Dell™ PowerVault™ Encryption Key Manager

使用手冊

Dell™ PowerVault™ Encryption Key Manager

使用手冊

目錄

圖

表

前言

關於本書

本手冊包含安裝和操作 Dell™ Encryption Key Manager 所需要的資訊及指示。它包括關於下列各項的概念和程序：

- 具有加密功能的 LTO 4 和 LTO 5 磁帶機
- 加密金鑰
- 數位憑證

本書適用對象

本書適用於負責重要資料之安全和備份的儲存體及安全管理者，以及在作業環境中協助設定和維護 Encryption Key Manager 伺服器的所有人員。它假設讀者有儲存裝置和網路的實務知識。

本書使用慣例和專有名詞

本書排版印刷慣例如下：

表 1. 本書排版印刷慣例

慣例	用途
粗體	**粗體**單字或字元代表必須按字面使用的系統元素，如指令名稱、檔名、旗標名稱、路徑名稱和所選的功能表選項。
等寬	範例、使用者指定的文字以及系統顯示的資訊，以**等寬**字體呈現。
斜體	*斜體*單字或字元代表您必須提供的變數值。
[項目]	表示選用項目。
{項目}	在格式和語法說明中，括住一份您必須從中選取項目的清單。
|	垂直線將選項清單中的項目分開。
<Key>	表示您按下的鍵。

注意事項

注意事項表示程式、裝置、系統或資料有可能毀損。驚嘆號符號可能會伴隨著注意事項，但並非必要。注意事項的範例如下：

警告: 如果您利用電動螺絲起子來執行這個程序，磁帶可能會毀損。

相關出版品

請參閱下列出版品，以取得詳細資訊：

- *Getting Started with the Dell™ PowerVault™ TL2000 and TL4000 Tape Libraries*提供安裝資訊。
- *Dell™ PowerVault™ TL2000 Tape Library and TL4000 Tape Library SCSI Reference* 提供用來控管 SCSI 介面行爲的受支援 SCSI 指令及通訊協定。

Linux 資訊

Red Hat 資訊

下列 URL 與 Red Hat Linux® 系統相關：

- **http://www.redhat.com**

SuSE 資訊

下列 URL 與 SuSE Linux 系統相關：

- **http://www.suse.com**

Microsoft Windows 資訊

下列 URL 用來存取 Microsoft® Windows® 系統的相關資訊：

- **http://www.microsoft.com**

線上支援

請造訪 **http://support.dell.com**，以取得下列相關出版品：

*Dell Encryption Key Manager 快速入門手冊*提供設定基本配置的資訊。

請造訪 **http://www.dell.com**，以取得下列相關出版品：

Library Managed Encryption for Tape 白皮書提供 LTO 磁帶加密的最佳作法建議。

請先閱讀以下說明

聯絡 Dell

美國地區的客戶，請撥 800-WWW-DELL (800-999-3355)。

註: 如果您沒有作用中的「網際網路」連線，您可以在購貨發票、裝箱單、帳單或 Dell 產品型錄上找到聯絡資訊。

Dell 提供數個線上以及電話的支援與服務。這些支援服務依國家和產品而異，部份的服務可能沒有在您所在的地區提供。若要針對銷售、技術支援或客戶服務問題聯絡 Dell，請進行下列動作：

1. 請造訪 **http://support.dell.com**。
2. 在頁面底端的**選擇國家/地區**下拉功能表中選取您的國家或地區。
3. 按一下頁面左邊的**聯絡我們**。
4. 依據您的需要選取適當的服務或支援鏈結。
5. 請選擇您想要的聯絡方式來聯絡 Dell。

第 1 章 磁帶加密概觀

在競爭激烈的商業環境中，資料是最有價值的其中一項資源。在我們這個安全意識高張的世界裡，既能保護資料、控制資料的存取及驗證資料的眞實性，又能維持資料的可用性，是優先的要務。資料加密便是解決其中多項需求的工具。Dell Encryption Key Manager（從現在開始，稱爲 Encryption Key Manager）簡化了加密作業。

LTO 4 及 LTO 5 磁帶機也在寫入至任何 LTO 4 及 LTO 5 資料卡匣時加密資料。這個新的功能爲儲存的資料新增了牢固的安全措施，不會因爲在伺服器上執行加密而帶來額外的處理成本，造成效能降低，也不需要支付專用設備的費用。

磁帶機加密解決方案由三個主要元素組成：

啓用加密的磁帶機

所有 LTO 4 和 LTO 5 磁帶機都必須透過程式庫介面加以啓用。

請參閱第 2-2 頁的『軟硬體需求』，以取得磁帶機的詳細資訊。

加密金鑰管理

加密包括在連續的各層上使用多種金鑰。這些金鑰的產生、維護、控制及傳輸，會隨著安裝了加密磁帶機的作業環境而不同。部分應用程式能夠執行金鑰管理。對於沒有這類應用程式或需要應用程式診斷加密的環境， Dell Encryption Key Manager 會執行所有必要的金鑰管理作業。 第 1-2 頁的『管理加密』詳細說明這些作業。

加密原則

這是用來實作加密的方法。其中包括控管加密哪些磁區的規則以及金鑰選擇的機制。如何設定以及在哪裡設定這些規則，會隨著作業環境而不同。請參閱第 1-2 頁的『管理加密』，以取得詳細資訊。

元件

Encryption Key Manager 是 Java 環境的一部分，利用 Java Security 元件來執行它的加密功能。（如需 Java Security 元件的詳細資訊，請參閱相關出版品章節。） Encryption Key Manager 有三個用來控制行爲的主要元件。這些元件如下：

Java 安全金鑰儲存庫

金鑰儲存庫定義爲 Java Cryptography Extension (JCE) 的一部分，是 Java Security 元件的一個元素，Java Security 元件又是 Java Runtime 環境的一部分。金鑰儲存庫用來存放憑證和金鑰（或憑證和金鑰的指標），供 Encryption Key Manager 執行加密作業。支援的 Java 金鑰儲存庫有許多類型，提供不同的作業性質以符合您的需求。這些性質的詳細討論資訊，請參閱第 2-3 頁的『金鑰儲存庫考量』。

保留金鑰儲存庫的資料極具重要性，不容忽視。當無法存取金鑰儲存庫時，您也無法將加密的磁帶解密。請仔細閱讀下列主題，以瞭解可用來保護金鑰儲存庫資料的方法。

配置檔 配置檔可讓您自訂 Encryption Key Manager 的行爲以符合組織的需要。本文件

會詳細說明這些選項，首先是第 2-1 頁的第 2 章, 『規劃 Encryption Key Manager 環境』，其次是第 4-1 頁的第 4 章, 『配置 Encryption Key Manager』，稍後在附錄 B 中會有全套配置選項的說明。

磁帶機表格

Encryption Key Manager 利用磁帶機表格來追蹤它支援的磁帶機。磁帶機表格是配置檔指定了其位置之不可編輯的二進位檔。您可以變更它的位置以符合您的需求。

KeyGroups.xml 檔

這個密碼保護的檔案包含所有加密金鑰群組的名稱，以及每個金鑰群組之相關加密金鑰的別名。

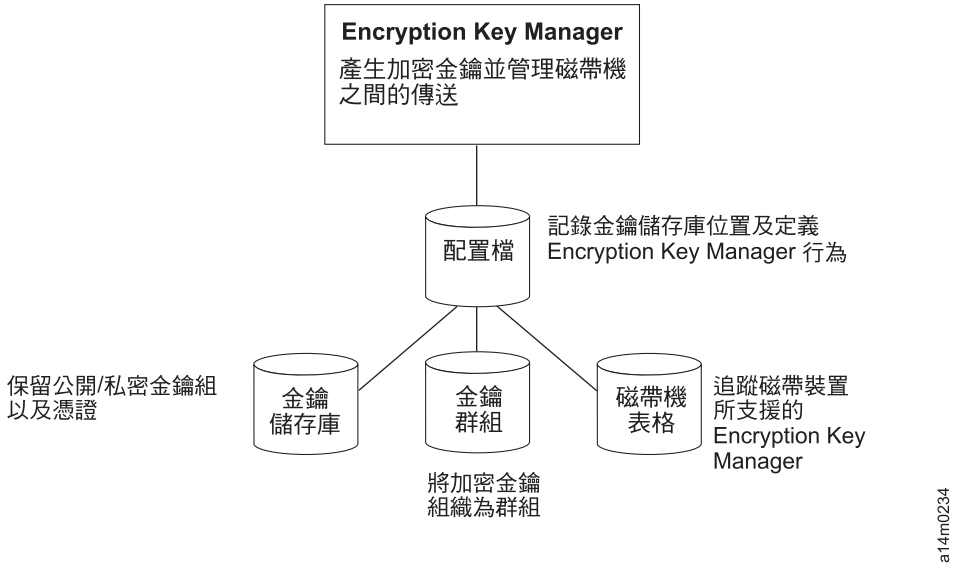

圖 1-1. Encryption Key Manager 的四個主要元件

管理加密

Dell Encryption Key Manager是一個Java™ 軟體程式，能夠協助啓用加密的磁帶機產生、保護、儲存和維護加密金鑰，以便用來加密和解密在磁帶媒體中讀寫的資訊（磁帶和卡匣格式）。 Encryption Key Manager在 Linux（SLES 和 RHEL） 和 Windows上運作，它設計成在背景中執行，作爲一項部署在企業內多個位置的共用資源。 指令行介面用戶端提供了一組健全的指令，用來自訂環境的 Encryption Key Manager 及監視它的作業。Dell Encryption Key Manager 圖形使用者介面 (GUI) 也提供許多自訂和監視功能。 Encryption Key Manager 利用一或多個金鑰儲存庫來保留所有加密作業所需要的憑證和金鑰（或憑證和金鑰的指標）。請參閱第 2-3 頁的『金鑰儲存庫考量』，以取得詳細資訊。

重要 Encryption Key Manager 主機伺服器配置資訊：建議您讓代管 Dell Encryption Key Manager 程式的機器使用 ECC 記憶體，將資料遺失的風險降低到最小。Encryption Key Manager 會執行要求產生加密金鑰及將這些金鑰傳給 LTO 4 和 LTO 5 磁帶機的功能。在 Encryption Key Manager 的處理期間，封裝（加密形式）的金鑰資料是在系統記憶體中。請注意，金鑰資料必須無誤地傳送到適當的磁帶機，才能夠回復（解密）寫在卡匣上的資料。如果由於某些原因，造成金鑰資料因系統記憶體位元錯誤而毀損，且該金鑰資料是用來將資料寫入卡匣中，則寫入這個卡匣的資料將無法復原（日後無法解密）。有一些適當的防護措施可確保不會發生這類的資料錯誤。不過，如果代管 Encryption Key Manager 的機器並未使用錯誤更正碼 (ECC) 記憶體，在系統記憶體內的金鑰資料仍有可能毀損，因而造成資料遺失。發生這個情況的機會不大，但對於代管重要應用程式（如 Encryption Key Manager）的機器，一律建議使用 ECC 記憶體。

Encryption Key Manager 是作爲一項背景處理程序來運作，在本身與磁帶庫之間，等待透過 TCP/IP 通訊路徑傳來的產生金鑰或擷取金鑰的要求。當磁帶機寫入加密資料時，它會先向 Encryption Key Manager 要求加密金鑰。收到要求時，Encryption Key Manager 會執行下列作業。

Encryption Key Manager 會從金鑰儲存庫中提取現有的 AES 金鑰，將它封裝起來，以便安全傳送給磁帶機，並在到達時解開，用來加密寫入磁帶的資料。

當 LTO 4 或 LTO 5 磁帶機讀取已加密的磁帶時，Encryption Key Manager會根據磁帶金鑰 ID 中的資訊，從金鑰儲存庫中提取必要的金鑰，將它封裝起來提供給磁帶機，以進行安全傳送。

總共有兩個加密管理方法可供選擇。這些方法不同之處，在於加密原則引擎放置的位置，解決方案的金鑰管理的執行位置，以及 Encryption Key Manager 如何連接磁帶機。作業環境決定了最適合的情況。金鑰管理和加密原則引擎可能在下列兩個環境層中的任何一層。

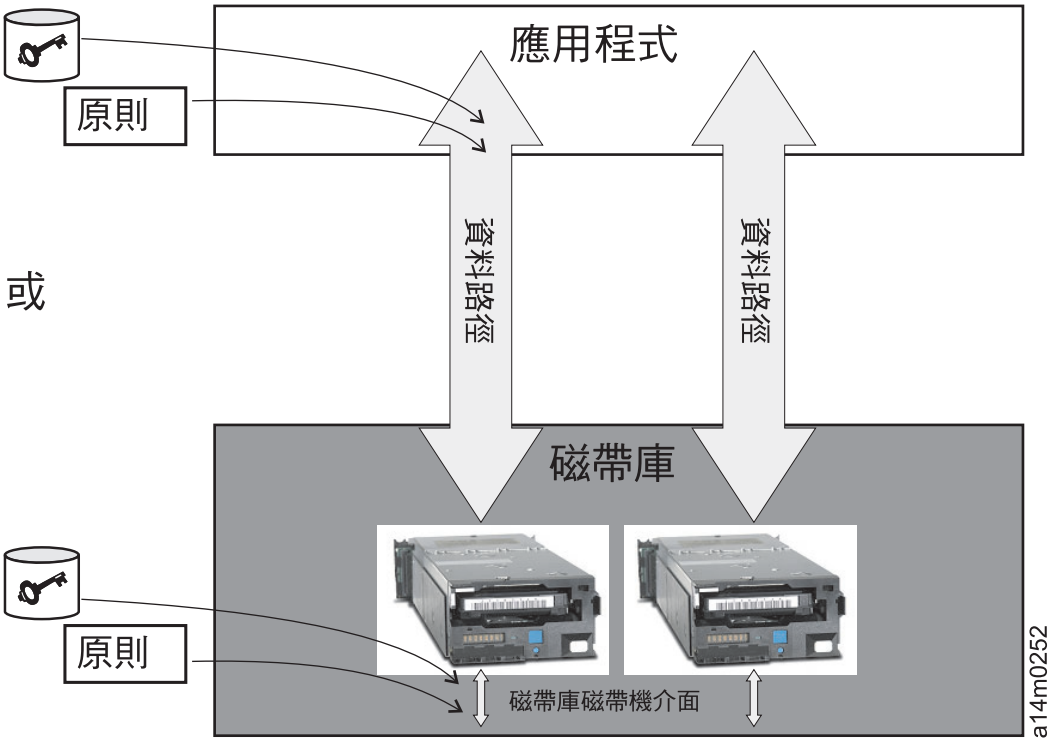

圖 1-2. 加密原則引擎和金鑰管理有兩個可能的位置。

應用程式層

應用程式（有別於金鑰管理程式）會起始磁帶儲存體的資料傳送。請參閱『應用程式管理的磁帶加密』，以瞭解支援的應用程式。

磁帶庫層

磁帶儲存體的外殼，如 Dell PowerVault TL2000/TL4000 和 ML6000 系列。現代磁帶庫含有其中每部磁帶機的內部介面。

應用程式管理的磁帶加密

當執行應用程式的作業環境已經能夠產生和管理加密原則和金鑰時，這個方法最適用。指定何時使用加密的原則是透過應用程式介面來定義的。原則和金鑰會通過應用程式層和加密磁帶機之間的資料路徑。加密是應用程式與啓用加密之磁帶機的互動結果，系統和磁帶庫層完全不需要改變。由於應用程式會管理加密金鑰，因此，使用應用程式方法所寫入和加密的磁區，只能由撰寫它們的相同應用程式利用應用程式管理的加密方法來讀取。

應用程式管理的磁帶加密不需要，也不使用 Encryption Key Manager。

可用來管理加密的最低應用程式版本如下：

- CommVault Galaxy 7.0 SP1
- Symantec Backup Exec 12

下列各項的 LTO 4 和 LTO 5 磁帶機支援應用程式管理的磁帶加密：

- Dell™ PowerVault™ TL2000 磁帶庫
- Dell™ PowerVault™ TL4000 磁帶庫
- Dell™ PowerVault™ ML6000 磁帶庫

請參閱磁帶備份軟體應用程式文件，以學習如何管理加密原則和金鑰。

磁帶庫管理的磁帶加密

請將這個適用於 LTO 4 和 LTO 5 磁帶機 的方法，用於 Dell™ PowerVault™ TL2000 磁帶庫、Dell™ PowerVault™ TL4000 磁帶庫 或 Dell™ PowerVault™ ML6000 磁帶庫。金鑰的產生和管理由 Encryption Key Manager 來執行，它是一個 Java 應用程式，在連接磁帶庫的主機上執行的 Java 應用程式。原則控制和金鑰在磁帶庫至磁帶機介面之間是透通的，因此，對應用程式而言，加密也是透通的。

關於加密金鑰

加密金鑰是專爲使資料混亂和還原而產生的隨機位元字串。加密金鑰是利用爲了確保每個金鑰是唯一與無法預期而設計的演算法來建立的。依照這個方式建構的金鑰越長，加密編碼的破解也就越難。IBM 和 T10 兩種加密方法都利用 256 位元 AES 演算法金鑰來加密資料。256 位元 AES 是美國政府目前承認和建議的加密標準，它接受三種不同的金鑰長度。256 位元金鑰是 AES 所容許的最長金鑰。

Encryption Key Manager 使用兩種類型的加密演算法：對稱演算法和非對稱演算法。對稱或私密金鑰加密的加密和解密都使用單一金鑰。對稱金鑰加密通常用來以有效的方式加密大量資料。256 位元 AES 金鑰是對稱金鑰。非對稱或公開/私密加密使用金鑰組。利用某一金鑰加密的資料，只能用公開/私密金鑰組中的另一金鑰來解密。產生非對稱金鑰組之後，公開金鑰用來加密，而私密金鑰用來解密。

Encryption Key Manager 同時使用對稱和非對稱金鑰；對稱加密用來高速加密使用者或主要資料，非對稱加密（必然較慢）用來保護對稱金鑰。

Encryption Key Manager 的加密金鑰可以由 Keytool 之類的公用程式產生。產生 AES 金鑰的責任及 AES 金鑰傳送給磁帶機的方式，會隨著加密管理的方法而不同。不過，瞭解 Encryption Key Manager 及其他應用程式使用加密金鑰的方式有何不同，可能會很有用。

Dell Encryption Key Manager 的加密金鑰處理

在磁帶庫管理的磁帶加密中，未加密的資料會傳送到 LTO 4 或 LTO 5 磁帶機，再利用 Encryption Key Manager 可用的金鑰儲存庫所預先產生的對稱「資料金鑰 (DK)」來轉換成密文，然後再寫入磁帶中。Encryption Key Manager 以循環方式來選取預先產生的 DK。未產生足夠數量的 DK 時，會在多個磁帶匣上重複使用 DK。Encryption Key Manager 以加密或封裝的形式，將 DK 傳送至 LTO 4 或 LTO 5 磁帶機。LTO 4 和 LTO 5 磁帶機會解除封裝此 DK，並用它來執行加密或解密。不過，封裝金鑰不會儲存在 LTO 4 或 LTO 5 磁帶匣上的任何位置。 在寫好已加密的磁區之後，DK 必須能夠根據別名或金鑰標籤來存取，且可供 Encryption Key Manager 使用，以便讀取磁區。第 1-6 頁的圖 1-3 說明這個程序。

Dell Encryption Key Manager 也提供了將 LTO 加密的對稱金鑰組織成金鑰群組的功能。依照這個方式，您便可以根據加密的資料類型、有權存取的使用者或其他有意義的性質來分組金鑰。請參閱第 3-13 頁的『建立和管理金鑰群組』，以取得詳細資訊。

其他應用程式的加密金鑰處理

在應用程式管理的磁帶加密中，未加密的資料會傳送到 LTO 4 和 LTO 5 磁帶機，且會利用應用程式提供的對稱 DK 轉換爲密文，然後再寫入磁帶中。DK 不會儲存在磁帶匣的任何位置。在寫好已加密的磁區之後，DK 必須放在應用程式能夠存取的位置，例如伺服器資料庫，以便讀取磁區。

LTO 4 和 LTO 5 磁帶機可以利用 Yosemite （適用於 Dell PowerVault TL2000 和 TL4000 磁帶庫）、CommVault 和 Symantec Backup Exec 之類的應用程式來進行應用程式管理的加密。

另外， LTO 4 和 LTO 5 磁帶機也可供使用 T10 指令集的應用程式來執行加密。T10 指令集使用應用程式所提供對稱的 256 位元 AES 金鑰。T10 可以每個磁帶匣使用多個唯一的 DK，甚至可以將加密資料和明碼資料寫到相同的磁帶匣中。當應用程式將磁帶匣加密時，它會利用應用程式所決定的方法來選取或產生一個 DK，並將它傳送到磁帶機。這個金鑰**不會**以非對稱公開金鑰進行封裝，**不會**儲存在磁帶匣中。在已加密的資料寫到磁帶之後，DK 必須放在應用程式能夠使用的位置，以便讀取資料。

圖 1-3 顯示應用程式管理和磁帶庫管理加密的磁帶加密程序。

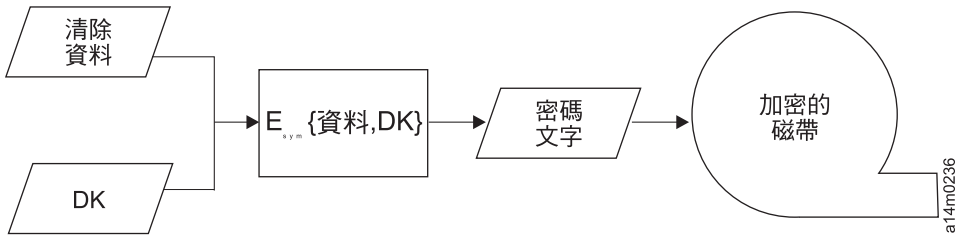

圖 1-3. 利用對稱加密金鑰加密. LTO 4 和 LTO 5 磁帶機上的磁帶庫管理和應用程式管理加密。

摘要

每個磁區所能使用的加密金鑰數目，會隨著磁帶機、加密標準和用來管理加密的方法而不同。對於 LTO 4 和 LTO 5 的透通加密而言，（也就是搭配 Encryption Key Manager來使用磁帶庫管理的加密）DK 的唯一性會隨著 Encryption Key Manager使用之預先產生的金鑰數目是否足夠而不同。

表 1-1. 加密金鑰摘要

	金鑰使用者	
加密管理方法	**IBM 加密**	**T10 加密**
磁帶庫管理加密	1 DK / 卡匣	N/A
應用程式管理加密	多重 DK / 卡匣	多重 DK / 卡匣
DK = 對稱 AES 256 位元 DK		

第 2 章 規劃 Encryption Key Manager 環境

這一節的目的在提供一些資訊來協助您判斷最適合您需求的 Encryption Key Manager 配置。當您規劃如何設定加密策略時，必須考量許多因素。

加密設定作業一覽

在使用磁帶機的加密功能之前，必須符合特定的軟硬體需求。下列核對清單可用來協助您符合這些需求。

Encryption Key Manager 設定作業

在磁帶加密之前，Encryption Key Manager 必須先配置好且在執行中，才能與加密磁帶機通訊。安裝磁帶機時，不一定要執行 Encryption Key Manager，但它必須在執行中，才能執行加密。

- 決定用來作爲 Encryption Key Manager 伺服器的系統平台。
- 如有必要，請升級伺服器作業系統。（請參閱第 2-2 頁的『軟硬體需求』。）
- 安裝 Java 無限制原則檔。（請參閱第 2-2 頁的『軟硬體需求』。）
- 升級 Encryption Key Manager JAR。（請參閱 第 3-1 頁的『下載最新版本的 Key Manager ISO Image』。）
- 建立金鑰、憑證和金鑰群組。

  第 3-5 頁的『利用 GUI 來建立配置檔、金鑰儲存庫和憑證』

  第 3-13 頁的『建立和管理金鑰群組』
- 如果您遵循第 3-5 頁的『利用 GUI 來建立配置檔、金鑰儲存庫和憑證』中的程序，除非您想要使用其他配置選項，否則不需要這些步驟：
  - 如有必要，請匯入金鑰和憑證。（請參閱 第 3-12 頁的『利用 Keytool -importseckey 匯入資料金鑰』。）
  - 定義配置內容檔。（請參閱第 4-1 頁的第 4 章, 『配置 Encryption Key Manager』。）
  - 向 Encryption Key Manager 定義磁帶機，或將 **drive.acceptUnknownDrives** 配置內容值設爲 on。（請參閱第 5-7 頁的『adddrive』來明確定義磁帶機，或參閱第 4-1 頁的『自動更新磁帶機表格』。）
  - 啓動 Encryption Key Manager 伺服器。（請參閱第 5-1 頁的『啓動、重新整理和停止金鑰管理程式伺服器』。）
  - 啓動指令行介面用戶端。（請參閱 第 5-5 頁的『指令行介面用戶端』。）

規劃磁帶庫管理的磁帶加密

如果要執行加密，您需要：

- 具備加密功能的 LTO 4 和 LTO 5 磁帶機
- 金鑰儲存庫
- Dell Encryption Key Manager

磁帶庫管理的磁帶加密作業

1. 安裝和連接 LTO 4 和 LTO 5 磁帶機。
   - 更新磁帶庫韌體（TL2000、TL4000、ML6000，如有必要）。請造訪 **http://support.dell.com**。
     - Dell™ PowerVault™ TL2000 磁帶庫 所需要的最低韌體版本 = 5.xx。
     - Dell™ PowerVault™ TL4000 磁帶庫 所需要的最低韌體版本 = 5.xx。
     - Dell™ PowerVault™ ML6000 磁帶庫 系列所需要的最低韌體版本是 = 415G.xxx。
   - 如有必要，請更新磁帶機韌體。所需要的最低韌體版本是 77B5。
2. 啓用 LTO 4 和 LTO 5 磁帶機和磁帶庫的磁帶庫管理磁帶加密（請參閱您的 Dell 磁帶庫資訊，以取得詳細資料）。
   - 新增 Encryption Key Manager 伺服器 IP 位址
3. 利用磁帶庫診斷功能來驗證 Encryption Key Manager 路徑和加密配置（請參閱您的 Dell 磁帶庫資訊，以取得詳細資料）。

軟硬體需求

註: 只有下列各平台的 IBM 版 Java Runtime Environment (JRE) 支援 Encryption Key Manager。

Linux 解決方案元件

作業系統

- RHEL 4
- RHEL 5
- SLES 9
- SLES 10
- SLES 11

Encryption Key Manager（在 Linux 上執行）

表 2-1. Linux 的基本軟體需求

平台	IBM Software Developer Kit	位置如下：
64 位元 AMD/Opteron/EM64T	Java 6.0 SR5	**http://support.dell.com**
32 位元 Intel® 相容		

磁帶庫

在 Dell PowerVault TL2000 磁帶庫、TL4000 磁帶庫和 ML6000 磁帶庫方面，請確定韌體是最新可用的層次。關於韌體更新，請造訪 **http://support.dell.com**。

磁帶機

在 LTO 4 和 LTO 5 磁帶機方面，請確定韌體是最新可用的層次。關於韌體更新，請造訪 **http://support.dell.com**。

Windows 解決方案元件

作業系統

Windows Server 2003、2008 和 2008 R2

Dell Encryption Key Manager

所需要的 Encryption Key Manager 最低版本是 2.1，建置日期是 20070914 或以後，以及下列 IBM Runtime Environment 之一：

表 2-2. Windows 的基本軟體需求

作業系統	IBM Runtime Environment
Windows 2003	• IBM® 64-bit Runtime Environment for Windows，AMD64/EM64T 架構，Java 2 Technology Edition 5.0 版 SR5 • IBM 32-bit Runtime Environment for Windows，Java 2 Technology Edition 5.0 版 SR5
Windows 2008 和 2008 R2	IBM 64-bit Runtime Environment for Windows，AMD64/EM64T 架構，Java 2 Technology Edition 6.0 版 SR5

磁帶庫

在Dell™ PowerVault™ TL2000 磁帶庫、Dell™ PowerVault™ TL4000 磁帶庫 和 Dell™ PowerVault™ ML6000 磁帶庫 方面，請確定韌體是最新可用的層次。關於韌體更新，請造訪 **http://support.dell.com**。

磁帶機

在 LTO 4 和 LTO 5 磁帶機方面，請確定韌體是最新可用的層次。關於韌體更新，請造訪 **http://support.dell.com**。

金鑰儲存庫考量

保留金鑰儲存庫的資料極具重要性，不容忽視。當無法存取金鑰儲存庫時，您也無法將加密的磁帶解密。請仔細閱讀下列主題，以瞭解可用來保護金鑰儲存庫資料的方法。

JCEKS 金鑰儲存庫

EKM 支援 JCEKS 金鑰儲存庫類型。

JCEKS（Unix 系統服務檔案型）是所有執行 EMK 的平台所支援的檔案型金鑰儲存庫。因此您可以輕鬆複製此金鑰儲存庫的內容以進行備份及回復，並保持兩個 EKM 實例同步化，以因應失效接手。JCEKS 爲安全提供了密碼型的金鑰儲存庫內容保護，也提供了相對較好的效能。可以使用 FTP 之類的檔案複製方法。

加密金鑰及 LTO 4 和 LTO 5 磁帶機

Dell Encryption Key Manager 及其支援的磁帶機都利用對稱的 256 位元 AES 金鑰來加密資料。這個主題說明您所應該瞭解的這些金鑰和憑證的相關事項。

在 LTO 磁帶匣的 LTO 4 或 LTO 5 磁帶機上執行加密作業時，Encryption Key Manager 只會使用 256 位元的 AES對稱資料金鑰。

當 LTO 4 或 LTO 5 要求金鑰時，Encryption Key Manager 會使用指定給磁帶機的別名。如果未指定任何別名給磁帶機，便會使用 symmetricKeySet 配置內容所指定之金鑰群組、金鑰別名清單或金鑰別名範圍中的別名。欠缺磁帶機的特定別名，便會以循環方式，從其他實體中選取別名，以均勻平衡金鑰的使用。

所選的別名會關聯於金鑰儲存庫中預先載入的對稱「資料金鑰 (DK)」。Encryption Key Manager 會將這個 DK（封裝了磁帶機能夠解密的不同金鑰）傳送給 LTO 4 或 LTO 5 磁帶機，以便將資料加密。這個 DK 並不以明碼方式透過 TCP/IP 來傳輸。所選的別名也會轉換成稱爲「資料金鑰 ID (DKi)」的實體，它會隨著加密資料寫入磁帶中。依照這個方式，當讀取 LTO 4 或 LTO 5 磁帶時，Encryption Key Manager 便能夠利用 DKi 來識別將資料解密所需要的正確 DK。

第 5-7 頁的『CLI 指令』中的 **adddrive** 和 **moddrive** 主題說明如何指定磁帶機的別名。請參閱第 3-9 頁的『在 LTO 4 和 LTO 5 產生金鑰和別名以進行加密』，其中包括在 symmetricKeySet 配置內容中匯入金鑰、匯出金鑰和指定預設別名的相關資訊。 第 3-13 頁的『建立和管理金鑰群組』顯示如何定義金鑰群組，以及如何將金鑰儲存庫中的別名移入其中。

圖 2-1 顯示如何處理加密寫入作業的金鑰。

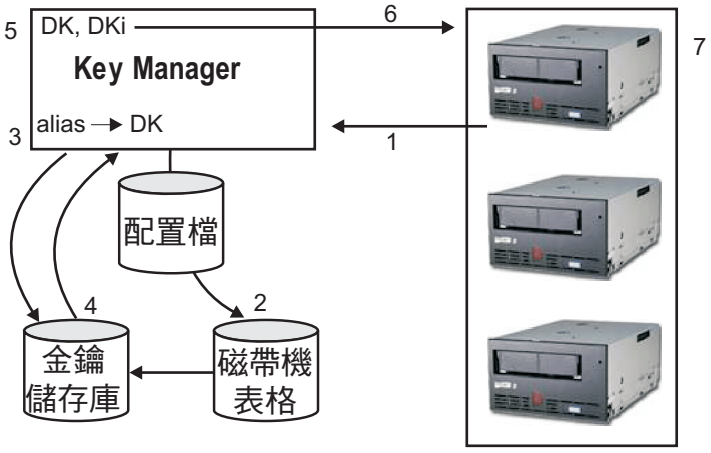

圖 2-1. LTO 4 或 LTO 5 磁帶機的加密寫入作業要求

1. 磁帶機要求金鑰來加密磁帶
2. Encryption Key Manager 驗證磁帶機表格中的磁帶裝置
3. 如果要求未指定任何別名，磁帶機表格也未指定任何別名，Encryption Key Manager會從 keyAliasList 中的別名集或金鑰群組中選取一個別名。
4. Encryption Key Manager 會從金鑰儲存庫中提取一個相對應的 DK。
5. Encryption Key Manager 將別名轉換成 DKi，並將 DK 與磁帶機能夠解密的金鑰一起封裝起來

6. Encryption Key Manager 將 DK 和 DKi 傳送到磁帶機
7. 磁帶機將 DK 解開，將加密的資料和 DKi 寫入磁帶中

圖 2-2 顯示如何處理加密讀取作業的金鑰。

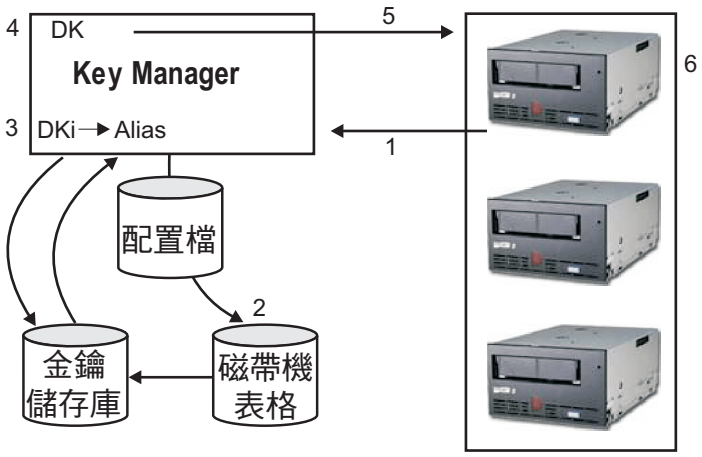

圖 2-2. LTO 4 或 LTO 5 磁帶機的加密讀取作業要求

1. 磁帶機收到讀取要求，將 DKi 傳送到 Encryption Key Manager
2. Encryption Key Manager 驗證磁帶機表格中的磁帶裝置
3. Encryption Key Manager 將 DKi 轉換成別名，再從金鑰儲存庫提取對應的 DK
4. Encryption Key Manager 將 DK 與磁帶機能夠解密的金鑰一起封裝起來
5. Encryption Key Manager 將封裝的 DK 傳送到磁帶機
6. 磁帶機將 DK 解開，利用它來將資料解密

備份金鑰儲存庫資料

註: 由於金鑰儲存庫中的金鑰相當重要，將這項資料備份在非加密裝置中，也非常重要，您才能在必要時回復它，或讀取這部磁帶機或磁帶庫的相關憑證所加密的磁帶。金鑰儲存庫若未適當備份，將會造成完全無法存取已加密的資料，且無可彌補。

您可以利用許多方式來備份這個金鑰儲存庫資訊。每個金鑰儲存庫類型都有它自己專有的性質。全體適用的一般準則如下：

- 所有已載入金鑰儲存庫的憑證都各保留一份副本（通常是 PKCS12 格式檔）。
- 利用系統備份功能（如 RACF）來建立金鑰儲存庫資訊的備份副本（請小心避免利用加密磁帶機來加密這個副本，因爲這會造成無法解密而無法進行回復）。
- 維護主要和次要 Encryption Key Manager 及金鑰儲存庫副本（用於備份及失效接手備援）。主要和次要的金鑰儲存庫都要備份，以增加備援能力。
- 對於 JCEKS 金鑰儲存庫，只要複製金鑰儲存庫檔，將明碼（未加密）副本儲存在保險庫之類的安全位置即可（請小心避免利用加密磁帶機來加密這個副本，因爲這會造成無法解密而無法進行回復）。

每當金鑰儲存庫資料有了變更，您至少應該備份一次。Encryption Key Manager 不會修改金鑰儲存庫資料。除了您套用的變更之外，金鑰儲存庫別無其他變更，因此，在變更金鑰儲存庫之後，請立刻進行複製。

利用 GUI 備份檔案

1. 如果 GUI 尚未啓動，請開啓它：

   Windows

   導覽至 c:\ekm\gui，並按一下 **LaunchEKMGui.bat**

   Linux 平台

   導覽至 /var/ekm/gui 然後輸入 `. ./LaunchEKMGui.sh`

2. 在 Encryption Key Manager GUI 左側導覽器中，選取 **Backup Critical Files**。
3. 在顯示的對話框中，輸入備份資料的路徑 (圖 2-3)。

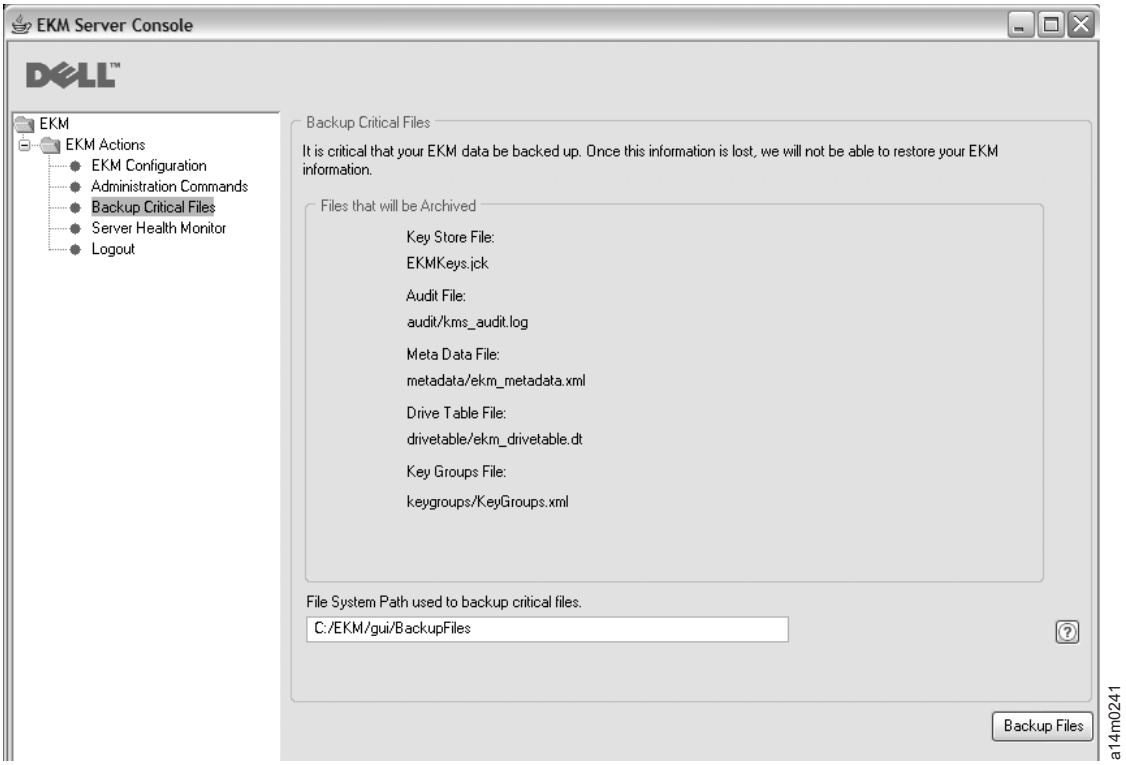

圖 2-3. Backup Critical Files 視窗

4. 按一下 **Backup Files**。
5. 這時會出現一則參考訊息來顯示結果。

多重備援金鑰管理程式

Encryption Key Manager 設計成能夠使用磁帶機和磁帶庫來提供備援，以提供高可用性，因此，您可以用多個金鑰管理程式來處理相同的磁帶機和磁帶庫。此外，這些金鑰管理程式也不需要在與磁帶機和磁帶庫的相同系統上。金鑰管理程式的數目上限會隨著磁帶庫或 Proxy 而不同。唯一需求是磁帶機能夠透過 TCP/IP 連線功能來使用它們。

這個方式可讓您擁有兩個互爲鏡映影像且內建金鑰儲存庫重要資訊備份的 Encryption Key Manager，以及在一個金鑰管理程式失去作用時的失效接手功能。當您配置裝置（或

Proxy）時，您可以將它指向兩個金鑰管理程式。如果一個金鑰管理程式因故無法使用，您的裝置（或磁帶庫）就使用替代的金鑰管理程式。

另外，您也可以將兩個 Encryption Key Manager 保持同步。在必要時使用這個重要功能，非常重要，一方面是它本來就能夠備份重要資料，另一方面也在於它的失效接手功能可使磁帶作業免於中斷。請參閱第 4-2 頁的『將兩部金鑰管理程式伺服器的資料同步化』。

註: 同步化不包括金鑰儲存庫。它們必須以手動方式來複製。

Encryption Key Manager 伺服器配置

Encryption Key Manager 可以安裝在單一伺服器或多重伺服器上。下列範例顯示一個及兩個金鑰管理程式的配置，但您的磁帶庫可能容許更多管理程式配置。

單一伺服器配置

單一伺服器配置（如圖 2-4 所示）是最簡易的 Encryption Key Manager 配置。不過，由於缺乏備援，不建議採用。在這個配置中，所有磁帶機都依賴單一金鑰管理程式伺服器，沒有任何備份。如果伺服器關閉，便無法使用金鑰儲存庫、配置檔、KeyGroups.xml 檔以及磁帶機表格，也會因而無法讀取任何已加密的磁帶。在單一伺服器配置中，您必須確定已在 Encryption Key Manager 以外的安全位置，維護了金鑰儲存庫、配置檔、KeyGroups.xml 檔以及磁帶機表格的備份副本，當遺失伺服器副本時，才能在置換伺服器上重建它的功能。

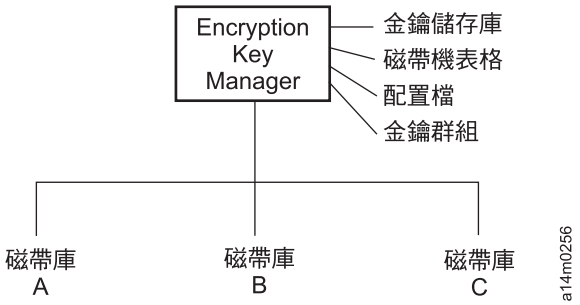

圖 2-4. 單一伺服器配置

雙伺服器配置

建議您採用雙伺服器配置。當主要金鑰管理程式因故無法存取時，這個 Encryption Key Manager 配置會自動進行失效接手，交給次要金鑰管理程式。

註: 利用不同的 Encryption Key Manager 伺服器來處理同一組磁帶機所發出的要求時，相關金鑰儲存庫中的資訊必須相同。不論連接的是哪一部金鑰管理程式伺服器，都能取得必要資訊來支援磁帶機的要求，所以此爲必要條件。

相同配置: 在兩部 Encryption Key Manager 伺服器含有相同配置的環境（如第 2-8 頁的圖 2-5 所示）中，如果主要金鑰管理程式關閉，處理程序會自動進行失效接手，交給次要金鑰管理程式。在這類配置中，兩部金鑰管理程式伺服器的同步化非常重要。您可以利用 **sync** 指令，將金鑰管理程式伺服器配置檔和磁帶機表格的更新，自動複製到另一部金鑰管理程式伺服器中，但金鑰儲存庫的更新，必須以使用的金鑰儲存庫專用

方法，才能複製到其他金鑰儲存庫中。金鑰儲存庫和金鑰群組 XML 檔必須以手動方式複製。請參閱第 4-2 頁的『將兩部金鑰管理程式伺服器的資料同步化』，以取得詳細資訊。

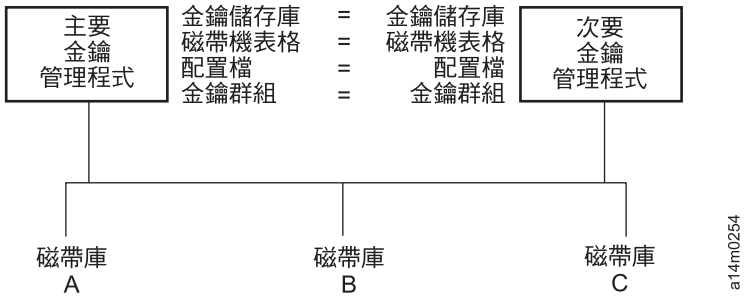

圖 2-5. 兩部伺服器共用配置

區分配置: 兩部 Encryption Key Manager 伺服器可以分享共同的金鑰儲存庫和磁帶機表格，且具有不同的兩個配置檔，而在各自的 XML 檔中，也分別定義兩組不同的金鑰群組。唯一需求是每部伺服器用來處理共用磁帶機的金鑰必須相同。這使得每部金鑰管理程式伺服器能夠有自己的一組內容。在這類型的配置中（如圖 2-6 所示），在金鑰管理程式伺服器之間，只有磁帶機表格應該同步化。（請參閱第 4-2 頁的『將兩部金鑰管理程式伺服器的資料同步化』，以取得詳細資訊。）請務必指定 `sync.type = drivetab`（不指定 config 或 all），以防止改寫配置檔。

註: 伺服器的配置無法局部分享。

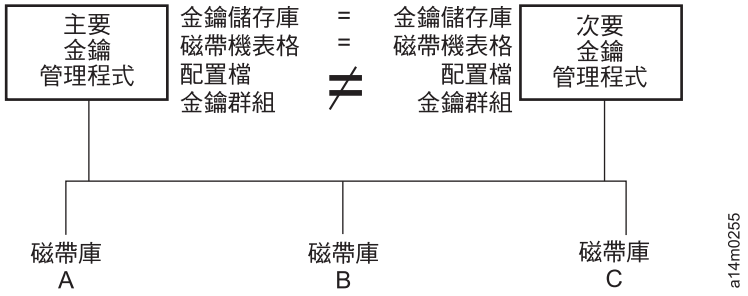

圖 2-6. 兩部配置不同的伺服器而存取相同的裝置

災難回復站台考量

如果您打算使用災難回復 (DR) 站台，Encryption Key Manager 提供了若干選項，使這個站台能夠讀取和寫入加密的磁帶。這些選項如下：

- 在 DR 站台建立 Encryption Key Manager 複本。

  在 DR 站台上，利用本端 Encryption Key Manager 的相同資訊（配置檔、磁帶機表格、金鑰群組 XML 檔，以及金鑰儲存庫）來設定 Encryption Key Manager 複本。之後，這個金鑰管理程式便已備妥，能夠接替其中一個現有正式作業金鑰管理程式來讀取和寫入加密的磁帶。
- 依照需要，建立三個 Encryption Key Manager 資料檔的備份副本，以便進行回復。

如果您建立 Encryption Key Manager 所需要的四個資料元素（配置檔、磁帶機表格、金鑰群組 XML 檔，以及金鑰儲存庫）的現行副本，您便能夠隨時啓動一個金鑰管理程式來作爲在 DR 站台的複本。（請記住，您不應利用 Encryption Key Manager 來加密這些檔案的副本，因爲沒有能夠運作的金鑰管理程式，便無法將它解密）。如果您的 DR 站台使用不同於主要站台的磁帶機，配置檔和磁帶機表格必須包含 DR 站台的正確資訊。

離站共用加密磁帶考量

註: 在取得商業夥伴的任何憑證之後，可將信任鏈往回檢查到最後簽署這個憑證的憑證管理中心 (CA)，以驗證這個憑證的有效性，這一點很重要。如果您信任這個 CA，便可以信任這個憑證。另外，如果憑證在傳輸中受到安全的保護，憑證的有效性也能夠得到驗證。這些方式若發生無法驗證憑證有效性的情況，可能開啓導致來「中間人 (Man-in-the-Middle)」攻擊。

共用 LTO 4 和 LTO 5 磁帶

如果要共用 LTO 4 或 LTO 5 磁帶上的加密資料，磁帶上用來加密資料的對稱金鑰副本也應該提供給其他組織，使他們能夠讀取磁帶。如果要共用對稱金鑰，其他組織也必須與您共用他們的公開金鑰。利用 Keytool 從 Encryption Key Manager 金鑰儲存庫匯出對稱金鑰時，這個公開金鑰將用來封裝對稱金鑰（請參閱第 3-12 頁的『利用 Keytool -exportseckey 匯出資料金鑰』）。當其他組織將對稱金鑰匯入 Encryption Key Manager 金鑰儲存庫時，會利用相對應的私密金鑰（請參閱第 3-12 頁的『利用 Keytool -importseckey 匯入資料金鑰』）將它解開。這可以確保在傳輸過程中，對稱金鑰是安全的，因爲只有私密金鑰的持有者才能夠將對稱金鑰解開。有了在 Encryption Key Manager 金鑰儲存庫內用來加密資料的對稱金鑰，其他組織便能夠讀取磁帶上的資料。

聯邦資訊存取安全標準 (FIPS) 140-2 注意事項

「聯邦資訊存取安全標準 (FIPS) 140-2」現在已日趨重要，美國聯邦政府要求它的所有加密提供者通過 FIPS 140 的驗證。在日漸增長的民間部門社群中，也採用了這個標準。在這個安全意識日漸高漲的世界裡，協力廠商根據政府標準來提供加密功能憑證，也令人感到更具價值。

Encryption Key Manager 本身並不提供加密功能，因此，不需要也不能取得 FIPS 140-2 憑證。不過，Encryption Key Manager 在 IBM Java 加密延伸元件中運用了 IBM JVM 的加密功能，可供選擇及使用具備 FIPS 140-2 層次 1 憑證的 IBMJCEFIPS 加密提供者。您將配置內容檔中的 **fips** 配置參數設爲 **on** 之後，Encryption Key Manager 會將 IBMJCEFIPS 提供者用在所有加密功能上。

請參閱特定軟硬體加密提供者的文件，以取得其產品是否通過 FIPS 140-2 驗證的相關資訊。

第 3 章 安裝 Encryption Key Manager 和金鑰儲存庫

Encryption Key Manager 檢附於 IBM Java 虛擬機器安裝架構，需要 IBM Software Developer Kit for Linux 以及 IBM Runtime Environment for Windows（請參閱第 2-2 頁的『軟硬體需求』）。請遵循作業系統所適用的程序：

- 『在 Linux 上安裝 Encryption Key Manager』
- 第 3-2 頁的『在 Windows 上安裝 Encryption Key Manager』

如果不確定您的 Encryption Key Manager 是否爲最新版本，『下載最新版本的 Key Manager ISO Image』會告訴您如何判斷是否有新版可用。建議您取得最新版本的 Encryption Key Manager，因爲它可能未包含在 Java 安裝架構中。 請造訪 **http://support.dell.com**，以取得詳細資訊。

重要 Encryption Key Manager 主機伺服器配置資訊：建議您讓代管 Dell Encryption Key Manager 程式的機器使用 ECC 記憶體，將資料遺失的風險降低到最小。Encryption Key Manager 會執行要求產生加密金鑰及將這些金鑰傳給 LTO 4 和 LTO 5 磁帶機的功能。在 Encryption Key Manager 的處理期間，封裝（加密形式）的金鑰資料是在系統記憶體中。請注意，金鑰資料必須無誤地傳送到適當的磁帶機，才能夠回復（解密）寫在卡匣上的資料。如果由於某些原因，造成金鑰資料因系統記憶體位元錯誤而毀損，且該金鑰資料是用來將資料寫入卡匣中，則寫入這個卡匣的資料將無法復原（日後無法解密）。有一些適當的防護措施可確保不會發生這類的資料錯誤。不過，如果代管 Encryption Key Manager 的機器並未使用錯誤更正碼 (ECC) 記憶體，在系統記憶體內的金鑰資料仍有可能毀損，因而造成資料遺失。發生這個情況的機會不大，但對於代管重要應用程式（如 Encryption Key Manager）的機器，一律建議使用 ECC 記憶體。

下載最新版本的 Key Manager ISO Image

如果要下載最新版本的 Dell ISO image，請移至http://support.dell.com。

在 Linux 上安裝 Encryption Key Manager

在 Linux 上，從 CD 安裝 Encryption Key Manager

1. 插入 Dell Encryption Key Manager CD，從 CD 根目錄中輸入 `Install_Linux`。

   安裝作業會將作業系統適用的所有內容（文件、GUI 檔和配置內容檔），從 CD 複製到您的硬碟中。安裝期間，會檢查系統來找出正確的 IBM Java Runtime Environment。如果找不到，會自動安裝它。

   安裝好之後，會啓動圖形使用者介面 (GUI)。

在 Linux 上，手動安裝 Software Developer Kit

如果您不是從 CD 安裝，請遵循這些步驟。

1. 從 **http://support.dell.com**中，根據您的作業系統來下載正確的 Runtime Environment for Java：
   - Java 6 SR 5（32 位元）或更新的版本

- Java 6 SR 5（64 位元）或更新的版本

2. 將 Java linux rpm 檔放在某個工作目錄中：

```
mordor:~ #/tape/Encryption/java/1.6.0# pwd
/tape/Encryption/java/1.6.0
mordor:~ #/tape/Encryption/java/1.6.0# ls
ibm-java-i386-jre-6.0-5.0.i386.rpm
```

3. 安裝 rpm 套件：

```
mordor:~ #rpm -ivh -nodeps ibm-java-i386-jre-6.0-5.0.i386.rpm
```

這會將檔案置於 **/opt/ibm/java-i386-60/** 目錄中：

```
mordor:~ #/opt/ibm/java-i386-60/jre # ls
.systemPrefs  bin  javaws  lib
```

4. 以所安裝之 Java 的 JAVA_HOME、CLASSPATH 和 bin 目錄來編輯（或建立，必要的話）**/etc/profile.local** 檔。新增下面這三行：

```
JAVA_HOME=/opt/ibm/java-i386-60/jre
CLASSPATH=/opt/ibm/java-i386-60/jre/lib
PATH=$JAVA_HOME:opt/ibm/java-i386-60/jre/bin/:$PATH
```

5. 登出再重新登入主機，使 **/etc/profile.local** 項目生效，或發出 export 指令行指令：

```
mordor:~ # export JAVA_HOME=/opt/ibm/java-i386-60/jre
mordor:~ # export CLASSPATH=/opt/ibm/java-i386-60/jre/lib
mordor:~ # export PATH=/opt/ibm/java-i386-60/jre/bin/:$PATH
```

6. 重新登入之後，發出 **java -version** 指令。您應該會見到下列結果：

```
mordor:~ # java -version
java version "1.6.0"
Java(TM) SE Runtime Environment (build pmz60sr5-20090529(SR5))
IBM J9 VM (build 2.4, J2RE 1.6.0 IBM J9 2.4 Linux x86-32 jvmxi3260-20090519_35743 (JIT enabled)
...
mordor:~ # which java
/opt/ibm/java-i386-60/jre/bin/java
```

在 Windows 上安裝 Encryption Key Manager

1. 插入 Dell Encryption Key Manager CD。

   安裝作業會將作業系統適用的所有內容（文件、GUI 檔和配置內容檔），從 CD 複製到您的硬碟中。安裝期間，會檢查系統來找出正確的 IBM Java Runtime Environment。如果找不到，會自動安裝它。

   安裝好之後，會啓動圖形使用者介面 (GUI)。

2. InstallShield Wizard 開啓之後，按 **Next**。
3. 閱讀 License Agreement，按一下 **Yes**。
4. Choose Destination Location 視窗開啓之後 (第 3-3 頁的圖 3-1)，選擇一個資料夾，將它記下來。您需要這個 Java 路徑，才能啓動 Encryption Key Manager。

圖 3-1. Choose Destination Location 視窗

按 **Next**。

5. 這時會開啓一個視窗，詢問您是否要以這個 Java Runtime Environment 爲預設的系統 JVM (圖 3-2)。

圖 3-2. 將這個版本的 JVM 設爲預設值

按一下 **No**。

6. 這時會開啓 Start Copying Files 視窗 (第 3-4 頁的圖 3-3)。請確定您已記下目標目錄。

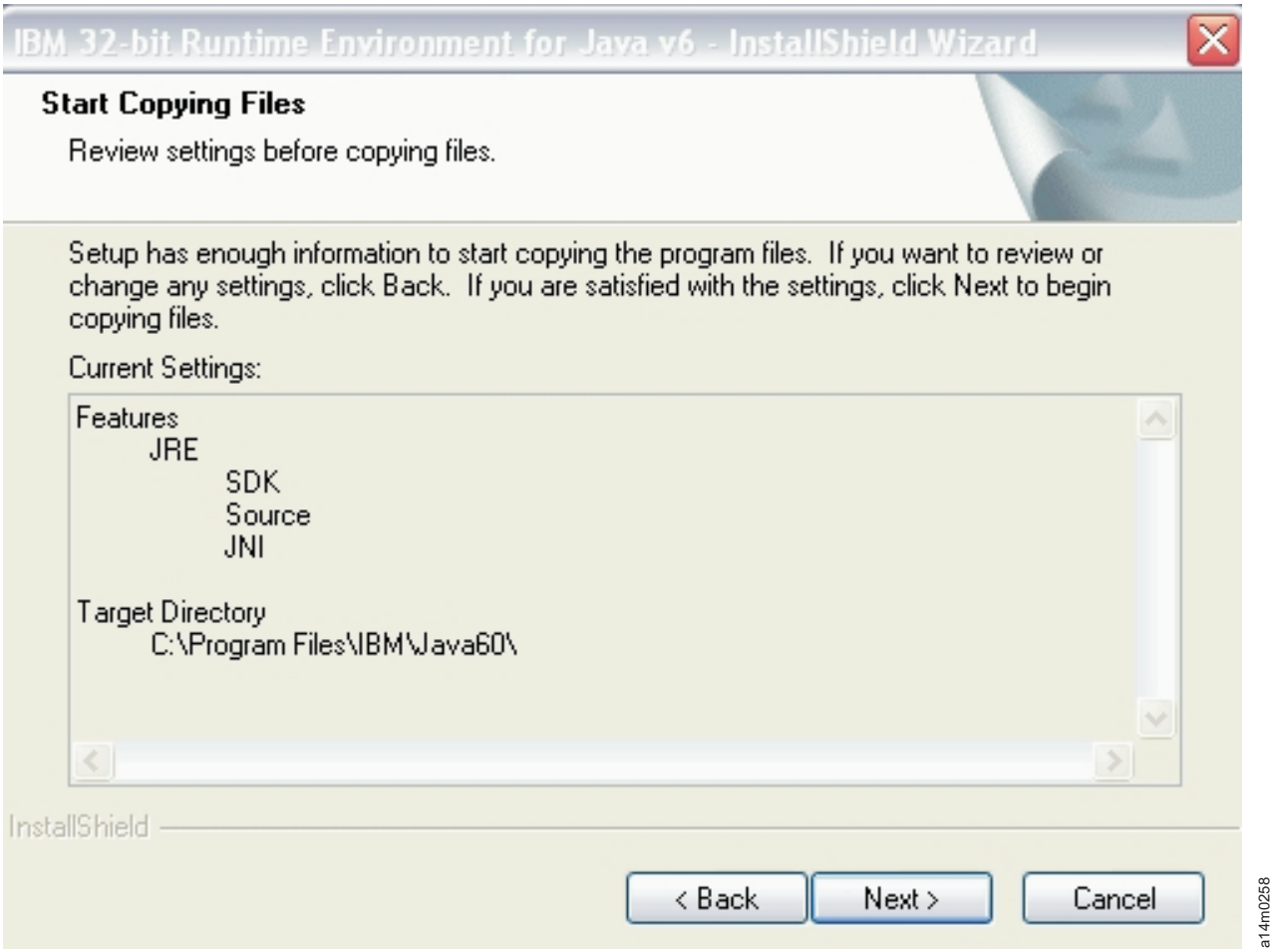

圖 3-3. Start Copying Files 視窗

按 **Next**。

7. 狀態視窗指出安裝進度。
8. 這時會開啓 Browser Registration 視窗。選擇搭配 Encryption Key Manager 使用的瀏覽器。按 **Next**。
9. InstallShield Wizard Complete 視窗開啓之後，按一下 **Finish**。

   安裝好之後，您可以開啓一個命令提示字元來查詢安裝的 Java 版本：

   ```
   C:\WinEKM>C:\"Program Files"\IBM\Java60\jre\bin\java -version
   java version "1.6.0"
   Java(TM) SE Runtime Environment (build pwi3260sr5-20090529_04(SR5))
   IBM J9 VM (build 2.4, J2RE 1.6.0 IBM J9 2.4 Windows Server 2003 x86-32 j9vmwi3223-20090
   519_35743 (JIT enabled, AOT enabled)
   ...
   ```

10. 依下列方式更新 PATH 變數：（對 Encryption Key Manager2.1 是必要的，對 05032007 及更早的建置日期而言是選用的）。

    如果您要從指令視窗呼叫 Java SDK，且希望從任何目錄執行 Java JRE 執行檔 (java.exe)，而不必輸入指令的完整路徑，您可以設定 PATH 變數。如果未設定 PATH 變數，則執行檔每次執行時，都必須指定完整路徑，例如：

    ```
    C:>\Program Files\IBM\Java60\jre\bin\java ...
    ```

如果要永久設定 PATH（對 Encryption Key Manager 2.1 而言是必要的），請新增 Java bin 目錄的完整路徑到 PATH 變數中。這個完整路徑通常看起來如下：

```
C:\Program Files\IBM\Java60\jre\bin
```

如果要在 Microsoft Windows 2003、2008 和 2008 R2 中永久設定 PATH，請執行下列動作：

註: 從指令行設定 PATH 變數無法運作。

a. 從「開始」功能表中，選取**設定**，再選取**控制台**。
b. 按兩下**系統**。
c. 按一下**進階**標籤。
d. 按一下**環境變數**。
e. 捲動「系統變數」清單來找出 Path 變數，並按一下**編輯**。
f. 將 IBM JVM 路徑新增到 Path 變數的開頭。

   預設安裝目錄是 C:\PROGRA~1\IBM\Java60\jre\bin。

   重要事項：在路徑結尾插入分號，將它與路徑清單中的其他目錄區隔開來。
g. 按一下**確定**。

利用 GUI 來建立配置檔、金鑰儲存庫和憑證

在啓動 Encryption Key Manager 之前，您必須先建立至少一個新的金鑰儲存庫，以及至少一個自簽憑證。您可以利用 Dell Encryption Key Manager 伺服器圖形使用者介面 (GUI) 來建立您的 Encryption Key Manager 配置內容檔、金鑰儲存庫、憑證，以及金鑰。這個程序也會同時建立一個簡式 CLI 配置內容檔。

1. 如果 GUI 尚未啓動，請開啓它：

   Windows
   導覽至 c:\ekm\gui，並按一下 **LaunchEKMGui.bat**

   Linux 平台
   導覽至 /var/ekm/gui 然後輸入 `. ./LaunchEKMGui.sh`
2. 在 GUI 左側導覽器中，選取 **EKM Configuration**。

3. 在 『EKM Server Configuration』 頁面 (圖 3-4) 的所有必要欄位（以星號表示）中輸入資料。爲了方便，部分欄位已自動填妥。請按一下任何資料欄位右側的問號來取得說明。按 **Next**。

註: 金鑰儲存庫密碼設定好之後，**請勿加以變更**，除非安全出現漏洞。這些密碼會成爲亂碼，以免出現任何安全漏洞。如果要變更金鑰儲存庫密碼，您必須利用 **keytool** 指令來個別變更這個金鑰儲存庫中的每個密碼。請參閱第 3-11 頁的『變更金鑰儲存庫密碼』。

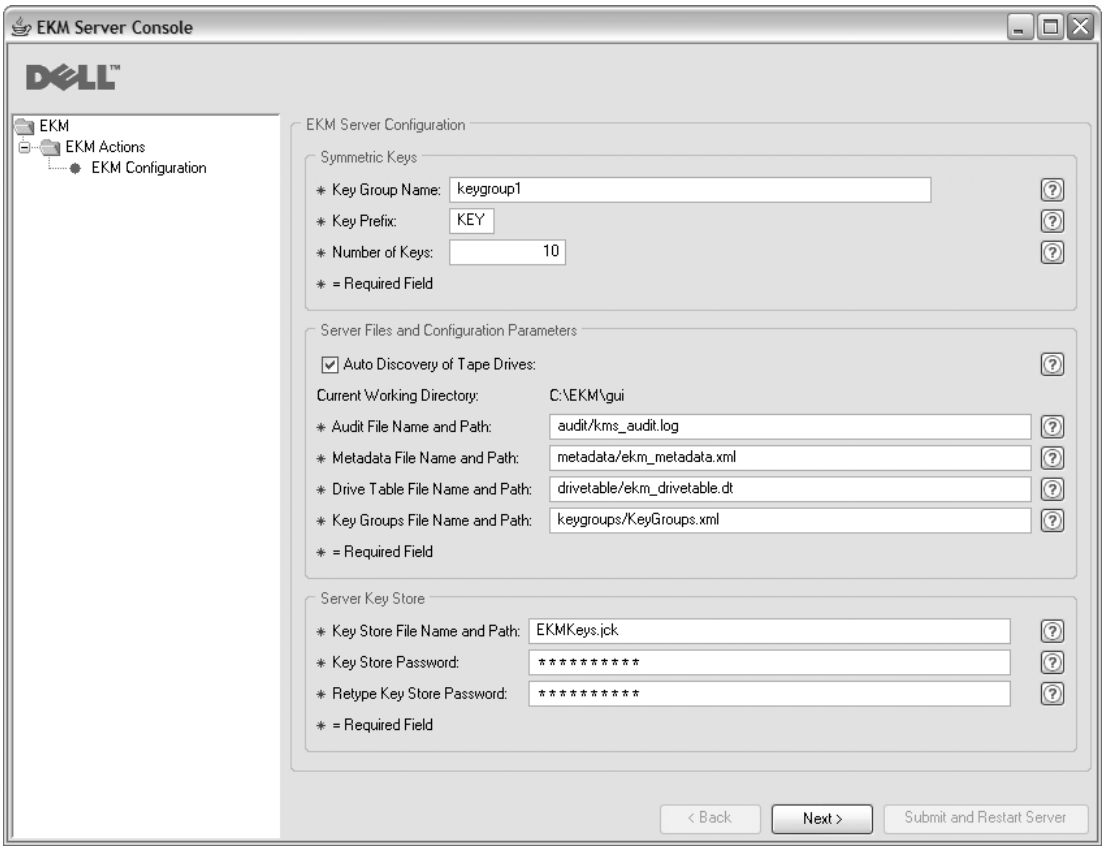

圖 3-4. EKM Server Configuration 頁面

雖然針對 Dell Encryption Key Manager 金鑰儲存庫而產生的金鑰數目沒有限制，但產生金鑰所需要的時間會依要求的金鑰數目而增加。Encryption Key Manager 花 15 秒來產生 10 個金鑰，花超出 30 分鐘來產生 10000 個金鑰。請注意，金鑰數目會受限於主機伺服器資源（伺服器中的記憶體）。Encryption Key Manager 應用程式在執行時，會維護系統記憶體內的金鑰儲存庫清單，以便在磁帶庫從磁帶機傳送金鑰要求時，能夠快速存取金鑰。

註: 在金鑰產生期間岔斷 Encryption Key Manager GUI，需要重新安裝 Encryption Key Manager。

如果您在 Encryption Key Manager 金鑰產生處理程序完成前加以停止，金鑰儲存庫檔會受到毀損。如果要從這個事件回復，請遵循下列步驟：

- 如果在 Encryption Key Manager 起始安裝期間，Encryption Key Manager 已岔斷，請導覽至 Encryption Key Manager 目錄所在的目錄（如 x:\ekm）。刪除目錄，再重新開始安裝。
- 如果在新增新的金鑰群組時岔斷了 Encryption Key Manager，請停止您的 Encryption Key Manager 伺服器，以最新的備份金鑰儲存庫（這個檔案在您的 x:\ekm\gui\backupfiles 資料夾中）來還原您的金鑰儲存庫檔案。請注意，備份檔的檔名含有日期和時間戳記（如 2007_11_19_16_38_31_EKMKeys.jck）。檔案複製到 x:\ekm\gui 目錄之後，必須移除日期和時間戳記。請重新啓動 Encryption Key Manager 伺服器，新增先前岔斷的金鑰群組。

4. 在 『EKM Server Certificate Configuration』 頁面 (圖 3-5) 上，輸入金鑰儲存庫別名及您想要的任何其他資料。按一下 **Submit and Restart Server**。

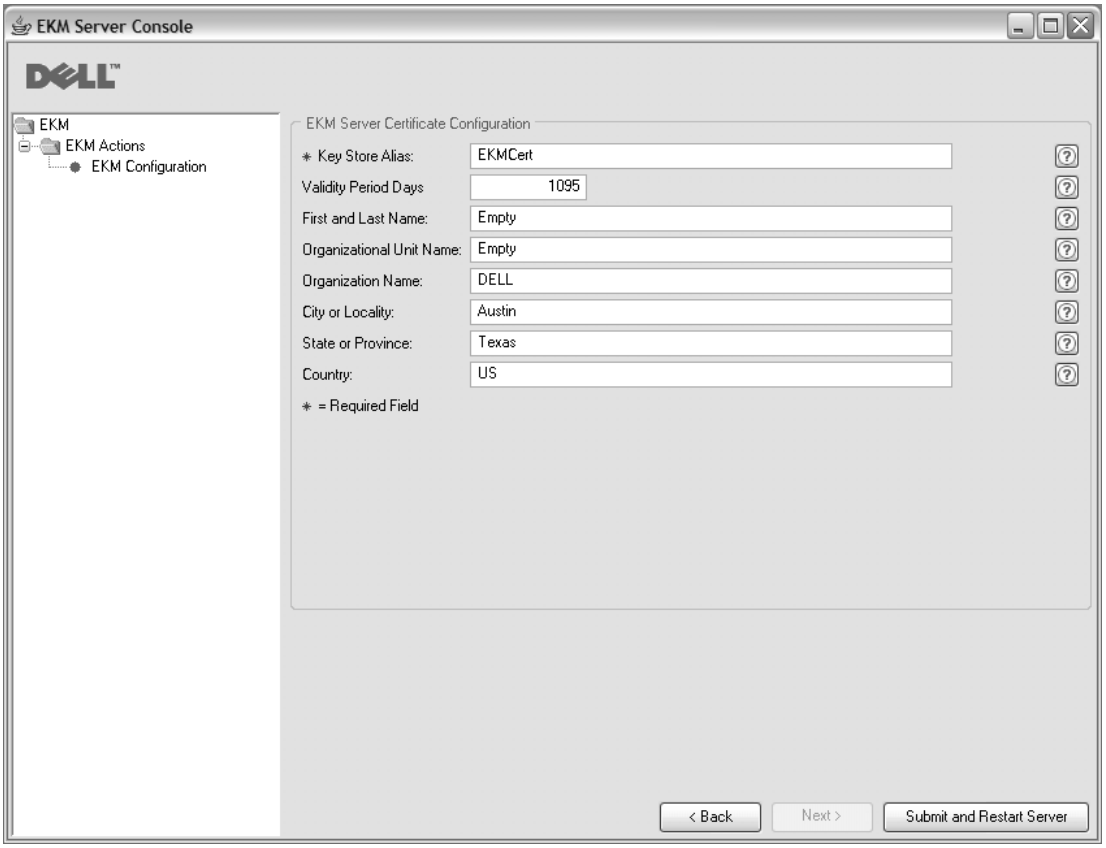

圖 3-5. EKM Server Certificate Configuration 頁面

5. 這時會開啓一個 『Backup Critical Files』 視窗 (第 3-8 頁的圖 3-6)，提示您備份 Encryption Key Manager 資料檔。

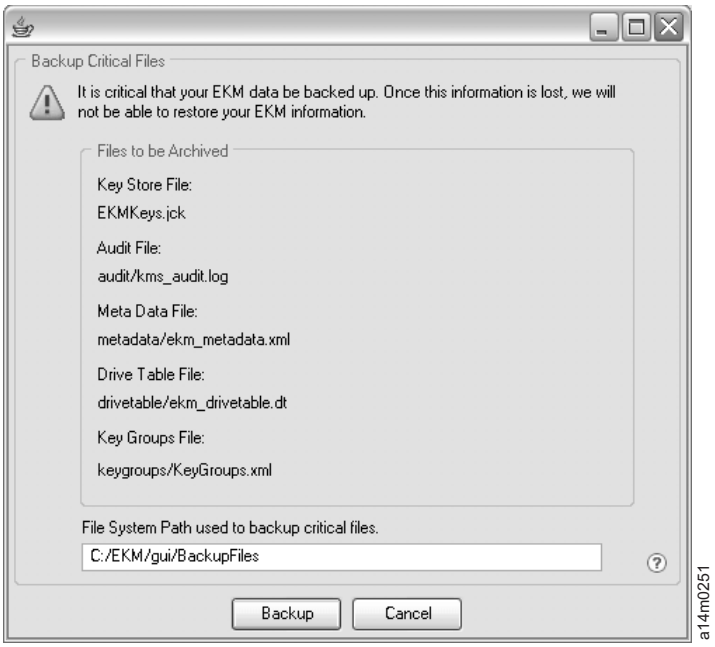

圖 3-6. Backup Critical Files 視窗

請驗證路徑，再按一下 **Backup**。這時會在背景啓動 Dell Encryption Key Manager 伺服器。

當在 『Backup Critical Files』 視窗中變更 Encryption Key Manager 伺服器配置或**備份**時，每當按一下 **OK**，Encryption Key Manager 都會產生一組備份檔。檔案只要列爲*將保存的檔案*，都會儲存在 c:/ekm/gui/BackupFlies 目錄中。每個檔名前面都會附加日期和時間。比方說，2007 年 11 月 26 日下午 2 點 58 分 46 秒所備份的一組檔案，名稱開頭都會含有下列日期和時間戳記：“2007_11_26_14_58_46_FileName。不會改寫備份檔。

6. 在 GUI 導覽器中，選取**伺服器性能監視器**來確認 Encryption Key Manager 伺服器已啓動。

如果要新增金鑰到現有的金鑰儲存庫中，請參閱第 3-14 頁的『利用 GUI 來定義金鑰群組和建立金鑰』。

如何找到正確的主機 IP 位址：

現行 Encryption Key Manager GUI 的限制，會造成會造成「伺服器性能監視器」無法顯示 Encryption Key Manager 主機 IP 位址：

- 如果主機配置了 IPv6 位址，Encryption Key Manager 應用程式將無法顯示 IP 位址。
- 如果 Encryption Key Manager 應用程式是安裝在 Linux 系統中，Encryption Key Manager 應用程式會顯示本端主機位址，而不是實際作用中的 IP 埠。

1. 如果要擷取主機系統的實際 IP 位址，請存取網路配置來尋找 IP 埠位址。
   - 在 Windows 系統中，開啓一個指令視窗，輸入 `ipconfig`。
   - 如果是 Linux，請輸入 `isconfig`。

如何識別 EKM SSL 埠

1. 利用指令行啓動 Encryption Key Manager 伺服器。
   - 在 Windows 上，cd c:\ekm 以進行導覽，並按一下 **startServer.bat**
   - 在 Linux 平台上，導覽至 /var/ekm，並輸入 startServer.sh
   - 請參閱第 5-1 頁的『啓動、重新整理和停止金鑰管理程式伺服器』，以取得詳細資訊。
2. 利用指令行來啓動 CLI 用戶端。
   - 在 Windows 上，cd c:\ekm 以進行導覽，並按一下 **startClient.bat**
   - 在 Linux 平台上，導覽至 /var/ekm，並輸入 startClient.sh
   - 請參閱第 5-5 頁的『指令行介面用戶端』，以取得詳細資訊。
3. 在 Encryption Key Manager 伺服器上，利用下列指令來登入 CLI 用戶端：

   ```
   login –ekmuser userID –ekmpassword password
   ```

   其中 *userID* = EKMAdmin，*password* = changeME（這是預設密碼。如果您先前變更了預設密碼，請使用您的新密碼。）

   成功登入之後，畫面上會顯示 User successfully logged in。
4. 輸入下列指令來識別 SSL 埠：

   ```
   status
   ```

   顯示的回應類似於：伺服器在執行中。 TCP port: 3801, SSL port: 443.

   請記下 SSL 配置埠，以確定它是用來配置磁帶庫管理加密設定的埠。
5. 從指令行登出。輸入下列指令：

   ```
   exit
   ```

   關閉指令視窗。

在 LTO 4 和 LTO 5 產生金鑰和別名以進行加密

「Dell Encryption Key Manager 伺服器 GUI」是產生對稱加密金鑰最簡單的方法（請參閱第 3-5 頁的『利用 GUI 來建立配置檔、金鑰儲存庫和憑證』）。另外，您也可以利用 Keytool 公用程式來產生對稱加密金鑰。在不同金鑰儲存庫之間匯入和匯出金鑰，Keytool 格外有用。請參閱第 3-12 頁的『利用 Keytool -importseckey 匯入資料金鑰』和第 3-12 頁的『利用 Keytool -exportseckey 匯出資料金鑰』，以取得詳細資料。

Keytool 是用來管理金鑰、憑證和別名的公用程式。 它讓您能夠建立、匯入和匯出加密資料金鑰，以及將它們儲存在金鑰儲存庫中。

金鑰儲存庫中的每個資料金鑰都是利用唯一別名來存取。別名是一個字元字串，如 123456tape。在 JCEKS 金鑰儲存庫中，123456Tape 同等於 123456tape，都可存取金鑰儲存庫中的相同項目。 當您利用 **keytool -genseckey** 指令來產生資料金鑰時，您在相同指令中指定相對應的別名。這個別名可讓您在正確的金鑰群組和金鑰儲存庫中識別正確的金鑰，以便在 LTO 4 和 LTO 5 磁帶上寫入和讀取加密資料。

註: 個別別名和別名範圍必須是唯一的。當金鑰是在給定的金鑰儲存庫/ Encryption Key Manager 實例上產生時，會強制執行。不過，在多重 Encryption Key Manager /金

鑰儲存庫環境中，您應該使用可跨越多個實例維護獨特性的命名慣例，以便在實例之間傳輸金鑰，並維護其參照獨特性。

產生金鑰和別名之後，更新 KeyManagerConfig.properties 檔中的 symmetricKeySet 內容來指定新的別名、別名範圍，或金鑰群組 GroupID、用來儲存對稱金鑰的檔名，以及金鑰群組定義所在的檔名。（請參閱第 3-13 頁的『建立和管理金鑰群組』，以取得詳細資料。）只有對 symmetricKeySet 中所指名的金鑰會進行驗證（檢查現有的別名及適當大小和演算法的對稱金鑰）。如果這個內容指定了無效的金鑰，金鑰管理程式便不會啓動，這時會建立一筆審核記錄。

另外，keytool 公用程式也提供來在金鑰儲存庫之間匯入和匯出資料金鑰。以下是各項作業的概觀。您可以發出 **keytool -ekmhelp** 來顯示下列各項討論所涵蓋的所有金鑰管理程式相關參數。

編輯配置內容檔

如果要變更 KeyManagerConfig.properties 或 ClientKeyManagerConfig.properties 檔，請執行下列動作：

1. 停止 Encryption Key Manager 伺服器。
2. 利用您選擇的文字編輯器，開啓 KeyManagerConfig.properties 檔來變更伺服器配置，如果是用戶端配置，則開啓 ClientKeyManagerConfig.properties 檔。請勿因爲 ^M 而利用 Windows 來編輯 Linux 機器的檔案。如果您使用 Windows，請利用 gvim/vim 來編輯檔案。
3. 根據這份文件所提供的指示來變更內容值。
4. 儲存檔案。
5. 重新啓動 Encryption Key Manager 伺服器。

如果您使用的不是 Keytool

如果您不使用 keytool 或 GUI 來產生金鑰和別名，就不能產生與 Encryption Key Manager 相容的一系列金鑰。 若要產生與 Encryption Key Manager 相容的個別金鑰，請確實以下列其中一項格式指定別名：

- 12 個或以下的可列印字元（例如，`abcdefghijk`）
- 3 個可列印字元，後面加兩個零，接著再加上 16 個十六進位數字（例如，`ABC00000000000000001`），總共是 21 個字元

利用 Keytool -genseckey 產生資料金鑰和別名

註: 在任何階段作業中初次使用 **keytool** 指令之前，請先執行 `updatePath` Script 來設定正確的環境。

Windows
導覽至 cd c:\ekm，並按一下 **updatePath.bat**

Linux 平台
導覽至 /var/ekm 然後輸入 `. ./updatePath.sh`

Keytool 公用程式會產生別名和對稱金鑰，以便在使用 LTO 4 和 LTO 5 磁帶的 LTO 4 和 LTO 5 磁帶機上進行加密。請利用 **keytool -genseckey** 指令來產生一或多個私密金鑰，將它們儲存在指定的金鑰儲存庫中。**keytool -genseckey** 所用的參數如下：

-genseckey [-v] [-protected]
[-alias <alias> | aliasrange <aliasRange>] [-keypass <keypass>]
[-keyalg <keyalg>] [-keysize <keysize>]
[-keystore <keystore>] [-storepass <storepass>]
[-storetype <storetype>] [-providerName <name>]
[-providerClass <provider_class_name> [-providerArg <arg>] ...
[-providerPath <pathlist>]

當產生 Encryption Key Manager的資料金鑰來用於 LTO 4 和 LTO 5 磁帶機以進行磁帶加密時，這些參數尤其重要：

-alias
指定單一資料金鑰的 *alias* 值，最多 12 個可列印的字元（如 `abcfrg` 或 `key123tape`）。

-aliasrange
當產生多個資料金鑰時，*aliasrange* 指定爲 3 個字元的英文字母字首，後面接著 16 字元（十六進位）字串系列上下限，並自動填入前導零來建構長 21 個字元的別名。比方說，指定 `key1-a` 會產生從 `KEY000000000000000001` 到 `KEY00000000000000000A` 的一系列別名。指定 `xyz01-FF` 的 *aliasrange* 值會產生 `XYZ000000000000000001` 到 `XYZ0000000000000000FF`，這會產生 255 個對稱金鑰。

-keypass
指定用來保護資料金鑰的密碼。這個密碼與金鑰儲存庫密碼**必須相同**。如果未指定任何密碼，系統會提示您輸入它。如果您在提示中按 **Enter** 鍵，金鑰密碼會設爲金鑰儲存庫所用的相同密碼。*keypass* 必須是至少 6 個字元長。

註: 金鑰儲存庫密碼設定好之後，**請勿加以變更**，除非安全出現漏洞。請參閱『變更金鑰儲存庫密碼』。

-keyalg
指定用來產生資料金鑰的演算法。這個值必須指定爲 `AES`。

-keysize
指定要產生的資料金鑰大小。金鑰大小必須指定爲 `256`。

能夠關聯於對稱金鑰的可接受別名範例如下：

```
abc000000000000000001
abc00a0120fa000000001
```

金鑰管理程式不接受的別名範例如下：

```
abcefghij1234567 ? 長度錯誤
abcg0000000000000001 ? 字首超出 3 個字元
```

如果別名存在於金鑰儲存庫中，Keytool 會擲出異常狀況，並停止作業。

變更金鑰儲存庫密碼

註: 金鑰儲存庫密碼設定好之後，**請勿加以變更**，除非安全出現漏洞。這些密碼會成爲亂碼，以免出現任何安全漏洞。如果要變更金鑰儲存庫密碼，您必須利用下列 **keytool** 指令來個別變更這個金鑰儲存庫中每個金鑰的密碼。

如果要變更金鑰儲存庫密碼，請輸入：

```
keytool -keypasswd  -keypass old_passwd -new new_passwd -alias alias
        -keystore keystorename -storetype keystoretype
```

您也必須編輯 KeyManagerConfig.properties 來變更在每個伺服器配置檔內容中，以下列其中一個方法來指定的金鑰儲存庫密碼：

- 刪除整個以亂碼呈現的密碼，讓 Encryption Key Manager 在下次啓動時出現提示。
- 刪除整個以亂碼呈現的密碼，以明碼來輸入新密碼。下次啓動時，它會顯示成亂碼。

利用 Keytool -importseckey 匯入資料金鑰

請利用 keytool -importseckey 指令，從匯入檔案匯入一個私密金鑰或一個批次的私密金鑰。 **keytool -importseckey** 所用的參數如下：

-importseckey [-v]
[-keyalias <keyalias>] [-keypass <keypass>]
[-keystore <keystore>] [-storepass <storepass>]
[-storetype <storetype>] [-providerName <name>]
[-importfile <importfile>] [-providerClass <provider_class_name>]
[providerArg <arg>]

當匯入 Encryption Key Manager的資料金鑰來用於 LTO 4 和 LTO 5 磁帶機以進行磁帶加密時，這些參數尤其重要：

-keyalias
指定金鑰儲存庫中之私密金鑰的別名，以解密 *importfile* 中的所有資料金鑰。

-importfile
指定含有要匯入之資料金鑰的檔案。

利用 Keytool -exportseckey 匯出資料金鑰

請利用 keytool -exportseckey 指令，將一個私密金鑰或一個批次的私密金鑰匯出到匯出檔中。 **keytool -exportseckey** 所用的參數如下：

-exportseckey [-v]
[-alias <alias> | aliasrange <aliasRange>] [-keyalias <keyalias>]
[-keystore <keystore>] [-storepass <storepass>]
[-storetype <storetype>] [-providerName <name>]
[-exportfile <exportfile>] [-providerClass <provider_class_name>]
[providerArg <arg>]

當匯出Encryption Key Manager的資料金鑰來用於 LTO 4 和 LTO 5 磁帶機以進行磁帶加密時，這些參數尤其重要：

-alias
指定單一資料金鑰的 *alias* 值，最多 12 個可列印的字元（如 `abcfrg` 或 `key123tape`）。

-aliasrange
當匯出多個資料金鑰時，*aliasrange* 指定爲 3 個字元的英文字母字首，後面接著 16 字元（十六進位）字串系列上下限，並自動填入前導零來建構長 21 個字元的別名。

比方說，指定 key1-a 會產生從 KEY000000000000000001 到 KEY00000000000000000A 的一系列別名。指定 xyz01-FF 的 *aliasrange* 值會產生 XYZ000000000000000001 至 XYZ0000000000000000FF

-exportfile
匯出資料金鑰時，指定用來儲存資料金鑰的檔案。

-keyalias
指定金鑰儲存庫中，用來加密所有資料金鑰的公開金鑰別名。請確定將將匯入對稱（資料）金鑰的金鑰儲存庫含有相對應的私密金鑰。

使用 JCEKS 金鑰儲存庫的 LTO 4 和 LTO 5 加密之範例別名和對稱金鑰設定

請使用 -aliasrange 選項來呼叫 **KeyTool**。

請注意，您必須依照下列方式，將金鑰演算法 (-keyalg) 指定爲 AES，將金鑰大小 (–keysize) 指定爲 256：

```
/bin/keytool –genseckey –v –aliasrange AES01-FF –keyalg AES –keysize 256
–keypass password -storetype jceks –keystore path/filename.jceks
```

這些 KeyTool 呼叫會產生 AES000000000000000001 至 AES0000000000000000FF 範圍內的 255 個循序別名，以及相關聯的 AES 256 位元對稱金鑰。兩者都可以依照需要重複累積許多次，以便設定健全的金鑰管理程式作業所需要全範圍及獨立式金鑰別名。比方說，如果要在 上，產生 LTO 4 和 LTO 5 額外的別名和對稱金鑰：

```
/bin/keytool –genseckey –v –alias abcfrg –keyalg AES –keysize 256
–keypass password -storetype jceks –keystore path/filename.jceks
```

這個呼叫會從上述 呼叫中，將獨立式別名 abcfrg 累加新增到已含有 255 個別名的指名金鑰儲存庫中，在 –keystore 選項所指名的 jceks 檔中，產生 256 個對稱金鑰。

請更新 KeyManagerConfig.properties 檔中的 symmetricKeySet 內容，新增下面這一行來符合上面所用的任何或所有別名範圍，以及用來儲存對稱金鑰的檔名。請注意，如果指定了無效的別名，Encryption Key Manager 不會啓動。驗證檢查失敗的其他原因可能包括位元大小不正確（AFS 金鑰大小必須是 256），或平台的演算法無效。-keyalg 必須是 AES，-keysize 必須是 256。**config.keystore.file** 指定的檔名應該符合 KeyTool 呼叫中的 –keystore <filename> 所指定的名稱：

symmetricKeySet = AES01-FF,abcfrg
config.keystore.file = <filename>.jceks

只有對 symmetricKeySet 中所指名的金鑰會進行驗證（檢查現有的別名及適當大小和演算法的對稱金鑰）。如果這個內容指定了無效的金鑰，Encryption Key Manager 便不會啓動，這時會建立一筆審核記錄。

建立和管理金鑰群組

Encryption Key Manager 可以讓您將 LTO 4 及 LTO 5 加密的對稱金鑰，組織爲幾個金鑰群組。依照這個方式，您可以根據加密的資料類型、有存取權的使用者或其他有意義的性質來分組金鑰。建立好金鑰群組之後，您便可以在 **adddrive** 指令中，利用 -symrec 關鍵字，將它關聯於特定的磁帶機。請參閱第 5-7 頁的『adddrive』，以瞭解語法。

如果要建置金鑰群組，您必須將它定義在 KeyGroups.xml 檔中。如果您遵循第 3-5 頁的『利用 GUI 來建立配置檔、金鑰儲存庫和憑證』中的程序，則已在 EKM Configuration 頁面指定這個檔案的位置。如果是手動建立配置檔，便依照下列方式，在配置內容檔中指定 KeyGroups.xml 檔的位置：

```
config.keygroup.xml.file = FILE:KeyGroups.xml
```

如果未指定這個參數，預設行爲是使用 Encryption Key Manager 啓動位置工作目錄中的 KeyGroups.xml 檔。如果這個檔案不存在，便會建立空的 KeyGroups.xml 檔。隨後啓動 Encryption Key Manager 伺服器時，**native_stderr.log** 可能會出現下列訊息：`[Fatal Error] :-1:-1: Premature end of file.`。這是剖析空白 KeyGroups.xml 檔時所發生的錯誤，除非 Encryption Key Manager 伺服器已配置成使用金鑰群組，否則，它無法阻止 Encryption Key Manager 伺服器啓動。

金鑰群組是利用 Dell Encryption Key Manager 伺服器 GUI 或利用下列 CLI 用戶端指令來建立的（請參閱第 5-7 頁的『CLI 指令』，以瞭解語法）：

利用 GUI 來定義金鑰群組和建立金鑰

您可以利用 GUI 來執行管理金鑰群組的所有必要作業。您也可以利用它來建立其他金鑰。

註: 當您在執行下列任何作業時，按一下 **Submit Changes**，會開啓一個備份對話框視窗 (第 3-8 頁的圖 3-6)，提示您備份 Encryption Key Manager 資料檔。請輸入備份資料的儲存路徑。按一下 **Submit**。之後，再驗證備份路徑，並按一下 **OK**。

如果要建立金鑰群組並將金鑰移入，或要新增金鑰到現有的金鑰群組中，請執行下列動作：

1. 如果 GUI 尚未啓動，請開啓它：

   Windows

   導覽至 c:\ekm\gui，並按一下 **LaunchEKMGui.bat**

   Linux 平台

   導覽至 /var/ekm/gui 然後輸入 `. ./LaunchEKMGui.sh`
2. 在 GUI 左側導覽器中，選取 **Administration Commands**。
3. 在視窗底端，按一下 **Create a Group of Keys** (第 3-15 頁的圖 3-7)。

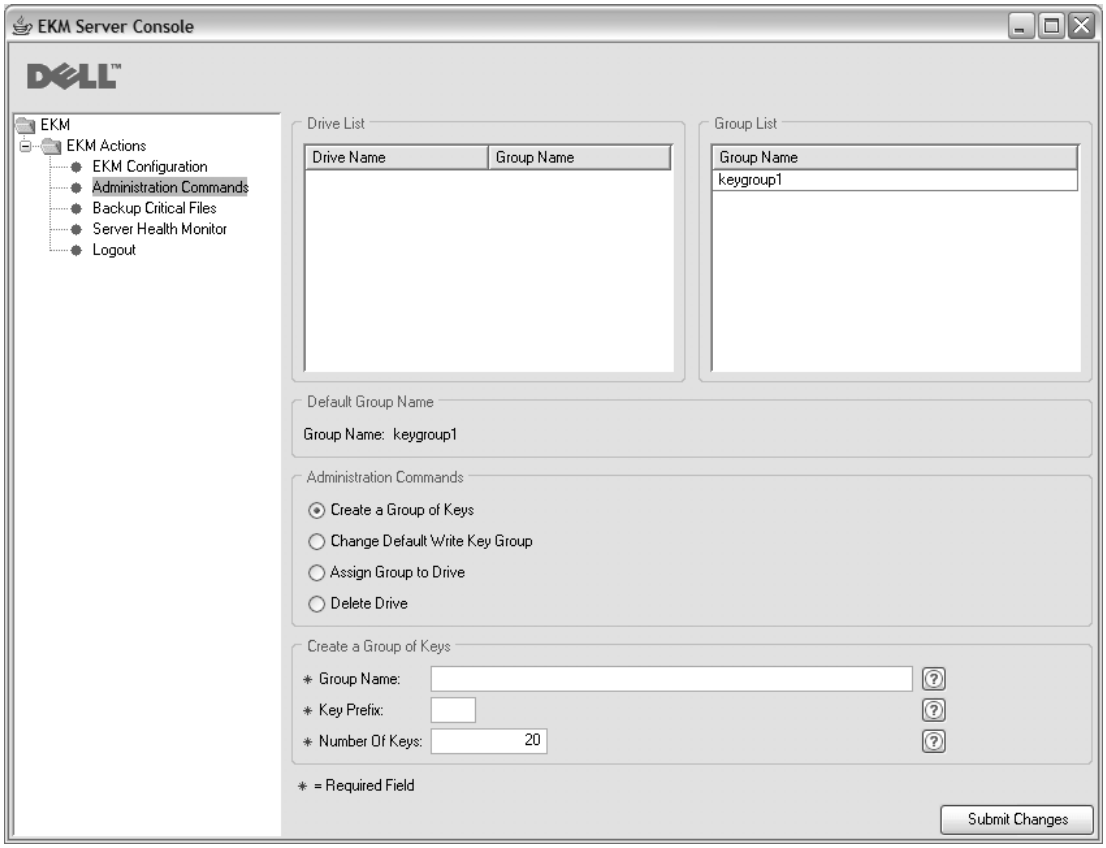

圖 3-7. 建立金鑰群組 (Create a Group of Keys)

4. 輸入新金鑰群組的名稱、金鑰別名要用的字首，以及群組將包含的金鑰數目。按一下 **Submit Changes**。

如果要變更預設金鑰群組，請執行下列動作：

1. 在 GUI 左側導覽器中，選取 **Administration Commands**。
2. 在視窗底端，按一下 **Change Default Write Key Group** (第 3-16 頁的圖 3-8)。

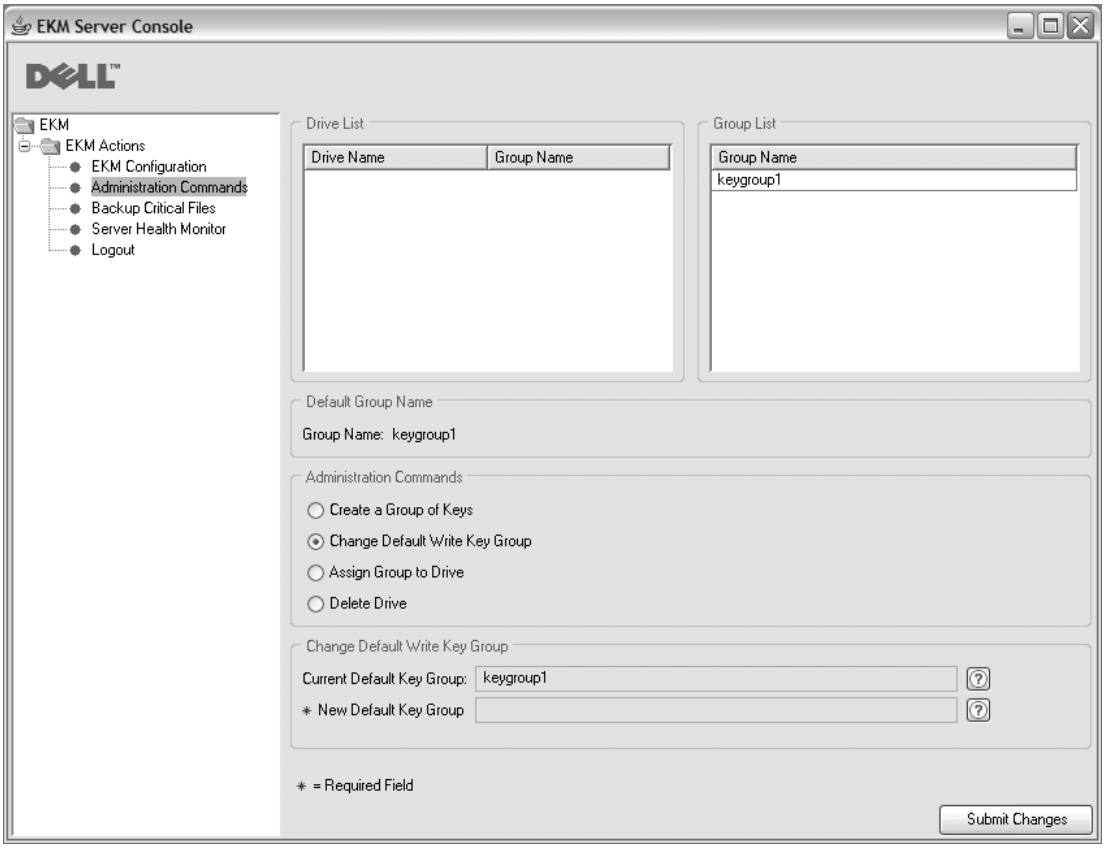

圖 3-8. 變更預設寫入金鑰群組 (Change Default Write Key Group)

3. 從右側 Group List 中，選取新的預設金鑰群組。
4. 在視窗底端，驗證現行及新的預設金鑰群組，並按一下 **Submit Changes**。

如果要將特定金鑰群組指派給特定的磁帶機，請執行下列動作：

1. 在 GUI 左側導覽器中，選取 **Administration Commands**。
2. 在視窗底端，按一下 **Assign Group to Drive** (第 3-17 頁的圖 3-9)。

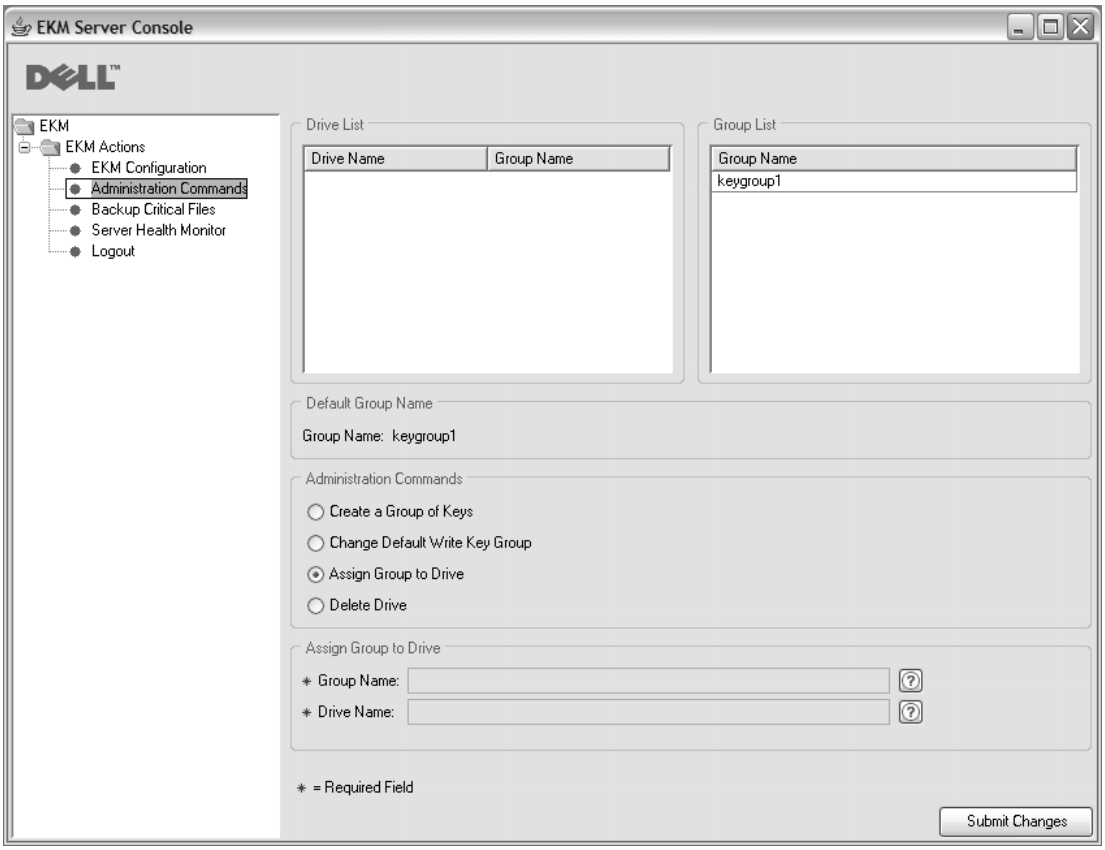

圖 3-9. 將群組指派給磁帶機 (Assign Group to Drive)

3. 從 Drive List 中，選取磁帶機。
4. 從 Group List 中，選取金鑰群組。
5. 在視窗底端，驗證磁帶機和金鑰群組，並按一下 **Submit Changes**。

如果要從磁帶機表格中刪除磁帶機，請執行下列動作：

1. 在 GUI 左側導覽器中，選取 **Administration Commands**。
2. 在視窗底端，按一下 **Delete Drive** (第 3-18 頁的圖 3-10)。

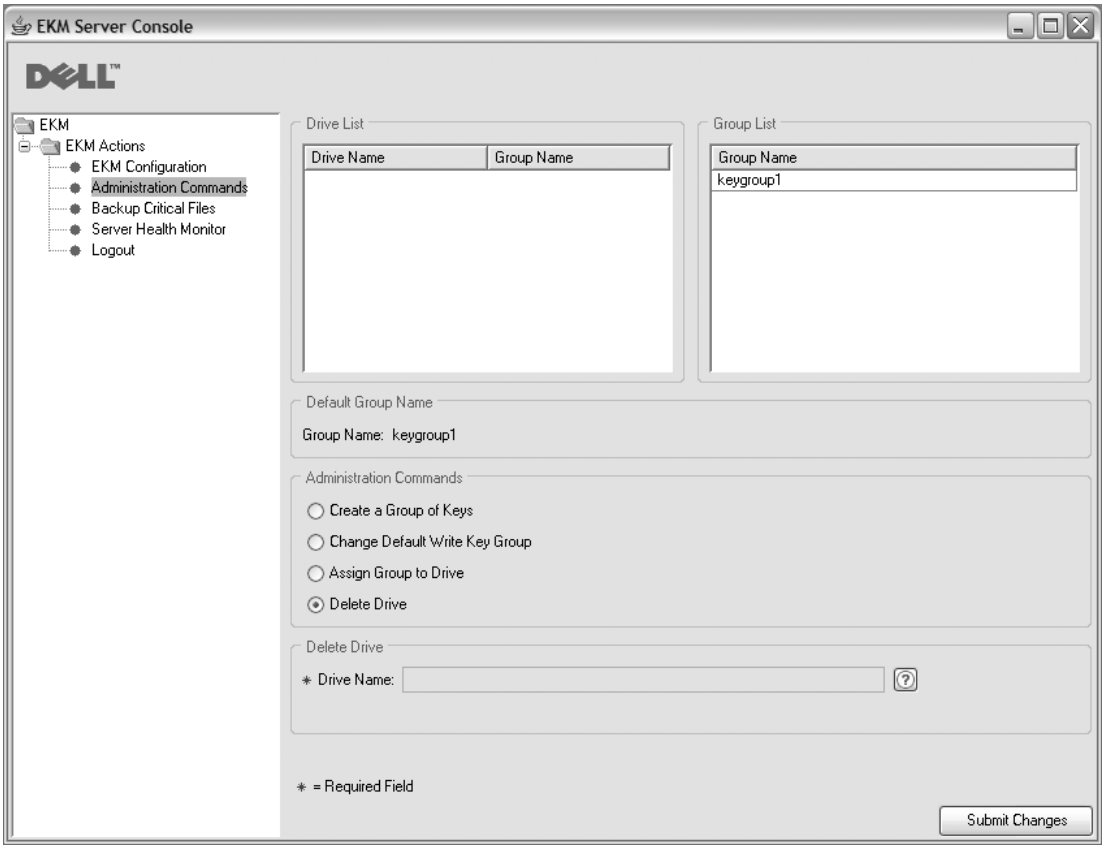

圖 3-10. 刪除磁帶機 (Delete Drive)

3. 從 Drive List 中，選取磁帶機。
4. 在視窗底端，驗證磁帶機名稱，並按一下 **Submit Changes**。

利用 CLI 指令來定義金鑰群組

Encryption Key Manager 的金鑰群組特性可讓您將金鑰集分組。

安裝和配置好 Encryption Key Manager 應用程式（金鑰儲存庫和產生的金鑰），Encryption Key Manager 伺服器也已啓動之後，請利用用戶端來登入伺服器，並遵循下列步驟：

1. 執行 **createkeygroup** 指令。

   這個指令會在 KeyGroups.xml 檔中建立起始金鑰群組物件。這個動作只執行一次。

   語法：**createkeygroup -password** *password*

   -password
   用來加密 KeyGroups.xml 檔中之金鑰儲存庫密碼，以便後來加以擷取的*密碼*。金鑰儲存庫會加密金鑰群組的金鑰，金鑰群組的金鑰又會加密每個個別的金鑰群組別名密碼。因此，KeyGroups.xml 檔不含任何明碼金鑰。

   範例： createkeygroup -password a75xynrd
2. 執行 **addkeygroup** 指令。

   這個指令會在 KeyGroups.xml 中，以唯一的「群組 ID」來建立金鑰群組的實例。

語法：**addkeygroup -groupID** *groupname*

-groupID
在 KeyGroup.xml 檔中，用來識別群組的唯一 *groupname*。

範例： addkeygroup -groupID keygroup1

3. 執行 **addkeygroupalias** 指令。

這個指令會建立金鑰儲存庫現有金鑰別名的新別名，以便新增到特定金鑰群組 ID 中。

語法：**addkeygroupalias -alias** *aliasname* **-groupID** *groupname*

-alias
金鑰的新 *aliasname*。這必須是完整的金鑰名稱，也就是說，Key00 必須輸入爲 key000000000000000000。

-groupID
在 KeyGroup.xml 檔中，用來識別群組的唯一 *groupname*。

範例： addkeygroupalias -alias key000000000000000000 -groupID keygroup1

註: 使用這個 CLI 指令時，每次只能新增一個金鑰。每個必須新增到金鑰群組中的個別金鑰，都必須執行這個指令。

4. 將金鑰群組關聯於新的或現有的磁帶機。
   a. 執行 **moddrive** 指令，將金鑰群組關聯於現有的磁帶機。

   這個指令會修改磁帶機表格中的磁帶機資訊。

   語法：**moddrive -drivename** *drivename* **-symrec** *alias*

   -drivename
   drivename 指定磁帶機的序號。

   -symrec
   指定磁帶機的*別名*（對稱金鑰的別名）或金鑰群組名稱。

   範例： moddrive -drivename 000123456789 -symrec keygroup1

   b. 執行 **adddrive** 指令來新增磁帶機到磁帶機表格中，並將它關聯於某個金鑰群組。

   這個指令可讓您新增磁帶機，並將它關聯於特定金鑰群組。

   語法：**adddrive -drivename** *drivename* **-symrec** *alias*

   -drivename
   drivename 指定要新增之磁帶機的 12 位數序號。

   註: 您必須在 10 位數序號前新增兩個零，以達到 12 位數。

   -symrec
   指定磁帶機的*別名*（對稱金鑰的別名）或 groupID。

   範例： adddrive -drivename 000123456789 -symrec keygroup1

如果要指定一個金鑰群組作爲磁帶機未定義別名時所用的預設值時，請將配置內容檔的 symmetrickeySet 內容設爲所要使用之金鑰群組的 GroupID。例如，

```
symmetricKeySet = keygroup1
```

GroupID 必須符合 KeyGroup.xml 檔中現有的金鑰群組 ID。否則，Encryption Key Manager 伺服器不會啓動。Encryption Key Manager 會追蹤金鑰群組內的金鑰用法。當您指定有效的 GroupID 時，Encryption Key Manager 會記錄上次使用的金鑰，再從指定的金鑰群組內隨機選取一個金鑰。

在金鑰群組之間複製金鑰

執行 **addaliastogroup** 指令。

這個指令會從現有（來源）金鑰群組中，將特定別名複製到新的（目標）金鑰群組。

語法：**addaliastogroup -aliasID** *aliasname* **-sourceGroupID** *groupname* **-targetGroupID** *groupname*

-aliasID
要新增之金鑰的 *aliasname*。

-sourceGroupID
用來識別要複製的別名之來源群組的唯一 *groupname*。

-targetGroupID
用來識別要新增別名之群組的唯一 *groupname*。

範例： addaliastogroup -aliasID aliasname -sourceGroupID keygroup1 -targetGroupID keygroup2

註: 在兩個群組內，金鑰都是可用的。

第 4 章 配置 Encryption Key Manager

利用 GUI 配置 Encryption Key Manager

建立配置內容檔最簡單的方法是遵循第 3-5 頁的『利用 GUI 來建立配置檔、金鑰儲存庫和憑證』中的程序來使用 Dell Encryption Key Manager GUI。如果已這麼做，您便已建立了配置檔，不需要任何其他配置。如果您想要使用其他 Encryption Key Manager 配置選項，下列資訊可能很有用。

配置策略

KeyManagerConfig.properties 檔中的部分配置設定提供了您應該知道的有效捷徑。

自動更新磁帶機表格

Encryption Key Manager 在配置檔 (drive.acceptUnknownDrives) 中提供的一個變數，在這個變數設爲 `true` 值之後，有新磁帶機連接 Dell Encryption Key Manager 時，會自動移入磁帶機表格。 這樣就不需要針對每個磁帶機或程式庫，使用 **adddrive** 指令。在這個模式下，並不需要使用CLI client commands，輸入每個裝置的 10 位數序號。 新的磁帶機會進行正常的公開/私密金鑰密碼交換來驗證磁帶機的身分。此驗證完成時，新的裝置就能根據儲存於現有磁帶上的 金鑰 ID，讀取這些磁帶（假設能夠在所配置的金鑰儲存庫中發現對應的金鑰資訊）。

註: Encryption Key Manager伺服器應在自動新增磁碟機之後，以 GUI 或 指令 第 5-13 頁的『refresh』 重新整理，確實將磁碟機儲存至磁碟機表格上。

在 LTO 4 及 LTO 5 磁碟機方面，您可以設定預設的對稱金鑰儲存區 (symmetricKeySet)，在新增的裝置上進行加密。換言之，您可以讓 Encryption Key Manager 在裝置進行連接時，利用相關聯的金鑰資料進行完整的裝置配置。 如果您選擇不在裝置新增到磁帶機表格時執行這個動作，也可以等磁帶機已新增到磁帶機表格之後，再利用 **moddrive** 指令來執行這個動作。

Encryption Key Manager 除了免除管理員爲每部磁帶機輸入 10 位數序號的麻煩外，同時也允許大型系統配置的預設環境。

另外，也請注意，這些便利的代價是安全程度會低一點。由於裝置是自動新增，應該關聯於某個憑證別名（能夠寫入含有這個憑證別名的磁帶），因此，會略過管理者手動新增裝置時所執行的附加安全檢查。請務必評估這個選項的優缺點來判斷，自動新增磁帶機資訊到磁帶機表格及隱含地授權新裝置存取憑證資訊，其安全風險是否可接受，這一點非常重要。

註: 依預設，drive.acceptUnknownDrives 內容會設爲 `false`。因此，Encryption Key Manager 不會自動將新磁帶機新增到磁帶機表格中。請選擇想採用的操作模式，再相應地變更配置。請參閱附錄 B，以取得詳細資料。

將兩部金鑰管理程式伺服器的資料同步化

您可以將兩部 Encryption Key Manager 伺服器的磁帶機表格和配置內容檔同步化。您可以利用 CLI 用戶端 **sync** 指令來手動完成這項作業，或在 KeyManagerConfig.properties 檔中設定四個內容來自動完成。

附註

這兩種同步化方法都不會處理金鑰儲存庫或金鑰群組 XML 檔。它們必須以手動方式來複製。

只有在 KeyManagerConfig.properties 檔中的 sync.ipaddress 內容指定了有效 IP 位址時，才會啓用自動同步化功能。請參閱『自動同步化』。

手動同步化

手動方法包括執行 CLI 用戶端 **sync** 指令。語法如下：

sync {**-all** | **-config** | **-drivetab**} **-ipaddr** *ip_addr* :*sslport* [**-merge** | **-rewrite**]

這個指令會將配置檔內容或磁帶機表格資訊（或兩者），從來源（或傳送端）伺服器傳送到 **–ipaddr** 參數所指定的目的地（或接收端）伺服器。接收端的 Encryption Key Manager 伺服器必須已啓動且在執行中。

必要欄位

-all
同時將配置內容檔和磁帶機表格資訊都傳送到 **-ipaddr** 所指定的伺服器。

-config
只將配置內容檔傳送到 **-ipaddr** 所指定的伺服器。

-drivetab
只將磁帶機表格資訊傳送到 **-ipaddr** 所指定的伺服器。

-ipaddr
ip_addr:sslport 指定接收端伺服器的位址和 ssl 埠。 *sslport* 應該符合接收端伺服器 KeyManagerConfig.properties 檔中的 『TransportListener.ssl.port』 所指定的值。

選用欄位

-merge
在接收端伺服器上，合併（新增）新的磁帶機表格資料與現行資料。（配置檔一律可以重新寫入。）這是預設值。

-rewrite
以新資料來取代接收端伺服器的現行資料。

自動同步化

磁帶機表格和內容檔可以從主要金鑰管理程式伺服器自動傳送到次要伺服器。次要伺服器必須在執行中，資料才能同步化。如果要將主要伺服器的資料自動同步化到次要伺服器中，便必須指定主要伺服器 KeyManagerConfig.properties 檔中的下列四個內容。次要或接收端伺服器內容檔不需要進行任何變更。

sync.ipaddress
指定接收端伺服器的位址和 ssl 埠，例如，

```
sync.ipaddress = backupekm.server.ibm.com:1443
```

如果未指定這個內容，或指定不正確，便會停用自動同步化。

sync.action
合併或重寫接收端伺服器中的現有資料。有效值如下：**merge**（預設值）和 **rewrite**。同步化配置內容一律會導致重寫。

sync.timeinhours
資料應該傳送的頻率。這個值是以整數來指定（時數）。時間間隔開始於伺服器啓動之時，也就是說，在伺服器執行了指定時數之後，便進行同步化。預設值是 24。

sync.type
應傳送的資料。有效值如下：**drivetab**（預設值）、**config** 和 **all**。

配置基礎

註: 如果您遵循第 3-5 頁的『利用 GUI 來建立配置檔、金鑰儲存庫和憑證』中的程序，這時已建立了基本配置，您不需要執行下列任何步驟。這項資訊顯示如何在不使用 GUI 的情況下執行這些作業，如果您想要利用其他配置選項，可能會很有用。

Windows 使用者附註: Windows 不接受目錄路徑含有空格的指令。當輸入指令時，可能需要指定爲這些目錄產生的簡短名稱，例如，`progra~1`，而不是 `Program Files`。如果要列出目錄簡短名稱，請發出 **dir /x** 指令。

這個程序包含配置 Encryption Key Manager 所需要的最少步驟。附錄 A 包括伺服器配置內容檔的範例。請參閱附錄 B，以取得伺服器和用戶端配置兩者所有內容的完整清單。

1. 利用 **Keytool** 來管理 JCEKS 金鑰儲存庫。當建立金鑰儲存庫時，請記下路徑和檔名，以及提供給憑證和金鑰的名稱。後面的步驟會用到這項資訊。
2. 如果金鑰儲存庫不存在，請建立一個。請將磁帶機要用的憑證和金鑰新增或匯入到這個新的金鑰儲存庫中。（請參閱第 3-9 頁的『在 LTO 4 和 LTO 5 產生金鑰和別名以進行加密』。）請記下提供給憑證和金鑰的名稱。後面的步驟會用到這項資訊。
3. 建立金鑰群組及移入金鑰別名。請參閱第 3-13 頁的『建立和管理金鑰群組』。
4. 利用您選擇的文字編輯器來開啓 **KeyManagerConfig.properties**，以指定下列內容。請注意，伺服器的現行設計非常嚴格。請勿因爲 ^M 而利用 Windows 來編輯 Linux 機器的檔案。如果您使用 Windows，請利用 gvim/vim 來編輯檔案。

   Windows 使用者附註: Java SDK 使用正斜線，在 Windows 上執行時也是如此。在 **KeyManagerConfig.properties** 檔中指定路徑時，請務必使用正斜線。在指令視窗中指定完整的路徑名稱時，請依照 Windows 的一般方式使用反斜線。

   a. **Audit.Handler.File.Directory** – 指定用來儲存審核日誌的位置。
   b. **Audit.metadata.file.name** – 指定 meta 資料 XML 檔的完整路徑和檔名。
   c. **Config.drivetable.file.url** – 指定 Encryption Key Manager 已知磁帶機之相關資訊的位置。在啓動伺服器或 CLI 用戶端之前，不需要這個檔案。如果它不存在，便會在 Encryption Key Manager 伺服器關閉期間建立它。

d. **TransportListener.ssl.keystore.name** - 指定步驟 1 所建立之金鑰儲存庫的路徑和檔名。

e. **TransportListener.ssl.truststore.name** - 指定步驟 1 所建立之金鑰儲存庫的路徑和檔名。

f. **Admin.ssl.keystore.name** - 指定步驟 1 所建立之金鑰儲存庫的路徑和檔名。

g. **Admin.ssl.truststore.name** - 指定步驟 1 所建立之金鑰儲存庫的路徑和檔名。

h. **config.keystore.file** - 指定步驟 1 所建立之金鑰儲存庫的路徑和檔名。

i. **drive.acceptUnknownDrives** - 指定 `true` 或 `false`。 true 值可讓連接 Encryption Key Manager 的新磁帶機自動新增到磁帶機表格中。 預設值是 false。

5. 下列選用的密碼項目可以新增，也可以省略。如果未在 **KeyManagerConfig.properties** 中指定這些項目，則在伺服器啓動期間，Encryption Key Manager 會提示您輸入金鑰儲存庫密碼。

   a. **Admin.ssl.keystore.password** - 指定步驟 1 所建立之金鑰儲存庫的密碼。

   b. **config.keystore.password** - 指定步驟 1 所建立之金鑰儲存庫的密碼。

   c. **TransportListener.ssl.keystore.password** - 指定步驟 1 所建立之金鑰儲存庫的密碼。

   新增到 **KeyManagerConfig.properties** 檔時，Encryption Key Manager 會使這些密碼成爲亂碼，以提高安全。

6. 如果要對照本端作業系統登錄來進行 CLI 用戶端鑑別，請選擇性地將 **Server.authMechanism** 內容設爲 `LocalOS` 值。如果未指定（或設爲 `EKM`），預設值是讓 CLI 用戶端使用者利用 usr/passwd 作爲 EKMAdmin/changeME 來登入金鑰管理程式伺服器。（**chgpasswd** 指令可以變更這個密碼。）

   當 **Server.authMechanism** 內容設爲 LocalOS 時，Linux 平台需要其他設定。如需詳細資訊，請參閱 **http://support.dell.com** 或產品檢附的 Dell Encryption Key Manager 媒體所提供的 Readme 檔。 第 5-5 頁的『鑑別 CLI 用戶端使用者』包含詳細資訊。

7. 將變更儲存在 **KeyManagerConfig.properties** 中。

8. 啓動 Encryption Key Manager 伺服器。如果要啓動伺服器，但不使用 GUI，

   Windows

   cd c:\ekm\ekmserver 以進行導覽，並按一下 **startServer.bat**

   Linux 平台

   導覽至 /var/ekm/ekmserver，輸入 `. ./startServer.sh`

   請參閱第 5-1 頁的『啓動、重新整理和停止金鑰管理程式伺服器』，以取得詳細資料。

9. 啓動 CLI 用戶端：

   Windows

   cd c:\ekm\ekmclient 以進行導覽，並按一下 **startClient.bat**

   Linux 平台

   導覽至 /var/ekm/ekmclient，輸入 `. ./startClient.sh`

   請參閱第 5-5 頁的『指令行介面用戶端』，以取得詳細資料。

10. 如果您在步驟 4(i) 指定 **drive.acceptUnknownDrives = false**，請在 # 提示之下，輸入下列指令來配置磁帶機：

    adddrive -drivename *drive_name* -recl *cert_name* -rec2 *cert_name*

    例如：

    ```
    # adddrive -drivename 000001365054 -rec1 key1c1 -rec2 key1c2
    ```

    後面接著

    ```
    # listdrives -drivename 000001365054
    ```

    這時會傳回

    ```
    Entry Key:  SerialNumber = 000001365054

    Entry Key:  AliasTwo = key1c2

    Entry Key:  AliasOne = key1c1
     Deleted : false
     Updated : true
     TimeStamp : Sun Jul 03 17:34:44 MST 2007
    ```

11. 請在 # 提示之下，輸入 **listdrives** 指令，確定已順利新增磁帶機。

第 5 章 管理 Encryption Key Manager

啓動、重新整理和停止金鑰管理程式伺服器

Encryption Key Manager 伺服器可以很容易的方式啓動和停止。

重新整理伺服器會使 Encryption Key Manager 將記憶體內的金鑰儲存庫現行內容、磁帶機表格及配置資訊傾出到個別的檔案中，再將它們重新載入記憶體中。在利用 CLI 用戶端進行了這些元件的任何變更之後，發出重新整理很有用。雖然 Encryption Key Manager 伺服器關閉時會自動儲存這些變更，但發出伺服器重新整理可以防止系統損毀或電源中斷造成這些變更遺失。

從 Dell Encryption Key Manager GUI 啓動 Encryption Key Manager 伺服器：

1. 如果 GUI 尚未啓動，請開啓它：

   Windows

   導覽至 c:\ekm\gui，並按一下 **LaunchEKMGui.bat**

   Linux 平台

   導覽至 /var/ekm/gui 然後輸入 `. ./LaunchEKMGui.sh`

2. 在 GUI 左側導覽器中，按一下**伺服器性能監視器**。
3. 在 『Server Status』 頁面 (圖 5-1) 中，按一下 **Start Server** 或 **Refresh Server**。

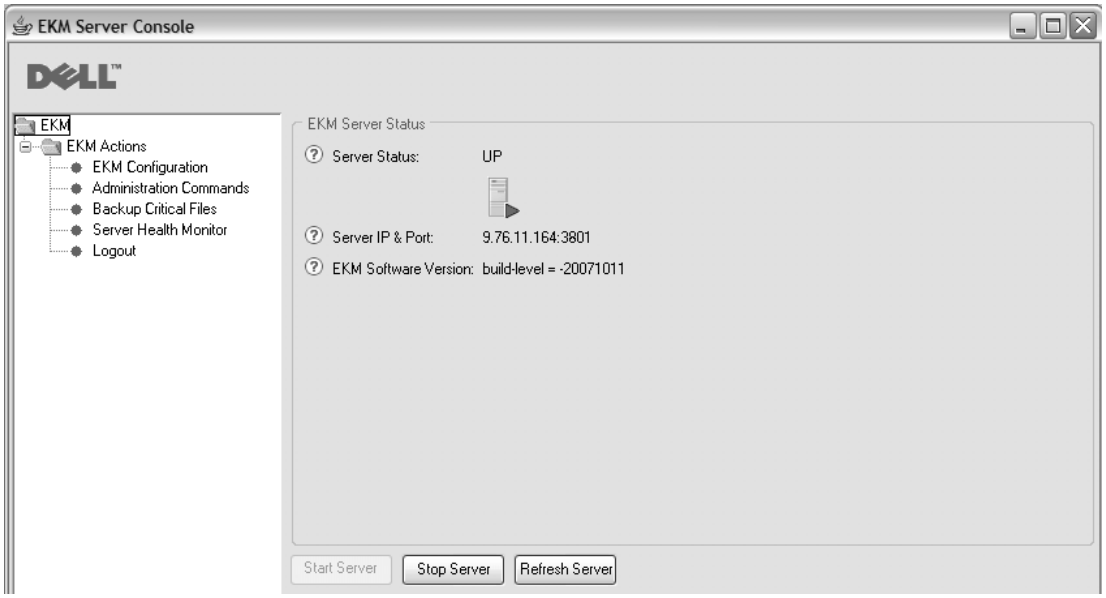

圖 5-1. Server Status

4. 伺服器狀態的變更反映在 Server Status 視窗中。請參閱圖 5-1。
5. 這時會顯示 Login 視窗 (第 5-2 頁的圖 5-2)。

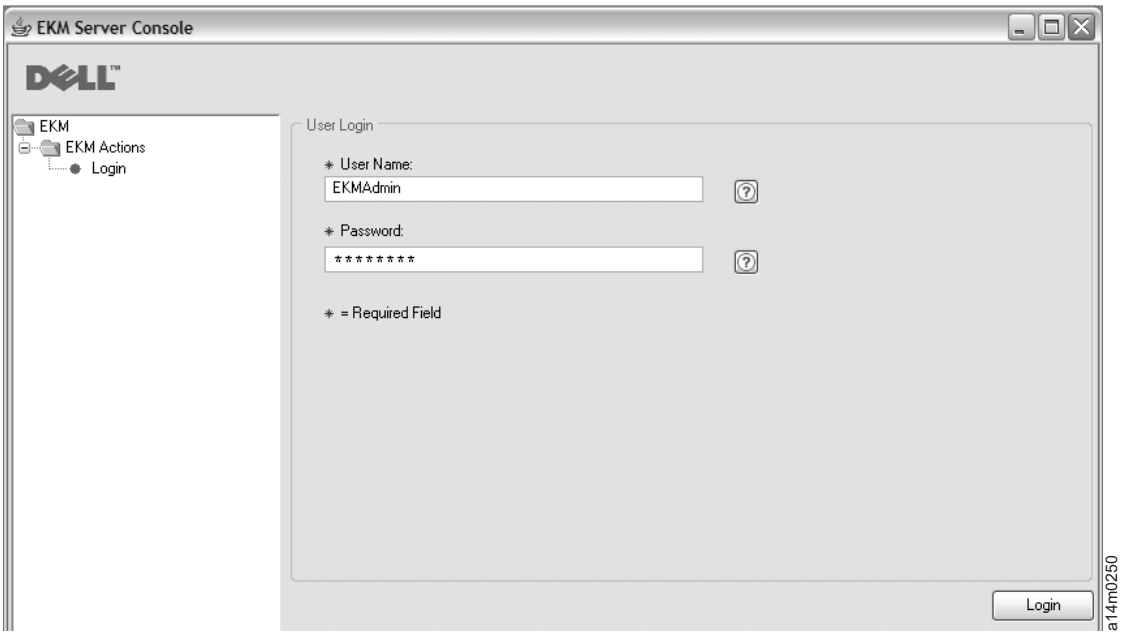

圖 5-2. Login 視窗

在 User Name 中，輸入 EKMAdmin。起始密碼是 changeME。登入之後，您可以利用 **chgpasswd** 指令來變更密碼。請參閱第 5-8 頁的『chgpasswd』。

註: • Dell Encryption Key Manager GUI 可能無法顯示主機 IP 位址

現行 GUI 的兩項限制，會造成**伺服器性能監視器**無法顯示 Encryption Key Manager 主機 IP 位址：

- 現行應用程式無法辨識 IPv6。如果主機配置了 IPV6 位址，Encryption Key Manager 應用程式將無法顯示 IP 位址。
- 如果 Encryption Key Manager 應用程式是安裝在 Linux 系統中，應用程式會顯示本端主機位址，而不是實際作用中的 IP 埠。

如果要擷取主機系統的實際 IP 位址，請存取網路配置來尋找 IP 埠位址。在 Windows 系統中，開啓一個指令視窗，輸入 ipconfig。如果是 Linux，請輸入 isconfig。

6. 按一下**登入**。

請利用相同的 Server Status 頁面來停止伺服器。

利用 Script 啓動金鑰管理程式伺服器

Windows

導覽至 cd c:\ekm\ekmserver，並按一下 **startServer.bat**

Linux 平台

導覽至 /var/ekm/ekmserver，輸入 . ./startServer.sh

如果要停止伺服器，請利用以下第 5-5 頁的『指令行介面用戶端』中所說明的任何方法來發出 **stopekm** 指令。另一個方法是將 **sigterm** 傳送給金鑰管理程式程序。 這可讓伺服器依正常程序關閉和結束。 請勿將 **sigkill** 傳送給金鑰管理程式程序。 **sigkill**

不會依正常程序來關閉程序。比方說，在 Linux 系統上，輸入 `kill -SIGTERM pid` 或 `kill -15 pid`。

在命令提示字元之下，啓動和停止金鑰管理程式伺服器

如果要從任何指令視窗或 Shell啓動 Encryption Key Manager 伺服器，請輸入：

```
java com.ibm.keymanager.EKMLaunch KeymanagerConfig.properties
```

這會在背景中啓動 Encryption Key Manager 伺服器。正確啓動之後，便可以利用 `ps -ef | grep java` 指令（Linux 平台）或 Windows 的「工作管理員」來顯示 Encryption Key Manager Java 程序。作爲一項「Windows 服務」來執行時，它會顯示爲 LaunchEKMService。

如果要停止伺服器，請利用以下第 5-5 頁的『指令行介面用戶端』中所說明的任何方法來發出 **stopekm** 指令。另一個方法是將 **sigterm** 傳送給金鑰管理程式程序。 這可讓伺服器依正常程序關閉和結束。 請勿將 **sigkill** 傳送給金鑰管理程式程序。 **sigkill** 不會依正常程序來關閉程序。比方說，在 Linux 系統上，輸入 `kill -SIGTERM pid` 或 `kill -15 pid`。

在 Windows 平台上，當 Dell Encryption Key Manager 啓動爲一項「Windows 服務」時，可以從「控制台」停止它。

安裝金鑰管理程式伺服器為 Windows 服務

將 Encryption Key Manager 伺服器安裝成主機伺服器中的一項服務，可確保在主機伺服器重新啓動時，會啓動 Encryption Key Manager 伺服器應用程式。

1. 從下載自「Dell 支援網站」(**http://support.dell.com**) 的版本中，將 LaunchEKMService.exe 執行檔解壓縮到暫存目錄中。
2. 如果要使服務能夠順利執行，您必須設定某些環境變數：
   a. 從「開始」功能表中，按一下**控制台**。
   b. 按兩下**系統**。
   c. 按一下**進階**標籤。
   d. 按一下**環境變數**。
   e. 在「系統變數」清單下，按一下**新增**。
   f. 指定 `JAVA_HOME` 作爲變數名稱，並輸入 IBM JVM 目錄。預設安裝目錄是 `C:\PROGRA~1\IBM\Java60`
   g. 按一下**確定**。
3. 利用這個程序來編輯系統 PATH 變數。

   註: 從指令行設定 PATH 變數無法運作。

   a. 從「開始」功能表中，按一下**控制台**。
   b. 按兩下**系統**。
   c. 按一下**進階**標籤。
   d. 按一下**環境變數**。
   e. 捲動「系統變數」清單來找出 **Path** 變數，並按一下**編輯**。
   f. 將 IBM JVM 路徑新增到 Path 變數的開頭。預設安裝目錄是`C:\PROGRA~1\IBM\Java60\jre\bin`

註: 在路徑結尾插入分號，將它與路徑清單中的其他目錄區隔。

g. 按一下**確定**。

4. 確定已完整定義 Encryption Key Manager 伺服器配置內容檔中的路徑。這個檔案的名稱是 KeyManagerConfig.properties，位在 `C:\ekm\gui` 目錄中。檔案中的所有下列路徑都應該檢查並更新，以確保它們有完整路徑（例如，使用 `c:\ekm\gui\EKMKeys.jck`，而不用 `gui\EKMKeys.jck`）。請參閱下列範例，以瞭解在使用預設安裝架構時，如何變更路徑。

   這些是使用預設安裝和金鑰儲存庫名稱時，它們應該指向的內容和完整路徑。您可以在 KeyManagerConfig.properties 檔中找到這些項目。

   config.keygroup.xml.file
   路徑應該改成：FILE:C:/ekm/gui/keygroups/KeyGroups.xml

   Admin.ssl.keystore.name
   路徑應該改成：C:/ekm/gui/EKMKeys.jck

   TransportListener.ssl.truststore.name
   路徑應該改成：C:/ekm/gui/EKMKeys.jck

   Audit.metadata.file.name
   路徑應該改成：C:/ekm/gui/metadata/ekm_metadata.xml

   Audit.handler.file.directory
   路徑應該改成：C:/ekm/gui/audit

   config.keystore.file
   路徑應該改成：C:/ekm/gui/EKMKeys.jck

   TransportListener.ssl.keystore.name
   路徑應該改成：C:/ekm/gui/EKMKeys.jck

   config.drivetable.file.url
   路徑應該改成：FILE:C:/ekm/gui/drivetable/ekm_drivetable.dt

   Admin.ssl.truststore.name
   路徑應該改成：C:/ekm/gui/EKMKeys.jck

5. **LaunchEKMServices.exe** 檔必須在命令提示字元之下執行。在 Windows 中，您可以導覽至**開始 >程式集>附屬應用程式>命令提示字元**來存取它。

6. 在命令提示字元之下，導覽至解壓縮 **LaunchEKMService.exe** 的暫存目錄。利用下列選項作爲參照來執行 **LaunchEKMService.exe** 檔。

   LaunchEKMService {**-help** | **-i** *config_file* | **-u**}

   -help
   顯示用法資訊。

   -i 將 Encryption Key Manager 安裝成一項「Windows 服務」。這個選項需要將配置內容檔的完整路徑名稱作爲引數傳入。預設路徑和檔名如下：`C:\ekm\gui\KeyManagerConfig.properties`。

   -u 如果不再需要當作一項服務來執行，請將金鑰管理程式「Windows 服務」解除安裝。請注意，EKMServer 服務必須已停止，才能解除安裝。當執行這個指令時，也可能會出現下列錯誤訊息：無法移除 `EKMServer`。錯誤 `0`。不過，仍可能已解除安裝這項服務。

如果要將 Encryption Key Manager 安裝成一項「Windows 服務」，請發出：

```
LaunchEKMService.exe -i config file
```

7. 利用上述指令安裝好服務之後，EKMServer 會出現在服務控制台中，您可以利用「服務控制台」來啓動和停止 Encryption Key Manager。

   註: 在第一次使用這項「Windows 服務」時，您必須利用控制台，以手動方式來啓動它。

指令行介面用戶端

Encryption Key Manager 伺服器啓動之後，您可以利用用戶端介面，在本端或遠端發出 CLI 指令。如果要發出 CLI 指令，您必須先啓動 CLI 用戶端。

鑑別 CLI 用戶端使用者

配置檔中的 Server.authMechanism 內容指定用來搭配本端/遠端用戶端的鑑別機制使用。當這個值設爲 `EKM` 時，CLI 用戶端使用者必須利用 user/password 作爲 EKMAdmin/changeME 來登入伺服器。（**chgpasswd** 指令可以變更這個密碼。請參閱第 5-8 頁的『chgpasswd』。） Server.authMechanism 內容的預設值是 `EKM`。

在 KeyManagerConfig.properties 檔中，將 Server.authMechanism 內容值指定爲 `LocalOS` 時，會對照本端作業系統登錄來進行用戶端鑑別。 CLI 用戶端使用者必須利用 OS user/password 來登入伺服器。請注意，只有能夠登入伺服器以及向伺服器提交指令的使用者/密碼，其使用者 ID 可用來執行伺服器，該使用者 ID 亦具備超級使用者/root 權限。

重要事項：變更 Encryption Key Manager 配置檔時，必須關閉 Encryption Key Manager 伺服器和 GUI。

如果在 Windows 中進行本端 OS 型鑑別，請依照下列方式，在 KeyManagerConfig.properties 中設定 Server.authMechanism=LocalOS：

1. 尋找 KeyManagerConfig.properties 檔（`c:\ekm\gui` 目錄）。
2. 利用您選擇的文字編輯器來開啓檔案（建議採用 WordPad）。
3. 尋找 `Server.authMechanism` 字串。如果這個字串不存在，請完全依照 `Server.authMechanism=LocalOS` 格式，將它新增到檔案中。
4. 儲存檔案。

現在，您的 Encryption Key Manager 伺服器使用者 ID 和密碼符合 OS 使用者帳戶。請注意，只有能夠登入伺服器以及向伺服器提交指令，且具備管理者專用權的使用者可以管理 Encryption Key Manager 伺服器

如果是 Linux 平台的本端 OS 型鑑別，便需要執行其他步驟：

1. 從 **http://support.dell.com** 下載 Dell Release R175158 (EKMServicesAndSamples)，並將檔案解壓縮到您選擇的目錄中。
2. 在下載中找出 LocalOS 目錄。
3. 將 libjaasauth.so 檔從平台適用的 JVM-JaasSetup 目錄中，複製到 *java_home*/jre/bin。
   - 在 32 位元的 Intel Linux 環境中，將LocalOS-setup/linux_ia32/libjaasauth.so 檔複製到 *java_home*/jre/bin/ 目錄，對於執行 1.6 JVM 的 32 位元 Intel Linux Kernel 而言，其中的 *java_home* 通常是 *java_install_path*/IBMJava-i386-60。

- 在 64 位元的 AMD64 Linux 環境中，將 LocalOS-setup/linux-x86_64/libjaasauth.so 檔複製到 *java_home*/jre/bin/ 目錄，對於執行 1.6 JVM 的 64 位元 Linux Kernel而言，其中的 *java_home* 通常是 *java_install_path*/IBMJava-x86_64-60。

如果是 Windows 平台，便不需要這個檔案。

安裝完成之後，您可以啓動 Encryption Key Manager 伺服器。現在，Encryption Key Manager 用戶端可以利用 OS 型使用者/密碼來登入。請注意，只有能夠登入伺服器以及向伺服器提交指令的使用者 ID，可用來執行伺服器，該使用者 ID 亦具備超級使用者/root 權限。

Dell 產品媒體及 **http://support.dell.com** 中的 Readme 檔提供了更詳細的安裝資料。

啓動指令行介面用戶端

註: Encryption Key Manager 伺服器和 Encryption Key Manager CLI 用戶端兩者內容檔中的 TransportListener.ssl.port 內容，必須設成相同的值，否則它們無法通訊。如果發生問題，請參閱第 6-2 頁的『爲 CLI 用戶端及 EKM 伺服器之間的通訊問題進行除錯』。

Encryption Key Manager CLI 用戶端和 Encryption Key Manager 伺服器利用 SSL 來維護它們的通訊安全。使用不含用戶端鑑別的預設 JSSE 配置時，Encryption Key Manager 伺服器之 TransportListener.ssl.keystore 的憑證必須在 TransportListener.ssl.truststore 中。如此一來，用戶端便知道它可以信任伺服器。如果 Encryption Key Manager CLI 用戶端與 Encryption Key Manager 伺服器在相同的系統上執行，便可以使用相同的配置內容檔。這使 Encryption Key Manager CLI 用戶端能夠使用 Encryption Key Manager 伺服器的相同金鑰儲存庫/信任儲存庫配置。如果它們不在相同的系統上，或您要用戶端使用不同的金鑰儲存庫，您必須從 Encryption Key Manager 伺服器配置內容檔所指定的 TransportListener.ssl.keystore 中匯出憑證。這些憑證必須匯入 Encryption Key Manager CLI 內容檔中 TransportListener.ssl.truststore 所指定的信任儲存庫。

您可以利用四種方式來啓動 CLI 用戶端和發出 CLI 指令。不論選擇哪個方式，您都必須指定 CLI 配置檔的名稱。請參閱「附錄 B」，以取得詳細資料。

使用 Script

Windows

導覽至 cd c:\ekm\ekmclient，並按一下 **startClient.bat**

Linux 平台

導覽至 /var/ekm/ekmclient，輸入 `. ./startClient.sh`

互動方式

如果要從任何指令視窗或 Shell 中，以互動方式來執行指令，請輸入：

```
java com.ibm.keymanager.KMSAdminCmd CLIconfiglfile_name -i
```

這時會出現 # 提示。在提交任何指令之前，您必須先利用下列指令，將 CLI 用戶端登入到金鑰管理程式伺服器中：

```
#login –ekmuser EKMAdmin –ekmpassword changeME
```

CLI 用戶端順利登入金鑰管理程式伺服器之後，您便可以執行任何 CLI 指令。完成之後，請利用 **quit** 或 **logout** 指令來關閉 CLI 用戶端。依預設，Encryption Key

Manager 伺服器會在 10 分鐘之後，關閉與未使用之用戶端的通訊 Socket。之後，凡是試圖輸入指令，都會使用戶端結束。如果要指定較長的 Encryption Key Manager 伺服器-用戶端 Socket 逾時期間，請修改 KeyManagerConfig.properties 檔中的 TransportListener.ssl.timeout 內容。

使用指令檔

如果要在檔案中向金鑰管理程式伺服器提交一批指令，請建立含有要發出之指令的檔案，如 *clifile* 。這個檔案中的第一個指令必須是 **login** 指令，因爲用戶端必須登入，才能執行任何指令。比方說，clifile 可能含有下列指令：

```
login -ekmuser EKMAdmin -ekmpassword changeME
listdrives
```

之後，如果要執行這個指令檔，請啓動 CLI 用戶端：

```
java com.ibm.keymanager.admin.KMSAdminCmd CLIconfiglfile_name –filename clifile
```

每次一個指令

您可以指定每個指令的 CLI userid_ID 和密碼，每次執行單一指令。請從任何指令視窗或 Shell 中輸入：

```
java com.ibm.keymanager.KMSAdminCmd ClientConfig.properties_name -listdrives
    -ekmuser EKMAdmin -ekmpassword changeME
```

（**chgpasswd** 指令可以變更這個密碼。）這時會執行這個指令，並結束用戶端階段作業。

CLI 指令

Encryption Key Manager 提供了一個指令集，供您從指令行介面用戶端用來與 Encryption Key Manager 伺服器互動，其中包括下列指令。

addaliastogroup

從現有（來源）金鑰群組中，將特定別名複製到新的（目標）金鑰群組。當您想要將某金鑰群組中的別名新增到另一個金鑰群組時，這很有用。

addaliastogroup -aliasID *aliasname* **-sourceGroupID** *groupname* **-targetGroupID** *groupname*

-aliasID
要新增之金鑰的 *aliasname*。

-sourceGroupID
用來識別要複製的別名之來源群組的唯一 *groupname*。

-targetGroupID
用來識別要新增別名之群組的唯一 *groupname*。

範例： addaliastogroup -aliasID aliasname -sourceGroupID keygroup1 -targetGroupID keygroup2

adddrive

在金鑰管理程式磁帶機表格中，增加新的磁帶機。請參閱第 4-1 頁的『自動更新磁帶機表格』，學習如何自動新增磁帶機到磁帶機表格中。請參閱第 2-4 頁的『加密金鑰及 LTO 4 和 LTO 5 磁帶機』，以取得別名需求的相關資訊。

adddrive -drivename *drivename* [**-rec1** *alias*] [**-rec2** *alias*][**-symrec** *alias*]

-drivename
drivename 指定所要新增之磁碟機的 12 位數序號。

註: 您必須在 10 位數序號前新增兩個零，以達到 12 位數。

-rec1
指定磁帶機憑證的*別名*（或金鑰標籤）。

-rec2
指定磁帶機憑證的第二個*別名*（或金鑰標籤）。

-symrec
指定磁帶機的*別名*（對稱金鑰的別名）或金鑰群組名稱。

範例： adddrive -drivename 000123456789 -rec1 alias1 -rec2 alias2

addkeygroup

利用「金鑰群組 XML」中唯一的「群組 ID」來建立金鑰群組的實例。

addkeygroup -groupID *groupname*

-groupID
用來識別 KeyGroup XML 檔中群組的唯一 *groupname*。

範例： addkeygroup -groupID keygroup1

addkeygroupalias

建立金鑰儲存庫現有金鑰別名的新別名，以便新增到特定金鑰群組 ID 中。

addkeygroupalias -alias *aliasname* **-groupID** *groupname*

-alias
金鑰的新 *aliasname*。

-groupID
用來識別 KeyGroup XML 檔中群組的唯一 *groupname*。

範例： addkeygroupalias -alias aliasname -groupID keygroup1

chgpasswd

變更 CLI 用戶端使用者 (EKMAdmin) 預設密碼。

chgpasswd -new *password*

-new
取代舊密碼的新*密碼*。

範例： chgpasswd -new ebw74jxr

createkeygroup

在 KeyGroup.xml 檔中，建立起始金鑰群組物件。只要執行一次。

createkeygroup -password *password*

-password
用來加密 KeyGroups.xml 檔中的金鑰儲存庫密碼，以便日後擷取的 *password*。金鑰儲存庫會加密金鑰群組的金鑰，金鑰群組的金鑰又會加密每個個別的金鑰群組別名密碼。因此，KeyGroups.xml 檔不含任何明碼金鑰。

範例： createkeygroup -password password

deletedrive

從金鑰管理程式磁帶機表格中，刪除磁帶機。 同等的指令爲 **deldrive** 和 **removedrive**。

deletedrive -drivename *drivename*

-drivename
drivename 指定要刪除之磁帶機的序號。

範例： deletedrive -drivename 000123456789

delgroupalias

從金鑰群組中，刪除金鑰別名。

delgroupalias -groupID *groupname* **-alias** *aliasname*

-groupID
在 KeyGroup.xml 檔中，用來識別群組的唯一 *groupname*。

-alias
要移除之金鑰別名的 *aliasname*。

範例： delgroupalias -groupID keygroup1 -alias aliasname

delkeygroup

刪除整個金鑰群組。

delkeygroup -groupID *groupname*

-groupID
在 KeyGroup.xml 檔中，用來識別群組的唯一 *groupname*。

範例： delkeygroup -groupID keygroup1

exit

結束 CLI 用戶端和停止 Encryption Key Manager 伺服器。同等的指令爲 **quit**。

範例： **exit**

export

將磁帶機表格或 Encryption Key Manager 伺服器配置檔匯出到指定的 URL。

export {**-drivetab**|**-config**} **-url** *urlname*

-drivetab
匯出磁帶機表格。

-config
匯出 Encryption Key Manager 伺服器配置檔。

-url
urlname 指定將寫入檔案的位置。

範例： export -drivetab -url FILE:///keymanager/data/export.table

help

顯示指令行介面指令名稱和語法。同等的指令是 **?**。

help

import

從指定的 URL 匯入磁帶機表格或配置檔。

import {**-merge**|**-rewrite**} {**-drivetab**|**-config**} **-url** *urlname*

-merge
合併新資料和現行資料。

-rewrite
以新資料來取代現行資料。

-drivetab
匯入磁帶機表格。

-config
匯入配置檔。

-url
urlname 用來指定取出新資料的位置。

範例： import -merge -drivetab -url FILE:///keymanager/data/export.table

list

列出 config.keystore.file 內容指名的金鑰儲存庫所包含的憑證。

list [**-cert** |**-key**|**-keysym**][**-alias** *alias* **-verbose** |**-v**]

-cert
列出指定金鑰儲存庫中的憑證。

-key
列出指定金鑰儲存庫中的所有金鑰。

-keysym
列出指定金鑰儲存庫中的對稱金鑰。

-alias
alias 指定清單專用的特定憑證。

-verbose|-v
顯示一或多個憑證的詳細資訊。

範例：
`list -v` 列出金鑰儲存庫中的所有項目。

`list -alias mycert -v` 列出 mycert 別名的所有可用資料（如果 config.keystore.file 金鑰儲存庫含有這個別名的話）。

listcerts

列出 config.keystore.file 內容指名的金鑰儲存庫所包含的憑證。

listcerts [**-alias** *alias* **-verbose** |**-v**]

-alias
alias 指定清單專用的特定憑證。

-verbose|-v
顯示一或多個憑證的詳細資訊。

範例： listcerts -alias alias1 -v

listconfig

列出記憶體內的 Encryption Key Manager 伺服器配置內容，反映 KeyManagerConfig.properties 檔的現行內容，加上 **modconfig** 指令所進行的任何更新。

listconfig

listdrives

列出磁帶機表格中的磁帶機。

listdrives [**-drivename** *drivename*]

-drivename
drivename 指定要列出之磁帶機的序號。

-verbose|-v
顯示一或多部磁帶機的詳細資訊。

範例： listdrives -drivename 000123456789

login

在 Encryption Key Manager 伺服器上，登入 CLI 用戶端。

login -ekmuser *userID* **-ekmpassword** *password*

-ekmuser
請在 *userID* 中指定 EKMadmin 或 localOS 使用者 ID 值，這會隨著所用的鑑別類型而不同（請參閱第 5-5 頁的『鑑別 CLI 用戶端使用者』）。

-ekmpassword
使用者 ID 的有效密碼。

範例： login -ekmuser EKMAdmin -ekmpassword changeME

logout

登出現行使用者。同等的指令是 **logoff**。只有在已啓用用戶端階段作業時，這些指令才有用。

範例： **logout**

modconfig

修改 Encryption Key Manager 伺服器配置內容檔 KeyManagerConfig.properties 中的內容。同等的指令是 **modifyconfig**。

modconfig {**-set** | **-unset**} **-property** *name* **-value** *value*

-set
將指定的內容設爲指定的值。

-unset
移除指定的內容。

-property
name 指定目標內容的名稱。

-value
在指定了 **-set** 之後，*value* 用來指定目標內容的新值。

範例： modconfig -set -property sync.timeinhours -value 24

moddrive

修改磁帶機表格中的磁帶機資訊。同等的指令是 **modifydrive**。

moddrive -drivename *drivename* {**-rec1** [*alias*] | **-rec2** [*alias*]| **-symrec** [*alias*]}

-drivename
drivename 指定磁帶機的序號。

-rec1
指定磁帶機憑證的*別名*（或金鑰標籤）。

-rec2
指定磁帶機憑證的第二個*別名*（或金鑰標籤）。

-symrec
指定磁帶機的*別名*（對稱金鑰的別名）或金鑰群組名稱。

範例： moddrive -drivename 000123456789 -rec1 newalias1

refresh

告知 Encryption Key Manager 以最新的配置參數來重新整理除錯、審核和磁帶機表格值。

範例： refresh

refreshks

重新整理金鑰儲存庫。如果在 Encryption Key Manager 伺服器執行時，已修改過 **config.keystore.file** 所指定的金鑰儲存庫，請利用這個指令來重新載入這個金鑰儲存庫。請只在必要時才使用這個指令，因爲它會降低效能。

範例： refreshks

status

顯示金鑰管理程式伺服器是已啓動或已停止。

範例： status

stopekm

停止 Encryption Key Manager 伺服器。

範例： stopekm

sync

以發出指令的金鑰管理程式伺服器的配置檔內容或磁帶機表格資訊（或兩者都包括在內）來同步化另一部 Encryption Key Manager 伺服器的配置檔內容或磁帶機表格資訊（或兩者都包括在內）。

註: 這兩種同步化方法都不會處理金鑰儲存庫或 KeyGroups.xml 檔。這些必須以手動方式來複製。

sync {**-all** | **-config** | **-drivetab**} **-ipaddr** *ip_addr* :*ssl*:*port* [**-merge** | **-rewrite**]

-all
將配置內容檔和磁帶機表格資訊兩者都傳送到 **-ipaddr** 指定的 Encryption Key Manager 伺服器。

-config
只將配置內容檔傳送到 **-ipaddr** 指定的 Encryption Key Manager 伺服器。

-drivetab
只將磁帶機表格資訊傳送到 **-ipaddr** 指定的 Encryption Key Manager 伺服器。

-ipaddr
ip_addr:ssl:port 指定接收端 Encryption Key Manager 伺服器的位址和 ssl 埠。*ssl:port* 應該符合接收端伺服器 KeyManagerConfig.properties 檔中的『TransportListener.ssl.port』 所指定的值。

-merge
合併新的磁帶機表格資料和現行資料。（配置檔一律可以重新寫入。）這是預設值。

-rewrite
以新資料來取代現行資料。

範例： sync -drivetab -ipaddr remoteekm.ibm.com:443 -merge

version

顯示 Encryption Key Manager 伺服器的版本。

範例： **version**

第 6 章 問題判斷

您可以啓用 Encryption Key Manager 的個別元件、多個元件或所有元件的除錯。

檢查這些重要檔案來瞭解 Encryption Key Manager 伺服器問題

在 Encryption Key Manager 無法啓動時，可以檢查三個檔案來判斷問題的原因。

- **native_stdout.log** 和 **native_stderr.log**
  - 由於 Encryption Key Manager 伺服器是在背景處理程序中執行，因此，沒有主控台可以顯示它的一般參考訊息和錯誤訊息。這些訊息會記載到這兩個檔案中。
  - 如果 Encryption Key Manager 伺服器內容檔含有 **debug.output.file** 內容，便會在除錯日誌的相同目錄中建立這兩個檔案。
  - 如果 Encryption Key Manager 伺服器內容檔不含 **debug.output.file** 內容，便會在工作目錄中建立這兩個檔案。
  - 每次啓動 Encryption Key Manager 伺服器時，都會刪除和重建這兩個檔案。
- **審核日誌**
  - 審核日誌含有 Encryption Key Manager 的處理程序進行中所記載的記錄。
  - 這個檔案的位置由 Encryption Key Manager 伺服器配置內容檔 **KeyManagerConfig.properties** 中的兩個內容來指定：
    - Audit.handler.file.directory – 指定審核日誌應該放在哪個目錄
    - Audit.handler.file.name – 指定審核日誌的檔名。
  - 如需審核的詳細資訊，請參閱第 7-1 頁的第 7 章, 『審核記錄』。

大於 127 字元的金鑰儲存庫密碼的日誌項目

Encryption Key Manager 安裝爲 Windows Service，且 KeyManagerConfig.properties 檔案中的金鑰儲存庫密碼長達 128 字元或以上時，Encryption Key Manager 因無法提示要輸入可接受長度的密碼而無法啓動。原生 Encryption Key Manager 日誌包含類似如下所示的項目：

native_stdout.log

```
伺服器已起始設定
預設的金鑰儲存庫無法載入ERROR! SEGMENT DATA CORRUPTED, SEGDATA=

ERROR! SEGMENT DATA CORRUPTED, SEGDATA=

ERROR! SEGMENT DATA CORRUPTED, SEGDATA=
```

native_stderr.log

```
於 com.ibm.keymanager.KeyManagerException：預設的金鑰儲存庫無法載入
於 com.ibm.keymanager.keygroups.KeyGroupManager.init(KeyGroupManager.java:605)
於 com.ibm.keymanager.EKMServer.c(EKMServer.java:243)
於 com.ibm.keymanager.EKMServer.<init>(EKMServer.java:753)
於 com.ibm.keymanager.EKMServer.a(EKMServer.java:716)
於 com.ibm.keymanager.EKMServer.main(EKMServer.java:129)
```

為 CLI 用戶端及 EKM 伺服器之間的通訊問題進行除錯

EKM CLI 用戶端及 EKM 伺服器之間的通訊，是透過伺服器及用戶端配置內容檔中的 TransportListener.ssl.port 內容所指定的埠來進行的，並受到 SSL 的保護。

下列是用戶端無法連接至 EKM 伺服器的幾個可能原因。其中包括如何判斷問題及更正問題的步驟。

- EKM 伺服器並未執行，因此用戶端沒有通訊的對象。
  1. 請在指令視窗中下達 **netstat –an** 指令，並確認由 TransportListener.ssl.port 及 TransportListener.tcp.port 內容在 EKM 伺服器內容檔中所指定的埠是否顯示出來。如果未顯示這些埠，就表示伺服器不在執行中
- EKM CLI 用戶端內容檔中的 TransportListener.ssl.host 內容並未指向 EKM 伺服器所在的正確主機上。
  1. EKM CLI 用戶端內容檔中，TransportListener.ssl.host 內容的值預設爲 `localhost`。請修改這個內容的值來指向正確的主機。
- EKM 伺服器和 EKM CLI 用戶端不在相同的埠上對談。
  1. 檢查 EKM 伺服器和 EKM CLI 用戶端內容檔中的 TransportListener.ssl.port 內容，是否設定爲相同的值。
- EKM 伺服器和 EKM CLI 用戶端找不到安全通訊所需要的一般憑證。
  1. 確定 CLI 用戶端內容 TransportListener.ssl.keystore 和 TransportListener.ssl.truststore 所指定的金鑰儲存庫含有與伺服器內容中之 Admin.ssl.keystore 和 Admin.ssl.truststore 金鑰儲存庫相同的憑證。
  2. 確定用戶端內容中的 TransportListener.ssl.keystore.password 有正確的密碼。
  3. 確定在這些金鑰儲存庫中，沒有任何已到期的憑證。 JSSE 不會使用過期憑證來進行安全通訊。
- EKM CLI 用戶端內容檔是唯讀的。
  1. 請檢查檔案的屬性或權限，確認正在執行 EKM CLI 用戶端的使用者具有存取或修改檔案的權限。
- EKM 伺服器內容檔雖有 Server.authMechanism = LocalOS，但是並未安裝 EKMServicesAndSamples 套件中的必要檔案或是檔案的安裝位置錯誤。
  1. 請參閱 EKMServiceAndSamples 套件檢附的 ReadMe，以取得關於鑑別的詳細資訊。

金鑰管理程式伺服器問題的除錯

與金鑰管理程式相關的問題，大部分都涉及金鑰管理程式伺服器的配置或啓動。請參閱「附錄 B，預設配置檔」，以取得指定除錯內容的相關資訊。

如果 Encryption Key Manager 無法啓動，請檢查防火牆。

可能是軟體防火牆或硬體防火牆的封鎖，使 Encryption Key Manager 無法存取埠。

EKM 伺服器未啓動。無法載入或找到 EKM.properties 配置。

1. 當啓動 KMSAdminCmd 或 EKMLaunch，但並未指定 **KeyManagerConfig.properties** 的完整路徑時，如果內容檔不在預設路徑上，就會發生這個錯誤。

Windows 的預設路徑是 **C:/Program Files/IBM/KeyManagerServer/**

Linux 平台的預設路徑是 **/opt/ibm/KeyManagerServer/**

2. 重新輸入指令來啓動 KMSAdminCmd，且併入 **KeyManagerConfig.properties** 檔的完整路徑。請參閱「附錄 B，Encryption Key Manager 配置內容檔」，以取得詳細資訊。

EKM 伺服器未啓動。您必須在配置檔中，指定 XML meta 資料檔的檔名。

配置檔遺漏 Audit.metadata.file.name 項目。

如果要更正這個問題，請新增 Audit.metadata.file.name 內容到 **KeyManagerConfig.properties** 配置檔中。

無法啓動 EKM.Mykeys。系統找不到指定的檔案。

1. 當 **KeyManagerConfig.properties** 中的金鑰儲存庫項目未指向現有檔案時，便會出現這個錯誤訊息。
2. 如果要更正這個問題，請確定 **KeyManagerConfig.properties** 檔中的下列項目指向有效的現有金鑰儲存庫檔：

   Admin.ssl.keystore.name

   TransportListener.ssl.truststore.name

   TransportListener.ssl.keystore.name

   Admin.ssl.truststore.name

   請參閱「附錄 B，Encryption Key Manager 配置內容檔」，以取得詳細資訊。

無法啓動 EKM。檔案不存在 = safkeyring://xxx/yyy

錯誤原因可能是在 Encryption Key Manager 環境 Shell Script 的 IJO 變數中，指定了錯誤的提供者。

對於 JCECCARACFKS 金鑰儲存庫，請使用：

```
-Djava.protocol.handler.pkgs=com.ibm.crypto.hdwrCCA.provider
```

對於 JCERACFKS 金鑰儲存庫，請使用：

```
-Djava.protocol.handler.pkgs=com.ibm.crypto.provider
```

無法啓動 EKM。金鑰儲存庫遭到篡改，或密碼不正確。

1. 如果內容檔中的一或多個這些項目含有錯誤值（請參閱「附錄 B，Encryption Key Manager 配置內容檔」），便會發生這個錯誤：

   config.keystore.password（對應於 config.keystore.file）

   admin.keystore.password（對應於 admin.keystore.name）

   transportListener.keystore.password（對應於 transportListener.keystore.name）
2. 如果伺服器啓動時，在密碼提示之下輸入錯誤密碼，也可能出現這個錯誤。
3. 如果配置不含其中的任何密碼，而內容檔中的三個金鑰儲存庫項目都是唯一的，則最多會向您發出三次提示。如果內容中的所有項目都相同，便提示一次。

無法啓動 EKM。金鑰儲存庫格式無效。

1. 當內容檔中的金鑰儲存庫項目之一指定了錯誤的金鑰儲存庫類型時，便可能出現這個錯誤。
2. 如果內容檔中的所有金鑰儲存庫項目都指向相同的檔案，Encryption Key Manager 會利用 config.keystore.type 值作爲所有金鑰儲存庫的金鑰儲存庫類型。
3. 當內容檔中沒有特定金鑰儲存庫的類型項目時，Encryption Key Manager 會假設類型爲 `jceks`。

無法啓動伺服器。接聽器執行緒未啓動，不在執行中。

發生這個錯誤有許多可能的原因：

1. **KeyManagerConfig.properties** 檔中的下列兩個項目指向相同的埠：

   TransportListener.ssl.port

   TransportListener.tcp.port

   每個傳輸接聽器都必須配置成利用本身的埠來接聽。
2. 這兩個項目或其中一個項目所配置的埠，已由金鑰管理程式伺服器的相同機器上所執行的另一項服務使用中。請找出沒有其他服務在使用的埠，利用這些埠來配置金鑰管理程式伺服器。
3. 在執行 Linux 作業系統的系統上，如果這兩個埠或其中一個埠小於 1024，且啓動金鑰管理程式伺服器的使用者不是 root，便可能發生這個錯誤。請修改 **KeyManagerConfig.properties** 中的傳輸接聽器項目來使用 1024 以上的埠。

“[Fatal Error] :-1:-1: Premature end of file.” 訊息出現在 native_stderr.log。

當 Encryption Key Manager 載入空白金鑰群組檔案時，會出現這個訊息。這個訊息來自 XML 剖析器，除非 Encryption Key Manager 配置成使用金鑰群組，且 **KeyManagerConfig.properties**（Encryption Key Manager 伺服器內容檔）中的 config.keygroup.xml.file 內容所指定的檔案已毀損，否則，它仍會啓動。

錯誤：無法在配置金鑰儲存庫中，利用 alias:MyKey 找到 Secretkey。

內容檔中的 symmetricKeySet 項目包含在 config.keystore.file 中並不存在的金鑰別名。

如果要更正這個問題，請修改配置檔中的 symmetricKeySet 項目，使它只包含 **KeyManagerConfig.properties** 中的 config.keystore.file 項目所指示之金鑰儲存庫檔案中現存的別名，或新增遺漏的對稱金鑰到金鑰儲存庫中。請參閱「附錄 B，Encryption Key Manager 配置內容檔」，以取得詳細資訊。

symmetricKeySet 中沒有對稱金鑰，因此無法支援 LTO 磁帶機。

這是一則參考訊息。 Encryption Key Manager 伺服器仍會啓動，但 Encryption Key Manager 的這個實例無法支援 LTO 磁帶機。如果未配置任何 LTO 磁帶機與這個 Encryption Key Manager 通訊，這便不是問題。

Encryption Key Manager 報告錯誤

這一節定義 Encryption Key Manager 所報告，且在磁帶機感應資料中傳回的錯誤訊息。它們通常稱爲錯誤症狀碼或 FSC。這份表格包含錯誤碼、失敗的簡要說明以及更正動作。請參閱「附錄 B，預設配置檔」，以取得指定除錯內容的相關資訊。

表 6-1. Encryption Key Manager 報告的錯誤

錯誤碼	說明	動作
EE02	加密讀取訊息失敗：DriverErrorNotifyParameterError：「收到不當的 ASC 和 ASCQ。ASC 和 ASCQ 不符合「金鑰建立/金鑰轉換/金鑰擷取」作業。」	磁帶機要求不支援的動作。確定您所執行的是最新版本的 Encryption Key Manager（請參閱 第 3-1 頁的『下載最新版本的 Key Manager ISO Image』以判斷是否爲最新版本）。請檢查磁帶機或 Proxy 伺服器韌體的版本，必要的話，將它們更新成最新的版本。請在金鑰管理程式伺服器上啓用除錯追蹤。請嘗試重建問題並收集除錯日誌。如果仍有問題，請參閱本出版品底端「請先閱讀」一節中的「聯絡 Dell」，以取得技術協助的相關資訊。
EE0F	加密邏輯錯誤：內部錯誤：「非預期的錯誤。EKM 內部程式設計錯誤。」	確定您所執行的是最新版本的 Encryption Key Manager（請參閱 第 3-1 頁的『下載最新版本的 Key Manager ISO Image』以判斷是否爲最新版本）。請檢查磁帶機或 Proxy 伺服器韌體的版本，必要的話，將它們更新成最新的版本。請在金鑰管理程式伺服器上啓用除錯追蹤。請嘗試重建問題並收集除錯日誌。如果仍有問題，請參閱本出版品底端「請先閱讀」一節中的「聯絡 Dell」，以取得技術協助的相關資訊。
	錯誤：呼叫 CSNDDSV，傳回硬體錯誤，returnCode 12 reasonCode 0。	如果使用硬體加密法，請確定已啓動 ICSF。
EE23	加密讀取訊息失敗：內部錯誤：「非預期的錯誤.....」	發生一般錯誤，無法剖析從磁帶機或 Proxy 伺服器收到的訊息。確定您所執行的是最新版本的 Encryption Key Manager（請參閱 第 3-1 頁的『下載最新版本的 Key Manager ISO Image』以判斷是否爲最新版本）。請在金鑰管理程式伺服器上啓用除錯。請嘗試重建問題並收集除錯日誌。如果仍有問題，請參閱本出版品底端「請先閱讀」一節中的「聯絡 Dell」，以取得技術協助的相關資訊。

表 6-1. Encryption Key Manager 報告的錯誤 (繼續)

錯誤碼	說明	動作
EE25	加密配置問題：發生與磁帶機表格相關的錯誤。	如果在 KeyManagerConfig.properties 檔中提供了 config.drivetable.file.url，請確定這個參數正確。在 Encryption Key Manager 伺服器上執行 `listdrives -drivename <drivename>` 指令，驗證磁帶機的配置是否正確（例如，磁帶機序號、別名、憑證等）。確定您所執行的是最新版本的 Encryption Key Manager（請參閱 第 3-1 頁的『下載最新版本的 Key Manager ISO Image』以判斷是否爲最新版本）。請檢查磁帶機或 Proxy 伺服器韌體的版本，必要的話，將它們更新成最新的版本。請啓用除錯追蹤，再重試作業。如果仍有問題，請參閱本出版品底端「請先閱讀」一節中的「聯絡 Dell」 ，以取得技術協助的相關資訊。
EE29	加密讀取訊息失敗：簽章無效	從磁帶機或 Proxy 伺服器收到的訊息與簽章不符。確定您所執行的是最新版本的 Encryption Key Manager（請參閱 第 3-1 頁的『下載最新版本的 Key Manager ISO Image』以判斷是否爲最新版本）。請在金鑰管理程式伺服器上啓用除錯。請嘗試重建問題並收集除錯日誌。如果仍有問題，請參閱本出版品底端「請先閱讀」一節中的「聯絡 Dell」 ，以取得技術協助的相關資訊。
EE2B	加密讀取訊息失敗：內部錯誤：「DSK 不含簽章，或無法驗證 DSK 中的簽章。」	確定您所執行的是最新版本的 Encryption Key Manager（請參閱 第 3-1 頁的『下載最新版本的 Key Manager ISO Image』以判斷是否爲最新版本）。請檢查磁帶機或 Proxy 伺服器韌體的版本，必要的話，將它們更新成最新的版本。請在金鑰管理程式伺服器上啓用除錯追蹤。請嘗試重建問題並收集除錯日誌。如果仍有問題，請參閱本出版品底端「請先閱讀」一節中的「聯絡 Dell」 ，以取得技術協助的相關資訊。
EE2C	加密讀取訊息失敗：QueryDSKParameterError：「從裝置剖析 QueryDSKMessage 時發生錯誤。非預期的 dsk 計數或非預期的有效負載。」	磁帶機要求 Encryption Key Manager 執行不支援的功能。確定您所執行的是最新版本的 Encryption Key Manager（請參閱 第 3-1 頁的『下載最新版本的 Key Manager ISO Image』以判斷是否爲最新版本）。請檢查磁帶機或 Proxy 伺服器韌體的版本，必要的話，將它們更新成最新的版本。請在金鑰管理程式伺服器上啓用除錯追蹤。請嘗試重建問題並收集除錯日誌。如果仍有問題，請參閱本出版品底端「請先閱讀」一節中的「聯絡 Dell」 ，以取得技術協助的相關資訊。

表 6-1. Encryption Key Manager 報告的錯誤 (繼續)

錯誤碼	說明	動作
EE2D	加密讀取訊息失敗：訊息類型無效	Encryption Key Manager 收到未依照順序的訊息，或收到不知如何處理的訊息。確定您所執行的是最新版本的 Encryption Key Manager（請參閱 第 3-1 頁的『下載最新版本的 Key Manager ISO Image』以判斷是否爲最新版本）。請在金鑰管理程式伺服器上啓用除錯。請嘗試重建問題並收集除錯日誌。如果仍有問題，請參閱本出版品底端「請先閱讀」一節中的「聯絡 Dell」 ，以取得技術協助的相關資訊。
EE2E	加密讀取訊息失敗：內部錯誤：簽章類型無效	從磁帶機或 Proxy 伺服器收到的訊息不含有效簽章類型。確定您所執行的是最新版本的 Encryption Key Manager（請參閱 第 3-1 頁的『下載最新版本的 Key Manager ISO Image』以判斷是否爲最新版本）。請在金鑰管理程式伺服器上啓用除錯。請嘗試重建問題並收集除錯日誌。如果仍有問題，請參閱本出版品底端「請先閱讀」一節中的「聯絡 Dell」 ，以取得技術協助的相關資訊。
EE30	禁止的要求。	要求磁帶機所不支援的作業。請針對目標磁帶機輸入正確且支援的指令。
EE31	加密配置問題：發生與金鑰儲存庫相關的錯誤。	請檢查您試圖使用或配置爲預設值的金鑰標籤。您可以利用 listcerts 指令列出 Encryption Key Manager 所能使用的憑證。如果知道您在嘗試使用預設值，請在 Encryption Key Manager 伺服器上執行 listdrives -drivename *drivename* 指令，驗證磁帶機的配置是否正確（例如，磁帶機序號及相關聯的別名/金鑰標籤正確）。如果問題所在的磁帶機沒有相關聯的別名/金鑰標籤，請檢查 default.drive.alias1 和 default.drive.alias2 的值。如果這沒有用，或別名/金鑰標籤已存在，請收集除錯日誌，並請參閱本出版品底端「請先閱讀」一節中的「聯絡 Dell」 在本出版品前面的「請先閱讀」一節中，，以取得技術協助的相關資訊。
EE32	金鑰儲存庫相關問題。	最可能的原因是利用含有不同金鑰的不同 Encryption Key Manager 來加密磁帶，或用來加密這個磁帶的金鑰已重新命名，或已從金鑰儲存庫中刪除。請發出 `list -keysym`，確定要求的別名存在於金鑰儲存庫中。
EEE1	加密邏輯錯誤：內部錯誤：「非預期的錯誤：EK/EEDK 旗標與子頁面衝突。」	確定您所執行的是最新版本的 Encryption Key Manager（請參閱 第 3-1 頁的『下載最新版本的 Key Manager ISO Image』以判斷是否爲最新版本）。請檢查磁帶機或 Proxy 伺服器韌體的版本，必要的話，將它們更新成最新的版本。請在金鑰管理程式伺服器上啓用除錯。請嘗試重建問題並收集除錯日誌。如果仍有問題，請參閱本出版品底端「請先閱讀」一節中的「聯絡 Dell」 ，以取得技術協助的相關資訊。

表 6-1. Encryption Key Manager 報告的錯誤 (繼續)

錯誤碼	說明	動作
EF01	加密配置問題：「未配置磁帶機。」	試圖與 Encryption Key Manager 通訊的磁帶機不在磁帶機表格中。如果在 KeyManagerConfig.properties 檔中提供了 config.drivetable.file.url，請確定這個參數正確。請執行 `listdrives` 指令來檢查磁帶機是否在清單中。如果不在，請以正確的磁帶機資訊，利用 `adddrive` 指令來手動配置磁帶機，或利用 `modconfig` 指令，將 ″drive.acceptUnknownDrives″ 內容設爲 true。請啓用除錯追蹤，再重試作業。如果仍有問題，請參閱本出版品前面『請先閱讀』一節中的『聯絡 Dell』 ，以取得技術協助的相關資訊。

訊息

Encryption Key Manager 可以產生下列訊息，這些訊息會顯示在管理主控台中。

未指定配置檔

文字

```
Configuration file not specified:  KeyManager Configuration file not specified when starting EKM.
```

說明

KMSAdmin 指令要求將配置檔當成指令行參數傳入。

系統回應

程式停止。

操作員回應

請提供配置檔，再重試指令。

無法新增磁帶機

文字

```
Failed to add drive. Drive already exists.
```

說明

磁帶機已配置了 Encryption Key Manager，並已在磁帶機表格中，所以 **adddrive** 指令失敗。

操作員回應

請執行 **listdrives** 指令，以瞭解磁帶機是否已配置 Encryption Key Manager。如果磁帶機已存在，您可以利用 **moddrive** 指令來變更磁帶機配置。請執行 **help**，以取得詳細資訊。

無法保存日誌檔

文字

```
Failed to archive the log file.
```

說明

無法重新命名日誌檔。

操作員回應

請檢查檔案許可權及這部磁帶機上的空間。

無法刪除配置

文字

```
"modconfig" command failed.
```

說明

無法利用 **modconfig** 指令來刪除 Encryption Key Manager 配置。

操作員回應

請利用 **help** 檢查指令語法，確定提供的參數正確。請檢查審核日誌，以取得詳細資訊。

無法刪除磁帶機項目

文字

```
"deldrive" command failed.
```

說明

deldrive 指令無法刪除磁帶機表格中的磁帶機項目。

操作員回應

請利用 **help** 檢查指令語法，確定提供的參數正確。請確定已利用 **listdrives** 指令，以 Encryption Key Manager 配置了磁帶機。請檢查審核日誌，以取得詳細資訊。

無法匯入

文字

```
"import" command failed.
```

說明

無法匯入磁帶機表格或配置檔。

系統回應

Encryption Key Manager 伺服器未啓動。

操作員回應

請確定指定的 URL 存在，且具有讀取許可權。請利用 **help** 檢查指令語法。請確定參數正確，再重試一次。

無法修改配置

文字

```
"modconfig" command failed.
```

說明

無法利用 **modconfig** 指令來修改 Encryption Key Manager 配置。

操作員回應

請利用 **help** 檢查指令語法，確定提供的參數正確。請檢查審核日誌，以取得詳細資訊。

檔名不能是空值

文字

```
File name was not supplied for audit log file.
```

說明

未利用 Encryption Key Manager 的配置內容來提供審核檔名稱。這個參數是必要的配置參數。

系統回應

程式停止。

操作員回應

請確認在提供給 Encryption Key Manager 的配置內容檔中定義了 `Audit.handler.file.name` 內容，再嘗試重新啓動它。

檔案大小限制不能是負數

文字

```
Maximum file size for audit log can not be a negative number.
```

說明

Encryption Key Manager 配置檔中的 `Audit.handler.file.size` 內容值必須是正數。

系統回應

Encryption Key Manager 未啓動。

操作員回應

請指定有效的數字給 Audit.handler.file.size，再嘗試重新啓動 Encryption Key Manager。

沒有要同步化的資料

文字

```
No data can be found to be synchronized with "sync".
```

說明

sync 指令無法識別任何要同步化的資料。

操作員回應

請確認提供的配置檔存在，且配置檔利用 *config.drivetable.file.url* 配置了正確的磁帶機表格。請利用 **help** 檢查語法，再重試 **sync** 指令。

無效輸入

文字

```
Invalid input parameters for the CLI.
```

說明

特定指令語法可能不正確。

操作員回應

請確定輸入的指令正確。請利用 **help** 檢查指令語法。請確定提供的參數正確，再重試一次。

配置檔中的 SSL 埠號無效

文字

```
Invalid SSL port number specified in the EKM configuration file.
```

說明

配置檔所提供的 SSL 埠號不是有效的號碼。

系統回應

Encryption Key Manager 未啓動。

操作員回應

啓動 Encryption Key Manager 時，請在配置檔的 TransportListener.ssl.port 內容中指定有效的埠號，再嘗試重新啓動。

配置檔中的 TCP 埠號無效

文字

```
Invalid TCP port number specified in the EKM configuration file.
```

說明

配置檔所提供的 TCP 埠號不是有效的號碼。

系統回應

Encryption Key Manager 未啓動。

操作員回應

啓動 Encryption Key Manager 時，請在配置檔的 TransportListener.tcp.port 內容中指定有效的埠號，再嘗試重新啓動。預設 TCP 埠號是 3801。

必須在配置檔中指定 SSL 埠號

文字

```
SSL port number is not configured in the properties file.
```

說明

SSL 埠號是配置內容檔中必須配置的內容。在多台伺服器環境中，Encryption Key Manager 伺服器利用它來彼此通訊。

系統回應

Encryption Key Manager 未啓動。

操作員回應

請在 TransportListener.ssl.port 內容中指定有效的埠號，再嘗試重新啓動 Encryption Key Manager。

必須在配置檔中指定 TCP 埠號

文字

```
TCP port number is not configured in the properties file.
```

說明

TCP 埠號是配置內容檔中必須配置的內容。它用於磁帶機和 Encryption Key Manager 之間的通訊。

系統回應

Encryption Key Manager 未啓動。

操作員回應

請在 TransportListener.tcp.port 內容中指定有效的埠號，再嘗試重新啓動 Encryption Key Manager。預設 TCP 埠號是 3801。

伺服器無法啓動

文字

```
EKM server failed to start.
```

說明

發生配置問題，Encryption Key Manager 伺服器無法啓動。

操作員回應

請檢查提供之配置檔中的參數。請查看日誌，以取得詳細資訊。

同步失敗

文字

```
"sync" command failed.
```

說明

將兩部 Encryption Key Manager 伺服器的資料同步化的同步作業失敗。

操作員回應

請確定指定給遠端 Encryption Key Manager 伺服器的 IP 位址正確，且這部電腦可供存取。請確定配置檔存在，且含有正確的磁帶機表格資訊。請利用 **help** 檢查 **sync** 指令語法。請查看日誌，以取得詳細資訊。

指定的審核日誌檔是唯讀的

文字

```
The audit log file can not be opened for writing.
```

說明

無法開啓 Encryption Key Manager 配置中 Audit.handler.file.name 內容所指定的審核日誌檔，以進行寫入。

系統回應

Encryption Key Manager 未啓動。

操作員回應

請檢查給定審核檔和目錄的許可權，再嘗試重新啓動 Encryption Key Manager。

無法載入管理金鑰儲存庫

文字

```
Keystore for Admin cannot be loaded.
```

說明

無法載入提供給 Encryption Key Manager 的管理金鑰儲存庫。在多台伺服器環境中，管理金鑰儲存庫是在 Encryption Key Manager 伺服器之間，用來進行伺服器端通訊。

系統回應

Encryption Key Manager 未啓動。

操作員回應

請檢查配置檔設定。請確定 Encryption Key Manager 配置檔中的 admin.keystore.file、admin.keystore.provider 和 admin.keystore.type 內容正確（請參閱「附錄 B」）、金鑰儲存庫檔存在，且有讀取權。請確定利用 admin.keystore.password 內容，或在指令行輸入，所提供給管理金鑰儲存庫的密碼正確。之後，再嘗試重新啓動 Encryption Key Manager。

無法載入金鑰儲存庫

文字

```
Keystore for EKM can not be loaded.
```

說明

無法載入指定給 Encryption Key Manager 的金鑰儲存庫。

系統回應

Encryption Key Manager 未啓動。

操作員回應

請檢查配置檔設定。請確定 Encryption Key Manager 配置檔中的 config.keystore.file、config.keystore.provider 和 config.keystore.type 內容正確、金鑰儲存庫檔存在，且有讀取權。請確定利用 config.keystore.password 內容，或在指令行輸入，而提供給 Encryption Key Manager 金鑰儲存庫的密碼正確。之後，再嘗試重新啓動。

無法載入傳輸金鑰儲存庫

文字

```
Transport keystore cannot be loaded.
```

說明

無法載入提供給 Encryption Key Manager 的傳輸金鑰儲存庫。在多台伺服器環境中，傳輸金鑰儲存庫是在 Encryption Key Manager 伺服器之間，用來進行用戶端通訊。

系統回應

Encryption Key Manager 未啓動。

操作員回應

請檢查配置檔設定。請確定 Encryption Key Manager 配置檔中的 transport.keystore.file、transport.keystore.provider 和 transport.keystore.type 內容正確、金鑰儲存庫檔存在，且有讀取權。請確定利用 transport.keystore.password 內容，或在指令行輸入，所提供給管理金鑰儲存庫的密碼正確。之後，再嘗試重新啓動 Encryption Key Manager。

不受支援的動作

文字

```
User entered action for the CLI which is not supported for EKM.
```

說明

Encryption Key Manager 不支援或無法解讀 **sync** 指令提供的動作。有效的動作是合併或重寫。

操作員回應

請利用 **help** 檢查指令語法，然後再試一次。

第 7 章 審核記錄

註: 本章所說明的審核記錄格式並不被視爲程式設計介面。這些記錄的格式在不同版本之間，可能有所不同。爲了預備某些需要剖析審核記錄的情況，本章說明了這些格式。

審核概觀

在 Encryption Key Manager 要求處理期間，發生各種可審核的事件時，審核子系統會將文字審核記錄寫到一組循序檔案中。審核子系統會寫入一個檔案中（目錄和檔名都可以配置）。這些檔案的檔案大小也可以配置。隨著記錄寫入檔案，在檔案大小到達可配置的大小之後，會關閉檔案，根據現行時間戳記來重新命名，再開啓另一個檔案，這時記錄會寫到新建的檔案中。因此，審核記錄的整體日誌會分成幾個可配置大小的檔案，它們的名稱依檔案大小超出可配置大小之時的時間戳記來循序指定。

如果要防止整體審核日誌的資訊量（跨越所有建立的循序檔案）過於龐大，超出檔案系統可用空間，您可以考慮建立 Script 或程式來監視所配置之審核目錄/資料夾/儲存器中的檔案集。當檔案關閉且根據時間戳記來命名時，應該會複製檔案內容，再附加到所需要的長期連續日誌位置，之後再加以清除。在執行時，請小心避免移除或變更 Encryption Key Manager 正在將記錄寫入其中的檔案（這個檔案的檔名不含時間戳記）。

審核配置參數

Encryption Key Manager 配置檔利用下列參數來控制將哪些事件記載到審核日誌中、在哪裡寫入審核日誌檔，以及審核日誌檔的大小上限。

Audit.event.types

語法

Audit.event.types={*type*[;*type*]}

用途

用來指定應該傳送到審核日誌的審核類型。配置參數可能的值如下：

all	所有事件類型
authentication	鑑別事件
data_synchronization	在 Encryption Key Manager 伺服器之間進行資訊同步化期間所發生的事件
runtime	發生的事件作爲傳送到 Encryption Key Manager 的處理作業和要求的一部份
configuration_management	進行配置變更時所發生的事件
resource_management	作爲 Encryption Key Manager 中之資源（磁帶機）設定時所發生的事件已變更

範例

這個配置值的範例規格如下：

```
Audit.event.types=all
```

另一個範例如下：

```
Audit.event.types=authentication;runtime;resource_management
```

Audit.event.outcome

語法

Audit.event.outcome={*outcome*[;*outcome*]}

用途

用來指出事件是因爲作業成功、作業不成功而發生，或是兩者都應該審核。請指定 **success** 來記載因作業成功而發生的事件。請指定 **failure** 來記載因作業不成功而發生的事件。

範例

這個配置值的範例規格如下：

```
Audit.event.outcome=failure
```

如果要啓用成功和不成功兩種情況：

```
Audit.event.outcome=success;failure
```

Audit.eventQueue.max

語法

Audit.eventQueue.max=*number_events*

用途

用來設定保留在記憶體佇列的事件物件數目上限。這是一個選用的參數，但建議採用。預設值是零。

範例

```
Audit.eventQueue.max=8
```

Audit.handler.file.directory

語法

Audit.handler.file.directory=*directoryName*

用途

這個參數用來指出審核記錄檔應該寫入哪個目錄。 請注意，如果目錄不存在，Encryption Key Manager 會試圖建立目錄。不過，如果不成功，Encryption Key Manager 便不會啓動。建議在執行 Encryption Key Manager 之前，目錄便已存在。另外，也請注意，用來執行 Encryption Key Manager 的使用者 ID 必須有指定目錄的寫入權。

範例

將目錄設爲 **/var/ekm/ekm1/audit**：

```
Audit.handler.file.directory=/var/ekm/ekm1/audit
```

Audit.handler.file.size

語法

Audit.handler.file.size=*sizeInKiloBytes*

用途

這個參數用來指出達到時會關閉審核檔再寫入新審核檔的大小限制。請注意，結果審核檔的實際大小有可能超出這個值幾個位元組，因爲檔案是在超出大小限制之後關閉。

範例

如果要將檔案大小上限設成大約 2 MB，請輸入：

```
Audit.handler.file.size=2000
```

Audit.handler.file.name

語法

Audit.handler.file.name=*fileName*

用途

請利用這個參數來指定在指定審核目錄內的基礎檔名，以便在建立審核日誌檔時，用來作爲基礎名稱。請注意，這個參數只能包含基本檔名，而不是完整路徑名稱。審核日誌檔的完整名稱會在這個名稱後加上檔案寫入時間的對應值。

爲了呈現這一點，請設想 Audit.handler.file.name 值設爲 **ekm.log** 的範例。檔案的完整名稱類似於：ekm.log.2315003554。附加的字串可以協助判斷審核日誌檔的建立順序 - 數值越高，審核日誌檔愈新。

範例

將基礎名稱設爲 **ekm.log** 的範例如下：

```
Audit.handler.file.name=ekm.log
```

Audit.handler.file.multithreads

語法

Audit.handler.file.multithreads={yes|true|no|false}

用途

如果指定爲 **true**，便會利用個別執行緒，將事件資料寫入審核日誌中，讓現行執行緒（作業）繼續執行，不需要等待審核日誌寫入完成。預設行爲是使用多個執行緒。

範例

將基礎名稱設爲 **true** 的範例如下：

```
Audit.handler.file.multithreads=true
```

Audit.handler.file.threadlifespan

語法

Audit.handler.file.threadlifespan=*timeInSeconds*

用途

這個參數用來指定爲了撰寫審核日誌項目，執行緒預期應該需要的最長時間。在清除處理程序期間，這個值可供執行緒在遭到岔斷之前，完成它們的作業。如果背景執行緒未在 threadlifespan 參數分配的時間內完成它的工作，在進行清除處理時，會將執行緒岔斷。

範例

如果要將執行緒撰寫審核日誌所應需要的預期時間設爲 10 秒，請指定：

```
Audit.handler.file.threadlifespan=10
```

審核記錄格式

所有審核記錄都使用這裡所說明的類似輸出格式。所有審核記錄都含有一些共用資訊（包括時間戳記和記錄類型）以及發生之審核事件特有的資訊。審核記錄的一般格式顯示如下：

```
AuditRecordType:[
  timestamp=timestamp
  Attribute Name=Attribute Value
  ...
  ]
```

在檔案中每個記錄都跨越多行，記錄的第一行從第一個字元開始是審核記錄類型，後面接著一個冒號 (:)，再接著一個左方括弧 ([)。關聯於相同審核記錄的後續各行縮排兩個 (2) 空格，使日誌記錄較容易閱讀。單一審核記錄的最後一行含有縮排兩個 (2) 空格的右方括弧 (])。每個審核記錄的行數會隨著審核記錄類型及審核記錄所提供的其他屬性資訊而不同。

審核記錄的時間戳記是以執行 Encryption Key Manager 之系統的系統時鐘爲基礎。如果這些記錄要根據時間戳記來關聯於其他系統所發生的事件，便應該使用某類型的時間同步化，以確保環境內各系統的時鐘會同步化到可接受的精確層次。

Encryption Key Manager 中的審核點

Encryption Key Manager 可以根據配置來撰寫要求處理期間發生的許多事件之審核記錄。這一節說明可審核的事件集及審核記錄配置種類，審核記錄配置種類必須啓用，這些審核記錄才能寫到審核檔中（請參閱表 7-1）。

表 7-1. Encryption Key Manager 寫入審核檔的審核記錄類型

審核記錄類型	審核類型	說明
鑑別	authentication	用來記載鑑別事件

表 7-1. Encryption Key Manager 寫入審核檔的審核記錄類型 (繼續)

審核記錄類型	審核類型	說明
資料同步化	data_synchronization	用來記載資料同步化處理程序
執行時期	runtime	用來記載處理要求期間，在 Encryption Key Manager 伺服器內發生的各種重要處理事件
資源管理	resource_management	用來記載資源如何配置給 Encryption Key Manager 的變更
配置管理	configuration_management	用來記載 Encryption Key Manager 伺服器的配置變更

審核記錄屬性

下列清單顯示每個審核記錄類型所能使用的屬性。

「鑑別」事件

這些記錄的格式如下：

```
Authentication event:[
  timestamp=timestamp
  event source=source
  outcome=outcome
  event type=SECURITY_AUTHN
  message=message
  authentication type=type
  users=users
  ]
```

請注意，只在有可用資訊之時，才會出現 message 值。

「資料同步化」事件

這些記錄的格式如下：

```
Data synchronization event:[
  timestamp=timestamp
  event source=source
  outcome=outcome
  event type=SECURITY_DATA_SYNC
  message=message
  action=action
  resource=resource
  user=user
  ]
```

請注意，只在有可用資訊之時，才會出現 message 和 user 值。

「執行時期」事件

這些記錄的格式如下：

```
Runtime event:[
  timestamp=timestamp
  event source=source
  outcome=outcome
  event type=SECURITY_RUNTIME
  message=message
```

```
  resource=resource
  action=action
  user=user
  ]
```

請注意，只在有可用資訊之時，才會出現 message 和 user 值。

「資源管理」事件

這些記錄的格式如下：

```
Resource management event:[
  timestamp=timestamp
  event source=source
  outcome=outcome
  event type=SECURITY_MGMT_RESOURCE
  message=message
  action=action
  user=user
  resource=resource
  ]
```

請注意，只在有可用資訊之時，才會出現 message 值。

「配置管理」事件

這些記錄的格式如下：

```
Configuration management event:[
  timestamp=timestamp
  event source=source
  outcome=outcome
  event type=SECURITY_MGMT_CONFIG
  message=message
  action=action
  command type=type
  user=user
  ]
```

請注意，只在有可用資訊之時，才會出現 message 值。

審核事件

表 7-2 說明導致建立審核記錄的事件。這份表格列出發生這個事件時所記載的審核記錄類型。

表 7-2. 審核記錄類型（依審核事件）

審核事件	審核記錄類型
已順利鑑別使用者	authentication
使用者鑑別失敗	authentication
資料已順利傳送到其他 EKM	data_synchronization
資料傳送到其他 EKM 時，發生錯誤	data_synchronization
已處理 sync 指令	data_synchronization
處理 sync 指令時，發生錯誤	data_synchronization
已啓動指令行處理程序	runtime
已收到 exit 指令	runtime
輸入不明指令	runtime

表 7-2. 審核記錄類型（依審核事件） (繼續)

審核事件	審核記錄類型
從磁帶機收到訊息	runtime
處理磁帶機訊息時，發生錯誤	runtime
從磁帶機收到的訊息發生錯誤	runtime
利用從磁帶機收到的資訊來更新磁帶機表格時，發生錯誤	runtime
從磁帶機表格擷取資訊時，發生錯誤	runtime
從金鑰儲存庫擷取資訊時，發生錯誤	runtime
處理來自金鑰儲存庫的憑證時，發生錯誤	runtime
尋找來自金鑰儲存庫的私密金鑰時，發生錯誤	runtime
計算加密值時，發生錯誤	runtime
已順利處理訊息交換	runtime
已啓動訊息處理程序	runtime
已啓動指令行處理程序	runtime
使用加密服務時，發現問題	runtime
探索到新的磁帶機	runtime
將磁帶機配置到磁帶機表格時，發生錯誤	runtime
已順利開始處理來自磁帶機的訊息	runtime
已收到和處理 stopekm 指令	runtime
已從磁帶機表格移除磁帶機	resource_management
從磁帶機表格移除磁帶機時，發生錯誤	resource_management
磁帶機表格匯入成功	resource_management
匯入磁帶機表格時，發生錯誤	resource_management
磁帶機表格匯出成功	resource_management
匯出磁帶機表格時，發生錯誤	resource_management
listcerts 指令成功	resource_management
磁帶機新增到磁帶機表格成功	resource_management
新增磁帶機到磁帶機表格時，發生錯誤	resource_management
listdrives 指令成功	resource_management
處理 listdrives 指令時，發生錯誤	resource_management
磁帶機表格修改成功	resource_management
修改磁帶機表格時，發生錯誤	resource_management
金鑰儲存庫開啓成功	resource_management
開啓金鑰儲存庫時，發生錯誤	resource_management
已變更配置內容	configuration_management
變更配置內容時，發生錯誤	configuration_management
已刪除配置內容	configuration_management
刪除配置內容時，發生錯誤	configuration_management
配置匯入成功	configuration_management
匯入配置時，發生錯誤	configuration_management
配置匯出成功	configuration_management

表 7-2. 審核記錄類型（依審核事件）(繼續)

審核事件	審核記錄類型
匯出配置時，發生錯誤	configuration_management
listconfig 指令成功	configuration_management

第 8 章 使用 meta 資料

您必須配置 Encryption Key Manager 來建立 XML 檔，使這個檔案能夠擷取要成爲加密資料的重要資訊，以及將它寫到磁帶中。可利用磁區序號來查詢這個檔案，以顯示磁區所用的別名或金鑰標籤。反過來說，也可利用別名來查詢這個檔案，以顯示這個金鑰標籤/別名的所有相關磁區。

註: 如果您未配置 meta 資料檔，Encryption Key Manager 便不會啓動。

執行加密處理程序時，Encryption Key Manager 會收集下列資料：

- 磁帶機序號
- 磁帶機 WorldWideName
- 建立日期
- 金鑰別名 1
- 金鑰別名 2
- DKi
- VolSer

收集的資料到了特定限制時，會寫到 XML 檔中。預設限制是 100 筆記錄，您可以在 Encryption Key Manager 內容檔 (KeyManagerConfig.properties) 中設定這項限制。寫入檔案之後，只要 Encryption Key Manager 在執行中，便可以查詢這個檔案。爲了防止檔案變得過於龐大，在達到檔案大小上限之後，會自動輪替爲新的檔案。輪替的檔案大小上限預設值是 1 MB，您也可以在 Encryption Key Manager 內容檔中設定這個上限。只有目前這個及上一個檔案版本會儲存起來。 Encryption Key Manager 配置內容檔中所設的值如下：

Audit.metadata.file.name
儲存 meta 資料的 XML 檔名稱。這是必要的。

Audit.metadata.file.size
檔案從目前這個版本輪替到上一版本之前的檔案大小上限（以 KB 爲單位指定）。這是選用的。預設值是 1024 (1MB)。

Audit.metadata.file.cachecount
寫入 meta 資料檔之前的快取記錄數目。這是選用的。預設值是 100。

XML 檔格式

這個檔案包含的記錄格式如下。

```
<KeyUsageEvent>
 <DriveSSN>FVTDRIVE0000</driveSSN>          -磁帶機序號
 <VolSer>TESTER</volSer>                    -磁區序號
 <DriveWWN>57574E414D453030</driveWWN>      -磁帶機 WWN
 <keyAlias2>cert2</keyAlias2>               -Key Alias1
```

<keyAlias1>cert1</keyAlias1> - keyAlias2
<dateTime>Tue Feb 20 09:18:07 CST 2007</dateTime> - 建立日期
</KeyUsageEvent>

附註：LTO 4 和 LTO 5 磁帶機只有 <keyAlias1></keyAlias1> 記錄，且會記錄 DKi。

查詢 meta 資料 XML 檔

請利用 EKMDataParser 工具來查詢 meta 資料檔。這個工具利用「文件物件模型 (DOM)」技術來剖析 XML 檔，無法從 Encryption Key Manager 指令行介面來執行。它的呼叫方式如下：

java com.ibm.keymanager.tools.EKMDataParser -filename *full_path_to_metadata_file* {**-volser** *volser* | **-keyalias** *alias*}

metadata_path
這是 **KeyManagerConfig.properties** 檔 Audit.metadata.file.name 中之 meta 資料檔所指定的相同目錄路徑。

-filename
filename 是必要的，必須是 XML meta 資料檔的名稱。通常符合 **KeyManagerConfig.properties** 檔中的 Audit.metadata.file.name 內容所指定的名稱。

-volser
您在 XML 檔中搜尋之磁帶匣的磁區序號。必須指定 **-volser** 或 **-keyalias**。

-keyalias
您在 XML 檔中搜尋的金鑰標籤或別名。必須指定 **-volser** 或 **-keyalias**。

範例

假設 **KeyManagerConfig.properties** 中的 meta 資料檔名內容 (Audit.metadata.file.name) 設爲 `metadata` 的值，且檔案是在 Encryption Key Manager 執行所在的本端目錄中，下列指令只會過濾（顯示）與 volser `72448` 相關的 XML 記錄：

```
<jvm_path>/bin/java com.ibm.keymanager.tools.EKMDataParser -filename metadata -volser 72448
```

輸出格式如下：

表 8-1. meta 資料查詢輸出格式

keyalias1	keyalias2	volSer	dateTime	driveSSN	dki
cert1	cert2	72448	Wed Mar 14 10:31:32 CDT 2007	FVTDRIVE0004	

從毀損的 meta 資料檔回復

如果 Encryption Key Manager 以不當的方式關機，或執行 Encryption Key Manager 的系統毀損，Encryption Key Manager meta 資料檔可能會毀損。以不當的方式編輯或修正 meta 資料檔也可能使它毀損。在 EKMDataParser 剖析 meta 資料檔之前，無法察覺毀損。 EKMDataParser 可能會失敗，且出現類似下列錯誤：

```
[Fatal Error] EKMData.xml:290:16: The end-tag for element type "KeyUsageEvent" must
 end with a '>' delimiter.
org.xml.sax.SAXParseException: The end-tag for element type "KeyUsageEvent" must
 end with a '>' delimiter.
at org.apache.xerces.parsers.DOMParser.parse(Unknown Source)
```

```
at org.apache.xerces.jaxp.DocumentBuilderImpl.parse(Unknown Source)
at javax.xml.parsers.DocumentBuilder.parse(Unknown Source)
at com.ibm.keymanager.tools.EKMDataParser.a(EKMDataParser.java:136)
at com.ibm.keymanager.tools.EKMDataParser.a(EKMDataParser.java:26)
at com.ibm.keymanager.tools.EKMDataParser.main(EKMDataParser.java:93)
```

如果發生這個錯誤，原因在於遺漏元素的 XML 結束標籤。您可以回復 Encryption Key Manager meta 資料檔，使 EKMDataParser 能夠重新剖析檔案。

1. 建立 Encryption Key Manager meta 資料檔的備份副本。
2. 編輯 Encryption Key Manager meta 資料檔。
3. 在 XML 中，每一段資料或事件都應該有起始標籤及對應的結束標籤。
   - 一些起始標籤的範例如下：
     - <KeyUsageEvent>
     - <driveSSN>
     - <keyAlias1>
   - 一些結束標籤的範例如下：
     - </KeyUsageEvent>
     - </driveSSN>
     - </keyAlias1>
4. 掃描檔案，尋找不相符的標籤。 EKMDataParser 的錯誤訊息會列出遺漏結束標籤的標籤。這應該會使搜尋簡單一些。
5. 找到不相符的標籤時，請暫時刪除事件，或新增必要的標籤來完成事件。
   - 比方說，下列 Encryption Key Manager meta 資料檔摘錄顯示沒有結束標籤的第一個 KeyUsageEvent：

     ```
     <KeyUsageEvent>
     <driveSSN>001310000109</driveSSN>
     <volSer>      </volSer>
     <driveWWN>5005076312418B07</driveWWN>
     <keyAlias1>key0000000000000000F</keyAlias1>
     <dki>6B65790000000000000000F</dki>
     <dateTime>Thu Aug 30 09:50:53 MDT 2007</dateTime>
     <KeyUsageEvent>
     <driveSSN>001310000100</driveSSN>
     <volSer>      </volSer>
     <driveWWN>5005076312418ABB</driveWWN>
     <keyAlias1>key00000000000000000000</keyAlias1>
     <dki>6B6579000000000000000000</dki>
     <dateTime>Thu Sep 06 16:49:39 MDT 2007</dateTime>
     </KeyUsageEvent>
     ```

     在 <dateTime>Thu Aug 30 09:50:53 MDT 2007</dateTime> and <KeyUsageEvent> 這兩行之間新增 </KeyUsageEvent>，便可完成第一個 <KeyUsageEvent>。

修復檔案毀損，可讓 EKMDataParser 順利剖析資料。

附錄 A. 範例檔

範例啓動常駐程式 Script

警告: 保留金鑰儲存庫的資料極爲重要，不容忽視。當無法存取金鑰儲存庫時，您也無法將加密的磁帶解密。請務必將金鑰儲存庫和密碼資訊儲存起來。

Linux 平台

以下是可讓您在背景中，用已獲證明的方式來啓動 EKM 的範例 Script。這份 Script 會啓動 EKM 以及透過 Script 來傳入金鑰儲存庫密碼 *keystore_password*。依照這個方式，便不需要將金鑰儲存庫密碼放在 EKM 配置檔中。（請參閱下列附註）。這個 Script 檔必須包含下列內容：

```
java com.ibm.keymanager.KMSAdminCmd KeyManagerConfig.properties <<EOF
startekm
keystore_password
status
EOF
```

註: 如果是利用 Script 將金鑰儲存庫密碼輸入 EKM（也就是說，EKM 配置檔不含金鑰儲存庫密碼），當備份 EKM 時，並不必然需要將各個檔案（配置檔、磁帶機表格和金鑰儲存庫備份檔）當作秘密來處理，不過，含有金鑰儲存庫密碼的 Script **必須**安全而靈活地儲存起來（比方說，將多份副本儲存在多個位置）。金鑰儲存庫密碼是機密資訊，必須依照這個方式來處理。安全地備份 Script 檔與備份含有金鑰儲存庫密碼的配置檔，有相同的選項。不過，Script 可以在 EKM 備份檔之外，獨自安全地備份和儲存/傳輸，這增加了一個間接的安全層次。最後，我們必須強調，不論金鑰儲存庫密碼是如何儲存（在 Script 中，或在 EKM 的配置檔中），它的儲存都必須既安全又靈活，以便隨時能夠回復金鑰儲存庫密碼。**失去金鑰儲存庫密碼的所有副本，即會遺失金鑰儲存庫中所有的金鑰，再也無法回復。**

範例配置檔

以下是範例 EKM 內容檔，所有金鑰儲存庫項目都指向相同的軟體金鑰儲存庫：

```
Admin.ssl.keystore.name = /keymanager/testkeys
Admin.ssl.keystore.type = jceks
Admin.ssl.truststore.name = /keymanager/testkeys
Admin.ssl.truststore.type = jceks
Audit.event.outcome = success,failure
Audit.event.types = all
Audit.eventQueue.max = 0
Audit.handler.file.directory = /keymanager/audit
Audit.handler.file.name = kms_audit.log
Audit.handler.file.size = 10000
Audit.metadata.file.name = /keymanager/metafile.xml
config.drivetable.file.url = FILE:///keymanager/drivetable
config.keystore.file = /keymanager/testkeys
config.keystore.provider = IBMJCE
config.keystore.type = jceks
fips = Off
```

```
TransportListener.ssl.ciphersuites = JSSE_ALL
TransportListener.ssl.clientauthentication = 0
TransportListener.ssl.keystore.name = /keymanager/testkeys
TransportListener.ssl.keystore.type = jceks
TransportListener.ssl.port = 443
TransportListener.ssl.protocols = SSL_TLS
TransportListener.ssl.truststore.name = /keymanager/testkeys
TransportListener.ssl.truststore.type = jceks
TransportListener.tcp.port = 3801
```

這是一個範例 EKM 內容檔，所有金鑰儲存庫項目都指向不同的金鑰儲存庫。粗體項目有別於上面第一個範例內容檔。

```
Admin.ssl.keystore.name = /keymanager/adminkeys.jceks
Admin.ssl.keystore.type = jceks
Admin.ssl.truststore.name = /keymanager/admintrustkeys
Admin.ssl.truststore.type = jceks
Audit.event.outcome = success,failure
Audit.event.types = all
Audit.eventQueue.max = 0
Audit.handler.file.directory = /keymanager/audit
Audit.handler.file.name = kms_audit.log
Audit.handler.file.size = 10000
Audit.metadata.file.name = /keymanager/metafile.xml
config.drivetable.file.url = FILE:///keymanager/drivetable
config.keystore.file = /keymanager/drive.keys
config.keystore.provider = IBMJCE
config.keystore.type = jceks
fips = Off
TransportListener.ssl.ciphersuites = JSSE_ALL
TransportListener.ssl.clientauthentication = 0
TransportListener.ssl.keystore.name = /keymanager/sslkeys
TransportListener.ssl.keystore.type = jceks
TransportListener.ssl.port = 443
TransportListener.ssl.protocols = SSL_TLS
TransportListener.ssl.truststore.name = /keymanager/ssltrustkeys
TransportListener.ssl.truststore.type = jceks
TransportListener.tcp.port = 3801
```

附錄 B. Encryption Key Manager 配置內容檔

Encryption Key Manager 需要兩個配置內容檔：Encryption Key Manager 伺服器一個，CLI 用戶端一個。這兩個檔案都以 Java.util.Properties 負載檔形式進行處理及剖析，並對內容格式與規格施予特定限制：

- 配置內容是以每行一個的方式記錄。給定內容的值會延伸至行末。
- 包含空格的內容值，如密碼等，不需要括在引號中。
- 金鑰儲存庫密碼的長度不得超過 127 個字元。
- 行末偶有的空格可解譯爲內容值的一部份。

您可以從下列網址的 EKMServicesandSamples 檔下載範例配置內容檔：**http://support.dell.com**。

Encryption Key Manager 伺服器配置內容檔

下列各項構成了 Encryption Key Manager 伺服器配置檔 (KeyManagerConfig.properties) 的完整內容集。檔案中內容設定的順序無關緊要。註解可出現在檔案中。如果要新增註解，請在行首第一直欄使用 『#』。

註: KeyManagerConfig.properties 檔所進行的任何變更，有可能在關機時遺失。因此，在編輯配置內容之前，請確定 Encryption Key Manager 伺服器不在執行中。如果要停止 Encryption Key Manager 伺服器，請從 CLI 用戶端發出 **stopekm** 指令。變更會在 Encryption Key Manager 伺服器重新啓動之時生效。

Admin.ssl.ciphersuites = *value*

指定 Encryption Key Manager 伺服器之間的通訊所用的密碼組合。密碼組合說明資料傳送所用的密碼演算法及信號交換通訊協定 Transport Layer Security (TLS) 和 Secure Socket Layer (SSL)。

必要 選用。

值 可能的值是 IBMJSSE2 所支援的任何密碼組合。

預設值 JSSE_ALL

Admin.ssl.keystore.name = *value*

這是 Secure Socket Layer 用戶端作業（如 Encryption Key Manager 伺服器之間的 **sync** 指令）所用的金鑰組和憑證的資料庫名稱。在同步作業中，Secure Socket 用戶端提供給 Secure Socket 伺服器的憑證是來自這個金鑰儲存庫。

必要 選用。只用於 **sync** 指令。預設值是 **config.keystore.file** 內容值。

Admin.ssl.keystore.password = password

用來存取 Admin.ssl.keystore.name 的密碼

必要 選用。如果未提供，在 Encryption Key Manager 啓動之時，會提示您輸入。指定之時，這個內容值會成爲亂碼，以提高安全程度，內容檔的這個段落名稱本身會取代成名稱爲 'Admin.ssl.keystore.password.obfuscated' 的新段落。

Admin.ssl.keystore.type = *value*
所用的金鑰儲存庫類型。

必要 選用。

預設值 jceks

Admin.ssl.protocols = *value*
安全通訊協定。

必要 選用。

值 SSL_TLS | SSL | TLS

預設值 SSL_TLS

Admin.ssl.timeout = *value*
指定 Socket 等待 read() 多久之後，便擲出 SocketTimeoutException。

必要 選用。

值 以分鐘爲單位指定。0 表示無逾時值

預設值 1

Admin.ssl.truststore.name = *value*
這是用來檢查伺服器提供給 Secure Sockets 用戶端之 Secure Sockets Server 憑證信任的資料庫檔案名稱。

必要 選用。只用於 **sync** 指令。預設值是 **config.keystore.file** 內容值。

Admin.ssl.truststore.type = *value*
所用的金鑰儲存庫類型。

必要 選用。

預設值 jceks

Audit.event.outcome = *value*
只記錄導致指定輸出的審核事件

必要 是。

值 success | failure。兩者都可以指定，以逗點或分號區隔。

預設值 success

Audit.event.Queue.max = 0
在沖寫到檔案之前，審核記憶體佇列中的事件物件數目上限。

必要 選用。建議採用。

值 0 - ? （0 表示立即沖寫。）

預設值 0

Audit.event.types = *value*
只記錄導致指定輸出的審核事件

必要 是。

值 all | authentication | authorization | data synchronization | runtime

| audit management | authorization terminate | configuration management | resource management | none。可以指定多個值，以逗點或分號區隔。

預設值 all

Audit.handler.file.directory = ../audit

用來放置 Audit.handler.file.name 的目錄

必要 選用。建議採用。

Audit.handler.file.multithreads = *value*

指定審核處理常式是否應該分派個別執行緒來處理審核記錄。

必要 選用。

值 true | false

預設值 true

Audit.handler.file.name = kms_audit.log

將記載審核項目的檔名。

必要 是。

Audit.handler.file.size = 100

Audit.Handler.file.name 在開始改寫之前，將成長的大小

必要 選用。建議採用。

值 0 - ?（以 KB 爲單位指定。）

預設值 100

Audit.handler.file.threadlifespan = *value*

限制審核記錄處理執行緒的生命期限。只在 audit.handler.file.multithreads= true 之時才有用。

必要 選用。

值 以毫秒爲單位指定。

預設值 10000

Audit.metadata.file.cachecount = 100

指定在寫入 meta 資料檔之前，儲存在記憶體內的記錄數目。

必要 否

預設值 100

Audit.metadata.file.name = *value*

指定用來儲存 meta 資料記錄的 XML 檔名。

必要 是。

Audit.metadata.file.size = 1024

指定在關閉檔案，啓動新檔案之前，XML meta 資料檔所可能達到的檔案大小上限（以 KB 爲單位指定）。只會儲存檔案的現行版本和舊版。

必要 否

預設值 1024

config.drivetable.file.url = FILE:../filedrive.table
含有序號、憑證等磁帶機相關資訊的檔案。

必要 是。

config.keygroup.xml.file = *value*
指定金鑰群組用來儲存個別別名的 XML 檔名稱。

必要 選用。

config.keystore.file = *value*
指定要用的金鑰儲存庫。

必要 是。

config.keystore.password = password
用來存取 config.keystore.file 的密碼指定之時，這個內容值會成爲亂碼，以提高安全程度，內容檔的這個段落名稱本身會取代成名稱爲 'config.keystore.password.obfuscated' 的新段落。

必要 選用。如果未提供，在 Encryption Key Manager 啓動之時，會提示您輸入。

config.keystore.provider = IBMJCE

必要 選用。

config.keystore.type = jceks

必要 選用。建議採用。

預設值 jceks

debug = *value*
啓用所指定 Encryption Key Manager 元件的除錯。

必要 選用。

值 all | audit | server | drivetable | config | admin | transport | logic | keystore | console | none。可以有多個值，以逗點區隔。

預設值 無

debug.output = *value*
將除錯輸出遞送到指定的位置。

必要 選用。

值 simple_file | console（不建議）。

debug.output.file = debug
除錯輸出寫入其中的路徑和檔名。

必要 選用。當 debug.output = simple_file 時，便是必要。檔案路徑必須存在。

drive.acceptUnknownDrives = *value*
自動將連接 Encryption Key Manager 的新磁帶機新增到磁帶機表格中

必要 是。

值 true | false

預設值 false

安全附註 - 這個設定與有效的 drive.default.alias1 設定一起使用，便可以新增連接 Encryption Key Manager 的磁帶機，且不需要管理者驗證這項新增就能夠運作。請參閱第 3 章的「自動更新磁帶機表格」，以取得詳細資訊。

fips = *value*

聯邦資訊存取安全標準 (FIPS)。請參閱第 2 章的「聯邦資訊存取安全標準 (FIPS) 140-2 注意事項」，以取得詳細資訊。

必要 選用。

值 on | off

預設值 off

maximum.threads = 200

Encryption Key Manager 所能建立的執行緒數目上限。

必要 選用。

Server.authMechanism = *value*

指定用來搭配本端/遠端用戶端的鑑別機制。當這個值設爲 EKM 時，CLI 用戶端使用者必須利用 usr/passwd 作爲 EKMAdmin/changeME 來登入伺服器。（chgpasswd 指令可以變更這個密碼。）當這個值指定爲 LocalOS 時，會對照本端作業系統登錄來進行用戶端鑑別。（請務必在變更 KeyManagerConfig.properties 檔之前，關閉 Encryption Key Manager 伺服器。）CLI 用戶端使用者必須利用 OS usr/passwd 來登入伺服器。如果是 Linux 平台的本端 OS 型鑑別，便需要執行其他步驟：

1. 從 **http://support.dell.com** 下載 Dell 版本 R175158 (EKMServicesAndSamples)，並將檔案解壓縮到您選擇的目錄中。
2. 將 EKMServiceAndSamples.jar 內容（可在 Dell 產品媒體及 **http://support.dell.com** 中取得） 解壓縮到暫存目錄中
3. 從適用於您的平台到 *java_home*/jre/bin。
   - 在 32 位元的 Intel Linux 環境中，將 LocalOS-setup/linux_ia32/libjaasauth.so 檔複製到 *java_home*/jre/bin/ 目錄，對於執行 1.4.2 JVM 的 32 位元 Intel Linux Kernel 而言，其中的 *java_home* 通常是 *java_install_path*/IBMJava2-i386-142。
   - 在 64 位元的 AMD64 Linux 環境中，將 LocalOS-setup/linux-x86_64/libjaasauth.so 檔複製到 *java_home*/jre/bin/ 目錄，對於執行 1.4.2 JVM 的 64 位元 AMD Linux Kernel 而言，其中的 *java_home* 通常是 *java_install_path*/IBMJava2-amd64-142。

如果是 Windows 平台，便不需要這個檔案。

安裝完成之後，您可以啓動 Encryption Key Manager 伺服器。現在，Encryption Key Manager 用戶端可以利用 OS 型使用者/密碼來登入。請注意，只有能夠登入伺服器以及向伺服器提交指令的使用者 ID，可用來執行伺服器，該使用者 ID 亦具備超級使用者/root 權限。

Dell 產品媒體及 **http://support.dell.com** 中的 Readme 檔提供了更詳細的安裝資料。

必要 選用。

值 EKM | LocalOS

預設值 EKM

Server.password = *value*

內部內容。不編輯。

symmetricKeySet = {GroupID | keyAliasList [, keyAliasList,]}

指定要於用 LTO 4 及 LTO 5 磁帶機的對稱金鑰別名及金鑰群組。

必要 選用。僅適用於 LTO 4 及 LTO 5 磁帶匣。

值

請指定一個值給 *GroupID*，指定一或多個值給 *keyAliasList*。

GroupID 指定一個金鑰群組名稱，以便準備對稱金鑰清單，以及作爲未指定磁帶機別名時的預設值。 *GroupID* 必須符合 KeyGroup.xml 檔中現有的金鑰群組 ID。否則，便會傳回 KeyManageException。如果指定了多個 *GroupID*，也會傳回 KeyManagerException。在指定有效的 *GroupID* 之後，每當從 KeyGroups.xml 呼叫 getKey 來取得對稱金鑰清單時，都會追蹤「金鑰群組 XML」中上一個使用的金鑰，並隨機選取下一個要使用的金鑰。每項 *keyAliasList* 規格都會包含 *keyAlias* 或 *keyAliasRange* 的值。

keyAlias 指定金鑰儲存庫中的對稱金鑰名稱或別名的 Backus-Naur Form (BNF)，最多 12 個字元長，也可以是正好 21 個字元的 sequentialKeyID。

keyAliasRange 指定 sequentialKeyID 和十六進位數，最多 18 個字元，用連字號 (-) 分開。如果指定 18 個字元，前兩個字元必須是 00。必須在單行指定，不能含有 cr-lf。

GroupID 指定別名群組的名稱。

範例 `symmetricKeySet = KMA0238ab34,KMB0000034acd2345678a,THZ001-FF` 此指示在 Encryption Key Manager 提供金鑰給 LTO 4 及 LTO 5 磁帶機時，要使用 KMA0238ab34、KMB0000034acd2345678a 及介於 THZ000000000000000001 至 THZ00000000000000000FF 範圍之間的別名。這些金鑰必須在內容檔中 **config.keystore.file** 所指定的金鑰儲存庫中。

sync.action = *value*

指定在自動同步化期間，應如何處理資料。

必要 選用。

值 rewrite | merge

預設值 merge

註: 合併配置資訊與重新編寫配置資訊相同。

sync.ipaddress = *ip_addr:ssl*

指定要自動同步化之遠端 Encryption Key Manager 的 IP 位址和埠。

必要 選用。如果未指定這個內容，或指定不正確，便會停用同步功能。

值 遠端伺服器的 IP 位址：SSL 埠號

sync.timeinhours = *value*

指定等待多少小時之後，便與遠端 Encryption Key Manager 自動同步化。

必要 選用。

值 以小時爲單位指定。

預設值 24

sync.type = *value*

指定要自動同步化的資料。

必要 選用。

值 config | drivetab | all

預設值 drivetab

TransportListener.ssl.ciphersuites = JSSE_ALL

Encryption Key Manager 伺服器之間的通訊所用的密碼組合。密碼組合說明資料傳送所用的密碼演算法及信號交換通訊協定 Transport Layer Security (TLS) 和 Secure Socket Layer (SSL)。

必要 選用。

值 值 - IBMJSSE2 所支援的任何密碼組合。

TransportListener.ssl.clientauthentication = 0

Encryption Key Manager 伺服器之間的通訊所需要的 SSL 鑑別。

必要 選用。

值 0 - 無用戶端鑑別（預設值）
1 - 伺服器想要執行用戶端的用戶端鑑別
2 - 伺服器必須執行用戶端的用戶端鑑別

TransportListener.ssl.keystore.name = *value*

Encryption Key Manager 伺服器用來保留「Secure Socket 伺服器」憑證和私密金鑰的資料庫名稱。這個憑證是提供給 Secure Socket 用戶端進行鑑別及信任檢查。 「Encryption Key Manager 用戶端」也利用這個金鑰儲存庫與「Encryption Key Manager 伺服器」交談，以及扮演 Secure Socket 用戶端的角色。

必要 是。

TransportListener.ssl.keystore.password = password

用來存取 TransportListener.ssl.keystore.name 的密碼。若指定此密碼，則這個內容的值在其他安全方面會變得模糊，且內容檔中的段落名稱，也會被名爲 'TransportListener.ssl.keystore.password.obfuscated' 的新段落所取代。

必要 選用。

TransportListener.ssl.keystore.type = jceks

必要 選用。建議採用。

值 JCEKS

TransportListener.ssl.port = *value*

Encryption Key Manager 伺服器將用來接聽其他 Encryption Key Manager 伺服器或 Encryption Key Manager CLI 用戶端所發出之要求的埠。

必要 是。

值 埠號，如 443。這必須符合 CLI 用戶端配置內容檔中的 TransportListener.ssl.port 內容。

TransportListener.ssl.protocols = SSL_TLS

安全通訊協定

必要 選用。

值 SSL_TLS（預設值）| SSL | TLS

TransportListener.ssl.timeout = 10

指定 Socket 等待 read() 多久之後，便擲出 SocketTimeoutException。

必要 選用。

值 以分鐘爲單位指定。

預設值 1

TransportListener.ssl.truststore.name = *value*

用來驗證其他用戶端與伺服器身分之公開金鑰和已簽章的憑證之資料庫名稱。如果 TransportListener.ssl.clientauthentication 內容**未**設爲預設值 0（無用戶端鑑別），扮演「Secure Socket 伺服器」的「Encryption Key Manager 伺服器」便必須利用這個檔案來鑑別用戶端。「Encryption Key Manager 用戶端」也利用這個信任儲存庫與「Encryption Key Manager 伺服器」交談，以及扮演 Secure Socket 用戶端的角色。

必要 是。

TransportListener.ssl.truststore.type = jceks

必要 選用。建議採用。

值 JCEKS

TransportListener.tcp.port = *value*

Encryption Key Manager 伺服器將用來接聽磁帶機發出之要求的埠。預設 TCP 埠號是 3801。

必要 是。

值 埠號，如 10。

TransportListener.tcp.timeout = *value*

指定 Socket 等待 read() 多久之後，便擲出 SocketTimeoutException。

必要 選用。

值 以分鐘爲單位指定。0 表示無逾時值。

預設值 10

CLI 用戶端配置內容檔

這個檔案 (ClientKeyManagerConfig.properties) 含有 KeyManagerConfig.properties 檔所包含之內容的子集。這個子集包括下列內容。

TransportListener.ssl.ciphersuites = JSSE_ALL

Encryption Key Manager 伺服器和 CLI 用戶端之間的通訊所用的密碼組合。密

碼組合說明資料傳送所用的密碼演算法及信號交換通訊協定 Transport Layer Security (TLS) 和 Secure Socket Layer (SSL)。

必要 選用。

值 這個值必須符合 Encryption Key Manager 伺服器內容檔 KeyManagerConfig.properties 中指定給 TransportListener.ssl.ciphersuites 的值。

TransportListener.ssl.host = *value*

使 Encryption Key Manager CLI 用戶端識別 Encryption Key Manager 伺服器。

必要 選用。

值 IP 位址或主機名稱

預設值 localhost

範例
```
TransportListener.ssl.host = 9.24.136.444
TransportListener.ssl.host = ekmsvr02
```

註: 在 KeyManagerConfig.properties 檔中，不用這個內容。

TransportListener.ssl.keystore.name = *value*

Encryption Key Manager 用戶端也利用這個金鑰儲存庫與 Encryption Key Manager 伺服器交談，以及扮演 Secure Socket 用戶端的角色。

必要 是。

TransportListener.ssl.keystore.type = jceks

金鑰儲存庫的類型。

必要 選用。建議採用。

預設值 jceks

TransportListener.ssl.port = *value*

這是 CLI 用戶端用來與 Encryption Key Manager 伺服器通訊的埠。

必要 是。

值 這個值必須符合 Encryption Key Manager 伺服器內容檔 KeyManagerConfig.properties 中指定給 TransportListener.ssl.port 的值。

TransportListener.ssl.protocols = SSL_TLS

安全通訊協定

必要 選用。

值 這個值必須符合 Encryption Key Manager 伺服器內容檔 KeyManagerConfig.properties 中指定給 TransportListener.ssl.protocols 的值。

TransportListener.ssl.truststore.name = *value*

用來驗證其他用戶端與伺服器身分之公開金鑰和已簽章的憑證之資料庫名稱。

必要 是。

TransportListener.ssl.truststore.type = jceks

信任儲存庫的類型。

必要	選用。建議採用。
預設值	jceks

您可以從 **http://support.dell.com** 的 EKMServicesAndSamples 檔下載範例配置內容檔。

附錄 C. 常見問題

可以使用應用程式型金鑰管理與磁帶庫管理之加密的某種組合嗎？

不可以。 當使用應用程式管理的加密時，在磁帶庫層次上，加密是透通的。同樣地，當使用磁帶庫管理的加密時，此程序在其他層次也是透通的。各種加密管理方法是互斥的。對於磁帶庫管理的加密，應用程式不需要進行任何改變。

在可能產生磁帶加密或解密要求的每個系統上，Encryption Key Manager 是否需要安裝好且在執行中？

當使用磁帶庫管理的加密時，產生磁帶機寫入要求的系統不一定是執行 Encryption Key Manager 的系統。不但如此，在每個存取加密磁帶機的系統上，也不一定要有執行中的 Encryption Key Manager 實例。

如果我併入 "drive.acceptUnknownDrives = True" 參數，我是否仍應在配置檔中併入 "config.drivetable.file.url = FILE:/filename" 參數？

必須一律指定 **config.drivetable.file.url**。它是磁帶機資訊的所在的位置。如果您設定 `drive.acceptUnknownDrives = True`，您也應該將 `drive.default.alias1` 和 `drive.default.alias2` 變數指定爲正確的憑證別名/金鑰標籤。

FILE:/filename 是 `config.drivetable.file.url` 內容的正確語法嗎？FILE:///filename 出現在範例檔，FILE:../ 出現在說明中。

範例正確。這是一個 URL 規格，只是不符合人們對於目錄結構規格的一般預期。

在執行於 Windows 之 Encryption Key Manager 實例的 KeyManagerConfig.properties 檔中指定完整路徑時，我必須使用正斜線或反斜線？

由於 KeyManagerConfig.properties 是一個 Java 內容檔，因此，在路徑名稱中只能辨識正斜線，即使在 Windows 中，也是如此。如果您在 KeyManagerConfig.properties 檔中使用反斜線，便會發生錯誤。

Encryption Key Manager 會執行任何「憑證撤消清冊 (CRL)」檢查嗎？

不，Encryption Key Manager 不會執行任何 CRL 檢查

當用來加密磁帶的憑證到期時，會發生什麼情況？Encryption Key Manager 會讀取先前加密的磁帶嗎？

對 Encryption Key Manager 而言，憑證是否過期，並不重要。它會繼續接受這些憑證，以及讀取先前加密的磁帶。不過，到期的憑證必須保留在金鑰儲存庫，才能讀取或附加先前加密的磁帶。

Encryption Key Manager 需要在更新時重新命名憑證嗎？

依預設，Encryption Key Manager 會配置成接受含有過期憑證的新金鑰要求。以這個方式來配置 Encryption Key Manager 時，就不需要更新憑證。如果停用這個功能，且這個私密金鑰/憑證配對仍必須用於新的金鑰要求，使用者便必須更新憑證。只會更新憑證（有效日期），不會更新相關聯的金鑰。

較新的 Encryption Key Manager 版本仍會讀取舊版軟體所建立的加密磁帶嗎？

是。不管版本爲何，Encryption Key Manager 都會接受憑證。

注意事項

商標

本文所用的商標：Dell、Dell 標誌和 PowerVault，皆爲 Dell Inc. 的商標。Microsoft 和 Windows 則是 Microsoft Corporation 的註冊商標。本文件中所使用的其他商標或商品名稱可能是擁有該標誌及名稱的實體或其產品名稱。Dell Inc. 未擁有非其自身所有之商標及商品名稱的所有權。

名詞解釋

本名詞解釋定義本出版品及其他相關出版品中所用的特殊詞彙、縮寫及字首語。

四劃

公開金鑰 (public key). 這是非對稱金鑰組之中的一個金鑰，通常用來加密。 Encryption Key Manager 會先利用公開金鑰來封裝（保護）AES 資料金鑰，再將它們儲存到磁帶匣。

五劃

加密 (encryption). 這是指將資料轉換成密碼。資料的加密和解密需要金鑰。加密提供保護來防止不具備金鑰的人或軟體試圖存取資料。

七劃

別名 (alias). 請參閱「金鑰標籤 (key label)」。

私密金鑰 (private key). 這是非對稱金鑰組之中的一個金鑰，通常用來解密。在解密之前，Encryption Key Manager 利用私密金鑰來解開受保護的 AES 資料金鑰。

八劃

金鑰標籤 (key label). 這是用來比對 EEDK 與解開受保護的對稱資料金鑰時所需要之私密金鑰 (KEK) 的專屬 ID。也稱爲別名或憑證標籤，這會隨著所用的金鑰儲存庫而不同。

金鑰儲存庫 (keystore). 這是私密金鑰及用來鑑別對應公開金鑰之相關 X.509 數位憑證鏈的資料庫。在某些環境中，也稱爲憑證儲存庫或金鑰環。

金鑰環 (key ring). 請參閱「金鑰儲存庫 (keystore)」。

九劃

重設金鑰 (rekey). 這是用來保護已加密磁帶所儲存的資料金鑰 (DK) 之非對稱「金鑰加密金鑰 (KEK)」的變更程序，讓不同的實體也能夠存取資料。

十六劃

憑證 (certificate). 這是一種數位文件，用來將公開金鑰連結到憑證擁有人的身分，使憑證擁有人能夠接受鑑別。

憑證標籤 (certificate label). 請參閱「金鑰標籤 (key label)」。

憑證儲存庫 (certificate store). 請參閱「金鑰儲存庫 (keystore)」。

A

AES. 進階加密標準 (Advanced Encryption Standard) 的字首語。這是美國政府採用爲加密標準的區塊密碼。

D

DK. 資料金鑰 (Data Key) 的字首語。這是用來加密資料的英數字串。

E

EEDK. 外部加密資料金鑰 (Externally Encrypted Data Key) 的字首語。這是指「金鑰加密金鑰」已加密（封裝）而尚未儲存到資料卡匣的資料金鑰。請參閱 KEK。

K

KEK. 金鑰加密金鑰 (Key Encrypting Key) 的字首語。這是用來加密資料金鑰的英數非對稱金鑰。請參閱 EEDK。

P

PKDS. 公開金鑰資料集 (Public Key Data Set) 的字首語。也是「PKA 加密金鑰資料集 (PKA cryptographic Key Data Set)」。

R

RSA. Rivest-Shamir-Adleman 演算法的字首語。這是用來進行加密和鑑別的非對稱公開金鑰密碼系統。 Ron Rivest、Adi Shamir 和 Leonard Adleman 在 1977 年發明這個系統。系統安全會隨著兩大質數乘積的分解難度而不同。

索引

索引順序以中文字，英文字，及特殊符號之次序排列。

〔十六劃〕

〔十七劃〕

〔十九劃〕

〔二十三劃〕

A

C

D

E

F

J

K

L

M

S

W

X