7.5

保護 *IBM WebSphere MQ* 安全

附註

使用本資訊及其支援的產品之前，請先閱讀第 275 頁的『注意事項』中的資訊。

目錄

安全.....5
安全概觀..... 5
概念和機制..... 5
IBM WebSphere MQ 安全機制..... 18
規劃安全需求.....38
規劃識別及鑑別.....39
規劃授權.....41
規劃機密性..... 49
規劃資料完整性.....55
規劃審核.....55
依拓蹼規劃安全.....56
防火牆和網際網路透通.....65
設定安全..... 65
在 UNIX 和 Linux 及 Windows 系統上設定安全.....65
在 HP NSS 上設定安全..... 89
設定 IBM WebSphere MQ MQI 用戶端安全.....89
在 UNIX, Linux, and Windows 系統上設定 SSL 或 TLS 的通訊.....91
使用 SSL 或 TLS.....92
識別及鑑別使用者..... 121
特許使用者.....123
使用 MQCSP 結構來識別及鑑別使用者..... 123
在安全結束程式中實作識別及鑑別.....123
訊息結束程式中的身分對映.....124
API 結束程式和 API 交互結束程式中的身分對映.....124
使用已撤銷的憑證.....125
授權存取物件.....132
在 UNIX、Linux 及 Windows 系統上使用 OAM 來控制對物件的存取權..... 132
授與對資源的必要存取權.....139
在 UNIX、 Linux 及 Windows 系統上管理 IBM WebSphere MQ 的權限.....165
使用 IBM WebSphere MQ 物件的權限..... 167
在安全結束程式中實作存取控制.....170
在訊息結束程式中實作存取控制.....172
在 API 結束程式和 API 交互結束程式中實作存取控制.....172
訊息機密性..... 172
使用 SSL 或 TLS 連接兩個佇列管理程式.....172
將用戶端安全連接至佇列管理程式..... 178
指定 CipherSpecs..... 182
重設 SSL 秘密金鑰..... 188
在使用者結束程式中實作機密性.....189
訊息的資料完整性..... 190
使用 SSL 或 TLS 連接兩個佇列管理程式.....190
將用戶端安全連接至佇列管理程式..... 198
指定 CipherSpecs..... 202
審核..... 206
保持叢集安全.....206
停止傳送訊息的未獲授權佇列管理程式.....206
停止在佇列上放置訊息的未獲授權佇列管理程式..... 206
授權將訊息放置在遠端叢集佇列上..... 207
防止佇列管理程式加入叢集..... 208
強制不要的佇列管理程式離開叢集..... 209
防止佇列管理程式接收訊息..... 209
SSL 和叢集..... 209

發佈/訂閱安全 ... 211
發佈/訂閱安全設定範例 ... 217
訂閱安全 ... 226
IBM WebSphere MQ Advanced Message Security ... 227
IBM WebSphere MQ Advanced Message Security 概觀 ... 227
安裝 IBM WebSphere MQ Advanced Message Security ... 248
使用金鑰儲存庫和憑證 ... 249
建立 IBM WebSphere MQ Advanced Message Security 安全原則 ... 259
問題及解決方案 ... 273

注意事項 ... 275
程式設計介面資訊 ... 276
商標 ... 276

安全

對於 IBM WebSphere MQ 應用程式的開發人員以及配置 IBM WebSphere MQ 權限的系統管理者來說，安全是一項重要考量。

安全概觀

此主題集合介紹 IBM WebSphere 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 WebSphere 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 WebSphere 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 WebSphere MQ 實作時，請考量您需要哪些安全機制來實作對您很重要的安全層面。 如需在閱讀這些主題之後考量事項的相關資訊，請參閱 第 38 頁的『規劃安全需求』。

相關概念

第 172 頁的『使用 SSL 或 TLS 連接兩個佇列管理程式』
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。

第 92 頁的『使用 SSL 或 TLS』
這些主題提供如何執行與搭配使用 SSL 或 TLS 與 IBM WebSphere MQ 相關的單一作業的指示。

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 WebSphere MQ 環境中的相關安全服務，因為 IBM WebSphere 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。

具有 IBM WebSphere MQ Advanced Message Security 的 IBM WebSphere MQ 不會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 WebSphere 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 WebSphere MQ 中的識別及鑑別』
在 IBM WebSphere 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 WebSphere 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 WebSphere 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 41 頁的『規劃授權』 及相關聯的子主題。

相關概念

第 18 頁的『IBM WebSphere MQ 中的審核』
IBM WebSphere 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 WebSphere MQ 的加密法概念。

實體 一詞是用來指佇列管理程式、 WebSphere MQ MQI 用戶端、個別使用者或任何能夠交換訊息的其他系統。

相關概念

第 19 頁的『IBM WebSphere MQ 中的加密法』
IBM WebSphere MQ 使用 Secure Sockets Layer (SSL) 及傳輸安全層 (TLS) 通訊協定提供加密法。

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)。

如需加密法的定義，請參閱 名詞解釋 。

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 和 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。

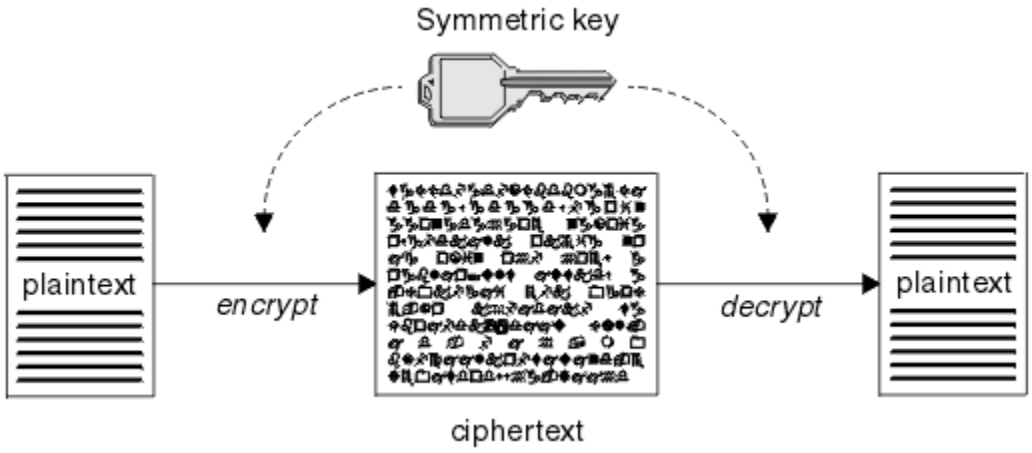


圖 *1:* 對稱金鑰加密法 *(symmetric key cryptography)*

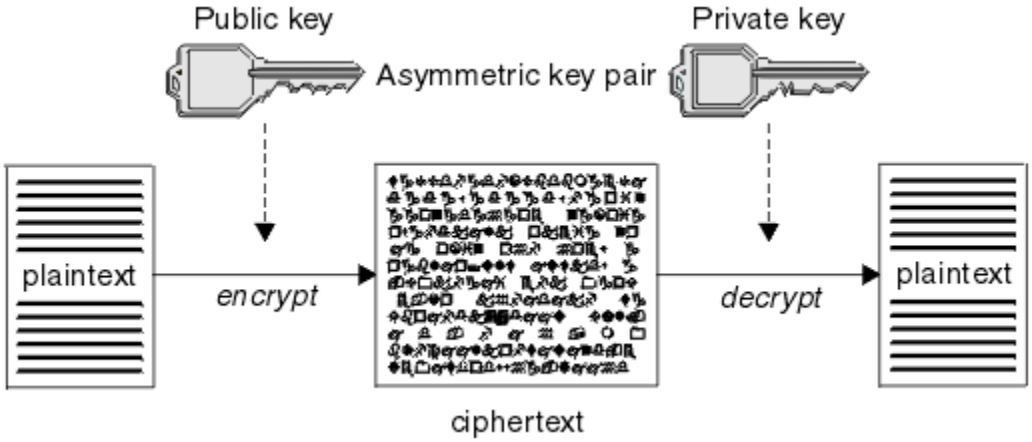


圖 *2:* 非對稱金鑰加密法 *(asymmetric key cryptography)*

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，由雜湊函數計算。 訊息摘要可以加密，形成數位簽章。

訊息本身的大小是可變的。 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，雜湊函數是符合兩個準則的轉換:

- 雜湊函數必須是單向。 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 和 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。

WebSphere MQ 所使用的數位憑證符合 X.509 標準，其中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，如果無法辨識，系統可以忽略它。

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

WebSphere 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 WebSphere MQ 是對等式系統，所以在 SSL 術語中被視為用戶端鑑別。 因此，任何用於 SSL 鑑別的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 WebSphere MQ 中憑證驗證原則的相關資訊，請參閱 第 28 頁的『IBM WebSphere 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 125 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。

在 DN 中通常會找到下列屬性類型:

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 WebSphere MQ 容許某些形態異常的 DN。 如需相關資訊，請參閱 WebSphere MQ 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。 憑證申請通常由您系統使用的憑證管理工具產生，例如 UNIX、 Linux®及 Windows 系統上的 iKeyman 工具，以及 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。

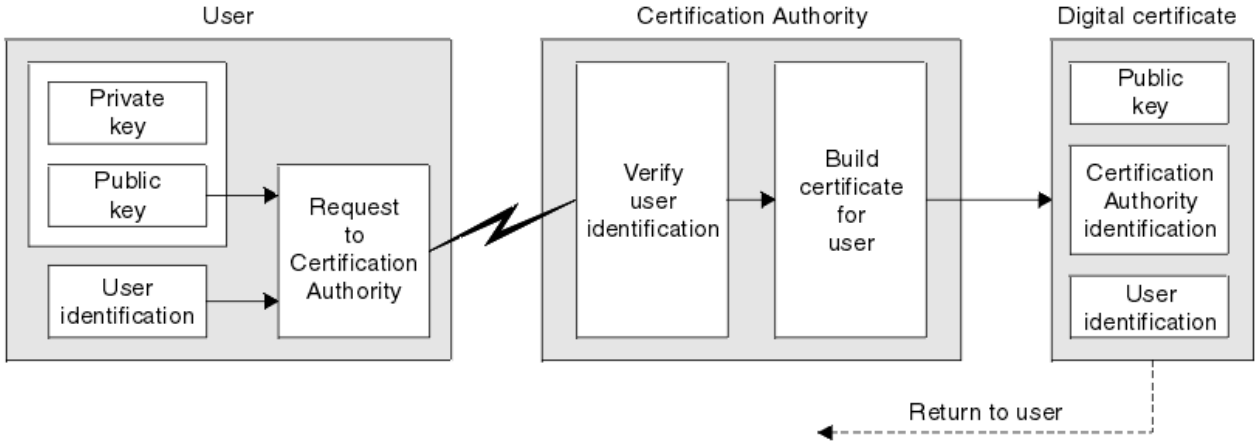


圖 *3:* 取得數位憑證

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。

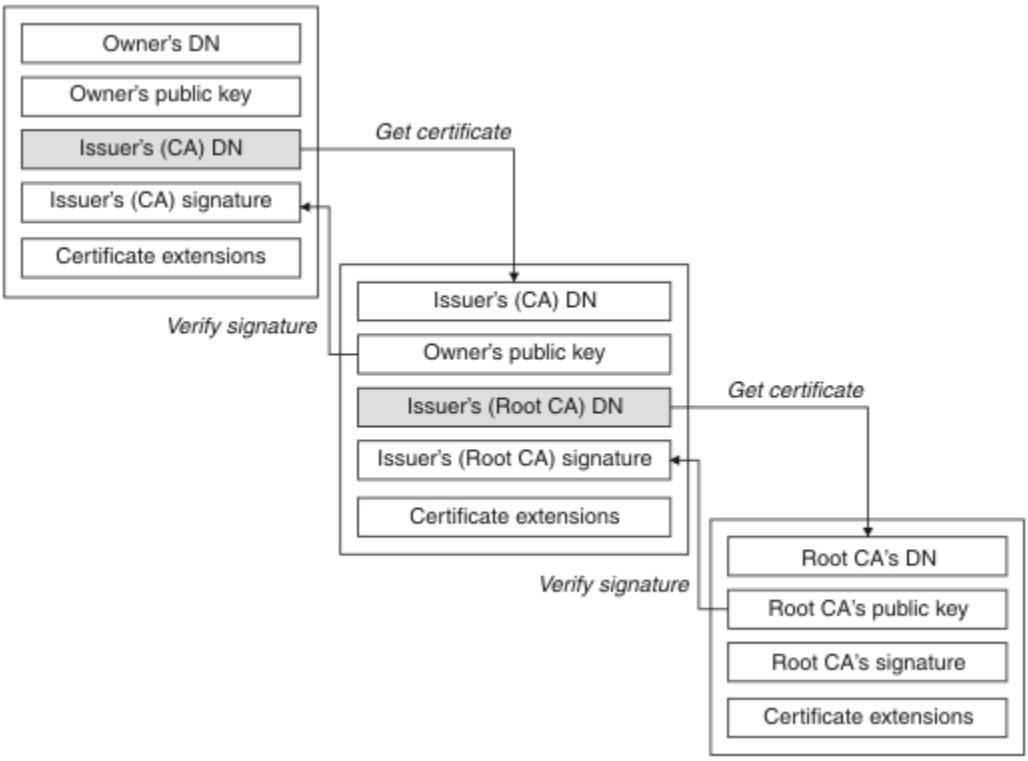


圖 *4:* 信任鏈

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。

如需憑證有效期限的定義，請參閱 名詞解釋 。

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。

WebSphere 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 UNIX、 Linux 及 Windows 系統) ，以檢查憑證是否已撤銷。 或者，他們可以在 LDAP 伺服器上存取 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 125 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』 。 RA 會驗證在要求數位憑證時所提供的資訊。 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: SSL 和 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 WebSphere MQ 同時支援 SSL 和 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 19 頁的『IBM WebSphere MQ 中的安全通訊協定』
IBM WebSphere MQ 同時支援傳輸層安全 (TLS) 及 Secure Sockets Layer (SSL) 通訊協定，以提供訊息通道及 MQI 通道的鏈結層次安全。

Secure Sockets Layer (SSL) 和傳輸層安全 *(TLS)* 概念

SSL 及 TLS 通訊協定可讓雙方識別及鑑別彼此，並以機密性及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。

SSL 及 TLS 通訊協定提供網際網路上的通訊安全，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。

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

IBM WebSphere MQ 支援的 CipherSpecs 完整清單提供於 第 182 頁的『指定 CipherSpecs』。

如需 SSL 通訊協定的相關資訊，請參閱 https://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt 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 或 *TLS* 信號交換的概觀

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

本節提供可讓 SSL 或 TLS 用戶端與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， SSL 或 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， SSL 信號交換所涉及的步驟如下:

1. SSL 或 TLS 用戶端會傳送 "client hello" 訊息，其中列出加密資訊 (例如 SSL 或 TLS 版本) ，並依照用戶端的喜好設定順序，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. SSL 或 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. SSL 或 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『SSL 和 TLS 如何提供識別、鑑別、機密性和完整性』。

4. SSL 或 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 SSL 或 TLS 伺服器傳送 "用戶端憑證要求"，則用戶端會傳送以用戶端私密金鑰加密的隨機位元組字串，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. SSL 或 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『SSL 和 TLS 如何提供識別、鑑別、機密性和完整性』。
7. SSL 或 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. SSL 或 TLS 伺服器會傳送 "已完成" 訊息給用戶端，以秘密金鑰加密，指出信號交換的伺服器部分已完成。
9. 在 SSL 或 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 SSL 或 TLS 信號交換。

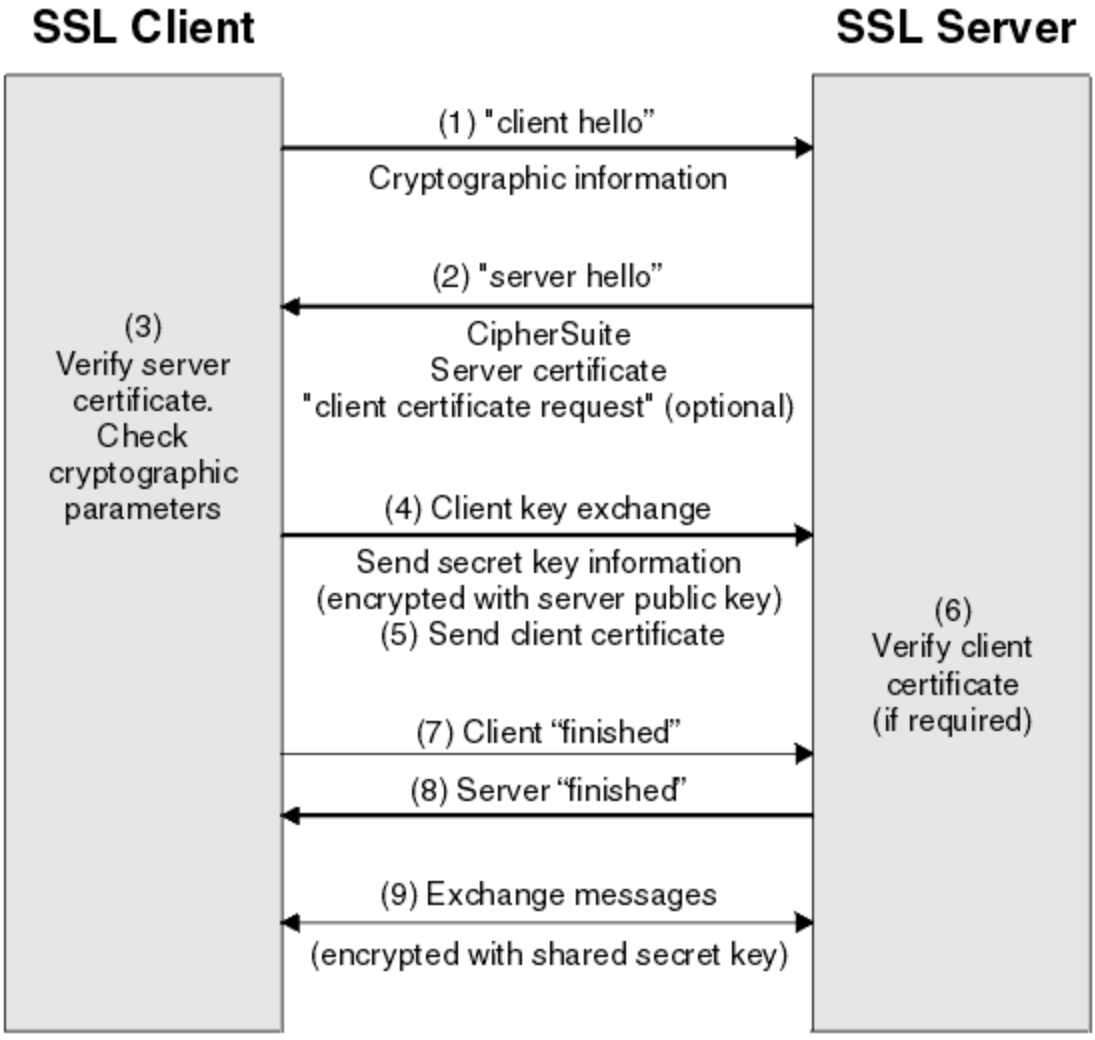


圖 *5: SSL* 或 *TLS* 信號交換的概觀

SSL 和 *TLS* 如何提供識別、鑑別、機密性和完整性

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 或 TLS 信號交換的概觀』。

SSL 和 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。

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的步驟 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 SSLor 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 SSL 或 TLS 用戶端，而 CA Y 會將憑證發出至 SSL 或 TLS 伺服器:

僅針對伺服器鑑別， SSL 或 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 SSL 或 TLS 用戶端需要:

- CA Y 的 CA 憑證

如果 SSL 或 TLS 伺服器需要用戶端鑑別，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

SSL 或 TLS 伺服器及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， SSL 或 TLS 用戶端會驗證伺服器的憑證，而 SSL 或 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 和 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 125 頁的『使用已撤銷的憑證』)。

重設秘密金鑰

在 SSL 或 TLS 信號交換期間，會產生 秘密金鑰 ，以加密 SSL 或 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。

SSL 和 TLS 如何提供機密性

SSL 和 TLS 使用對稱和非對稱加密的組合來確保訊息隱私權。 在 SSL 或 TLS 信號交換期間， SSL 或 TLS 用戶端及伺服器同意加密演算法及共用秘密金鑰僅用於一個階段作業。 在 SSL 或 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。 SSL 支援廣泛的加密演算法。 由於 SSL 和 TLS 在傳輸共用秘密金鑰時使用非對稱加密，因此沒有金鑰配送問題。 如需加密技術的相關資訊，請參閱 第 7 頁的『加密法』。

SSL 和 TLS 如何提供完整性

SSL 和 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 190 頁的『訊息的資料完整性』。

如果通道定義中的 CipherSpec 使用雜湊演算法 (如 第 182 頁的『指定 CipherSpecs』中的表格所述)，則使用 SSL 或 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 (或 SSL) 連線的兩端必須同意相同的 CipherSpec ，才能進行通訊。

重要: 在處理 IBM WebSphere 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 182 頁的『指定 CipherSpecs』。

CipherSuite 是 SSL 或 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，但只有在針對 TLS 或 SSL 連線指定時，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 SSL_RSA_WITH_RC4_128_MD5 指定:

- RSA 金鑰交換和鑑別演算法
- RC4 加密演算法，使用 128 位元金鑰
- MD5 MAC 演算法

有數種演算法可用於金鑰交換及鑑別，但 RSA 演算法目前是最廣泛使用的演算法。 所使用的加密演算法和 MAC 演算法有更多種類。

SSL 和 *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。

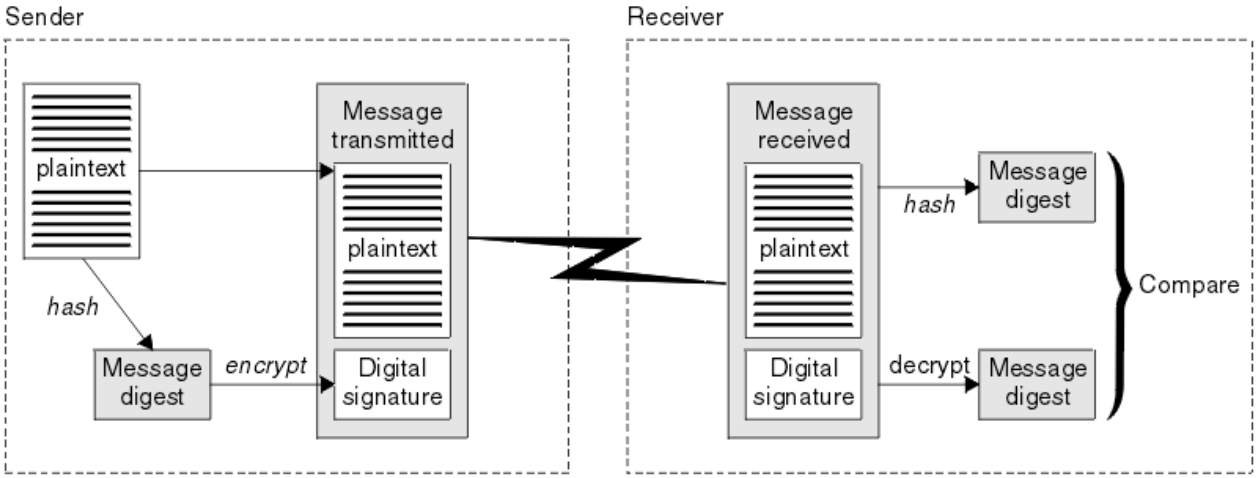


圖 *6:* 數位簽章處理程序

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。

IBM WebSphere 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。

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 版搭配使用的 Suite B 相容設定檔。 配置為「套組 B」相容作業時，只會使用上面列出的受限加密演算法集。
2. 與 TLS 1.0 版或 TLS 1.1 版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 Windows、 UNIX 及 Linux 系統上， WebSphere 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 25 頁的『IBM WebSphere MQ 中的 NSA Suite B 加密法』。

相關資訊

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 WebSphere MQ 安全機制

此主題集合說明如何在 IBM WebSphere MQ 中實作各種安全概念。

IBM WebSphere 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 WebSphere MQ 中的識別及鑑別

在 IBM WebSphere 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 WebSphere 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 WebSphere 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 WebSphere MQ 環境中可以執行的動作。

以下是 IBM WebSphere 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 WebSphere 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 41 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 WebSphere MQ 中的審核

IBM WebSphere MQ 可以發出事件訊息，以記錄發生異常活動。

以下是 IBM WebSphere 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 WebSphere MQ 中的機密性

您可以透過加密訊息，在 IBM WebSphere MQ 中實作機密性。

以下是一些範例，說明如何在 IBM WebSphere MQ 環境中確保機密性:

- 在傳送端 MCA 從傳輸佇列取得訊息之後， IBM WebSphere MQ 會使用 SSL 或 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 WebSphere 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 IBM WebSphere MQ Advanced Message Security 來加密儲存在佇列中的訊息。

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 WebSphere 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。

以下是一些範例，說明如何在 IBM WebSphere MQ 環境中確保資料完整性:

- 您可以使用 SSL 或 TLS 來偵測在透過網路傳輸訊息時是否刻意修改訊息內容。 在 SSL 和 TLS 中，訊息摘要演算法會偵測傳輸中已修改的訊息。 除了 TLS_RSA_WITH_NULL_NULL 未提供訊息資料完整性之外，所有 IBM WebSphere MQ CipherSpecs 都提供訊息摘要演算法。
- 當訊息儲存在本端佇列時， IBM WebSphere 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 IBM WebSphere MQ 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。

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 WebSphere MQ 中的加密法

IBM WebSphere MQ 使用 Secure Sockets Layer (SSL) 及傳輸安全層 (TLS) 通訊協定提供加密法。

如需相關資訊，請參閱 第 19 頁的『IBM WebSphere MQ 中的安全通訊協定』。

相關概念

第 6 頁的『加密概念』
此主題集合說明適用於 WebSphere MQ 的加密法概念。

IBM WebSphere MQ 中的安全通訊協定

IBM WebSphere MQ 同時支援傳輸層安全 (TLS) 及 Secure Sockets Layer (SSL) 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 SSL 或 TLS 通訊協定來提供鏈結層次安全。呼叫端 MCA 是 SSL 或 TLS 用戶端，而回應端 MCA 是 SSL 或 TLS 伺服器。 WebSphere MQ 支援 SSL 通訊協定 3.0 版，以及傳輸層安全 (TLS) 通訊協定 1.0 版和 1.2 版。 您可以提供 CipherSpec 作為通道定義的一部分，來指定 SSL 或通訊協定所使用的加密演算法。

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 SSL 或 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 WebSphere MQ 程式碼代表 WebSphere MQ 用戶端應用程式的使用者運作。 在 SSL 或 TLS 信號交換期間， WebSphere 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 WebSphere MQ 用戶端使用者作為 SSL 或 TLS 用戶端時，不需要有相關聯的個人數位憑證，除非在通道的伺服器端指定 SSLCAUTH (必要)。

數位憑證儲存在 金鑰儲存庫 中。 佇列管理程式屬性 *SSLKeyRepository* 指定保留佇列 管理程式的 數位憑證之金鑰儲存庫的位置。 在 WebSphere MQ 用戶端系統上， MQSSLKEYR 環境變數指定存放 使用者的 數位憑證之金鑰儲存庫的位置。 或者， WebSphere MQ 用戶端應用程式可以在 MQCONNX 呼叫中 SSL 及 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。

相關概念

第 13 頁的『加密安全通訊協定: SSL 和 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 WebSphere MQ 同時支援 SSL 和 TLS。

IBM WebSphere MQ 支援 *SSL* 和 *TLS*

IBM WebSphere MQ 同時支援 Secure Sockets Layer (SSL) 通訊協定和傳輸層安全 (TLS) 通訊協定。

如需 SSL 及 TLS 通訊協定的相關資訊，請參閱相關資訊。

IBM WebSphere MQ 提供下列 SSL 版本 3.0 及 TLS 1.0 及 TLS 1.2:

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

UNIX, Linux, and Windows 及 HP Integrity NonStop Server 系統
對於 UNIX, Linux, and Windows 和 HP Integrity NonStop Server 系統， SSL 和 TLS 支援隨 IBM WebSphere MQ 一起安裝。

如需 IBM WebSphere MQ SSL 和 TLS 支援的任何必要條件的相關資訊，請參閱 IBM WebSphere MQ 的系統需求。

SSL 或 *TLS* 金鑰儲存庫
相互鑑別的 SSL 或 TLS 連線在連線的每一端都需要金鑰儲存庫 (在不同平台上可以用不同名稱來識別)。 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 SSL 和 TLS 的平台和環境上使用的特定商店名稱如下:

Java 和 JMS	金鑰儲存庫和信任儲存庫
Windows UNIX Linux Windows UNIX Linux	金鑰資料庫檔
Windows 、 UNIX and Linux 系統	

如需相關資訊，請參閱 第 9 頁的『數位憑證』 和 第 13 頁的『Secure Sockets Layer (SSL) 和傳輸層安全 (TLS) 概念』。

相互鑑別的 SSL 或 TLS 連線在連線的每一端都需要一個金鑰儲存庫。 金鑰儲存庫可能包含:

- 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 WebSphere MQ MQI 用戶端相關聯。 如果需要交互鑑別，則個人憑證在 SSL 或 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 21 頁的『保護 IBM WebSphere MQ 金鑰儲存庫』。

金鑰儲存庫的位置視您使用的平台而定:

Windows、 UNIX and Linux 系統

在 Windows、 UNIX and Linux 系統上，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 UNIX 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 WebSphere MQ 安裝在預設位置，則 Windows 上的對等路徑是 C:\Program Files\IBM\WebSphere MQ\Qmgrs\QM1\ssl\key.kdb。

在 Windows 、 UNIX and Linux 系統上，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: 在 Windows、 UNIX and Linux 系統上， PKCS #11 加密硬體卡可以包含金鑰資料庫檔中所保留的憑證和金鑰。 當憑證和金鑰保留在 PKCS #11 卡上時， WebSphere MQ 仍需要同時存取金鑰資料庫檔和密碼隱藏檔。

在 Windows 及 UNIX 系統上，金鑰資料庫也包含與佇列管理程式或 WebSphere MQ MQI 用戶端相關聯之個人憑證的私密金鑰。

保護 *IBM WebSphere MQ* 金鑰儲存庫

IBM WebSphere 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 WebSphere 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。

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「 WebSphere MQ 探險家」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中「 WebSphere MQ 探險家」一節的 MQCMD_REFRESH_SECURITY 指令 及 重新整理 *SSL* 或 *TLS* 安全 主題。

相關概念

第 21 頁的『重新整理用戶端的 SSL 金鑰儲存庫內容和 SSL 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* 金鑰儲存庫內容和 *SSL* 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。

您無法在 WebSphere MQ 用戶端上重新整理安全; 用戶端沒有同等的 REFRESH SECURITY TYPE (SSL) 指令 (如需相關資訊，請參閱 重新整理安全)。

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 21 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。

聯邦資訊存取安全標準 *(FIPS)*

本主題介紹美國國家標準與技術機構 (US National Institute of Standards and Technology) 的「聯邦資訊存取安全標準 (FIPS) Cryptomodule 驗證程式」， 以及可在 Windows、 UNIX and Linux 及 z/OS 系統的 SSL 或 TLS 通道上使用的加密函數。

在 UNIX、 Linux 及 Windows 系統上， IBM WebSphere MQ SSL 或 TLS 連線的 FIPS 140-2 相符性位於這裡 第 22 頁的『 UNIX、 Linux 及 Windows 的聯邦資訊存取安全標準 (FIPS)』。

如果存在加密硬體，則 IBM WebSphere 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 WebSphere 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 90 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 95 頁的『使用 iKeyman、 iKeycmd、runmqakm 和 runmqckm』
在 UNIX、 Linux 及 Windows 系統上，使用 iKeyman GUI 或從指令行使用 iKeycmd 或 runmqakm 來管理金鑰及數位憑證。

相關工作

在適用於 Java 的 WebSphere MQ 類別中啟用 SSL

搭配使用 Secure Sockets Layer (SSL) 與適用於 JMS 的 WebSphere MQ 類別

相關參考

JMS 物件的 SSL 內容

相關資訊

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)」。

UNIX、 *Linux* 及 *Windows* 的聯邦資訊存取安全標準 *(FIPS)*

當 Windows、 UNIX and Linux 系統上的 SSL 或 TLS 通道需要加密法時， WebSphere MQ 會使用稱為 IBM Crypto for C (ICC) 的加密法套件。 在 Windows(UNIX and Linux 平台) 上， ICC 軟體已通過美國國家標準與技術機構層次 140-2 的「聯邦資訊存取安全標準 (FIPS) Cryptomodule 驗證程式」。

在 Windows UNIX and Linux 系統上， WebSphere MQ SSL 或 TLS 連線的 FIPS 140-2 標準如下:

- 對於所有 IBM WebSphere 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 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 WebSphere MQ MQI 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之 Java 應用程式的 IBM WebSphere MQ 類別，如果符合下列條件，連線會使用 JRE 的 SSL 和 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。
  - 您已指定只使用 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之 JMS 應用程式的 IBM WebSphere MQ 類別，如果符合下列條件，連線會使用 JRE 的 SSL 和 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。
  - 您已指定只使用 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。
  - 您已指定只使用 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。
  - 您已指定只使用 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 AIX、 Linux、 HP-UX、Solaris、 Windows 及 z/OS 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 SSL 及 TLS 連線，經 FIPS 140-2 認證的元件稱為 *ICC*。 此元件的版本決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: Licensed Materials-Property of IBM
@ (#) ICC
@ (#) © Copyright IBM Corp. 2002 , 2024.
@ (#) 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 WebSphere 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。

註: 在 Intel 系統上執行時，針對 FIPS 140-2 相容作業配置的 32 位元 Solaris x86 SSL 及 TLS 用戶端失敗。 由於未在 Intel 晶片組上載入符合 FIPS 140-2 標準的 GSKit-Crypto Solaris x86 32 位元程式庫檔案，因此才會發生此故障。 在受影響的系統上，將在用戶端錯誤日誌中報告錯誤 AMQ9655。 若要解決此問題，請停用 FIPS 140-2 相符性或重新編譯用戶端應用程式 64 位元，因為 64 位元程式碼不受影響。

符合 FIPS 140-2 標準運作時施行三重 DES 演算法限制

當 WebSphere 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， WebSphere MQ 不會重設秘密階段作業金鑰，因此必須配置這項重設。 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 188 頁的『重設 SSL 和 TLS 秘密金鑰』。

WebSphere MQ 會產生已符合規則 1 和 2 的三重 DES 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 90 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 95 頁的『使用 iKeyman、 iKeycmd、runmqakm 和 runmqckm』
在 UNIX、 Linux 及 Windows 系統上，使用 iKeyman GUI 或從指令行使用 iKeycmd 或 runmqakm 來管理金鑰及數位憑證。

相關工作

在適用於 Java 的 WebSphere MQ 類別中啟用 SSL

搭配使用 Secure Sockets Layer (SSL) 與適用於 JMS 的 WebSphere MQ 類別

相關參考

JMS 物件的 SSL 內容

相關資訊

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 WebSphere MQ MQI 用戶端上的 *SSL* 及 *TLS*

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

IBM WebSphere MQ 為 Windows UNIX and Linux 系統上的 IBM WebSphere MQ MQI 用戶端提供 SSL 及 TLS 支援。 如果您使用適用於 Java 的 IBM WebSphere MQ 類別，請參閱 使用適用於 Java 的 WebSphere MQ 類別 ，如果您使用適用於 JMS 的 IBM WebSphere MQ 類別，請參閱 使用適用於 JMS 的 WebSphere MQ 類別。 此區段的其餘部分不適用於 Java 或 JMS 環境。

您可以在 IBM WebSphere 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 WebSphere MQ MQI 用戶端的金鑰儲存庫。 您有三個選項可指定通道使用 SSL:

- 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 SSL。

只要通道另一端未指定 SSL ，您可以繼續執行現有的 IBM WebSphere MQ MQI 用戶端應用程式 (不使用 SSL)。

如果在用戶端機器上對「SSL 金鑰儲存庫」的內容、「SSL 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 SSL 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。 一旦所有連線都結束，請重新啟動 SSL 通道。 會使用所有新的 SSL 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 WebSphere MQ MQI 用戶端在具有加密硬體的 Windows UNIX and Linux 系統上執行時，您會使用 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 WebSphere MQ MQI 用戶端支援 SSL 秘密金鑰重設及 FIPS。 如需相關資訊，請參閱 第 188 頁的『重設 SSL 和 TLS 秘密金鑰』 及 第 22 頁的『 UNIX、 Linux 及 Windows 的聯邦資訊存取安全標準 (FIPS)』。

如需 IBM WebSphere MQ MQI 用戶端的 SSL 支援的相關資訊，請參閱 第 89 頁的『設定 IBM WebSphere MQ MQI 用戶端安全』 。

相關工作

使用配置檔來配置用戶端

指定 *MQI* 通道使用 *SSL*

若要讓 MQI 通道使用 SSL ，用戶端連線通道的 *SSLCipherSpec* 屬性值必須是用戶端平台上 IBM WebSphere 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 WebSphere MQ MQI 用戶端應用程式發出 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 WebSphere MQ 用戶端存取的任何項目。

您無法使用 MQSERVER 環境變數，在使用 SSL 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 187 頁的『指定 IBM WebSphere MQ MQI 用戶端的 CipherSpec』
您有三個選項可用來指定 IBM WebSphere MQ MQI 用戶端的 CipherSpec 。

IBM WebSphere MQ 中的 *CipherSpecs* 和 *CipherSuites*

IBM WebSphere MQ 同時支援 SSL 和 TLS CipherSpecs， 以及 RSA 和 Diffie-Hellman 演算法。

WebSphere MQ 支援 SSL V3 及 TLS V1.0 及 V1.2 CipherSpecs。

WebSphere MQ 支援 RSA 和 Diffie-Hellman 金鑰交換及鑑別演算法。 SSL 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 WebSphere MQ 中的 *NSA Suite B* 加密法

本主題提供如何在 Windows、 Linux 及 UNIX 系統上配置 IBM WebSphere MQ 以符合 Suite 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 WebSphere 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 WebSphere MQ Version 7.5 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 WebSphere 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 及 MQSeries 產品 READMEs)。

在 Windows、 UNIX 及 Linux 系統上， IBM WebSphere 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 *1:* 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 WebSphere MQ 的預設行為不符合「套組 B」標準，但 IBM WebSphere MQ 可以配置為符合 Windows、 UNIX 及 Linux 系統上的任一或兩個安全層次。 在成功配置 IBM WebSphere 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 WebSphere MQ CipherSpecs 的相關資訊，請參閱 第 182 頁的『指定 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，WebSphere MQ 必須配置成使用遠端友機的已配置套組 B 安全層次容許數位簽章演算法和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: iKeycmd 和 **iKeyman** 工具不支援建立套組 B 相容作業的數位憑證。

建立及要求數位憑證

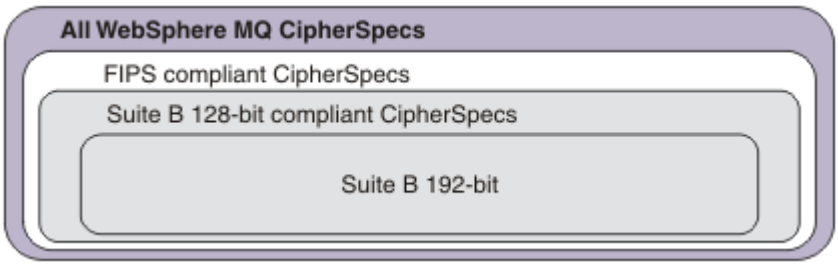
若要建立用於套組 B 測試的自簽數位憑證，請參閱 第 102 頁的『在 UNIX, Linux, and Windows 系統上建立自簽個人憑證』

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 104 頁的『在 UNIX, Linux, and Windows 系統上要求個人憑證』。

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

「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 WebSphere 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 WebSphere 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 WebSphere MQ

如需如何在 Windows、UNIX 及 Linux 上配置 IBM WebSphere MQ 以符合 Suite B 標準作業的相關資訊，請參閱 第 27 頁的『為套組 B 配置 IBM WebSphere MQ』。

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

相關概念

第 90 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。

為套組 *B* 配置 *IBM WebSphere MQ*
IBM WebSphere MQ 可以配置為在 UNIX, Linux, and Windows 系統上遵循 NSA Suite B 標準來運作。

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 WebSphere MQ 可以配置為遵循「套組 B」來操作，以提供加強的安全等級。 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 Suite B 配置及其對 SSL 和 TLS 通道的影響的相關資訊，請參閱 第 25 頁的『IBM WebSphere 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，以針對套組 B 相容作業配置佇列管理程式

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. 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。

.NET

對於 .NET 未受管理用戶端，內容 **MQC.ENCRYPTION_POLICY_SUITE_B** 指出所需的 Suite B 安全類型。

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

IBM WebSphere 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 UNIX, Linux, and Windows 系統，憑證驗證原則由 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 UNIX, Linux, and Windows 系統上配置憑證驗證原則的相關資訊，請參閱 第 28 頁的『在 IBM WebSphere MQ 中配置憑證驗證原則』。 如需「基本」和「標準」憑證驗證原則之間差異的相關資訊，請參閱 UNIX、 Linux 和 Windows 系統上的憑證驗證和信任原則設計 。

在 *IBM WebSphere MQ* 中配置憑證驗證原則

您可以用四種方式來指定使用哪一個 SSL/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 28 頁的『IBM WebSphere MQ 中的憑證驗證原則』。

IBM WebSphere 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 WebSphere 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。

在舊版 IBM WebSphere MQ 中，所有支援的 SSL 及 TLS CipherSpecs 都使用 RSA 演算法來進行數位簽章及金鑰合約。 所有支援的數位憑證類型都與所有支援的 CipherSpecs 相容，因此可以變更任何通道的 CipherSpec ，而不需要變更數位憑證。

在 IBM WebSphere MQ v7.5 中，只能將受支援 CipherSpecs 的子集與所有受支援類型的數位憑證搭配使用。 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。 如果您想要使用基於 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 SSL 3.0 及 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` 是您需要顯示之數位簽章演算法的憑證標籤。

註: 雖然 **iKeycmd (runmqckm)** 工具及 **iKeyman (strmqikm)** GUI 可用來檢視數位簽章演算法的選項，但 **runmqakm** 工具提供更廣泛的範圍。

執行 **runmqakm** 指令會產生輸出，顯示使用指定的簽章演算法:

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

```
    66 6A 65 CB C3 C4 E2 CC F5 49 39 A3 8B 93 5A DD
    B0 21 0B A8 B2 59 5B 24 59 50 44 89 DC 78 19 51
Trust Status : Enabled
```

`Signature Algorithm` 行顯示使用 `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

並非所有 CipherSpecs 都可以與所有數位憑證搭配使用。 CipherSpec 有三種類型，以 CipherSpec 名稱字首表示。 每一種類型的 CipherSpec 對可能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 WebSphere MQ SSL 及 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 *2: 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_	CipherSpecs，使用 RSA 公開金鑰、橢圓曲線秘密金鑰及橢圓曲線數位簽章演算法。	RSA	RSA	ECDHE
3	(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

註: 只有 UNIX, Linux, and Windows 平台上的 WebSphere MQ 佇列管理程式及 MQI 用戶端才支援類型 1 及 2 CipherSpecs 。

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。

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 with MD5" 或 "ECDSA with 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 要求個人憑證必須具有 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 WebSphere 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。

WebSphere MQ 佇列管理程式只能使用單一個人憑證來識別本身。 這表示佇列管理程式上的所有通道都將使用相同的數位憑證，因此每一個佇列管理程式一次只能使用一種 CipherSpec 類型。 同樣地， WebSphere MQ 用戶端應用程式只能使用單一個人憑證來識別本身。 這表示單一應用程式程序內的所有 SSL 和 TLS 連線都將使用相同的數位憑證，因此每個用戶端應用程式程序一次只能使用一種 CipherSpec 類型。

這三種類型的 CipherSpec 不會直接交互作業: 這是現行 SSL 和 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。 ECDHE_ECDSA_AES_128_CBC_SHA256 是類型 1 CipherSpec，因此 QM1 必須具有具有橢圓曲線金鑰及 ECDSA 型數位簽章的個人憑證。 因此，所有與 QM1 直接通訊的用戶端及其他佇列管理程式都必須具有滿足類型 1 CipherSpec 需求的數位憑證。 連接至佇列管理程式 QM1 的其他通道可以使用其他 CipherSpecs (例如 ECDHE_ECDSA_3DES_EDE_CBC_SHA256) ，但它們只能使用第 1 類 CipherSpecs 來與 QM1 進行通訊。

規劃 WebSphere MQ 網路時，請仔細考量哪些通道需要 SSL 或 TLS ，並確保所有需要交互作業的用戶端及佇列管理程式都使用相同類型的 CipherSpecs 及適當的數位憑證。 IETF 標準 RFC 4492、RFC 5246 及 RFC 6460 說明「橢圓曲線」 CipherSpecs 在 TLS 1.2 中的詳細用法。

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的憑證標籤。

執行 **runmqakm** 指令將產生顯示「公開金鑰類型」的輸出:

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

在此情況下，「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。

您也可以使用具有相同參數的 **iKeycmd (runmqckm)** 工具。 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 **iKeyman (strmqikm)** GUI 來檢視數位簽章演算法。 不過，建議您使用 **runmqakm** 工具來檢視數位憑證，因為它支援範圍更廣的演算法。

橢圓曲線 CipherSpecs 與 NSA Suite B

當 WebSphere 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 25 頁的『IBM WebSphere 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 *3:* 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 182 頁的『指定 CipherSpecs』
在 **DEFINE CHANNEL** MQSC 指令或 **ALTER CHANNEL** MQSC 指令中使用 **SSLCIPH** 參數，以指定 CipherSpec 。

第 90 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 25 頁的『IBM WebSphere MQ 中的 NSA Suite B 加密法』
本主題提供如何在 Windows、 Linux 及 UNIX 系統上配置 IBM WebSphere MQ 以符合 Suite 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。

IBM WebSphere MQ 中支援的 *CipherSpec* 值

預設 CipherSpecs 集僅容許下列值:

TLS 1.0

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA

TLS 1.2

- ECDHE_ECDSA_AES_128_CBC_SHA256
- ECDHE_ECDSA_AES_256_CBC_SHA384
- ECDHE_ECDSA_AES_128_GCM_SHA256
- ECDHE_ECDSA_AES_256_GCM_SHA384
- ECDHE_RSA_AES_128_CBC_SHA256
- ECDHE_RSA_AES_256_CBC_SHA384
- ECDHE_RSA_AES_128_GCM_SHA256
- ECDHE_RSA_AES_256_GCM_SHA384

- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_GCM_SHA384

啟用已淘汰的 CipherSpecs

依預設，不容許您在通道定義上指定已淘汰的 CipherSpec 。 如果您嘗試指定已淘汰的 CipherSpec，則會在佇列管理程式的錯誤日誌中收到訊息 AMQ9788 。

您可以透過編輯 `qm.ini` 檔案來重新啟用已淘汰的 CipherSpecs 。 在 `qm.ini` 檔的 SSL 段落內，新增下列這一行:

```
SSL:
AllowWeakCipherSpec=Yes
```

您也可以將環境變數 *AMQ_SSL_WEAK_CIPHER_ENABLE* 設為任何值，以在伺服器上執行時期重新啟用一個以上已淘汰的 CipherSpecs 。 不論 `qm.ini` 檔案中指定的值為何，此環境變數都會啟用 CipherSpecs 。

通道鑑別記錄

若要在通道層次對授與用來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。

如果您已設定管理工具（如「IBM WebSphere MQ Explorer」），以及用於該特定用途的通道，則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 *4:* 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。 但可能會存在以下情況：您遇到從不應該存取 WebSphere MQ 系統的 IP 位址進行的連線嘗試，在可以更新防火牆之前，必須暫時封鎖此位址。 這些連線嘗試甚至可能不是來自 WebSphere MQ 通道，而是來自以 WebSphere MQ 接聽器為目標的其他 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 152 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 153 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 156 頁的『封鎖用戶端主張使用者 ID 的存取』 。

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 155 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 156 頁的『封鎖存取 SSL 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 157 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 154 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

如果要指定在來自 WebSphere MQ MQI 用戶端的連線使用某個使用者 ID 時，將會使用不同的指定 MCAUSER，請設定 USERMAP 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 155 頁的『將用戶端主張的使用者 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 155 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 154 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。

如果有許多通道鑑別記錄符合某個通道名稱、IP 位址、佇列管理程式名稱或是 SSL 或 TLS DN，則會使用最具體的相符項。 視為最具體的相符項將按以下方式來決定。

- 針對通道名稱：
  - 最具體的相符項是沒有萬用字元的名稱，例如 A.B.C。
  - 最通用的相符項是單一星號 (*)，它符合所有通道名稱。
  - 在最左側位置具有星號的型樣，比在最左側位置具有已定義值的型樣更通用。 因此，*.B.C 比 A.* 更通用。
  - 在第二個位置具有星號的型樣，比在第二個位置具有已定義值的型樣更通用，對於後續每個位置，情況亦類似。 因此，A.*.C 比 A.B.* 更通用。
  - 如果兩個以上的型樣在相同位置都有一個星號，則星號後面節點較少的型樣更通用。 因此 A. * 比 A.*.C 更通用
- 針對 IP 位址：
  - 最具體的相符項是沒有萬用字元的名稱，例如 192.0.2.6。
  - 最通用的相符項是單一星號 (*)，它符合所有通道名稱。
  - 在最左側位置具有星號的型樣，比在最左側位置具有已定義值的型樣更通用。 因此，*.0.2.6 比 192.* 更通用。
  - 在第二個位置具有星號的型樣，比在第二個位置具有已定義值的型樣更通用，對於後續每個位置，情況亦類似。 因此，192.*.2.6 比 192.0.* 更通用。
  - 如果兩個以上的型樣在相同位置都有一個星號，則星號後面節點較少的型樣更通用。 因此 192 .* 比 192.*.2.*. 更通用
  - 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。
  - 只要在緊鄰尾端星號的位置沒有星號，則型樣可以包含其他星號。 例如，192.*.2.* 有效，但 192.0.*.* 是無效的。
  - IPv6 位址型樣不得包含雙冒號及尾端星號，因為所產生的位址可能會不明確。 例如，2001::* 可以擴充為 2001:0000:*、2001:0000:0000:* 等
- 針對佇列管理程式名稱：
  - 最具體的相符項是沒有萬用字元的名稱，例如 192.0.2.6。
  - 最通用的相符項是單一星號 (*)，它符合所有通道名稱。

– 在最左側位置具有星號的型樣，比在最左側位置具有已定義值的型樣更通用。 因此，*QUEUEMANAGER 比 QUEUEMANAGER* 更通用。
– 在第二個位置具有星號的型樣，比在第二個位置具有已定義值的型樣更通用，對於後續每個位置，情況亦類似。 因此，Q*MANAGER 比 QUEUE* 更通用。
– 如果兩個以上的型樣在相同位置都有一個星號，則星號後面字元較少的型樣更通用。 因此，Q* 比 Q*MGR 更通用。

- 對於 SSL 或 TLS「識別名稱 (DN)」，子字串的優先順序如下：

表 *5:* 子字串的優先順序

訂購	**DN 子字串**	**姓名**
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=	一般名稱
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 日	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，如果隨 SSL 或 TLS 憑證一起提供的 DN 中包含子字串 O=IBM 及 C=UK，則 WebSphere MQ 會優先於 C=UK 的通道鑑別記錄，來使用 O=IBM 的通道鑑別記錄（如果兩者同時存在）。

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 46 頁的『遠端傳訊的安全』

本節處理安全的遠端傳訊層面。

IBM WebSphere MQ 中的訊息安全

IBM WebSphere MQ 基礎架構中的訊息安全由個別授權元件 IBM WebSphere MQ Advanced Message Security 提供。

IBM WebSphere MQ Advanced Message Security (AMS) 會擴充 IBM WebSphere 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 227 頁的『IBM WebSphere MQ Advanced Message Security』

IBM WebSphere MQ Advanced Message Security (AMS) 是 IBM WebSphere MQ Advanced Message Security 的個別授權元件，可為流經 IBM WebSphere MQ Advanced Message Security 網路的機密資料提供高階保護，同時不會影響終端應用程式。

規劃安全需求

此主題集合說明在 IBM WebSphere MQ 環境中規劃安全時需要考量的事項。

您可以將 IBM WebSphere 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。

WebSphere MQ 提供一系列鏈結層次安全服務，包括支援 Secure Sockets Layer (SSL) 和傳輸層安全 (TLS)。

在實作 WebSphere 時，您必須考量安全的某些方面。 在 UNIX、 Linux 及 Windows 系統上，如果您忽略這些層面而不執行任何動作，則無法使用 WebSphere MQ。

以下說明安全考量。

管理 WebSphere MQ 的權限

WebSphere MQ 管理者需要有下列權限:

- 發出指令以管理 WebSphere MQ
- 使用 IBM WebSphere MQ Explorer

如需相關資訊，請參閱:

- 第 165 頁的『在 UNIX, Linux, and Windows 系統上管理 IBM WebSphere MQ 的權限』

使用 WebSphere MQ 物件的權限

應用程式可以透過發出 MQI 呼叫來存取下列 WebSphere 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 WebSphere MQ 物件，以及存取通道和鑑別資訊物件。 這些物件可以受 WebSphere 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 42 頁的『授權應用程式使用 IBM WebSphere MQ』。

通道安全性

與訊息通道代理程式 (MCA) 相關聯的使用者 ID 需要權限，才能存取各種 WebSphere MQ 資源。 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 56 頁的『通道授權』。

其他考量

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

- 第 63 頁的『佇列管理程式叢集的安全』
- 第 64 頁的『IBM WebSphere MQ 發佈/訂閱的安全』
- 第 65 頁的『IBM WebSphere MQ 網際網路透通的安全』

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 WebSphere 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 SSL 或 TLS 的 IBM WebSphere 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 WebSphere MQ 進行鑑別。 如需相關資訊，請參閱:

- 第 154 頁的『將遠端佇列管理程式對映至 MCAUSER 使用者 ID』
- 第 155 頁的『將用戶端主張的使用者 ID 對映至 MCAUSER 使用者 ID』
- 第 155 頁的『將 SSL 或 TLS 識別名稱對映至 MCAUSER 使用者 ID』
- 第 157 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。 此圖顯示連接至伺服器的 IBM WebSphere MQ MQI 用戶端。 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」。

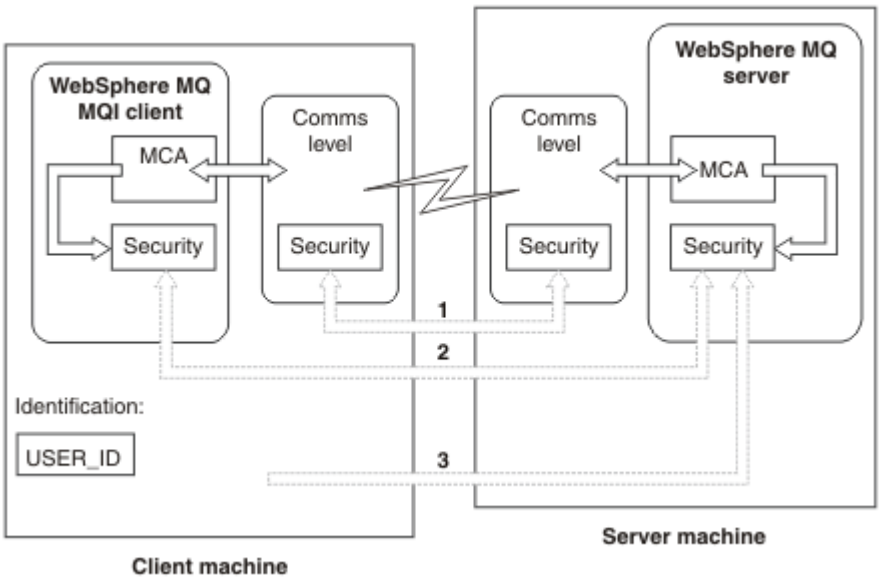


圖 7: 主從式連線中的安全

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 SSL 或 TLS。 如需相關資訊，請參閱 第 13 頁的『加密安全通訊協定: SSL 和 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 SSL/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 33 頁的『通道鑑別記錄』中提供完整說明。

3. 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。

4. 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 WebSphere 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。

   Windows 用戶端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。

   為了協助識別 HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端，用戶端會傳遞執行用戶端應用程式所使用的 OSS Safeguard 別名。 此 ID 通常格式為 <PRIMARYGROUP>.<ALIAS>。 必要的話，您可以使用通道鑑別記錄或安全結束程式，將此使用者 ID 對映至佇列管理程式上的替代使用者 ID。 如需訊息結束程式的相關資訊，請參閱 第 124 頁的『訊息結束程式中的身分對映』。 如需定義通道鑑別記錄的相關資訊，請參閱 第 155 頁的『將用戶端主張的使用者 ID 對映至 MCAUSER 使用者 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 WebSphere MQ MQI 用戶端的身分。

使用者 *ID*

如果 IBM WebSphere MQ MQI 用戶端位於 Windows 上，且 IBM WebSphere MQ 伺服器位於 Windows 上，且可以存取用戶端使用者 ID 定義所在的網域，則 IBM WebSphere MQ 支援最多 20 個字元的使用者 ID。 在 UNIX and Linux 平台和配置上，長度上限為 12 個字元。

如果用戶端以包含 @ 字元 (例如 abc@d) 的使用者 ID 執行，則 WebSphere MQ for Windows 伺服器不支援 Windows 用戶端的連線。 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。

請注意， UNKNOWN 是保留的使用者 ID，而 NOBODY 使用者 ID 也對 WebSphere MQ 具有特殊意義。 在稱為 UNKNOWN 或 NOBODY 的作業系統中建立使用者 ID 可能會產生非預期的結果。

雖然使用者 ID 用於鑑別，但群組用於授權 (Windows 除外)。

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。

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 WebSphere MQ 物件，包括從 IBM WebSphere MQ MQI 用戶端連接的使用者。

必須授與個人或應用程式存取權，才能使用 IBM WebSphere 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 WebSphere 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 WebSphere 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 WebSphere MQ 的權限

IBM WebSphere 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 WebSphere MQ 管理者需要下列權限:

- 發出指令以管理 IBM WebSphere MQ
- 使用 IBM WebSphere MQ Explorer

如需相關資訊，請參閱適合您作業系統的主題。

在 *UNIX* 和 *Windows* 系統上管理 *IBM WebSphere MQ* 的權限

IBM WebSphere MQ 管理者是 mqm 群組的成員。 此群組具有所有 IBM WebSphere MQ 資源的存取權，並且可以發出 IBM WebSphere 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 UNIX 及 Windows 系統上成為 IBM WebSphere MQ 管理者，使用者必須是 *mqm* 群組的成員。 當您安裝 WebSphere 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 UNIX 系統上的 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 WebSphere MQ 控制指令，且他們有權只執行他們已獲授與存取權的指令。

此外，在 Windows 系統上， SYSTEM 及 Administrator 帳戶對 IBM WebSphere MQ 資源具有完整存取權。

mqm 群組的所有成員都可以存取系統上的所有 WebSphere 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 群組的成員也可以存取所有 WebSphere MQ 資源。

管理者可以使用控制指令 **runmqsc** 來發出 WebSphere 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。

「WebSphere MQ 探險家」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「WebSphere MQ 探險家」來管理本端系統上的佇列管理程式。 當使用「WebSphere MQ 探險家」來管理另一個系統上的佇列管理程式時，管理者必須具有必要權限，才能讓遠端佇列管理程式處理 PCF 指令。

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 42 頁的『授權應用程式使用 IBM WebSphere 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。

如需在 UNIX 和 Windows 系統上管理 WebSphere MQ 所需之權限的相關資訊，請參閱相關資訊。

授權應用程式使用 IBM WebSphere 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 WebSphere 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 WebSphere 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，應用程式包括使用者及供應商所撰寫的應用程式。

使用 IBM WebSphere MQ classes for Java、 IBM WebSphere MQ classes for JMS、 IBM WebSphere MQ classes for .NET 或 Message Service Clients for C/C++ and .NET 的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，需要權限才能存取這些 WebSphere MQ 物件。 如需這些使用者 ID 及其所需權限的相關資訊，請參閱 第 56 頁的『通道授權』。

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 WebSphere MQ。 IBM WebSphere 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 WebSphere 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。 會對與遠端佇列管理程式同名的傳輸佇列執行權限檢查。 在 UNIX, Linux, and Windows 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 WebSphere 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 43 頁的『訊息環境定義』。

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。

在 *UNIX*、 *Linux* 及 *Windows* 系統上使用 *IBM WebSphere MQ* 物件的權限

IBM WebSphere 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。

1. 鑑別。

   IBM WebSphere 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* 屬性具有 MQCSP_AUTH_USER_ID_AND_PWD 值，則由 OAM 在其 MQZID_AUTHENTICATE_USER 函數中執行檢查。 這是檢查: MQCSP 結構中的使用者 ID 會與 *IdentityContext* (MQZIC) 中的使用者 ID 相互比較，以判斷它們是否相符。 如果它們不相符，則檢查會失敗。

此基本檢查不是要作為使用者的完整鑑別。 例如，透過檢查 MQCSP 結構中提供的密碼，不會檢查使用者的確實性。 此外，如果應用程式省略 MQCSP 結構，則不會執行任何檢查。

如果透過授權服務元件在佇列管理程式中需要更完整的鑑別服務，則 IBM WebSphere MQ 隨附的 OAM 不會提供此服務。 您必須撰寫新的授權服務元件，或從供應商取得授權服務元件。

2. 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 UNIX、 Linux 及 Windows 系統上，當應用程式發出 MQI 呼叫來存取 IBM WebSphere 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。

在 Windows 上， OAM 會授與 Administrators 群組成員存取所有 IBM WebSphere 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 WebSphere MQ 資源具有完整存取權。

當 PCF 指令在其中一個 IBM WebSphere 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。

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 WebSphere MQ 產品所提供的元件。

IBM WebSphere 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 WebSphere MQ 物件的存取控制清單 (ACL)。 在 UNIX 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 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 及 Linux 已提高此限制，但 IBM WebSphere 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 WebSphere MQ 會將它取代為值 "UNKNOWN"。 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 WebSphere 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。

在 UNIX、 Linux 及 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。

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。

使用 *PCF* 存取 *OAM* 指令
在 UNIX、 Linux 及 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 *6: 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 。

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 WebSphere 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 SSL 或 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。

UNIX and Linux 系統上物件的安全

如果此 ID 將使用 IBM WebSphere 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。

UNIX and Linux 系統上的使用者 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 系統上物件的安全

如果此 ID 將使用 IBM WebSphere MQ 管理指令，則在 Windows 系統上，管理使用者必須同時隸屬於 mqm 群組和 administrators 群組。

Windows 系統上的使用者 ID

在 Windows 系統上， 如果未安裝訊息結束程式，則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*

Windows 以外的平台， UNIX and Linux 系統會在訊息中對使用者 ID 使用大寫字元。

To allow Windows, UNIX and Linux systems to use lowercase user IDs in messages, the following conversions are carried out by the message channel agent (MCA) on these platforms:

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 UNIX、 Linux 及 Windows 系統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 WebSphere MQ 提供可安裝的授權服務。 您可以選擇安裝替代服務。

IBM WebSphere 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 WebSphere 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 *7:* 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 48 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	在任何其他方法未設定已使用的 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 48 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 WebSphere MQ MQI 用戶端，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。

在 Windows 平台上，請確定 MCAUSER 欄位限制為 12 個字元，因為任何額外字元都會被截斷，而可能導致授權失敗。

當使用用戶端安全結束程式時， IBM WebSphere 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:

- 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 WebSphere 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 WebSphere 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。

對於 UNIX、 Linux 及 Windows 系統上的伺服器，從用戶端收到的 `MCAUserIdentifier` 欄位內容會變更為小寫。

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。

對於 UNIX 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，當使用 MQ JMS 連結應用程式時所傳遞的使用者 ID，是執行應用程式之 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 SSL 或 TLS，在此情況下，您必須計劃使用數位憑證。 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 49 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 53 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 58 頁的『使用 SSL 保護通道』
IBM WebSphere MQ 中的 SSL 支援會使用佇列管理程式鑑別資訊物件，以及各種 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 49 頁的圖 8 中說明。

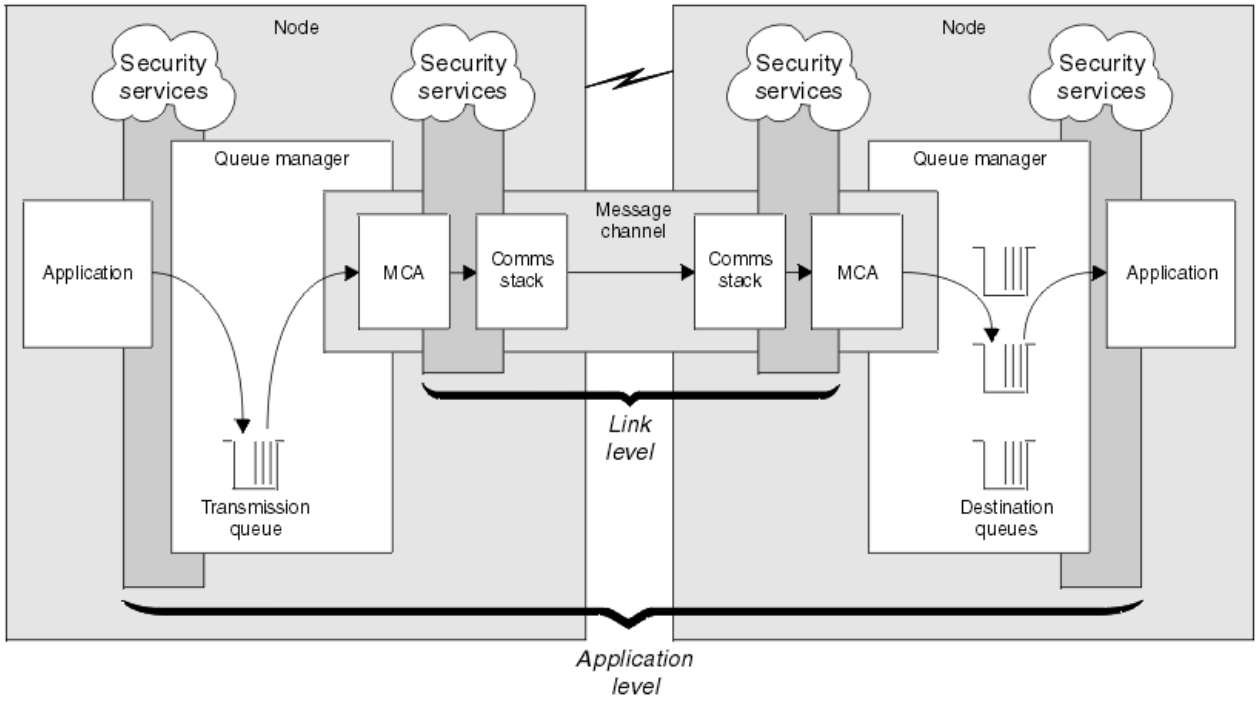


圖 *8:* 鏈結層次安全及應用程式層次安全

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。

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 49 頁的圖 8 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， WebSphere MQ 所提供的存取控制機制可能被視為足以保護儲存在其佇列上的訊息。 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 WebSphere 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 49 頁的圖 8 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 WebSphere MQ 提供的鏈結層次安全

IBM WebSphere MQ 中提供機密性和資料完整性的主要方法是使用 SSL 或 TLS。 如需在 IBM WebSphere MQ 中使用 SSL 和 TLS 的相關資訊，請參閱 第 20 頁的『IBM WebSphere MQ 支援 SSL 和 TLS』。 對於鑑別， IBM WebSphere 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 33 頁的『通道鑑別記錄』。

提供您自己的鏈結層次安全

這個主題集合說明如何提供您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 53 頁的『通道結束程式』中引進。 同一主題也說明 IBM WebSphere 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、另一個支援 WebSphere MQ 的產品，或任何這些一起運作的組合，都可以直接或間接呼叫服務。 第 49 頁的圖 8 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*

IBM WebSphere MQ Advanced Message Security (AMS) 是 IBM WebSphere MQ 的個別授權元件，可為流經 IBM WebSphere MQ 網路的機密資料提供高階保護，同時不會影響終端應用程式。

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 WebSphere 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。 IBM WebSphere MQ 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。

IBM WebSphere MQ Advanced Message Security 以下列方式延伸 IBM WebSphere 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 WebSphere MQ 伺服器和用戶端。
- 它與 IBM WebSphere MQ 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 227 頁的『IBM WebSphere MQ Advanced Message Security』。

提供您自己的應用程式層次安全

這個主題集合說明如何提供您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 WebSphere MQ 提供兩個結束程式 :API 結束程式和 API 交互結束程式。

這些結束程式可以提供識別及鑑別、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 WebSphere 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 WebSphere 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

第 53 頁的圖 9 中說明這四種類型的通道結束程式，並在下列主題中說明。

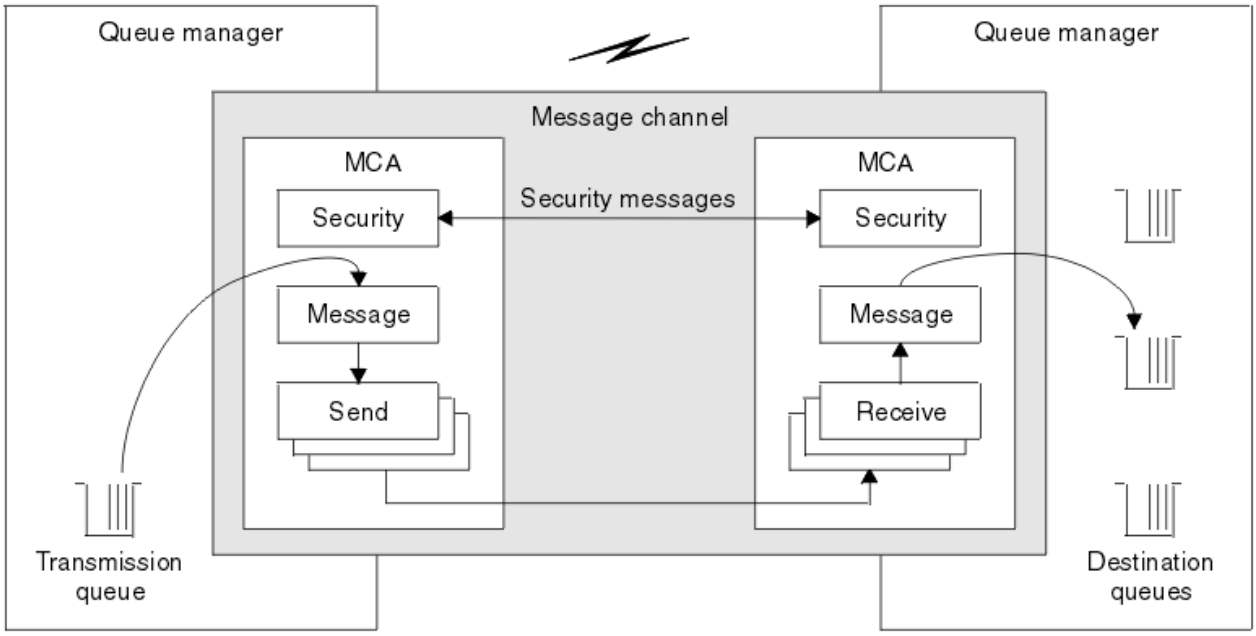


圖 *9:* 訊息通道上的安全、訊息、傳送及接收結束程式

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 51 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 通道的目的是啟用 MQI 呼叫的輸入及輸出參數，以在 IBM WebSphere MQ MQI 用戶端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 51 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 WebSphere 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 WebSphere 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 51 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，您可以選擇使用 IBM WebSphere MQ Advanced Message Security 來數位簽署訊息，以防止未獲授權的修改。 如果標準機能無法滿足您的需求，您也可以使用 API 結束程式。

在鏈結層次，您可以選擇使用 SSL 或 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 58 頁的『使用 SSL 保護通道』
IBM WebSphere MQ 中的 SSL 支援會使用佇列管理程式鑑別資訊物件，以及各種 MQSC 指令。 您也必須考量使用數位憑證。

第 19 頁的『IBM WebSphere 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 52 頁的『規劃 Advanced Message Security』
IBM WebSphere MQ Advanced Message Security (AMS) 是 IBM WebSphere MQ 的個別授權元件，可為流經 IBM WebSphere MQ 網路的機密資料提供高階保護，同時不會影響終端應用程式。

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 WebSphere 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 WebSphere MQ 介面對 IBM WebSphere MQ 所做的變更
配置 IBM WebSphere 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 WebSphere MQ 所做的變更
部分變更可能會影響 IBM WebSphere MQ 的行為方式，但無法由 IBM WebSphere MQ 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 WebSphere 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 WebSphere MQ 可以配置成發出檢測事件。

IBM WebSphere 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及取得訊息)，請配置 IBM WebSphere 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 WebSphere 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 WebSphere 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，您需要具有各種 IBM WebSphere MQ 資源存取權的使用者 ID。

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 WebSphere MQ 中，通道可以受 SSL 或 TLS 支援保護。

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。 如需進一步資訊，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。
- 使用與接收端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「 WebSphere 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，它將具有完整管理權限且不需要任何特殊授權

    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。
  – 如果您使用缺少 WebSphere 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。 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。 例如，發出下列指令:

```
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 33 頁的『通道鑑別記錄』
若要在通道層次對授與用來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。

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 WebSphere MQ 通道起始程式不是 IBM WebSphere MQ 物件; 對它們的存取權不受 OAM 控制。 IBM WebSphere 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 74 頁的表 10。

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 53 頁的『通道結束程式』。

使用 *SSL* 保護通道

IBM WebSphere MQ 中的 SSL 支援會使用佇列管理程式鑑別資訊物件，以及各種 MQSC 指令。 您也必須考量使用數位憑證。

SSL 支援的指令及屬性

Secure Sockets Layer (SSL) 通訊協定提供通道安全，防止竊聽、竄改及模擬。 IBM WebSphere MQ SSL 支援可讓您在通道定義上指定特定通道使用 SSL 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 SSL:

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 SSL:

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/SSL 憑證檢查。

SSLCRYP
在 Windows、 UNIX and Linux 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 SSL 的通道無法建立 SSL 連線，是否報告 SSL 事件訊息。

SSLFIPS
指定在 IBM WebSphere 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 Windows(UNIX and Linux 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。 (IBM Global Security Kit (GSKit) 可讓您在 Windows、UNIX and Linux 系統上使用 SSL 安全保護。)

SSLRKEYC
在重新協議秘密金鑰之前，要在 SSL 交談內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 SSL:

SSLCAUTH
定義 IBM WebSphere MQ 是否需要並驗證來自 SSL 用戶端的憑證。

SSLCIPH
指定加密強度和功能 (CipherSpec) ，例如 NULL_MD5 或 RC4_MD5_US。 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。 它也說明用於在 UNIX and Linux 系統上管理憑證的 iKeycmd 指令，以及用於在 UNIX、 Linux 及 Windows 系統上管理憑證的 runmqakm 工具。 請參閱下列小節:

- setmqaut
- dspmqaut

- dmpmqaut
- rcrmqobj
- rcdmqimg
- dspmqfls
- 管理金鑰和憑證

如需使用 SSL 的通道安全概觀，請參閱

- 第 20 頁的『IBM WebSphere MQ 支援 SSL 和 TLS』

如需與 SSL 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 SSL 相關聯的 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

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:

Windows UNIX Linux 在 Windows、 UNIX 及 Linux 系統上，您可以使用自簽憑證。 請參閱第 102 頁的『在 UNIX, Linux, and Windows 系統上建立自簽個人憑證』以取得相關指示。

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 29 頁的『IBM WebSphere 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 WebSphere MQ 中憑證驗證原則的相關資訊，請參閱 第 28 頁的『IBM WebSphere MQ 中的憑證驗證原則』。

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， WebSphere 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。 如需可在您的平台上使用的更強加密形式 (例如 z/OS 上的三重 DES 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。

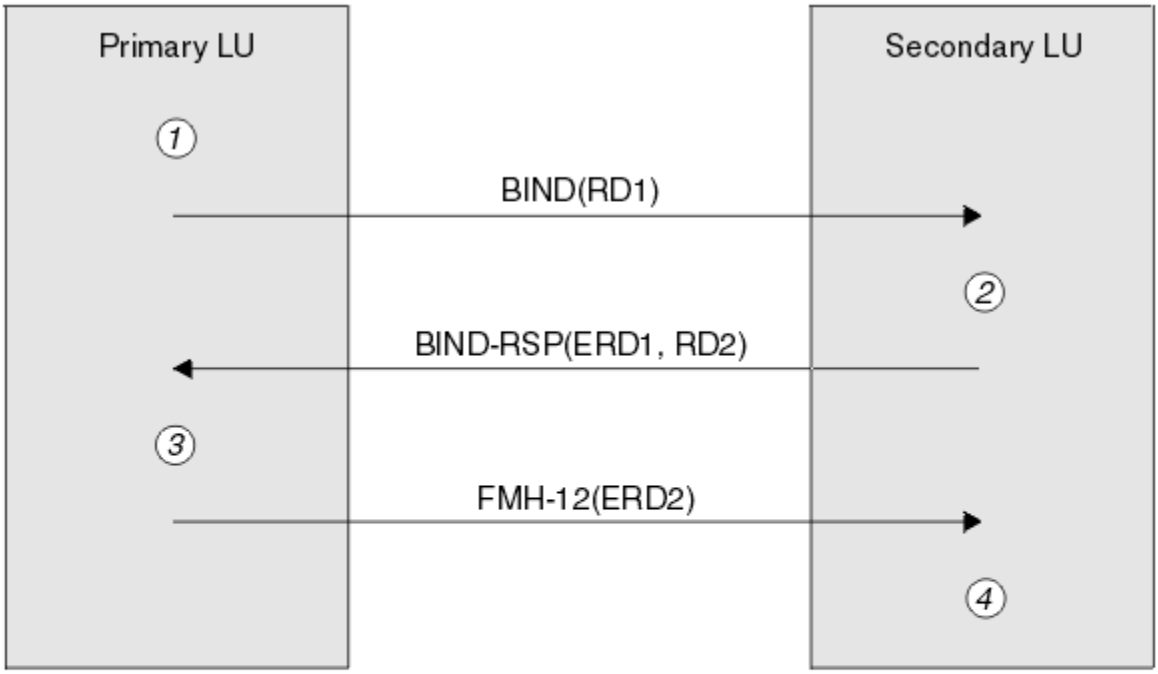
階段作業層次鑑別

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 61 頁的圖 10 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 *10:* 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 61 頁的圖 10 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 WebSphere 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)。

在 *UNIX* 系統及 *Windows* 系統上，支援 *IBM WebSphere MQ* 中的交談層次鑑別
在 UNIX, Linux, and Windows 上，請利用這個主題來取得交談層次鑑別如何運作的概觀。

第 62 頁的圖 11 說明 IBM WebSphere MQ for WebSphere MQ (在 UNIX 系統上) 及 WebSphere MQ for Windows 中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。

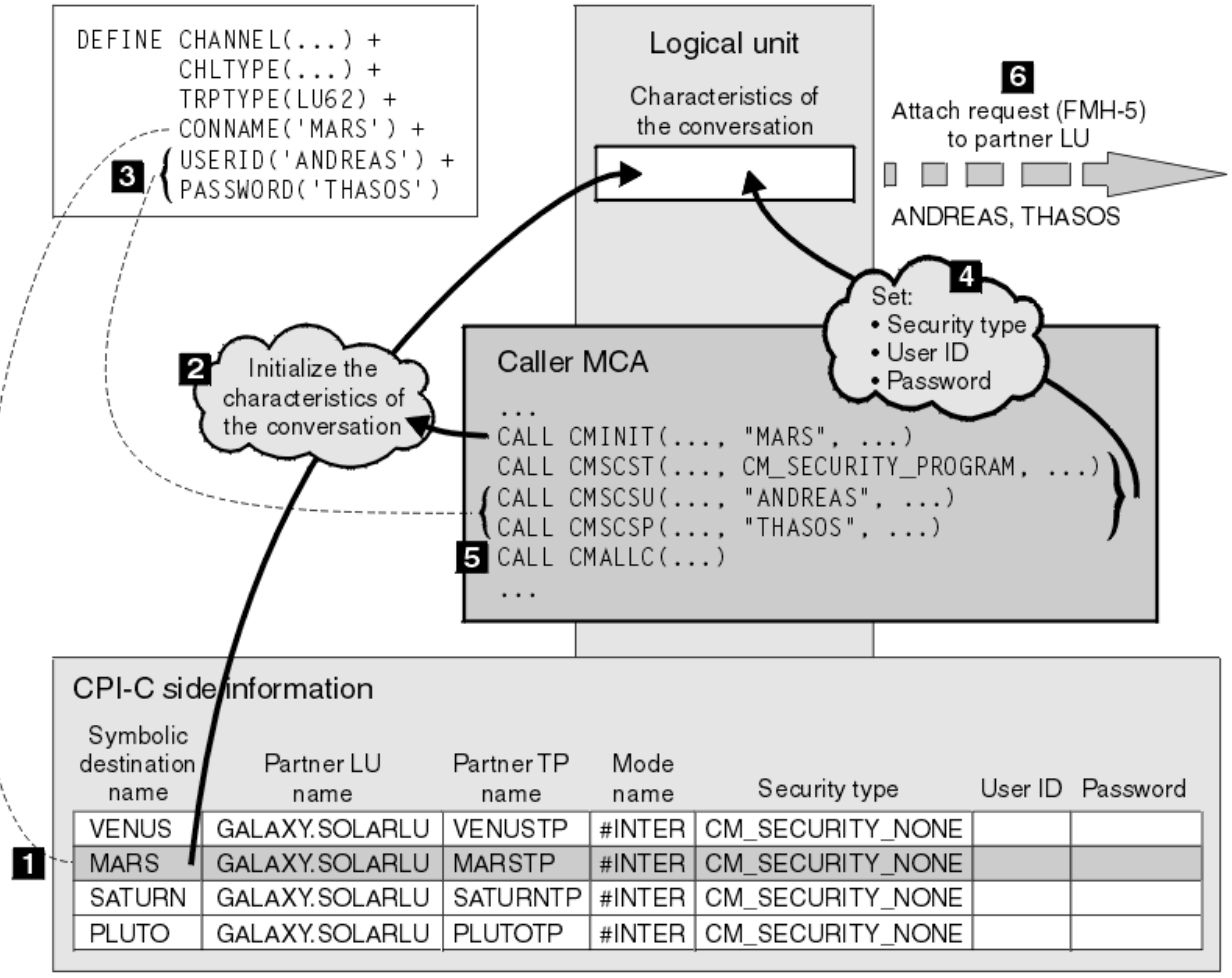


圖 *11: WebSphere MQ* 支援交談層次鑑別

在 IBM i、 UNIX 系統及 Windows 系統上， MCA 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。

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 WebSphere 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， WebSphere 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， WebSphere 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。

相關工作

第 209 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 WebSphere MQ 發佈/訂閱的安全

如果您使用「 IBM WebSphere 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 WebSphere 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 WebSphere 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 WebSphere MQ Multicast 可能需要安全處理程序的原因。

IBM WebSphere 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 WebSphere MQ 應用程式 (例如， LLM 應用程式，如需相關資訊，請參閱 與 WebSphere 的多重播送交互作業能力 MQ 低延遲傳訊)，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 WebSphere 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 47 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 6 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 WebSphere MQ 內的 IP 位址，可能是在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 152 頁的『封鎖特定 IP 位址』。

IBM WebSphere MQ 網際網路透通的安全

網際網路透通可以簡化透過防火牆的通訊，但這會影響安全。

IBM WebSphere MQ 網際網路透通是 SupportPac MS81 中提供的 IBM WebSphere MQ 基本產品延伸。

WebSphere MQ 網際網路透通可讓兩個佇列管理程式交換訊息，或讓 WebSphere 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。 它可讓 WebSphere MQ 通道通訊協定流入及流出防火牆更簡單且更容易管理，方法是透過在 HTTP 內進行通道傳輸或作為 Proxy。 使用 Secure Sockets Layer (SSL) ，它也可以用來加密及解密透過網際網路傳送的訊息。

當 WebSphere MQ 系統與 IPT 通訊時，除非您在 IPT 中使用 SSLProxyMode ，否則請確定 WebSphere MQ 所使用的 CipherSpec 符合 IPT 所使用的 CipherSuite :

- 當 IPT 充當 SSL 或 TLS 伺服器，且 WebSphere MQ 連接作為 SSL 或 TLS 用戶端時， WebSphere MQ 使用的 CipherSpec 必須對應於相關 IPT 金鑰環中啟用的 CipherSuite 。
- 當 IPT 充當 SSL 或 TLS 用戶端並連接至 WebSphere MQ SSL 或 TLS 伺服器時， IPT CipherSuite 必須符合接收端 WebSphere MQ 通道上定義的 CipherSpec 。

如果您從 IPT 移轉至整合 WebSphere MQ SSL 及 TLS 支援，請使用 iKeyman 從 IPT 傳送數位憑證。

如需相關資訊，請參閱 WebSphere MQ Internet Pass-Thru (SupportPac MS81)。

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。

在 UNIX, Linux, and Windows 系統上設定安全

UNIX, Linux, and Windows 系統特定的安全考量。

IBM WebSphere 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 WebSphere MQ
您可以定義一組可以發出指令來管理 IBM WebSphere MQ 的使用者。

誰可以使用 IBM WebSphere 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 WebSphere 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 WebSphere 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。

使用終端機服務連接至 IBM WebSphere MQ

如果您使用「終端機服務」，**Create global objects** 使用者權限可能會導致問題。

如果您使用「終端機服務」連接至 Windows 系統，且在建立或啟動佇列管理程式時發生問題，這可能是因為使用者權限 **Create global objects** 在 Windows 的最新版本中。

Create global objects 使用者權限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 **Create global objects** 使用者權限。

依預設，管理者會套用 **Create global objects** 使用者權限，因此在使用「終端機服務」連接時，管理者可以建立及啟動佇列管理程式，而不會變更使用者權限。

當您使用終端機服務時，如果管理 WebSphere MQ 的各種方法都沒有作用，請嘗試設定 **Create global objects** 使用者權限:

1. 開啟「系統管理工具」畫面:

   Windows 2003 和 Windows XP
   使用 **控制台** > **系統管理工具**存取此畫面。

   Windows Vista 和 Windows Server 2008
   使用 **控制台** > **系統和維護** > **系統管理工具**來存取此畫面。
2. 按兩下**本機安全性原則**。
3. 展開 `Local Policies`。
4. 按一下 `User Rights Assignment`。
5. 將新的使用者或群組新增至 **Create global objects** 原則。

在 Windows 上建立及管理群組

這些指示會引導您完成在工作站或成員伺服器機器上管理群組的程序。

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「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不會生效。

Windows 2003 和 Windows XP
使用 **控制台** > **系統管理工具** > **電腦管理**來存取此畫面。

Windows Vista 和 Windows Server 2008
使用 **控制台** > **系統及維護** > **系統管理工具** > **電腦管理**來存取此畫面。

Windows 7
使用 **系統管理工具** > **電腦管理** 存取此畫面

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**。

在 *Windows* 上將使用者新增至群組

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。

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。

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。

在 HP-UX 上建立及管理群組

在 HP-UX 上，如果您不是使用 NIS 或 NIS + ，請使用「系統管理程式 (SAM)」來使用群組。

在 *HP-UX* 上建立群組

使用「系統管理管理程式」將使用者新增至群組

程序

1. 從「系統管理管理程式 (SAM)」中，按兩下使用者和群組的帳戶。

2. 按兩下群組。
3. 從「動作」下拉清單中選取「新增」，以顯示「新增群組」畫面。
4. 輸入群組名稱，並選取您要新增至群組的使用者。
5. 按一下套用以建立群組。

結果
您現在已建立群組。

在 *HP-UX* 上新增使用者至群組
使用「系統管理管理程式」將使用者新增至群組。

程序
1. 從「系統管理管理程式 (SAM)」中，按兩下使用者和群組的帳戶。
2. 按兩下群組。
3. 強調顯示群組名稱，並從「動作」下拉清單中選取「修改」，以顯示「修改現有群組」畫面。
4. 選取您要新增至群組的使用者，然後按一下新增。
5. 如果您要將其他使用者新增至群組，請針對每一個使用者重複步驟 4。
6. 當您完成將名稱新增至清單時，請按一下確定。

結果
您現在已將使用者新增至群組。

在 *HP-UX* 上顯示群組中的人員
使用「系統管理管理程式」顯示群組中的人員

程序
1. 從「系統管理管理程式 (SAM)」中，按兩下使用者和群組的帳戶。
2. 按兩下群組。
3. 強調顯示群組名稱，並從「動作」下拉清單中選取「修改」，以顯示「修改現有群組」畫面，其中顯示群組中的使用者清單。

結果
即會顯示群組成員。

在 *HP-UX* 上從群組中移除使用者
使用「系統管理管理程式」從群組中移除使用者。

程序
1. 從「系統管理管理程式 (SAM)」中，按兩下使用者和群組的帳戶。
2. 按兩下群組。
3. 強調顯示群組名稱，並從「動作」下拉清單中選取「修改」，以顯示「修改現有群組」畫面。
4. 選取您要從群組中移除的使用者，然後按一下移除。
5. 如果您要從群組中移除其他使用者，請針對每一個使用者重複步驟 4。
6. 當您完成從清單中移除名稱時，請按一下確定。

結果
您現在已從群組中移除使用者

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 來處理群組。

建立群組

使用 SMITTY 建立群組。

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. 選取新增群組，然後按 Enter 鍵。
4. 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)。
5. 按 Enter 鍵以建立群組。

結果

您現在已建立群組。

將使用者新增至群組

使用 SMITTY 將使用者新增至群組。

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. 選取變更/顯示群組性質，然後按 Enter 鍵。
4. 輸入群組名稱，以顯示群組成員的清單。
5. 新增您要新增至群組的使用者名稱，以逗點區隔。
6. 按 Enter 鍵將名稱新增至群組。

顯示群組中的人員

顯示使用 SMITTY 的群組中的人員。

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. 選取變更/顯示群組性質，然後按 Enter 鍵。
4. 輸入群組名稱，以顯示群組成員的清單。

結果

即會顯示群組成員。

從群組中移除使用者

使用 SMITTY 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. 選取變更/顯示群組性質，然後按 Enter 鍵。
4. 輸入群組名稱，以顯示群組成員的清單。
5. 刪除您要從群組中移除的使用者名稱。

6. 按 Enter 鍵以從群組中移除名稱。

結果
您現在已從群組中移除使用者。

在 Solaris 上建立及管理群組

在 Solaris 上，如果您不是使用 NIS 或 NIS + ，請使用 /etc/group 檔案來使用群組。

在 *Solaris* 上建立群組
使用 **groupadd** 指令來建立群組。

程序

請鍵入下列指令： groupadd *group-name*
其中 *group-name* 是群組的名稱。

結果
檔案 /etc/group 檔案保留群組資訊。

在 *Solaris* 上將使用者新增至群組
使用 **usermod** 指令將使用者新增至群組。

程序

若要將成員新增至增補群組，請執行 usermod 指令，並列出使用者目前是其成員的增補群組，以及使用者要成為其成員的增補群組。
例如，如果使用者是群組 groupa 的成員，而且也要成為 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。

在 *Solaris* 上顯示群組中的人員
若要探索誰是群組成員，請在 /etc/group 檔案中查看該群組的項目。

在 *Solaris* 上從群組中移除使用者
使用 **usermod** 指令從群組中移除使用者。

程序

若要從增補群組中移除成員，請執行 **usermod** 指令，列出您希望使用者保留其成員的增補群組。
比方說，如果使用者的主要群組是 users ，且使用者也是群組 mqm、 groupa 和 groupb 的成員，則為了從 mqm 群組中移除使用者，會使用下列指令: usermod -G groupa,groupb *user-name*，其中 *user-name* 是使用者名稱。

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。

在 *Linux* 上建立群組
使用 **groupadd** 指令來建立群組。

程序

若要建立新群組，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。

結果
檔案 /etc/group 檔案保留群組資訊。

將使用者新增至 *Linux* 上的群組
使用 **usermod** 指令將使用者新增至群組。

程序

若要將成員新增至增補群組，請執行 usermod 指令，並列出使用者目前是其成員的增補群組，以及使用者要成為其成員的增補群組。
例如，如果使用者是群組 groupa 的成員，而且要同時成為 groupb 的成員，則會使用下列指令: `usermod -G groupa,groupb` *`user-name`*
，其中 *user-name* 是使用者名稱。

在 *Linux* 上顯示群組中的人員
使用 **getent** 指令顯示群組中的人員。

程序

若要顯示誰是群組成員，請鍵入下列指令: `getent group` *`group-name`*
，其中 *group-name* 是群組的名稱。

從群組中移除使用者
使用 **usermod** 指令從群組中移除使用者。

程序

若要從增補群組中移除成員，請執行 **usermod** 指令，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，則為了從 mqm 群組中移除使用者，會使用下列指令: `usermod -G groupa,groupb` *`user-name`*
，其中 *user-name* 是使用者名稱。

授權如何運作
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之 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。

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，而不是對已解析名稱之後所產生的名稱進行權限檢查。 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 73 頁的表 8、 第 73 頁的表 9、 第 74 頁的表 10 和 第 74 頁的表 11 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 *8: MQCONN* 呼叫所需的安全授權

需要授權:	**佇列物件 (第 75 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 *9: MQOPEN* 呼叫所需的安全授權

需要授權:	**佇列物件 (第 75 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 75 頁的『2』)**	MQZAO_ 輸入	不適用	不適用
MQOO_OUTPUT (一般佇列) **(第 75 頁的『3』)**	MQZAO_OUTPUT	不適用	不適用
MQOO_PASS_ IDENTITY_CONTEXT **(第 75 頁的『4』)**	MQZAO_PASS_ IDENTITY_CONTEXT	不適用	不檢查
MQOO_PASS_ALL_ CONTEXT **(第 75 頁的『4』， 第 75 頁的『5』)**	MQZAO_PASS _ALL_CONTEXT	不適用	不檢查
MQOO_SET_ IDENTITY_CONTEXT **(第 75 頁的『4』， 第 75 頁的『5』)**	MQZAO_SET_ IDENTITY_CONTEXT	不適用	MQZAO_SET_ IDENTITY_CONTEXT **(第 75 頁的『6』)**

表 *9: MQOPEN* 呼叫所需的安全授權 *(繼續)*			
需要授權:	**佇列物件 (第 75 頁的『1』)**	**程序物件**	**佇列管理程式物件**
MQOO_SET_ ALL_CONTEXT **(第 75 頁的『4』, 第 75 頁的『7』)**	MQZAO_SET_ ALL_CONTEXT	不適用	MQZAO_SET_ ALL_CONTEXT **(第 75 頁的『6』)**
MQOO_OUTPUT (傳輸佇列) **(第 75 頁的『8』)**	MQZAO_SET_ ALL_CONTEXT	不適用	MQZAO_SET_ ALL_CONTEXT **(第 75 頁的『6』)**
MQOO_SET	MQZAO_SET	不適用	不檢查
MQOO_ALTERNATE_USER _AUTHORITY	**(第 75 頁的『9』)**	**(第 75 頁的『9』)**	MQZAO_ALTERNATE_USE R_AUTHORITY **(第 75 頁的『9』, 第 75 頁的『10』)**

表 *10: MQPUT1* 呼叫所需的安全授權			
需要授權:	**佇列物件 (第 75 頁的『1』)**	**程序物件**	**佇列管理程式物件**
MQPMO_PASS_ IDENTITY_CONTEXT	MQZAO_PASS_ IDENTITY_CONTEXT **(第 75 頁的『11』)**	不適用	不檢查
MQPMO_PASS_ALL _CONTEXT	MQZAO_PASS_ ALL_CONTEXT **(第 75 頁的『11』)**	不適用	不檢查
MQPMO_SET_ IDENTITY_CONTEXT	MQZAO_SET_ IDENTITY_CONTEXT **(第 75 頁的『11』)**	不適用	MQZAO_SET_ IDENTITY_CONTEXT **(第 75 頁的『6』)**
MQPMO_SET_ ALL_CONTEXT	MQZAO_SET_ ALL_CONTEXT **(第 75 頁的『11』)**	不適用	MQZAO_SET_ ALL_CONTEXT **(第 75 頁的『6』)**
(傳輸佇列) **(第 75 頁的『8』)**	MQZAO_SET_ ALL_CONTEXT	不適用	MQZAO_SET_ ALL_CONTEXT **(第 75 頁的『6』)**
MQPMO_ALTERNATE_USE R_AUTHORITY	**(第 75 頁的『12』)**	不適用	MQZAO_ALTERNATE_USE R_AUTHORITY **(第 75 頁的『10』)**

表 *11: MQCLOSE* 呼叫所需的安全授權			
需要授權:	**佇列物件 (第 75 頁的『1』)**	**程序物件**	**佇列管理程式物件**
MQCO_DELETE	MQZAO_DELETE **(第 75 頁的『13』)**	不適用	不適用
MQCO_DELETE _PURGE	MQZAO_DELETE **(第 75 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 75 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. 只有在下列兩項都成立時，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。

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

物件	需要授權
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 80 頁的『1』)

物件	需要授權
佇列	MQZAO_CREATE (第 80 頁的『2』)
主題	MQZAO_CREATE (第 80 頁的『2』)
處理程序	MQZAO_CREATE (第 80 頁的『2』)
佇列管理程式	不適用
名稱清單	MQZAO_CREATE (第 80 頁的『2』)
鑑別資訊	MQZAO_CREATE (第 80 頁的『2』)
通道	MQZAO_CREATE (第 80 頁的『2』)
用戶端連線通道	MQZAO_CREATE (第 80 頁的『2』)
接聽器	MQZAO_CREATE (第 80 頁的『2』)
服務	MQZAO_CREATE (第 80 頁的『2』)
通訊資訊	MQZAO_CREATE (第 80 頁的『2』)

DEFINE 物件 REPLACE (第 80 頁的『1』, 第 80 頁的『3』)

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
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
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

物件	需要授權
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
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
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 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

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
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 86 頁的『2』)**

複製 物件 (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

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

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
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

物件	需要授權
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 物件 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
服務	不檢查

物件	需要授權
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
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
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	
重設叢集	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	
回復佇列管理程式叢集	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 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」。

Windows 上安全的特殊考量

部分安全功能在不同 Windows 版本上的行為不同。

IBM WebSphere 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。 這個主題集合包含當您在 Windows 環境中執行 IBM WebSphere MQ 時，這些差異如何影響 IBM WebSphere MQ 安全的說明。

SSPI 通道結束程式

WebSphere MQ for Windows 提供安全結束程式，可同時在訊息及 MQI 通道上使用。 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)」，它提供 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。 WebSphere MQ 7.0 所支援的所有 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 WebSphere MQ MQI 用戶端應用程式。 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。 WebSphere MQ 7.0 支援的所有 Windows 平台都支援 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 WebSphere MQ MQI 用戶端應用程式彼此鑑別。 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。

當您在 *Windows* 上取得「找不到群組」錯誤時

當 Windows 伺服器升級至網域控制站或從網域控制站降級時，因為 WebSphere MQ 無法存取本端 mqm 群組，所以可能會發生此問題。 若要補救此問題，請重建本端 mqm 群組。

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.
```

變更伺服器與網域控制站之間的機器狀態可能會影響 WebSphere MQ 的作業，因為 WebSphere MQ 使用本端定義的 mqm 群組。 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。

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，您必須在新建立的本端 mqm 群組中恢復特許 WebSphere 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 WebSphere MQ 使用者 ID 必要的權限層次。

當您在 *Windows* 上使用 *IBM WebSphere MQ* 及網域控制站時發生問題

當 Windows 伺服器升級至網域控制站時，安全設定可能會發生某些問題。

將 Windows 2000、 Windows 2003 或 Windows Server 2008 伺服器升級至網域控制站時，您會看到一個選項，可讓您選取與使用者和群組許可權相關的預設或非預設安全設定。 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。 因為 WebSphere MQ 依賴群組成員資格資訊來實作其安全原則，所以執行 WebSphere MQ 作業的使用者 ID 必須能夠決定其他使用者的群組成員資格。

在 Windows 2000 上，使用預設安全選項建立網域時， WebSphere 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。 然後產品會正常安裝，建立預設物件，必要的話，佇列管理程式可以決定本端及網域使用者的存取權限。

在 Windows 2000 上，當使用非預設安全選項建立網域時，或在 Windows 2003 及 Windows Server 2008 上使用預設安全選項建立網域時，安裝期間 WebSphere MQ 所建立的使用者 ID 無法一律判定所需的群組成員資格。 在此情況下，您需要知道:

- Windows 2000 (含非預設) 或 Windows 2003 及 Windows Server 2008 (含預設安全許可權) 的行為
- 如何容許網域 mqm 群組成員讀取群組成員資格
- 如何配置 IBM WebSphere MQ Windows 服務在網域使用者下執行

Windows 2000 網域 *(*含非預設*)* 或 *Windows 2003* 及 *Windows Server 2008* 網域 *(*含預設安全許可權*)*
在這些作業系統上安裝 WebSphere MQ 時的行為，會因執行安裝的是本端使用者還是網域使用者而異。

如果 **本端** 使用者安裝 WebSphere MQ，「準備 WebSphere MQ 精靈」會偵測到為 IBM WebSphere 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。 「準備 WebSphere MQ 精靈」會詢問使用者有關網路配置的問題，以判斷在 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 WebSphere 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。 「準備 WebSphere MQ 精靈」會提示使用者輸入此使用者的帳戶詳細資料。 其線上說明提供可傳送至網域管理者所需網域使用者帳戶的詳細資料。

如果 **網域** 使用者安裝 WebSphere MQ，「準備 WebSphere MQ 精靈」會偵測到為 IBM WebSphere 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，「準備

WebSphere MQ 精靈」一律會提示使用者輸入網域使用者帳戶的帳戶詳細資料，以供 IBM WebSphere MQ Windows 服務使用。

當 IBM WebSphere MQ Windows 服務需要使用網域使用者帳戶時，在使用「準備 WebSphere MQ 精靈」配置之前， WebSphere MQ 無法正確運作。 在使用適當的帳戶配置 Windows 服務之前，「準備 WebSphere MQ 精靈」不容許使用者繼續執行其他作業。

如果已使用非預設安全許可權來配置 Windows 2000 網域，則讓 WebSphere MQ 正常運作的一般解決方案是使用適當的網域使用者帳戶來配置它，如上所述。

如需相關資訊，請參閱 建立及設定 WebSphere MQ 的網域帳戶 。

配置 *IBM WebSphere MQ* 服務以在 *Windows* 上以網域使用者身分執行
使用「準備 IBM WebSphere MQ 」精靈來輸入網域使用者帳戶的帳戶詳細資料。 或者，您可以使用「電腦管理」畫面來變更安裝特定 IBM WebSphere MQ 服務的 **登入** 詳細資料。

如需相關資訊，請參閱 變更 IBM WebSphere MQ Windows 服務使用者帳戶的密碼

將安全範本檔案套用至 *Windows*

套用範本可能會影響套用至 WebSphere MQ 檔案和目錄的安全設定。 如果您使用高度安全的範本，請先套用它，再安裝 WebSphere MQ。

Windows 支援文字型安全範本檔，您可以使用「安全配置及分析」MMC 嵌入式管理單元，將統一安全設定套用至一部以上電腦。 特別是， Windows 提供數個範本，其中包括一系列安全設定，目的是提供特定安全層次。 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 WebSphere MQ 檔案和目錄的安全設定。 如果您想要使用「高度安全」範本，請先配置您的機器，然後再安裝 WebSphere MQ。

如果您將高度安全的範本套用至已安裝 WebSphere MQ 的機器，則會移除您在 WebSphere MQ 檔案和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)。

巢狀群組

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 WebSphere 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 伺服器 2003、Windows XP、Windows Vista 及 Windows Server 2008 都遵循 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 WebSphere MQ 所使用的 ID 施加進一步限制。 WebSphere 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， WebSphere 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，該區域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。

為連接至 *IBM WebSphere MQ* 的 *Windows* 應用程式配置其他權限

執行 IBM WebSphere MQ 處理程序的帳戶可能需要其他授權，才能授與對應用程式程序的 SYNCHRONIZE 存取權。

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 WebSphere MQ ，則可能會遇到問題。

IBM WebSphere MQ 需要應用程式程序的「同步化」存取權，才能協調特定動作。 APAR IC35116 已變更 IBM WebSphere MQ ，以便指定適當的專用權。 不過，執行 IBM WebSphere 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。

當伺服器應用程式第一次嘗試連接至佇列管理程式時， IBM WebSphere MQ 會修改程序，將 SYNCHRONIZE 權限授與 IBM WebSphere MQ 管理者。 若要配置對執行 IBM WebSphere MQ 處理程序之使用者 ID 的其他權限，請完成下列步驟:

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 IBM WebSphere 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。

在 HP Integrity NonStop Server 上設定安全

HP Integrity NonStop Server 系統特有的安全考量。

HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端同時支援傳輸層安全 (TLS) 及 Secure Sockets Layer (SSL) 通訊協定，以在您連接至佇列管理程式時提供鏈結層次安全。 使用 OpenSSL 實作來支援這些通訊協定。 OpenSSL 需要隨機資料來源，以提供高度加密作業。

OpenSSL

IBM WebSphere MQ Client for HP Integrity NonStop Server 的 OpenSSL 安全概觀。

OpenSSL 工具箱是 Secure Sockets Layer (SSL) 和傳輸層安全 (TLS) 通訊協定的開放程式碼實作，用於透過網路進行安全通訊。

工具箱由 OpenSSL 專案開發。 如需 OpenSSL 專案的相關資訊，請參閱 https://www.openssl.org。 HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端包含已修改版本的 OpenSSL 程式庫及 **openssl** 指令。 這些程式庫和 **openssl** 指令是從 OpenSSL 工具箱 1.0.1c 移轉而來，並且僅作為物件程式碼提供。 未提供原始碼。

IBM WebSphere MQ 用戶端應用程式會視需要動態載入 OpenSSL 程式庫。 僅支援 IBM WebSphere MQ 提供的 OpenSSL 程式庫與 IBM WebSphere MQ 用戶端應用程式搭配使用。

openssl 指令 (可用於憑證管理) 安裝在 OSS 目錄 `opt_installation_path/opt/mqm/bin` 中。

使用 **openssl** 指令，您可以建立及管理具有各種一般資料格式的金鑰及數位憑證，並執行簡式憑證管理中心 (CA) 作業。

OpenSSL 所處理之金鑰及憑證資料的預設格式為 Privacy Enhanced Mail (PEM) 格式。 PEM 格式的資料是 base64 編碼的 ASCII 資料。 因此，可以使用文字型系統 (例如電子郵件) 來傳送資料，並且可以使用文字編輯器和 Web 瀏覽器來剪下和貼上資料。 PEM 是一種網際網路標準，用於文字型加密交換，並在網際網路 RFC 1421、1422、1423 及 1424 中指定。 IBM WebSphere MQ 假設副檔名為 `.pem` 的檔案包含 PEM 格式的資料。 PEM 格式的檔案可以包含多個憑證及其他編碼物件，且可以包含註解。

其他作業系統上的 IBM WebSphere MQ SSL 支援可能需要使用「識別編碼規則 (DER)」來編碼檔案中的金鑰和憑證資料。 DER 是一組在安全通訊中使用 ASN.1 表示法的編碼規則。 使用 DER 編碼的資料是二進位資料，使用 DER 編碼的金鑰和憑證資料格式也稱為 PKCS#12 或 PFX。 包含此資料的檔案通常具有副檔名 `.p12` 或 `.pfx`。 **openssl** 指令可以在 PEM 與 PKCS#12 格式之間轉換。

Entropy 常駐程式

OpenSSL 需要隨機資料來源，以提供高度加密作業。 亂數產生是一種通常由作業系統或全系統常駐程序提供的功能。 HP Integrity NonStop Server 作業系統在作業系統內不提供此功能。

當您使用 IBM WebSphere MQ client for HP Integrity NonStop Server 隨附的 SSL 和 TLS 支援時，需要稱為 entropy 常駐程式的處理程序來提供隨機資料的來源。 當您啟動需要 SSL 或 TLS 的用戶端通道時，OpenSSL 預期熵常駐程式正在 `/etc/egd-pool` 的 OSS 檔案系統中的 Socket 上執行並提供其服務。

HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端未提供 entropy 常駐程式。 HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端已使用下列熵常駐程式進行測試:

- amqjkdm0 (如 IBM WebSphere MQ 5.3 伺服器所提供)
- `/usr/local/bin/prngd` (0.9.27 版，由 HP Integrity NonStop Server Open Source Technical Library 提供)

設定 IBM WebSphere MQ MQI 用戶端安全

您必須考量 IBM WebSphere MQ MQI 用戶端安全，讓用戶端應用程式不會對伺服器上的資源具有無限制存取權。

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。

  您應該在 SSL 或 TLS 內使用交互鑑別。 如需相關資訊，請參閱 第 92 頁的『使用 SSL 或 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 132 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 155 頁的『將用戶端主張的使用者 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 33 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。

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 SSL 或 TLS 通道必須以三種方式僅使用 FIPS 認證的 CipherSpecs，並依優先順序列出:

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 SSL 或 TLS 連線，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 SSL 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 SSL 或 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 WebSphere 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，而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。 WebSphere MQ 不保證會拒絕所有這類連線，您必須負責判斷 WebSphere MQ 配置是否符合 FIPS 標準。

相關概念

第 22 頁的『 UNIX、 Linux 及 Windows 的聯邦資訊存取安全標準 (FIPS)』
當 Windows、 UNIX and Linux 系統上的 SSL 或 TLS 通道需要加密法時， WebSphere MQ 會使用稱為 IBM Crypto for C (ICC) 的加密法套件。 在 Windows(UNIX and Linux 平台) 上， ICC 軟體已通過美國國家標準與技術機構層次 140-2 的「聯邦資訊存取安全標準 (FIPS) Cryptomodule 驗證程式」。

用戶端配置檔的 SSL 段落

相關參考

FipsRequired (MQLONG)

MQSSLFIPS

在 AIX 上使用多個 GSKit V8.0 安裝來執行 SSL 或 TLS 用戶端應用程式

在具有多個 GSKit V8.0 安裝的 AIX 系統上執行時， AIX 上的 SSL 或 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 V8.0 安裝的 AIX 系統上執行用戶端應用程式時，使用 SSL 或 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 WebSphere MQ 用戶端從兩個不同的 GSKit V8.0 安裝中載入一組混合的程式庫。 在 Db2 環境中執行 IBM WebSphere 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包含 IBM WebSphere MQ 媒體庫目錄，以便優先使用 IBM WebSphere 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 WebSphere MQ 環境)

在 UNIX, Linux, and Windows 系統上設定 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 UNIX、Linux 和 Windows 系統上，您可以使用自簽憑證來執行測試。

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 95 頁的『在 UNIX, Linux, and Windows 系統上使用 SSL 或 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 WebSphere MQ 實作， SSL 或 TLS 伺服器一律會從用戶端要求憑證。

在 UNIX、 Linux 及 Windows 系統上，只有在 SSL 或 TLS 用戶端具有以正確 IBM WebSphere MQ 格式標示的憑證時，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 WebSphere 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 WebSphere 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，請參閱第 172 頁的『使用 SSL 或 TLS 連接兩個佇列管理程式』。

使用 SSL 或 TLS

這些主題提供如何執行與搭配使用 SSL 或 TLS 與 IBM WebSphere 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 121 頁的『識別及鑑別使用者』
- 第 132 頁的『授權存取物件』
- 第 172 頁的『訊息機密性』
- 第 190 頁的『訊息的資料完整性』
- 第 206 頁的『保持叢集安全』

在 HP Integrity NonStop Server 上使用 SSL 或 TLS

說明 HP Integrity NonStop Server OpenSSL 安全實作的 IBM WebSphere MQ 用戶端，包括安全服務、元件、支援的通訊協定版本、支援的 CipherSpecs 及不受支援的安全功能。

IBM WebSphere MQ SSL & TLS support provides the following security services for client channels:

- 鑑別伺服器，以及選擇性地鑑別用戶端。
- 加密及解密流經通道的資料。
- 對流經通道的資料進行完整性檢查。

HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端所提供的 SSL 及 TLS 支援包含下列元件:

- OpenSSL 程式庫及 **openssl** 指令。
- IBM WebSphere MQ password stash 指令 **amqrsslc**。

IBM WebSphere MQ Client for HP Integrity NonStop Server 未提供 SSL 或 TLS 用戶端通道作業的下列必要元件:

- 熵常駐程式，用於提供 OpenSSL 加密法的隨機資料來源。

支援的通訊協定版本

HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端支援下列通訊協定版本:

- SSL 3.0
- TLS 1.0
- TLS 1.2

支援的 CipherSpecs

HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端支援下列 CipherSpecs 版本:

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

不受支援的安全功能

HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端目前不支援:

- PKCS#11 加密硬體支援
- LDAP 憑證撤銷清冊檢查
- OCSP 線上憑證狀態通訊協定檢查

- FIPS 140-2， NSA SUITE B 密碼組合控制項

憑證管理

使用一組檔案來儲存數位憑證及憑證撤銷資訊。

IBM WebSphere MQ SSL 和 TLS 支援使用一組檔案來儲存數位憑證和憑證撤銷資訊。 這些檔案位於透過 MQCONNX 呼叫 MQSCO 結構中的 KeyRepository 欄位以程式化方式指定的目錄中，由 *MQSSLKEYR* 環境變數所傳遞，或位於使用 SSLKeyRepository 屬性之 `mqclient.ini` 的 SSL 段落中。

MQSCO 結構優先於 MQSSLKEYR 環境變數，後者優先於 ini 檔案段落值。

重要: 金鑰儲存庫位置指定目錄位置，而不是 HP Integrity NonStop Server 平台上的檔名。

HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端會在金鑰儲存庫位置中使用下列區分大小寫的具名檔案:

- 第 94 頁的『個人憑證儲存庫』
- 第 94 頁的『憑證信任儲存庫』
- 第 94 頁的『通行詞組隱藏檔』
- 第 95 頁的『憑證撤銷清冊檔案』

個人憑證儲存庫

個人憑證儲存庫檔案 `cert.pem`。

此檔案包含個人憑證及加密的私密金鑰，以供用戶端使用，採用 PEM 格式。 當您使用不需要用戶端鑑別的 SSL 或 TLS 通道時，此檔案的存在是選用的。 如果通道需要用戶端鑑別，且在通道定義上指定 SSLCAUTH (REQUIRED)，則此檔案必須存在且同時包含憑證及已加密私密金鑰。

必須對此檔案設定檔案許可權，以容許對憑證儲存庫擁有者的讀取權。

正確格式化的 `cert.pem` 檔案必須正好包含兩個具有下列標頭和標底的區段:

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

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

已加密私密金鑰的通行詞組儲存在通行詞組隱藏檔 `Stash.sth` 中。

憑證信任儲存庫

憑證信任儲存庫檔案 `trust.pem`。

此檔案包含驗證用戶端連接的佇列管理程式所使用的個人憑證所需的憑證 (PEM 格式)。 憑證信任儲存庫對於所有 SSL 或 TLS 用戶端通道都是必要的。

必須設定檔案許可權，以限制此檔案的寫入權。

正確格式化的 `trust.pem` 檔案必須包含一個以上具有下列標頭和標底的區段:

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

通行詞組隱藏檔

通行詞組隱藏檔 `Stash.sth`。

此檔案是 IBM WebSphere MQ 專用的二進位格式，包含已加密的通行詞組，可在您存取 `cert.pem` 檔案中保留的私密金鑰時使用。 私密金鑰本身儲存在 `cert.pem` 憑證儲存庫中。

此檔案是透過搭配使用 IBM WebSphere MQ **amqrsslc** 指令行工具與 **-s** 參數來建立或變更。 例如，其中 /home/alice 目錄包含 cert.pem 檔案:

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

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

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

必須對此檔案設定檔案許可權，以容許對相關聯個人憑證儲存庫擁有者的讀取權。

憑證撤銷清冊檔案
憑證撤銷清冊檔案 crl.pem。

此檔案包含用戶端用來驗證數位憑證的憑證撤銷清冊 (CRL) ，採用 PEM 格式。 此檔案的存在是選用的。 如果此檔案不存在，則在您驗證憑證時，不會執行任何憑證撤銷檢查。

必須設定檔案許可權，以限制此檔案的寫入權。

正確格式化的 crl.pem 檔案必須包含一個以上具有下列標頭和標底的區段:

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

在 UNIX, Linux, and Windows 系統上使用 SSL 或 TLS

在 UNIX、 Linux 和 Windows 系統上， Secure Sockets Layer (SSL) 支援隨 IBM WebSphere 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。

使用 *iKeyman*、 *iKeycmd*、*runmqakm* 和 *runmqckm*

在 UNIX、 Linux 及 Windows 系統上，使用 iKeyman GUI 或從指令行使用 iKeycmd 或 runmqakm 來管理金鑰及數位憑證。

- 若為 **UNIX and Linux** 系統:
  - 使用 **strmqikm** 指令來啟動 iKeyman GUI。
  - 使用 **runmqckm** 指令，以 iKeycmd 指令行介面來執行作業。
  - 使用 **runmqakm** 指令，以 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 SSL 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 。

  如果您要使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 iKeycmd 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 iKeycmd 程式在這些平台上是 32 位元。

  在下列平台上， JRE 在舊版產品中是 32 位元，但在 IBM WebSphere MQ Version 7.5 中是 64 位元，您可能需要安裝適用於 **iKeyman** 和 **iKeycmd** JRE 定址模式的其他 PKCS#11 驅動程式。 這是因為 PKCS#11 驅動程式必須使用與 JRE 相同的定址模式。 下表顯示 IBM WebSphere MQ Version 7.5 JRE 定址模式。

表 *12: IBM WebSphere MQ Version 7.5 JRE* 定址模式

平台	**JRE 定址模式**
Windows (32 位元或 64 位元)	32
Linux for System x（32 位元）	32
Linux for System x 64 位元	64

表 12: IBM WebSphere MQ Version 7.5 JRE 定址模式 (繼續)	
平台	**JRE 定址模式**
Linux for System p	64
Linux for System z	64
HP-UX	64
Solaris Sparc	64
Solaris x86-64	64
AIX	64

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
```

- 若為 **Windows** 系統:
  - 使用 **strmqikm** 指令來啟動 iKeyman GUI。
  - 使用 **runmqckm** 指令，以 iKeycmd 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 SSL 憑證，請使用 **runmqakm** 指令，而非 **runmqckm** 或 **strmqikm** 指令。

若要在 UNIX、 Linux 或 Windows 系統上要求 SSL 追蹤，請參閱 strmqtrc。

相關參考

runmqckm 及 runmqakm 指令

在 *UNIX, Linux, and Windows* 系統上設定金鑰儲存庫

您可以使用 iKeyman 使用者介面，或使用 **iKeycmd** 或 **runmqakm** 指令，來設定金鑰儲存庫。

關於這項作業

SSL 或 TLS 連線在連線的每一端都需要 金鑰儲存庫 。 每一個 IBM WebSphere MQ 佇列管理程式及 IBM WebSphere 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 20 頁的『SSL 或 TLS 金鑰儲存庫』。

在 UNIX, Linux, and Windows 系統上，數位憑證儲存在使用 **iKeyman** 使用者介面或使用 **iKeycmd** 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 WebSphere MQ MQI client 相關聯。 SSL 和 TLS 會使用該憑證進行鑑別。 在 UNIX, Linux, and Windows 系統上， IBM WebSphere MQ 使用 ibmwebspheremq 作為標籤字首，以避免與其他產品的憑證混淆。 字首後面接著佇列管理程式或 IBM WebSphere MQ MQI client 使用者登入 ID 的名稱，變更為小寫。請確保以小寫形式指定整個憑證標籤。

金鑰資料庫檔名包含路徑和系統名稱:

- 在 UNIX 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 WebSphere MQ 的安裝目錄。 例如， C:\program files\IBM\WebSphere 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 WebSphere 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 100 頁的『在 UNIX、 Linux 或 Windows 系統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 **iKeyman** 或 **iKeycmd** 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 **iKeyman** 或 **iKeycmd** 的使用者 ID 必須是 mqm 群組的成員。 對於 IBM WebSphere MQ MQI client，如果您執行 **iKeyman** 或 **iKeycmd** 的使用者 ID 不同於執行用戶端的使用者 ID，則必須變更檔案許可權，以讓 IBM WebSphere MQ MQI client 在執行時期存取金鑰資料庫檔案。 如需相關資訊，請參閱第 98 頁的『在 Windows 上存取及保護金鑰資料庫檔的安全』或第 99 頁的『在 UNIX and Linux 系統上存取金鑰資料庫檔案並保護其安全』。

在 **iKeyman** 或 **iKeycmd** 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 **iKeyman** 或 **iKeycmd** 8.0 版中，金鑰資料庫不會自動移入資料，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: 由於 GSKit 8.0 版的行為已變更，導致 CA 憑證不再自動新增至儲存庫，因此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 99 頁的『使用 GSKit 8.0 版將預設 CA 憑證新增至 UNIX, Linux, and Windows 系統上的空金鑰儲存庫』。

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:
   - 在 UNIX, Linux, and Windows 系統上:

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

   - 使用 runmqakm:

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

   其中:

   -db 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 WebSphere MQ，它必須是 cms。)

   -stash
   將金鑰資料庫密碼儲存至檔案。

-fips
停用使用 BSafe 加密程式庫。 只會使用 ICC 元件，且這個元件必須在 FIPS 模式中順利起始設定。 處於 FIPS 模式時， ICC 元件會使用 FIPS 140-2 驗證的演算法。 如果 ICC 元件未在 FIPS 模式中起始設定，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 UNIX 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. 執行 **strmqikm** 指令，以啟動 iKeyman 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 UNIX and Linux 系統上) 或 C:\Program Files\IBM\WebSphere 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 WebSphere MQ 用戶端: /var/mqm/ssl (在 UNIX and Linux 系統上) 或 C:\mqm\ssl (在 Windows 系統上)。
8. 按一下**開啟**。
   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: 如果您不隱藏密碼，則嘗試啟動 SSL 或 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 98 頁的『在 Windows 上存取及保護金鑰資料庫檔的安全』 或 第 99 頁的『在 UNIX and Linux 系統上存取金鑰資料庫檔案並保護其安全』中所述。

在 *Windows* 上存取及保護金鑰資料庫檔的安全

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。

在 *UNIX and Linux* 系統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。

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 -----)。

使用 *GSKit 8.0* 版將預設 *CA* 憑證新增至 *UNIX, Linux, and Windows* 系統上的空金鑰儲存庫

遵循此程序，將一或多個預設 CA 憑證新增至具有 GSKit 第 8 版的空金鑰儲存庫。

在 GSKit 7.0 版中，建立新的金鑰儲存庫時的行為是自動為常用憑證管理中心新增一組預設 CA 憑證。 對於 GSKit 第 8 版，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 iKeyman

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 iKeyman GUI (在 UNIX、 Linux 及 Windows 系統上)。
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 iKeycmd 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 WebSphere MQ 信任該 CA 憑證所簽署的所有個人憑證。 請仔細考量您想要信任哪些「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。

在 *UNIX, Linux, and Windows* 系統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 WebSphere MQ 探險家」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 UNIX and Linux 系統上: `/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 WebSphere MQ 安裝所在的高階目錄。

在 *UNIX*、 *Linux* 或 *Windows* 系統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 UNIX and Linux 系統上:

```
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\WebSphere MQ\Qmgrs\QM1\ssl\Mykey')
```

金鑰資料庫檔具有完整檔名: `C:\Program Files\IBM\WebSphere 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「WebSphere 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 112 頁的『將個人憑證匯入 UNIX, Linux, and Windows 系統上的金鑰儲存庫』中所述。

在 *UNIX, Linux, and Windows* 系統上尋找 *IBM WebSphere MQ MQI* 用戶端的金鑰儲存庫

金鑰儲存庫的位置由 MQSSLKEYR 變數提供，或在 MQCONNX 呼叫中指定。

請檢查 MQSSLKEYR 環境變數，以取得 IBM WebSphere MQ MQI 用戶端金鑰資料庫檔的位置。 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 101 頁的『指定 UNIX, Linux, and Windows 系統上 IBM WebSphere MQ MQI 用戶端的金鑰儲存庫位置』中所述。 MQCONNX 呼叫中設定的值會置換 MQSSLKEYR 的值。

指定 *UNIX, Linux, and Windows* 系統上 *IBM WebSphere MQ MQI* 用戶端的金鑰儲存庫位置

IBM WebSphere MQ MQI 用戶端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。

您可以使用下列兩種方式之一，來指定 IBM WebSphere MQ MQI 用戶端金鑰資料庫檔的位置：

- 設定 MQSSLKEYR 環境變數。 例如，在 UNIX and Linux 系統上：

  ```
  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\WebSphere MQ\ssl\key
  ```

  金鑰資料庫檔具有完整檔名：

  ```
  C:\Program Files\IBM\WebSphere 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 。

當憑證或憑證儲存庫的變更在 *UNIX*、 *Linux* 或 *Windows* 系統上生效時。

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 SSL 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 SSL 通道的要求時。
- 當發出 MQSC 指令 REFRESH SECURITY TYPE (SSL) 來重新整理 Websphere MQ SSL 環境時。
- 對於用戶端應用程式程序，當程序中的最後一個 SSL 連線關閉時。 下一個 SSL 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 SSL 通道時。 如果處理程序儲存區處理程序已執行 SSL 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 SSL 通道時。 如果通道起始程式處理程序已執行 SSL 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 SSL 通道的要求時。 如果接聽器已執行 SSL 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 WebSphere MQ 探險家」或 PCF 指令來重新整理 WebSphere MQ SSL 環境。

在 *UNIX, Linux, and Windows* 系統上建立自簽個人憑證

您可以使用 iKeyman、 iKeycmd 或 runmqakm 來建立自簽憑證。

註: IBM WebSphere 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 第 173 頁的『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。 WebSphere MQ 支援三種不同類型的 CipherSpec。 如需詳細資料，請參閱 第 29 頁的『IBM WebSphere MQ 中的數位憑證及 CipherSpec 相容性』 主題中的 第 30 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 。

使用 iKeyman

iKeyman 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 SSL 或 TLS 憑證，請使用 **runmqakm** 指令。

使用下列程序來取得佇列管理程式或 WebSphere MQ MQI 用戶端的自簽憑證:

1. 使用 **strmqikm** 指令啟動 iKeyman 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. 選取您要儲存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**。 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，鍵入:
   - 對於佇列管理程式， ibmwebspheremq 後面接著小寫的佇列管理程式名稱。 例如，若為 QM1、 ibmwebspheremqqm1 或
   - 若為 WebSphere MQ 用戶端， ibmwebspheremq 後面接著小寫的登入使用者 ID ， 例如 ibmwebspheremqmyuserid 。
10. 在 **Distinguished name** 或任何 **Subject alternative name** 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。

使用指令行

使用下列指令，以使用 iKeycmd 或 runmqakm 來建立自簽個人憑證:

- 在 UNIX、 Linux 及 Windows 系統上使用 iKeycmd :

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

您可以使用 -san_dsname *DNS_names* 、 -san_emailaddr *email_addresses* 或 -san_ipaddr *IP_addresses* 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
```

-db *filename*	CMS 金鑰資料庫的完整檔名。
-pw *password*	CMS 金鑰資料庫的密碼。
-label *label*	附加至憑證的金鑰標籤。
-dn *distinguished_name*	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 或 DC 屬性。
-size *key_size*	金鑰大小。 對於 iKeycmd，此值可以是 512 或 1024。 若為 runmqakm ，值可以是 512、1024、2048 或 4096。
-x509version *version*	要建立的 X.509 憑證版本。 值可以是 1、2 或 3。 預設是 3。
-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。 此模式會停用 BSafe 加密檔案庫。 只會使用 ICC 元件，且這個元件必須在 FIPS 模式中順利起始設定。 在 FIPS 模式中時，ICC 元件會使用已通過 FIPS 140-2 驗證的演算法。 如果 ICC 元件未在 FIPS 模式中起始設定，則 **runmqakm** 指令會失敗。
-sig_alg	對於 runmqakm ，這是在建立自簽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立的自簽憑證相關聯的簽章。 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。
-sig_alg	對於 iKeycmd，這是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、 SHA1WithRSA、 SHA256_WITH_RSA、 SHA256WithRSA 、 SHA2WithRSA、 SHA384_WITH_RSA、 SHA384WithRSA、 SHA512_WITH_RSA、 SHA512WithRSA、 SHA_WITH_DSA、 SHA_WITH_RSA 、 SHAWithDSA 或 SHAWithRSA。 預設值為 SHA1WithRSA。
-san_dnsname *DNS_names*	所建立項目的 DNS 名稱清單 (以逗點或空格區隔)。
-san_emailaddr *email_addresses*	所建立項目的電子郵件位址清單 (以逗點或空格區隔)。
-san_ipaddr *IP_addresses*	所建立項目的 IP 位址清單 (以逗點或空格區隔)。

distributed 在 *UNIX, Linux, and Windows* 系統上要求個人憑證

您可以使用 **strmqikm** (iKeyman) 來要求個人憑證 GUI，或從指令行使用 **runmqckm** 或 **runmqakm** 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 iKeyman GUI 或從指令行要求個人憑證，但需遵循下列考量:

- WebSphere 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。 WebSphere MQ 支援三種不同類型的 CipherSpec。 如需詳細資料，請參閱 第 29 頁的『IBM WebSphere MQ 中的數位憑證及 CipherSpec 相容性』 主題中的 第 30 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。
- 只有 runmqakm 指令會提供符合 FIPS 標準的選項。
- 如果您使用加密硬體，請參閱 第 118 頁的『要求 PKCS #11 硬體的個人憑證』。

使用 *iKeyman* 使用者介面

關於這項作業

iKeyman 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 SSL 或 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 iKeyman 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，輸入下列標籤:
   - 若為佇列管理程式，請輸入 ibmwebspheremq，後面接著已變更為小寫的佇列管理程式名稱。 例如，對於稱為 QM1 的佇列管理程式，請輸入 ibmwebspheremqqm1。
   - 若為 IBM WebSphere MQ MQI client，請輸入 ibmwebspheremq，後面接著您的登入使用者 ID (全部都是小寫); 例如 ibmwebspheremqmyuserid。
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 104 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。

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 -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。

-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。

-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。 此模式會停用 BSafe 加密檔案庫。 只會使用 ICC 元件，且這個元件必須在 FIPS 模式中順利起始設定。 處於 FIPS 模式時， ICC 元件會使用 FIPS 140-2 驗證的演算法。 如果 ICC 元件未在 FIPS 模式中起始設定，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 、 SHA1WithRSA、 SHA256_WITH_RSA、 SHA256WithRSA、 SHA2WithRSA、 SHA384_WITH_RSA 、 SHA384WithRSA、 SHA512_WITH_RSA 、 SHA512WithRSA、 SHA_WITH_DSA、 SHA_WITH_RSA、 SHAWithDSA 或 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)。

在 *UNIX, Linux, and Windows* 系統上更新現有的個人憑證

您可以使用 iKeyman 使用者介面，或使用 **iKeycmd** 或 **runmqakm** 指令，來更新個人憑證。

開始之前

如果您需要對個人憑證使用較大的金鑰大小，則下面說明的更新步驟無法運作，因為重建的憑證申請是從現有金鑰產生的。

遵循 第 104 頁的『在 UNIX, Linux, and Windows 系統上要求個人憑證』 中說明的步驟，使用您需要的金鑰大小來建立新的憑證申請。 此程序會取代您現有的金鑰。

關於這項作業

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 *iKeyman* 使用者介面

關於這項作業

iKeyman 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 SSL 或 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. 在 UNIX, Linux, and Windows 系統上使用 **strmqikm** 指令，以啟動 iKeyman 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 106 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 **iKeycmd** 或 **runmqakm** 指令來要求個人憑證:

- 在 UNIX, Linux, and Windows 系統上使用 **iKeycmd** :

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

- 使用 runmqakm:

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

其中:

-db 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。

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 107 頁的『將個人憑證接收至 UNIX、 Linux 及 Windows 系統上的金鑰儲存庫』中說明的步驟將它新增至金鑰資料庫。

將個人憑證接收至 *UNIX*、 *Linux* 及 *Windows* 系統上的金鑰儲存庫

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 iKeyman

如果您需要以符合 FIPS 標準的方式管理 SSL 憑證，請使用 runmqakm 指令。 iKeyman 不提供符合 FIPS 標準的選項。

請確定要匯入的憑證檔具有現行使用者的寫入權，然後對佇列管理程式或 WebSphere MQ MQI 用戶端使用下列程序，將個人憑證接收到金鑰資料庫檔中:

1. 使用 **strmqikm** 指令來啟動 iKeyman GUI (在 Windows UNIX and Linux 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**，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

使用下列指令，透過 iKeycmd 將個人憑證新增至金鑰資料庫檔:

- 在 UNIX、 Linux 和 Windows 上，發出下列指令:

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

其中:

`-file` *filename*	是包含個人憑證之檔案的完整檔名。
`-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。
`-format` *ascii*	是憑證的格式。 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` 。

如果您使用加密硬體，請參閱 第 120 頁的『將個人憑證匯入 PKCS #11 硬體』。

從金鑰儲存庫擷取 *CA* 憑證

請遵循此程序來擷取 CA 憑證。

使用 iKeyman

如果您需要以符合 FIPS 標準的方式管理 SSL 憑證，請使用 runmqakm 指令。 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 iKeyman 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 iKeycmd 來擷取 CA 憑證:

- 在 UNIX、 Linux 和 Windows 上:

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

其中:

-db *filename*	是 CMS 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。

在 *UNIX*、 *Linux* 及 *Windows* 系統上從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 iKeyman

如果您需要以符合 FIPS 標準的方式管理 SSL 憑證，請使用 runmqakm 指令。 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** 指令來啟動 iKeyman GUI (在 UNIX、 Linux 及 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. 選取您要從中擷取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 iKeycmd 或 runmqakm 來擷取自簽憑證的公用部分:

- 在 UNIX、 Linux 和 Windows 上:

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

- 使用 runmqakm:

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

其中:

-db *filename*	是 CMS 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。

在 *UNIX, Linux, and Windows* 系統上，將 *CA* 憑證 *(*或自簽憑證的公用部分*)* 新增至金鑰儲存庫

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: 如果您要新增的憑證是在憑證鏈中，則還必須新增在該憑證鏈中位於其上方的所有憑證。 您必須確定該憑證是採用 ASCII (UTF-8) 或二進位 (DER) 編碼，因為 IBM「廣域安全工具箱 (GSKit)」不支援其他編碼類型的憑證。 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）。

使用 iKeyman

如果您需要以符合 FIPS 標準的方式管理 SSL 憑證，請使用 runmqakm 指令。 iKeyman 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** 指令來啟動 iKeyman GUI (在 UNIX、 Linux 及 Windows 系統上)。
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**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

請使用下列指令，以使用 iKeycmd 新增 CA 憑證:

- 在 UNIX、 Linux 和 Windows 上，發出下列指令:

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

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。
-file *filename*	是包含憑證的檔案名稱。
-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。

從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 iKeyman

如果您需要以符合 FIPS 標準的方式管理 SSL 憑證，請使用 runmqakm 指令。 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 iKeyman GUI (在 Windows UNIX and Linux 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

使用下列指令，以使用 iKeycmd 來匯出個人憑證:

- 在 UNIX、 Linux 和 Windows 上:

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

其中:

-db *filename*	是 CMS 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。
-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。

將個人憑證匯入 *UNIX, Linux, and Windows* 系統上的金鑰儲存庫

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 110 頁的『在 UNIX, Linux, and Windows 系統上，將 CA 憑證 (或自簽憑證的公用部分) 新增至金鑰儲存庫』)。

PKCS #12 檔案應視為暫時檔案，並在使用後刪除。

使用 iKeyman

如果您需要以符合 FIPS 標準的方式管理 SSL 憑證，請使用 runmqakm 指令。 iKeyman 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 iKeyman 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，則此視窗容許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。 這可用來將個人憑證標籤變更為 WebSphere MQ 所需要的標籤，以便將憑證與特定佇列管理程式或用戶端相關聯 (例如 ibmwebspheremqqm1)。
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 iKeycmd 匯入個人憑證，請使用下列指令:

- 在 UNIX、 Linux 和 Windows 上:

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

其中:

`-file` *filename*	是包含 PKCS #12 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。

iKeycmd 不提供用來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。

從 *Microsoft .pfx* 檔案匯入

遵循此程序，使用 iKeyman 從 Microsoft .pfx 檔案進行 mport。 您無法使用 runmqakm 來匯入 .pfx 檔。

.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** 指令來啟動 iKeyman GUI (在 Linux、 UNIX 或 Windows 上)。 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金鑰 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**。 一般名稱的格式必須是 ibmwebspheremq，後面接著佇列管理程式名稱或 WebSphere MQ MQI 用戶端使用者登入 ID (小寫)。
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 iKeycmd 來變更憑證標籤。

從 *PKCS #7* 檔案匯入

iKeyman 和 iKeycmd 工具不支援 PKCS #7 (.p7b) 檔案。 使用 runmqckm 工具從 PKCS #7 檔案匯入憑證。

使用下列指令，從 PKCS #7 檔案新增 CA 憑證:

```
runmqckm -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:

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

-db *filename*	是包含 PKCS #7 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。

從 *UNIX, Linux, and Windows* 系統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 iKeyman

如果您需要以符合 FIPS 標準的方式管理 SSL 憑證，請使用 runmqakm 指令。 iKeyman 不提供符合 FIPS 標準的選項。

1. 使用 **strmqikm** 指令來啟動 iKeyman GUI (在 UNIX、 Linux 及 Windows 系統上)。
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 111 頁的『從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 iKeycmd 或 runmqakm 來刪除憑證:

- 在 UNIX、 Linux 和 Windows 上:

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

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。 此模式會停用 BSafe 加密檔案庫。 只會使用 ICC 元件，且這個元件必須在 FIPS 模式中順利起始設定。 在 FIPS 模式中時，ICC 元件會使用已通過 FIPS 140-2 驗證的演算法。 如果 ICC 元件未在 FIPS 模式中起始設定，則 **runmqakm** 指令會失敗。

產生金鑰儲存庫保護的高保護性密碼

您可以使用 **runmqakm** 指令來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 UNIX 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 iKeyman 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。

配置 *UNIX, Linux, and Windows* 系統上的加密硬體

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。

您可以使用下列一種方法，為 UNIX、 Linux 或 Windows 系統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 WebSphere MQ Explorer 來配置 UNIX、 Linux 或 Windows 系統上的加密硬體。 如需相關資訊，請參閱線上說明。

您可以使用下列一種方法，在 UNIX、 Linux 或 Windows 系統上配置 WebSphere MQ 用戶端的加密硬體:

- 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_PCS11 版本，則必須完全以小寫形式指定 PKCS #11 記號標籤。
- 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。

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 116 頁的『在 PKCS #11 硬體上管理憑證』。

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 WebSphere 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:
   - 在 UNIX, Linux, and Windows 系統上:

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

   - 使用 runmqakm:

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

     其中:

-db 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 WebSphere MQ，它必須是 cms。)

-stash
將金鑰資料庫密碼儲存至檔案。

-fips
停用使用 BSafe 加密程式庫。 只會使用 ICC 元件，且這個元件必須在 FIPS 模式中順利起始設定。 處於 FIPS 模式時， ICC 元件會使用 FIPS 140-2 驗證的演算法。 如果 ICC 元件未在 FIPS 模式中起始設定，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 UNIX 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. 執行 **strmqikm** 指令，以啟動 iKeyman 使用者介面。
4. 按一下 **金鑰資料庫檔** > **開啟**。
5. 按一下 **金鑰資料庫類型** ，然後選取 **PKCS11Direct**。
6. 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。

   如果您要使用儲存在 PKCS #11 加密硬體上的憑證或金鑰，請注意 iKeycmd 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 iKeycmd 程式在這些平台上是 32 位元。
7. 在 **位置** 欄位中，輸入路徑:

   - 例如，在 UNIX 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。
8. 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。
9. 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 118 頁的『13』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**。
10. 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。
11. 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 WebSphere MQ MQI 用戶端: /var/mqm/ssl

    按一下**確定**。 這時會開啟「密碼提示」視窗。
12. 請輸入密碼。

如果您在步驟 第 117 頁的『9』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 117 頁的『9』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 SSL 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**。 即會開啟一個視窗，確認密碼位於檔案 `key.sth` 中 (除非您指定不同的詞幹名稱)。

13. 按一下**確定**。 即會顯示金鑰資料庫內容頁框。

要求 *PKCS #11* 硬體的個人憑證

針對佇列管理程式或 IBM WebSphere MQ MQI 用戶端，使用此程序來要求加密硬體的個人憑證。

使用 *iKeyman* 使用者介面

關於這項作業

註: WebSphere 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 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 116 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，輸入下列標籤:
   - 若為佇列管理程式，請輸入 `ibmwebspheremq`，後面接著已變更為小寫的佇列管理程式名稱。 例如，對於稱為 `QM1` 的佇列管理程式，請輸入 `ibmwebspheremqqm1`。
   - 若為 IBM WebSphere MQ MQI client，請輸入 `ibmwebspheremq`，後面接著您的登入使用者 ID (全部都是小寫); 例如 `ibmwebspheremqmyuserid`。
4. 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』。
5. 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。
6. 按一下**確定**。
   就會開啟確認視窗。
7. 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 118 頁的『5』中選擇的檔案中。
8. 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 -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。

-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。

-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。 此模式會停用 BSafe 加密檔案庫。 只會使用 ICC 元件，且這個元件必須在 FIPS 模式中順利起始設定。 處於 FIPS 模式時， ICC 元件會使用 FIPS 140-2 驗證的演算法。 如果 ICC 元件未在 FIPS 模式中起始設定，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 、 SHA1WithRSA、 SHA256_WITH_RSA、 SHA256WithRSA、 SHA2WithRSA、 SHA384_WITH_RSA 、 SHA384WithRSA、 SHA512_WITH_RSA 、 SHA512WithRSA、 SHA_WITH_DSA、 SHA_WITH_RSA、 SHAWithDSA 或 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)。

將個人憑證匯入 *PKCS #11* 硬體
請針對佇列管理程式或 IBM WebSphere MQ MQI 用戶端使用此程序，將個人憑證匯入加密硬體。

使用 *iKeyman*

程序

若要從 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 116 頁的『在 PKCS #11 硬體上管理憑證』。
2. 按一下 **接收**。 即會開啟「從檔案接收憑證」視窗。
3. 選取新個人憑證的 **資料類型**；例如， `Base64-encoded ASCII data` 代表副檔名為 `.arm` 的檔案。
4. 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。
5. 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。
6. 請按一下**是**或**否**。 這時會開啟「輸入標籤」視窗。
7. 鍵入標籤。
   例如，您可以使用與要求個人憑證時相同的標籤。 請注意，標籤必須採用正確的 IBM WebSphere MQ 格式:
   - 若為佇列管理程式， `ibmwebspheremq` 後面接著小寫的佇列管理程式名稱。 例如，對於稱為 QM1 的佇列管理程式，標籤會是: `ibmwebspheremqqm1`。
   - 若為 IBM WebSphere MQ MQI 用戶端， `ibmwebspheremq` 後面接著小寫的登入使用者 ID。 例如，對於使用者 ID `MyUserID`，標籤會是: `ibmwebspheremqmyuserid`。
8. 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。

使用指令行

程序

若要從指令行要求個人憑證，請完成下列步驟:

1. 開啟針對您環境所配置的指令視窗。
2. 輸入適合您作業系統及配置的指令:
   - 在 Windows、 UNIX and Linux 系統上，使用下列其中一個指令:

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

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

   其中:

   -file *filename*
   指定包含個人憑證之檔案的完整檔名。

   -crypto *path*
   指定硬體隨附之 PKCS #11 檔案庫的完整路徑。

   -tokenlabel *hardware_token*
   指定在安裝期間提供給加密硬體儲存部分的標籤。

-pw *hardware_password*
指定用於存取硬體的密碼。

-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
指定在 FIPS 模式中執行指令此模式會停用 BSafe 加密程式庫。 只會使用 ICC 元件，且這個元件必須在 FIPS 模式中順利起始設定。 處於 FIPS 模式時， ICC 元件會使用 FIPS 140-2 驗證的演算法。 如果 ICC 元件未在 FIPS 模式中起始設定，則 **runmqakm** 指令會失敗。

識別及鑑別使用者

您可以使用 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 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。

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 WebSphere MQ 用戶端應用程式的使用者。 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 WebSphere MQ MQI 用戶端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 86 頁的『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。

Secure Sockets Layer (SSL) 和傳輸層安全 (TLS) 使用 PKI 技術，如剛才說明的技術。 如需 SSL 和 TLS 如何執行鑑別的相關資訊，請參閱 第 13 頁的『Secure Sockets Layer (SSL) 和傳輸層安全 (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 189 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 60 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 124 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。

特許使用者

特許使用者是對 WebSphere MQ 具有完整管理權限的使用者。

除了下表中列出的使用者之外，對佇列具有 +crt 權限之任何群組的成員都是間接管理者。 同樣地，任何對佇列管理程式具有 +set 權限，且對指令佇列具有 +put 權限的使用者都是管理者。

您不應該將這些專用權授與一般使用者及應用程式。

表 *13:* 依平台列出特許使用者.

特許使用者的表格。 在 Windows、SYSTEM、mqm 群組的所有成員以及 Administrators 群組的所有成員都是特許使用者。 在 UNIX and Linux 系統上， mqm 群組的所有成員都是特許使用者。 在 IBM i 上，設定檔 (使用者) qmqm 及 qmqmadm、qmqmadm 群組的所有成員，以及使用 *ALLOBJ 設定所定義的任何使用者都是特許使用者。

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
UNIX and Linux 系統	• mqm 群組的成員

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。

IBM WebSphere MQ 隨附的授權服務元件稱為「物件權限管理程式 (OAM)」。 OAM 會根據 MQCSP 中包含的 ID 來授權使用者，但不會驗證密碼。 可以在授權服務中實作密碼驗證，方法是將鏈結結束程式與 OAM 搭配使用，或將 OAM 取代為替代授權服務。

您可以在安全結束程式中變更 MQCSP。

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 WebSphere MQ MQI 用戶端應用程式的使用者來運作。 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 WebSphere MQ MQI 用戶端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 86 頁的『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。

Secure Sockets Layer (SSL) 和傳輸層安全 (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『Secure Sockets Layer (SSL) 和傳輸層安全 (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 189 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 60 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 124 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 SSL 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。

在 UNIX、 Linux 及 Windows 系統上， WebSphere MQ SSL 支援使用 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 WebSphere MQ classes for Java 和 IBM WebSphere 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小節中所述，配置 OCSP。

在 z/Os 和 IBM i WebSphere MQ SSL 支援僅使用 LDAP 伺服器上的 CRL 和 ARL 來檢查已撤銷的憑證。

如需憑證的相關資訊

權限，請參閱 第 9 頁的『數位憑證』。

撤銷的憑證及 OCSP

IBM WebSphere 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 Windows、 UNIX and Linux 系統上的 WebSphere MQ 。

若要使用 OCSP 檢查數位憑證的撤銷狀態，WebSphere MQ 可以使用兩種方法，來判定要聯絡哪一個 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 或 SSL 憑證是否已撤銷，可能是因為您是在測試環境中執行，所以可以在 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 WebSphere MQ 無法存取 OCSP 回應者。
- OCSP 回應端已傳送回應，但 WebSphere 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 WebSphere MQ 收到**不明**的 OCSP 結果，則其行為視 OCSPAuthentication 屬性的設定而定。若為佇列管理程式，此屬性保留在 UNIX and Linux 系統或 Windows 登錄之 `qm.ini` 檔案的 SSL 段落中。 您可以使用「IBM WebSphere MQ 檔案總管」來設定它。 若為用戶端，這個屬性存放在用戶端配置檔的 SSL 段落中。

如果收到的結果為**不明**，且 OCSPAuthentication 設為 REQUIRED（預設值），則 WebSphere MQ 會拒絕連線，並發出類型 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。

如果收到的結果為**不明**，且 OCSPAuthentication 設為 OPTIONAL，則 WebSphere MQ 會容許啟動 SSL 通道，且不會產生任何警告或 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 WebSphere 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；您建立 SSL 連線所採取的步驟，即足以驗證 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（且該憑證是由 SSL 連線已經信任的 CA 所簽署），所以不需要任何其他憑證設定。
- 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 檢查的用戶端或佇列管理程式之金鑰資料庫；請參閱第 110 頁的『在 UNIX, Linux, and Windows 系統上，將 CA 憑證 (或自簽憑證的公用部分) 新增至金鑰儲存庫』。 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。 如果不新增這個憑證，WebSphere MQ 就無法驗證 OCSP 回應的數位簽章，且 OCSP 檢查會導致「不明」結果，這可能會造成 IBM WebSphere MQ 關閉通道（視 OCSPAuthentication 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，只有在為整個 Java 虛擬機器 (JVM) 處理程序啟用 OCSP 時，WebSphere MQ 才能針對 SSL 及 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，受限於已生效的任何「Java 安全管理程式」原則。

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。 在部分情況中，當 JVM 與使用 SSL 或 TLS 安全 Socket 的其他應用程式碼共用時，此配置可能會有不想要的副作用。 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 臨時修正程式及產品維護時，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 AMS Java 攔截程式，如在原生攔截程式中啟用 OCSP 檢查中所述，請小心避免使用與金鑰儲存庫配置檔中的 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

WebSphere 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。

WebSphere 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 WebSphere MQ MQI 用戶端收到憑證時，會檢查 CRL 以確認該憑證仍然有效。 WebSphere MQ 會先檢查快取中是否有快取。 如果 CRL 不在快取中， WebSphere MQ 會依照 LDAP CRL 伺服器位置在 *SSLCRLNamelist* 屬性指定的鑑別資訊物件名單中的出現順序來詢問它們，直到 WebSphere 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。

如需 LDAP 的相關資訊，請參閱 搭配使用輕量型目錄存取通訊協定服務與 WebSphere MQ for Windows。

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。 為了與 WebSphere MQ SSL 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 128 頁的圖 12 顯示範例 LDIF 檔案，您可以建立該檔案作為 LDAP 伺服器的輸入，以載入 CA1 發出的 CRL 及 ARL， 它是一個虛構的「憑證管理中心」， 具有識別名稱 "CN=CA1， OU=Test， O=IBM， C=GB"，由 IBM 內的「測試」組織設定。

```
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 *12:* 憑證管理中心的 *LDIF* 檔案範例。 這可能因實作不同而異。

第 129 頁的圖 13 顯示當您載入 第 128 頁的圖 12 中顯示的範例 LDIF 檔案時， LDAP 伺服器所建立的 DIT 結構，以及 CA2 的類似檔案 (也是 IBM 內 PKI 組織所設定的虛擬「憑證管理中心」)。

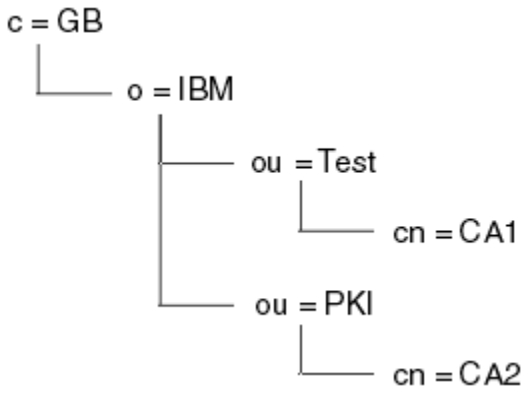


圖 *13: LDAP* 目錄資訊樹狀結構的範例

WebSphere 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。 WebSphere 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 129 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。

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 *SSLCRLNamelist* 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。

   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。

   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。
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 *SSLCRLNamelist* 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。

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。

使用 *IBM WebSphere MQ Explorer* 存取 *CRL* 及 *ARL*
您可以使用 IBM WebSphere 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. 將 CRL (LDAP) 物件的名稱 (從步驟 第 130 頁的『2.a』) 新增至清單。
   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. 在 **CRL 名單** 欄位中輸入名單名稱 (來自步驟 第 130 頁的『3.a』)。

使用 *IBM WebSphere MQ MQI* 用戶端存取 *CRL* 及 *ARL*
您有三個選項可指定 LDAP 伺服器保留 CRL 供 IBM WebSphere MQ MQI 用戶端檢查。

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 WebSphere MQ MQI 用戶端通道存取 LDAP CRL。

OCSP 回應者及保留 *CRL* 之 *LDAP* 伺服器的位置
在 IBM WebSphere MQ MQI 用戶端系統上，您可以指定 OCSP 回應端以及保存憑證撤銷清冊 (CRL) 之「輕量型目錄存取通訊協定 (LDAP)」伺服器的位置。

您可以使用三種方式來指定這些位置，在這裡以遞減優先順序列出。

當 WebSphere MQ MQI 用戶端應用程式發出 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 結構都包含 WebSphere MQ MQI 用戶端存取 OCSP 回應端或 LDAP 伺服器 (保存 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， WebSphere MQ MQI 用戶端可以存取 OCSP 回應端或 LDAP 伺服器，且會保留 CRL，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。

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 WebSphere MQ MQI 用戶端存取 OCSP 回應端或保留 CRL 的 LDAP 伺服器，則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:

在伺服器平台 AIX、 HP-UX、 Linux、 Solaris 及 Windows 上

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 **SSLCRLNameList** 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。 如果一個 LDAP 伺服器在需要時無法使用， WebSphere MQ MQI 用戶端可以嘗試存取另一個 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 **SSLCRLNameList** 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 WebSphere MQ MQI 用戶端可以使用 CCDT 中的憑證撤銷位置，來存取保留 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 **SSLCRLNameList** 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。

在 Windows 上使用 Active Directory

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。

使用適用於 *Java* 的 *IBM WebSphere MQ* 類別及適用於 *JMS* 的 *IBM WebSphere MQ* 類別來存取 *CRL* 及 *ARL*

Java 的 IBM WebSphere MQ 類別和 JMS 的 IBM WebSphere MQ 類別存取 CRL 與其他平台不同。

如需搭配使用 CRL 及 ARL 與 IBM WebSphere MQ classes for Java 的相關資訊，請參閱 使用憑證撤銷清冊。

如需搭配使用 CRL 及 ARL 與適用於 JMS 的 IBM WebSphere MQ 類別的相關資訊，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 WebSphere 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 Script (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 。

在有可用的平台上，您也可以使用「 WebSphere MQ 探險家」。

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。

在 UNIX, 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。 它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系統，以及授與使用者管理 IBM WebSphere MQ 及使用 IBM WebSphere 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。

在 UNIX、 Linux 和 Windows 系統上使用 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，用來授與及撤銷 WebSphere 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 165 頁的『在 UNIX, Linux, and Windows 系統上管理 IBM WebSphere MQ 的權限』中所述。 獲授權管理 WebSphere MQ 的使用者 ID 具有佇列管理程式的 超級使用者 權限，這表示您不需要將發出任何 MQI 要求或指令的進一步許可權授與他們。

授與 UNIX, Linux, and Windows 系統上 IBM WebSphere MQ 物件的存取權

使用 **setmqaut** 控制指令或 **MQCMD_SET_AUTH_REC** PCF 指令，為使用者及使用者群組提供 IBM WebSphere MQ 物件的存取權。

如需 **setmqaut** 控制指令及其語法的完整定義，請參閱 setmqaut; 如需 **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 133 頁的『在 UNIX, Linux, and Windows 系統上使用 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 UNIX and Linux 系統上，此指令也會撤銷與 fvuser 相同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。

在 UNIX, Linux, and Windows 系統上使用 OAM 通用設定檔

OAM 通用設定檔可讓您一次設定使用者對許多物件的權限，而不必在建立個別物件時對個別物件發出個別 **setmqaut** 指令。

在 **setmqaut** 指令中使用通用設定檔可讓您為所有符合該設定檔的物件設定通用權限。

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; * 在此環境定義中使用一律表示一個限定元。)
- 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 UNIX 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，在上述範例中，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 **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 UNIX and Linux 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: 僅限 WebSphere 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
```

在 *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; * 在此環境定義中使用一律表示一個限定元。)
- 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 UNIX 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

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，在上述範例中，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

使用 **dmpmqaut** 控制指令或 **MQCMD_INQUIRE_AUTH_RECS** PCF 指令來傾出與指定設定檔相關的現行授權。

如需 **dmpmqaut** 控制指令及其語法的完整定義，請參閱 dmpmqaut; 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: UNIX 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: 僅限 WebSphere 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
```

顯示存取設定

使用 **dspmqaut** 控制指令或 **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。

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 **MQCMD_INQUIRE_ENTITY_AUTH** PCF 指令及其語法的完整定義，請參閱 查詢實體權限。

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
```

變更及撤銷 IBM WebSphere 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** 指令。 若要撤銷特定使用者的存取權，該使用者是具有授權的群組成員，請從群組中移除該使用者。

從群組中移除使用者的處理程序說明如下:

- 第 66 頁的『在 Windows 上建立及管理群組』
- 第 68 頁的『在 HP-UX 上建立及管理群組』
- 第 70 頁的『在 AIX 上建立及管理群組』
- 第 71 頁的『在 Solaris 上建立及管理群組』
- 第 71 頁的『在 Linux 上建立及管理群組』

.

建立 IBM WebSphere 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 **MQCMD_INQUIRE_ENTITY_AUTH** PCF 指令及其語法的完整定義，請參閱 查詢實體權限。

在 Windows 上，請先刪除對應於特定 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。

在 UNIX, Linux, and Windows 系統上防止安全存取檢查

若要關閉所有安全檢查，您可以停用 OAM。 這可能適用於測試環境。 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 MQSNOAUT (如果您這樣做，則稍後無法新增 OAM):

  如需設定 MQSNAUT 變數之 含意 的相關資訊，請參閱 環境變數 。
- 編輯佇列管理程式配置檔以移除服務。 (如果您這樣做，則稍後無法新增 OAM。)

如果您在停用 OAM 時使用 setmqaut 或 dspmqaut，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。

警告: 當移除 OAM 時，無法將它放回現有的佇列管理程式。 此 是因為 OAM 需要在建立物件時就定位。 移除 OAM 之後，如果要再次使用 WebSphere MQ，則需要重建佇列管理程式。

相關概念

可安裝的服務

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 WebSphere MQ 系統。

關於這項作業

在這項作業期間，您決定需要採取哪些動作，將適當的安全層次套用至 WebSphere MQ 安裝的元素。 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 140 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 144 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 148 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 149 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 150 頁的『移除佇列管理程式的連線功能』 中所述

b) 是: 請參閱 第 151 頁的『容許使用者應用程式連接至佇列管理程式』。

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 *14:* 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 140 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 141 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 141 頁的『授與對部分通道的有限管理存取權』 中所述
佇列管理程式	授與局部管理存取權給佇列管理程式，如 第 142 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 143 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 143 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 144 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。

程序

- 若為 UNIX、 Linux 及 Windows 系統，請發出下列指令:

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

- 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。

  ReqdAction
  您容許群組採取的動作:

  – 在 UNIX、 Linux 及 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。

  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。

  若為 *ReqdAction* 的其他值，則為 QLOCAL、QALIAS、QMODEL 或 QREMOTE 其中一個值。

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。

程序

- 若為 UNIX、 Linux 及 Windows 系統，請發出下列指令:

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

- 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:

  – 在 UNIX、 Linux 及 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。

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。

程序

- 若為 UNIX、 Linux 及 Windows 系統，請發出下列指令:

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

- 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。

  ReqdAction
  您容許群組採取的動作:

  – 在 UNIX、 Linux 及 Windows 系統上，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。

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。

程序

- 若為 UNIX、 Linux 及 Windows 系統，請發出下列指令:

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

- 若為 IBM i，請發出下列指令:

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

結果

若要判定使用者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:

- 在 UNIX、 Linux 及 Windows 系統上，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。

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。

程序

- 若為 UNIX、 Linux 及 Windows 系統，請發出下列指令:

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

- 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:

  – 在 UNIX、 Linux 及 Windows 系統上，下列授權的任何組合: + chg、 + clr、 + crt、 + dlt、 + dsp。 authorization + alladm 相當於 + chg + clr + dlt + dsp。

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。

程序

- 若為 UNIX、 Linux 及 Windows 系統，請發出下列指令:

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

- 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:

– 在 UNIX、 Linux 及 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。

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。

註: 服務物件不存在於 z/OS 上。

程序

- 若為 UNIX、 Linux 及 Windows 系統，請發出下列指令:

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

- 若為 IBM i，請發出下列指令:

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

結果

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:

- 在 UNIX、 Linux 及 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。

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 *15:* 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 145 頁的『授與部分佇列的完整管理存取權』 中所述

表 *15:* 授與對佇列管理程式資源子集的完整管理存取權 (繼續)	
使用者需要管理此類型的物件	**執行此動作**
主題	授與必要主題的完整管理存取權，如 第 145 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 146 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 146 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 147 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 147 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 148 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。

程序

- 若為 UNIX、 Linux 及 Windows 系統，請發出下列指令:

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

- 若為 IBM i，請發出下列指令:

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

- 若為 z/OS，請發出下列指令:

```
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。 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。

程序

- 若為 UNIX、 Linux 及 Windows 系統，請發出下列指令:

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

- 若為 IBM i，請發出下列指令:

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

- 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)
```

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。

程序

- 若為 UNIX、 Linux 及 Windows 系統，請發出下列指令:

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

- 若為 IBM i，請發出下列指令:

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

- 若為 z/OS，請發出下列指令:

```
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。 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。

程序
- 若為 UNIX、 Linux 及 Windows 系統，請發出下列指令:

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

- 若為 IBM i，請發出下列指令:

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

- 若為 z/OS，請發出下列指令:

```
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。 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。

程序
- 若為 UNIX、 Linux 及 Windows 系統，請發出下列指令:

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

- 若為 IBM i，請發出下列指令:

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

- 若為 z/OS，請發出下列指令:

```
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。 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。

程序
- 若為 UNIX、 Linux 及 Windows 系統，請發出下列指令:

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

- 若為 IBM i，請發出下列指令:

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

- 若為 z/OS，請發出下列指令:

```
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。 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。

程序
- 若為 UNIX、 Linux 及 Windows 系統，請發出下列指令:

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

- 若為 IBM i，請發出下列指令:

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

- 若為 z/OS，請發出下列指令:

```
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。 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。

程序

- 使用精靈:

  a) 在 WebSphere MQ 探險家 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。

- 若為 UNIX 及 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
setmqaut -m QMgrName -n ** -t topic -g GroupName +dsp
setmqaut -m QMgrName -n ** -t channel -g GroupName +dsp +inq
setmqaut -m QMgrName -n ** -t clntconn -g GroupName +dsp
setmqaut -m QMgrName -n ** -t authinfo -g GroupName +dsp
setmqaut -m QMgrName -n ** -t listener -g GroupName +dsp
setmqaut -m QMgrName -n ** -t namelist -g GroupName +dsp
setmqaut -m QMgrName -n ** -t process -g GroupName +dsp
setmqaut -m QMgrName -n ** -t service -g GroupName +dsp
setmqaut -m QMgrName -t qmgr -g GroupName +dsp +inq +connect
```

  只有在您想要使用「 MQ 探險家」時，才需要 SYSTEM.ADMIN.COMMAND.QUEUE 及 SYSTEM.MQEXPLORER.REPLY.MODEL 的特定權限。

- 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')
```

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

  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。

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

請使用「新增角色型權限」精靈或適合您作業系統的指令。

程序

- 使用精靈:

  a) 在 WebSphere MQ 探險家 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。

- 若為 UNIX 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.QUEUE -t queue -g GroupName +dsp +inq +get
setmqaut -m QMgrName -n '**' -t topic -g GroupName +alladm
setmqaut -m QMgrName -n @class -t topic -g GroupName +crt
setmqaut -m QMgrName -n '**' -t channel -g GroupName +alladm
setmqaut -m QMgrName -n @class -t channel -g GroupName +crt
setmqaut -m QMgrName -n '**' -t clntconn -g GroupName +alladm
setmqaut -m QMgrName -n @class -t clntconn -g GroupName +crt
setmqaut -m QMgrName -n '**' -t authinfo -g GroupName +alladm
setmqaut -m QMgrName -n @class -t authinfo -g GroupName +crt
setmqaut -m QMgrName -n '**' -t listener -g GroupName +alladm
setmqaut -m QMgrName -n @class -t listener -g GroupName +crt
setmqaut -m QMgrName -n '**' -t namelist -g GroupName +alladm
setmqaut -m QMgrName -n @class -t namelist -g GroupName +crt
setmqaut -m QMgrName -n '**' -t process -g GroupName +alladm
setmqaut -m QMgrName -n @class -t process -g GroupName +crt
setmqaut -m QMgrName -n '**' -t service -g GroupName +alladm
setmqaut -m QMgrName -n @class -t service -g GroupName +crt
setmqaut -m QMgrName -t qmgr -g GroupName +alladm +conn
```

- 若為 Windows 系統，請發出與 UNIX and Linux 系統相同的指令，但使用設定檔名稱 @CLASS 而非 @class。
- 若為 IBM i，請發出下列指令:

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

- 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。 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。

程序

- 若為 UNIX、 Linux 及 Windows 系統，請發出下列指令:

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

- 若為 IBM i，請發出下列指令:

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

- 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。 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 157 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 151 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 157 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 158 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 162 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 164 頁的『授與查詢佇列管理程式的權限』。
您具有使用程序物件的應用程式	請參閱 第 164 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 165 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 SSL 或 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. 將 SSL 或 TLS 與通道鑑別記錄搭配使用:
   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 SSL 或 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 長度欄位中傳送至 MQCD 結構中結束程式的 SSL DN 值，撰寫安全結束程式來指派 MCAUSER 值。
3. 將 SSL 或 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 SSL 或 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 152 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 123 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。 如果您在下面顯示的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

將用戶端主張的使用者 *ID* 對映至 *MCAUSER* 使用者 *ID*
您可以使用通道鑑別記錄，根據從用戶端收到的原始使用者 ID 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 是用戶端所主張的使用者 ID。

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

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
) 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 WebSphere MQ 規則。 請參閱 WebSphere SSLPEER 值的 MQ 規則。

user 是要用於所有使用指定 DN 之連線的使用者 ID。

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。 如果您在下面顯示的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

封鎖用戶端主張使用者 *ID* 的存取

您可以使用通道鑑別記錄來防止用戶端主張的使用者 ID 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 是用戶端所主張的使用者 ID。

相關參考

SET CHLAUTH

封鎖存取 *SSL* 識別名稱

您可以使用通道鑑別記錄來防止「SSL 識別名稱」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')
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 WebSphere MQ 規則。 請參閱 WebSphere SSLPEER 值的 MQ 規則。

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。

程序

- 若為 UNIX、 Linux 及 Windows 系統，請發出下列指令:

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

- 若為 IBM i，請發出下列指令:

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

- 若為 z/OS，請發出下列指令:

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

這些指令提供權限來連接批次、 CICS、 IMS 及通道起始程式 (CHIN)。 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。

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 158 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 159 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 160 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 207 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 161 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 161 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 162 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。

程序

- 若為 UNIX、 Linux 及 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(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。

程序

- 若為 UNIX、 Linux 及 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
```

- 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。

程序
- 若為 UNIX、 Linux 及 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
```

- 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')
```

- 若為 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。

程序
- 若為 UNIX、 Linux 及 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。

程序
- 若為 UNIX、 Linux 及 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 161 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 UNIX、 Linux 及 Windows 系統上，您也可以使用 SET AUTHREC 指令。

程序

- 若為 UNIX、 Linux 及 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  QMgrNameObjectProfile UACC(NONE)
PERMIT QMgrNameObjectProfile CLASS(MQADMIN)
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 *16:* 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 163 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 163 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。

程序
- 若為 UNIX、 Linux 及 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。

程序
- 若為 UNIX、 Linux 及 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。

程序

- 若為 UNIX、 Linux 及 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。

程序

- 若為 UNIX、 Linux 及 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。

程序

- 若為 UNIX、 Linux 及 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。

在 UNIX, Linux, and Windows 系統上管理 IBM WebSphere MQ 的權限

IBM WebSphere MQ 管理者可以使用所有 IBM WebSphere 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 WebSphere MQ 管理者有權使用所有 WebSphere MQ 指令 (包括將 WebSphere MQ 權限授與其他使用者的指令)

若要成為 IBM WebSphere MQ 管理者，您必須是稱為 *mqm* 群組的特殊群組成員 (或 Windows 系統上的 Administrators 群組成員)。 安裝 WebSphere MQ 時會自動建立 mqm 群組; 請將更多使用者新增至群組，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 WebSphere 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 WebSphere MQ 資源。 用於管理權限記錄的 PCF 指令可供已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。

管理者可以使用控制指令 **runmqsc** 來發出 IBM WebSphere 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。 「WebSphere MQ 探險家」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「WebSphere MQ 探險家」來管理本端系統上的佇列管理程式。 當使用「IBM WebSphere MQ 探險家」來管理另一個系統上的佇列管理程式時，管理者必須具有必要權限，遠端佇列管理程式才能處理 PCF 指令。

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進行操作的 PCF 指令，請參閱 使用 WebSphere 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

此外，在 Windows 系統上， SYSTEM 帳戶對 WebSphere MQ 資源具有完整存取權。

在 UNIX 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 WebSphere 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，如 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 WebSphere 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，但在 Windows 系統上，如果您具有相同名稱的本端使用者及網域鑑別使用者，則可能會拒絕權限。 這說明於第 169 頁的『主體和群組』。

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「管理者」群組的成員也一樣。 如果您的使用者 ID 位於 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 WebSphere MQ 管理指令 (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，即可執行下列動作:

- 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 57 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 呼叫上使用捷徑連結)。
- 使用 crtmqcvx 指令來建立程式碼片段，以針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 WebSphere 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 版已提高此限制，但 WebSphere MQ 在所有 UNIX and Linux 平台上持續觀察到 12 個字元的限制。 如果您使用大於 12 個字元的使用者 ID ， WebSphere MQ 會將它取代為值 UNKNOWN。 請勿定義值為 UNKNOWN 的使用者 ID。

管理 mqm 群組

mqm 群組中的使用者已獲授與 WebSphere 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 WebSphere MQ 管理者的帳戶。

這些作業說明如下:

- 在 Windows 上建立及管理群組

- 在 HP-UX 上建立及管理群組
- 在 AIX
- 在 Solaris 上建立及管理群組
- 在 Linux 上建立及管理群組

如果網域控制站在 Windows 2000 或 Windows 2003 上執行，則網域管理者可能必須設定一個特殊帳戶供 WebSphere MQ 使用。 這在 配置 WebSphere MQ 帳戶中有說明。

在 UNIX, Linux, and Windows 系統上使用 IBM WebSphere MQ 物件的權限

所有物件都受到 IBM WebSphere 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 WebSphere MQ 保護，應用程式需要獲得存取它們的許可權。 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授與建立物件的使用者 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 165 頁的『在 UNIX, Linux, and Windows 系統上管理 IBM WebSphere MQ 的權限』

IBM WebSphere MQ 管理者可以使用所有 IBM WebSphere 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。

在 UNIX, Linux, and Windows 系統上進行安全檢查時

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 WebSphere 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 WebSphere MQ; WebSphere MQ 會假設使用者已登入基礎作業系統，且已經過該系統的鑑別。

開啟物件 (MQOPEN 或 MQPUT1 呼叫)

WebSphere MQ 物件是透過開啟物件並對其發出指令來存取。 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)。

WebSphere 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 75 頁的『1』中有說明。

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，會針對在主題樹狀結構中找到的主題物件執行權限檢查，該主題樹狀結構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。

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。

IBM WebSphere MQ 在 UNIX, Linux, and Windows 系統上如何實作存取控制

IBM WebSphere 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 WebSphere 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 WebSphere MQ 的一部分。 WebSphere 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 139 頁的『在 UNIX, Linux, and Windows 系統上防止安全存取檢查』中所述)，否則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。 使用者必須具備正確的權限，才能存取 WebSphere MQ 物件。 第 132 頁的『在 UNIX、 Linux 和 Windows 系統上使用 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。

WebSphere 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 132 頁的『在 UNIX、 Linux 和 Windows 系統上使用 OAM 來控制對物件的存取權』。

WebSphere MQ 會向 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。 WebSphere MQ 遵循 OAM 的決策; 如果 OAM 無法做出決策，則 WebSphere MQ 不容許存取。

識別 UNIX, Linux, and Windows 系統上的使用者 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 WebSphere 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 UNIX 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 WebSphere 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 170 頁的『UNIX、 Linux 及 Windows 系統上的環境定義權限』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。

主體和群組

主體可以屬於群組。 您可以將特定資源的存取權授與群組，而非個人，以減少所需的管理量。 在 UNIX and Linux 系統上，所有「存取控制清單 (ACL)」都以群組為基礎，但在 Windows 系統上， ACLS 則以使用者 ID 和群組為基礎。

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。 此程序說明如下:

- 在 Windows 上建立及管理群組
- 在 HP-UX 上建立及管理群組
- 在 AIX
- 在 Solaris 上建立及管理群組
- 在 Linux 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 PCF 對等項目) ，否則在重新啟動佇列管理程式之前，無法辨識您對主體群組成員資格所做的任何變更。

UNIX and Linux 系統

所有 ACL 都以群組為基礎。 當使用者獲授與特定資源的存取權時，使用者 ID 的主要群組會併入 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，以將 WebSphere MQ 資源的存取權授與沒有特定授權的使用者。

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。

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 系統

ACL 同時以使用者 ID 和群組為基礎。 除了個別使用者 ID 也可以顯示在 ACL 之外，檢查與 UNIX 系統的檢查相同。 您可以在不同網域上具有相同使用者 ID 的不同使用者。 WebSphere MQ 允許使用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\GroupName
```

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含下列設定: `GroupModel=GlobalGroups`; 請參閱 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 WebSphere 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，則當系統具有不在本端 mqm 群組中的同名本端使用者時，指令會失敗。

Windows 安全 *ID (SID)*

WebSphere Windows 上的 MQ 使用可用的 SID。 如果授權要求未提供 Windows SID ， WebSphere MQ 會單獨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 包含在定義使用者的 Windows 安全帳戶管理程式 (SAM) 資料庫上識別完整使用者帳戶詳細資料的資訊。 在 WebSphere MQ for Windows 上建立訊息時， WebSphere MQ 會將 SID 儲存在訊息描述子中。 當 WebSphere 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， WebSphere 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 WebSphere MQ 權限。 若要防止此問題，請在每一個授權要求中包括 SID; WebSphere 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。

UNIX、 Linux 及 Windows 系統上的替代使用者權限

您可以指定在存取 WebSphere 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 WebSphere 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。

UNIX、 Linux 及 Windows 系統上的環境定義權限

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 中欄位的起始值取決於 WebSphere MQ 管理者所建立的通道定義。 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。 *MCAUserIdentifier* 包含 MCA 使用者 ID 的前 12 個位元組。 如果 MCA 使用者 ID 不是空白，則會指定訊息通道代理程式用於授權存取 MQ 資源的使用者 ID。 在 Windows 平台上，請確定 MCAUSER 少於 12 個字元。

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。

在 IBM i、 UNIX、 Linux 及 Windows 系統上，除非 *MCAUserIdentifier* 值空白，否則當 MCA 在連接至佇列管理程式之後嘗試存取佇列管理程式的資源時，佇列管理程式會使用 *MCAUserIdentifier* 值作為權限檢查的使用者 ID。 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。

在 z/OS 上，如果佇列管理程式不是空白，則可能會使用 *MCAUserIdentifier* 值進行權限檢查。 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:

- 如果 MQI 通道的用戶端端沒有安全結束程式，則當用戶端應用程式發出 MQCONN 呼叫時，與 WebSphere MQ 用戶端應用程式相關聯的使用者 ID 會從用戶端連線 MCA 流向伺服器連線 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 WebSphere MQ MQI 用戶端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 SSL，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 WebSphere MQ v7.1 之前所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。

WebSphere MQ 物件權限管理程式使用者鑑別

在 WebSphere MQ MQI 用戶端連線上，可以使用安全結束程式來修改或建立在物件權限管理程式 (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，並在具有替代使用者權限的情況下，要求佇列管理程式在代表用戶端存取 WebSphere MQ 資源時使用此使用者 ID 進行權限檢查。

如果通道定義中的 PUTAUT 參數設為 CTX (或 z/OS 上的 ALTMCA)，則當 MCA 開啟目的地佇列時，會使用每則送入訊息的 *UserIdentifier* 欄位中的使用者 ID 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 124 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 WebSphere 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。

訊息機密性

若要維護機密性，請加密訊息。 視您的需求而定，在 WebSphere MQ 中有各種加密訊息的方法。

您選擇的 CipherSpec 可決定您具有的機密性層次。

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 WebSphere MQ 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。

如果您只需要在透過通道傳輸訊息時加密訊息，因為您在佇列管理程式上有足夠的安全保護，您可以使用 SSL 或 TLS，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。

如需 WebSphere MQ Advanced Message Security 的相關資訊，請參閱 第 52 頁的『規劃 Advanced Message Security』。第 20 頁的『IBM WebSphere MQ 支援 SSL 和 TLS』說明使用 SSL 和 TLS 搭配 WebSphere MQ。 在訊息加密中使用結束程式的相關說明，請參閱 第 189 頁的『在使用者結束程式中實作機密性』。

使用 SSL 或 TLS 連接兩個佇列管理程式

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。 您也必須取得並管理數位憑證。 在測試系統上，您可以使用自簽憑證或本端憑證管理中心 (CA) 發出的憑證。 在正式作業系統上，請勿使用自簽憑證。 如需相關資訊，請參閱 ..//zs14140_.dita。

如需建立及管理憑證的完整資訊，請參閱 第 95 頁的『在 UNIX, Linux, and Windows 系統上使用 SSL 或 TLS』。

此主題集合介紹設定 SSL 通訊所涉及的作業，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 WebSphere MQ 實作， SSL 或 TLS 伺服器一律會向用戶端要求憑證。

附註:

1. 在此環境定義中， SSL 用戶端是指起始信號交換的連線。
2. 如需進一步詳細資料，請參閱 名詞解釋 。

在 UNIX、 Linux 及 Windows 系統上，只有在 SSL 或 TLS 用戶端具有以正確 WebSphere MQ 格式標示的憑證時，它才會傳送憑證， `ibmwebspheremq` 後面接著變更為小寫的佇列管理程式名稱。 例如，若為 `QM1`，則為 `ibmwebspheremqqm1`。

WebSphere MQ 在標籤上使用 `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。 如需匿名連接佇列管理程式的相關資訊，亦即當 SSL 或 TLS 用戶端未傳送憑證時，請參閱 第 177 頁的『使用單向鑑別來連接兩個佇列管理程式』。

使用自簽憑證進行兩個佇列管理程式的交互鑑別

請遵循下列範例指示，使用自簽 SSL 或 TLS 憑證來實作兩個佇列管理程式之間的交互鑑別。

關於這項作業

測試情境:

- 您有兩個佇列管理程式 QM1 及 QM2，它們需要安全地通訊。 您需要在 QM1 與 QM2 之間執行交互鑑別。
- 您已決定使用自簽憑證來測試安全通訊。

產生的配置如下所示:

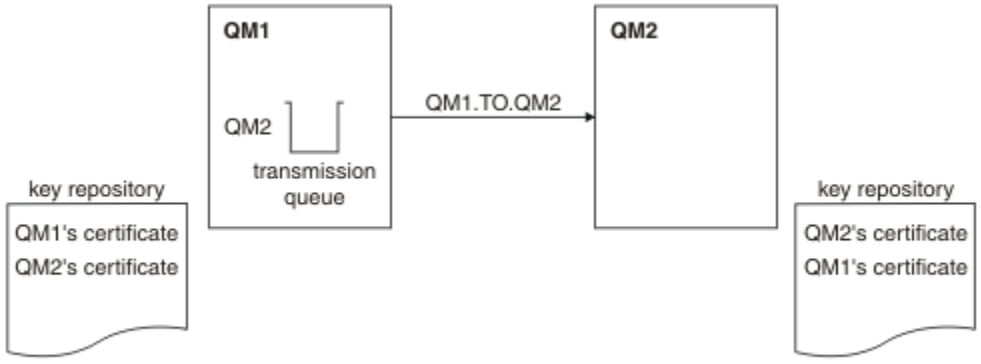


圖 *14:* 此作業產生的配置

在 第 173 頁的圖 14 中， QM1 的金鑰儲存庫包含 QM1 的憑證，以及來自 QM2 的公用憑證。 QM2 的金鑰儲存庫包含 QM2 的憑證，以及來自 QM1 的公用憑證。

程序

1. 根據作業系統，在每一個佇列管理程式上準備金鑰儲存庫:
   - 在 UNIX、 Linux 及 Windows 系統上。
2. 為每一個佇列管理程式建立自簽憑證:
   - 在 UNIX、 Linux 及 Windows 系統上。
3. 擷取每一個憑證的副本:
   - 在 UNIX、 Linux 及 Windows 系統上。
4. 使用 FTP 中所述。
5. 將夥伴憑證新增至每一個佇列管理程式的金鑰儲存庫:
   - 在 UNIX、 Linux 及 Windows 系統上。
6. 在 QM1 上，發出類似下列範例的指令來定義傳送端通道及相關聯的傳輸佇列:

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

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

   此範例使用 CipherSpec RC4_MD5。 通道每一端的 CipherSpecs 必須相同。
7. 在 QM2 上，發出類似下列範例的指令來定義接收端通道:

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

   通道必須與您在步驟 6 中定義的傳送端通道同名，並使用相同的 CipherSpec。
8. 啟動通道中所述。

結果

金鑰儲存庫和通道如 第 173 頁的圖 14 中所示建立。

下一步

使用 DISPLAY 指令來檢查作業是否已順利完成。 如果作業成功，則產生的輸出類似於下列範例中所示的輸出。

從佇列管理程式 QM1，輸入下列指令:

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

產生的輸出類似下列範例:

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

從佇列管理程式 QM2 中，輸入下列指令:

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

產生的輸出類似下列範例:

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

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

在每一種情況下， SSLPEER 的值必須符合在步驟 2 中建立之夥伴憑證中的 DN 值。 發證者名稱符合同層級名稱，因為憑證是自簽憑證。

SSLPEER 是選用的。 如果指定，則必須設定其值，以便容許夥伴憑證中的 DN (在步驟 2 中建立)。 如需使用 SSLPEER 的相關資訊，請參閱 WebSphere SSLPEER 值的 MQ 規則 。

使用 CA 簽章憑證來進行兩個佇列管理程式的交互鑑別

遵循下列範例指示，使用 CA 簽署的 SSL 或 TLS 憑證來實作兩個佇列管理程式之間的交互鑑別。

關於這項作業

測試情境:

- 您有兩個佇列管理程式，稱為 QMA 和 QMB ，需要安全通訊。 您需要在 QMA 與 QMB 之間執行交互鑑別。
- 未來您計劃在正式作業環境中使用此網路，因此您已決定從頭開始使用 CA 簽章憑證。

產生的配置如下所示:

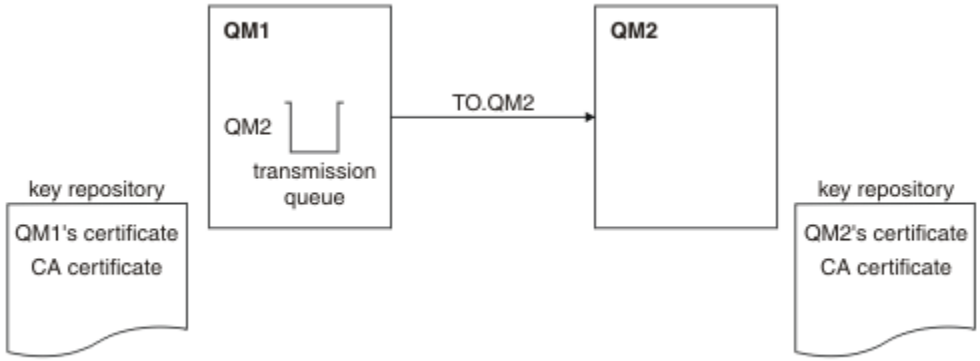


圖 *15:* 此作業產生的配置

在 第 175 頁的圖 15 中， QMA 的金鑰儲存庫包含 QMA 的憑證及 CA 憑證。 QMB 的金鑰儲存庫包含 QMB 的憑證及 CA 憑證。 在此範例中， QMA 的憑證和 QMB 的憑證都由相同的 CA 發出。 如果 QMA 的憑證和 QMB 的憑證是由不同的 CA 發出，則 QMA 和 QMB 的金鑰儲存庫必須同時包含這兩個 CA 憑證。

程序

1. 根據作業系統，在每一個佇列管理程式上準備金鑰儲存庫:
   - 在 UNIX、 Linux 及 Windows 系統上。
2. 要求每一個佇列管理程式的 CA 簽章憑證。
   您可以對兩個佇列管理程式使用不同的 CA。
   - 在 UNIX、 Linux 及 Windows 系統上。
3. 將憑證管理中心憑證新增至每一個佇列管理程式的金鑰儲存庫:
   如果佇列管理程式使用不同的憑證管理中心，則必須將每一個憑證管理中心的 CA 憑證新增至兩個金鑰儲存庫。

- 在 UNIX、 Linux 及 Windows 系統上。

4. 將 CA 簽章憑證新增至每一個佇列管理程式的金鑰儲存庫:
   - 在 UNIX、 Linux 及 Windows 系統上。
5. 在 QMA 上，發出類似下列範例的指令來定義傳送端通道及相關聯的傳輸佇列:

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

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

   此範例使用 CipherSpec RC4_MD5。 通道每一端的 CipherSpecs 必須相同。
6. 在 QMB 上，發出類似下列範例的指令來定義接收端通道:

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

   通道必須與您在步驟 6 中定義的傳送端通道同名，並使用相同的 CipherSpec。
7. 啟動通道:

結果

金鑰儲存庫和通道如 第 175 頁的圖 15 中所示建立。

下一步

使用 DISPLAY 指令來檢查作業是否已順利完成。 如果作業成功，產生的輸出會如下列範例所示。

從佇列管理程式 QMA 中，輸入下列指令:

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

產生的輸出類似下列範例:

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

從佇列管理程式 QMB 中，輸入下列指令:

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

產生的輸出類似下列範例:

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

在每一種情況下， SSLPEER 值必須符合在步驟 2 中建立之夥伴憑證中的「識別名稱 (DN)」值。 發證者名稱符合步驟 4 中所新增簽署個人憑證之 CA 憑證的主體 DN。

使用單向鑑別來連接兩個佇列管理程式

請遵循下列範例指示來修改具有交互鑑別的系統，以容許佇列管理程式使用單向鑑別來連接至另一個系統；亦即，當 SSL 或 TLS 用戶端未傳送憑證時。

關於這項作業

測試情境：

- 您的兩個佇列管理程式 (QM1 及 QM2) 已設定為在 第 175 頁的『使用 CA 簽章憑證來進行兩個佇列管理程式的交互鑑別』 中。
- 您想要變更 QM1 ，讓它使用單向鑑別連接至 QM2。

產生的配置如下所示：

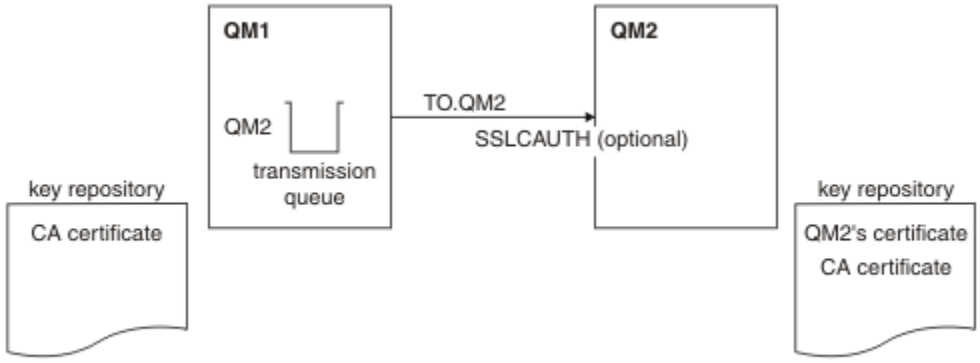


圖 *16:* 容許單向鑑別的佇列管理程式

程序

1. 根據作業系統，從其金鑰儲存庫中移除 QM1 的個人憑證：
   - 在 UNIX、 Linux 及 Windows 系統上。 憑證標示如下：
     - `ibmwebspheremq` 後面接著佇列管理程式的名稱，並轉換成小寫。 例如，若為 `QM1` ，則為 `ibmwebspheremqqm1`。
2. 選擇性的: 在 QM1 上，如果先前已執行任何 SSL 或 TLS 通道，請重新整理 SSL 或 TLS 環境中所述。
3. 容許接收端中所述。

結果

金鑰儲存庫和通道已變更，如 第 177 頁的圖 16 中所示

下一步

如果傳送端通道正在執行中，且您發出 REFRESH SECURITY TYPE (SSL) 指令 (在步驟 2 中) ，則通道會自動重新啟動。 如果傳送端通道不在執行中，請啟動它。

在通道的伺服器端，通道狀態顯示畫面上存在對等節點名稱參數值表示已傳送用戶端憑證。

請發出一些 DISPLAY 指令來驗證作業已順利完成。 如果作業成功，則產生的輸出類似於下列範例中所示的輸出：

從 QM1 佇列管理程式中，輸入下列指令：

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

產生的輸出將類似於下列範例：

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(TO.QMB) SSLPEER
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QM2)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMB,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QM2)
```

從 QM2 佇列管理程式中，輸入下列指令:

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

產生的輸出將類似於下列範例:

```
DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QMA)                            SSLCERTI( )
   SSLPEER( )                              STATUS(RUNNING)
   SUBSTATE(RECEIVE)                       XMITQ( )
```

在 QM2 上， SSLPEER 欄位是空的，表示 QM1 未傳送憑證。 在 QM1 上， SSLPEER 的值符合 QM2 個人憑證中 DN 的值。

將用戶端安全連接至佇列管理程式

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。 您也必須取得並管理數位憑證。 在測試系統上，您可以使用自簽憑證或本端憑證管理中心 (CA) 發出的憑證。 在正式作業系統上，請勿使用自簽憑證。 如需相關資訊，請參閱 ..//zs14140_.dita。

如需建立及管理憑證的完整資訊，請參閱 第 95 頁的『在 UNIX, Linux, and Windows 系統上使用 SSL 或 TLS』。

此主題集合介紹設定 SSL 通訊所涉及的作業，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 WebSphere MQ 實作， SSL 或 TLS 伺服器一律會向用戶端要求憑證。

在 UNIX, Linux, and Windows 系統上，只有在 SSL 或 TLS 用戶端具有以正確 WebSphere MQ 格式標示的憑證時，它才會傳送憑證，即 `ibmwebspheremq` 後面接著您的登入使用者 ID 變更為小寫，例如 `ibmwebspheremqmyuserid`。

WebSphere MQ 在標籤上使用 `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。 如需匿名連接佇列管理程式的相關資訊，請參閱 第 181 頁的『以匿名方式將用戶端連接至佇列管理程式』。

使用自簽憑證進行用戶端及佇列管理程式的交互鑑別

遵循下列範例指示，使用自簽 SSL 或 TLS 憑證來實作用戶端與佇列管理程式之間的交互鑑別。

關於這項作業

測試情境:

- 您具有需要安全通訊的用戶端 C1 及佇列管理程式 QM1。 您需要在 C1 與 QM1 之間執行交互鑑別。
- 您已決定使用自簽憑證來測試安全通訊。

DCM on IBM i 不支援自簽憑證，因此這項作業不適用於 IBM i 系統。

產生的配置如下所示:

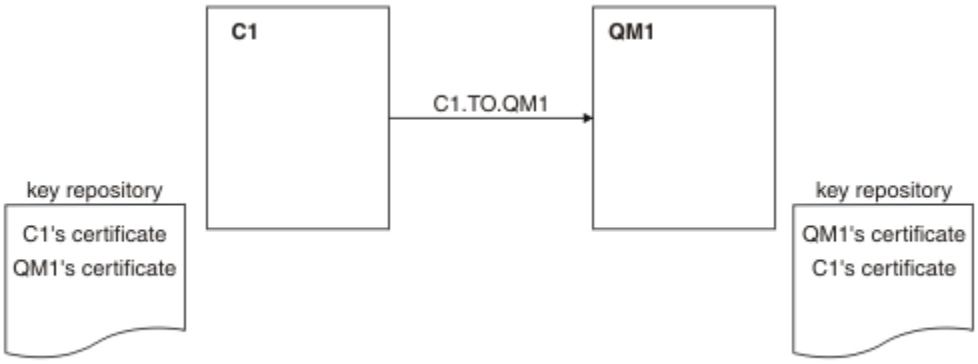


圖 *17:* 此作業產生的配置

在 第 179 頁的圖 17 中， QM1 的金鑰儲存庫包含 QM1 的憑證及來自 C1 的公用憑證。 C1 的金鑰儲存庫包含 C1 的憑證，以及來自 QM1 的公用憑證。

程序

1. 根據作業系統，在用戶端及佇列管理程式上準備金鑰儲存庫:
   - 在 UNIX、 Linux 及 Windows 系統上。
2. 為用戶端及佇列管理程式建立自簽憑證:
   - 在 UNIX、 Linux 及 Windows 系統上。
3. 擷取每一個憑證的副本:
   - 在 UNIX、 Linux 及 Windows 系統上。
4. 使用 FTP 中所述。
5. 將夥伴憑證新增至用戶端及佇列管理程式的金鑰儲存庫:
   - 在 UNIX、 Linux 及 Windows 系統上。
6. 在佇列管理程式上發出 REFRESH SECURITY TYPE (SSL) 指令。
7. 以下列其中一種方式來定義用戶端連線通道:
   - 在 C1 上搭配使用 MQCONNX 呼叫與 MQSCO 結構，如 在 WebSphere MQ MQI 用戶端上建立用戶端連線通道中所述。
   - 使用用戶端通道定義表，如 在伺服器上建立伺服器連線及用戶端連線定義 中所述。
8. 在 QM1 上，發出類似下列範例的指令來定義伺服器連線通道:

   ```
   DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC4_MD5_US)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
   ```

   通道必須與您在步驟 6 中定義的用戶端連線通道同名，並使用相同的 CipherSpec。

結果

金鑰儲存庫和通道如 第 179 頁的圖 17 中所示建立。

下一步

使用 DISPLAY 指令來檢查作業是否已順利完成。 如果作業成功，則產生的輸出類似於下列範例中所示的輸出。

從佇列管理程式 QM1，輸入下列指令:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

產生的輸出類似下列範例:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                       CURRENT
   SSLCERTI("CN=QM1,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5E:02,CN=QM2,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                           SUBSTATE(RECEIVE)
```

設定通道定義的 SSLPEER 過濾器屬性是選用的。 如果設定通道定義 SSLPEER，則其值必須符合在步驟 2 中建立的夥伴憑證中的主體 DN。 成功連線之後， DISPLAY CHSTATUS 輸出中的 SSLPEER 欄位會顯示遠端用戶端憑證的主體 DN。

使用 CA 簽章憑證進行用戶端及佇列管理程式的交互鑑別

請遵循下列範例指示，使用 CA 簽署的 SSL 或 TLS 憑證來實作用戶端與佇列管理程式之間的交互鑑別。

關於這項作業

測試情境:

- 您具有需要安全通訊的用戶端 C1 及佇列管理程式 QM1。 您需要在 C1 與 QM1 之間執行交互鑑別。
- 未來您計劃在正式作業環境中使用此網路，因此您已決定從頭開始使用 CA 簽章憑證。

產生的配置如下所示:

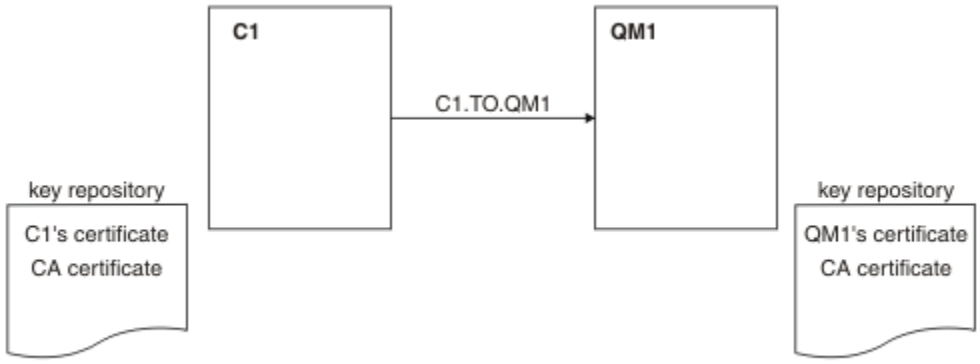


圖 *18:* 此作業產生的配置

在 第 180 頁的圖 18 中， C1 的金鑰儲存庫包含 C1 的憑證及 CA 憑證。 QM1 的金鑰儲存庫包含 QM1 的憑證及 CA 憑證。 在此範例中， C1 的憑證和 QM1 的憑證都由相同的 CA 發出。 如果 C1 的憑證和 QM1 的憑證是由不同的 CA 發出，則 C1 和 QM1 的金鑰儲存庫必須同時包含這兩個 CA 憑證。

程序

1. 根據作業系統，在用戶端及佇列管理程式上準備金鑰儲存庫:
   - 在 UNIX、 Linux 及 Windows 系統上。

2. 要求用戶端及佇列管理程式的 CA 簽章憑證。
   您可以對用戶端及佇列管理程式使用不同的 CA。
   - 在 UNIX、 Linux 及 Windows 系統上。
3. 將憑證管理中心憑證新增至用戶端及佇列管理程式的金鑰儲存庫。
   如果用戶端及佇列管理程式使用不同的「憑證管理中心」，則必須將每一個「憑證管理中心」的 CA 憑證新增至這兩個金鑰儲存庫。
   - 在 UNIX、 Linux 及 Windows 系統上。
4. 將 CA 簽章憑證新增至用戶端及佇列管理程式的金鑰儲存庫:
   - 在 UNIX、 Linux 及 Windows 系統上。
5. 以下列其中一種方式來定義用戶端連線通道:
   - 在 C1 上搭配使用 MQCONNX 呼叫與 MQSCO 結構，如 在 WebSphere MQ MQI 用戶端上建立用戶端連線通道中所述。
   - 使用用戶端通道定義表，如 在伺服器上建立伺服器連線及用戶端連線定義 中所述。
6. 在 QM1 上，發出類似下列範例的指令來定義伺服器連線通道:

   ```
   DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC2_MD5_EXPORT)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
   ```

   通道必須與您在步驟 6 中定義的用戶端連線通道同名，並使用相同的 CipherSpec。

結果
金鑰儲存庫和通道如 第 180 頁的圖 18 中所示建立。

下一步
使用 DISPLAY 指令來檢查作業是否已順利完成。 如果作業成功，產生的輸出會如下列範例所示。

從佇列管理程式 QM1 中，輸入下列指令:

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

產生的輸出類似下列範例:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                     CURRENT
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMA,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

DISPLAY CHSTATUS 輸出中的 SSLPEER 欄位會顯示在步驟 2 中建立之遠端用戶端憑證的主體 DN。 發證者名稱符合步驟 4 中所新增簽署個人憑證之 CA 憑證的主體 DN。

以匿名方式將用戶端連接至佇列管理程式
遵循這些範例指示來修改具有交互鑑別的系統，以容許佇列管理程式匿名連接至另一個。

關於這項作業
測試情境:

- 您的佇列管理程式及用戶端 (QM1 及 C1) 已設定在 第 180 頁的『使用 CA 簽章憑證進行用戶端及佇列管理程式的交互鑑別』中。
- 您想要變更 C1 ，以便它以匿名方式連接至 QM1。

產生的配置如下所示:

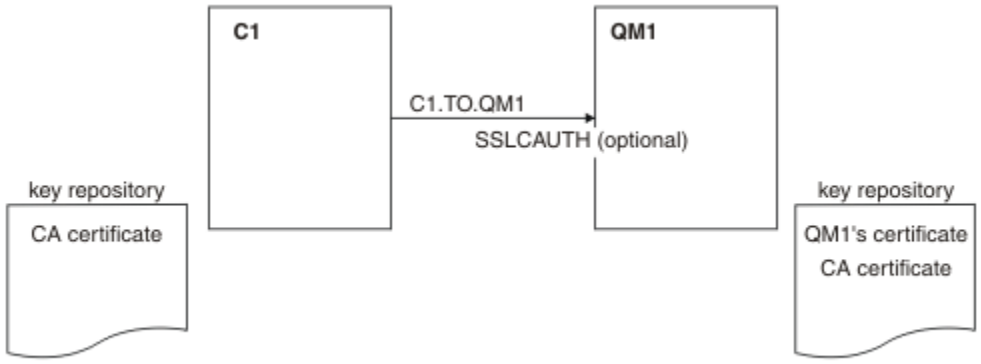


圖 *19:* 容許匿名連線的用戶端及佇列管理程式

程序

1. 根據作業系統，從 C1 的金鑰儲存庫中移除個人憑證:
   - 在 UNIX、 Linux 及 Windows 系統上。 憑證標示如下:
     - `ibmwebspheremq` 後面接著您的登入使用者 ID 會轉換成小寫，例如 `ibmwebspheremqmyuserid`。
2. 重新啟動用戶端應用程式，或讓用戶端應用程式關閉並重新開啟所有 SSL 或 TLS 連線。
3. 發出下列指令，在佇列管理程式上容許匿名連線:

```
ALTER CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) SSLCAUTH(OPTIONAL)
```

結果

金鑰儲存庫和通道已變更，如 第 182 頁的圖 19 中所示

下一步

在通道的伺服器端，通道狀態顯示畫面上存在對等節點名稱參數值表示已傳送用戶端憑證。

請發出一些 DISPLAY 指令來驗證作業已順利完成。 如果作業成功，則產生的輸出類似於下列範例中所示的輸出:

從佇列管理程式 QM1，輸入下列指令:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

產生的輸出將類似於下列範例:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                     CURRENT
   SSLCERTI( )                             SSLPEER( )
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

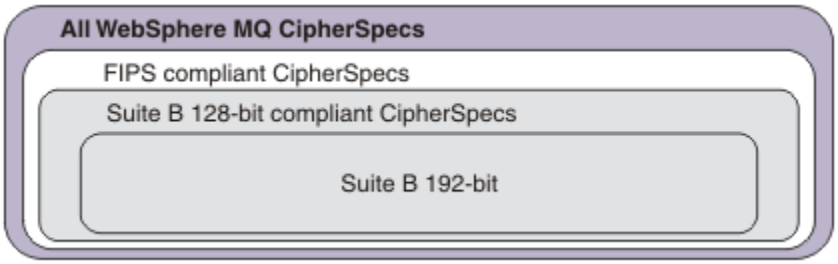
SSLCERTI 和 SSLPEER 欄位是空的，表示 C1 未傳送憑證。

指定 CipherSpecs

在 **DEFINE CHANNEL** MQSC 指令或 **ALTER CHANNEL** MQSC 指令中使用 **SSLCIPH** 參數，以指定 CipherSpec 。

您可以與 IBM WebSphere MQ 搭配使用的部分 CipherSpecs 符合 FIPS 標準。 其他 (例如 NULL_MD5) 則不是。 同樣地，部分符合 FIPS 標準的 CipherSpecs 也符合套組 B 標準，但其他不符合套組 B 標準。 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:



下表列出可與 IBM WebSphere MQ SSL 及 TLS 支援搭配使用的密碼規格。 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。 SSL 信號交換期間使用的金鑰大小是儲存在憑證中的大小，除非如表格中所述由 CipherSpec 決定。

CipherSpec 名稱	使用的通訊協定	MAC 演算法	加密演算法	加密位元	FIPS[1]	套組 B 128 位元	套組 B 192 位元
NULL_MD5 [a]	SSL 3.0	MD5	無	0	否	否	否
NULL_SHA [a]	SSL 3.0	SHA-1	無	0	否	否	否
RC4_MD5_EXPORT [2] [a]	SSL 3.0	MD5	RC4	40	否	否	否
RC4_MD5_US [a]	SSL 3.0	MD5	RC4	128	否	否	否
RC4_SHA_US [a]	SSL 3.0	SHA-1	RC4	128	否	否	否
RC2_MD5_EXPORT [2] [a]	SSL 3.0	MD5	RC2	40	否	否	否
DES_SHA_EXPORT [2] [a]	SSL 3.0	SHA-1	DES	56	否	否	否
RC4_56_SHA_EXPORT1024 [3] [b]	SSL 3.0	SHA-1	RC4	56	否	否	否
DES_SHA_EXPORT1024 [3] [b]	SSL 3.0	SHA-1	DES	56	否	否	否
TLS_RSA_WITH_AES_128_CBC_SHA [a]	TLS 1.0	SHA-1	AES	128	是	否	否
TLS_RSA_WITH_AES_256_CBC_SHA [4] [a]	TLS 1.0	SHA-1	AES	256	是	否	否
TLS_RSA_WITH_DES_CBC_SHA [a]	TLS 1.0	SHA-1	DES	56	否 [5]	否	否
FIPS_WITH_DES_CBC_SHA [b]	SSL 3.0	SHA-1	DES	56	否 [6]	否	否
TLS_RSA_WITH_AES_128_GCM_SHA256 [b]	TLS 1.2	AEAD AES-128 GCM	AES	128	是	否	否
TLS_RSA_WITH_AES_256_GCM_SHA384 [b]	TLS 1.2	AEAD AES-256 GCM	AES	256	是	否	否
TLS_RSA_WITH_AES_128_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	128	是	否	否
TLS_RSA_WITH_AES_256_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	256	是	否	否
ECDHE_ECDSA_RC4_128_SHA256 [b]	TLS 1.2	SHA-1	RC4	128	否	否	否
ECDHE_RSA_RC4_128_SHA256 [b]	TLS 1.2	SHA_1	RC4	128	否	否	否

CipherSpec 名稱	使用的通訊協定	MAC 演算法	加密演算法	加密位元	FIPS[1]	套組 B 128 位元	套組 B 192 位元
ECDHE_ECDSA_AES_128_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	128	是	否	否
ECDHE_ECDSA_AES_256_CBC_SHA384 [b]	TLS 1.2	SHA-384	AES	256	是	否	否
ECDHE_RSA_AES_128_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	128	是	否	否
ECDHE_RSA_AES_256_CBC_SHA384 [b]	TLS 1.2	SHA-384	AES	256	是	否	否
ECDHE_ECDSA_AES_128_GCM_SHA256 [b]	TLS 1.2	AEAD AES-128 GCM	AES	128	是	是	否
ECDHE_ECDSA_AES_256_GCM_SHA384 [b]	TLS 1.2	AEAD AES-256 GCM	AES	256	是	否	是
ECDHE_RSA_AES_128_GCM_SHA256 [b]	TLS 1.2	AEAD AES-128 GCM	AES	128	是	否	否
ECDHE_RSA_AES_256_GCM_SHA384 [b]	TLS 1.2	AEAD AES-256 GCM	AES	256	是	否	否
TLS_RSA_WITH_NULL_SHA256 [b]	TLS 1.2	SHA-256	無	0	否	否	否
ECDHE_RSA_NULL_SHA256 [b]	TLS 1.2	SHA-1	無	0	否	否	否
ECDHE_ECDSA_NULL_SHA256 [b]	TLS 1.2	SHA-1	無	0	否	否	否
TLS_RSA_WITH_NULL_NULL [b]	TLS 1.2	無	無	0	否	否	否
TLS_RSA_WITH_RC4_128_SHA256 [b]	TLS 1.2	SHA-1	RC4	128	否	否	否

附註:

1. 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)。
2. 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。
3. 信號交換金鑰大小是 1024 位元。
4. 這個 CipherSpec 無法用來保護從「WebSphere MQ 探險家」到佇列管理程式的連線，除非將適當的未限定原則檔套用至「探險家」所使用的 JRE。
5. 在 2007 年 5 月 19 日之前，這個 CipherSpec 已經過 FIPS 140-2 認證。
6. 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。
7. 在連線因 AMQ9288 錯誤而終止之前，這個 CipherSpec 可用來傳送最多 32 GB 的資料。 若要避免發生此錯誤，請避免使用三重 DES 演算法，或在使用這個 CipherSpec 時啟用秘密金鑰重設。

平台支援:

- a 可在所有受支援平台上使用。
- b 僅適用於 UNIX, Linux, and Windows 平台。

相關概念

第 29 頁的『IBM WebSphere 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 WebSphere 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

已淘汰 CipherSpecs

必要的話，您可以與 WebSphere MQ 搭配使用的已淘汰 CipherSpecs 清單。

如需如何啟用已淘汰 CipherSpecs 的相關資訊，請參閱 第 32 頁的『IBM WebSphere MQ 中支援的 CipherSpec 值』。

下表列出您可以與 WebSphere MQ TLS 支援搭配使用的已淘汰 CipherSpecs：

平台支援 第 186 頁的『1』	CipherSpec 名稱	使用的通訊協定	資料完整性	加密演算法	加密位元	FIPS 第 186 頁的『2』	套組 B	淘汰時更新
全部	DES_SHA_EXPORT 第 186 頁的『3』	SSL 3.0	SHA-1	DES	56	否	否	7.5.0.6
Windows UNIX Linux	DES_SHA_EXPORT1024 第 186 頁的『4』	SSL 3.0	SHA-1	DES	56	否	否	7.5.0.6
Windows UNIX Linux	FIPS_WITH_DES_CBC_SHA	SSL 3.0	SHA-1	DES	56	否 第 186 頁的『6』	否	7.5.0.6
Windows UNIX Linux	FIPS_WITH_3DES_EDE_CBC_SHA	SSL 3.0	SHA-1	3DES	168	否 第 186 頁的『7』	否	7.5.0.8
全部	NULL_MD5	SSL 3.0	MD5	無	0	否	否	7.5.0.6
全部	NULL_SHA	SSL 3.0	SHA-1	無	0	否	否	7.5.0.6
全部	RC2_MD5_EXPORT 第 186 頁的『3』	SSL 3.0	MD5	RC2	40	否	否	7.5.0.7
全部	RC4_MD5_EXPORT 第 186 頁的『3』	SSL 3.0	MD5	RC4	40	否	否	7.5.0.7
全部	RC4_MD5_US	SSL 3.0	MD5	RC4	128	否	否	7.5.0.7
全部	RC4_SHA_US	SSL 3.0	SHA-1	RC4	128	否	否	7.5.0.7
Windows UNIX Linux	RC4_56_SHA_EXPORT1024 第 186 頁的『4』	SSL 3.0	SHA-1	RC4	56	否	否	7.5.0.7
全部	TRIPLE_DES_SHA_US	SSL 3.0	SHA-1	3DES	168	否	否	7.5.0.8
全部	TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	SHA-1	DES	56	否 第 186 頁的『5』	否	7.5.0.6
Windows UNIX Linux	ECDHE_ECDSA_NULL_SHA256	TLS 1.2	SHA-1	無	0	否	否	7.5.0.6

平台支援 第 186 頁的『1』	CipherSpec 名稱	使用的通訊協定	資料完整性	加密演算法	加密位元	FIPS 第 186 頁的『2』	套組 B	淘汰時更新
Windows UNIX Linux	ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	SHA-1	RC4	128	否	否	7.5.0.7
Windows UNIX Linux	ECDHE_RSA_NULL_SHA256	TLS 1.2	SHA-1	無	0	否	否	7.5.0.6
Windows UNIX Linux	ECDHE_RSA_RC4_128_SHA256	TLS 1.2	SHA-1	RC4	128	否	否	7.5.0.7
Windows UNIX Linux	TLS_RSA_WITH_NULL_NULL	TLS 1.2	無	無	0	否	否	7.5.0.6
全部	TLS_RSA_WITH_NULL_SHA256	TLS 1.2	SHA-256	無	0	否	否	7.5.0.6
Windows UNIX Linux	TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	SHA-1	RC4	128	否	否	7.5.0.7
全部	TLS_RSA_WITH_3DES_EDE_CBC_SHA 第 186 頁的『8』	TLS 1.0	SHA-1	3DES	168	是	否	7.5.0.8
Windows UNIX Linux	ECDHE_ECDSA_3DES_EDE_CBC_SHA256 第 186 頁的『8』	TLS 1.2	SHA-1	3DES	168	是	否	7.5.0.8
Windows UNIX Linux	ECDHE_RSA_3DES_EDE_CBC_SHA256 第 186 頁的『8』	TLS 1.2	SHA-1	3DES	168	是	否	7.5.0.8

附註:

1. 如果未指出特定平台，CipherSpec 可使用於所有平台。
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. 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。
4. 信號交換金鑰大小是 1024 位元。
5. 在 2007 年 5 月 19 日之前，這個 CipherSpec 已經過 FIPS 140-2 認證。
6. 在 2007 年 5 月 19 日之前，這個 CipherSpec 已經過 FIPS 140-2 認證。 FIPS_WITH_DES_CBC_SHA 名稱是歷史名稱，反映出這個 CipherSpec 先前（但已不再）符合 FIPS 標準的事實。 這個 CipherSpec 已淘汰，不建議使用它。
7. FIPS_WITH_3DES_EDE_CBC_SHA 名稱是歷史名稱，反映出這個 CipherSpec 先前（但已不再）符合 FIPS 標準的事實。 這個 CipherSpec 的用法已淘汰。
8. 在連線因 AMQ9288 錯誤而終止之前，這個 CipherSpec 可用來傳送最多 32 GB 的資料。 若要避免發生此錯誤，請避免使用三重 DES 演算法，或在使用這個 CipherSpec 時啟用秘密金鑰重設。

使用 IBM WebSphere MQ Explorer 取得 CipherSpecs 的相關資訊

您可以使用 IBM WebSphere MQ Explorer 來顯示 CipherSpecs 的說明。

使用下列程序來取得 第 182 頁的『指定 CipherSpecs』中 CipherSpecs 的相關資訊:

1. 開啟「 **IBM WebSphere MQ 探險家** 」， 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。

用於指定 CipherSpecs 的替代方案

對於作業系統提供 SSL 支援的那些平台，您的系統可能支援新的 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。

註: 本節不適用於 UNIX、 Linux 或 Windows 系統，因為 CipherSpecs 隨附於 WebSphere MQ 產品，因此新的 CipherSpecs 在出貨之後不會變成可用。

對於作業系統提供 SSL 支援的那些平台，您的系統可能支援 第 182 頁的『指定 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 必須 對應於 SSL CipherSpec ，它既有效又受系統執行的 SSL 版本支援。

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，請參閱適當的產品說明文件 (搜尋 IBM i 產品說明文件中的 *cipher_spec*)。

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

代表十六進位值的兩個字元字串。 十六進位碼對應於 SSL 通訊協定中定義的值。

如需相關資訊，請參閱 *z/OS Cryptographic Services System SSL Programming*(SC24-5901) 的 API 參考資料章節中 gsk_environment_open () 的說明，其中列出所有支援的 SSL V3.0 及 TLS V1.0 密碼規格 (格式為 2 位數十六進位碼)。

WebSphere MQ 叢集的考量

使用 WebSphere MQ 叢集，最安全的是在 第 182 頁的『指定 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 209 頁的『SSL 和叢集』。

指定 IBM WebSphere MQ MQI 用戶端的 CipherSpec

您有三個選項可用來指定 IBM WebSphere MQ MQI 用戶端的 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 WebSphere MQ classes for Java 和 IBM WebSphere MQ classes for JMS 來指定 CipherSuite

IBM WebSphere MQ classes for Java 和 IBM WebSphere MQ classes for JMS 指定不同於其他平台的 CipherSuites 。

如需使用 IBM WebSphere MQ classes for Java 來指定 CipherSuite 的相關資訊，請參閱 Secure Sockets Layer (SSL) 支援。

如需使用 IBM WebSphere MQ classes for JMS 來指定 CipherSuite 的相關資訊，請參閱 Using Secure Sockets Layer (SSL) with WebSphere MQ classes for JMS。

重設 SSL 和 TLS 秘密金鑰

IBM WebSphere MQ 支援重設佇列管理程式及用戶端上的秘密金鑰。

當指定的已加密資料位元組數已透過通道流動時，或在通道閒置一段時間之後，會重設秘密金鑰。

金鑰重設值一律由 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。

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. 在 MQCONNX 呼叫 MQSCO 結構中使用 KeyReset 計數欄位
2. 使用環境變數 MQSSLRESET
3. 透過在 MQI 用戶端配置檔中設定 SSLKeyReset 計數屬性

這些變數可以設為 0 到 999 999 999 範圍內的整數，代表在重新協議 SSL 或 TLS 秘密金鑰之前，在 SSL 或 TLS 交談內所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 SSL 或 TLS 秘密金鑰。 如果您指定 1 位元組到 32 KB 範圍內的 SSL 或 TLS 秘密金鑰重設計數，則 SSL 或 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 SSL 或 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。

如需 MQSCO 結構的完整資料，請參閱 KeyReset 計數 (MQLONG)。 如需 MQSSLRESET 的完整資料，請參閱 MQSSLRESET。 如需用戶端配置檔中使用 SSL 或 TLS 的相關資訊，請參閱 用戶端配置檔的 SSL 段落。

Java

對於 IBM WebSphere 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:

- 透過在 MQEnvironment 類別中設定 sslReset 計數欄位。
- 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。

sslReset 計數欄位或環境內容 MQC.SSL_RESET_COUNT_PROPERTY 的值代表在重新協議秘密金鑰之前，WebSphere MQ 類別 for Java 用戶端程式碼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 WebSphere MQ for Java 用戶端類別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 WebSphere 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 WebSphere MQ classes for JMS 所傳送及接收的控制資訊。 例如，若要配置 ConnectionFactory 物件 (可用來透過啟用 SSL 或 TLS 的 MQI 通道建立連線，且具有在 4 MB 資料傳送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 未受管理用戶端，整數內容 SSLKeyReset 計數指出在重新協議秘密金鑰之前，在 SSL 或 TLS 交談內傳送及接收的未加密位元組數。

如需在 IBM WebSphere 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。

XMS .NET

若為 XMS .NET 未受管理用戶端，請參閱 IBM WebSphere MQ 佇列管理程式的安全連線。

相關參考
ALTER QMGR
DISPLAY QMGR

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 189 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。

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 SSL 或 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 WebSphere MQ Advanced 訊息服務 (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。

使用 SSL 或 TLS 連接兩個佇列管理程式

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。 您也必須取得並管理數位憑證。 在測試系統上，您可以使用自簽憑證或本端憑證管理中心 (CA) 發出的憑證。 在正式作業系統上，請勿使用自簽憑證。 如需相關資訊，請參閱 ../zs14140_.dita。

如需建立及管理憑證的完整資訊，請參閱 第 95 頁的『在 UNIX, Linux, and Windows 系統上使用 SSL 或 TLS』。

此主題集合介紹設定 SSL 通訊所涉及的作業，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 WebSphere MQ 實作， SSL 或 TLS 伺服器一律會向用戶端要求憑證。

附註:

1. 在此環境定義中， SSL 用戶端是指起始信號交換的連線。
2. 如需進一步詳細資料，請參閱 名詞解釋 。

在 UNIX、 Linux 及 Windows 系統上，只有在 SSL 或 TLS 用戶端具有以正確 WebSphere MQ 格式標示的憑證時，它才會傳送憑證， `ibmwebspheremq` 後面接著變更為小寫的佇列管理程式名稱。 例如，若為 `QM1`，則為 `ibmwebspheremqqm1`。

WebSphere MQ 在標籤上使用 `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。 如需匿名連接佇列管理程式的相關資訊，亦即當 SSL 或 TLS 用戶端未傳送憑證時，請參閱 第 177 頁的『使用單向鑑別來連接兩個佇列管理程式』。

數位憑證標籤，瞭解需求

設定 SSL 及 TLS 以使用數位憑證時，您可能必須遵循特定的標籤需求，視使用的平台及用來連接的方法而定。

關於這項作業

何謂憑證標籤?

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 主旨識別名稱 或 主旨通用名稱 欄位不同。 請注意， 主旨識別名稱 和 主旨通用名稱 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。

如何使用憑證標籤?

IBM WebSphere MQ 使用憑證標籤來尋找在 SSL 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。

憑證標籤遵循命名慣例; 您需要確保使用與所使用平台相對應的正確標籤命名慣例。

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

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 WebSphere MQ 實作， SSL 或 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， 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，且在 SSLCAUTH 參數設為 REQUIRED 或設定 SSLPEER 參數值的情況下定義作為 SSL 或 TLS 伺服器的通道結尾，則鑑別會失敗。

如需使用單向鑑別連接佇列管理程式的相關資訊，亦即當 SSL 或 TLS 用戶端未傳送憑證時，請參閱 第 177 頁的『使用單向鑑別來連接兩個佇列管理程式』。

、 *UNIX, Linux, and Windows* 系統

關於這項作業

在 、 UNIX, Linux, and Windows 系統上，只有在伺服器找到以正確 IBM WebSphere MQ 格式標示的憑證時， SSL 或 TLS 伺服器才會將憑證傳送至用戶端。 在這些系統上，正確的格式為 ibmwebspheremq，後面接著佇列管理程式的名稱變更為小寫。

例如，對於名為 QM1 的佇列管理程式，憑證標籤需求為:

```
ibmwebspheremqqm1
```

如果在佇列管理程式的金鑰儲存庫中找不到符合正確大小寫及格式之必要標籤的憑證，則會發生錯誤，且 SSL 或 TLS 信號交換會失敗。

IBM WebSphere MQ 用戶端

關於這項作業

從 IBM WebSphere MQ 用戶端應用程式連接時，只有在 SSL 或 TLS 用戶端有一個憑證具有 ibmwebspheremq 格式的標籤，後面接著執行用戶端應用程式程序之使用者的使用者名稱時，才會傳送憑證。

例如，對於使用者名稱 wasadmin，憑證標籤需求如下所示，並轉換為小寫:

```
   ibmwebspheremqwasadmin
```

上述標籤需求適用於 Message Service Client for C、或 C++ 及 .NET。

IBM WebSphere MQ Java 或 *IBM WebSphere MQ JMS* 用戶端

關於這項作業

IBM WebSphere MQ Java 或 IBM WebSphere MQ JMS 用戶端會在 SSL 或 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，因此不符合憑證標籤需求。

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。

如果要進行疑難排解，或更充分地瞭解 IBM WebSphere MQ Java 用戶端應用程式與特定 JSSE 提供者一起執行的信號交換，您可以設定下列指令來啟用除錯:

```
javax.net.debug=ssl
```

在 JVM 環境中。

您可以在指令行上使用 -Djavax.net.debug=ssl ，或在應用程式內設定變數，或透過配置來設定變數。

相關概念

第 112 頁的『將個人憑證匯入 UNIX, Linux, and Windows 系統上的金鑰儲存庫』
請遵循此程序來匯入個人憑證

使用自簽憑證進行兩個佇列管理程式的交互鑑別

請遵循下列範例指示，使用自簽 SSL 或 TLS 憑證來實作兩個佇列管理程式之間的交互鑑別。

關於這項作業

測試情境:

- 您有兩個佇列管理程式 QM1 及 QM2，它們需要安全地通訊。 您需要在 QM1 與 QM2 之間執行交互鑑別。
- 您已決定使用自簽憑證來測試安全通訊。

產生的配置如下所示:

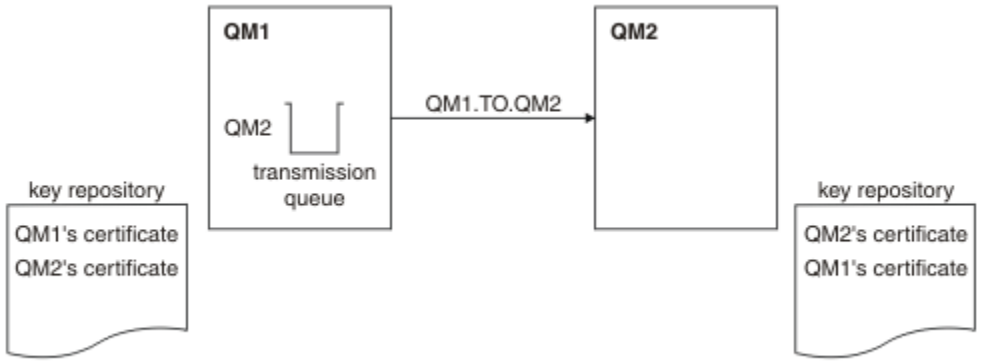


圖 *20:* 此作業產生的配置

在 第 173 頁的圖 14 中， QM1 的金鑰儲存庫包含 QM1 的憑證，以及來自 QM2 的公用憑證。 QM2 的金鑰儲存庫包含 QM2 的憑證，以及來自 QM1 的公用憑證。

程序

1. 根據作業系統，在每一個佇列管理程式上準備金鑰儲存庫:
   - 在 UNIX、 Linux 及 Windows 系統上。
2. 為每一個佇列管理程式建立自簽憑證:
   - 在 UNIX、 Linux 及 Windows 系統上。
3. 擷取每一個憑證的副本:
   - 在 UNIX、 Linux 及 Windows 系統上。
4. 使用 FTP 中所述。
5. 將夥伴憑證新增至每一個佇列管理程式的金鑰儲存庫:
   - 在 UNIX、 Linux 及 Windows 系統上。
6. 在 QM1 上，發出類似下列範例的指令來定義傳送端通道及相關聯的傳輸佇列:

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

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

   此範例使用 CipherSpec RC4_MD5。 通道每一端的 CipherSpecs 必須相同。
7. 在 QM2 上，發出類似下列範例的指令來定義接收端通道:

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

   通道必須與您在步驟 6 中定義的傳送端通道同名，並使用相同的 CipherSpec。
8. 啟動通道中所述。

結果

金鑰儲存庫和通道如 第 173 頁的圖 14 中所示建立。

下一步

使用 DISPLAY 指令來檢查作業是否已順利完成。 如果作業成功，則產生的輸出類似於下列範例中所示的輸出。

從佇列管理程式 QM1，輸入下列指令:

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

產生的輸出類似下列範例:

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

從佇列管理程式 QM2 中，輸入下列指令:

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

產生的輸出類似下列範例:

```
DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(QM2.TO.QM1)                     CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QM1)
   SSLCERTI("CN=QM1,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QM1,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
   XMITQ( )
```

在每一種情況下， SSLPEER 的值必須符合在步驟 2 中建立之夥伴憑證中的 DN 值。 發證者名稱符合同層級名稱，因為憑證是自簽憑證。

SSLPEER 是選用的。 如果指定，則必須設定其值，以便容許夥伴憑證中的 DN (在步驟 2 中建立)。 如需使用 SSLPEER 的相關資訊，請參閱 WebSphere SSLPEER 值的 MQ 規則 。

使用 CA 簽章憑證來進行兩個佇列管理程式的交互鑑別

遵循下列範例指示，使用 CA 簽署的 SSL 或 TLS 憑證來實作兩個佇列管理程式之間的交互鑑別。

關於這項作業

測試情境:

- 您有兩個佇列管理程式，稱為 QMA 和 QMB ，需要安全通訊。 您需要在 QMA 與 QMB 之間執行交互鑑別。
- 未來您計劃在正式作業環境中使用此網路，因此您已決定從頭開始使用 CA 簽章憑證。

產生的配置如下所示:

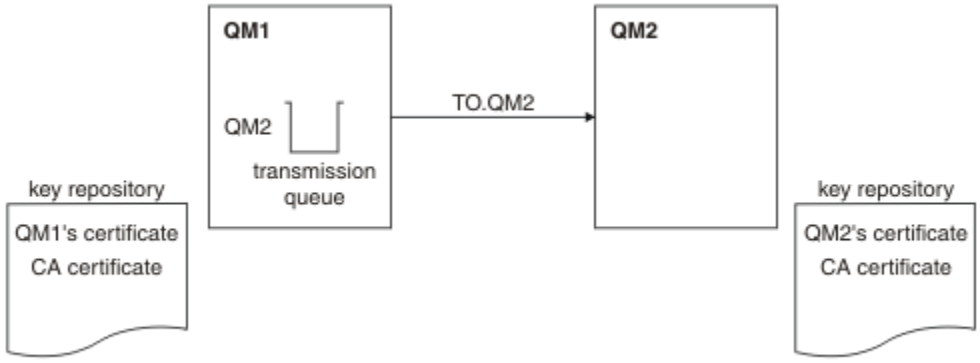


圖 *21:* 此作業產生的配置

在 第 175 頁的圖 15 中， QMA 的金鑰儲存庫包含 QMA 的憑證及 CA 憑證。 QMB 的金鑰儲存庫包含 QMB 的憑證及 CA 憑證。 在此範例中， QMA 的憑證和 QMB 的憑證都由相同的 CA 發出。 如果 QMA 的憑證和 QMB 的憑證是由不同的 CA 發出，則 QMA 和 QMB 的金鑰儲存庫必須同時包含這兩個 CA 憑證。

程序

1. 根據作業系統，在每一個佇列管理程式上準備金鑰儲存庫:
   - 在 UNIX、 Linux 及 Windows 系統上。
2. 要求每一個佇列管理程式的 CA 簽章憑證。
   您可以對兩個佇列管理程式使用不同的 CA。
   - 在 UNIX、 Linux 及 Windows 系統上。
3. 將憑證管理中心憑證新增至每一個佇列管理程式的金鑰儲存庫:
   如果佇列管理程式使用不同的憑證管理中心，則必須將每一個憑證管理中心的 CA 憑證新增至兩個金鑰儲存庫。
   - 在 UNIX、 Linux 及 Windows 系統上。
4. 將 CA 簽章憑證新增至每一個佇列管理程式的金鑰儲存庫:
   - 在 UNIX、 Linux 及 Windows 系統上。
5. 在 QMA 上，發出類似下列範例的指令來定義傳送端通道及相關聯的傳輸佇列:

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

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

   此範例使用 CipherSpec RC4_MD5。 通道每一端的 CipherSpecs 必須相同。
6. 在 QMB 上，發出類似下列範例的指令來定義接收端通道:

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

   通道必須與您在步驟 6 中定義的傳送端通道同名，並使用相同的 CipherSpec。
7. 啟動通道:

結果

金鑰儲存庫和通道如 第 175 頁的圖 15 中所示建立。

下一步

使用 DISPLAY 指令來檢查作業是否已順利完成。 如果作業成功，產生的輸出會如下列範例所示。

從佇列管理程式 QMA 中，輸入下列指令:

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

產生的輸出類似下列範例:

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

從佇列管理程式 QMB 中，輸入下列指令:

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

產生的輸出類似下列範例:

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

在每一種情況下， SSLPEER 值必須符合在步驟 2 中建立之夥伴憑證中的「識別名稱 (DN)」值。 發證者名稱符合步驟 4 中所新增簽署個人憑證之 CA 憑證的主體 DN。

使用單向鑑別來連接兩個佇列管理程式

請遵循下列範例指示來修改具有交互鑑別的系統，以容許佇列管理程式使用單向鑑別來連接至另一個系統; 亦即，當 SSL 或 TLS 用戶端未傳送憑證時。

關於這項作業

測試情境:

- 您的兩個佇列管理程式 (QM1 及 QM2) 已設定為在 第 175 頁的『使用 CA 簽章憑證來進行兩個佇列管理程式的交互鑑別』中。
- 您想要變更 QM1 ，讓它使用單向鑑別連接至 QM2。

產生的配置如下所示:

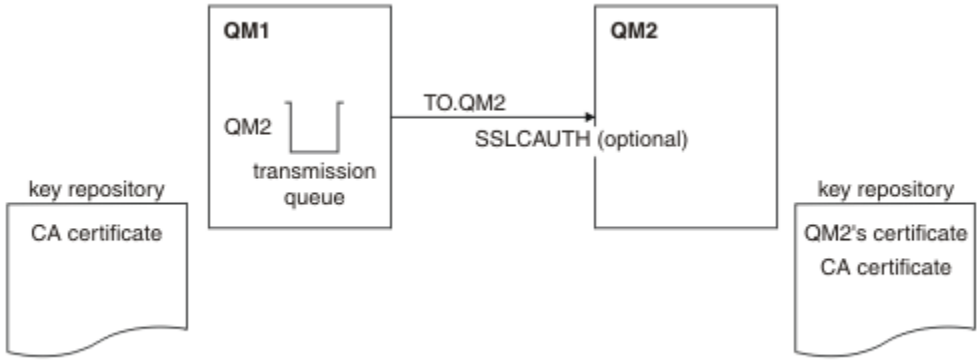


圖 *22:* 容許單向鑑別的佇列管理程式

程序

1. 根據作業系統，從其金鑰儲存庫中移除 QM1 的個人憑證:
   - 在 UNIX、 Linux 及 Windows 系統上。 憑證標示如下:
     - `ibmwebspheremq` 後面接著佇列管理程式的名稱，並轉換成小寫。 例如，若為 `QM1` ，則為 `ibmwebspheremqqm1`。
2. 選擇性的: 在 QM1 上，如果先前已執行任何 SSL 或 TLS 通道，請重新整理 SSL 或 TLS 環境中所述。
3. 容許接收端中所述。

結果

金鑰儲存庫和通道已變更，如 第 177 頁的圖 16 中所示

下一步

如果傳送端通道正在執行中，且您發出 REFRESH SECURITY TYPE (SSL) 指令 (在步驟 2 中) ，則通道會自動重新啟動。 如果傳送端通道不在執行中，請啟動它。

在通道的伺服器端，通道狀態顯示畫面上存在對等節點名稱參數值表示已傳送用戶端憑證。

請發出一些 DISPLAY 指令來驗證作業已順利完成。 如果作業成功，則產生的輸出類似於下列範例中所示的輸出:

從 QM1 佇列管理程式中，輸入下列指令:

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

產生的輸出將類似於下列範例:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(TO.QMB) SSLPEER
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QM2)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMB,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QM2)
```

從 QM2 佇列管理程式中，輸入下列指令:

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

產生的輸出將類似於下列範例:

```
DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QMA)                            SSLCERTI( )
   SSLPEER( )                              STATUS(RUNNING)
   SUBSTATE(RECEIVE)                       XMITQ( )
```

在 QM2 上， SSLPEER 欄位是空的，表示 QM1 未傳送憑證。 在 QM1 上， SSLPEER 的值符合 QM2 個人憑證中 DN 的值。

將用戶端安全連接至佇列管理程式

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。 您也必須取得並管理數位憑證。 在測試系統上，您可以使用自簽憑證或本端憑證管理中心 (CA) 發出的憑證。 在正式作業系統上，請勿使用自簽憑證。 如需相關資訊，請參閱 ../zs14140_.dita。

如需建立及管理憑證的完整資訊，請參閱 第 95 頁的『在 UNIX, Linux, and Windows 系統上使用 SSL 或 TLS』。

此主題集合介紹設定 SSL 通訊所涉及的作業，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 WebSphere MQ 實作， SSL 或 TLS 伺服器一律會向用戶端要求憑證。

在 UNIX, Linux, and Windows 系統上，只有在 SSL 或 TLS 用戶端具有以正確 WebSphere MQ 格式標示的憑證時，它才會傳送憑證，即 `ibmwebspheremq` 後面接著您的登入使用者 ID 變更為小寫，例如 `ibmwebspheremqmyuserid`。

WebSphere MQ 在標籤上使用 `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。 如需匿名連接佇列管理程式的相關資訊，請參閱 第 181 頁的『以匿名方式將用戶端連接至佇列管理程式』。

使用自簽憑證進行用戶端及佇列管理程式的交互鑑別

遵循下列範例指示，使用自簽 SSL 或 TLS 憑證來實作用戶端與佇列管理程式之間的交互鑑別。

關於這項作業

測試情境:

- 您具有需要安全通訊的用戶端 C1 及佇列管理程式 QM1。 您需要在 C1 與 QM1 之間執行交互鑑別。
- 您已決定使用自簽憑證來測試安全通訊。

DCM on IBM i 不支援自簽憑證，因此這項作業不適用於 IBM i 系統。

產生的配置如下所示:

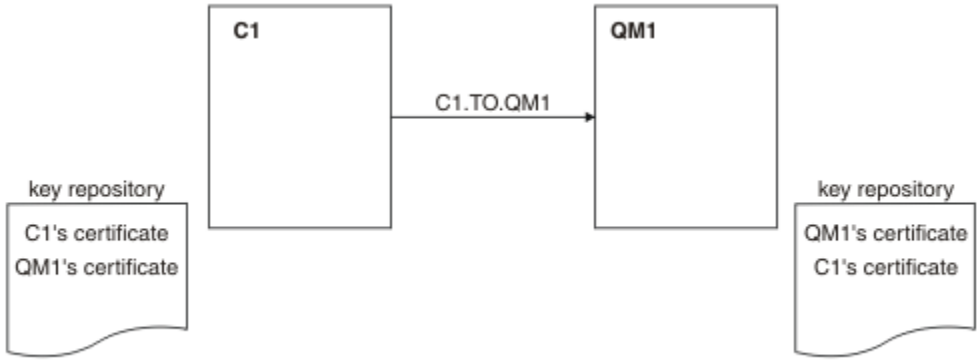


圖 *23:* 此作業產生的配置

在 第 179 頁的圖 17 中， QM1 的金鑰儲存庫包含 QM1 的憑證及來自 C1 的公用憑證。 C1 的金鑰儲存庫包含 C1 的憑證，以及來自 QM1 的公用憑證。

程序

1. 根據作業系統，在用戶端及佇列管理程式上準備金鑰儲存庫:
   - 在 UNIX、 Linux 及 Windows 系統上。
2. 為用戶端及佇列管理程式建立自簽憑證:
   - 在 UNIX、 Linux 及 Windows 系統上。
3. 擷取每一個憑證的副本:
   - 在 UNIX、 Linux 及 Windows 系統上。
4. 使用 FTP 中所述。
5. 將夥伴憑證新增至用戶端及佇列管理程式的金鑰儲存庫:
   - 在 UNIX、 Linux 及 Windows 系統上。
6. 在佇列管理程式上發出 REFRESH SECURITY TYPE (SSL) 指令。
7. 以下列其中一種方式來定義用戶端連線通道:
   - 在 C1 上搭配使用 MQCONNX 呼叫與 MQSCO 結構，如 在 WebSphere MQ MQI 用戶端上建立用戶端連線通道中所述。
   - 使用用戶端通道定義表，如 在伺服器上建立伺服器連線及用戶端連線定義 中所述。
8. 在 QM1 上，發出類似下列範例的指令來定義伺服器連線通道:

   ```
   DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC4_MD5_US)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
   ```

   通道必須與您在步驟 6 中定義的用戶端連線通道同名，並使用相同的 CipherSpec。

結果

金鑰儲存庫和通道如 第 179 頁的圖 17 中所示建立。

下一步

使用 DISPLAY 指令來檢查作業是否已順利完成。 如果作業成功，則產生的輸出類似於下列範例中所示的輸出。

從佇列管理程式 QM1，輸入下列指令:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

產生的輸出類似下列範例:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                    CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                     CURRENT
   SSLCERTI("CN=QM1,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5E:02,CN=QM2,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

設定通道定義的 SSLPEER 過濾器屬性是選用的。 如果設定通道定義 SSLPEER，則其值必須符合在步驟 2 中建立的夥伴憑證中的主體 DN。 成功連線之後， DISPLAY CHSTATUS 輸出中的 SSLPEER 欄位會顯示遠端用戶端憑證的主體 DN。

使用 CA 簽章憑證進行用戶端及佇列管理程式的交互鑑別

請遵循下列範例指示，使用 CA 簽署的 SSL 或 TLS 憑證來實作用戶端與佇列管理程式之間的交互鑑別。

關於這項作業

測試情境:

- 您具有需要安全通訊的用戶端 C1 及佇列管理程式 QM1。 您需要在 C1 與 QM1 之間執行交互鑑別。
- 未來您計劃在正式作業環境中使用此網路，因此您已決定從頭開始使用 CA 簽章憑證。

產生的配置如下所示:

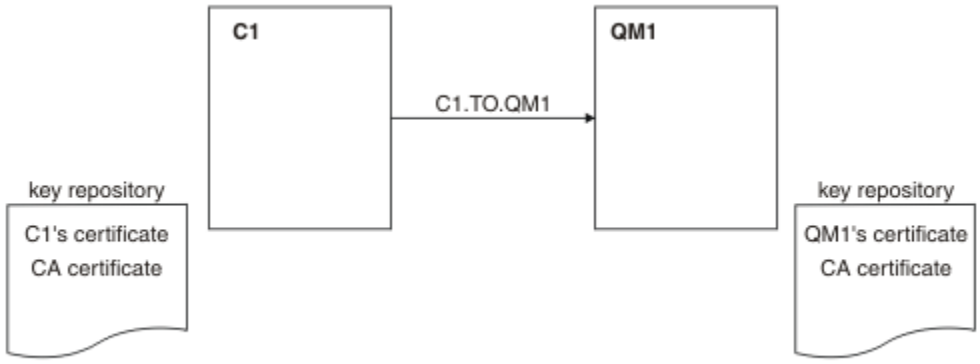


圖 *24:* 此作業產生的配置

在 第 180 頁的圖 18 中， C1 的金鑰儲存庫包含 C1 的憑證及 CA 憑證。 QM1 的金鑰儲存庫包含 QM1 的憑證及 CA 憑證。 在此範例中， C1 的憑證和 QM1 的憑證都由相同的 CA 發出。 如果 C1 的憑證和 QM1 的憑證是由不同的 CA 發出，則 C1 和 QM1 的金鑰儲存庫必須同時包含這兩個 CA 憑證。

程序

1. 根據作業系統，在用戶端及佇列管理程式上準備金鑰儲存庫:
   - 在 UNIX、 Linux 及 Windows 系統上。
2. 要求用戶端及佇列管理程式的 CA 簽章憑證。
   您可以對用戶端及佇列管理程式使用不同的 CA。
   - 在 UNIX、 Linux 及 Windows 系統上。
3. 將憑證管理中心憑證新增至用戶端及佇列管理程式的金鑰儲存庫。
   如果用戶端及佇列管理程式使用不同的「憑證管理中心」，則必須將每一個「憑證管理中心」的 CA 憑證新增至這兩個金鑰儲存庫。

- 在 UNIX、 Linux 及 Windows 系統上。

4. 將 CA 簽章憑證新增至用戶端及佇列管理程式的金鑰儲存庫:
   - 在 UNIX、 Linux 及 Windows 系統上。
5. 以下列其中一種方式來定義用戶端連線通道:
   - 在 C1 上搭配使用 MQCONNX 呼叫與 MQSCO 結構，如 在 WebSphere MQ MQI 用戶端上建立用戶端連線通道中所述。
   - 使用用戶端通道定義表，如 在伺服器上建立伺服器連線及用戶端連線定義 中所述。
6. 在 QM1 上，發出類似下列範例的指令來定義伺服器連線通道:

```
DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC2_MD5_EXPORT)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
```

   通道必須與您在步驟 6 中定義的用戶端連線通道同名，並使用相同的 CipherSpec。

結果

金鑰儲存庫和通道如 第 180 頁的圖 18 中所示建立。

下一步

使用 DISPLAY 指令來檢查作業是否已順利完成。 如果作業成功，產生的輸出會如下列範例所示。

從佇列管理程式 QM1 中，輸入下列指令:

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

產生的輸出類似下列範例:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                     CURRENT
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMA,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

DISPLAY CHSTATUS 輸出中的 SSLPEER 欄位會顯示在步驟 2 中建立之遠端用戶端憑證的主體 DN。 發證者名稱符合步驟 4 中所新增簽署個人憑證之 CA 憑證的主體 DN。

以匿名方式將用戶端連接至佇列管理程式

遵循這些範例指示來修改具有交互鑑別的系統，以容許佇列管理程式匿名連接至另一個。

關於這項作業

測試情境:

- 您的佇列管理程式及用戶端 (QM1 及 C1) 已設定在 第 180 頁的『使用 CA 簽章憑證進行用戶端及佇列管理程式的交互鑑別』中。
- 您想要變更 C1 ，以便它以匿名方式連接至 QM1。

產生的配置如下所示:

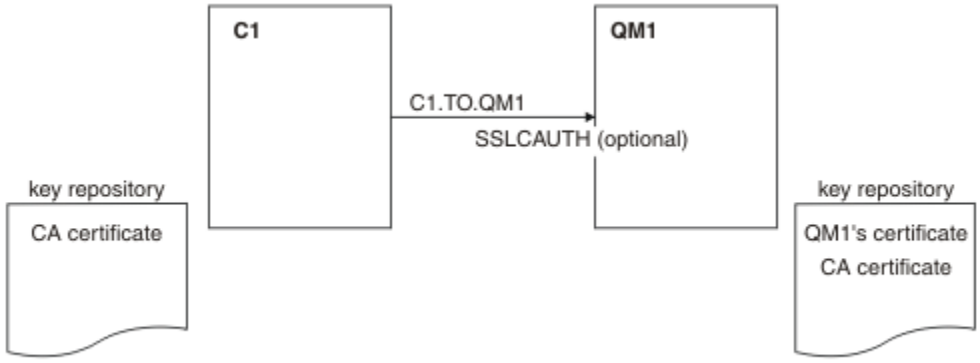


圖 *25:* 容許匿名連線的用戶端及佇列管理程式

程序

1. 根據作業系統，從 C1 的金鑰儲存庫中移除個人憑證:
   - 在 UNIX、 Linux 及 Windows 系統上。 憑證標示如下:
     - `ibmwebspheremq` 後面接著您的登入使用者 ID 會轉換成小寫，例如 `ibmwebspheremqmyuserid`。
2. 重新啟動用戶端應用程式，或讓用戶端應用程式關閉並重新開啟所有 SSL 或 TLS 連線。
3. 發出下列指令，在佇列管理程式上容許匿名連線:

```
ALTER CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) SSLCAUTH(OPTIONAL)
```

結果

金鑰儲存庫和通道已變更，如 第 182 頁的圖 19 中所示

下一步

在通道的伺服器端，通道狀態顯示畫面上存在對等節點名稱參數值表示已傳送用戶端憑證。

請發出一些 DISPLAY 指令來驗證作業已順利完成。 如果作業成功，則產生的輸出類似於下列範例中所示的輸出:

從佇列管理程式 QM1，輸入下列指令:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

產生的輸出將類似於下列範例:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                     CURRENT
   SSLCERTI( )                             SSLPEER( )
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

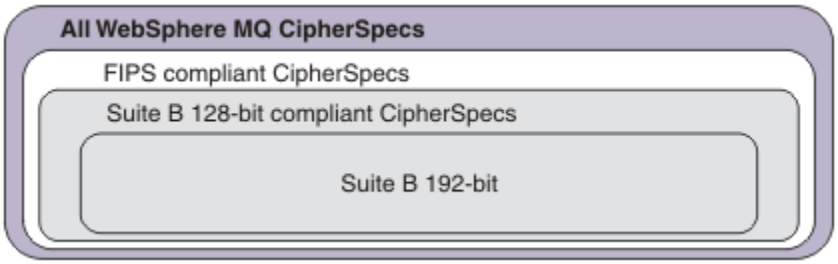
SSLCERTI 和 SSLPEER 欄位是空的，表示 C1 未傳送憑證。

指定 CipherSpecs

在 **DEFINE CHANNEL** MQSC 指令或 **ALTER CHANNEL** MQSC 指令中使用 **SSLCIPH** 參數，以指定 CipherSpec 。

您可以與 IBM WebSphere MQ 搭配使用的部分 CipherSpecs 符合 FIPS 標準。 其他 (例如 NULL_MD5) 則不是。 同樣地，部分符合 FIPS 標準的 CipherSpecs 也符合套組 B 標準，但其他不符合套組 B 標準。 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:



下表列出可與 IBM WebSphere MQ SSL 及 TLS 支援搭配使用的密碼規格。 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。 SSL 信號交換期間使用的金鑰大小是儲存在憑證中的大小，除非如表格中所述由 CipherSpec 決定。

CipherSpec 名稱	使用的通訊協定	MAC 演算法	加密演算法	加密位元	FIPS[1]	套組 B 128 位元	套組 B 192 位元
NULL_MD5 [a]	SSL 3.0	MD5	無	0	否	否	否
NULL_SHA [a]	SSL 3.0	SHA-1	無	0	否	否	否
RC4_MD5_EXPORT [2] [a]	SSL 3.0	MD5	RC4	40	否	否	否
RC4_MD5_US [a]	SSL 3.0	MD5	RC4	128	否	否	否
RC4_SHA_US [a]	SSL 3.0	SHA-1	RC4	128	否	否	否
RC2_MD5_EXPORT [2] [a]	SSL 3.0	MD5	RC2	40	否	否	否
DES_SHA_EXPORT [2] [a]	SSL 3.0	SHA-1	DES	56	否	否	否
RC4_56_SHA_EXPORT1024 [3] [b]	SSL 3.0	SHA-1	RC4	56	否	否	否
DES_SHA_EXPORT1024 [3] [b]	SSL 3.0	SHA-1	DES	56	否	否	否
TLS_RSA_WITH_AES_128_CBC_SHA [a]	TLS 1.0	SHA-1	AES	128	是	否	否
TLS_RSA_WITH_AES_256_CBC_SHA [4] [a]	TLS 1.0	SHA-1	AES	256	是	否	否
TLS_RSA_WITH_DES_CBC_SHA [a]	TLS 1.0	SHA-1	DES	56	否 [5]	否	否
FIPS_WITH_DES_CBC_SHA [b]	SSL 3.0	SHA-1	DES	56	否 [6]	否	否
TLS_RSA_WITH_AES_128_GCM_SHA256 [b]	TLS 1.2	AEAD AES-128 GCM	AES	128	是	否	否
TLS_RSA_WITH_AES_256_GCM_SHA384 [b]	TLS 1.2	AEAD AES-256 GCM	AES	256	是	否	否
TLS_RSA_WITH_AES_128_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	128	是	否	否
TLS_RSA_WITH_AES_256_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	256	是	否	否
ECDHE_ECDSA_RC4_128_SHA256 [b]	TLS 1.2	SHA-1	RC4	128	否	否	否
ECDHE_RSA_RC4_128_SHA256 [b]	TLS 1.2	SHA_1	RC4	128	否	否	否

CipherSpec 名稱	使用的通訊協定	MAC 演算法	加密演算法	加密位元	FIPS[1]	套組 B 128 位元	套組 B 192 位元
ECDHE_ECDSA_AES_128_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	128	是	否	否
ECDHE_ECDSA_AES_256_CBC_SHA384 [b]	TLS 1.2	SHA-384	AES	256	是	否	否
ECDHE_RSA_AES_128_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	128	是	否	否
ECDHE_RSA_AES_256_CBC_SHA384 [b]	TLS 1.2	SHA-384	AES	256	是	否	否
ECDHE_ECDSA_AES_128_GCM_SHA256 [b]	TLS 1.2	AEAD AES-128 GCM	AES	128	是	是	否
ECDHE_ECDSA_AES_256_GCM_SHA384 [b]	TLS 1.2	AEAD AES-256 GCM	AES	256	是	否	是
ECDHE_RSA_AES_128_GCM_SHA256 [b]	TLS 1.2	AEAD AES-128 GCM	AES	128	是	否	否
ECDHE_RSA_AES_256_GCM_SHA384 [b]	TLS 1.2	AEAD AES-256 GCM	AES	256	是	否	否
TLS_RSA_WITH_NULL_SHA256 [b]	TLS 1.2	SHA-256	無	0	否	否	否
ECDHE_RSA_NULL_SHA256 [b]	TLS 1.2	SHA-1	無	0	否	否	否
ECDHE_ECDSA_NULL_SHA256 [b]	TLS 1.2	SHA-1	無	0	否	否	否
TLS_RSA_WITH_NULL_NULL [b]	TLS 1.2	無	無	0	否	否	否
TLS_RSA_WITH_RC4_128_SHA256 [b]	TLS 1.2	SHA-1	RC4	128	否	否	否

附註:

1. 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)。
2. 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。
3. 信號交換金鑰大小是 1024 位元。
4. 這個 CipherSpec 無法用來保護從「WebSphere MQ 探險家」到佇列管理程式的連線，除非將適當的未限定原則檔套用至「探險家」所使用的 JRE。
5. 在 2007 年 5 月 19 日之前，這個 CipherSpec 已經過 FIPS 140-2 認證。
6. 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。
7. 在連線因 AMQ9288 錯誤而終止之前，這個 CipherSpec 可用來傳送最多 32 GB 的資料。 若要避免發生此錯誤，請避免使用三重 DES 演算法，或在使用這個 CipherSpec 時啟用秘密金鑰重設。

平台支援:

- a 可在所有受支援平台上使用。
- b 僅適用於 UNIX, Linux, and Windows 平台。

相關概念

第 29 頁的『IBM WebSphere 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 WebSphere 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

使用 IBM WebSphere MQ Explorer 取得 CipherSpecs 的相關資訊

您可以使用 IBM WebSphere MQ Explorer 來顯示 CipherSpecs 的說明。

使用下列程序來取得 第 182 頁的『指定 CipherSpecs』中 CipherSpecs 的相關資訊:

1. 開啟「 **IBM WebSphere MQ 探險家** 」， 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。

用於指定 CipherSpecs 的替代方案

對於作業系統提供 SSL 支援的那些平台，您的系統可能支援新的 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。

註: 本節不適用於 UNIX、 Linux 或 Windows 系統，因為 CipherSpecs 隨附於 WebSphere MQ 產品，因此新的 CipherSpecs 在出貨之後不會變成可用。

對於作業系統提供 SSL 支援的那些平台，您的系統可能支援 第 182 頁的『指定 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 必須 對應於 SSL CipherSpec ，它既有效又受系統執行的 SSL 版本支援。

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，請參閱適當的產品說明文件 (搜尋 IBM i 產品說明文件中的 *cipher_spec*)。

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
代表十六進位值的兩個字元字串。 十六進位碼對應於 SSL 通訊協定中定義的值。

如需相關資訊，請參閱 *z/OS Cryptographic Services System SSL Programming*(SC24-5901) 的 API 參考資料章節中 gsk_environment_open () 的說明，其中列出所有支援的 SSL V3.0 及 TLS V1.0 密碼規格 (格式為 2 位數十六進位碼)。

WebSphere MQ 叢集的考量

使用 WebSphere MQ 叢集，最安全的是在 第 182 頁的『指定 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 209 頁的『SSL 和叢集』。

指定 IBM WebSphere MQ MQI 用戶端的 CipherSpec

您有三個選項可用來指定 IBM WebSphere MQ MQI 用戶端的 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 WebSphere MQ classes for Java 和 IBM WebSphere MQ classes for JMS 來指定 CipherSuite

IBM WebSphere MQ classes for Java 和 IBM WebSphere MQ classes for JMS 指定不同於其他平台的 CipherSuites 。

如需使用 IBM WebSphere MQ classes for Java 來指定 CipherSuite 的相關資訊，請參閱 Secure Sockets Layer (SSL) 支援。

如需使用 IBM WebSphere MQ classes for JMS 來指定 CipherSuite 的相關資訊，請參閱 Using Secure Sockets Layer (SSL) with WebSphere MQ classes for JMS。

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 WebSphere 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 SSL 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 WebSphere 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 WebSphere 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

在您的平台上，授權存取權以連接至佇列管理程式，並放置至該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 UNIX、 Linux 及 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
```

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。

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 162 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。

相關資訊

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 SSL/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。

Windows UNIX Linux **Windows、 UNIX and Linux 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。

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)
```

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 和叢集

為叢集配置 SSL 時，請注意 CLUSRCVR 通道定義會以自動定義的 CLUSSDR 通道傳送至其他佇列管理程式。 如果 CLUSRCVR 通道使用 SSL ， 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 SSL。

如需 SSL 的相關資訊，請參閱 WebSphere MQ SSL 和 TLS 支援。 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 WebSphere 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 SSL 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 SSL 支援的存取權。 此 SSL 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 WebSphere 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=WebSphere 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 UNIX、 Linux 或 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 SSL 支援實作此 CipherSpec 的平台上，並且可以使用自訂值來指定它。 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 RC4_MD5_US)。

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

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 WebSphere MQ 版本中可能是新的，且在舊版中不受支援。 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。

將叢集中的所有佇列管理程式升級至 WebSphere MQ V6 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 SSL 從其中每一個憑證及金鑰運作。

關於這項作業

一次變更一個 CLUSRCVR ，並容許變更在變更下一個之前流經叢集。 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。

程序

1. 按您喜歡的任何順序將 CLUSRCVR 通道切換至 SSL。

   變更會在未變更為 SSL 的通道上以相反方向流動。
2. 將所有手動 CLUSSDR 通道切換至 SSL。

   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。

   註: 對於大型叢集，使用 **REFRESH CLUSTER** 指令會干擾進行中的叢集，而此後每隔 27 天，當叢集物件自動將狀態更新傳送給所有相關的佇列管理程式時，會再次造成干擾。 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。

相關概念

第 182 頁的『指定 CipherSpecs』
在 **DEFINE CHANNEL** MQSC 指令或 **ALTER CHANNEL** MQSC 指令中使用 **SSLCIPH** 參數，以指定 CipherSpec 。

第 29 頁的『IBM WebSphere 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 WebSphere 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 或 TLS

若要關閉 SSL 或 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)。
   您可以按任何您喜歡的順序關閉叢集接收端通道上的 SSL 或 TLS。

   請注意，變更會在您讓 SSL 或 TLS 處於作用中狀態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. 關閉所有手動叢集傳送端通道上的 SSL 或 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 212 頁的圖 26 中說明，並在下列範例中說明。

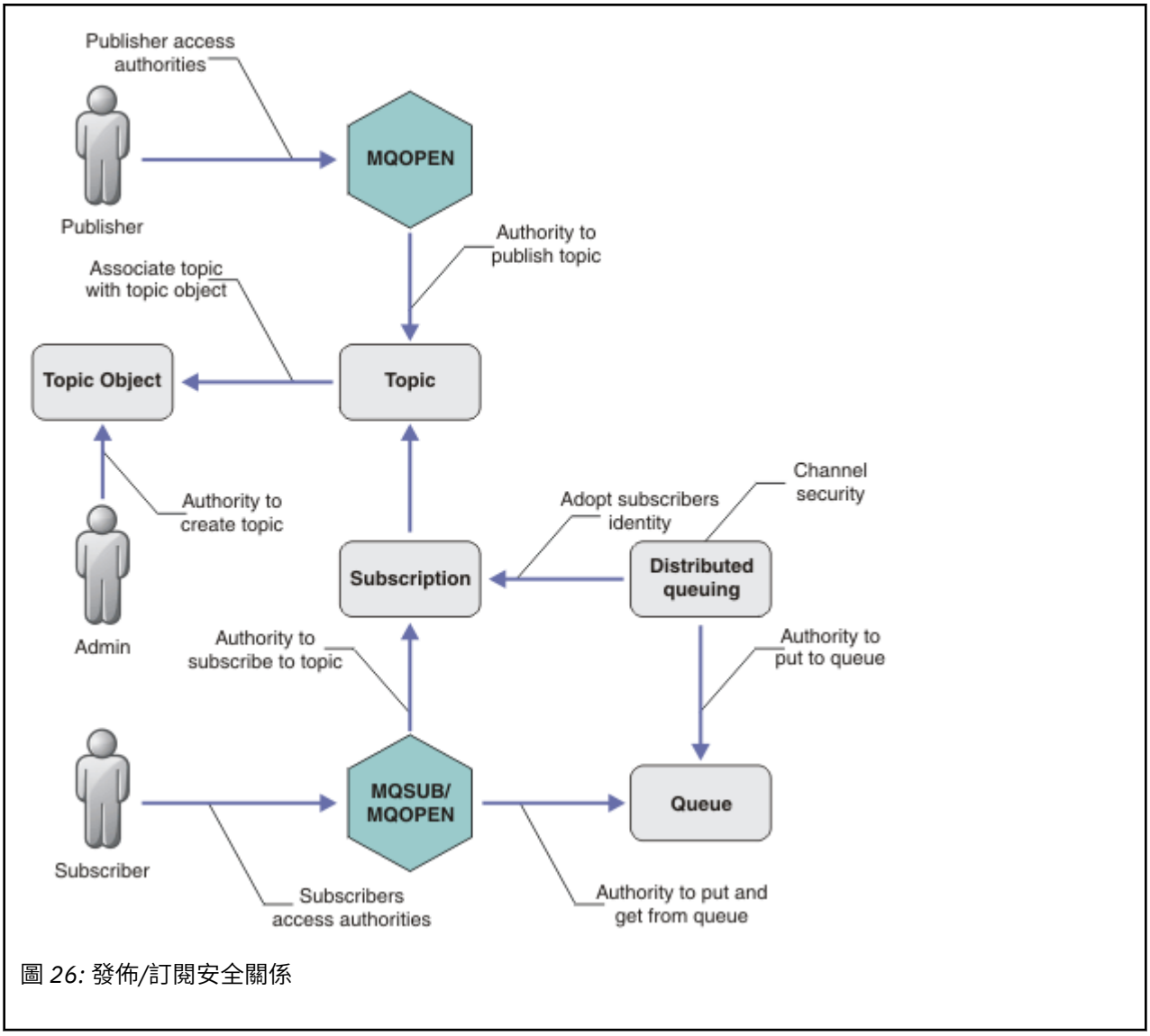


圖 *26:* 發佈/訂閱安全關係

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 214 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 217 頁的『授與使用者訂閱主題的存取權』 和 第 222 頁的『授與使用者發佈至主題的存取權』。

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 215 頁的『使用主題物件名稱訂閱』

使用主題訂閱

第 215 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 216 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 226 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 216 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。

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 WebSphere MQ 叢集配置中，通道是使用 PUTAUT=DEF 來配置，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。

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. WebSphere 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。

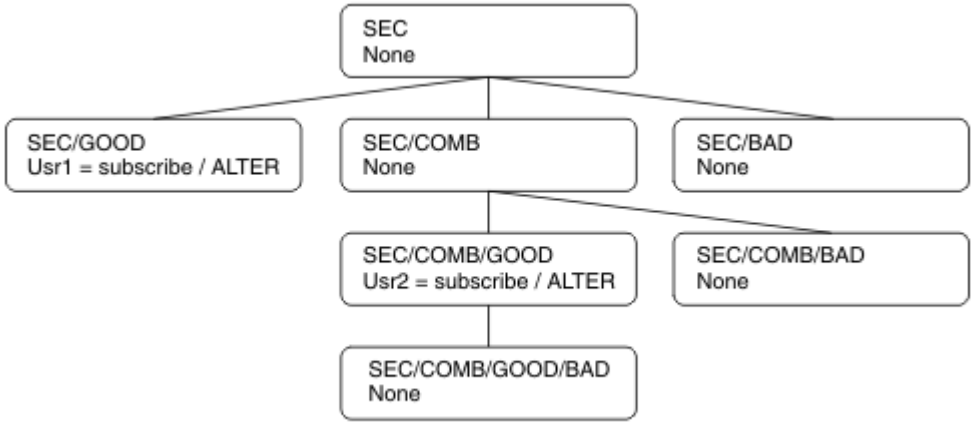
安全屬性是存取控制清單，指出特定作業系統使用者 ID 或安全群組對主題物件具有的權限。

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 *17:* 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
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。

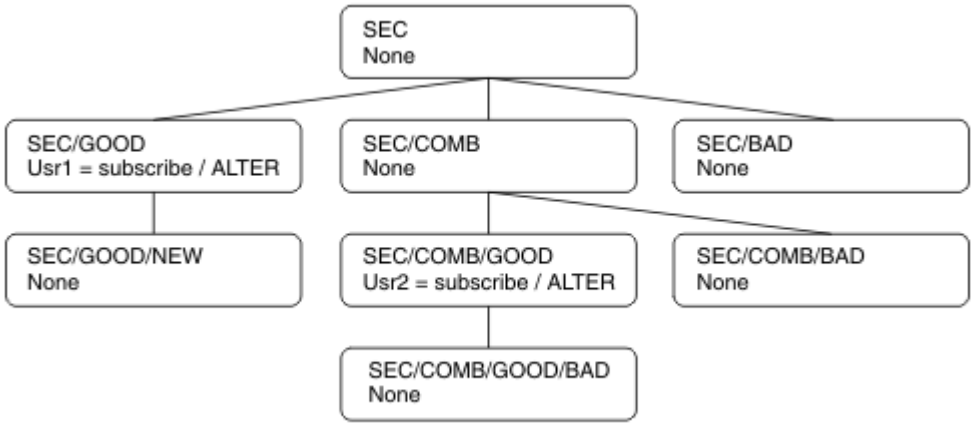
使用主題節點所在的主題字串來訂閱

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 217 頁的表 18 所示:

表 *18:* 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

本節說明在主題上具有存取控制設定的實務範例，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。

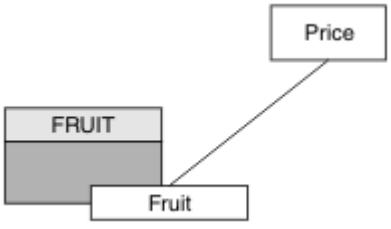


圖 *27:* 主題物件存取範例

表 19: 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:
   - 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:

     Windows UNIX Linux **Windows、 UNIX and Linux 系統**

     ```
     setmqaut -t topic -n FRUIT -p USER1 +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:

- Windows UNIX Linux 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"
```

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 217 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。

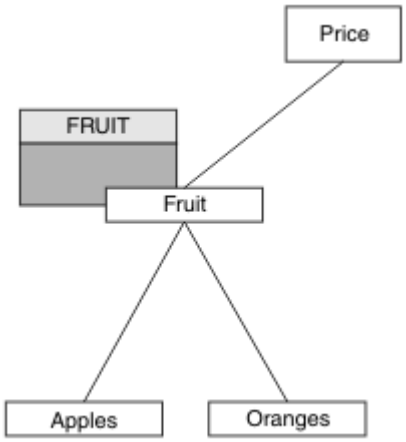


圖 *28:* 在主題樹狀結構內授與主題存取權的範例

表 *20:* 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和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 218 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。

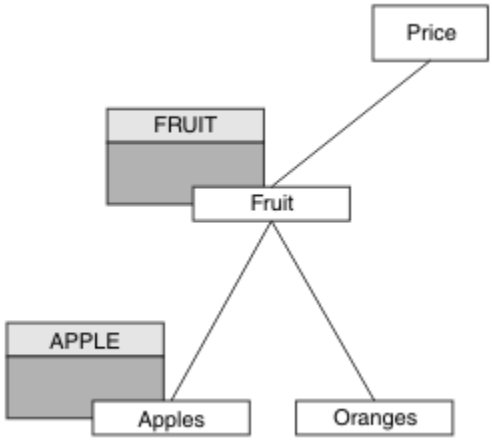


圖 *29:* 授與主題樹狀結構內特定主題的存取權

表 *21:* 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	Apple
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:
   - 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:

     Windows UNIX Linux **Windows、 UNIX and Linux 系統**

     ```
     setmqaut -t topic -n APPLE -p USER2 +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，且不會將回覆碼傳送至 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:

- Windows UNIX Linux 在 Windows、UNIX 及 Linux 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"
```

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，以及如何避免 z/OS 上的其他 RACF ICH408I 訊息。

開始之前

本主題加強 第 219 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。

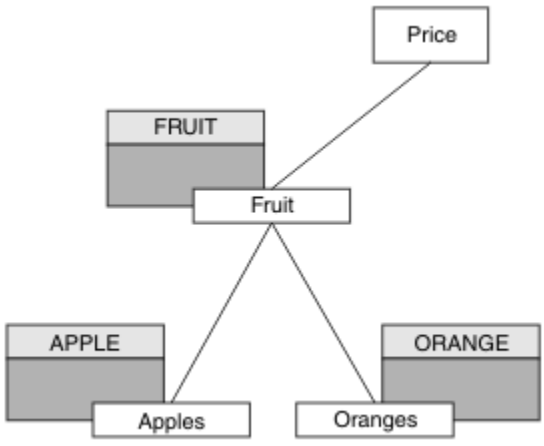


圖 *30:* 授與存取控制以避免其他訊息的範例。

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:

   - 其他平台:

     使用平台的授權指令來設定同等存取權:

     Windows、 UNIX and Linux 系統

```
setmqaut -t topic -n ORANGE -p USER1 +sub
setmqaut -t topic -n APPLE -p USER1 +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:

- 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"
```

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:

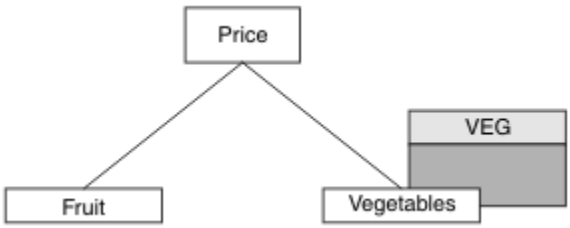


圖 *31:* 授與主題的發佈存取權

表 *22:* 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:
   - 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:

     Windows UNIX Linux **Windows、 UNIX and Linux 系統**

     ```
     setmqaut -t topic -n VEG -p USER1 +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:

- Windows UNIX Linux 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"
```

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 222 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。

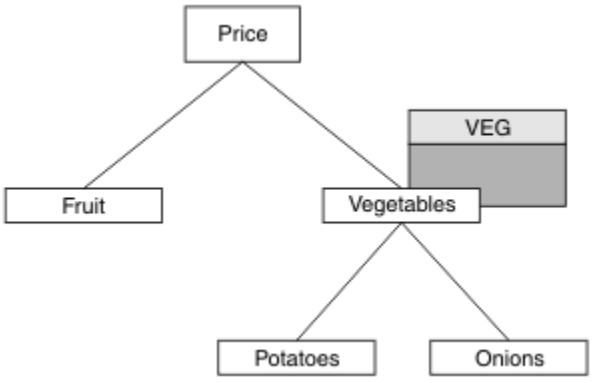


圖 *32:* 授與主題樹狀結構內主題的發佈存取權

表 *23:* 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和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 223 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。

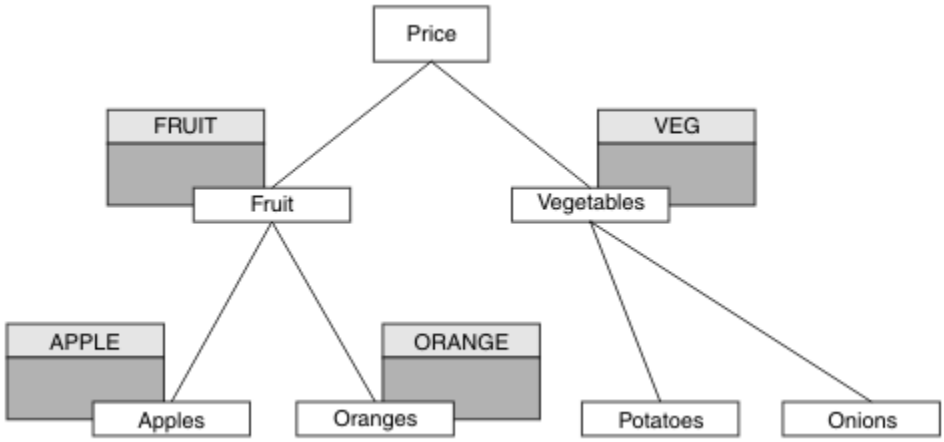


圖 *33:* 授與發佈及訂閱的存取權

表 *24:* 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		Apple
價格/水果/場	USER1		橙色

程序

授與存取權，如下所示:

- 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:

  Windows UNIX Linux **Windows、 UNIX and Linux 系統**

  ```
  setmqaut -t topic -n FRUIT -p USER1 +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:

- Windows UNIX Linux 在 Windows、UNIX 及 Linux 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"
  ```

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 *25:* 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		Apple
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 *26:* 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。

IBM WebSphere MQ Advanced Message Security

IBM WebSphere MQ Advanced Message Security (AMS) 是 IBM WebSphere MQ Advanced Message Security 的個別授權元件，可為流經 IBM WebSphere MQ Advanced Message Security 網路的機密資料提供高階保護，同時不會影響終端應用程式。

IBM WebSphere MQ Advanced Message Security 概觀

IBM WebSphere MQ 應用程式可以使用 IBM WebSphere MQ 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)。

相關參考

IBM WebSphere MQ AMS 訊息中使用的 GSKit 回覆碼

在 7.0.1 版與 7.5 版之間變更的行為

由於 IBM Advanced Message Security 在 WebSphere MQ 7.5 中變成元件， IBM WebSphere MQ AMS 功能的部分層面已變更，這可能會影響現有的應用程式、管理 Script 或管理程序。

在將佇列管理程式升級至 7.5 版之前，請仔細檢閱下列變更清單。 決定在開始將系統移轉至 IBM WebSphere MQ 7.5:

- IBM WebSphere MQ AMS 安裝是 WebSphere MQ 安裝程序的一部分。
- IBM WebSphere MQ AMS 安全功能隨其安裝而啟用，並受安全原則控制。 您不需要啟用攔截程式，即可讓 IBM WebSphere MQ AMS 開始截取資料。
- WebSphere MQ 7.5 中的 IBM WebSphere MQ AMS 不需要像在獨立式 IBM WebSphere MQ AMS 版本中一樣使用 **cfgmqs** 指令。

IBM WebSphere MQ Advanced Message Security 的特性及功能

Advanced Message Security 擴充 WebSphere 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， IBM WebSphere MQ 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。

以下是 IBM WebSphere MQ AMS 函數的完整清單:

- 保護 WebSphere 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 WebSphere MQ 的現有專有應用程式和客戶撰寫的應用程式。

錯誤處理

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 WebSphere MQ 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。

WebSphere MQ AMS 使用 SYSTEM.PROTECTION.ERROR.QUEUE 作為其錯誤處理佇列。 由 IBM WebSphere MQ AMS 放置到 SYSTEM.PROTECTION.ERROR.QUEUE 之前有 MQDLH 標頭。

您的 WebSphere 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。

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

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 249 頁的『使用金鑰儲存庫和憑證』
為了向 WebSphere 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。

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 在 Java 和原生應用程式中都支援自簽憑證

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。

物件權限管理程式

「物件權限管理程式 (OAM)」是 WebSphere MQ 產品隨附的授權服務元件。

對 Advanced Message Security 實體的存取權是透過 WebSphere 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 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

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 WebSphere MQ 應用程式設計介面 (API):

- 訊息佇列介面 (MQI)
- WebSphere MQ Java 訊息服務 (JMS) 1.0.2 和 1.1。
- WebSphere MQ Java 基礎類別
- 未受管理模式中 .Net 的 WebSphere MQ 類別

註: Advanced Message Security 支援 X.509 相容憑證管理中心。

已知限制

瞭解 IBM WebSphere MQ Advanced Message Security 的限制。

- 不支援下列 IBM WebSphere MQ 選項:

- 發佈/訂閱。
- 通道資料轉換。
- 配送清單。
- 應用程式訊息分段
- 在 HP-UX 平台上使用 API 結束程式來使用非執行緒應用程式。
- IBM WebSphere MQ classes for .NET in a managed mode (用戶端或連結連線)。
- .NET (XMS) 應用程式的訊息服務用戶端。
- 適用於 C/C++ (XMS supportPac IA94) 應用程式的訊息服務用戶端。

- 所有 Java 應用程式都相依於 IBM Java 執行時期。

  IBM WebSphere MQ Advanced Message Security 不支援其他供應商提供的 JRE。
- 在用戶端模式下使用 IBM WebSphere MQ Advanced Message Security 的 JMS 和 Java 用戶端應用程式。

  任何 JMS 或 Java 用戶端應用程式 (包括 IBM WebSphere MQ Explorer 和 IBM WebSphere MQ Managed File Transfer 代理程式) 都無法在用戶端模式下搭配使用 IBM WebSphere MQ Advanced Message Security 與 Version 7.5 之前的 WebSphere MQ 佇列管理程式。

  為了使用訊息保護原則，這些應用程式需要與 IBM WebSphere MQ Version 7.5 佇列管理程式互動，或以本端連結模式連接至與應用程式相同機器上的佇列管理程式。
- 您應該避免將兩個以上具有相同「識別名稱」的憑證放置在單一金鑰儲存庫檔中，因為未定義 IBM WebSphere MQ Advanced Message Security interceptor 對這類憑證的運作。
- IBM WebSphere MQ Version 7.5 資源配接器不支援 IBM WebSphere MQ Advanced Message Security。如果需要將訊息保護與在應用程式伺服器環境內執行的 IBM WebSphere MQ classes for JMS 或 IBM WebSphere MQ classes for Java 應用程式搭配使用，則:
  - 應用程式伺服器必須配置成使用 Version 8.0 或更新版本的資源配接器。
  - 或必須使用「訊息通道代理程式 (MCA)」截取。

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 達成的商業目標。

Windows 平台快速入門手冊

使用本手冊來快速配置 IBM 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 WebSphere 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 WebSphere MQ 指令的相關資訊，請參閱 setmqenv。

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 WebSphere MQ 介面進入 WebSphere 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 WebSphere MQ 中完成，並在下列步驟中配置。

您可以使用「WebSphere 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 (稱為 TEST.Q)，也可以使用在 \WebSphere 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. 您也必須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
```

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 234 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 240 頁的『Java 用戶端快速入門手冊』)。 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 located.This 是透過 `keystore.conf` 檔案來完成，該檔案以純文字形式保留該資訊。 每一位使用者都必須有個別的 `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，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 249 頁的『金鑰儲存庫配置檔的結構 (keystore.conf)』。
- *%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 249 頁的『使用金鑰儲存庫和憑證』。
- 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/alice/AMS/bobkey.kdb" -pw passw0rd
-label Bob_Cert -target bob_public.arm

runmqakm -cert -add -db "C:/Documents and Settings/bob/AMS/alicekey.kdb" -pw passw0rd -label
Bob_Cert -file bob_public.arm
```

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。

適用於 *UNIX* 平台的快速入門手冊

請使用本手冊來快速配置 IBM Advanced Message Security ，以在 UNIX 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元件:

- 執行時期
- 伺服器
- 範例程式
- IBM Global Security Kit
- MQ 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 系統的 IBM WebSphere MQ 元件
- HP-UX 系統的 IBM WebSphere MQ 元件
- AIX 系統的 IBM WebSphere MQ 元件
- Solaris 系統的 IBM WebSphere 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 WebSphere MQ 介面進入 WebSphere 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 WebSphere MQ 中完成，並在下列步驟中配置。

您可以使用「WebSphere 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 (稱為 TEST.Q)，也可以使用在 <MQ_INSTALL_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. 您也必須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
```

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 239 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 240 頁的『Java 用戶端快速入門手冊』)。 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 249 頁的『金鑰儲存庫配置檔的結構 (keystore.conf)』。
- 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 249 頁的『使用金鑰儲存庫和憑證』。

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。

Java 用戶端快速入門手冊

請使用本手冊來快速配置 IBM Advanced Message Security ，以針對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 UNIX) 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 WebSphere MQ 介面進入 WebSphere 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 WebSphere 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

在我們的實務範例中，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，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. 依照您平台的 **快速入門手冊** (Windows 或 UNIX) 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
```

3. 您也必須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
```

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 243 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，我們使用以 Java 撰寫並使用用戶端連結來連接的範例應用程式。 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 金鑰儲存庫和憑證。 這會顯示在 **快速入門手冊** 中 (Windows 或 UNIX)。

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 UNIX) 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，此目錄將稱為 *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 UNIX 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。
- 如果您已具有 keystore.conf ，因為您已遵循 **快速入門手冊** (Windows 或 UNIX) ，您可以編輯現有的手冊，以在上述各行中新增。
- 如需相關資訊，請參閱第 249 頁的『金鑰儲存庫配置檔的結構 (keystore.conf)』。

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 GSKit Keyman (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， WebSphere 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: WebSphere 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 UNIX)。

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 WebSphere MQ classes for JMS 所使用的環境變數中所示，以確保包含 samples 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。

保護遠端佇列

若要完全保護遠端佇列連線，必須在訊息傳輸至的遠端佇列及本端佇列上設定相同的原則。

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 WebSphere 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， WebSphere 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 會根據本端佇列上設定的原則來嘗試解碼訊息。 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 *WebSphere Message Broker* 遞送受保護的訊息

IBM Advanced Message Security 可以保護已安裝 WebSphere Message Broker 8.0.0.1 版 (或更新版本) 之基礎架構中的訊息。 在 WebSphere Message Broker 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 WebSphere Message Broker 的訊息安全，您可以容許 WebSphere Message Broker 在不知道其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-Message Broker* 無法查看訊息內容

開始之前

您應該將 WebSphere Message Broker 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，這些標頭和內容仍然不受保護，且可由 WebSphere Message Broker 讀取。 Advanced Message Security 僅由 *alice*、 *bob* 及 *cecil* 使用。 不需要針對 WebSphere Message Broker 安裝或配置它。

WebSphere Message Broker 會從未受保護的別名佇列接收受保護的訊息，以避免嘗試將訊息解密。 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 WebSphere Message Broker 遞送，到達目標佇列時未變更。 因此，它仍由原始作者簽署 (*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)。 WebSphere Message Broker 會將遞送的訊息放入未受保護的別名。 收件者會從受保護的輸出佇列中擷取訊息， IBM WebSphere MQ AMS 將在其中透通地解密訊息。

程序

1. 依照 **快速入門手冊** (Windows 或 UNIX) 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 UNIX)。
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 WebSphere Message Broker 中建立訊息流程，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。 若要這樣做，請執行下列動作:

   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 及 DEF 的 MQOutput 節點，並指派別名佇列 ABOB、 ACECIL 及 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。

[1] 塞西爾

f) 將 bob 輸出端連接至 BOB 節點。

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 WebSphere Message Broker 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 WebSphere Message Broker 從 AIN 別名佇列以受保護的形式擷取。 WebSphere Message Broker 會決定在何處遞送訊息來讀取 routeTo 內容，因為所有內容都未加密。 WebSphere Message Broker 會將訊息放在適當未受保護的別名上，避免進一步保護它。 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-Message Broker* 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 WebSphere Message Broker。 另一個群組獲授權接收 WebSphere Message Broker 所建立的訊息。 無法竊聽參與方與 WebSphere Message Broker 之間的傳輸。

請記住，只有在開啟佇列時， WebSphere Message Broker 才會讀取保護原則及憑證，因此您必須在對保護原則進行任何更新之後重新載入執行群組，變更才會生效。

```
mqsireload execution-group-name
```

如果 WebSphere Message Broker 被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 WebSphere Message Broker 服務的使用者配置 Advanced Message Security 。 請注意，將訊息放入/取得佇列的使用者不一定是同一位使用者，也不一定是建立及部署 WebSphere Message Broker 應用程式的使用者。

程序

1. 依照 **快速入門手冊** (Windows 或 UNIX) 中的說明，將 阿利切、 波布、 塞西爾 和 達韋 以及 WebSphere Message Broker 服務使用者配置成使用 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*、 *bob*、 *cecil* 及 *dave* 憑證給 WebSphere Message Broker 服務使用者。

作法是將每一個識別 *alice*、 *bob*、 *cecil* 及 *dave* 的憑證解壓縮至外部檔案，然後將解壓縮的憑證新增至 WebSphere Message Broker 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 UNIX)。

3. 提供 WebSphere Message Broker 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 WebSphere Message Broker 服務使用者的憑證，才能正確加密訊息。 WebSphere Message Broker 服務使用者需要 *alice* 及 *bob* 憑證來驗證訊息作者。 WebSphere Message Broker 服務使用者需要 *cecil* 的 及 *dave* 的 憑證來加密它們的訊息。 *cecil* 和 *dave* 需要 WebSphere Message Broker 服務使用者的憑證，以驗證訊息是否來自 WebSphere Message Broker。
4. 定義名為 IN 的本端佇列，並定義安全原則，並將 *alice* 及 *bob* 指定為作者，並將 WebSphere Message Broker 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，並定義安全原則，並將 WebSphere Message Broker 的服務使用者指定為作者，並將 *cecil* 和 *dave* 指定為收件者:

   ```
   setmqspl -m QMgrName -p OUT -s MD5 -a "CN=broker,O=IBM,C=GB" -e AES256
   -r "CN=cecil,O=IBM,C=GB" -r "CN=dave,O=IBM,C=GB"
   ```
6. 在 WebSphere Message Broker 中，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 WebSphere Message Broker 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 WebSphere Message Broker 讀取它。 WebSphere Message Broker 只會接受來自 *alice* 及 *bob* 的訊息，並會拒絕任何其他訊息。 將適當地處理接受的訊息，然後使用 *cecil* 的 及 *dave* 的 金鑰進行簽署及加密，再將其放入輸出佇列 OUT。 只有 *cecil* 和 *dave* 能夠讀取它，不會拒絕 WebSphere Message Broker 未簽署的訊息。

將 *IBM WebSphere MQ Advanced Message Security* 與 *IBM WebSphere MQ 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 IBM WebSphere MQ Managed File Transfer 傳送的資料提供訊息隱私權。

開始之前

確保您已在 WebSphere MQ 安裝上安裝 Advanced Message Security 元件，該元件管理您想要保護的 IBM WebSphere MQ Managed File Transfer 所使用的佇列。

如果您的 IBM WebSphere MQ 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 GSKit 元件。

關於這項作業

當兩個 IBM WebSphere MQ Managed File Transfer 代理程式之間的資料傳送岔斷時，機密資料可能在用來管理傳送的基礎 WebSphere MQ 佇列上仍未受保護。 此實務範例說明如何配置及使用 Advanced Message Security 來保護 IBM WebSphere MQ 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 IBM WebSphere MQ 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 hubQM，如實務範例 使用 Script 進行基本檔案傳送中所述。 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，其中使用群組 FTAGENTS 中的使用者 `ftagent` 來執行 IBM WebSphere MQ Managed File Transfer 代理程式處理程序。 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 IBM WebSphere MQ 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平台適用的 **快速入門手冊** (Windows 或 UNIX)。
- 如果 IBM WebSphere MQ Managed File Transfer 代理程式以用戶端模式執行，則 第 240 頁的『Java 用戶端快速入門手冊』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=IBM, L=Hursley, ST=Hampshire, C=GB
   ```
2. Create a `keystore.conf` file to identify the location of the keystore and the certificate within it as detailed in the appropriate Quick Start Guide.

2. 配置訊息保護

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 IBM WebSphere MQ 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=IBM,
   L=Hursley, ST=Hampshire, C=GB"
   -e AES128 -r "CN=ftagent, OU=MFT, O=IBM, L=Hursley, ST=Hampshire, C=GB"
   ```
3. 確保執行 IBM WebSphere MQ Managed File Transfer 代理程式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 IBM WebSphere MQ 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。

安裝 IBM WebSphere MQ Advanced Message Security
在各種平台上安裝 IBM WebSphere MQ Advanced Message Security 元件。

關於這項作業

如需完整安裝程序，請參閱 安裝 IBM WebSphere MQ Advanced Message Security。

相關工作

解除安裝 IBM WebSphere MQ Advanced Message Security

使用金鑰儲存庫和憑證

為了向 WebSphere 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 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。

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:

- 在 UNIX 平台上: `$HOME/.mqs/keystore.conf`
- 在 Windows 平台上: *%HOMEDRIVE%%HOMEPATH%*`\.mqs\keystore.conf`

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 UNIX and Linux 上: `export MQS_KEYSTORE_CONF=`*`path/filename`*
  - 在 Windows 上: `set MQS_KEYSTORE_CONF=`*`path\filename`*

    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)。

相關概念

第 260 頁的『傳送端識別名稱』

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。

第 261 頁的『接收端識別名稱』

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。

金鑰儲存庫配置檔的結構 (keystore.conf)

金鑰儲存庫配置檔 (`keystore.conf`) 將 Advanced Message Security 指向適當金鑰儲存庫的位置。

配置 CMS 和 Java (JKS 和 JCEKS) 有兩種類型。 視金鑰儲存庫的類型而定， CMS 配置項目會以 `cms.` 作為字首，而 Java 會以 `jks.` 或 `jceks.` 作為字首。

視配置檔的類型而定，配置檔可以具有下列其中一個結構:

```
cms.keystore = /<dir>/<keystore_file>
cms.certificate = certificate_label

jceks.keystore = <dir>/Keystore
jceks.certificate = <certificate_label>
jceks.encrypted = no
jceks.keystore_pass = <password>
jceks.key_pass = <password>
jceks.provider = IBMJCE

jks.keystore = <dir>/Keystore
jks.certificate = <certificate_label>
jks.encrypted = no
jks.keystore_pass = <password>
jks.key_pass = <password>
jks.provider = IBMJCE
```

配置檔參數定義如下:

keystore
金鑰儲存庫檔的路徑。

重要:

- 金鑰儲存庫檔的路徑不得包含副檔名。
- 對於 Java 金鑰儲存庫檔， IBM WebSphere MQ AMS 支援下列檔案格式: `.jks`、 `.jceks`、 `.jck`。

certificate
憑證標籤。

encrypted
密碼的狀態。

keystore_pass
金鑰儲存庫檔的密碼。

註:

- 對於 CMS 金鑰儲存庫， IBM WebSphere MQ AMS 依賴於隱藏檔（`.sth`），而 JKS 和 JCEKS 可能同時需要憑證和使用者私密金鑰的密碼。
- 以純文字儲存密碼是安全風險。

key_pass
使用者私密金鑰的密碼。

重要: 以純文字形式儲存密碼可能會造成安全風險。

provider
實作金鑰儲存庫憑證所需之加密演算法的 Java 安全提供者。

註: 目前 IBMJCE 是 Advanced Message Security 支援的唯一提供者。

重要: 儲存在金鑰儲存庫中的資訊對於使用 WebSphere MQ 傳送的資料安全流程非常重要，因此安全管理者在將檔案許可權指派給這些檔案時必須特別注意。

以下是 `keystore.conf` 檔的範例:

```
cms.keystore = c:\Documents and Settings\Alice\AliceKeystore
cms.certificate = AliceCert

jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = no
jceks.keystore_pass = <password>
jceks.key_pass = <password>
jceks.provider = IBMJCE
```

相關工作
第 258 頁的『在 Java 中保護密碼』
將金鑰儲存庫和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金鑰儲存庫檔中提供的使用者金鑰來編碼這些密碼。

訊息通道代理程式 (MCA) 截取

MCA 截取可讓在 IBM WebSphere 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 IBM WebSphere MQ 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。

無法在用戶端啟用 IBM WebSphere MQ AMS 時， MCA 截取旨在提供 IBM WebSphere MQ AMS 功能。 請注意，使用 MCA 截取及啟用 IBM WebSphere MQ 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。

如果您使用 Version 7.5 或更新版本用戶端從舊版產品連接至佇列管理程式，則會報告 2085 (MQRC_UNKNOWN_OBJECT_NAME) 錯誤，您需要在用戶端停用 IBM WebSphere MQ Advanced Message Security。 如需相關資訊，請參閱第 253 頁的『在用戶端停用 IBM WebSphere MQ Advanced Message Security』。

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 IBM WebSphere MQ AMS 金鑰儲存庫的相關資訊，請參閱第 249 頁的『使用金鑰儲存庫和憑證』。

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 251 頁的『IBM WebSphere MQ AMS MCA 截取範例』。

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 WebSphere MQ AMS MCA 截取範例

關於如何設定 IBM WebSphere MQ 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。

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 Version 7.5 之前， IBM WebSphere MQ 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 Version 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 Version 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/mq75
KEYSTORELOC=/home/testusr/ssl/ams1
mkdir -p $KEYSTORELOC
chmod -R 777 $KEYSTORELOC
chown -R mqm:mqm $KEYSTORELOC
export PATH=$PATH:$INSTLOC/gskit8/bin
echo "PATH = $PATH"
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$INSTLOC/gskit8/lib64

gsk8capicmd_64 -keydb -create -db $KEYSTORELOC/alicekey.kdb -pw passw0rd -stash
gsk8capicmd_64 -keydb -create -db $KEYSTORELOC/bobkey.kdb -pw passw0rd -stash
gsk8capicmd_64 -cert -create -db $KEYSTORELOC/alicekey.kdb -pw passw0rd
-label alice_cert -dn "cn=alice,O=IBM,c=IN" -default_cert yes
gsk8capicmd_64 -cert -create -db $KEYSTORELOC/bobkey.kdb -pw passw0rd
-label bob_cert  -dn "cn=bob,O=IBM,c=IN"  -default_cert yes
```

2. 在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 UNIX)。

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 Version 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 WebSphere MQ Version 7.5 安裝架構瀏覽佇列:

```
/opt/mq75/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 WebSphere MQ Version 7.5 安裝執行 **amqsgetc** :

```
/opt/mqm/samp/bin/amqsgetc TESTQ TESTQMGR
```

相關工作

第 240 頁的『Java 用戶端快速入門手冊』
請使用本手冊來快速配置 IBM Advanced Message Security ， 以針對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 229 頁的『已知限制』
瞭解 IBM WebSphere MQ Advanced Message Security 的限制。

在用戶端停用 IBM WebSphere MQ Advanced Message Security

如果您是使用 Version 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，則需要在用戶端停用 IBM WebSphere MQ Advanced Message Security (AMS)。

關於這項作業

從 Version 7.5 開始， IBM WebSphere MQ 用戶端會自動啟用 IBM WebSphere 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 Version 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，您可以在用戶端停用 AMS，如下所示:

- 若為 Java 用戶端，請使用下列任何方式:
  - V7.5.0.4 透過設定環境變數 AMQ_DISABLE_CLIENT_AMS。
  - V7.5.0.4 透過設定 Java 系統內容 com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS。
  - V7.5.0.5 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:
  - V7.5.0.4 透過設定環境變數 AMQ_DISABLE_CLIENT_AMS。
  - V7.5.0.5 透過使用 DisableClientAMS 內容，在 `mqclient.ini` 檔案中的 **Security** 段落下。

程序

- 若要在用戶端停用 AMS，請使用下列其中一個選項:

  V7.5.0.4 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 Version 7.5 或更新版本、 IBM WebSphere MQ classes for Java 或 IBM WebSphere MQ classes for JMS 用戶端。

  您也可以使用 AMQ_DISABLE_CLIENT_AMS 來停用 C 用戶端的 AMS 功能。

  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
  ```

  V7.5.0.4 com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS 系統內容

  若為 IBM WebSphere MQ classes for JMS 及 IBM WebSphere MQ classes for Java 用戶端，請將 Java 應用程式的 Java 系統內容 com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS 設為值 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)。

  V7.5.0.5 mqclient.ini 檔案中的 DisableClientAMS 內容

  若為 IBM WebSphere MQ classes for JMS 及 IBM WebSphere MQ classes for Java 用戶端，以及 C 用戶端，請在 `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 第 274 頁的『使用 JMS 時開啟受保護佇列時發生問題』。

相關概念

第 250 頁的『訊息通道代理程式 (MCA) 截取』
MCA 截取可讓在 IBM WebSphere 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 WebSphere 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 29 頁的『IBM WebSphere 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，金鑰用法必須設定如下: 對於 X.509 V3 或更新版本標準中用於保護品質完整性的憑證，如果設定金鑰用法延伸，它們必須至少包含下列其中一項:

- **nonRepudiation**
- **digitalSignature**

為了保護隱私權的品質，如果已設定金鑰用法延伸，則它們還必須包括 **keyEncipherment** 延伸。

相關概念

第 262 頁的『保護品質』
Advanced Message Security 資料保護原則暗示保護品質 (QOP)。

IBM WebSphere MQ Advanced Message Security 中的憑證驗證方法

您可以使用 IBM WebSphere MQ 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。

IBM WebSphere MQ 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。

IBM WebSphere MQ AMS 可以配置為進行 OCSP 及/或 CRL 檢查。 如果這兩種方法都已啟用，則基於效能原因， IBM WebSphere MQ 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， IBM WebSphere MQ AMS 會使用 CRL 檢查。

相關概念

第 255 頁的『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。

第 256 頁的『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。

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。

在原生攔截程式中啟用 *OCSP* 檢查

若要在 Advanced Message Security 中啟用「線上憑證狀態通訊協定 (OCSP)」檢查，您必須修改金鑰儲存庫配置檔。

程序

將下列選項新增至金鑰儲存庫配置檔：

註： 表格中提供的個別選項值為預設值。

您必須指定下列其中一個值：

- `ocsp.enable=on`
- `ocsp.url=<reposnder_URL>`
- `ocsp.http.proxy.host=<OCSP_proxy>`

選項	說明
`ocsp.enable=off`	如果要檢查的憑證，其「權限資訊存取延伸」為 PKIX_AD_OCSP 存取方法，其中包含「OCSP 回應端」所在的 URI，則啟用 OCSP 檢查。 可能的值：`on/off`。
`ocsp.url=<reposnder_URL>`	OCSP 回應端的 URL 位址。
`ocsp.http.proxy.host=<OCSP_proxy>`	OCSP Proxy 伺服器的 URL 位址。
`ocsp.http.proxy.port=<port_number>`	OCSP Proxy 伺服器的埠號。
`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。
`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。

在 *Java* 中啟用 *OCSP* 檢查

若要在 Advanced Message Security 中啟用 Java 的 OCSP 移入，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已設定「權限資訊存取 (AIA)」。

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";
```

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，其中包含授信憑證或用來主張憑證信任的授信主要憑證。IBM WebSphere MQ AMS 使用 PKIX 驗證演算法來驗證憑證路徑。 當建立並驗證鏈結時， IBM WebSphere MQ 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， IBM WebSphere MQ 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，根據配置檔中指定的值， IBM WebSphere MQ AMS 會從檔案或 LDAP 下載 CRL。 IBM WebSphere MQ 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 IBM WebSphere MQ AMS 不會執行 CRL 有效性檢查。 對於每一個 CA 憑證， IBM WebSphere MQ AMS 會使用 CA 的「識別名稱」來查詢 LDAP 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` 。

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， Advanced Message Security 才能從「輕量型目錄存取通訊協定 (LDAP)」伺服器下載 CLR。

程序

將下列選項新增至配置檔:

註: 表格中提供的個別選項值為預設值。

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。 您最多可以指定 11 部伺服器。 當 LDAP 連線失敗時，會使用多部 LDAP 主機來確保透通失效接手。 預期所有 LDAP 伺服器都是抄本，且包含相同的資料。 當 IBM WebSphere MQ AMS Java 攔截程式順利連接至 LDAP 伺服器時，它不會嘗試從提供的其餘伺服器下載 CRL。
`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。
`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。
`crl.http.max_response_size=204800`	可從 GSKit 接受的 HTTP 伺服器擷取的 CRL 大小上限 (以位元組為單位)。
`crl.http.timeout=30`	伺服器回應的等待時間 (以秒為單位) ， 在此之後 IBM WebSphere MQ AMS 逾時。
`crl.http.cache_size=0`	HTTP 快取記憶體大小 (以位元組為單位)。

在 *Java* 中啟用憑證撤銷清冊支援
若要在 Advanced Message Security 中啟用 CRL 支援，您必須修改金鑰儲存庫配置檔，以容許 IBM WebSphere MQ AMS 從「輕量型目錄存取通訊協定 (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=<*host_name*>	LDAP 主機名稱。
crl.ldap.port=<*port_number*>	LDAP 伺服器埠號。 您最多可以指定 11 部伺服器。 當 LDAP 連線失敗時，會使用多部 LDAP 主機來確保透通失效接手。 預期所有 LDAP 伺服器都是抄本，且包含相同的資料。 當 IBM WebSphere MQ AMS Java 攔截程式順利連接至 LDAP 伺服器時，它不會嘗試從提供的其餘伺服器下載 CRL。 Java 不使用 crl.ldap.user 和 crl.ldaworldp.pass 值。 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 屬性必須可供全球讀取。
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。 如果設為 true，則在執行憑證撤銷檢查時，會使用憑證的 CRL 配送點延伸中的 URL 來尋找 CRL。 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。
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 0 會停用快取; -1 表示生命期限無限制。 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。

在 Java 中保護密碼

將金鑰儲存庫和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金鑰儲存庫檔中提供的使用者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檔案。 本章中說明的密碼保護只是額外的保護措施。

程序

1. 編輯 keystore.conf 檔案，以併入金鑰儲存庫和使用者標籤的路徑。

```
jceks.keystore = c:/Documents and Setting/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.provider = IBMJCE
```

2. 若要執行工具，請發出:

```
java -cp com.ibm.mq.jmqi.jar com.ibm.mq.ese.config.KeyStoreConfigProtector keystore_password
private_key_password
```

即會產生具有已加密密碼的輸出，並可複製到 keystore.conf 檔案。

若要自動將輸出複製到 keystore.conf 檔案，請執行:

```
java -cp com.ibm.mq.jmqi.jar com.ibm.mq.ese.config.KeyStoreConfigProtector keystore_password
private_key_password >> ~/<path_to_keystore>/keystore.conf
```

註:

如需各種平台上 keystore.conf 的預設位置清單，請參閱 第 249 頁的『使用金鑰儲存庫和憑證』。

範例

以下是這類輸出的範例:

```
#Fri Jul 30 15:20:29 CEST 2010
jceks.key_pass=MMXh997n5ZOr8uRlJmc5qity9MN2CggGBMKCDxdbn1AyPklvdgTsOLG6X3C1YT7oDzwaqZFlOR4t\r\nm
Zsc7JGAx8nqqxLnAucdGn0NWo6xnjZB1n501YGol2k/
PhaQHhFXKMAU9dKg0f8djOtCAOlX4ETe\r\nfYl9LBUt2wk87uM7dSs\=
jceks.keystore_pass=OIdeayBnSCfLG4cFuxEVrk6SYyAsdSPpDqgPf16s9s1M04cqZjNbhgjoA2EXonudHZHH+4s2drvQ
\r\nCUvQgu9GuaBMJK2F2OjtHJJ1Y4BVeLW2c2okgawo/
W2J1AdUYKkJ0raYTkDouLaTYTQeulyG0xIl\r\niD2si1xUCxhYvvyhbbY\=
jceks.encrypted=yes
```

管理 IBM WebSphere MQ Advanced Message Security 安全原則

IBM WebSphere MQ 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。

安全原則概觀

IBM Advanced Message Security 安全原則是概念性物件，說明以加密方式加密及簽署訊息的方式。

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 262 頁的『保護品質』
Advanced Message Security 資料保護原則暗示保護品質 (QOP)。

第 262 頁的『安全原則屬性』
您可以使用 Advanced Message Security 來選取特定演算法或方法，以保護資料。

原則名稱

原則名稱是唯一名稱，可識別特定的 Advanced Message Security 原則及其適用的佇列。

原則名稱必須與其套用的佇列名稱相同。 Advanced Message Security (IBM WebSphere MQ AMS) 原則與佇列之間存在一對一對映。

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 IBM WebSphere MQ 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。

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。

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指定至少一個收件者 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。

容錯

容錯屬性指出 IBM 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。

傳送端識別名稱

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。

在擷取訊息之前， IBM Advanced Message Security (IBM WebSphere MQ AMS) 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的有效傳送端，而且將訊息放在佇列上的使用者不在有效傳送端清單中，則 IBM WebSphere MQ AMS 會將錯誤傳回給取得應用程式，並將訊息放在其錯誤佇列上。

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，則假設使用者憑證是受信任的，任何使用者都可以將已保護資料的訊息放在佇列上。

傳送端識別名稱的格式如下:

```
CN=Common Name,O=Organization,C=Country
```

重要:

- 所有 DN 都必須是大寫。 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。
- IBM WebSphere MQ AMS 僅支援具有 Latin-1 字集值的 DN。 若要使用集合中的字元來建立 DN，您必須先使用開啟 UTF-8 編碼或使用 iKeyman 公用程式使用 UNIX 平台來建立以 UTF-8 編碼建立的 DN 來建立憑證。 然後，您必須從開啟 UTF-8 編碼的 UNIX 平台建立原則，或使用 WebSphere MQ 的 IBM WebSphere MQ AMS 外掛程式。

相關概念

第 261 頁的『接收端識別名稱』
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。

接收端識別名稱

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 都必須是大寫。 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。 若要使用集合中的字元來建立 DN，您必須先使用開啟 UTF-8 編碼或使用 iKeyman 公用程式使用 UNIX 平台來建立以 UTF-8 編碼建立的 DN 來建

立憑證。 然後，您必須從開啟 UTF-8 編碼的 UNIX 平台建立原則，或使用 WebSphere MQ 的 Advanced Message Security 外掛程式。

相關概念

第 260 頁的『傳送端識別名稱』
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。

安全原則屬性

您可以使用 Advanced Message Security 來選取特定演算法或方法，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。 下表呈現 Advanced Message Security 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。

保護品質

Advanced Message Security 資料保護原則暗示保護品質 (QOP)。

Advanced Message Security 中的三種保護品質層次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。
- 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"。

不規定簽章演算法或加密演算法的原則具有 QOP NONE。 Advanced Message Security 對於具有 QOP 為 NONE 之原則的佇列，不提供資料保護。

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都從下列位置執行:

- 在 UNIX 平台上: `<MQInstallRoot>/bin`
- 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。

相關工作

第 263 頁的『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 264 頁的『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 264 頁的『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 265 頁的『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。

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 WebSphere MQ 物件的命名規則。
- 您必須具有必要的 `+connect +inq +chg` 權限才能建立安全原則。 如需授權變更指令的完整語法，請參閱 **`setmqaut`**。
- 請確定您具有在 WebSphere MQ 佇列及佇列管理程式上運作的必要許可權。 如需相關資訊，請參閱 第 267 頁的『授與 OAM 許可權』

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 SHA1 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 SHA1 -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 DES 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 MD5 演算法簽署 :CN=phil， O=IBM， C=US

```
$ setmqspl -m QMGR -p MY.OTHER.QUEUE -s MD5 -e DES -r CN=john,O=IBM,C=US -r CN=jeff,O=IBM,C=US
-a CN=phil,O=IBM,C=US
```

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

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 +connect +inq +chg 權限，才能建立安全原則。 如需授權變更指令的完整語法，請參閱 **setmqaut** 。

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

以下是在名為 QMGR 的佇列管理程式上建立名為 MYQUEUE 的原則的範例，該原則指定將針對具有 DN:CN=bob， O=IBM， C=US 的憑證使用 RC2 演算法來加密訊息，並針對具有 DN:CN=Jeff， O=IBM， C = US 的憑證使用 SHA1 演算法來簽署。

```
setmqspl -m QMGR -p MYQUEUE -e RC2 -s SHA1 -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 SHA1 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

相關參考

setmqspl

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。
- 您必須具有套用至佇列管理程式的必要 +connect +inq +dsp 許可權，才能顯示及傾出安全原則。 如需授權變更指令的完整語法，請參閱 **setmqaut** 。

關於這項作業

以下是 dspmqspl 指令旗標的清單:

表 *27: 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

在此範例中，我們將為 venus.queue.manager 建立兩個安全原則:

```
setmqspl -m venus.queue.manager -p AMS_POL_04_ONE -s MD5 -a "CN=signer1,O=IBM,C=US" -e NONE
setmqspl -m venus.queue.manager -p AMS_POL_06_THREE -s MD5 -a "CN=another signer,O=IBM,C=US" -e
```

```
NONE
```

此範例顯示一個指令，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: MD5
Encryption algorithm: NONE
Signer DNs:
  CN=signer1,O=IBM,C=US
Recipient DNs: -
Toleration: 0
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Policy Details:
Policy name: AMS_POL_06_THREE
Quality of protection: INTEGRITY
Signature algorithm: MD5
Encryption algorithm: NONE
Signer DNs:
  CN=another signer,O=IBM,C=US
Recipient DNs: -
Toleration: 0
```

此範例顯示一個指令，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: MD5
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 MD5 -a "CN=signer1,O=IBM,C=US" -e NONE

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:

- 在 Windows 平台上，執行 policies.bat
- 在 UNIX 平台上:
  1. 以屬於 mqm WebSphere MQ 管理群組的使用者身分登入。
  2. 發出 . policies.sh。

相關參考

dspmqspl 指令屬性的完整清單

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 +connect +inq +chg 權限，才能建立安全原則。 如需授權變更指令的完整語法，請參閱 **setmqaut** 。

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

系統佇列保護

系統佇列可啟用 WebSphere 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 WebSphere 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 263 頁的『建立安全原則』。

若要在 Windows 平台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
```

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。 作為回覆傳送至「WebSphere MQ 探險家」暫時動態佇列的訊息不受 WebSphere MQ AMS 保護。

Advanced Message Security 安全原則的變更需要您重新啟動 WebSphere MQ 指令伺服器

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
- SYSTEM.ADMIN.CONFIG.EVENT
- SYSTEM.ADMIN.LOGGER.EVENT
- SYSTEM.ADMIN.PERFM.EVENT
- SYSTEM.ADMIN.PUBSUB.EVENT
- SYSTEM.ADMIN.QMGR.EVENT
- SYSTEM.ADMIN.STATISTICS.QUEUE
- SYSTEM.ADMIN.TRACE.ROUTE.QUEUE
- SYSTEM.AUTH.DATA.QUEUE
- SYSTEM.BROKER.ADMIN.STREAM
- SYSTEM.BROKER.CONTROL.QUEUE
- SYSTEM.BROKER.DEFAULT.STREAM
- SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS
- SYSTEM.CHANNEL.INITQ
- SYSTEM.CHANNEL.SYNCQ
- SYSTEM.CICS.INITIATION.QUEUE

- SYSTEM.CLUSTER.COMMAND.QUEUE
- SYSTEM.CLUSTER.HISTORY.QUEUE
- SYSTEM.CLUSTER.REPOSITORY.QUEUE
- SYSTEM.CLUSTER.TRANSMIT.QUEUE
- SYSTEM.DEAD.LETTER.QUEUE
- SYSTEM.DURABLE.SUBSCRIBER.QUEUE
- SYSTEM.HIERARCHY.STATE
- SYSTEM.INTER.QMGR.CONTROL
- SYSTEM.INTER.QMGR.FANREQ
- SYSTEM.INTER.QMGR.PUBS
- SYSTEM.INTERNAL.REPLY.QUEUE
- SYSTEM.PENDING.DATA.QUEUE
- SYSTEM.PROTECTION.ERROR.QUEUE
- SYSTEM.PROTECTION.POLICY.QUEUE
- SYSTEM.RETAINED.PUB.QUEUE
- SYSTEM.SELECTION.EVALUATION.QUEUE
- SYSTEM.SELECTION.VALIDATION.QUEUE

授與 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， IBM Advanced Message Security 會根據「物件權限管理程式 (OAM)」，以及不屬於 mqm 群組 (即 WebSphere MQ 管理群組) 或無權讀取所授與的安全原則設定的使用者每次嘗試執行這些指令都會導致錯誤。

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
```

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。

WebSphere MQ 所產生的指令及配置事件是傳送至專用佇列的 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 263 頁的『建立安全原則』，產生兩則 WebSphere MQ 事件訊息:
  - 配置事件
  - 指令事件
- 第 264 頁的『變更安全原則』，它會產生三則 WebSphere 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 264 頁的『顯示及傾出安全原則』，產生一則 WebSphere MQ 事件訊息:
  - 指令事件
- 第 265 頁的『移除安全原則』，產生兩則 WebSphere MQ 事件訊息:
  - 配置事件
  - 指令事件

啟用及停用事件記載

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。 如果要停用配置事件，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。 若要啟用指令 (DISPLAY MQSC 指令及 Inquire PCF 指令除外) 的指令事件，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

控制 Websphere MQ 中的配置、指令及日誌程式事件

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 WebSphere 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: 對於當作訊息 (MQEVO_MSG) 接收的指令，則為來自指令訊息 MD 的應用程式身分資料 (ApplIdentity 資料)。

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: 對於接收為訊息 (MQEVO_MSG) 的指令，來自指令訊息 MD 的應用程式類型 (PutAppl 類型)。

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 (MQEVO_MSG) 接收的指令，來自指令訊息 MD 的應用程式名稱 (PutAppl 名稱)。

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: 對於當作訊息 (MQEVO_MSG) 接收的指令，來自指令訊息 MD 的應用程式原始資料 (ApplOriginData)。

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。

Command

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
這些定義於 WebSphere MQ 7.5 `cmqcfc.h`

已傳回: 始終。

CommandData 包含包含 PCF 指令的 PCF 元素。

配置事件訊息格式

配置事件是標準 Advanced Message Security 格式的 PCF 訊息。

如需 MQMD 訊息描述子的可能值，請參閱 事件訊息 MQMD (訊息描述子)。

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
```

訊息緩衝區是由 MQCFH 結構及其後面的參數結構所組成。 如需可能的 MQCFH 值，請參閱 事件訊息 MQCFH (PCF 標頭)。

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「WebSphere MQ 探險家」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 WebSphere MQ 7.5 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 WebSphere MQ 7.5 或 cmqc.h 檔案中定義的數值。

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 WebSphere MQ 7.5 或 cmqc.h 檔案中定義的數值。

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 WebSphere MQ 7.5 或 cmqc.h 檔案中定義的數值。

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** -在 WebSphere MQ 7.5 或 cmqc.h 檔案中定義的數值。

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** -在 WebSphere MQ 7.5 或 cmqc.h 檔案中定義的數值。

已傳回: 每當 WebSphere 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** -定義在 WebSphere MQ 7.5 或 cmqc.h 檔案中的數值。

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 WebSphere 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 WebSphere MQ 7.5 或 cmqc.h 檔案中定義的數值。

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 WebSphere MQ 原則中定義時。

問題及解決方案

本節說明如何解決任何 IBM 安裝可能產生的問題。使用此資訊可識別並解決與 Advanced Message Security 相關的任何問題。

com.ibm.security.pkcsutil.PKCSException: 加密內容時發生錯誤

錯誤 `com.ibm.security.pkcsutil.PKCSException: Error encrypting contents` 表示 IBM Advanced Message Security 在存取加密演算法時發生問題。

如果 Advanced Message Security 傳回下列錯誤:

```
DRQJP0103E   The IBM WebSphere MQ Advanced Message Security Java interceptor failed to protect
message.
  com.ibm.security.pkcsutil.PKCSException: Error encrypting contents
(java.security.InvalidKeyException: Illegal key size or default parameters)
```

驗證 `JAVA_HOME/lib/security/local_policy.jar/*.policy` 中的 JCE 安全原則是否授與對 MQ AMS 原則中所使用簽章演算法的存取權。

如果現行安全原則中未指定您要使用的簽章演算法，請從下列位置下載正確的 Java 原則檔:

- IBM Java 的 SDK 原則檔案 1.4.2。
- IBM Java 的 SDK 原則檔案 5.0。
- IBM Java 的 SDK 原則檔案 6.0。
- IBM Java 的 SDK 原則檔案 7.0。

OSGi 支援

如果要搭配使用 OSGi 軟體組與 IBM Advanced Message Security ，則需要其他參數。

在 OSGi 軟體組啟動期間執行下列參數:

```
-Dorg.osgi.framework.system.packages.extra=com.ibm.security.pkcs7
```

在 keystore.conf 中使用加密密碼時，必須在 OSGi 軟體組執行時新增下列陳述式:

```
-Dorg.osgi.framework.system.packages.extra=com.ibm.security.pkcs7,com.ibm.misc
```

限制: 對於 OSGi 軟體組內所保護的佇列， IBM WebSphere MQ AMS 只支援使用 MQ 基本 Java 類別來進行通訊。

使用 JMS 時開啟受保護佇列時發生問題

當您在使用 IBM WebSphere MQ Advanced Message Security 時開啟受保護的佇列時，可能會發生各種問題。

您正在執行 JMS ，並且收到錯誤 2085 (MQRC_UNKNOWN_OBJECT_NAME) 以及錯誤 JMSMQ2008。

您已驗證您已依照 第 240 頁的『Java 用戶端快速入門手冊』中的說明來設定 IBM WebSphere MQ Advanced Message Security 。

可能的原因是您正在使用非 IBM Java 執行時期環境。 這是 第 229 頁的『已知限制』中說明的已知限制。

您尚未設定 AMQ_DISABLE_CLIENT_AMS 環境變數。

解析問題

有四個選項可解決此問題:

1. 在支援的 IBM Java 執行時期環境 (JRE) 下啟動 JMS 應用程式。
2. 將應用程式移至佇列管理程式執行所在的相同機器，並讓它使用連結模式連線來連接。

   連結模式連線使用平台原生程式庫來執行 IBM WebSphere MQ API 呼叫。 因此，會使用原生 AMS 攔截程式來執行 AMS 作業，而不需要依賴 JRE 的功能。
3. 請使用 MCA 攔截程式，因為這容許在訊息到達佇列管理程式時立即簽署及加密訊息，而不需要用戶端執行任何 AMS 處理程序。

   假設在佇列管理程式上套用保護，則必須使用替代機制來保護從用戶端傳送至佇列管理程式的訊息。 最常見的作法是在應用程式所使用的伺服器連線通道上配置 SSL/TLS 加密。
4. 如果您不想使用 IBM WebSphere MQ Advanced Message Security，請設定 AMQ_DISABLE_CLIENT_AMS 環境變數。

如需進一步資訊，請參閱第 250 頁的『訊息通道代理程式 (MCA) 截取』。

註: 對於 MCA 攔截程式將訊息遞送至其中的每一個佇列，必須備妥安全原則。 換句話說，目標佇列需要備妥 AMS 安全原則，且簽章者和收件者的識別名稱 (DN) 必須符合指派給 MCA 攔截程式之憑證的識別名稱 (DN)。 亦即，在佇列管理程式所使用的 `keystore.conf` 中，由 `cms.certificate.channel.SYSTEM.DEF.SVRCONN` 內容指定之憑證的 DN。

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。 本文件的使用者應驗證其特定環境適用的資料。

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 IBM 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 (http://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: