訊息。

請注意，指定 NONE 以外的加密演算法的原則也必須至少指定一個「收件者 DN」及簽章演算法，因為 Advanced Message Security 加密訊息也已簽署。

重要: 用於訊息放置及取得功能的保護品質必須相符。 如果佇列與佇列中的訊息之間有原則保護品質不符，則不會接受訊息並傳送至錯誤處理佇列。 此規則同時適用於本端及遠端佇列。

AMS 中的容錯

容錯屬性指出 Advanced Message Security 是否可以接受未指定安全原則的訊息。

從具有加密訊息原則的佇列擷取訊息時，如果訊息未加密，則會將訊息傳回給呼叫端應用程式。 有效的值包括：

0
否 (**default**)。

1
是。

未指定容錯值或指定 0 的原則，意味著放置在與原則相關聯之佇列上的訊息必須符合原則規則。

容錯是選用的，用於協助進行配置轉出，其中原則已套用至佇列，但那些佇列已包含未指定安全原則的訊息。

AMS 中的寄件者識別名稱

傳送端識別名稱 (DN) 可識別獲授權將訊息放置在佇列上的使用者。 在將訊息放入佇列之前，傳送端會使用其憑證來簽署訊息。

Advanced Message Security (AMS) 在擷取訊息之前，不會檢查有效使用者是否已將訊息放置在受資料保護的佇列上。 此時，如果原則規定一個以上有效的傳送端，且將訊息放置在佇列上的使用者不在有效傳送端清單中，則 AMS 會將錯誤傳回給接收端應用程式，並將訊息放置在 AMS 錯誤佇列上。

原則可以指定 0 個以上的傳送端 DN。 如果未指定原則的傳送端 DN ，則只要傳送端的憑證是授信的，任何傳送端都可以將受資料保護的訊息放入佇列。 將公用憑證新增至接收端應用程式可用的金鑰儲存庫，即可信任傳送端的憑證。

傳送端識別名稱的格式如下:

```
CN=Common Name,O=Organization,C=Country
```

重要:

- 所有「DN 元件」名稱都必須大寫。 DN 中的所有元件名稱 ID 必須依下表中顯示的順序來指定:

元件名稱	值
CN	此 DN 物件的通用名稱，例如完整名稱或裝置的預期用途。
OU	DN 物件所屬組織內的單位，例如公司部門或產品名稱。
O	DN 物件所屬的組織，例如公司。
L	DN 物件所在的地區 (城市或自治市)。
ST	DN 物件所在的州/省 (縣/市) 名稱。
C	識別名稱 (DN) 物件所在的國家/地區。

- 如果為原則指定一個以上的傳送端 DN，則只有那些使用者可以將訊息放在與該原則相關聯的佇列上。
- 指定的傳送端 DN 必須完全符合與放置訊息之使用者相關聯的數位憑證所包含的 DN。
- AMS 僅支援具有 Latin-1 字集值的 DN。 若要使用集合的字元來建立 DN ，您必須先使用開啟或使用 **strmqikm** GUI 的 UTF-8 編碼 AIX and Linux ，來建立使用 UTF-8 編碼所建立的 DN 的憑證。 然後，您必須從開啟 UTF-8 編碼的 Linux 或 AIX 平台建立原則，或使用 AMS 外掛程式來 IBM MQ。
- AMS 用來將傳送端名稱從 x.509 格式轉換為 DN 格式的方法，一律使用 ST = 代表州/省 (縣/市) 值。
- 下列特殊字元需要跳出字元:

```
, (comma)
+ (plus)
" (double quote)
\ (backslash)
< (less than)
> (greater than)
; (semicolon)
```

- 如果「識別名稱」包含內含的空白，您應該以雙引號括住 DN。

相關概念

第 559 頁的『AMS 中的收件者識別名稱』

收件者識別名稱 (DN) 可識別獲授權從佇列擷取訊息的使用者。

AMS 中的收件者識別名稱

收件者識別名稱 (DN) 可識別獲授權從佇列擷取訊息的使用者。

原則可以指定零個以上的接收端 DN。 收件者識別名稱具有下列格式:

```
CN=Common Name,O=Organization,C=Country
```

重要:

- 所有「DN 元件」名稱都必須大寫。 DN 中的所有元件名稱 ID 必須依下表中顯示的順序來指定:

元件名稱	值
CN	此 DN 物件的通用名稱，例如完整名稱或裝置的預期用途。
OU	DN 物件所屬組織內的單位，例如公司部門或產品名稱。
O	DN 物件所屬的組織，例如公司。
L	DN 物件所在的地區 (城市或自治市)。
ST	DN 物件所在的州/省 (縣/市) 名稱。
C	識別名稱 (DN) 物件所在的國家/地區。

- 如果沒有為原則指定任何接收端 DN，則任何使用者都可以從與該原則相關聯的佇列中取得訊息。
- 如果為原則指定一個以上的接收端 DN，則只有那些使用者可以從與該原則相關聯的佇列中取得訊息。
- 指定的接收端 DN 必須完全符合與取得訊息之使用者相關聯的數位憑證所包含的 DN。
- Advanced Message Security 僅支援具有 Latin-1 字集值的 DN。 若要使用集合的字元來建立 DN，您必須先使用 UTF-8 編碼建立的 DN，並使用 AIX 或 Linux 開啟 UTF-8 編碼或使用 **strmqikm** GUI。 然後，您必須從已開啟 UTF-8 編碼的 Linux 或 AIX 平台建立原則，或使用 IBM MQ 的 Advanced Message Security 外掛程式。

相關概念

第 558 頁的『AMS 中的寄件者識別名稱』

傳送端識別名稱 (DN) 可識別獲授權將訊息放置在佇列上的使用者。 在將訊息放入佇列之前，傳送端會使用其憑證來簽署訊息。

AMS 中的安全原則屬性

您可以使用 Advanced Message Security 來選取特定的演算法或方法，以保護資料。

安全原則是一個概念性物件，說明訊息加密及簽署的方式。

表 *107: AMS* 中的安全原則屬性	
屬性	**說明**
原則名稱	佇列管理程式的原則唯一名稱。
簽章演算法	傳送之前用來簽署訊息的加密演算法。
加密演算法	在傳送之前用來加密訊息的加密演算法。
收件者清單	訊息潛在接收端的憑證識別名稱 (DN) 清單。
簽章 DN 核對清單	要在訊息擷取期間驗證的簽章 DN 清單。

在 Advanced Message Security 中，訊息會使用對稱金鑰來加密，而對稱金鑰會使用收件者的公開金鑰來加密。 公開金鑰使用 RSA 演算法進行加密，有效長度最多為 2048 位元的金鑰。 實際非對稱金鑰加密取決於憑證金鑰長度。

支援的對稱金鑰演算法如下:

- RC2
- DES
- 3DES
- AES128
- AES256

Advanced Message Security 也支援下列加密雜湊函數:

- MD5

- SHA-1
- SHA-2 系列(F):
  - SHA256
  - SHA384 (可接受的金鑰長度下限-768 位元)
  - SHA512 (可接受的金鑰長度下限-768 位元)

註: 用於訊息放置及取得功能的保護品質必須相符。 如果佇列與佇列中的訊息之間有原則保護品質不符，則不會接受訊息並傳送至錯誤處理佇列。 此規則同時適用於本端及遠端佇列。

AMS 中的保護品質

Advanced Message Security 資料保護原則暗示保護品質 (QOP)。

Advanced Message Security 中的三個保護品質層次由 IBM MQ 9.0 及更新版本中的第四個層次補充，且全都取決於用來簽署及加密訊息的加密演算法:

- 隱私權-必須簽署並加密放置在佇列上的訊息。
- 完整性-放置在佇列上的訊息必須由傳送者簽署。
- 機密性-佇列上放置的訊息必須加密。 如需相關資訊，請參閱 第 498 頁的『AMS 提供的保護品質』
- 無-沒有適用的資料保護。

規定在佇列上放置訊息時必須簽署其 QOP 為 INTEGRITY 的原則。 INTEGRITY 的 QOP 表示原則規定簽章演算法，但未規定加密演算法。 受完整性保護的訊息也稱為 "簽署"。

規定在佇列上放置訊息時必須簽署及加密的原則，其 QOP 為 PRIVACY。 PRIVACY 的 QOP 表示當原則規定簽章演算法及加密演算法時。 受隱私權保護的訊息也稱為 "SEALE"。

規定在佇列上放置訊息時必須加密的原則，其 QOP 為 CONFIDENTIALITY。 「機密性」的 QOP 表示原則規定加密演算法。

不規定簽章演算法或加密演算法的原則具有 QOP NONE。 對於具有 QOP 為 NONE 之原則的佇列，Advanced Message Security 不提供資料保護。

在 AMS 中管理安全原則

安全原則是一個概念性物件，說明訊息加密及簽署的方式。

與安全原則相關的所有管理作業執行所在的位置，視您使用的平台而定。

- ALW 在 AIX, Linux, and Windows 上，您可以使用 DELETE POLICY、 DISPLAY POLICY 及 SET POLICY (或同等的 PCF) 指令來管理安全原則。
  - Linux AIX 在 AIX and Linux 上，可以從 *MQ_INSTALLATION_PATH*/bin 執行管理作業。
  - Windows 在 Windows 平台上，可以從任何位置執行管理作業，因為安裝時會更新 PATH 環境變數。
- IBM i 在 IBM i 上，當安裝 IBM MQ 時， DSPMQMSPL、 SETMQMSPL 及 WRKMQMSPL 指令會安裝至 QSYS 系統程式庫中，以取得系統的主要語言。

  根據語言特性載入，將其他國家語言版本安裝至 QSYS29xx 程式庫。 例如，以美式英文作為主要語言，以韓文作為第二語言的機器，會將美式英文指令安裝至 QSYS ，並將 QSYS2962 中的韓文第二語言負載設為 2962 ，這是韓文的語言負載。
- z/OS 在 z/OS 上，管理指令是使用訊息安全原則公用程式 (CSQ0UTIL) 來執行。 在 z/OS 上建立、修改或刪除原則時，除非停止並重新啟動佇列管理程式，或使用 z/OS MODIFY 指令來重新整理 Advanced Message Security 原則配置，否則 Advanced Message Security 無法辨識這些變更。 例如:

```
F <qmgr ssid>AMSM,REFRESH POLICY
```

相關工作

第 562 頁的『在 AMS 中建立安全原則』
安全原則定義放置訊息時保護訊息的方式，或如何在接收訊息時保護訊息。

第 562 頁的『在 AMS 中變更安全原則』
您可以使用 Advanced Message Security 來變更已定義之安全原則的詳細資料。

第 563 頁的『在 AMS 中顯示及傾出安全原則』
根據您提供的指令行參數，使用 **dspmqspl** 指令來顯示所有安全原則的清單或具名原則的詳細資料。

第 565 頁的『在 AMS 中移除安全原則』
若要移除 Advanced Message Security 中的安全原則，您必須使用 setmqspl 指令。

操作 Advanced Message Security

相關參考

訊息安全原則公用程式 (CSQ0UTIL)

在 *AMS* 中建立安全原則

安全原則定義放置訊息時保護訊息的方式，或如何在接收訊息時保護訊息。

開始之前

建立安全原則時必須符合一些進入條件:

- 佇列管理程式必須在執行中。
- 安全原則的名稱必須遵循 IBM MQ 物件命名規則。
- 您必須具有必要的權限，才能連接至佇列管理程式及建立安全原則:
  - z/OS 在 z/OS 上，授與 訊息安全原則公用程式 (CSQ0UTIL)中所記載的權限。
  - Multi 在 z/OS 以外的其他平台上，您必須使用 setmqaut 指令授與必要的 + connect、+ inq 及 + chg 權限。

  如需配置安全的相關資訊，請參閱 第 106 頁的『設定安全』。
- z/OS 在 z/OS 上，請確定已根據 CSQ4INSM 中的定義來定義必要的系統物件。

範例

以下是在佇列管理程式 QMGR 上建立原則的範例。 原則指定使用 SHA256 演算法簽署訊息，並針對 DN 為 CN=joe、O=IBM、C=US 及 DN 為 CN=jane、O=IBM、C = US 的憑證使用 AES256 演算法進行加密。 此原則附加至 MY.QUEUE:

```
setmqspl -m QMGR -p MY.QUEUE -s SHA256 -e AES256 -r CN=joe,O=IBM,C=US -r CN=jane,O=IBM,C=US
```

以下是在佇列管理程式 QMGR 上建立原則的範例。 原則指定針對具有 DN 的憑證使用 3DES 演算法來加密訊息 :CN=john， O=IBM， C=US and CN=Jeff， O=IBM， C=US，並針對具有 DN 的憑證使用 SHA256 演算法簽署 :CN=phil， O=IBM， C=US

```
setmqspl -m QMGR -p MY.OTHER.QUEUE -s SHA256 -e 3DES -r CN=john,O=IBM,C=US -r
CN=jeff,O=IBM,C=US -a CN=phil,O=IBM,C=US
```

註:

- 用於訊息放置及取得的保護品質必須相符。 如果為訊息定義的原則保護品質低於為佇列定義的保護品質，則會將訊息傳送至錯誤處理佇列。 此原則同時適用於本端及遠端佇列。

相關參考

setmqspl 指令屬性的完整清單

在 *AMS* 中變更安全原則

您可以使用 Advanced Message Security 來變更已定義之安全原則的詳細資料。

開始之前

- 您要操作的佇列管理程式必須在執行中。
- 您必須具有必要的權限，才能連接至佇列管理程式及建立安全原則。
  - z/OS 在 z/OS 上，授與 訊息安全原則公用程式 (CSQ0UTIL)中所記載的權限。
  - Multi 在 z/OS 以外的其他平台上，您必須使用 setmqaut 指令授與必要的 + connect、+ inq 及 + chg 權限。

  如需配置安全的相關資訊，請參閱 第 106 頁的『設定安全』。

關於這項作業

若要變更安全原則，請將 `setmqspl` 指令套用至提供新屬性的現有原則。

範例

Here is an example of creating a policy named MYQUEUE on a queue manager named QMGR, specifying that messages are to be encrypted using the 3DES algorithm for authors (**-a**) having certificates with Distinguished Name (DN) of CN=alice,O=IBM,C=US and signed with the SHA256 algorithm for recipients (**-r**) having certificates with DN of CN=jeff,O=IBM,C=US.

```
setmqspl -m QMGR -p MYQUEUE -e 3DES -s SHA256 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

若要變更此原則，請發出 `setmqspl` 指令，其中包含範例中僅變更您要修改的值的所有屬性。 在此範例中，先前建立的原則會附加至新佇列，且其加密演算法會變更為 AES256:

```
setmqspl -m QMGR -p MYQUEUE -e AES256 -s SHA256 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

相關參考

setmqspl (設定安全原則)

在 *AMS* 中顯示及傾出安全原則

根據您提供的指令行參數，使用 **dspmqspl** 指令來顯示所有安全原則的清單或具名原則的詳細資料。

開始之前

- 若要顯示安全原則詳細資料，佇列管理程式必須存在且在執行中。
- 您必須具有必要的權限，才能連接至佇列管理程式及建立安全原則。
  - z/OS 在 z/OS 上，授與 訊息安全原則公用程式 (CSQ0UTIL)中所記載的權限。
  - Multi 在 z/OS 以外的其他平台上，您必須使用 setmqaut 指令授與必要的 + connect、+ inq 及 + chg 權限。

  如需配置安全的相關資訊，請參閱 第 106 頁的『設定安全』。

關於這項作業

以下是 **dspmqspl** 指令旗標的清單:

表 *108:* ***dspmqspl*** 指令旗標。

指令旗標	說明
-m	佇列管理程式名稱 (必要)。
-p	原則名稱。
-export	新增此旗標會產生可輕鬆套用至不同佇列管理程式的輸出。

範例

下列範例顯示如何為 venus.queue.manager 建立兩個安全原則:

```
setmqspl -m venus.queue.manager -p AMS_POL_04_ONE -s sha256 -a "CN=signer1,O=IBM,C=US" -e NONE
setmqspl -m venus.queue.manager -p AMS_POL_06_THREE -s sha256 -a "CN=another signer,O=IBM,C=US"
-e NONE
```

此範例顯示一個指令，其中顯示針對 venus.queue.manager 定義的所有原則及其產生的輸出的詳細資料:

```
dspmqspl -m venus.queue.manager

Policy Details:
Policy name: AMS_POL_04_ONE
Quality of protection: INTEGRITY
Signature algorithm: SHA256
Encryption algorithm: NONE
Signer DNs:
  CN=signer1,O=IBM,C=US
Recipient DNs: -
Toleration: 0
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Policy Details:
Policy name: AMS_POL_06_THREE
Quality of protection: INTEGRITY
Signature algorithm: SHA256
Encryption algorithm: NONE
Signer DNs:
  CN=another signer,O=IBM,C=US
Recipient DNs: -
Toleration: 0
```

此範例顯示一個指令，該指令顯示針對 venus.queue.manager 定義的所選安全原則的詳細資料及其產生的輸出:

```
dspmqspl -m venus.queue.manager -p AMS_POL_06_THREE

Policy Details:
Policy name: AMS_POL_06_THREE
Quality of protection: INTEGRITY
Signature algorithm: SHA256
Encryption algorithm: NONE
Signer DNs:
  CN=another signer,O=IBM,C=US
Recipient DNs: -
Toleration: 0
```

在下一個範例中，我們先建立安全原則，然後使用 **-export** 旗標來匯出原則:

```
setmqspl -m venus.queue.manager -p AMS_POL_04_ONE -s SHA256 -a "CN=signer1,O=IBM,C=US" -e NONE

dspmqspl -m venus.queue.manager -export
```

z/OS 在 z/OS 上， CSQ0UTIL 會將匯出的原則資訊寫入 EXPORT DD。

Multi 在 z/OS 以外的平台上，將輸出重新導向至檔案，例如:

```
dspmqspl -m venus.queue.manager -export > policies.[bat|sh]
```

如果要匯入安全原則，請執行下列動作:

- Linux AIX 在 AIX and Linux 上:
  1. 以屬於 mqm IBM MQ 管理群組的使用者身分登入。
  2. 發出 . policies.sh。
- Windows 在 Windows 上，執行 policies.bat。
- z/OS 在 z/OS 上，使用 CSQ0UTIL 公用程式，將包含匯出原則資訊的資料集指定為 SYSIN。

相關參考
dspmqspl 指令屬性的完整清單

在 *AMS* 中移除安全原則
若要移除 Advanced Message Security 中的安全原則，您必須使用 `setmqspl` 指令。

開始之前
管理安全原則時必須符合一些進入條件:

- 佇列管理程式必須在執行中。
- 您必須具有必要的權限，才能連接至佇列管理程式及建立安全原則。
  - z/OS 在 z/OS 上，授與 訊息安全原則公用程式 (CSQ0UTIL)中所記載的權限。
  - Multi 在 z/OS 以外的其他平台上，您必須使用 setmqaut 指令授與必要的 + connect、+ inq 及 + chg 權限。

  如需配置安全的相關資訊，請參閱 第 106 頁的『設定安全』。

關於這項作業
搭配使用 **setmqspl** 指令與 **-remove** 選項。

範例
以下是移除原則的範例:

```
setmqspl -m QMGR -remove -p MY.OTHER.QUEUE
```

相關參考
setmqspl 指令屬性的完整清單

AMS 中的系統佇列保護
系統佇列可啟用 IBM MQ 與其輔助應用程式之間的通訊。 每當建立佇列管理程式時，也會建立系統佇列來儲存 IBM MQ 內部訊息和資料。 您可以使用 Advanced Message Security 來保護系統佇列，以便只有授權使用者才能存取或解密它們。

系統佇列保護遵循與一般佇列保護相同的型樣。 請參閱 第 562 頁的『在 AMS 中建立安全原則』。

Windows 若要在 Windows 上使用系統佇列保護，請將 `keystore.conf` 檔案複製到下列目錄:

```
c:\Documents and Settings\Default User\.mqs\keystore.conf
```

z/OS 在 z/OS 上，若要為 `SYSTEM.ADMIN.COMMAND.QUEUE` 提供保護，指令伺服器必須具有 `keystore` 及 `keystore.conf` 的存取權，其中包含金鑰及配置，以便指令伺服器可以存取金鑰及憑證。 對 `SYSTEM.ADMIN.COMMAND.QUEUE` 安全原則所做的所有變更都需要重新啟動指令伺服器。

根據原則設定，會簽署或簽署及加密從指令佇列傳送及接收的所有訊息。 如果管理者定義授權簽章者，則指令伺服器不會執行未通過簽章者識別名稱 (DN) 檢查的指令訊息，且不會遞送至 Advanced Message Security 錯誤處理佇列。 作為 IBM MQ Explorer 暫時動態佇列的回覆傳送的訊息不受 AMS 保護。

安全原則不會影響下列 SYSTEM 佇列:

- SYSTEM.ADMIN.ACCOUNTING.QUEUE
- SYSTEM.ADMIN.ACTIVITY.QUEUE
- SYSTEM.ADMIN.CHANNEL.EVENT
- SYSTEM.ADMIN.COMMAND.EVENT
- z/OS SYSTEM.ADMIN.COMMAND.QUEUE

- SYSTEM.ADMIN.CONFIG.EVENT
- SYSTEM.ADMIN.LOGGER.EVENT
- SYSTEM.ADMIN.PERFM.EVENT
- SYSTEM.ADMIN.PUBSUB.EVENT
- SYSTEM.ADMIN.QMGR.EVENT
- SYSTEM.ADMIN.STATISTICS.QUEUE
- SYSTEM.ADMIN.TRACE.ROUTE.QUEUE
- SYSTEM.AUTH.DATA.QUEUE
- SYSTEM.BROKER.ADMIN.STREAM
- z/OS SYSTEM.BROKER.CLIENTS.DATA
- SYSTEM.BROKER.CONTROL.QUEUE
- SYSTEM.BROKER.DEFAULT.STREAM
- SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS
- z/OS SYSTEM.BROKER.SUBSCRIPTIONS.DATA
- SYSTEM.CHANNEL.INITQ
- SYSTEM.CHANNEL.SYNCQ
- z/OS SYSTEM.CHLAUTH.DATA.QUEUE
- SYSTEM.CICS.INITIATION.QUEUE
- SYSTEM.CLUSTER.COMMAND.QUEUE
- SYSTEM.CLUSTER.HISTORY.QUEUE
- SYSTEM.CLUSTER.REPOSITORY.QUEUE
- SYSTEM.CLUSTER.TRANSMIT.QUEUE
- z/OS SYSTEM.COMMAND.INPUT
- z/OS SYSTEM.DDELAY.LOCAL.QUEUE
- SYSTEM.DEAD.LETTER.QUEUE
- SYSTEM.DURABLE.SUBSCRIBER.QUEUE
- SYSTEM.HIERARCHY.STATE
- SYSTEM.INTER.QMGR.CONTROL
- SYSTEM.INTER.QMGR.FANREQ
- SYSTEM.INTER.QMGR.PUBS
- SYSTEM.INTERNAL.REPLY.QUEUE
- z/OS SYSTEM.JMS.PS.STATUS.QUEUE
- z/OS SYSTEM.JMS.REPORT.QUEUE
- SYSTEM.PENDING.DATA.QUEUE
- SYSTEM.PROTECTION.ERROR.QUEUE
- SYSTEM.PROTECTION.POLICY.QUEUE
- z/OS SYSTEM.QSG.CHANNEL.SYNCQ
- z/OS SYSTEM.QSG.TRANSMIT.QUEUE
- z/OS SYSTEM.QSG.UR.RESOLUTION.QUEUE
- SYSTEM.RETAINED.PUB.QUEUE

- z/OS SYSTEM.RETAINED.PUB.QUEUE
- SYSTEM.SELECTION.EVALUATION.QUEUE
- SYSTEM.SELECTION.VALIDATION.QUEUE

相關參考
系統和預設對象

V 9.2.3 Multi 串流佇列及 AMS

可以串流重複的 Advanced Message Security (AMS) 受保護訊息。

如果佇列定義了 AMS 原則，導致放置在該佇列中的訊息被簽署及/或加密，您也可以配置佇列的 **STREAMQ** 屬性，將每一個受保護訊息的副本放置在第二個佇列中。 使用針對原始佇列所配置的相同原則來簽署及/或加密重複的串流訊息。

在下列範例中，您將配置兩個佇列: QUEUE1 和 QUEUE2。 QUEUE1 將其 **STREAMQ** 屬性配置成將串流訊息放置到 QUEUE2:

```
DEFINE QLOCAL(QUEUE2)

DEFINE QLOCAL(QUEUE1) STREAMQ(QUEUE2)
```

具有憑證 CN=bob,O=IBM,C=GB 的使用者將 AMS 受保護訊息放置到 QUEUE1 。

具有憑證 CN=alice,O=IBM,C=GB 的應用程式將耗用來自 QUEUE1 的訊息。 具有憑證 CN=fred,O=IBM,C=GB 的個別應用程式將耗用來自 QUEUE2 的訊息。

QUEUE1 已套用下列 AMS 隱私權原則:

```
SET POLICY(QUEUE1) SIGNALG(SHA256) SIGNER('CN=bob,O=IBM,C=GB') ENCALG(AES256)
RECIP('CN=alice,O=IBM,C=GB') RECIP('CN=fred,O=IBM,C=GB') ACTION(ADD
```

如果已在 QUEUE1 的原則中配置加密演算法，則原則中列出的收件者必須同時包括來自 QUEUE1 的原始訊息收件者，以及將要耗用來自 QUEUE2 的重複訊息的收件者。

當應用程式嘗試使用來自 QUEUE2 的訊息時，它會執行完整性檢查，及/或根據已在 QUEUE2 上設定的原則來解密訊息。 如果應用程式想要耗用來自 QUEUE2 的串流訊息，您必須在 QUEUE2 上設定適當的原則，以容許檢查訊息的完整性並正確解密。

尤其是簽署演算法、簽章者及加密演算法必須與套用至 QUEUE1 的原則相同。 QUEUE2 的原則收件者必須包括使用來自 QUEUE2 的訊息的收件者身分。

註: 套用至 QUEUE2 的原則不需要列出 QUEUE1 上原則集中指定的所有收件者。

例如，可以在 QUEUE2 上設定下列原則，以容許具有憑證識別名稱 CN=fred,O=IBM,C=GB 的應用程式從中讀取 AMS 保護的訊息:

```
SET POLICY(QUEUE2) SIGNALG(SHA256) SIGNER('CN=bob,O=IBM,C=GB') ENCALG(AES256)
RECIP('CN=fred,O=IBM,C=GB') ACTION(ADD)
```

相關概念
串流佇列

在 AMS 中授與 OAM 許可權

檔案許可權授權所有使用者執行 setmqspl 和 dspmqspl 指令。 不過， Advanced Message Security 依賴「物件權限管理程式 (OAM)」，而且每次嘗試由不屬於 mqm 群組 (即 IBM MQ 管理群組) 的使用者執行這些指令，或無權讀取所授與的安全原則設定，都會導致錯誤。

程序

若要將必要的許可權授與使用者，請執行:

```
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
```

註: 只有在您想要使用 Advanced Message Security 7.0.1 將用戶端連接至佇列管理程式時，才需要設定這些 OAM 權限。

小心: SYSTEM.PROTECTION.POLICY.QUEUE 並非在所有狀況下都是必要的。 IBM MQ 可透過快取原則來最佳化效能，以便您無需在 SYSTEM.PROTECTION.POLICY.QUEUE 。

IBM MQ 不會快取所有可用的原則。 如果有大量原則， IBM MQ 會快取有限數目的原則。 因此，如果佇列管理程式定義的原則數目較少，則不需要提供瀏覽選項給 SYSTEM.PROTECTION.POLICY.QUEUE。

不過，如果定義了大量原則，或您使用舊用戶端，您應該提供此佇列的瀏覽權限。 SYSTEM.PROTECTION.ERROR.QUEUE 用來放置 AMS 程式碼所產生的錯誤訊息。 只有在您嘗試將錯誤訊息放入佇列時，才會檢查此佇列的放置權限。 當您嘗試從 AMS 受保護佇列中放置或取得訊息時，不會檢查您對佇列的放置權限。

在 AMS 中授與安全許可權

使用指令資源安全時，您必須設定許可權，以容許 Advanced Message Security 運作。 本主題在範例中使用 RACF 指令。 如果您的企業使用不同的外部安全管理程式 (ESM) ，則必須使用該 ESM 的對等指令。

授與安全許可權有三個層面:

- 第 568 頁的『ASM 位址空間』
- 第 568 頁的『CSQ0UTIL』
- 第 569 頁的『使用已定義 Advanced Message Security 原則的佇列』

附註: 範例指令使用下列變數。

1. *QMgrName* -佇列管理程式的名稱。

   z/OS 在 z/OS 上，此值也可以是佇列共用群組的名稱。
2. *username* -這可以是群組名稱。
3. 範例顯示 MQQUEUE 類別。 這也可以是 MXQUEUE、GMQQUEUE 或 GMXQUEUE。 如需進一步資訊，請參閱第 169 頁的『佇列安全的設定檔』。

此外，如果設定檔已存在，則不需要 RDEFINE 指令。

ASM 位址空間

您需要對執行 Advanced Message Security 位址空間的使用者名稱發出一些 IBM MQ 安全。

- 對於佇列管理程式的批次連線，請發出

```
RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
            PERMIT QMgrName.BATCH CLASS(MQCONN) ID(username) ACCESS(READ)
```

- 用於存取 SYSTEM.PROTECTION.POLICY.QUEUE，問題:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
            PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

CSQ0UTIL

容許使用者執行 **setmqspl** 及 **dspmqspl** 指令的公用程式需要下列許可權，其中使用者名稱是工作使用者 ID:

- 對於佇列管理程式的批次連線，請發出:

```
  RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
            PERMIT QMgrName.BATCH CLASS(MQCONN) ID(username) ACCESS(READ)
```

- 若要存取 **setmqpol** 指令所需的 SYSTEM.PROTECTION.POLICY.QUEUE，請發出:

```
  RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
              PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(ALTER)
```

- 若要存取 **dspmqpol** 指令所需的 SYSTEM.PROTECTION.POLICY.QUEUE，請發出:

```
  RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
              PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

使用已定義 Advanced Message Security 原則的佇列

當應用程式使用已定義原則的佇列時，該應用程式需要其他許可權才能容許 Advanced Message Security 保護訊息。

應用程式需要:

- 對 SYSTEM.PROTECTION.POLICY.QUEUE。 請發出下列指令來執行此動作:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
          PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

- 放置對 SYSTEM.PROTECTION.ERROR.QUEUE。 請發出下列指令來執行此動作:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.ERROR.QUEUE UACC(NONE)
            PERMIT QMgrName.SYSTEM.PROTECTION.ERROR.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

IBM i 在 IBM i 上設定 AMS 的憑證和金鑰儲存庫配置檔

設定 Advanced Message Security 保護時的第一項作業是建立憑證，並將該憑證與您的環境相關聯。 關聯是透過整合檔案系統 (IFS) 中所保留的檔案來配置。

程序

1. 若要使用 IBM i 隨附的 OpenSSL 工具來建立自簽憑證，請從 QShell 發出下列指令:

   ```
   /QOpenSys/usr/bin/openssl req -x509 -newkey rsa:2048 -keyout
   $HOME/private.pem -out $HOME/mycert.pem -nodes -days 365
   ```

   指令會提示輸入新自簽憑證的各種識別名稱屬性，包括:

   - 通用名稱 (CN =)
   - 組織 (O =)
   - 國家 (C =)

   這會以 PEM (Privacy Enhanced Mail) 格式建立未加密的私密金鑰和相符憑證。

   為了簡單起見，只要輸入通用名稱、組織及國家/地區的值。 這些屬性和值在建立原則時非常重要。

   透過在指令行上使用 **-config** 參數指定自訂 openssl 配置檔，可以自訂其他提示及屬性。 如需配置檔語法的詳細資料，請參閱 OpenSSL 文件。

   例如，下列指令會新增其他 X.509 v3 憑證延伸:

   ```
   /QOpenSys/usr/bin/openssl req -x509 -newkey rsa:2048
   -keyout $HOME/private.pem -out $HOME/mycert.pem -nodes -days 365 -config myconfig.cnf
   ```

   其中 myconfig.cnf 是包含下列項目的 ASCII 串流檔:

   ```
   [req]
   distinguished_name = req_distinguished_name
   x509_extensions = myextensions
   ```

```
[req_distinguished_name]
countryName = Country Name (2 letter code)
countryName_default = GB
stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = Hants
localityName = Locality Name (eg, city)
localityName_default = Hursley
organizationName = Organization Name (eg, company)
organizationName_default = IBM United Kingdom
organizationalUnitName = Organizational Unit Name (eg, department)
organizationalUnitName_default = IBM MQ Development
commonName = Common Name (eg, Your Name)

[myextensions]
keyUsage = digitalSignature,nonRepudiation,dataEncipherment,keyEncipherment
extendedKeyUsage = emailProtection
```

2. AMS 要求憑證和私密金鑰都保留在相同的檔案中。 請發出下列指令來達到此目的:

```
cat $HOME/mycert.pem >> $HOME/private.pem
```

$HOME 中的 private.pem 檔案現在包含相符的私密金鑰和憑證，而 mycert.pem 檔案包含您可以加密其訊息並驗證簽章的所有公用憑證。

這兩個檔案需要在預設位置中建立金鑰儲存庫配置檔 keystore.conf，以與您的環境相關聯。

依預設， AMS 會在起始目錄的 .mqs 子目錄中尋找金鑰儲存庫配置。

3. 在 QShell 中，建立 keystore.conf 檔:

```
mkdir -p $HOME/.mqs
echo "pem.private = $HOME/private.pem" > $HOME/.mqs/keystore.conf
echo "pem.public = $HOME/mycert.pem" >> $HOME/.mqs/keystore.conf
echo "pem.password = unused" >> $HOME/.mqs/keystore.conf
```

IBM i 在 *IBM i* 上建立 *AMS* 的原則

在建立原則之前，您需要建立佇列來保留受保護的訊息。

程序

1. 在指令行提示中輸入;

```
CRTMQMQ QNAME(PROTECTED) QTYPE(*LCL) MQMNAME (mqmname)
```

其中 mqmname 是佇列管理程式的名稱。

使用 DSPMQM 指令來檢查佇列管理程式是否能夠使用安全原則。 確定 **Security Policy Capability** 顯示 **YES*。

您可以定義最簡單的原則是完整性原則，這是透過使用數位簽章演算法 (但沒有加密演算法) 來建立原則來達成。

訊息已簽署但未加密。 如果要加密訊息，您必須指定加密演算法，以及一或多個預期的訊息收件者。

公用金鑰儲存庫中預期訊息收件者的憑證是透過識別名稱來識別。

2. 在 QShell 中使用下列指令，在 $HOME 中顯示公用金鑰儲存庫中憑證的識別名稱 mycert.pem :

```
/QOpenSys/usr/bin/openssl x509 -in $HOME/mycert.pem -noout -subject -nameopt RFC2253
```

您需要輸入識別名稱作為預期的收件者，且原則名稱必須符合要保護的佇列名稱。

3. 在 CL 命令提示字元中輸入，例如:

```
SETMQMSPL POLICY(PROTECTED) MQMNAME (mqmname)SIGNALG(*SHA256) ENCALG(*AES256) RECIP('CN=..,
O=.., C=..')
```

其中 mqmname 是佇列管理程式的名稱。

建立原則之後，任何透過該佇列名稱放置、瀏覽或破壞性移除的訊息都會受限於 AMS 原則。

相關參考
顯示訊息佇列管理程式 (DSPMQM)
設定 MQM 安全原則 (SETMQMSPL)

IBM i 在 *IBM i* 上測試 *AMS* 的原則
使用產品隨附的範例應用程式來測試您的安全原則。

關於這項作業

您可以使用 IBM MQ 隨附的範例應用程式 (例如 AMQSPUT4、 AMQSGET4、 AMQSGBR4 及 WRKMQMMSG 等工具) 來放置、瀏覽及取得使用 PROTECTED 佇列名稱的訊息。

如果已正確配置所有項目，則此使用者的應用程式行為與未受保護佇列的應用程式行為應該沒有差異。

未設定 Advanced Message Security 的使用者，或沒有解密訊息所需的私密金鑰的使用者將無法檢視訊息。 使用者收到完成碼 RCFAIL，相當於 MQCC_FAILED (2) 和原因碼 RC2063 (MQRC_SECURITY_ERROR)。

若要查看 AMS 保護是否有效，請將部分測試訊息放入 PROTECTED 佇列，例如使用 AMQSPUT0。 然後，您可以建立別名佇列，以在靜止時瀏覽原始受保護資料。

程序

若要將必要的許可權授與使用者，請執行:

```
CRTMQMQ QNAME(ALIAS) QTYPE(*ALS) TGTQNAME(PROTECTED) MQMNAME(yourqm)
```

使用 ALIAS 佇列名稱瀏覽 (例如使用 AMQSBCG4 或 WRKMQMMSG) 應該會顯示較大的 `scrambled` 訊息，其中 PROTECTED 佇列的瀏覽會顯示明碼訊息。

可以看到加擾的訊息，但無法使用 ALIAS 佇列來辨識原始明碼，因為 AMS 沒有施行符合此名稱的原則。 因此，會傳回原始受保護資料。

相關參考
設定 MQM 安全原則 (SETMQMSPL)
使用 MQ 訊息 (WRKMQMMSG)

AMS 的指令及配置事件

使用 Advanced Message Security，您可以產生指令及配置事件訊息，這些訊息可以記載並作為審核的原則變更記錄。

IBM MQ 產生的指令及配置事件是傳送至發生事件之佇列管理程式上專用佇列的 PCF 格式訊息。

配置事件訊息會傳送至 SYSTEM.ADMIN.CONFIG.EVENT 佇列。

指令事件訊息會傳送至 SYSTEM.ADMIN.COMMAND.EVENT 佇列。

不論您用來管理 Advanced Message Security 安全原則的工具為何，都會產生事件。

在 Advanced Message Security 中，安全原則上不同動作所產生的事件有四種類型:

- 第 562 頁的『在 AMS 中建立安全原則』，產生兩則 IBM MQ 事件訊息:
  - 配置事件
  - 指令事件
- 第 562 頁的『在 AMS 中變更安全原則』，它會產生三則 IBM MQ 事件訊息:
  - 包含舊安全原則值的配置事件
  - 包含新安全原則值的配置事件
  - 指令事件
- 第 563 頁的『在 AMS 中顯示及傾出安全原則』，產生一則 IBM MQ 事件訊息:
  - 指令事件

- 第 565 頁的『在 AMS 中移除安全原則』，產生兩則 IBM MQ 事件訊息:
  - 配置事件
  - 指令事件

啟用及停用 *AMS* 的事件記載

您可以使用佇列管理程式屬性 **CONFIGEV** 及 **CMDEV** 來控制指令及配置事件。 若要啟用這些事件，請將適當的佇列管理程式屬性設為 ENABLED。 若要停用這些事件，請將適當的佇列管理程式屬性設為 DISABLED。

程序

配置事件

若要啟用配置事件，請將 **CONFIGEV** 設為 ENABLED。 若要停用配置事件，請將 **CONFIGEV** 設為 DISABLED。 例如，您可以使用下列 MQSC 指令來啟用配置事件:

```
ALTER QMGR CONFIGEV (ENABLED)
```

指令事件

若要啟用指令事件，請將 **CMDEV** 設為 ENABLED。 若要對指令啟用指令事件，但 **DISPLAY MQSC** 指令及 Inquire PCF 指令除外，請將 **CMDEV** 設為 NODISPLAY。 若要停用指令事件，請將 **CMDEV** 設為 DISABLED。 例如，您可以使用下列 MQSC 指令來啟用指令事件:

```
ALTER QMGR CMDEV (ENABLED)
```

相關工作

在 IBM MQ 中控制配置、指令及日誌程式事件

AMS 的指令事件訊息格式

指令事件訊息包含 MQCFH 結構及其後的 PCF 參數。

以下是選取的 MQCFH 值:

```
Type = MQCFT_EVENT;
Command = MQCMD_COMMAND_EVENT;
MsgSeqNumber = 1;
Control = MQCFC_LAST;
ParameterCount = 2;
CompCode = MQCC_WARNING;
Reason = MQRC_COMMAND_PCF;
```

註: ParameterCount 值是 2，因為 MQCFGR 類型 (群組) 一律有兩個參數。 每一個群組都由適當的參數組成。 事件資料由兩個群組組成: CommandContext 和 CommandData。

CommandContext 包含:

EventUserID

說明:	發出已產生事件的指令或呼叫的使用者 ID。 (這是用來檢查發出指令或呼叫之權限的相同使用者 ID; 對於從佇列接收的指令，這是來自指令訊息 MD 的使用者 ID (UserIdentifier))。
ID:	MQCACF_EVENT_USER_ID。
資料類型:	MQCFST。
長度上限:	MQ_USER_ID_length。
已傳回:	始終。

EventOrigin

說明:	導致事件的動作來源。
ID:	MQIACF_EVENT_ORIGIN。

資料類型: MQCFIN。

值:

MQEVO_CONSOLE
主控台指令-指令行。

MQEVO_MSG
來自 IBM MQ Explorer 外掛程式的指令訊息。

已傳回: 始終。

EventQMgr

說明: 輸入指令或呼叫所在的佇列管理程式。 (執行指令且產生事件的佇列管理程式位於事件訊息的 MD 中)。

ID: MQCACF_EVENT_Q_MGR。

資料類型: MQCFST。

長度上限: MQ_Q_MGR_NAME_LENGTH。

已傳回: 始終。

EventAccountingToken

說明: 對於當作訊息 (MQEVO_MSG) 接收的指令，則是來自指令訊息 MD 的帳戶記號 (AccountingToken)。

ID: MQBACF_EVENT_ACCOUNTING_TOKEN。

資料類型: MQCFBS。

長度上限: MQ_ACCOUNTING_TOKEN_LENGTH。

已傳回: 僅當 EventOrigin 為 MQEVO_MSG 時。

EventIdentityData

說明: 對於作為訊息 (MQEVO_MSG) 接收的命令，來自命令訊息的 MD 的應用程式身分資料 (ApplIdentityData)。

ID: MQCACF_EVENT_APPL_IDENTITY。

資料類型: MQCFST。

長度上限: MQ_APPL_IDENTITY_DATA_LENGTH。

已傳回: 僅當 EventOrigin 為 MQEVO_MSG 時。

EventApplType

說明: 對於作為訊息 (MQEVO_MSG) 接收的命令，來自命令訊息的 MD 的應用程式類型 (PutApplType)。

ID: MQIACF_EVENT_APPL_TYPE。

資料類型: MQCFIN。

已傳回: 僅當 EventOrigin 為 MQEVO_MSG 時。

EventApplName

說明: 對於作為訊息接收的命令 (MQEVO_MSG)，命令訊息的 MD 中的應用程式名稱 (PutApplName)。

ID: MQCACF_EVENT_APPL_NAME。

資料類型: MQCFST。

長度上限: MQ_APPL_NAME_LENGTH。

已傳回: 僅當 EventOrigin 為 MQEVO_MSG 時。

EventApplOrigin

說明: 對於作為訊息 (MQEVO_MSG) 接收的命令，應用程式原始資料 (ApplOriginData) 來自命令訊息的 MD。

ID: MQCACF_EVENT_APPL_ORIGIN。

資料類型: MQCFST。

長度上限: MQ_APPL_ORIGIN_DATA_LENGTH。

已傳回: 僅當 EventOrigin 為 MQEVO_MSG 時。

指令

說明: 指令碼。

ID: MQIACF_COMMAND。

資料類型: MQCFIN。

值: **MQCMD_INQUIRE_PROT_POLICY 數值 205**
MQCMD_CREATE_PROT_POLICY 數值 206
MQCMD_DELETE_PROT_POLICY 數值 207
MQCMD_CHANGE_PROT_POLICY 數值 208
這些定義在 IBM MQ 8.0 `cmqcfc.h` 中

已傳回: 始終。

CommandData 包含包含 PCF 指令的 PCF 元素。

AMS 的配置事件訊息格式

配置事件是標準 Advanced Message Security 格式的 PCF 訊息。

在 事件訊息 MQMD (訊息描述子)中可以找到 MQMD 訊息描述子的可能值。

以下是選取的 MQMD 值:

```
Format = MQFMT_EVENT
Peristence = MQPER_PERSISTENCE_AS_Q_DEF
PutApplType = MQAT_QMGR           //for both CLI and command server
```

訊息緩衝區由 MQCFH 結構及其後面的參數結構組成。 可以在 事件訊息 MQCFH (PCF 標頭)中找到可能的 MQCFH 值。

以下是選取的 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}
```

下列 MQCFH 參數如下:

EventUserID

說明: 發出已產生事件的指令或呼叫的使用者 ID。 (這是用來檢查發出指令或呼叫之權限的相同使用者 ID; 對於從佇列接收的指令，這是來自指令訊息 MD 的使用者 ID (UserIdentifier))。

ID: **MQCACF_EVENT_USER_ID**

資料類型: MQCFST。

長度上限: MQ_USER_ID_length。

已傳回: 始終。

SecurityId

說明: MQMD.AccountingToken ; 如果是本端指令，則為 Windows SID。

ID: **MQBACF_EVENT_SECURITY_ID**

資料類型: MQCBS。

長度上限: MQ_SECURITY_ID_LENGTH。

已傳回: 始終。

EventOrigin

說明: 導致事件的動作來源。

ID: **MQIACF_EVENT_ORIGIN**

資料類型: MQCFIN。

值:

MQEVO_CONSOLE
主控台指令-指令行。

MQEVO_MSG
來自 IBM MQ Explorer 外掛程式的指令訊息。

已傳回: 始終。

EventQMgr

說明: 輸入指令或呼叫所在的佇列管理程式。 (執行指令且產生事件的佇列管理程式位於事件訊息的 MD 中)。

ID: **MQCACF_EVENT_Q_MGR**

資料類型: MQCFST

長度上限: MQ_Q_MGR_NAME_LENGTH

已傳回: 始終。

ObjectType

說明: 物件類型。

ID: **MQIACF_OBJECT_TYPE**

資料類型: MQCFIN

值:

MQOT_PROT_POLICY
Advanced Message Security 保護原則。 **1019** -在 IBM MQ 8.0 或 `cmqc.h` 檔案中定義的數值。

已傳回: 始終。

PolicyName

說明: Advanced Message Security 原則名稱。

ID: **MQCA_POLICY_NAME**。

資料類型: MQCFST。

值: **2112** -在 IBM MQ 8.0 或 cmqc.h 檔中定義的數值。

長度上限: MQ_OBJECT_NAME_LENGTH。

已傳回: 始終。

PolicyVersion

說明: Advanced Message Security 原則版本。

ID: **MQIA_POLICY_VERSION**

資料類型: MQCFIN

值 **238** -在 IBM MQ 8.0 或 cmqc.h 檔中定義的數值。

已傳回: 一律

TolerateFlag

說明: Advanced Message Security 原則容錯旗標。

ID: **MQIA_TOLERATE_UNPROTECTED**

資料類型: MQCFIN

值 **235** -在 IBM MQ 8.0 或 cmqc.h 檔中定義的數值。

已傳回: 始終。

SignatureAlgorithm

說明: Advanced Message Security 原則簽章演算法。

ID: **MQIA_SIGNATURE_ALGORITHM**

資料類型: MQCFIN

值: **236** - IBM MQ 8.0 或 cmqc.h 檔中定義的數值。

已傳回: 每當 Advanced Message Security 原則中定義簽章演算法時

EncryptionAlgorithm

說明: Advanced Message Security 原則加密演算法。

ID: **MQIA_ENCRYPTION_ALGORITHM**

資料類型: MQCFIN

值: **237** - IBM MQ 8.0 或 cmqc.h 檔中定義的數值。

已傳回: 每當 IBM MQ 原則中定義加密演算法時

SignerDNs

說明: 所容許簽章者的主旨 DistinguishedName 。

ID: **MQCA_SIGNER_DN**

資料類型: MQCFSL

值: **2113** -在 IBM MQ 8.0 或 cmqc.h 檔中定義的數值。

長度上限: 原則中最長的簽章者 DN，但不再是 MQ_DISTINGUISHED_NAME_LENGTH

已傳回: 每當在 IBM MQ 原則中定義時。

RecipientDNs

說明: 所容許簽章者的主旨 DistinguishedName。

ID: **MQCA_RECIPIENT_DN**

資料類型: MQCFSL

值: **2114** -在 IBM MQ 8.0 或 `cmqc.h` 檔中定義的數值。

長度上限: 原則中最長的接收端 DN，但不再是 MQ_DISTINGUISHED_NAME_LENGTH。

已傳回: 每當在 IBM MQ 原則中定義時。

注意事項

本資訊係針對 IBM 在美國所提供之產品與服務所開發。

在其他國家中，IBM 可能不會提供本書中所提的各項產品、服務或功能。 請洽當地 IBM 業務代表，以取得當地目前提供的產品和服務之相關資訊。 這份文件在提及 IBM 的產品、程式或服務時，不表示或暗示只能使用 IBM 的產品、程式或服務。 只要未侵犯 IBM 的智慧財產權，任何功能相當的產品、程式或服務都可以取代 IBM 的產品、程式或服務。 不過，任何非 IBM 的產品、程式或服務，使用者必須自行負責作業的評估和驗證責任。

本文件所說明之主題內容，IBM 可能擁有其專利或專利申請案。 提供本文件不代表提供這些專利的授權。 您可以書面提出授權查詢，來函請寄到：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

如果是有關雙位元組 (DBCS) 資訊的授權查詢，請洽詢所在國的 IBM 智慧財產部門，或書面提出授權查詢，來函請寄到：

智慧財產權授權
法務部與智慧財產權法律
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

下列段落不適用於英國，若與任何其他國家之法律條款抵觸，亦不適用於該國： International Business Machines Corporation 只依 "現況" 提供本出版品，不提供任何明示或默示之保證，其中包括且不限於不侵權、可商用性或特定目的之適用性的隱含保證。 有些地區在特定交易上，不允許排除明示或暗示的保證，因此，這項聲明不一定適合您。

這項資訊中可能會有技術上或排版印刷上的訛誤。 因此，IBM 會定期修訂；並將修訂後的內容納入新版中。 IBM 隨時會改進及/或變更本出版品所提及的產品及/或程式，不另行通知。

本資訊中任何對非 IBM 網站的敘述僅供參考，IBM 對該網站並不提供任何保證。 這些網站所提供的資料不是 IBM 本產品的資料內容，如果要使用這些網站的資料，您必須自行承擔風險。

IBM 得以各種適當的方式使用或散布由您提供的任何資訊，無需對您負責。

如果本程式的獲授權人為了 (i) 在個別建立的程式和其他程式（包括本程式）之間交換資訊，以及 (ii) 相互使用所交換的資訊，因而需要相關的資訊，請洽詢：

IBM Corporation
軟體交互作業能力協調程式，部門 49XA
3605 公路 52 N
Rochester， MN 55901
U.S.A.

在適當條款與條件之下，包括某些情況下（支付費用），或可使用此類資訊。

IBM 基於雙方之 IBM 客戶合約、IBM 國際程式授權合約或任何同等合約之條款，提供本資訊所提及的授權程式與其所有適用的授權資料。

本文件中所含的任何效能資料都是在受管制的環境下判定。 因此不同作業環境之下所得的結果，可能會有很大的差異。 有些測定已在開發階段系統上做過，不過這並不保證在一般系統上會出現相同結果。 甚至有部分的測量，是利用插補法而得的估計值， 實際結果可能有所不同。 本書的使用者應依自己的特定環境，查證適用的資料。

本文件所提及之非 IBM 產品資訊，取自產品的供應商，或其發佈的聲明或其他公開管道。 IBM 並未測試過這些產品，也無法確認這些非 IBM 產品的執行效能、相容性或任何對產品的其他主張是否完全無誤。 有關非 IBM 產品的性能問題應直接洽詢該產品供應商。

有關 IBM 未來方針或目的之所有聲明，僅代表 IBM 的目標與主旨，隨時可能變更或撤銷，不必另行通知。

這份資訊含有日常商業運作所用的資料和報告範例。 為了要使它們儘可能完整，範例包括個人、公司、 品牌和產品的名稱。 這些名稱全屬虛構，如與實際公司的名稱和住址雷同，純屬巧合。

著作權授權:

本資訊含有原始語言之範例應用程式，用以說明各作業平台中之程式設計技術。 您可以基於研發、使用、銷售或散布符合作業平台（撰寫範例程式的作業平台）之應用程式介面的應用程式等目的，以任何形式複製、修改及散布這些範例程式，而不必向 IBM 付費。 這些範例並未在所 有情況下完整測試。 因此，IBM 不保證或暗示這些程式的可靠性、有用性或功能。

若 貴客戶正在閱讀本項資訊的電子檔，可能不會有照片和彩色說明。

程式設計介面資訊

程式設計介面資訊 (如果有提供的話) 旨在協助您建立與此程式搭配使用的應用軟體。

本書包含預期程式設計介面的相關資訊，可讓客戶撰寫程式以取得 WebSphere MQ 的服務。

不過，本資訊也可能包含診斷、修正和調整資訊。 提供診斷、修正和調整資訊，是要協助您進行應用軟體的除錯。

重要: 請勿使用此診斷、修改及調整資訊作為程式設計介面，因為它可能會變更。

商標

IBM、 IBM 標誌 ibm.com 是 IBM Corporation 在全球許多適用範圍的商標。 IBM 商標的最新清單可在 Web 的 "Copyright and trademark information"www.ibm.com/legal/copytrade.shtml 中找到。 其他產品和服務名稱，可能是 IBM 或其他公司的商標。

Microsoft 及 Windows 是 Microsoft Corporation 在美國及/或其他國家或地區的商標。

UNIX 是 The Open Group 在美國及/或其他國家/地區的註冊商標。

Linux 是 Linus Torvalds 在美國及/或其他國家或地區的註冊商標。

本產品包含 Eclipse Project (https://www.eclipse.org/) 所開發的軟體。

Java 和所有以 Java 為基礎的商標及標誌是 Oracle 及/或其子公司的商標或註冊商標。

產品編號:

(1P) P/N: