AIX 7.2 版

裝置管理

請注意

使用本資訊及其支援的產品之前，請先閱讀第 195 頁的『注意事項』中的資訊。

此版本適用於 AIX 7.1 版，以及所有後續版次和修改內容，除非新版中另有指示。

目錄

關於本文件

本文件包含使用者和系統管理者所需的完整資訊，當您執行諸如備份及還原系統、管理實體及邏輯儲存體、調整適當的分頁空間大小等作業時，需要使用這些資訊來選取適當的選項。它提供如何執行管理邏輯磁區、儲存體及資源等作業的完整資訊。系統使用者可以學習如何執行諸如 執行指令、處理程序、處理檔案及目錄，以及基本列印等作業。

其他有助於使用者及系統管理者的主題包括：建立及調整分頁空間大小、管理虛擬記憶體、備份及還原系統、管理硬體及虛擬裝置、使用「系統資源控制器 (SRC)」、保護檔案安全、使用儲存媒體、自訂環境檔案，以及撰寫 Shell Script。您也可以在作業系統隨附的文件 CD 中找到本文件。

強調顯示

下列為本文件所使用的強調顯示慣例：

粗體	指定指令、子常式、關鍵字、檔案、結構、目錄及系統已預先定義其名稱的其他項目。亦指定圖形物件，如使用者選取的按鈕、標籤及圖示。
斜體	指定由使用者提供實際名稱或值的參數。
等寬字體	指定特定資料值的範例、類似您可能看到顯示文字的範例、類似您以程式設計者身分可能寫出程式碼部分的範例、系統的訊息，或您應實際鍵入的資訊。

AIX 中區分大小寫

AIX® 作業系統中的所有項目皆區分大小寫，亦即區分大寫和小寫字母。例如，您可以使用 **ls** 指令來列出檔案。如果您鍵入 LS，則系統會回應找不到該指令。同樣地，**FILEA**、**FiLea** 及 **filea** 即使位於相同目錄中，仍是三個不同的檔名。為了避免執行不想要的動作，請務必使用正確的大小寫。

ISO 9000

本產品的開發和製造過程使用 ISO 9000 註冊的品質系統。

裝置管理

您可以使用指令來管理可在 AIX 中使用的不同裝置。您可以管理的部分裝置包括「邏輯磁區管理程式」、檔案系統、磁帶機及印表機。

新增功能

閱讀「作業系統」和裝置管理主題集合的新增或重大變更相關資訊。

如何查看新增功能或變更資訊

在 PDF 檔案中，您可能會在新增及變更資訊周遭看見修訂標記（>| 及 |<）。

2018 年 2 月

下列資訊為對此主題集合所進行更新項目的摘要：

· 已在第 138 頁的『設定 iSCSI 卸載配接卡』主題中，更新設定 iSCSI 卸載配接卡的說明。

2017 年 12 月

下列資訊為對此主題集合所進行更新項目的摘要：

· 已更新第 174 頁的『儲存體資料快取的限制』主題中關於儲存體資料快取的限制。

2017 年 10 月

下列資訊為對此主題集合所進行更新項目的摘要：

· 已新增第 31 頁的『邏輯磁區儲存體的空間收回支援』主題。
· 已更新 第 171 頁的『USB 裝置支援』主題中 AIX 作業系統支援的 USB 裝置相關資訊。

2017 年 6 月

下列資訊為對此主題集合所進行更新項目的摘要：

· 已在第 179 頁的『監視快取統計資料』主題中新增統計資料監視的相關資訊。
· 已在第 144 頁的『管理具備 MPIO 功能的裝置』主題中新增 **lsmpio** 指令的相關資訊。

邏輯磁區管理程式

可讓您建立及控制邏輯磁區儲存體的作業系統指令、程式庫子常式及其他工具的集合稱為「邏輯磁區管理程式 (LVM)」。

LVM 透過對映更為簡單且有彈性的儲存體空間邏輯 視圖與實際實體 磁碟之間的資料，來控制磁碟資源。LVM 透過使用一層裝置驅動程式程式碼來執行此動作，該程式碼在傳統裝置驅動程式之上執行。

LVM 由邏輯磁區裝置驅動程式 (LVDD) 與 LVM 子常式介面程式庫組成。邏輯磁區裝置驅動程式 (LVDD) 是虛擬裝置驅動程式，會管理並處理所有的 I/O。它會將邏輯位址轉換為實體位址，並將 I/O 要求傳送到特定的裝置驅動程式。*LVM* 子常式介面程式庫 包含系統管理指令使用的常式，以執行系統邏輯及實體磁區的系統管理作業。

邏輯磁區管理程式概念

在您可以開始使用「邏輯磁區管理程式」之前，您必須瞭解基本機制及術語。

轉開處理程序
轉開處理程序是 LVM 用來確定磁區群組已經可以使用，並且包含最新資料的其中一種機制。

varyonvg 及 **varyoffvg** 指令可啟動或停止（使可用或不可用）您已為系統定義的磁區群組。必須先轉開磁區群組，系統才可以存取之。於轉開處理程序期間， LVM 會從定義於磁區群組的實體磁區讀取管理資料。此管理資料包含磁區群組描述子區域 (VGDA) 及磁區群組狀態區域 (VGSA)，儲存於磁區群組的每一個實體磁區上。

VGDA 包含說明磁區群組中每一個邏輯磁區的實體分割區與邏輯分割區對映的資訊，以及包括時間戳記的其他重要資訊。VGSA 包含的資訊則是諸如嘗試對磁區群組進行轉開作業時，哪些實體分割區是過時的，哪些實體磁區已遺失（即不可用或非作用中）的資訊。

如果轉開作業無法存取定義於磁區群組中的一或多個實體磁區，則該指令會顯示為該磁區群組定義的所有實體磁區名稱及其狀態。 這有助於您決定是否要轉斷此磁區群組。

Quorum
Quorum 是 LVM 用來確定磁區群組已準備好可以使用，並且包含最新資料的其中一種機制。

Quorum 是選定的作用中「磁區群組描述子區域 (VGDA)」及「磁區群組狀態區域 (VGSA)」數目。Quorum 可確保在磁碟故障時 VGDA/VGSA 區域的資料完整性。磁區群組中的每一個實體磁碟都至少有一個 VGDA/VGSA。當在單一磁碟上建立磁區群組時，它最初有兩個 VGDA/VGSA 區域位於磁碟上。如果磁區群組是由兩個磁碟所組成，一個磁碟仍有兩個 VGDA/VGSA 區域，但是另一個磁碟僅有一個 VGDA/VGSA。當磁區群組由三個或更多個磁碟組成時，每一個磁碟都僅配置一個 VGDA/VGSA。

當 LVM 無法讀取至少一半的磁碟（代表它們的 VGDA/VGSA 區域）時，Quorum 會遺失。在具有兩個磁碟的磁區群組中，如果僅具有一個 VGDA/VGSA 的磁碟遺失，則 Quorum 仍會存在，因為仍可抵達三個 VGDA/VGSA 區域中的兩個。如果具有兩個 VGDA/VGSA 區域的磁碟遺失，則上述說法不再成立。組成磁區群組的磁碟越多，一個磁碟故障時遺失 Quorum 的機會越小。

當 Quorum 遺失時，磁區群組會自行轉斷，因此 LVM 將無法再存取磁碟。這會防止對該磁區群組進行進一步的磁碟 I/O，以便在發生實體問題時，不會遺失資料或假冒寫入資料。此外，由於轉斷的關係，會以錯誤日誌通知使用者：發生了硬體錯誤，且必須執行服務程式。

有時，即使在 Quorum 遺失時，也要繼續操作磁區群組。在這些狀況中，可以停用磁區群組的 Quorum 檢查。 此類磁區群組稱為非 *Quorum* 磁區群組。發生非 Quorum 磁區群組的最常見情況是已鏡映邏輯磁區。當磁碟遺失時，如果邏輯磁區的副本位於可以存取的未停用磁碟上，則資料不會遺失。不過，當資料（包括副本）位於已不可用的一或多個磁碟上時，在非 Quorum 磁區群組（鏡映或非鏡映）中可能會有案例。在這些案例中，即使磁區群組繼續轉開，也可能無法存取資料。

鏡映儲存區
鏡映儲存區可能將磁區群組的實體磁區分成個別儲存區。

鏡映儲存區由一或多個實體磁區組成。每一個實體磁區一次只能屬於一個鏡映儲存區。建立邏輯磁區時，正建立之邏輯磁區的每一個副本都可以指派給鏡映儲存區。指派給鏡映儲存區的邏輯磁區副本僅從該鏡映儲存區中的實體磁區配置分割區。這能夠限制邏輯磁區副本可以使用的磁碟。沒有鏡映儲存區，限制在建立或擴充邏輯磁區時用於配置的實體磁區的唯一方法是使用對映檔。因此，使用鏡映儲存區會大大簡化此處理程序。鏡映儲存區可以使用 **extendvg** 指令或 **chpv** 指令建立。

當建立新的鏡映儲存區時，您必須指定鏡映儲存區名稱。鏡映儲存區名稱必須符合下列規則：

· 只能包含英數字元、_（底線）、-（減號）或 .（句號）字元。

· 必須小於或等於 15 個字元。

· 在磁區群組中必須是唯一的。

鏡映儲存區用於磁區群組之後，磁區群組不能再匯入不支援鏡映儲存區的 AIX 版本。這包括 AIX 6.1.1.0 版之前的任何版本。此外，為了搭配使用鏡映儲存區與加強的並行模式 LVM，叢集中的所有節點都必須支援鏡映儲存區。

鏡映儲存區嚴格程度

鏡映儲存區嚴格程度可以用於對鏡映儲存區使用施行更嚴格的限制。鏡映儲存區嚴格程度可以具有下列三個值的其中一個：

off
當鏡映儲存區嚴格程度設為 off 時，對鏡映儲存區使用不設定任何限制。這是預設值。

on

當鏡映儲存區嚴格程度設為 on 時，磁區群組中建立的每一個邏輯磁區都必須指派給鏡映儲存區。

super

當鏡映儲存區嚴格程度設為 super 時，會套用下列限制：

· 本端及遠端實體磁區不能屬於相同的鏡映儲存區。

註： 如需本端及遠端實體磁區的相關資訊，請參閱 HACMP/XD GLVM 文件。

· 磁區群組中最多可以有三個鏡映儲存區。

· 每一個鏡映儲存區必須至少包含磁區群組中每一個邏輯磁區的一個副本。

地理邏輯磁區管理程式

「地理邏輯磁區管理程式 (GLVM)」可讓您在地理位置較遠的地方維護資料的鏡映副本。

GLVM 可以協助保護商業資料遠離災難，方法是將重要資料鏡映至遠端災難回復站台。如果災難（如火災或水災）損毀了正式作業站台上的資料，則您可能在災難回復站台上有資料的備份副本。

資料是透過標準 TCP/IP 網路來進行鏡映。正式作業及災難回復站台不需要位於相同的實體網路上。允許兩個站台之間的路由器及閘道。不使用過長的磁碟纜線，而是將 TCP/IP 網路及「遠端實體磁區 (RPV)」裝置驅動程式用於遠端磁碟存取。

使用者會配置地理位置較遠的磁碟作為遠端實體磁區，然後將那些遠端實體磁區與邏輯實體磁區結合，以形成地理鏡映的磁區群組。這些磁區群組是受到「邏輯磁區管理程式 (LVM)」管理，且其運作方式與一般磁區群組類似。GLVM 同時支援同步及非同步遠端鏡映。

非 Quorum 磁區群組

當磁區群組缺少「磁區群組描述子區域 VGDA」或「磁區群組狀態區 VGSA」的 Quorum 時，「邏輯磁區管理程式 (LVM)」會自動停止該磁區群組。不過，只要有一對 VGDA/VGSA 是完整的，您就可以選擇讓群組處於線上的選項。這個選項會產生非 *Quorum* 磁區群組。

LVM 需要存取非 Quorum 磁區群組中的所有磁碟，才能允許重新啟動。這樣可確保 VGDA 及 VGSA 保持最新。

您可以在每一個邏輯磁區都有至少兩個副本的系統中，產生非 Quorum 邏輯磁區。 若發生磁碟故障，則只要有一個作用中磁碟，磁區群組即會保持作用中。

註： 使用者定義以及 **rootvg** 磁區群組皆可在非 Quorum 狀態下操作， 但是用來將使用者定義的磁區群組及 **rootvg** 磁區群組配置成非 Quorum 的方法， 與在硬體故障後用於回復的方法卻不同。請確定您有對適當的磁區群組使用正確的方法。

即使當您正在使用非 Quorum 磁區群組，也有可能會失去 Quorum，並在 **errpt** 指令輸出中看到下列訊息：

```
QUORUM LOST,  VOLUME GROUP CLOSING LVM.
```

當所有實體磁區都處於 missing 狀態時，即會發出這則訊息，並且 LVM 會自動轉斷磁區群組。

訊息指出 QUORUM LOST，因為磁區群組上停用的 Quorum 將 Quorum 需求減為 1。您可以使用 **lsvg *vgname*** 指令，顯示 QUORUM: 欄位中的 Quorum 值。在所有實體磁區都是 missing 的情況下， 即使違反這個最低 Quorum 需求，也會產生失去 Quorum 的訊息，並自動轉斷磁區群組。

將磁區群組轉換為非 Quorum 狀態

您可以將磁區群組變更為非 Quorum 狀態，以便即使沒有 Quorum，仍可以繼續使用資料。

此程序經常用於具有下列配置的系統：

· 發生邏輯磁區鏡映的雙磁碟磁區群組

· 發生一或兩次邏輯磁區鏡映的三磁碟磁區群組

當這些情況下的磁區群組可以在非 Quorum 狀態下操作時，即使發生磁碟故障，只要磁區群組中至少一個磁碟處於作用中，磁區群組仍會保持作用中。

若要能夠回復非 Quorum 群組，請確定下列項目：

- 如果系統使用 JFS 或 JFS2 檔案系統，請鏡映 JFS 日誌邏輯磁區。
- 將鏡映的副本存放在個別的磁碟中。如果您不確定配置，請鍵入下列指令以檢查每一個邏輯分割區的實體位置（PV1、PV2 及 PV3）。（若要將副本存放在個別的磁碟中，則 PV1、PV2 及 PV3 直欄必須含有不同的 hdisk 號碼。）

  ```
  lslv -m LVName
  ```

  如果邏輯磁區的僅有副本是位在無法使用的同一磁碟上，則不論其磁區群組是處於 Quorum 或非 Quorum 狀態，使用者都不能使用該磁區。

使用者定義的磁區群組及 rootvg 磁區群組都可以在非 Quorum 狀態下操作，但其配置及回復方法有所不同。

若要啟動非 Quorum 的使用者定義磁區群組，則必須能夠存取磁區群組的所有實體磁區，否則啟動會失敗。因為在最後一個磁碟變更不可存取前，非 Quorum 磁區群組會持續連線，所以在啟動時，必須能使每一個磁碟成為可存取的。

注意： 遺漏與 rootvg 磁區群組相關的磁碟時，請勿開啟系統電源，除非是遺漏的磁碟無法修復時。「邏輯磁區管理程式 (LVM)」一律使用 -f 旗標，以強制啟動（轉開）非 Quorum rootvg；此作業含有風險。必須強制啟動 LVM，因為除非已啟動 rootvg，否則作業系統無法啟動。換句話說，即使僅單一磁碟是可存取的，LVM 仍會進行最後的嘗試來啟動（轉開）非 Quorum rootvg。

相關概念

配接卡或電源供應器故障時的高可用性
若要防止配接卡或電源供應器發生故障，請根據您的需求來執行下列其中一項或多項動作。

配置邏輯磁區管理程式

「邏輯磁區管理程式 (LVM)」會與基本作業系統一起安裝，不需要進一步配置。不過，在 LVM 可以使用磁碟之前，必須先配置它們並將它們定義為實體磁區。

相關工作

定義應用程式的原始邏輯磁區
原始邏輯磁區是實體與邏輯磁碟空間的區域，此區域受應用程式（如資料庫或分割區）的直接控制，而非直接受制於作業系統或檔案系統。

LVM 維護指令及捷徑

將維護 LVM 控制之實體（實體與邏輯磁區、磁區群組及檔案系統）時，可能需要的最簡單作業在下表中加以分組。

表 *1.* 管理邏輯磁區及儲存體作業

作業	*SMIT* 捷徑	指令或檔案
啟動磁區群組	**smit varyonvg**	
將不含資料的硬式磁碟新增至現有磁區群組	**smit extendvg**	
將不含資料的硬式磁碟新增至新磁區群組	**smit mkvg**	
新增邏輯磁區[附註 1]	**smit mklv**	
新增磁區群組	**smit mkvg**	
新增並啟動新的磁區群組	**smit mkvg**	
變更邏輯磁區以使用資料配置	**smit chlv1**	

表 *1.* 管理邏輯磁區及儲存體作業 *(*繼續*)*		
作業	***SMIT* 捷徑**	**指令或檔案**
變更磁區群組的名稱[附註 2]	1. **smit varyoffvg** 2. **smit exportvg** 3. **smit importvg** 4. **smit mountfs**	1. **varyoffvg** *OldVGName* 2. **exportvg** *OldVGName* 3. **importvg** *NewVGName* 4. **mount all**
變更磁區群組以使用自動啟動	**smit chvg**	
變更或設定邏輯磁區原則	**smit chlv1**	
複製邏輯磁區至新的邏輯磁區[附註 3]	**smit cplv**	
複製邏輯磁區至相同大小的現有邏輯磁區**警告 1**	**smit cplv**	
複製邏輯磁區至較小的現有邏輯磁區**警告 1** 附註 3	不要使用 SMIT **警告 2**	1. 建立邏輯磁區。例如：**mklv -y hdisk***N* **vg004** 2. 在新邏輯磁區上建立新的檔案系統。例如：**crfs -v jfs -d hdisk***N* **-m /doc -A yes** 3. 裝載檔案系統。例如：**mount /doc** 4. 在新的裝載點建立目錄。例如：**mkdir /doc/options** 5. 將檔案系統從來源轉送至目標邏輯磁區。例如：**cp -R /usr/adam/oldoptions/* \ /doc/options**
複製邏輯磁區至較大的現有邏輯磁區**警告 1**	**smit cplv**	
取消啟動磁區群組	**smit varyoffvg**	
啟用寫入驗證及變更排程法原則	**smit chlv1**	
增加邏輯磁區的大小上限	**smit chlv1**	
增加邏輯磁區的大小	**smit extendlv**	
按磁區群組列出所有邏輯磁區	**smit lslv2**	
列出系統中的所有實體磁區	**smit lspv2**	
列出所有磁區群組	**smit lsvg2**	
列出實體磁區的狀態、邏輯磁區或分割區	**smit lspv**	
列出磁區群組的內容	**smit lsvg1**	
列出邏輯磁區的狀態或對映	**smit lslv**	
鏡映含有或不含資料配置的邏輯磁區	**smit mklvcopy**	
關閉抽取式磁碟電源	**smit offdsk**	僅熱抽取特性可以使用

表 1. 管理邏輯磁區及儲存體作業 (繼續)		
作業	***SMIT* 捷徑**	**指令或檔案**
開啟抽取式磁碟電源	**smit ondsk**	僅熱抽取特性可以使用
從磁區群組移除鏡映	**smit unmirrorvg**	
移除磁區群組	**smit reducevg2**	
重組磁區群組	**smit reorgvg**	
取消配置磁碟並關閉磁碟電源	**smit rmvdsk1** 或 **smit rmvdsk**，然後 **smit opendoor**	

小心：

1. 使用此程序來複製到現有的邏輯磁區將會改寫該磁區上的資料，而不要求使用者確認。
2. 請勿使用 SMIT 程序或 **cplv** 指令，將較大的邏輯磁區複製到較小的磁區。這麼做會造成檔案系統損毀，因為部分資料（包括超級區塊）並未複製到較小的邏輯磁區。

註：

1. 在您建立邏輯磁區之後，將會關閉此狀態，這是因為沒有任何 LVM 結構正在使用該邏輯磁區。它將持續關閉，直到已在邏輯磁區上裝載檔案系統，或開啟邏輯磁區執行原始 I/O。
2. 您不能變更 **rootvg** 的名稱，也不能匯入或匯出它。
3. 您必須具有足夠的直接存取記憶體，以複製特定的邏輯磁區。

在系統仍可用時新增磁碟

下列程序說明如何使用熱抽取特性（可讓您無需關閉系統電源即可新增磁碟）來開啟及配置磁碟。

您可以新增磁碟來額外增加儲存體，或是修正磁碟故障。此特性只適用於某些系統上。

1. 在機箱的可用介面槽中安裝磁碟。如需安裝程序的詳細資訊，請參閱機器的服務手冊。
2. 藉由在指令行上鍵入下列捷徑，來開啟新磁碟的電源：

```
smit ondsk
```

此時，可將磁碟新增至系統但它尚不可用。接下來的動作要視新磁碟是否包含資料而定。

· 如果磁碟沒有任何資料，請使用下列其中一項將它當作實體磁區新增至磁區群組：

– 若要將磁碟新增至現有的磁區群組，請在指令行上鍵入下列捷徑：

```
smit extendvg
```

– 若要將磁碟新增至新的磁區群組，請在指令行上鍵入下列捷徑：

```
smit mkvg
```

· 如果磁碟包含資料，則匯入資料。

變更邏輯磁區的名稱

下列程序說明如何重新命名邏輯磁區，而不會遺失邏輯磁區上的資料。

在下列範例中，邏輯磁區名稱將從 `lv00` 變更為 `lv33`。

1. 卸載與邏輯磁區相關的所有檔案系統，請鍵入：

```
unmount /FSname
```

其中 *FSname* 是檔案系統的完整名稱。

註：

a. 如果您嘗試卸載的檔案系統目前在使用中，則 **unmount** 指令會失敗。僅當沒有開啟檔案系統的任何檔案且該裝置上沒有使用者的現行目錄時，才會執行 **unmount** 指令。

b. **unmount** 指令的另一個名稱是 **umount**。這兩個名稱是可交換的。

2. 重新命名邏輯磁區，請鍵入：

```
chlv -n NewLVname OldLVname
```

其中的 **-n** 旗標指定新邏輯磁區名稱 (*NewLVname*)， *OldLVname* 是您要變更的名稱。例如：

```
chlv -n lv33 lv00
```

註： 如果重新命名 JFS 或 JFS2 日誌，則系統會提示您在使用重新命名之日誌裝置的所有檔案系統上執行 **chfs** 指令。

3. 重新裝載於步驟第 6 頁的『1』 中卸載的檔案系統，請鍵入：

```
mount /test1
```

此時，邏輯磁區會重新命名並可供使用。

複製邏輯磁區至另一個實體磁區

視您的需要，有數個方式可以在保持檔案系統完整性的同時，複製邏輯磁區至另一個實體磁區。

有多種方法，可將邏輯磁區或 JFS 複製至另一個實體磁區。請選擇最適合您的目的之方法。

複製邏輯磁區

最簡單的方法是使用 **cplv** 指令來複製原始邏輯磁區，並在目的地實體磁區上建立新的邏輯磁區。

1. 停止使用邏輯磁區。卸載檔案系統（如適用的話）並停止存取邏輯磁區的任何應用程式。
2. 選取其容量可包含原始邏輯磁區中所有資料的實體磁區。

小心： 如果您從包含資料的較大邏輯磁區複製到較小的邏輯磁區，則可能會毀損檔案系統，這是因為部分資料（包括超級區塊）可能會遺失。

3. 複製原始邏輯磁區（在此範例中，稱為 **lv00**）並建立一個新的，請使用下列指令：

註： 如果建立新的邏輯磁區且磁區群組以並行模式轉開，則下列 **cplv** 指令會失敗。

```
cplv lv00
```

4. 裝載檔案系統（如適用），並重新啟動應用程式以開始使用邏輯磁區。

此時，可使用邏輯磁區副本。

在原始邏輯磁區仍可用時複製邏輯磁區

如果您的環境需要持續使用原始邏輯磁區，則可以使用 **splitlvcopy** 指令複製內容，如下列範例所示。

1. 使用下列 SMIT 捷徑，鏡映邏輯磁區：

```
smit mklvcopy
```

2. 停止使用邏輯磁區。卸載檔案系統（如適用）並停止存取邏輯磁區的任何應用程式，或將其置於靜止模式。

小心： 下一步驟使用 **splitlvcopy** 指令。分割邏輯磁區前一律先關閉它們，並在使用此指令前先卸載任何包含的檔案系統。如果多個處理程序同時存取開啟的邏輯磁區，則分割該邏輯磁區會毀損檔案系統，並使原始邏輯磁區與副本之間失去一致性。

3. 以 root 權限，複製原始邏輯磁區 (oldlv) 至新的邏輯磁區 (newlv)，請使用下列指令：

```
splitlvcopy -y newlv oldlv
```

-y 旗標可指定新的邏輯磁區名稱。如果 oldlv 磁區沒有邏輯磁區控制區塊，則 **splitlvcopy** 指令會順利完成，但會產生一則訊息，告知 newlv 磁區已經建立但沒有邏輯磁區控制區塊。

4. 裝載檔案系統（如適用），並重新啟動應用程式以開始使用邏輯磁區。

此時，可使用邏輯磁區副本。

複製原始邏輯磁區至另一個實體磁區

若要複製原始邏輯磁區至另一個實體磁區，請執行下列步驟：

1. 使用下列指令，在磁區群組中的新實體磁區上建立邏輯磁區的鏡映副本：

```
mklvcopy LogVol_name 2 new_PhysVol_name
```

2. 使用下列指令同步化新鏡映副本中的分割區：

```
syncvg -l LogVol_name
```

3. 使用下列指令將邏輯磁區副本從實體磁區中移除：

```
rmlvcopy LogVol_name 1 old_PhysVol_name
```

此時，可使用原始邏輯磁區副本。

在使用者定義的磁區群組的專用磁碟上建立檔案系統日誌

JFS 或 JFS2 檔案系統日誌是檔案系統交易記錄的一種格式化清單。日誌可確定檔案系統的完整性（但不一定能確定資料的完整性），以防系統在交易完成前當機。

安裝系統上，會在 rootvg 的 hd8 上建立專用磁碟。下列程序幫助您在其他磁區群組的個別磁碟上建立 JFS 日誌。建立 JFS2 日誌時，程序需要下列變更：

- 日誌裝置類型是 jfs2log。
- **logform** 指令需要 -V jfs2 選項來指定 JFS2 日誌裝置。
- **crfs** 指令必須指定 jfs2，而非 jfs。

註： JFS2 日誌不一定要位於與檔案系統不同的磁碟上。唯一需求是，日誌裝置必須在與檔案系統相同的磁區群組上。在此程序中，JFS2 日誌必須在個別磁碟上，才能改進效能。

在某些狀況下，為使用者定義的磁區群組建立檔案系統日誌檔可以增進效能，例如，若您有一個 NFS 伺服器，且想要處理此伺服器的交易而不與其他處理程序競爭。

您可以使用下列程序，建立具有兩個實體磁區（hdisk1 及 hdisk2）的磁區群組 (fsvg1)。檔案系統位在 hdisk2（裝載在 /u/myfs 上的 256 MB 檔案系統）上，而日誌位在 hdisk1 上。 根據預設值，JFS 日誌大小是 4 MB。您可以將較少使用的程式（例如，/blv）放在與日誌相同的實體磁區上，而不會影響效能。

若要使用 SMIT 及指令行介面，為使用者定義的磁區群組建立 JFS 日誌，請遵循下列步驟：

1. 使用 SMIT 捷徑新增磁區群組（在此範例中，為 fsvg1）：

```
smit mkvg
```

2. 使用 SMIT 捷徑將新的邏輯磁區新增至此磁區群組：

```
smit mklv
```

3. 在**新增邏輯磁區**畫面上，將您的資料新增至下列欄位。例如：

```
邏輯磁區名稱                          fsvg1log
邏輯分割區數目                        1
實體磁區名稱                          hdisk1
邏輯磁區類型                          jfslog
實體磁區上的位置                      center
```

4. 在您設定欄位後，按 Enter 鍵接受變更並結束 SMIT。
5. 在指令行上鍵入下列指令：

```
/usr/sbin/logform /dev/fsvg1log
```

6. 接收到下列提示時，鍵入 **y** 然後按 Enter 鍵：

```
Destroy /dev/fsvg1log
```

不論此提示中的字眼為何，都不會有任何損毀。對此提示回應 y 時，系統會為 JFS 日誌格式化邏輯磁區，以便它可以記錄檔案系統交易。

7. 使用下列 SMIT 捷徑新增另一個邏輯磁區：

```
smit mklv
```

8. 鍵入相同的磁區群組名稱，如在步驟 2 所使用的名稱（在此範例中為 `fsvg1`）。在「邏輯磁區」畫面上，將您的資料新增至下列欄位。請記得為此邏輯磁區指定一個與步驟 3 中不同的實體磁區。例如：

```
邏輯磁區名稱                          fslv1
邏輯分割區數目                        64
實體磁區名稱                          hdisk2
邏輯磁區類型                          jfs
```

在您設定欄位後，按 Enter 鍵接受變更並結束 SMIT。

9. 使用下列指令順序，將檔案系統新增至新的邏輯磁區、指定日誌並裝載新的檔案系統：

```
crfs -v jfs -d LogVolName -m FileSysName -a logname=FSLogPath
mount FileSysName
```

其中的 *LogVolName* 是您在步驟 2 建立的邏輯磁區名稱；*FileSysName* 是您要在此邏輯磁區上裝載的檔案系統名稱；而 *FSLogPath* 是步驟 2 建立的邏輯磁區名稱。例如：

```
crfs -v jfs -d fslv1 -m /u/myfs -a logname=/dev/fsvg1log
mount /u/myfs
```

10. 若要驗證您是否已正確地設定檔案系統與日誌，請鍵入下列指令（替換您的磁區群組名稱）：

```
lsvg -l fsvg1
```

此輸出會顯示您建立的兩個邏輯磁區及其檔案系統類型，如下列範例所示：

```
LV 名稱             類型    ...
/dev/fsvg1log       jfslog  ...
fslv1               jfs     ...
```

您已建立一個磁區群組，它在個別實體磁區上包含至少兩個邏輯磁區，而且其中一個邏輯磁區包含檔案系統日誌。

註：為了提供備援，您可以在邏輯磁區層次上提供 JFS2 日誌裝置的鏡映。不過，提供鏡映並不是一般作法，而且也沒有必要。

匯入或匯出磁區群組

下列表格說明如何使用匯入及匯出，來將使用者定義的磁區群組從一個系統移至另一個系統。(rootvg 磁區群組無法匯出或匯入。)

匯出程序會從系統中移除磁區群組的定義。匯入程序用來將磁區群組引入其新系統。

您亦可使用匯入程序在磁區群組曾經與系統相關聯但已匯出時，將該磁區群組重新引入系統。您亦可使用匯入及匯出，藉由將要新增的磁碟放入它自己的磁區群組，來將包含資料的實體磁區新增至磁區群組。

小心：如果新系統上已存在與匯入邏輯磁區同名的邏輯磁區，則 **importvg** 指令會變更匯入邏輯磁區的名稱。如果 **importvg** 指令必須重新命名邏輯磁區，則它會將錯誤訊息列印到標準錯誤。如果沒有發生衝突，則 **importvg** 指令亦會在 /etc/filesystems 檔案中建立檔案裝載點及項目。

匯入及匯出磁區群組作業		
作業	*SMIT* 捷徑	指令或檔案
匯入磁區群組	**smit importvg**	
匯出磁區群組	1. 卸載磁區群組中邏輯磁區的檔案系統：**smit umntdsk** 2. 轉斷磁區群組：**smit varyoffvg** 3. 匯出磁區群組：**smit exportvg**	

小心：具有分頁空間磁區的磁區群組，不能在分頁空間作用中時匯出。匯出具有作用中分頁空間的磁區群組之前，請鍵入下列指令，以確定分頁空間不會在系統起始設定時自動啟動：

```
chps -a n paging_space name
```

然後，重新啟動系統，使分頁空間成為非作用中。

移轉實體磁區的內容

若要將屬於一或多個指定邏輯磁區的實體分割區，從某一實體磁區移至磁區群組中的一或多個其他實體磁區，請使用下列指示。您亦可以使用此程序，在置換或修復故障的磁碟前，先搬移此故障磁碟中的資料。此程序可以用於 root 磁區群組或使用者定義之磁區群組中的實體磁區。

小心：從實體磁區中移轉開機邏輯磁區時，必須清除來源中的開機記錄，否則會使系統懸置。執行 **bosboot** 指令時，也必須執行下列程序之步驟 4 中說明的 **chpv -c** 指令。

1. 如果您要將資料移轉至新的磁碟，請執行下列步驟。否則，請繼續步驟 2。
   a. 檢查系統可辨識且可用的磁碟，請鍵入：

   ```
   lsdev -Cc disk
   ```

   輸出如下：

   ```
   hdisk0 Available 10-60-00-8,0  16 Bit  LVD  SCSI Disk Drive
   hdisk1 Available 10-60-00-9,0  16 Bit  LVD  SCSI Disk Drive
   hdisk2 Available 10-60-00-11,0 16 Bit  LVD  SCSI Disk Drive
   ```

   b. 如果磁碟有列出來，且是處於可用狀態，則檢查它是否不屬於另一個磁區群組，請鍵入：

   ```
   lspv
   ```

   輸出如下：

   ```
   hdisk0          0004234583aa7879       rootvg          active
   hdisk1          00042345e05603c1       none            active
   hdisk2          00083772caa7896e       imagesvg        active
   ```

   在此範例中， hdisk1 可用作目的地磁碟，這是因為第三個欄位顯示磁區群組沒有在使用它。

   如果新磁碟未列出或不可用，則您需要配置磁碟或邏輯磁區儲存體。

   c. 將新磁碟新增至磁區群組，請鍵入：

   ```
   extendvg VGName diskname
   ```

其中 *VGName* 是磁區群組名稱，*diskname* 是新磁碟名稱。在前一步驟中顯示的範例中，*diskname* 會由 hdisk1 取代。

2. 來源及目的地實體磁區必須都在同一磁區群組中。若要判斷這兩個實體磁區是否都在該磁區群組中，請鍵入：

```
lsvg -p VGname
```

其中 *VGname* 是磁區群組名稱。root 磁區群組的輸出如下：

```
rootvg:
PV_NAME        PV STATE       TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk0         active         542         85          00..00..00..26..59
hdisk1         active         542         306         00..00..00..00..06
```

請記下 FREE PPs 的數目。

3. 檢查您在目標磁碟上是否有足夠的空間，可以容納您想要移動的來源：
   a. 鍵入下列指令以判斷來源磁碟上的實體分割區數：

```
lspv SourceDiskName | grep "USED PPs"
```

其中 *SourceDiskName* 是來源磁碟名稱，例如 hdisk0。輸出如下：

```
USED PPs:      159 (636 MB)
```

在本例中，您在目的地磁碟上需要 159 FREE PPs，才能順利完成移轉。

   b. 將來源磁碟的 USED PP 數目與目的地磁碟上的 FREE PP 數目相比較（步驟 2）。如果 FREE PP 的數目大於 USED PP 的數目，則您有足夠的空間進行移轉。

4. 僅當您要從 rootvg 磁區群組中的磁碟移轉資料時，才需要遵循此步驟。
   如果您要從使用者定義之磁區群組中的磁碟移轉資料，請繼續步驟 5。

   檢查開機邏輯磁區 (**hd5**) 是否在來源磁碟上，請鍵入：

```
lspv -l SourceDiskNumber | grep hd5
```

如果您未收到任何輸出，則表示開機邏輯磁區不在來源磁碟上。請繼續步驟 5。

如果您收到類似下面的輸出：

```
hd5            2   2   02..00..00..00..00   /blv
```

然後執行下列指令：

```
migratepv -l hd5 SourceDiskName DestinationDiskName
```

您會接收到一則訊息，警告您要在目的地磁碟上執行 **bosboot** 指令。您亦必須執行 **mkboot -c** 指令以清除來源上的開機記錄。鍵入下列指令順序：

```
bosboot -a -d /dev/DestinationDiskName

bootlist -m normal DestinationDiskName
mkboot -c -d /dev/SourceDiskName
```

5. 鍵入下列 SMIT 捷徑以移轉資料：

```
smit migratepv
```

6. 列出實體磁區，然後選取您之前檢查的來源實體磁區。
7. 前往**目的地**實體磁區欄位。如果您接受預設值，則磁區群組中的所有實體磁區均可用於轉送。否則，請選取一或多個具有足夠空間的磁碟，來容納您要移動的分割區（從步驟 4）。
8. 如果您想要的話，請前往「只移動屬於此**邏輯磁區**的資料」欄位，然後列出並選取邏輯磁區。只移動配置到指定邏輯磁區的實體分割區，這些分割區是位在選取為來源實體磁區的實體磁區上。
9. 請按 Enter 鍵，移動實體分割區。

此時，資料目前位於新的（目的地）磁碟。不過，原始（來源）磁碟仍在磁區群組中。如果磁碟仍是可靠的，則您可繼續使用它作為緊急備用磁碟。建議執行下列步驟，特別是在磁碟發生故障時：

1. 若要從磁區群組移除來源磁碟，請鍵入：

```
reducevg VGNname SourceDiskName
```

2. 若要從系統實際移除來源磁碟，請鍵入：

```
rmdev -l SourceDiskName -d
```

配置磁碟

您可以利用各種方法來配置新的磁碟。

您可以下列任何一種方式來配置新的磁碟。

- 如果可以關機並關閉系統電源，請使用「方法 1」。將實體磁碟連接到系統時，如果可能的話，應當關機並關閉系統電源。
- 如果您無法關閉系統且瞭解新磁碟的相關明細（如子類別、類型、母項節點名稱及其連接的位置），則使用第 2 種方法。
- 如果您無法關閉系統，並且只知道磁碟的位置，請使用第 3 種方法。

配置磁碟後，雖然它通常都可供使用，但是「邏輯磁區管理程式」要求將它進一步識別為實體磁區。

第 1 種方法

如果在連接磁碟之前能夠關機並關閉系統電源，請使用下列方法：

1. 實際將新磁碟連接到系統，然後根據系統隨附的文件來開啟磁碟及系統的電源。
2. 系統開機期間，讓「配置管理程式」(**cfgmgr**) 自動配置磁碟。
3. 系統開機之後，以 root 權限在指令行上鍵入 **lspv** 指令以尋找新磁碟的名稱。

   系統傳回類似下列其中個項目：

```
   hdisk1  none                      none
```

   或：

```
   hdisk1   00005264d21adb2e     none
```

   第一個欄位可識別磁碟的系統指派名稱。第二個欄位顯示實體磁區 ID (PVID)（如果有的話）。如果新磁碟沒有出現在 **lspv** 輸出中，請參照 安裝與移轉。

此時磁碟可以由系統來使用，但它需要 PVID，以供 LVM 使用。如果新磁碟沒有 PVID，則請參閱第 13 頁的『使可用磁碟成為實體磁區』。

第 2 種方法

當您無法關閉系統並瞭解關於新磁碟的下列資訊時，請使用下列方法：

- 磁碟的連接方式（子類別）
- 磁碟類型（類型）
- 磁碟連接哪個系統附件（母項節點名稱）
- 磁碟的邏輯位址（連接的位置）。

請執行下列動作：

1. 實際將新磁碟連接到系統，然後根據系統隨附的文件來開啟磁碟及系統的電源。
2. 若要配置磁碟並確定它可以當成實體磁區使用，請搭配使用 **mkdev** 指令與所顯示的旗標，如以下範例所示：

```
mkdev -c disk -s scsi -t 2200mb -p scsi3 \
-w 6,0 -a pv=yes
```

此範例將新增 2.2 GB 磁碟，其 SCSI ID 是 6 且 scsi3 SCSI 匯流排的邏輯單元號碼是 0。**-c** 旗標可定義裝置類別。**-s** 旗標可定義子類別。**-t** 旗標可定義裝置類型。**-p** 旗標可定義您要指派的母項裝置名稱。**-w** 旗標可依 SCSI ID 及邏輯單元號碼指定磁碟位置。**-a** 旗標可指定裝置屬性值配對 pv=yes，它會將磁碟變成實體磁區，並使用唯一實體磁區 ID 將開機記錄寫入磁碟（如果它還沒有記錄的話）。

此時，將磁碟既定義為可用的裝置，又定義為實體磁區。您可以在指令行上鍵入 **lspv** 指令以列出新的磁碟項目。如果新磁碟沒有出現在 **lspv** 輸出中，請參照 安裝與移轉。

第 3 種方法

當您無法關閉系統且只知道磁碟位置時，請使用下列方法：

1. 實際將新磁碟連接到系統，然後根據系統隨附的文件來開啟磁碟及系統的電源。
2. 若要檢查系統上已經配置了哪些實體磁碟，請在指令行上鍵入 **lspv** 指令。如需 **lspv** 指令的相關資訊，請參閱 lspv 指令 主題。輸出如下：

   ```
   hdisk0       000005265ac63976    rootvg
   ```

3. 在指令行上鍵入 **cfgmgr** 以進入「配置管理程式」。「配置管理程式」會自動偵測及配置系統上所有新連接的裝置，包括新磁碟。如需 **cfgmgr** 指令的相關資訊，請參閱 **cfgmgr**。
4. 若要確認已配置了新磁碟，請重新鍵入 **lspv** 指令。輸出將類似下列其中一項：

   ```
   hdisk1    none                none
   ```

   或

   ```
   hdisk1   00005264d21adb2e     none
   ```

   第一個欄位可識別磁碟的系統指派名稱。第二個欄位顯示實體磁區 ID (PVID)（如果有的話）。如果新磁碟沒有出現在 **lspv** 輸出中，請參照 安裝與移轉。

此時磁碟可以由系統來使用，但它需要 PVID，以供 LVM 使用。如果新磁碟沒有 PVID，則請參閱第 13 頁的『使可用磁碟成為實體磁區』。

使可用磁碟成為實體磁區

將磁碟指派到磁區群組並可供 LVM 使用之前，必須先將它配置為實體磁區。

使用下列指示來配置實體磁區：

1. 請確定磁碟是作業系統已知，且可以使用，而且目前未被作業系統或任何應用程式使用。在指令行上鍵入 **lspv** 指令。輸出如下：

   ```
      hdisk1  none                    none
   ```

   針對下列項目檢查輸出：

   - 如果新磁碟的名稱沒有出現在指令輸出中，請參照第 12 頁的『配置磁碟』。
   - 如果輸出的第二個欄位顯示了系統產生的實體磁區 ID (PVID)（例如，00005264d21adb2e），則磁碟已經配置為實體磁區，您無需完成此程序。
   - 如果輸出的第三個欄位顯示了磁區群組名稱（例如，rootvg），則該磁碟目前正被使用中，對於此程序來說，它不是一個適當的選擇。

   如果新的磁碟沒有 PVID 且未在使用中，則繼續下一步。
2. 若要將可用的磁碟變更為實體磁區，請在指令行上鍵入 **chdev** 指令。例如：

   ```
   chdev -l hdisk3 -a pv=yes
   ```

   -l 旗標可指定磁碟的裝置名稱。**-a** 旗標可指定裝置屬性值配對 pv=yes，它會將磁碟變成實體磁區，並使用唯一實體磁區 ID 將開機記錄寫入磁碟（如果它還沒有記錄的話）。

此時，會將磁碟定義為實體磁區。您可以在指令行上鍵入 **lspv** 指令以列出新的磁碟項目。

變更 rootvg 的 PVID 和 VGID

您可以在系統開機階段期間，變更 rootvg 磁區群組的實體磁區 ID (PVID) 和磁區群組 ID (VGID)。

若要變更 rootvg 的 PVID 和 VGID，請使用值 2 來設定 *sys0 dev ghostdev* 屬性，然後重新啟動系統。*sys0 device ghostdev* 屬性是位元旗標。

- 若要設定 *sys0 device ghostdev* 屬性來變更 rootvg 磁區群組的 PVID 和 VGID，請輸入下列指令：

```
chdev -l sys0 -a ghostdev=2
```

註： 在 **ipl_varyon** 指令變更 rootvg 中的所有磁碟的 PVID 和 VGID 之後，會取消設定值為 2 的 *sys0 device ghostdev* 屬性值。如果用來變更所有 rootvg 磁碟的 PVID 的 **chdev** 指令失敗，則 **ipl_varyon** 指令會傳送一則警告訊息，並繼續轉接 rootvg。如果用來變更 rootvg 中的所有磁碟的 PVID 的 **chdev** 指令失敗，而且您想要在下一次重新開機期間變更 PVID 和 VGID，請再次將 *sys0 device ghostdev* 屬性設為 2。

- 若要列出 *ghostdev* 屬性的值，請輸入下列指令：

```
lsattr -E -l sys0 -a ghostdev
```

取代鏡映磁區群組中故障的實體磁區

下列程序會取代鏡映磁區群組內故障的實體磁區 (PV)。**replacepv** 指令提供了方法來取代大部分配置中故障的 PV。對於無法使用 **replacepv** 指令的配置，也會提供替代程序。

此作法範例情節中的資訊已使用特定版本的 AIX 進行測試。依據您的 AIX 版本及層次，所得到的結果可能大不相同。

簡介

使用故障 PV 的所有邏輯磁區在其他可用 PV 上具有有效的副本（專用的傾出邏輯磁區可能除外）。

使用 replacepv 指令取代故障 PV

如果無法符合下列任一個必要條件，請參閱替代程序。

- 包含故障 PV 的磁區群組不是 rootvg。
- 取代的 PV 可新增至含有故障 PV 的磁區群組（這不一定行得通，需視 PV 大小及磁區群組性質而定，如每個 PV 的最大 PP 數目）。
- 取代的 PV 必須能夠與故障的 PV 同時配置到系統中。
- 取代的 PV 名稱可以不同於故障 PV 的名稱。
- 取代的 PV 大小至少必須是故障的 PV 大小。
- 包含故障 PV 的磁區群組不可以是 Snapshot 磁區群組或具有 Snapshot 磁區群組。

請完成下列步驟，並假設故障 PV 為 hdisk2，而取代的 PV 為 hdisk10：

1. 如果取代的 PV 尚未安裝在系統上，請執行必要步驟來安裝它。若要使用配置管理程式來定義新的 PV，請執行下列指令：

   ```
   cfgmgr
   ```

   使用 **lspv** 指令來決定指派給 PV 的名稱。在這個範例中，假設新的 PV 名為 hdisk10。
2. 若要以步驟 1 定義的 PV 來取代故障的 PV，請執行下列指令：

   ```
   replacepv hdisk2 hdisk10
   ```

   當指令執行時，hdisk2 會取代為 hdisk10，並且 hdisk2 不再指定給磁區群組。
3. 若要取消定義故障的 PV，請執行下列指令：

   ```
   rmdev -dl hdisk2
   ```
4. 從系統實際移除故障磁碟。

5. 完成下列步驟來驗證程序是否順利完成：

   · 若要檢查所有邏輯磁區是否已鏡映到所要的新建 PV，請執行下列指令：

   ```
   lslv lvname
   ```

   檢查受到故障 PV 影響的每一個邏輯磁區的 COPIES 屬性，以確定所要的副本份數已經存在。如果邏輯磁區的副本份數低於所需的數目，請使用 **mklvcopy** 指令，建立額外的副本。

   · 若要驗證所有邏輯磁區是否已同步，而且沒有陳舊的分割區，請執行下列指令：

   ```
   lspv hdisk10
   ```

   檢查取代的 PV 之 STALE PARTITIONS 屬性，以確定計數為零。如果有陳舊分割區，請使用 **syncvg** 指令來同步化分割區。

步驟 5 完成了故障 PV 的取代程序。

在配置不允許使用 replacepv 指令時取代故障 PV

假設故障實體磁區 hdisk0 及其鏡映 hdisk1 都是 *yourvg* 磁區群組的一部分。

1. 若要從故障 PV 移除鏡映副本，請執行下列指令：

   ```
   unmirrorvg yourvg hdisk0
   ```

2. 如果 rootvg 上發生 PV 故障，請執行下列指令，從開機清單移除 hdisk0：

   註： 如果您的配置使用 hdisk0 及 hdisk1 以外的開機裝置，請將它們新增到指令。

   ```
   bootlist -om normal hdisk1
   ```

   這個步驟需要 hdisk1 將可開機裝置保留在 rootvg 中。完成此步驟之後，請確定 hdisk0 沒有在輸出中出現。

3. 如果 rootvg 上發生 PV 故障，請從故障 PV 重建任何專用的傾出裝置。

   如果故障 PV 上已有專用的傾出裝置，您可以使用 **mklv** 指令，在現有的 PV 上建立新的邏輯磁區。您可以使用 **sysdumpdev** 指令，將新的邏輯磁區設定為主要傾出裝置。

4. 若要取消定義故障的 PV，請執行下列指令：

   註： 如果故障 PV 是用來啟動系統的 PV，則移除磁碟裝置項目也會移除 /dev/ipldevice 固定鏈結。

   ```
   reducevg yourvg hdisk0
   rmdev -dl hdisk0
   ```

5. 如果故障 PV 是最近使用的開機裝置，請執行下列指令，重建在步驟 4 中移除的 /dev/ipldevice 固定鏈結：

   ```
   ln /dev/rhdisk1 /dev/ipldevice
   ```

   請注意位於 PV 名稱字首的 r。

   若要驗證是否已重建 /dev/ipldevice 固定鏈結，請執行下列指令：

   ```
   ls /dev/ipldevice
   ```

6. 取代故障磁碟。
7. 若要定義新的 PV，請執行下列指令：

   ```
   cfgmgr
   ```

   cfgmgr 指令會指定 PV 名稱給取代的 PV。指定的 PV 名稱可能與先前指定給故障 PV 的 PV 名稱相同。在這個範例中，假設裝置 hdisk0 指定給取代的 PV。

8. 若要新增 PV 到磁區群組，請執行下列指令：

```
extendvg yourvg hdisk0
```

您可能會遇到下列錯誤訊息：

```
0516-050 這個磁區群組中剩餘的描述子空間不足。請嘗試新增較小的 PV 或使用另一個磁區群組。
```

如果遇到這個錯誤，且無法新增 PV 到磁區群組，則您可以嘗試將邏輯磁區鏡映到已存在於磁區群組的另一個 PV，或新增較小的 PV。如果無法使用任一個選項，則您可以使用 **chvg** 指令， 將磁區群組升級到 Big-type 或 Scalable-type 磁區群組，來略過這個限制。

9. 鏡映磁區群組。

註：如果所有下列條件都存在，就不能使用 **mirrorvg** 指令：

· 目標系統是邏輯分割區 (LPAR)。

· 開機邏輯磁區副本（預設為 hd5）位於故障 PV。

· 自上次冷開機後，取代 PV 的配接卡已動態配置到 LPAR。

如果上述所有條件都存在，請使用 **mklvcopy** 指令來重建每個邏輯磁區的鏡映副本，如下所示：

a. 建立開機邏輯磁區的副本，以確保將其配置給一連串的實體分割區。

b. 建立其餘邏輯磁區的副本，然後使用 **syncvg** 指令來同步化這些副本。

c. 藉由關閉並重新啟動 LPAR，而非使用 shutdown 或 reboot 指令來重新開機，使磁碟成為可開機磁碟。此關機不必立即執行，但是需要執行，這樣系統才能從新的 PV 開機。

否則，搭配下列指令使用新 PV，在磁區群組中建立邏輯磁區的新副本：

註：根據預設值，**mirrorvg** 指令會停用 Quorum。若為 rootvg，則要使用 **-m** 選項，以確保新邏輯磁區副本對映到 hdisk0 的方式與工作中磁碟相同。

```
mirrorvg yourvg hdisk0
```

10. 如果您的配置保留了部分邏輯磁區的副本，您可能需要使用下列指令來重建那些副本：

```
mklvcopy -k
```

11. 如果 rootvg 上發生 PV 故障，請執行下列指令來起始設定開機記錄：

```
bosboot -a
```

12. 如果 rootvg 上發生 PV 故障，請執行下列指令來更新開機清單：

註：如果您的配置使用 hdisk0 及 hdisk1 以外的開機裝置，請將它們新增到指令。

```
bootlist -om normal hdisk0 hdisk1
```

13. 驗證程序是否順利完成。

· 若要驗證所有邏輯磁區是否鏡映到新的 PV，請執行下列指令：

```
lslv lvname
```

檢查受到故障 PV 影響的每一個邏輯磁區的 COPIES 屬性，以確定所要的副本份數已經存在。如果邏輯磁區的副本份數低於所需的數目，請使用 **mklvcopy** 指令，建立額外的副本。

· 若要驗證所有邏輯磁區分割區是否已同步化，請執行下列指令來檢查是否沒有陳舊的分割區：

```
lspv hdisk0
```

檢查取代的 PV 之 STALE PARTITIONS 屬性，以確定計數為零。如果有陳舊分割區，請使用 **syncvg** 指令來同步化分割區。

如果 rootvg 上發生 PV 故障，請使用下列步驟來驗證此程序的其他方面：

· 若要驗證開機清單，請執行下列指令：

```
bootlist -om normal
```

· 若要驗證傾出裝置，請執行下列指令：

```
sysdumpdev -l
```

· 若要驗證可開機 PV 的清單，請執行下列指令：

```
ipl_varyon -i
```

· 若要驗證 /dev/ipl_device，請執行下列指令：

```
ls -i /dev/rhdisk1 /dev/ipldevice
```

確保 **ls** 的輸出中，這兩個項目具有相同的 inode 號碼。

這個步驟會完成整個程序。

相關資訊

Logical Volume Manager from A to Z: Introduction and Concepts

遺失實體磁區時通知管理者

在無法存取邏輯磁區時，雖然 AIX 會記載錯誤，但還是可能有些情況下是偵測不到錯誤的。

例如，如果實體磁區是鏡映磁區群組的一部分，則因為使用者仍可存取完整的資料副本，所以不會注意到有問題發生。在這個狀況下， 自動通知就可以在使用者工作發生任何中斷之前，將問題警示給管理者。

下列程序說明如何設定缺少實體磁區時的自動通知。您只要修改下列程序，就可以追蹤您認為重要的其他錯誤。

此作法範例情節中的資訊已使用特定版本的 AIX 進行測試。依據您的 AIX 版本及層次，所得到的結果可能大不相同。

1. 以 root 權限來備份 /etc/objrepos/errnotify ODM 檔案。

   您可以隨意命名此備份。在下列範例中， 備份會將目前日期附加到 errnotify 檔名：

   ```
   cd /etc/objrepos
   cp errnotify errnotify_current_date
   ```

2. 請使用您喜好的編輯器來建立一個名為 /tmp/pvmiss.add 的檔案，以包含下列段落：

   ```
   errnotify:
           en_pid = 0
           en_name = "LVM_SA_PVMISS"
           en_persistenceflg = 1
           en_label = "LVM_SA_PVMISS"
           en_crcid = 0
           en_type = "UNKN"
           en_alertflg = ""
           en_resource = "LVDD"
           en_rtype = "NONE"
           en_rclass = "NONE"
           en_method = "/usr/lib/ras/pvmiss.notify $1 $2 $3 $4 $5 $6 $7 $8 $9"
   ```

   當您完成本主題中的所有步驟之後，錯誤通知常駐程式就會以通知訊息內之錯誤日誌項目的詳細資訊，來自動擴充這個 Script 的 $1 到 $9。

3. 請使用您喜好的編輯器來建立一個名為 /usr/lib/ras/pvmiss.notify 的檔案，使其具有下列內容：

   ```
   #!/bin/ksh
   exec 3>/dev/console
   print -u3 "?"
   print -u3 - "-------------------------------------------"
   print -u3 "ALERT!  ALERT!  ALERT!  ALERT!  ALERT!  ALERT!"
   print -u3 ""
   print -u3 "Desc: PHYSICAL VOLUME IS MISSING. SEE ERRPT."
   print -u3 ""
   ```

```
print -u3 "Error label: $9"
print -u3 "Sequence number: $1"
print -u3 "Error ID: $2"
print -u3 "Error class: $3"
print -u3 "Error type: $4"
print -u3 "Resource name: $6"
print -u3 "Resource type: $7"
print -u3 "Resource class: $8"
print -u3 - "-------------------------------------------"
print -u3 "?"
mail - "PHSYICAL VOLUME DECLARED MISSING" root <<-EOF
-------------------------------------------
ALERT!  ALERT!  ALERT!  ALERT!  ALERT!  ALERT!
Desc: PHYSICAL VOLUME IS MISSING. SEE ERRPT.
Error label: $9
Sequence number: $1
Error ID: $2
Error class: $3
Error type: $4
Resource name: $6
Resource type: $7
Resource class: $8
-------------------------------------------
EOF
```

4. 儲存檔案並結束編輯器。
5. 為您建立的檔案設定適當的許可權。
   例如：

```
chmod 755 /usr/lib/ras/pvmiss.notify
```

6. 鍵入下列指令，將您已在步驟 2 中建立的 LVM_SA_PVMISS 定義新增至 ODM：

```
odmadd /tmp/pvmiss.add
```

現在，只要發生 LVM_SA_PVMISS 錯誤，系統就會執行 /usr/lib/ras/pvmiss.notify Script。此 Script 會將訊息傳送到主控台，也會傳送郵件給 root 使用者。

相關概念

邏輯磁區儲存體

邏輯磁區是位於實體磁區上的資訊群組。

相關資訊

odmadd 指令

從磁區群組中分割鏡映磁碟

Snapshot 支援會協助您保護鏡映磁區群組的一致性，以避免可能發生的磁碟故障。

使用 Snapshot 功能，您可以分割一或多個鏡映磁碟，以其作為磁區群組的可靠（從 LVM meta 資料的立場來看）時間點備份，且在需要時，以可靠的方式將分割的磁碟重新整合到磁區群組中。在下面的程序中，您首先需從磁區群組中分割出鏡映磁碟，再將分割的磁碟合併到原始磁區群組中。為進一步確保 Snapshot 的可靠性，請務必卸載檔案系統，且使用原始邏輯磁區的應用程式必須處於已知狀態（即您需要使用備份時，應用程式可從中回復的狀態）。

如果符合下列任一項，則將無法分割磁區群組：

· 磁碟已經遺失。

· 最後一個非陳舊分割區位於分割的磁區群組上。

· 如果您不搭配使用強制旗標 (**-f**) 與 **splitvg** 指令，磁區群組中會存在陳舊分割區。

此外，在加強或典型並行模式中無法使用 Snapshot 功能（尤其是 **splitvg** 指令）。無法讓分割的磁區群組並行或強制並行，而分割及原始磁區群組所允許的變更也有所限制。如需詳細資料，請閱讀 **chvg** 指令說明。

此作法範例情節中的資訊已使用特定版本的 AIX 進行測試。依據您的 AIX 版本及層次，所得到的結果可能大不相同。

1. 確定磁區群組已完全鏡映，且鏡映在僅包含此鏡映集的磁碟或磁碟集上。

2. 若要啟用 Snapshot 支援，請使用下列指令將原始磁區群組 (origVG) 分割到其他磁碟或磁碟集上：

```
splitvg origVG
```

此時，您便有了原始磁區群組的可靠時間點備份。不過請注意，您不能變更分割磁區群組上的配置。

3. 重新啟動分割磁碟，並使用下列指令將它合併到原始磁區群組中：

```
joinvg origVG
```

此時，分割磁區群組便與原始磁區群組重新整合在一起。

相關概念

邏輯磁區儲存體

邏輯磁區是位於實體磁區上的資訊群組。

相關資訊

chvg 指令

recreatevg 指令

splitvg 指令

Logical Volume Manager from A to Z: Introduction and Concepts

為 LVM 進行疑難排解

有數種常見類型的 LVM 問題，您可以對它們進行疑難排解。

為磁碟機問題進行疑難排解

本資訊會顯示如何診斷並修正硬式磁碟機問題。

如果您懷疑磁碟機有機械方面的故障或已發生故障，請使用下列程序在磁碟上執行診斷程式：

1. 以 root 權限，在指令行上鍵入下列 SMIT 捷徑：

```
smit diag
```

2. 選取**現行 shell 診斷**，以進入 AIX 診斷工具。
3. 在您閱讀「診斷作業指示」畫面之後，請按 Enter 鍵。
4. 選取**偵錯常式**。
5. 選取**系統驗證**。
6. 向下捲動清單，以尋找並選取您要測試的磁碟機。
7. 選取**確認**。

根據診斷結果，您應該可以判斷磁碟的狀況：

- 如果您偵測到磁碟機發生故障或已經故障，則從該磁碟回復資料非常重要。移轉是從故障的磁碟中回復資料的較常用方式。下列程序說明如果無法順利完成移轉，如何在邏輯磁區中回復或還原資料。
- 如果磁碟機發生故障，且您可以修復磁碟機而無需將它重新格式化，則不會遺失任何資料。
- 如果磁碟機必須重新格式化或替換，請製作備份（如果可能的話），然後在替換它之前， 先從其磁區群組與系統配置中移除磁碟機。單一副本檔案系統中的部分資料可能會遺失。

磁碟機空間

如果您用盡磁碟機上的空間，有數種方式可以補救此問題。您可以自動追蹤及移除不想要的檔案、限制使用者存取某些目錄或從另一部磁碟機中裝載空間。

您必須具有 root 使用者、系統群組或管理群組權限，才能執行這些作業。

自動清除檔案系統的指令

使用 **skulker** 指令，移除不想要的檔案以清除檔案系統。

請從指令行鍵入下列指令：

```
skulker -p
```

skulker 指令是用來定期清除檔案系統中已作廢或不需要的檔案。候選的檔案包括 /tmp 目錄中的檔案、比指定的經歷時間還要舊的檔案、a.out 檔案、核心檔案或 ed.hup 檔案。如需 **skulker** 指令的相關資訊，請參閱 **skulker**。

通常，**skulker** 指令是每日執行， 是在離峰時間由 **cron** 指令所執行的統計作業程序的一部分。

限制使用者存取某些目錄
您可以藉由限制存取某些目錄並監視磁碟使用情況，來釋出磁碟空間，以及使其保持可用。

1. 要限制使用者存取某些目錄時，請鍵入：

   ```
   chmod 755 DirName
   ```

   此指令會設定擁有者 (root) 的讀取及寫入許可權，並設定群組及其他人的唯讀許可權。*DirName* 是您要限制的目錄完整路徑名稱。
2. 將下行新增至 **/var/spool/cron/crontabs/adm** 檔案，以監視個別使用者的磁碟使用情況：

   ```
   0 2 * * 4 /usr/sbin/acct/dodisk
   ```

   這一行會在每個星期四 (4) 的上午二點 (0 2) 執行 **dodisk** 指令。**dodisk** 指令會起始磁碟使用情形統計作業。在 **cron** 指令於離峰時段期間執行的統計作業程序過程中，通常會執行此指令。

從另一部磁碟機裝載空間
您可以藉由從另一部磁碟機裝載空間，在磁碟機上取得更多空間。

您可以使用下列方法從一部磁碟機將空間裝載到另一部磁碟機：

- 使用 **smit mountfs** 捷徑。
- 使用 **mount** 指令。例如：

  ```
  mount -n nodeA -vnfs /usr/spool /usr/myspool
  ```

  mount 指令可讓檔案系統能在特定位置上使用。

不需重新格式化的磁碟機復原
如果您修復故障的磁碟，並將它放回系統而沒有重新格式化，則您可以讓系統自動啟動，並在開機時重新同步化磁碟機上陳舊的實體分割區。陳舊實體分割區中包含系統無法使用的資料。

如果您對陳舊的實體分割區有疑問，請在指令行上鍵入下列指令：

```
lspv -M PhysVolName
```

其中 *PhysVolName* 是實體磁區名稱。 **lspv** 指令輸出將列出實體磁區上的所有分割區。下列是範例輸出的摘要：

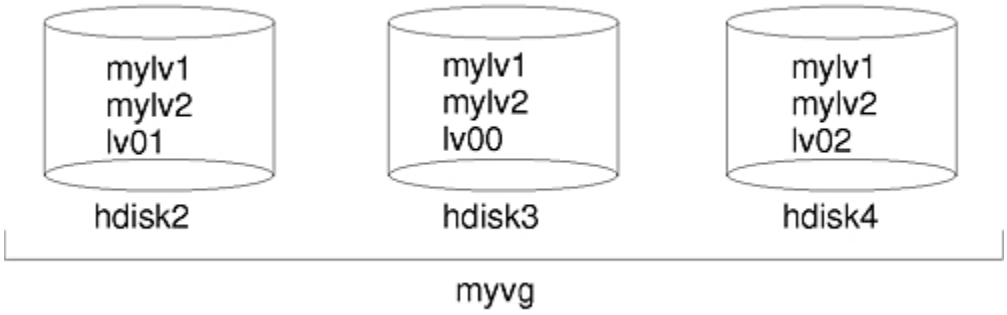
```
hdisk16:112     lv01:4:2        stale
hdisk16:113     lv01:5:2        stale
hdisk16:114     lv01:6:2        stale
hdisk16:115     lv01:7:2        stale
hdisk16:116     lv01:8:2        stale
hdisk16:117     lv01:9:2        stale
hdisk16:118     lv01:10:2       stale
```

第一個直欄顯示實體分割區，且第二個直欄顯示邏輯分割區。第三個直欄中記錄了所有陳舊的實體分割區。

使用重新格式化或更換磁碟機進行回復
當必須重新格式化或替換故障的磁碟時，您可以從故障的磁碟機中回復資料。

小心： 在您重新格式化或更換磁碟機之前，請從發生故障的磁碟中移除對未鏡映檔案系統的所有參照，並從磁區群組與系統配置中移除該磁碟。如果您沒有這麼做，則會在 ODM（物件資料管理程式）及系統配置資料庫中造成問題。這些必要步驟的指示包含在下列程序中，位於替換或重新格式化已故障或即將故障的磁碟之前下方。

下列程序使用一個實務範例，其中稱為 *myvg* 的磁區群組包含三個磁碟機，它們叫做 *hdisk2*、*hdisk3* 及 *hdisk4*。在此實務範例中，*hdisk3* 發生故障。*hdisk2* 上包含非鏡映的邏輯磁區 *lv01* 及 *mylv* 邏輯磁區的副本。*mylv* 邏輯磁區是鏡映且有三個副本，每一個副本在其磁碟中都佔用兩個實體分割區。發生故障的 *hdisk3* 包含 *mylv* 的另一個副本，以及稱為 *lv00* 的非鏡映邏輯磁區。最後，*hdisk4* 包含 *mylv* 的第三個副本，以及稱為 *lv02* 的邏輯磁區。下圖顯示此範例情節。



此程序分為下列幾個主要部分：

· 在更換或重新格式化故障的磁碟之前為保護資料而執行的作業

· 要重新格式化或更換磁碟時所遵循的程序

· 在重新格式化或更換磁碟之後為回復資料而執行的作業

更換或重新格式化已故障或即將故障的磁碟之前：

1. 以 root 權限登入。
2. 如果您不熟悉故障磁碟機上的邏輯磁區，請使用作業磁碟來檢視故障磁碟的內容。
   例如，若要使用 *hdisk4* 來查看 *hdisk3*，請在指令行上鍵入下列指令：

   ```
   lspv -M -n hdisk4 hdisk3
   ```

   lspv 指令會顯示磁區群組間的實體磁區相關資訊。輸出如下：

   ```
   hdisk3:1        mylv:1
   hdisk3:2        mylv:2
   hdisk3:3        lv00:1
   hdisk3:4-50
   ```

   第一個直欄顯示實體分割區，第二個直欄顯示邏輯分割區。分割區 4 到 50 是可用的。
3. 如果可能，請備份故障裝置上所有單一副本邏輯磁區。如需指示，請參閱備份使用者檔案或檔案系統。
4. 如果您具有單一副本檔案系統，請將它們從磁碟中卸載。
   (您可以從 **lspv** 指令的輸出來識別單一副本檔案系統。單一副本檔案系統的邏輯分割區數目與輸出中的實體分割區數目相同。）鏡映檔案系統不必卸載。

   在該範例情節中，故障磁碟 hdisk3 上的 lv00 是單一副本檔案系統。若要卸載它，請鍵入下列指令：

   ```
   unmount /dev/lv00
   ```

   假設 /etc/filesystems 檔案不只是位於故障的磁碟上，如果您不知道檔案系統的名稱，請在指令行上鍵入 mount，以列出所有已裝載的檔案系統，並尋找與邏輯磁區相關聯的名稱。您也可以針對 /etc/filesystems 檔案使用 **grep** 指令，僅列出與邏輯磁區相關的檔案系統名稱（如果有的話）。例如：

   ```
   grep lv00 /etc/filesystems
   ```

   輸出類似下列範例：

```
dev             = /dev/lv00
log             = /dev/loglv00
```

附註：

a. 如果您嘗試卸載的檔案系統正在使用中，則 **unmount** 指令會失敗。唯有當沒有開啟檔案系統的任何檔案且該裝置上沒有使用者的現行目錄時，才會執行 **unmount** 指令。

b. **unmount** 指令的另一個名稱是 **umount**。這兩個名稱是可交換的。

5. 鍵入 **rmfs** 指令， 從故障的實體磁區中移除所有單一副本檔案系統：

```
rmfs /FSname
```

6. 移除故障磁碟上所有鏡映的邏輯磁區。

註：您不可以對 rootvg 磁區群組的實體磁區中的 hd5 和 hd7 邏輯磁區使用 **rmlvcopy**。系統不允許您移除這些邏輯磁區，因為這些是唯一副本。

rmlvcopy 指令會從每一個邏輯分割區中移除副本。例如，請鍵入：

```
rmlvcopy mylv 2 hdisk3
```

移除 *hdisk3* 上的副本，就可以將屬於 *mylv* 邏輯磁區的每一個邏輯分割區的副本數從三個減成兩個（一個在 *hdisk4* 上，另一個在 *hdisk2* 上）。

7. 如果故障磁碟是 root 磁區群組的一部分，而且包含邏輯磁區 hd7，請在指令行上鍵入下列指令，來移除主要傾出裝置 (hd7)：

```
sysdumpdev -P -p /dev/sysdumpnull
```

sysdumpdev 指令會變更執行系統的主要或次要傾出裝置位置。當您重新開機時，傾出裝置會返回它原來的位置。

註：您可以選擇傾出到 DVD 裝置。如需如何將 DVD 配置成傾出裝置的相關資訊，請參閱 **sysdumpdev**。

8. 使用下列指令來移除磁碟上的任何分頁空間：

```
rmps PSname
```

其中 *PSname* 是要移除之分頁空間的名稱，它實際上是分頁空間所在之邏輯磁區的名稱。

如果 **rmps** 指令未順利完成，則在繼續此程序之前，必須使用 **smit chps** 捷徑來取消啟動主要分頁空間，並重新開機。如果有分頁空間處於作用中，則步驟 10 中的 **reducevg** 指令可能失敗。

9. 使用 **rmlv** 指令，從磁區群組中移除任何其他的邏輯磁區（例如，不包含檔案系統的邏輯磁區）。
例如，請鍵入：

```
rmlv -f lv00
```

10. 使用 **reducevg** 指令，從磁區群組中移除故障的磁碟。
例如，請鍵入：

```
reducevg -df myvg hdisk3
```

如果您無法執行 **reducevg** 指令，或該指令未順利完成，在您重新格式化或更換磁碟之後，可利用步驟 13 中的程序來協助清除 VGDA/ODM 資訊。

替換或重新格式化已故障或即將故障的磁碟：

11. 下一個步驟取決於您是否要重新格式化或更換磁碟，以及您所使用的硬體類型：

· 如果您要重新格式化磁碟機，請使用下列程序：

a. 以 root 權限，在指令行上鍵入下列 SMIT 捷徑：

```
smit diag
```

b. 選取**現行 shell 診斷**，以進入「AIX 診斷」工具。

c. 在您閱讀**診斷作業指示**畫面之後，請按 Enter 鍵。

d. 選取**作業選項**。

e. 向下捲動作業清單，以尋找並選取**格式化媒體**。

f. 選取您要重新格式化的磁碟。確認要重新格式化磁碟之後，將會消除磁碟上的所有內容。

重新格式化磁碟之後，請繼續步驟 12。

- 如果您的系統支援熱交換磁碟，請使用第 25 頁的『在系統仍可用時從磁碟故障中回復』中的程序，然後繼續步驟 13。
- 如果系統不支援熱抽換磁碟，請執行下列步驟：
  - 使用 SMIT 捷徑 **smit rmvdsk** 來關閉舊磁碟機的電源。將資料庫欄位中的 KEEP 定義變更為 No。
  - 請洽詢下一層次的系統支援來更換磁碟機。

更換或重新格式化已故障或即將故障的磁碟之後：

12. 請遵循第 12 頁的『配置磁碟』及第 13 頁的『使可用磁碟成為實體磁區』中的指示。
13. 如果在磁碟格式化之前，您無法對舊磁區群組中的磁碟使用 **reducevg** 指令（步驟 10），則下列程序可協助清除 VGDA/ODM 資訊。

    a. 如果磁區群組只含有一個已重新格式化的磁碟，請鍵入：

    ```
    exportvg VGName
    ```

    其中 *VGName* 是磁區群組名稱。

    b. 如果磁區群組是由多個磁碟所組成，請在指令行上鍵入下列指令：

    ```
    varyonvg VGName
    ```

    系統會顯示遺失或無法使用磁碟的相關訊息，並且會列出新（或重新格式化）的磁碟。請記下新磁碟的實體磁區 ID (PVID)，它是列在 **varyonvg** 訊息中。它是在遺漏的磁碟名稱與標籤 PVNOTFND 之間 16 個字元的字串。例如：

    ```
    hdisk3 00083772caa7896e PVNOTFND
    ```

    請鍵入：

    ```
    varyonvg -f VGName
    ```

    遺漏的磁碟現在已顯示 PVREMOVED 標籤。例如：

    ```
    hdisk3 00083772caa7896e PVREMOVED
    ```

    然後，鍵入指令：

    ```
    reducevg -df VGName PVID
    ```

    其中 PVID 是實體磁區 ID（在此範例情節中是 00083772caa7896e）。
14. 若要將新磁碟機新增到磁區群組中，請使用 **extendvg** 指令。

    例如，請鍵入：

    ```
    extendvg myvg hdisk3
    ```
15. 若要在新（或重新格式化）的磁碟機上重建單一副本邏輯磁區，請使用 **mklv** 指令。

    例如，請鍵入：

    ```
    mklv -y lv00 myvg 1 hdisk3
    ```

    此範例會在 *hdisk3* 磁碟機上重建 lv00 邏輯磁區。1 表示此邏輯磁區未鏡映。
16. 若要在邏輯磁區上重建檔案系統，請使用 **crfs** 指令。

例如，請鍵入：

```
crfs -v jfs -d lv00 -m /dev/lv00
```

17. 若要從備份媒體中還原單一副本檔案系統資料，請參閱從備份映像檔中還原使用者檔案。
18. 若要重建邏輯磁區的鏡映副本，請使用 **mklvcopy** 指令。
    例如，請鍵入：

    ```
    mklvcopy mylv 3 hdisk3
    ```

    此範例會在 *hdisk3* 上建立 *mylv* 邏輯磁區之鏡映的第三個分割區。
19. 若要使新的鏡映與其他鏡映上的資料同步化（在此範例中，為 *hdisk2* 和 *hdisk4*），請使用 **syncvg** 指令。
    例如，請鍵入：

    ```
    syncvg -p hdisk3
    ```

因此，所有鏡映的檔案系統都必須還原且保持最新。如果您可以備份單一副本檔案系統，則它們也已備妥可供使用。您必須能夠繼續進行系統的正常使用。

從故障的磁碟機中回復的範例
若要從故障的磁碟機中回復，請退回您之前進入的方法； 亦即，列出您逐步建立磁區群組的步驟，然後退回去。

下面範例是此技術的圖例。它顯示如何建立鏡映邏輯磁區，然後如何變更它，並在磁碟故障時，一次退回一個步驟。

註： 下列範例說明特定的案例。它不能用來當成一般回復程序基礎的一般原型。

1. 系統管理員 Jane 鍵入下列指令，在 hdisk1 上建立了名為 **workvg** 的磁區群組：

   ```
   mkvg -y workvg hdisk1
   ```

2. 然後她鍵入下列指令，又為此磁區群組建立了兩個磁碟：

   ```
   extendvg workvg hdisk2
   extendvg workvg hdisk3
   ```

3. Jane 建立了 40 MB、有三個副本的邏輯磁區。
   每一個副本各在構成 **workvg** 磁區群組的三個磁碟上。她使用了下列指令：

   ```
   mklv -y testlv workvg 10
   mklvcopy testlv 3
   ```

Jane 建立鏡映的 workvg 磁區群組之後，hdisk2 失敗。因此，她執行了下列步驟來進行回復：

1. 她鍵入下列指令，從 hdisk2 移除邏輯磁區副本：

   ```
   rmlvcopy testlv 2 hdisk2
   ```

2. 她鍵入下列指令，將 hdisk2 從系統中分離，以便更新 ODM 與 VGDA：

   ```
   reducevg workvg hdisk2
   ```

3. 她鍵入下列指令，從系統配置中移除 hdisk2，以準備置換：

   ```
   rmdev -l hdisk2 -d
   ```

4. 她鍵入下列指令，選擇了關閉系統：

   ```
   shutdown -F
   ```

5. 替換磁碟。新磁碟與先前磁碟 hdisk2 的 SCSI ID 不同。

6. 她重新啟動系統。

   因為您有新磁碟（系統會發現此磁碟上的新 PVID），系統會選擇第一個開啟的 hdisk 名稱。因為在步驟 3 中使用了 **-d** 旗標，而釋出了名稱 hdisk2，因此系統會選擇 hdisk2 作為新磁碟的名稱。如果未使用 **-d** 旗標，則會選擇 hdisk4 作為新名稱。

7. Jane 鍵入了下列指令，將此磁碟新增到 **workvg** 磁區群組中：

   ```
   extendvg workvg hdisk2
   ```

8. 她鍵入下列指令，建立了邏輯磁區的兩個鏡映副本：

   ```
   mklvcopy testlv 3
   ```

   「邏輯磁區管理程式 (LVM)」已自動將第三個邏輯磁區副本放在新的 hdisk2 上。

在系統仍可用時從磁碟故障中回復
您可以使用熱移除功能，從磁碟故障中回復。

使用可熱抽取特性從磁碟故障中回復的程序大部分與第 20 頁的『不需重新格式化的磁碟機復原』中所述的相同，但下列是例外：

1. 若要卸載磁碟上的檔案系統，請使用裝載 JFS 或 JFS2 程序。
2. 若要從磁區群組或從作業系統中移除磁碟，請使用第 39 頁的『移除不含資料的磁碟』程序。
3. 若要用新的磁碟替換故障的磁碟，您不必關閉系統。請使用下列順序的程序：
   a) 第 28 頁的『邏輯磁區儲存體』
   b) 第 12 頁的『配置磁碟』
   c) 請繼續第 20 頁的『使用重新格式化或更換磁碟機進行回復』的步驟 13。

當磁區群組包含一個磁碟時替換磁碟
如果可以存取磁區群組中發生錯誤的磁碟，請使用下列其中一個程序。

· 第 10 頁的『移轉實體磁區的內容』

如果磁碟發生錯誤而無法存取，請遵循下列步驟：

1. 匯出磁區群組。
2. 更換磁碟機。
3. 從存在的備份媒體中重建資料。

實體及邏輯磁區錯誤
有數種常見的實體及邏輯磁區錯誤，您可以對它們進行疑難排解。

熱點問題
如果您注意到在存取邏輯磁區時發現效能退化，則可能是在發生過多磁碟 I/O 的邏輯磁區中有熱點存在。

如需相關資訊，請參閱第 49 頁的『邏輯磁區中的熱點管理』。

***LVCB* 警告**
如果 LVCB 包含無效資訊，即會產生警告。

邏輯磁區控制區塊 (LVCB) 是邏輯磁區的第一個區塊。LVCB 的大小是磁區群組內實體磁區的區塊大小。 此區域保留了重要資訊，如邏輯磁區的建立日期、鏡映副本的相關資訊及 JFS 中可能的裝載點。需要特定的 LVM 指令來更新 LVCB，作為 LVM 中演算法的一部分。讀取並分析舊 LVCB 以查看其是否有效。如果該資訊是有效的 LVCB 資訊，則會更新 LVCB。如果該資訊無效，則不執行 LVCB 更新，且您可能接收到下列訊息：

```
警告，無法寫入 lv 控制區塊資料。
```

大部分情況下，當資料庫程式略過 JFS，並存取作為儲存媒體的原始邏輯磁區時，會產生此訊息。發生這種狀況時，資料庫的相關資訊會逐字地改寫 LVCB。對於原始邏輯磁區來說，這並不嚴重。改寫 LVCB 之後，使用者仍可以：

· 擴充邏輯磁區

- 建立邏輯磁區的鏡映副本
- 移除邏輯磁區
- 建立日誌型檔案系統來裝載邏輯磁區

刪除 LVCB 時有限制。已刪除 LVCB 的邏輯磁區可能未順利匯入其他系統。匯入期間，LVM **importvg** 指令會掃描磁區群組中所有已定義邏輯磁區的 LVCB，以取得邏輯磁區的相關資訊。如果 LVCB 不存在，已匯入的磁區群組仍會將邏輯磁區定義給正在存取此磁區群組的新系統，且使用者仍可存取原始邏輯磁區。不過，通常會發生下列狀況：

- 會遺失所有 JFS 資訊，且不會將相關裝載點匯入新系統。在此情況下，您必須建立新的裝載點，且不能確保檔案系統中儲存之先前資料的可用性。
- 找不到有關邏輯磁區的部分非 JFS 資訊。當發生此狀況時，系統會使用預設邏輯磁區資訊來大量輸入 ODM 資訊。因此，**lslv** 指令的部分輸出可能與實際的邏輯磁區不一致。如果原始磁碟上仍存在任何邏輯磁區副本，則此資訊不會正確反映在 ODM 資料庫中。使用 **rmlvcopy** 及 **mklvcopy** 指令重新建置所有邏輯磁區副本並同步化 ODM。

實體分割區限制

設計「邏輯磁區管理程式 (LVM)」時，每個邏輯分割區會對映一個實體分割區 (PP)。且每個實體分割區會對映一些磁區。LVM 的設計會將 LVM 每磁碟可追蹤之實體分割區的數目限制為 1016。在大部分狀況下，並非全部的 1016 個追蹤分割區都由某個磁碟所使用。

當超出此限制時，您可能會看到類似下列的訊息：

```
0516-1162 extendvg：警告，PPsize 的「實體分割區大小」
需要為  PVname 建立 TotalPPs 個分割區。磁區群組
VGname 的限制是每個實體磁區的 LIMIT 實體分割區。
請使用 chvg 指令並加上 -t 選項來嘗試為此磁區群組變更每個實體磁區的最大實體分割區數目。
```

其中：

PPsize
是 1 MB 到 1 GB 的二次方之間。

Total PP
是給定 *PPsize* 後，此磁碟上實體分割區的總數。

PVname
是實體磁區的名稱，例如：hdisk3。

VGname
是磁區群組名稱。

LIMIT
是 1016 或 1016 的倍數。

在下列案例中強制實施此限制：

1. 當使用 **mkvg** 指令建立磁區群組時，您已指定磁區群組中磁碟上的實體分割區數目超過 1016。若要避免此限制，您可以從範圍 1、2、4（預設值）、8、16、32、64、128、256、512 或 1024 MB 的實體分割區大小中選取，並使用 **mkvg -s** 指令來建立磁區群組。另外，您可以使用適當的因數，使每個磁碟的分割區數目是 1016 的倍數，並使用 **mkvg -t** 指令來建立磁區群組。
2. 使用 **extendvg** 指令將磁碟新增到預先存在的磁區群組時，新磁碟導致超出 1016 的限制。若要解決此狀況，請使用 **chvg -t** 指令轉換現有的磁區群組，以容納每個磁碟包含的分割區為 1016 的倍數。另外，您可以用更大的分割區大小來重建允許新磁碟的磁區群組，或可以為新磁碟建立包含更大實體大小的單機磁區群組。

分割區限制與 rootvg

如果安裝程式碼偵測到 rootvg 磁碟機大於 4 GB，它就會變更 **mkvg-s** 值，直到整個磁碟容量可對映到可用的 1016 個磁軌為止。此安裝變更同時暗示，亦會以該實體分割區大小來定義新增至 rootvg 的所有其他磁碟（無論其大小為何）。

分割區限制與 RAID 系統

對於使用磁碟陣列 (RAID) 的系統， LVM 所使用的 /dev/hdiskX 名稱可能由許多非 4 GB 的磁碟所組成。在此狀況下，基本需求仍是 1016。LVM 不知道實際組成 /dev/hdiskX 之個別磁碟的大小。LVM 的 1016 限制是基於 /dev/hdiskX 的已辨識大小，而非組成 /dev/hdiskX 的實際實體磁碟。

裝置配置資料庫同步化

系統故障可導致裝置配置資料庫與 LVM 不一致。您可以將裝置配置資料庫與 LVM 資訊同步化。

當裝置配置資料庫變成與 LVM 不一致時，邏輯磁區指令會產生這樣的錯誤訊息：

```
0516-322 「裝置配置資料庫」不一致 ...
```

或

```
0516-306 在「裝置配置資料庫」中找不到邏輯磁區 LVname。
```

（其中名為 *LVname* 的邏輯磁區通常可用）。

小心： 請勿移除磁區群組或邏輯磁區的 /dev 項目。請勿使用「物件資料管理程式」來變更磁區群組或邏輯磁區的資料庫項目。

若要以 root 權限來同步化裝置配置資料庫與 LVM 資訊，請在指令行上鍵入下列指令：

```
synclvodm -v VGName
```

其中 *VGName* 是您要同步化之磁區群組的名稱。

修正磁區群組錯誤

使用下列方法，可以修正磁區群組錯誤。

如果 **importvg** 指令未正確地運作，請嘗試重新整理裝置配置資料庫。

置換轉開失敗

小心： 忽略轉開失敗是不尋常的作業；請在繼續之前檢查所有其他可能的問題來源，如硬體、纜線、配接卡及電源。轉開處理程序期間忽略 Quorum 失敗僅用在緊急狀況下，且僅當作最後的手段（例如，從故障的磁碟援救資料）。當忽略 Quorum 失敗時，對於包含在所選之 VGDA 及 VGSA 副本中的管理資料，將無法保證資料完整性。

當您選擇忽略 Quorum 的不存在以強制轉開磁區群組時，此轉開處理程序期間遺失之所有實體磁區的 PV STATE 都將變更為已移除。這表示將所有 VGDA 及 VGSA 副本從這些實體磁區中移除。完成此動作之後，這些實體磁區將不再參與 Quorum 檢查，在磁區群組中也不允許它們成為作用中，直到您將它們傳回磁區群組為止。如果磁區群組未遺失 Quorum，則會忽略 varyonvg -f 旗標（用來置換 Quorum 遺失）。

在下列一或多個狀況下，您可能會想要忽略轉開失敗，以便可以存取磁區群組中可用磁碟上的資料：

- 無法使用的實體磁區似乎已永久損壞。
- 您可以確認當磁區群組上次轉開時，至少有一個目前可存取的實體磁區（必須亦包含良好的 VGDA 及 VGSA 副本）在線上。取消配置並關閉遺失實體磁區的電源，直到可診斷並修復它們。

使用下列程序以避免在一個磁碟遺失或可能很快故障並需要修復時遺失 Quorum：

1. 若要從磁區群組中暫時移除磁區，請鍵入：

   ```
   chpv -vr PVname
   ```

   當此指令完成時，實體磁區 *PVname* 就不再作為 Quorum 檢查的因素。不過，在雙磁碟磁區群組中，如果您在包含兩個 VGDA/VGSA 的磁碟上嘗試 **chpv** 指令，則此指令會失敗。該指令不允許您讓 Quorum 遺失。
2. 如果您需要移除磁碟以進行修復，請關閉系統電源，並移除磁碟。（請參閱第 19 頁的『為磁碟機問題進行疑難排解』，以取得指示。） 修正磁碟，並將磁碟送回給系統之後，請繼續下一個步驟。

3. 若要使磁碟重新可用於磁區群組以進行 Quorum 檢查，請鍵入：

```
chpv -v a PVname
```

註：chpv 指令僅用於 Quorum 檢查變更。位於磁碟上的資料仍在原處，且如果未將磁碟送回給系統，則必須將資料移至或複製到其他磁碟。

VGDA 警告
在部分案例中，使用者在將新磁碟新增到現有磁區群組，或建立新磁區群組時，會遇到問題。LVM 所提供的訊息是：

```
0516-1163 extendvg：VGname 已擁有實體磁區最大數目。當每個實體磁區之實體分割區的最大數目是 LIMIT 時，
磁區群組 VGname 之實體磁區的最大數目會是 MaxDisks。
```

其中：

VGname
是磁區群組名稱。

LIMIT
是 1016 或 1016 的倍數。

MaxDisks
是磁區群組中磁碟的最大數目。例如，如果每個磁碟有 1016 個實體分割區 (PP)，則 *MaxDisk* 是 32；如果有 2032 個，那麼 *MaxDisk* 是 16。

您可以修改 `image.data` 檔案，然後使用替代磁碟安裝，或使用 **mksysb** 指令還原系統，以將磁區群組重建為大的磁區群組。如需相關資訊，請參閱安裝與移轉。

在舊版的 AIX 上， 當限制小於 32 個磁碟時，此 VGDA 上限說明就不包括 **rootvg**。為提供給使用者更多的可用磁碟空間，在建立 rootvg 時，**mkvg -d** 指令會使用安裝功能表中選取之磁碟數目作為參照數目。一個磁碟時，此 **-d** 數字是 7，每多選取一個磁碟，該數字增加 1。例如，如果選取兩個磁碟，則該數字是 8，如果選取三個磁碟，則該數字是 9，以此類推。

邏輯磁區儲存體

邏輯磁區是位於實體磁區上的資訊群組。

結構階層是用來管理磁碟儲存體。每一個個別磁碟機，稱為實體磁區 (PV) 都具有名稱，例如 `/dev/hdisk0`。使用中的每個實體磁區都屬於某個磁區群組 (VG)。 磁區群組中的所有實體磁區被分成相同大小的實體分割區 (PP)。基於空間配置目的，每一個實體磁區都將分成五個區域（**outer_edge**、**inner_edge**、**outer_middle**、**inner_middle** 及 **center**）。 每一個範圍中的實體分割區數，因磁碟機容量總計而異。

在每一個磁區群組內，都定義了一或多個邏輯磁區。邏輯磁區上的資料對使用者而言似乎是連續的，但在實體磁區上可能是不連續的。這可讓檔案系統、分頁空間及其他邏輯磁區重新調整大小或重新定位，也可以跨越多個實體磁區以及抄寫彼此的內容，從而在儲存資料時，擁有更高的彈性及可用性。

每一個邏輯磁區都由一或多個邏輯分割區所組成。 每一個邏輯分割區至少對應一個實體分割區。 如果對邏輯磁區指定鏡映，則會配置其他實體分割區來儲存每一個邏輯分割區的其他副本。 雖然邏輯分割區是連續編號，但基礎實體分割區不一定要連續。

邏輯磁區在系統方面具有多種用途（如分頁），但每個邏輯磁區的用途僅限一種。 許多邏輯磁區都包含單一日誌型檔案系統（JFS 或 JFS2）。每一個 JFS 是由頁面大小 (4 KB) 區塊的儲存區所組成。 資料要寫入到檔案中時，會為這個檔案配置一或多個其他區塊。這些區塊可能彼此不連續，或與先前配置到檔案的其他區塊不連續。 特定的檔案系統可定義其片段大小為 4 KB（512 位元組、1 KB、2 KB）以下。

安裝之後，系統會具有一個磁區群組（rootvg 磁區群組），由啟動系統所需的一組基本邏輯磁區，以及 在安裝 Script 指定的任何其他邏輯磁區所組成。連接到系統的任何其他實體磁區，都可新增到磁區群組（使用 **extendvg** 指令）。您可以將實體磁區新增到 rootvg 磁區群組或其他磁區群組（使用 **mkvg** 指令定義）。邏輯磁區可以使用指令、功能表驅動式「系統管理介面工具 (SMIT)」介面來進行自訂。

相關工作

遺失實體磁區時通知管理者
在無法存取邏輯磁區時，雖然 AIX 會記載錯誤，但還是可能有些情況下是偵測不到錯誤的。

從磁區群組中分割鏡映磁碟
Snapshot 支援會協助您保護鏡映磁區群組的一致性，以避免可能發生的磁碟故障。

減少 root 磁區群組中檔案系統的大小
將所有檔案系統大小減至最小的最簡單方法是，從備份還原基本作業系統時， 將 **SHRINK** 選項設為 **yes**。

準備安裝裝置

在系統上安裝裝置包括識別裝置的連接位置、裝置的實際連接裝置，以及使用「配置管理程式」或 SMIT 來配置裝置。

註： 下列程序需要將系統關機以安裝裝置。不是所有裝置安裝都需要關閉系統。請參照特定裝置隨附的文件。

本主題說明所有裝置通用的安裝作業。因為您可以在系統上安裝的裝置種類非常多，所以只提供一般程序。如需特定資訊，請參閱特定裝置隨附的安裝指示。

1. 停止在主機上執行的所有應用程式，並使用 **shutdown** 指令關閉主機。
2. 關閉主機與所有連接的裝置。
3. 拔掉主機與所有連接裝置的插頭。
4. 使用裝置設定與操作員手冊中所說明的程序，連接新裝置到系統。
5. 插入主機與所有連接裝置的插頭。
6. 開啟所有連接裝置，並仍關閉主機。
7. 當所有裝置完成開機自我測試 (POST) 後，打開主機。

「配置管理程式」會自動掃描連接的裝置，並配置它偵測到的任何新裝置。使用預設屬性配置新裝置，並記錄在自訂的配置資料庫中，使裝置處於**可用**狀態。

您可以使用 SMIT 捷徑 **smit dev** 來手動配置裝置。如果您需要自訂裝置屬性或如果無法自動配置裝置，請參閱該裝置隨附的裝置文件，了解特定的配置基本需求。

配置可讀寫光碟裝置

有兩種方法可以配置可讀寫光碟裝置。

可讀寫光碟裝置必須連接至系統，並開啟電源。

第 1 種方法

方法 1 是兩種方法中較快速的。它只配置指定的可讀寫光碟裝置。若要使用此方法，您必須提供下列資訊：

項目	說明
子類別	定義如何連接磁碟機。
類型	指定可讀寫光碟裝置的類型。
母項名稱	指定磁碟機連接的系統附件。
連接位置	指定磁碟機的邏輯位址。

輸入下列指令以配置可讀寫光碟裝置：

```
mkdev -c rwoptical -s Subclass -t Type -p ParentName -w WhereConnected
```

下面是一個可讀寫光碟裝置的範例，其中的 SCSI ID 是 6、邏輯單元號碼 0 且連接至第三個 (scsi3) SCSI 匯流排：

```
mkdev -c rwoptical -s scsi -t osomd -p scsi3 -w 6,0 -a pv=yes
```

第 2 種方法

方法 2 使用「配置管理程式」，以搜尋現行配置、偵測任何新裝置並自動配置裝置。當您只知道可讀寫光碟裝置的少數資訊時，可以使用此方法。

1. 使用配置管理程式，以配置系統上所有新偵測到的裝置（包括可讀寫光碟裝置），請鍵入：

```
cfgmgr
```

2. 鍵入下列指令，以列出所有目前已配置的可讀寫光碟裝置之名稱、位置碼及類型：

```
lsdev -C -c rwoptical
```

3. 利用與要新增的磁碟機之位置相符的位置碼，判斷新配置的可讀寫光碟裝置名稱。

配置大量裝置

裝置包含硬體元件（如印表機、磁碟機、配接卡、匯流排及機殼），以及虛擬裝置（如錯誤特殊檔案及空值特殊檔案）。 裝置驅動程式位於 /usr/lib/drivers 目錄中。

AIX 在不同系統中可支援的裝置個數，取決於幾個重要因素。下列因素會影響支援裝置的檔案系統：

- 配置大量裝置需於 ODM 裝置配置資料庫中儲存更多資訊。可能亦需要更多裝置特殊檔案。 因此，檔案系統中需要有更多的空間及 i-node。
- 於 ODM 裝置配置資料庫中，部分裝置比其他裝置需要更多的空間。 所使用之特殊檔案或 i-node 的數目亦會隨裝置的不同而不同。 因此，檔案系統所需的空間及 i-node 數量要根據系統上的裝置類型而定。
- 多路徑 I/O (MPIO) 裝置比非 MPIO 裝置需要更多的空間，因為裝置本身的 資訊及到裝置之每條路徑的資訊皆儲存於 ODM 中。粗略而言，假設每條路徑佔據 裝置五分之一的空間。例如，具有五條路徑的 MPIO 裝置與兩個非 MPIO 裝置的空間相等。
- AIX 會將邏輯裝置及實體裝置併入 ODM 裝置配置資料庫中。 邏輯裝置包括磁區群組、邏輯磁區、網路介面等。於某些情況下， 邏輯與實體裝置間的關係對受支援裝置的總數有極大地影響。例如， 若您為連接至系統的每個實體磁碟定義一個具有兩個邏輯磁區的磁區群組，則會導致每個磁碟有四個 AIX 裝置。另一方面，若您為每個實體磁碟 定義一個具有六個邏輯磁區的磁區群組，則每個磁碟將有八個 AIX 裝置。 因此，僅可連接半數磁碟。
- 變更預設設定的裝置屬性會導致更大的 ODM 裝置配置資料庫，並可能導致支援更少的裝置。
- 裝置越多，所需的實際記憶體就越多。

AIX 使用兩個檔案系統來支援裝置：

- 於未裝載分頁空間及磁碟檔案系統的環境中開機期間，會使用 RAM 檔案系統。 RAM 檔案系統大小為系統記憶體大小的 25%，最大為 128 MB。RAM 檔案系統中之 每 KB 配置一個 i-node。AIX 作業系統的最小系統記憶體需求為 256 MB，其轉換為 64 MB 具有 65536 個 i-node 的最小 RAM 檔案系統大小。 若系統記憶體大小為 512 MB 或更大，則 RAM 檔案系統將會是其最大大小 128 MB，具有 131072 個 i-node。 若支援附屬裝置所需的 RAM 檔案系統空間數量或 i-node 數目超過配置給 RAM 磁碟的數量，則系統可能無法開機。若是這樣的狀況，則必須移除部分裝置。
- 只要 rootvg 中有未配置的分割區，就可增加磁碟上根檔案系統 (rootvg) 的空間及 i-node。若使用最大 RAM 檔案系統大小，很可能可配置最多 25,000 個 AIX 裝置。 這些數目包括實體及邏輯裝置。根據本節中所提到的各種因素，您的系統可能可配置多於或少於此數目的裝置。

註 : 若系統中有大量的裝置，配置時間越長，則會導致越長的開機時間。

新增抽取式媒體磁碟機

您可以新增抽取式媒體磁碟機。

下列程序使用 SMIT 將光碟機新增至系統。其他類型的抽取式媒體磁碟機是使用不同的捷徑來新增，但全都遵循相同的一般程序。您還可以使用「配置管理程式」或 **mkdev** 指令，來新增抽取式媒體磁碟機。

此作法範例情節中的資訊已使用特定版本的 AIX 進行測試。依據您的 AIX 版本及層次，所得到的結果可能大不相同。

1. 若要將光碟機新增至系統，請根據系統隨附的文件來安裝硬體。

2. 使用 root 權限，鍵入下列 SMIT 捷徑：

```
smit makcdr
```

3. 在下一個畫面中，從可用的支援磁碟機清單中選取磁碟機類型。
4. 在下一個畫面中，從可用的清單中選取母項配接卡。
5. 在下一個畫面中，至少需從可用的清單中選取連線位址。您也可以使用此畫面來選取其他選項。完成後，請按 Enter 鍵，SMIT 即會加入新的光碟機。

此時，系統會辨識出新的光碟機。若要新增可讀寫光碟裝置，請使用 **smit makomd** 捷徑。若要新增磁帶機，請使用 **smit maktpe** 捷徑。

如需相關資訊，請參閱 *Commands Reference, Volume 3* 中的 **mkdev** 指令說明。

邏輯磁區儲存體的空間收回支援

在含有 7200-01 技術層次的 AIX 7.2 或更新版本中，「邏輯磁區管理程式 (LVM)」支援對能夠收回空間的實體磁區進行空間收回。

LVM 會通知磁碟驅動程式，磁碟驅動程式接下來會通知儲存體子系統已不再使用分割區空間，因此該儲存體子系統可以收回已配置的空間。 磁碟驅動程式可協助 LVM 偵測實體磁區的空間收回功能。 LVM 和檔案系統配置指令（例如 `rmlv` 指令、`rmlvcopy` 指令及 `chfs(shrink fs)` 指令）可在釋出後起始分割區的空間收回。 LVM 會在 `varyonvg` 或 `extendvg` 指令執行期間開啟實體磁區時，偵測該磁區的空間收回功能。LVM 還會在磁區群組處於線上狀態時嘗試偵測它。 如果狀態變更偵測需要重新開啟實體磁區，則管理者必須針對磁區群組執行 `varyoffvg` 指令，然後執行 `varyonvg` 指令。

在 AIX 7.2 技術層次 1 之前建立的磁區群組可能具有可用的分割區空間，而該分割區空間不適用於自動收回。管理者可以在那些可用分割區上建立及刪除虛擬的邏輯磁區，以收回此空間。 但是將為分割區自動收回在安裝 AIX 7.2 技術層次 1 之後仍然可用的空間。

在 `rmlv` 之類的指令完成執行之後，用來收回空間的 LVM 處理程序將在背景執行。如果系統在 LVM 處理程序對所有分割區完成收回處理程序之前當機，則會釋出這些分割區，但不會收回擱置中分割區的空間。 如果出現此情境，您可以建立及刪除虛擬的邏輯磁區，以從剩餘的分割區中收回空間。

LVM 處理程序不會延遲處理 `varyoffvg` 指令或 `reducevg` 指令，即使空間收回處理程序處於擱置狀態也一樣。 將捨棄空間收回處理程序，而不是等待該處理程序完成。

註： 指令只會等待已提交給磁碟驅動程式的任何未完成之空間收回要求。

您可以從儲存體子系統使用空間收回功能，來收回從實體磁區釋出的空間。每一個儲存體子系統都預期收回要求與特定數目的實體區塊一致，而且實體區塊數目將根據儲存體子系統而有所不同。因此，有時無法從分割區收回區塊（所有或部分），因為收回大小未與分割區的實體區塊一致。 某些儲存體子系統支援收回大於 LVM 分割區大小的區塊大小，但無法進行局部區塊收回。 在此情境中，LVM 可能無法累計足夠的連續可用分割區，以產生單一收回要求。 因此，當您刪除多個 LVM 分割區時，可能無法收回儲存體子系統中的相等數量空間。 您可以搭配使用 `lvmstat` 指令與 **-r** 選項，來取得 LVM 所產生之空間收回要求的相關資訊。

相關資訊

varyoffvg 指令

邏輯磁區儲存體概念

邏輯磁區（可跨越實體磁區）是由配置到實體分割區上的邏輯分割區所組成。

下圖說明基本邏輯儲存體概念之間的關係。

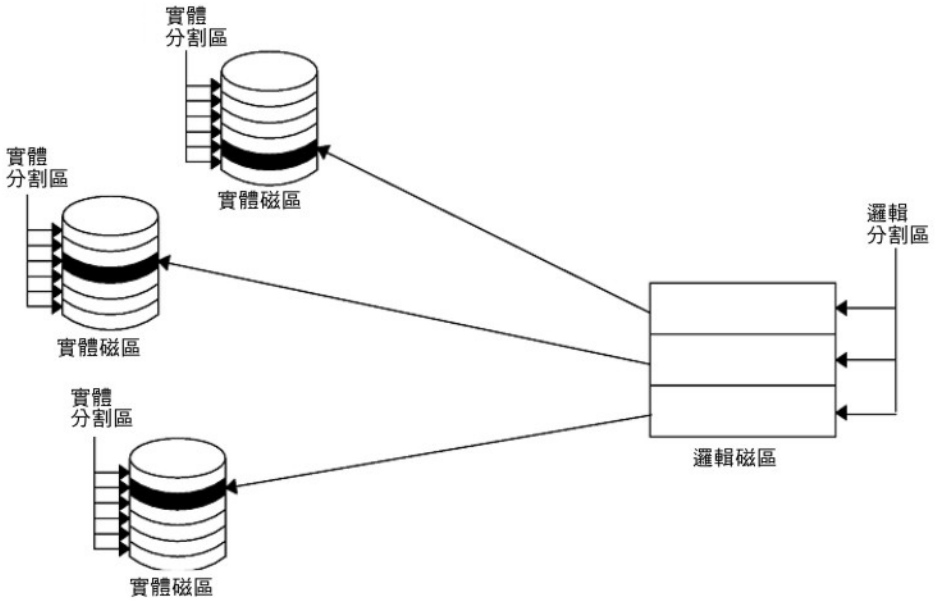


圖 *1.* 磁區群組

實體磁區
磁碟必須先指定為實體磁區並置於可用狀態，方可指派給磁區群組。

實體磁區上記載了特定的配置及識別資訊。此資訊包含對系統而言是唯一的實體磁區 ID。

LVM 可利用磁碟陣列 (RAID) 新增至邏輯單元號碼 (LUN) 所產生的其他空間，方法是將實體分割區新增到與 LUN 關聯的實體磁區。

磁區群組
磁區群組 是 1 到 32 個大小不同之實體磁區的集合。

大型磁區群組可含有 1 到 128 個實體磁區。可調整的磁區群組最多可有 1024 個實體磁區。對於各別系統而言，實體磁區僅可屬於一個磁區群組；最多可有 255 個作用中的磁區群組。

把實體磁區指派給磁區群組時，其儲存媒體的實體區塊，將以建立磁區群組時所指定的大小組織成實體分割區。

安裝系統時，會自動建立一個磁區群組（root 磁區群組，稱為 rootvg），其包含啟動系統所需的一組基本邏輯磁區，以及安裝 Script 所指定的任何其他邏輯磁區。 rootvg 包括分頁空間、日誌、開機資料及傾出儲存體，分別存於各自獨立的邏輯磁區中。 rootvg 的屬性與使用者定義的磁區群組不同。例如，無法匯入或匯出 rootvg。 對 rootvg 執行指令或程序時，您必須熟悉其獨特性質。

您可使用 **mkvg** 指令建立磁區群組。 使用 **extendvg** 指令，可將實體磁區新增至磁區群組； 使用 **chvg** 指令，可利用實體磁區變更後大小的空間； 使用 **reducevg** 指令，可將實體磁區從磁區群組中移除。其他磁區群組可以使用的指令包括：列出 (**lsvg**)、 移除 (**exportvg**)、安裝 (**importvg**)、重組 (**reorgvg**)、同步化 (**syncvg**)、 使其可用 (**varyonvg**)，及使其不可用 (**varyoffvg**)。

小型系統可能僅需要一個磁區群組來包含連接到系統的所有實體磁區。不過，基於安全考量，因為每一個磁區群組都可具有其自己的安全許可權，所以您可能想要建立個別的磁區群組。個別的磁區群組還使得維護更為容易，這是因為除了維修中群組之外，其他群組仍可保持作用中。因為 rootvg 必須總是在線上，所以它僅包含系統作業所需之最小數目的實體磁區。

您可以使用 **migratepv** 指令，將資料從一個實體磁區移到同一磁區群組中的其他實體磁區上。此指令可讓您釋放實體磁區，以便可將它從磁區群組中移除。 例如，您可從要被替換的實體磁區移出資料。

使用較小的實體及邏輯磁區限制所建立的磁區群組，可轉換成可保留更多實體磁區及更多邏輯磁區的格式。此作業需要在磁區群組的每一個實體磁區上有足夠的可用分割區，以擴充磁區群組描述子區域 (VGDA)。所需的可用分割區數目取決於現行 VGDA 的大小及實體分割區大小。由於 VGDA 位於磁碟的邊緣，且需要連續的空間，所以在磁碟的邊緣需要有可用分割區。 若那些分割區配置以供使用者所使用，則會將其移轉至同一磁碟的其他可用分割區。剩餘的實體分割區會重新編號，以反映供 VGDA 使用而減少的分割區。 重新編號會變更此磁區群組的所有實體磁區中邏輯與實體分割區的對映。若您已儲存了邏輯磁區的對映，以供可能需要的回復作業使用，請於完成轉換作業之後，重新產生對映。另外，若使用了對映選項進行 磁區群組的備份，且您計劃要使用那些對映進行還原，則還原作業可能會失敗，原因是分割區號碼可能 不再存在（由於減少的緣故）。建議於轉換之前進行備份；若利用了對映選項，則應緊接在轉換之後進行 備份。因為 VGDA 空間已經顯著地增加，所以每一次 VGDA 更新作業（建立邏輯磁區、變更邏輯磁區、 新增實體磁區等）皆可能執行相當長的時間。

實體分割區

當將實體磁區新增至磁區群組時，會將實體磁區分割成連續的、 等大小的空間單位，稱為實體分割區。實體分割區是儲存體空間配置的最小單位，也是實體磁區上的連續空間。

實體磁區會繼承磁區群組的實體分割區大小，該大小僅可在建立磁區群組時（例如，使用 **mkvg -s** 指令）設定。 下列圖例顯示實體磁區上實體分割區與磁區群組的關係。

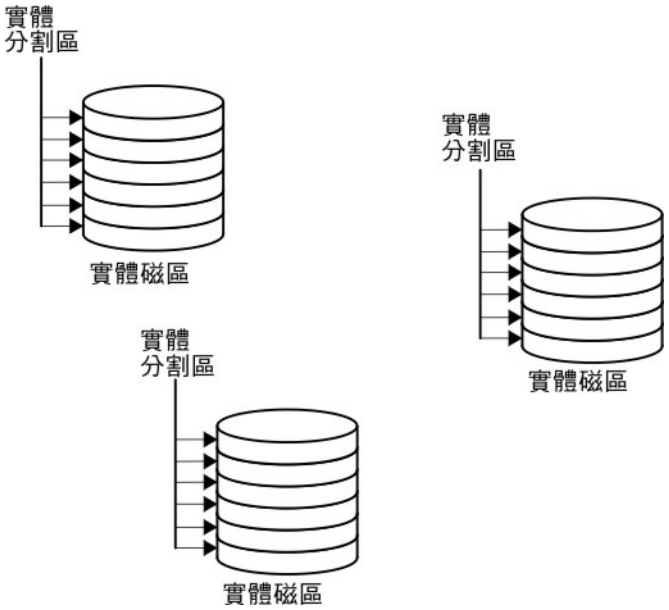


圖 *2.* 包含三個實體磁區的磁區群組

邏輯磁區

在建立磁區群組之後，可以在該磁區群組中建立邏輯磁區。

雖然邏輯磁區可以位於非連續的實體分割區，甚至可以位於多個實體磁區，但是對於使用者及應用程式而言，它似乎是單一、連續、可擴充的磁區。 您可使用 **mklv** 指令，建立額外的邏輯磁區。此指令可讓您指定邏輯磁區的名稱，並定義其性質，包括為其配置的邏輯分割區號碼及位置。

建立邏輯磁區之後，可以使用 **chlv** 指令變更其名稱及性質，且可以使用 **extendlv** 指令增加為其配置的邏輯分割區數目。除非指定更大的值，否則於建立時邏輯磁區的預設最大大小為 512 個邏輯分割區。 使用 **chlv** 指令，可以置換此限制。

註 : 建立邏輯磁區之後，即會關閉使用 **lslv** 指令可看到的性質 LV STATE。 它會在（例如）邏輯磁區中已建立了檔案系統並裝載了邏輯磁區後開啟。

亦可使用 **cplv** 指令複製邏輯磁區； 使用 **lslv** 指令進行列出； 使用 **rmlv** 指令進行移除； 分別使用 **mklvcopy** 及 **rmlvcopy** 指令增加及減少它們維護的副本份數。 重組磁區群組時，亦可將「邏輯磁區」重新定位。

系統可讓您為每一個標準磁區群組定義最多 255 個邏輯磁區（若是大磁區群組則為 511 個，若是可調整的磁區群組則為 4095 個）， 但您實際可定義的數目，取決於為該磁區群組定義的實體儲存體總量，以及您所定義的邏輯磁區大小。

邏輯分割區
當建立邏輯磁區時，您可以為邏輯磁區指定邏輯分割區 的數目。

一個邏輯分割區可以是一個、兩個或三個實體分割區，依您要維護之資料的案例數目而定。指定一個案例表示只有一個邏輯磁區副本（預設值）。在此狀況下，此為一個邏輯分割區與一個實體分割區的直接對映。 每一個案例（包括第一個），都稱為副本。實體分割區的位置（即實體分割區之間的實際距離）由您建立邏輯磁區時指定的選項決定。

檔案系統
邏輯磁區可以定義實體分割區層次的磁碟空間配置。資料管理的細微層次由較高層次軟體元件（如「虛擬記憶體管理程式」或檔案系統）完成。因此，磁碟演化的最後一步是建立檔案系統。

您可以為每一個邏輯磁區都建立一個檔案系統。若要建立檔案系統，請使用 **crfs** 指令。

邏輯儲存體管理的限制
下表顯示邏輯儲存體管理的限制。

雖然每一個磁區群組的實體磁區預設最大數目為 32 個（若是大磁區群組則為 128 個，若是可調整的磁區群組則為 1024 個），但是您可以在使用 **mkvg** 指令時， 為使用者定義的磁區群組設定最大數目。不過，對於 rootvg，系統會在安裝期間自動將此變數設為最大值。

種類	限制
磁區群組	· 32 位元核心的 255 磁區群組 · 64 位元核心的 4096 磁區群組 **註：** 64 位元核心上的裝置表會將作用中主要編號的數目限制為 1024。因此，作用中磁區群組的數目會限制為小於 1024 個磁區群組。
實體磁區	每一磁區群組（MAXPVS/磁區群組因數）個。若是標準磁區群組，MAXPVS 為 32， 若是大磁區群組，則為 128，若是可調整的磁區群組，則為 1024。
實體分割區	正常及大磁區群組：每一實體磁區（1016 x 磁區群組因數）個， 每一個大小最高達 1024 MB。可調整的磁區群組：2097152 個分割區的大小最高達 128 GB。可調整的磁區群組沒有磁區群組因數。
邏輯磁區	每一磁區群組的 MAXLVS 數，在標準磁區群組為 255， 若是大磁區群組，則為 511，若是可調整的磁區群組，則為 4095。

若在強制實施每一個實體磁區有 1016 個實體分割區的限制之前，您已建立了磁區群組，則除非您將磁區群組轉換為受支援的狀態，否則不會正確追蹤磁區群組中的陳舊分割區（不再包含最新的資料）。您可以使用 **chvg -t** 指令轉換磁區群組。會根據預設值選擇適當的因數值，以在磁區群組中容納最大的磁碟。

例如，如果您使用 9 GB 磁碟及 4 MB 分割區大小建立磁區群組，則此磁區群組將大約有 2250 個分割區。使用轉換因數 3 (1016 * 3 = 3048) 可讓所有的 2250 個分割區都得到正確追蹤。使用較高因數轉換標準或大磁區群組，可以包含較大的磁碟分割區（最大為 1016* 因數）。您亦可在建立磁區群組時指定較高的因數，以使用較小的分割區大小容納較大的磁碟。

這些作業會減少您可新增至磁區群組的磁碟總數。您可新增之磁碟的新最大數目將是 MAXPVS/因數。例如，對於一般磁區群組，因數 2 會將磁區群組中磁碟的最大數目降至 16 (32/2)。對於大磁區群組，因數 2 會將磁區群組中磁碟的最大數目降至 64 (128/2)。

LVM 裝置大小限制

下列限制為 LVM 架構限制。如果需要「LVM 錯誤區塊重新定位」，則 PV 大小不能大於 128 GB。如需特定儲存裝置的大小限制，請參閱儲存裝置文件。

下列大小限制適用於 64 位元核心：

原始 VG
PV 限制：1GB (PP) * 16256（PPs/PV，因數=16）= 15.9 TB

LV 限制：1GB (PP) * 32512 (PPs/VG) = 31.8 TB

大型 VG
PV 限制：1GB (PP) * 65024（PPs/PV，因數=64）= 63.5 TB

LV 限制：1GB (PP) * 130048 (PPs/VG) = 127 TB

SVG
PV 及 LV 限制：128GB (PP) * 2048K (PPs/PV) = 256 PB

下列大小適用於 32 位元核心：

所有 VG 類型
PV 限制：< 1 TB

LV 限制：< 1 TB

配置邏輯磁區儲存體

使用「邏輯磁區儲存體」，您可以鏡映磁區群組、定義邏輯磁區，以及移除執行中系統的磁碟。

鏡映磁區群組

下列範例情節說明如何鏡映一般磁區群組。

下列指示顯示如何使用「系統管理介面工具 (SMIT)」來鏡映 root 磁區群組。

（在**磁區**配置區中選取磁區群組，然後從**已選取**功能表選擇**鏡映**）。有經驗的管理者可以使用 **mirrorvg** 指令。

1. 以 root 權限，藉由使用下列 SMIT 捷徑將磁碟新增至磁區群組：

   ```
   smit extendvg
   ```

2. 鍵入下列 SMIT 捷徑，將磁區群組鏡映到新磁碟上：

   ```
   smit mirrorvg
   ```

3. 在第一個畫面中，選取用於鏡映的磁區群組。
4. 在第二個畫面中，可以定義鏡映選項或接受預設值。如需要，可使用線上輔助說明。

註：當您完成 SMIT 畫面並按一下「確定」或結束時，基礎的指令可能需要相當長的時間才能完成。時間長度受錯誤檢查、磁區群組中邏輯磁區的大小及數目，以及同步化新鏡映之邏輯磁區所需時間等因素的影響。

此時，會依您在 SMIT 畫面中所指定的選項，來鏡映邏輯磁區的所有變更。

鏡映 root 磁區群組

下列範例情節說明如何鏡映 root 磁區群組 (rootvg)。

註：鏡映 root 磁區群組需要具有進階系統管理經驗。如果未正確執行，則可能會使系統無法啟動。

在下列範例情節中，rootvg 包含在 hdisk01 中，並會對稱為 hdisk11 的磁碟執行鏡映：

1. 檢查 AIX 是否支援 hdisk11 作為開機裝置：

   ```
   bootinfo -B hdisk11
   ```

   如果此指令傳回 1 的值，則已選取的磁碟可由 AIX 來啟動。任何其他值都表示 hdisk11 不是用於 rootvg 鏡映的候選磁碟。

2. 使用下列指令來擴充 rootvg 以包含 hdisk11：

```
extendvg rootvg hdisk11
```

如果您接收到下列錯誤訊息：

```
0516-050 此磁區群組中剩餘的描述子空間不足，請嘗試
新增較小的 PV 或使用另一個磁區群組。
```

或者類似下面的訊息：

```
0516-1162 extendvg：警告，大小為 16 的「實體分割區」需要
為 hdisk11 建立 1084 個分割區。磁區群組 rootvg 的限制
是每實體磁區 1016 個實體分割區。將 chvg 指令與 -t 選項
搭配使用，以嘗試變更此磁區群組的每「實體磁區」
的實體分割區上限。
```

您有下列幾個選擇：

· 將 rootvg 鏡映到已屬於 rootvg 的空磁碟上。

· 使用較小的磁碟。

· 使用下列程序來變更 rootvg 支援的最大分割區數目：

a. 檢查目的地磁碟所需之實體分割區數目及 rootvg 目前支援之最大數目的訊息。

b. 使用 **chvg -t** 指令，將 rootvg 中目前允許的最大分割區數目（在上述範例中為 1016）與大於目的地磁碟所需實體分割區的數目（在上述範例中為 1084）相乘。例如：

```
chvg -t 2 rootvg
```

c. 在步驟 2 開始時，重新發出 **extendvg** 指令。

3. 使用精確的對映選項鏡映 rootvg，如下列指令所示：

```
mirrorvg -m rootvg hdisk11
```

當磁區群組是 rootvg 時此指令會關閉 Quorum。如果您不使用精確的對映選項，則必須驗證開機邏輯磁區的新副本 hd5 是由連續的分割區組成。

4. 使用下列指令起始設定所有的開機記錄及裝置：

```
bosboot -a
```

5. 使用下列指令起始設定開機清單：

```
bootlist -m normal hdisk01 hdisk11
```

註：

a. 雖然 **bootlist** 指令會將 hdisk11 識別為替代開機磁碟，但是它無法保證如果 hdisk01 失敗時，系統會將 hdisk11 用作開機裝置。在此狀況下，您可能必須從產品媒體開機、選取**維護**，然後重新發出 **bootlist** 指令，而不指定故障的磁碟。

b. 如果硬體模型不支援 **bootlist** 指令，您仍可鏡映 rootvg，但必須在原始磁碟無法使用時主動選取替代開機磁碟。

定義應用程式的原始邏輯磁區

原始邏輯磁區是實體與邏輯磁碟空間的區域，此區域受應用程式（如資料庫或分割區）的直接控制，而非直接受制於作業系統或檔案系統。

略過檔案系統可以增進控制應用程式（尤其是資料庫應用程式）的效能。不過，增進量取決於資料庫大小或應用程式的驅動程式等因素。

註：您必須適當地為應用程式提供新的原始邏輯磁區之字元或區塊特殊裝置檔案。當應用程式嘗試開啟、讀取及寫入等作業時，它會鏈結到此裝置檔案。

小心： 每一個邏輯磁區在第一個區塊中都有一個邏輯磁區控制區塊 (LVCB)。LVCB 的大小是磁區群組內實體磁區的區塊大小。資料會從實體磁區的第二個區塊開始。 在原始邏輯磁區中，LVCB 是未受保護的。如果應用程式改寫 LVCB，通常會使更新 LVCB 的指令失敗並產生訊息。雖然邏輯磁區可能可以繼續正確運作，且改寫也是可允許的事件，但不建議改寫 LVCB。

下列指示使用 SMIT 及指令行介面來定義原始邏輯磁區。

此作法範例情節中的資訊已使用特定版本的 AIX 進行測試。依據您的 AIX 版本及層次，所得到的結果可能大不相同。

1. 以 root 權限，尋找您能夠鍵入下列 SMIT 捷徑在其上建立原始邏輯磁區的可用實體分割區：

   ```
   smit lspv
   ```

2. 選取磁碟。
3. 接受第二個對話框（狀態）中的預設值，然後按一下**確定**。
4. 將 **FREE PPs** 欄位中的值乘以 **PP SIZE** 欄位中的值， 取得可供所選磁碟之原始邏輯磁區使用的 MB 總數。

   如果可用空間數量不足，請選取其他磁碟，直到您找到有足夠可用空間的磁碟為止。
5. 結束 SMIT。
6. 使用 **mklv** 指令建立原始邏輯磁區。

   下列指令會在使用 38 個 4 MB 實體分割區的 db2vg 磁區群組中建立名為 lvdb2003 的原始邏輯磁區：

   ```
   mklv -y lvdb2003 db2vg 38
   ```

   使用 **-y** 旗標為邏輯磁區提供名稱，不要使用系統產生的名稱。

此時，會建立原始邏輯磁區。如果您列出磁區群組的內容，則會顯示原始邏輯磁區及預設類型（即 JFS）。邏輯磁區的此類項目只是一個標籤。它不表示已經為原始邏輯磁區裝載了檔案系統。

請參閱您的應用程式指示，了解如何開啟 /dev/*rawLVname* 及如何使用此原始空間。

相關概念

配置邏輯磁區管理程式

「邏輯磁區管理程式 (LVM)」會與基本作業系統一起安裝，不需要進一步配置。不過，在 LVM 可以使用磁碟之前，必須先配置它們並將它們定義為實體磁區。

相關資訊

mklv 指令

Logical Volume Manager from A to Z: Introduction and Concepts

解除 root 磁區群組鏡映

您可以解除 root 磁區群組的鏡映。

小心： 解除 root 磁區群組鏡映需要具有資深系統管理經驗。如果執行不正確，您的系統會變得無法開機。

在下列範例情節中，root 磁區群組位於 hdisk01，而鏡映的 root 磁區群組位於 hdisk11。這個範例會移除 hdisk11 上的鏡映。不論您最後開機的磁碟為何，這個程序皆相同。

1. 使用下列指令，解除 hdisk11 上的 root 磁區群組鏡映：

   ```
   unmirrorvg rootvg hdisk11
   ```

   unmirrorvg 指令會將 Quorum 交還 root 磁區群組。
2. 使用下列指令，可以減少 root 磁區群組中的磁碟：

   ```
   reducevg rootvg hdisk11
   ```

3. 使用下列指令，可以重新起始設定剩餘磁碟的開機記錄：

   ```
   bosboot -a -d /dev/hdisk01
   ```

4. 使用下列指令，可以修改開機清單，以便從清單移除已解除鏡映的磁碟：

```
bootlist -m normal hdisk01
```

這時磁碟就會解除鏡映。

在系統仍可用時移除磁碟

下列程序說明如何使用熱抽取特性來移除磁碟，這可讓您無需關閉系統即可移除磁碟。此特性只適用於某些系統上。

當您要執行下列動作時，熱抽取非常有用：

· 基於安全或維護目的，移除個別非 rootvg 磁區群組中包含資料的磁碟。

· 將磁碟永久地從磁區群組中移除。

· 更正磁碟故障。

移除含有資料的磁碟

使用這個程序可以移除包含資料的磁碟，而無需關閉系統。

您正在移除的磁碟必須在個別的非 rootvg 磁區群組中。當您要將磁碟移至另一個系統時使用此程序。

1. 若要列出與要移除之磁碟相關的磁區群組，請鍵入：

```
smit lspv
```

輸出將如下：

```
PHYSICAL VOLUME:    hdisk2                  VOLUME GROUP:     imagesvg
PV IDENTIFIER:      00083772caa7896e        VG IDENTIFIER
0004234500004c00000000e9b5cac262

PV STATE:           active
STALE PARTITIONS:   0                       ALLOCATABLE:      yes
PP SIZE:            16 megabyte(s)          LOGICAL VOLUMES:  5
TOTAL PPs:          542 (8672 megabytes)    VG DESCRIPTORS:   2
FREE PPs:           19 (304 megabytes)      HOT SPARE:        no
USED PPs:           523 (8368 megabytes)
FREE DISTRIBUTION:  00..00..00..00..19
USED DISTRIBUTION:  109..108..108..108..90
```

磁區群組名稱列在 VOLUME GROUP 欄位中。在此範例中，磁區群組是 imagesvg。

2. 若要驗證磁碟是否在個別的非 rootvg 磁區群組中，請鍵入：

```
smit lsvg
```

然後選取與磁碟相關的磁區群組（在此範例中為 imagesvg）。輸出將如下：

```
VOLUME GROUP:   imagesvg                VG IDENTIFIER:
0004234500004c00000000e9b5cac262

VG STATE:       active                  PP SIZE:        16 megabyte(s)
VG PERMISSION:  read/write              TOTAL PPs:      542 (8672 megabytes)
MAX LVs:        256                     FREE PPs:       19 (304 megabytes)
LVs:            5                       USED PPs:       523 (8368 megabytes)
OPEN LVs:       4                       QUORUM:         2
TOTAL PVs:      1                       VG DESCRIPTORS: 2
STALE PVs:      0                       STALE PPs:      0
ACTIVE PVs:     1                       AUTO ON:        yes
MAX PPs per PV: 1016                    MAX PVs:        32
LTG size:       128 kilobyte(s)         AUTO SYNC:      no
HOT SPARE:      no
```

在此範例中，TOTAL PVs 欄位指示僅有一個與 imagesvg 相關的實體磁區。因為此磁區群組中的所有資料都包含在 hdisk2 上，所以可使用此程序來移除 hdisk2。

3. 若要卸載磁碟上邏輯磁區的任何檔案系統，請鍵入：

```
smit umountfs
```

4. 若要取消啟動磁區群組，請鍵入：

```
smit varyoffvg
```

5. 若要匯出磁區群組，請鍵入：

```
smit exportvg
```

6. 若要移除磁碟，請鍵入：

```
smit rmvdsk
```

7. 查看您要移除之磁碟的 LED 顯示器。請確定黃色 LED 已熄滅（不亮）。
8. 實際移除磁碟。如需有關移除程序的詳細資訊，請參閱機器的服務手冊。

此時，磁碟已實際地且在邏輯上從系統中移除。如果您要永久地移除此磁碟，則此程序已完成。

移除不含資料的磁碟

下列程序說明如何移除不含資料或沒有您想保留之資料的磁碟。

小心：下列程序會消除磁碟中的任何資料。

1. 若要卸載磁碟上邏輯磁區的任何檔案系統，請鍵入：

```
smit umountfs
```

2. 若要取消啟動磁區群組，請鍵入：

```
smit varyoffvg
```

3. 若要匯出磁區群組，請鍵入：

```
smit exportvg
```

4. 若要移除磁碟，請鍵入：

```
smit rmvdsk
```

5. 查看您要移除之磁碟的 LED 顯示器。請確定黃色 LED 已熄滅（不亮）。
6. 實際移除磁碟。
   如需有關移除程序的詳細資訊，請參閱機器的服務手冊。

此時，磁碟已實際地且在邏輯上從系統中移除。如果您要永久地移除此磁碟，則此程序已完成。

移除檔案系統來移除邏輯磁區

下列程序說明如何移除 JFS 或 JFS2 檔案系統、其相關的邏輯磁區、`/etc/filesystems` 檔案中的相關段落，以及選擇性地移除裝載檔案系統的裝載點（目錄）。

小心：當移除檔案系統時，會損毀指定的檔案系統及邏輯磁區中的所有資料。

如果您想要移除裝載了不同類型的檔案系統的邏輯磁區，或不含檔案系統的邏輯磁區，您可以只移除邏輯磁區。

若要透過 SMIT 移除日誌型檔案系統，請使用下列程序：

1. 使用類似下面範例的指令，以卸載常駐於邏輯磁區上的檔案系統：

```
umount /adam/usr/local
```

   註：umount 指令不可在使用中的裝置上使用。如果有任何檔案因任何理由而開啟，或如果使用者的現行目錄是在該裝置上，則裝置是在使用中。
2. 若要移除檔案系統，請鍵入下列捷徑：

```
smit rmfs
```

3.

1. 選取要移除的檔案系統名稱。
2. 前往**移除裝載點**欄位，並切換至您的喜好設定。如果選取**是**，則基本的指令亦會移除裝載檔案系統的裝載點（目錄）（如果目錄為空）。
3. 按 Enter 鍵移除檔案系統。SMIT 會提示您確認是否要移除該檔案系統。
4. 確認要移除該檔案系統。順利移除檔案系統後，SMIT 會顯示一則訊息。

此時，檔案系統、其資料及其相關邏輯磁區會完全從系統中移除。

相關工作

只移除邏輯磁區

使用這個程序，可以移除裝載不同檔案系統類型的邏輯磁區，或不含檔案系統的邏輯磁區。

只移除邏輯磁區

使用這個程序，可以移除裝載不同檔案系統類型的邏輯磁區，或不含檔案系統的邏輯磁區。

小心：移除邏輯磁區會損毀指定的檔案系統及邏輯磁區中的所有資料。

下列程序說明如何移除邏輯磁區及任何相關的檔案系統。您可以使用此程序移除非 JFS 檔案系統或不包含檔案系統的邏輯磁區。下列程序在說明如何移除邏輯磁區之後，會說明如何移除 `/etc/filesystems` 檔案中任何非 JFS 檔案系統的段落。

若要透過 SMIT 移除邏輯磁區，請使用下列程序：

1. 如果邏輯磁區不包含檔案系統，請跳至步驟 4。
2. 卸載與該邏輯磁區相關的所有檔案系統，請鍵入：

   ```
   unmount /FSname
   ```

   其中 */FSname* 是檔案系統的完整路徑名稱。

   註：

   a. 如果您嘗試 **unmount** 的檔案系統目前在使用中，則 **unmount** 指令會失敗。僅當沒有開啟檔案系統的任何檔案且該裝置上沒有使用者的現行目錄時，才會執行 **unmount** 指令。

   b. **unmount** 指令的另一個名稱是 **umount**。這兩個名稱是可交換的。
3. 若要列出您需要瞭解之檔案系統的資訊，請鍵入下列捷徑：

   ```
   smit lsfs
   ```

   下列是部分清單：

   ```
   Name            Nodename   Mount Pt         ...
   /dev/hd3        --         /tmp             ...
   /dev/locallv    --         /adam/usr/local  ...
   ```
4. 假設第二個列出的項目使用標準命名慣例，則檔案系統會命名為 `/adam/usr/local`，且邏輯磁區是 `locallv`。若要驗證此項，請鍵入下列捷徑：

   ```
   smit lslv2
   ```

   下列是部分清單：

   ```
   imagesvg:
   LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT POINT
   hd3                 jfs        4     4     1    open/syncd    /tmp
   locallv             mine       4     4     1    closed/syncd  /adam/usr/local
   ```

5. 若要移除邏輯磁區，請在指令行上鍵入下列捷徑：

   ```
   smit rmlv
   ```

6. 選取要移除的邏輯磁區名稱。
7. 前往**移除裝載點**欄位，並切換至您的喜好設定。如果選取**是**，則基本的指令亦會移除裝載檔案系統的裝載點（目錄）（如果有且目錄為空）。
8. 按 Enter 鍵即可移除邏輯磁區。SMIT 會提示您確認是否要移除該邏輯磁區。
9. 確認要移除該邏輯磁區。順利移除邏輯磁區後，SMIT 會顯示一則訊息。
10. 如果邏輯磁區上裝載了非 JFS 檔案系統，請移除該檔案系統及 `/etc/filesystems` 檔案中的相關段落，如下列範例所示：

    ```
    rmfs /adam/usr/local
    ```

    或者，您可以使用檔案系統名稱，如下所示：

    ```
    rmfs /dev/locallv
    ```

此時，會移除邏輯磁區。如果邏輯磁區包含了非 JFS 檔案系統，則該系統的段落亦會從 `/etc/filesystems` 檔案中移除。

相關工作

移除檔案系統來移除邏輯磁區
下列程序說明如何移除 JFS 或 JFS2 檔案系統、其相關的邏輯磁區、`/etc/filesystems` 檔案中的相關段落，以及選擇性地移除裝載檔案系統的裝載點（目錄）。

重新調整 *RAID* 磁區群組大小

在使用多磁碟機陣列 (RAID) 的系統上，**chvg** 及 **chpv** 指令選項提供您將磁碟新增至 RAID 群組，以及無需岔斷系統的使用或可用性，即可增加 LVM 使用之實體磁區的大小。

註：

1. 此特性在磁區群組以典型或增強並行模式啟動時無法使用。
2. 使用下列程序無法調整 rootvg 磁區群組的大小。
3. 使用下列程序無法調整具有作用中分頁空間的磁區群組大小。

啟動（轉開）磁區群組時，會自動檢查磁區群組中所有磁碟的大小。如果偵測到成長，則系統會產生一則參考訊息。

下列程序說明如何在 RAID 環境中增加磁碟：

1. 若要檢查磁碟成長及調整大小（必要的話），請鍵入下列指令：

   ```
   chvg -g VGname
   ```

   其中 *VGname* 是磁區群組名稱。此指令會檢查磁區群組中的所有磁碟。如果任何磁碟大小有所增加，則會嘗試將實體分割區新增至實體磁區。必要的話，會判斷適當的 1016 限制乘數並將該磁區群組轉換為大的磁區群組。
2. 若要關閉磁區群組的 LVM 錯誤區塊重新定位，請鍵入下列指令：

   ```
   chvg -b ny VGname
   ```

   其中 *VGname* 是磁區群組名稱。

磁區群組策略

磁碟故障是儲存體系統中最常見的硬體故障，其次是配接卡及電源供應器故障。磁碟故障防護主要與邏輯磁區的配置有關。

若要防止配接卡及電源供應器故障，請考量所有特定磁區群組的特殊硬體配置。此類配置包括兩個配接卡及每一配接卡至少一個磁碟（配接卡之間存在鏡映），以及非 quorum 磁區群組配置。此配置的其他費用不適

於所有站台或系統。 僅當高（直至最後一秒）可用性優先考慮時才建議此配置。根據配置，高可用性可以應付最新備份與現行資料項目之間發生的硬體故障。高可用性不適用於意外刪除的檔案。

建立個別磁區群組的時機
有數個理由，說明您為什麼要將實體磁區組織到與 rootvg 不同的磁區群組中的原因。

- 更安全且更容易維護。
  - 作業系統更新、重新安裝及損毀回復會更加安全，因為您可從作業系統分隔使用者檔案系統，以便在這些作業期間不會危害使用者檔案。
  - 維護更加容易，因為您可更新或重新安裝作業系統，而無需還原使用者資料。例如，在更新前，您可藉由卸載使用者定義之磁區群組的檔案系統，來將此群組從系統中移除。 使用 **varyoffvg** 指令將之取消啟動，然後使用 **exportvg** 指令匯出群組。 更新系統軟體之後，您可使用 **importvg** 指令重新引入使用者定義的磁區群組，然後重新裝載其檔案系統。
- 若為不同的實體分割區大小。同一磁區群組中所有實體磁區必須有相同的實體分割區大小。若要讓實體磁區有不同的實體分割區大小，請將每一個大小置於個別的磁區群組中。
- 需要不同的 Quorum 性質。如果您有要為其建立非 quorum 磁區群組的檔案系統，請為該資料維護個別的磁區群組；所有其他檔案系統應保留在操作於 quorum 下的磁區群組中。
- 出於安全考量。例如，您可能想在晚上移除磁區群組。
- 於系統之間切換實體磁區。如果在可從多個系統存取的配接卡上為每一個系統建立個別的磁區群組，則您可在該配接卡上存取的系統之間切換實體磁區，而無需岔斷任一正常作業（請參閱 **varyoffvg**、**exportvg**、**importvg** 及 **varyonvg** 指令）。

磁碟故障時的高可用性
用於防止磁碟故障的主要方法與邏輯磁區配置設定（如鏡映）有關。

雖然磁區群組考量是較次要的，但有重大的經濟牽連，因為它們涉及每一磁區群組的實體磁區數目：

- 只要磁碟的 Quorum (51%) 存在，Quorum 配置（預設值）就會使磁區群組處於作用中（轉開）。在大部分情況下，您至少需要三個具有鏡映副本的磁碟在磁區群組中，才能防止磁碟發生故障。
- 只要在磁碟上有一個 VGDA 可用，非 Quorum 配置就會讓磁區群組處於作用中（轉開）。使用此配置，您僅需兩個具有鏡映副本的磁碟在磁區群組中，即可防止磁碟發生故障。

當決定每一個磁區群組中的磁碟數目時，您亦必須規劃鏡映資料的空間。請記住，您僅可在同一磁區群組中的磁碟之間鏡映及移動資料。如果站台使用大型檔案系統，則尋找在其上鏡映的磁碟空間之後可能會成為問題。請注意邏輯磁區副本之跨磁碟設定與邏輯磁區之磁碟內部配置可用性的牽連。

配接卡或電源供應器故障時的高可用性
若要防止配接卡或電源供應器發生故障，請根據您的需求來執行下列其中一項或多項動作。

- 使用位於相同或不同底座中的兩個配接卡。如果一個底座中有電源供應器故障，則將配接卡置於不同的底座中可避免失去兩個配接卡。
- 使用兩個配接卡，將每一個配接卡至少連接一個磁碟。這會藉由仍然在磁區群組中保有 Quorum 來防止任一配接卡（或電源供應器，如果配接卡在不同的櫃子中）發生故障， 這是假設磁碟 A（配接卡 A）上的邏輯磁區與磁碟 B（配接卡 B）上的邏輯磁區之間為交互鏡映（邏輯分割區的副本不可共用同一實體磁區）。這表示您將連接到配接卡 A 之磁碟上的邏輯磁區複製到位於配接卡 B 上的磁碟，且您亦將連接到配接卡 B 之磁碟上的邏輯磁區複製到位於配接卡 A 上的磁碟。
- 從兩個配接卡將所有磁碟配置到同一磁區群組。這會確保萬一配接卡發生故障（或者萬一電源供應器發生故障，如果櫃子是分開的），至少一個邏輯磁區副本仍然完整。
- 讓磁區群組成為非 Quorum 磁區群組。只要一個「磁區群組描述子區域 (VGDA)」在磁區群組中的任何磁碟上是可存取的，這就會允許磁區群組仍處於作用中。
- 若磁區群組中有兩個磁碟，則會在配接卡之間施行交互鏡映。如果在每一個配接卡上都有多個磁碟可使用，則施行雙倍鏡映。在該狀況下，您會在使用同一配接卡的磁碟及使用不同配接卡的磁碟上建立鏡映的副本。

相關概念
將磁區群組轉換為非 Quorum 狀態
您可以將磁區群組變更為非 Quorum 狀態，以便即使沒有 Quorum，仍可以繼續使用資料。

邏輯磁區策略

這裡說明的原則有助於您擬定邏輯磁區的使用策略，從而找到一種適合於您的站台之可用性、效能及成本的組合。

可用性是存取資料的能力，即使其相關的磁碟已失敗或不可存取。資料可透過資料的副本進行存取，這些副本是於正常的系統作業期間，在單獨的磁碟及配接卡上 製作並維護的。諸如鏡映及使用緊急備用磁碟之類的技術有助於確保資料可用性。

效能是存取資料的平均速度。寫入驗證及鏡映等原則可以增強可用性，但是會增加系統處理負荷，因而降低效能。鏡映會使邏輯磁區的大小變成兩倍或三倍。一般而言，增加可用性會降低效能。磁碟分段可以增加效能。鏡映可以使用磁碟分段。您可偵測及補救發生於下面狀況之熱點問題：磁碟上的部分邏輯分割區具有過多的磁碟 I/O， 從而使系統效能顯著降低。

藉由控制資料在磁碟上及磁碟間的配置，您可以調整儲存體系統，以取得可能的最佳效能。請參閱 效能管理，以取得如何將儲存體系統效能增至最大的詳細資訊。

使用以下各節權衡效能、可用性及成本之間的利弊。請記得增加可用性通常會降低效能，反之亦然。不過，如果 LVM 選擇最不繁忙磁碟上的副本進行「讀取」，則鏡映可能會增加效能。

> **附註：**鏡映不會避免因軟體問題而意外刪除或遺失個別檔案所造成的損失。 這些檔案僅可透過傳統的磁帶或磁碟備份得到還原。

鏡映或分段的需求

請判斷儲存於邏輯磁區的資料是否物有所值，可以抵得上鏡映的處理程序及磁碟空間的花費。如果您有效能敏感的大型循序存取檔案系統，則您可能要考慮磁碟分段。

效能與鏡映也不總是對立的。如果邏輯分割區的不同案例（副本）位於不同的實體磁區上（最好是連接到不同的配接卡），則 LVM 可以讀取最不繁忙磁碟上的副本而增進「讀取」效能。「寫入」效能則不然，因為您必須更新所有的副本，所以「寫入」效能總是花費相同的成本（除非磁碟連接到不同的配接卡）。而對於「讀取」作業，僅讀取一個副本即可。

AIX LVM 支援下列 RAID 選項：

表 *2.* 邏輯磁區管理程式支援 *RAID*

項目	**說明**
RAID 0	分段
RAID 1	鏡映
RAID 10 或 0+1	鏡映與分段

雖然鏡映可以增進儲存體系統的可用性，但是這並不表示可以用它替代慣用磁帶備份安排。

您可以鏡映 rootvg，但若您如此做的話，請建立個別的傾出邏輯磁區。 傾出至鏡映邏輯磁區會導致不一致的傾出。而且，由於預設傾出裝置為 主要分頁邏輯磁區，所以若要鏡映分頁邏輯磁區，請建立個別的傾出邏輯磁區。

通常，無論何時更新邏輯分割區上的資料，都會自動更新包含該邏輯分割區的所有實體分割區。 然而，因系統故障或因實體磁區於更新時無法使用，實體分割區可能變得過時（不再包含最新資料）。LVM 可以藉由將現行資料從保持最新的實體分割區複製到過時的分割區，將過時的分割區重新整理成一致狀態。此處理程序稱為鏡映同步化。系統重新啟動、實體磁區返回線上狀態或您發出 **syncvg** 指令時，可能發生重新整理。

若要影響開機邏輯磁區之實體分割區結構的任何變更發生作用，您需於變更後執行 **bosboot** 指令。這表示像變更開機邏輯磁區鏡映這樣的動作需要 **bosboot**。

鏡映寫入磁碟的排程法原則

對於僅有一個實體副本的資料，邏輯磁區裝置驅動程式 (LVDD) 會將邏輯「讀取」或「寫入」要求位址轉換為實體位址，並呼叫適當的實體裝置驅動程式來服務該要求。此單一副本或非鏡映原則會為「寫入」要求處理錯誤的區塊重新定位，並將所有的「讀取」錯誤傳回呼叫處理程序。

如果您使用鏡映邏輯磁區，則下列寫入磁碟的排程法原則，可以設定給具多個備份的邏輯磁區：

循序排程法原則
依次執行對多個副本或鏡映的「寫入」。代表單一邏輯分割區鏡映副本的多個實體分割區被指定為主要、次要及第三位。在循序排程法中，會依次寫入實體分割區。系統會等待完成一個實體分割區的「寫入」作業後，再開始下一個的「寫入」作業。當完成對所有鏡映的所有寫入作業後，「寫入」作業才算完成。

平行排程法原則
同時啟動邏輯分割區中所有實體分割區的「寫入」作業。當花費時間最長的實體分割區的「寫入」作業完成時，「寫入」作業即告完成。 使用平行排程法原則來指定鏡映邏輯磁區可能會增進 I/O 讀取作業效能， 這是因為多個副本可讓系統直接對此邏輯磁區之最不繁忙的磁碟進行讀取作業。

平行寫入循序讀取排程法原則
同時啟動邏輯分割區中所有實體分割區的「寫入」作業。讀取的主要副本總是最先讀取。如果未順利完成該「讀取」作業，則會繼續讀取下一個副本。在對下一個副本重試「讀取」作業期間，LVM 會透過硬體重新定位，更正失敗的主要副本。這樣可以修補錯誤區塊，供將來存取。

平行寫入循環讀取排程法原則
同時啟動邏輯分割區中所有實體分割區的「寫入」作業。「讀取」作業會在鏡映副本之間來回切換。

錯誤區塊原則
指出是否要啟用磁區群組以進行錯誤區塊重新定位。預設值是 *yes*。針對磁區群組將此值設為 yes 時，即可重新定位錯誤區塊。當此值設為 no 時，此原則會置換邏輯磁區設定。變更此值時，所有的邏輯磁區會以之前的設定繼續執行。此值指出所要求的 I/O 是否必須直接導向至重新定位的區塊。如果值設為 yes，磁區群組便允許執行錯誤區塊 重新定位。如果值設為 no，錯誤區塊重新定位便不會完成。LVM 只會在硬體重新定位失敗時，才執行軟體重新定位。否則，LVM 錯誤區塊重新配置 (BBR) 旗標將無作用。

註 : 除非磁區群組及邏輯磁區的錯誤區塊原則設定都設為 *yes*，否則會停用錯誤區塊重新定位。

邏輯磁區的鏡映寫入一致性原則
當啟用 (ON)「鏡映寫入一致性 (MWC)」時，會識別系統或磁區群組未適當關機時可能不一致的邏輯分割區。當將磁區群組轉接回線上時，會使用此資訊來使邏輯分割區一致。 此稱為主動 *MWC*。

當邏輯磁區是在使用主動 MWC 時，此邏輯磁區的要求會保留在排程層內，直到可以在目標實體磁區上更新 MWC 快取記憶體區塊為止。在已更新 MWC 快取記憶體區塊後，該要求會繼續進行實體資料「寫入」作業。在可以進行「寫入」作業前，僅有資料實際所在的磁碟才必須寫入 MWC 快取記憶體區塊。

當使用主動 MWC 時，系統效能可能會受到不利的影響。不利影響的原因為「寫入」所要求之記載或日誌登載（於此期間「邏輯追蹤群組 (LTG)」處於作用中）的額外負擔。磁區群組可使用的 LTG 大小為 128 K、256 K、512 K、1024 K、2 MB、4 MB、8 MB 及 16 MB。

註 : 若要使 LTG 大小大於 128 K，磁區群組中所包含的磁碟必須支援來自磁碟的策略常式之此大小的 I/O 要求。LTG 是包含於邏輯磁區的連續區塊，以 LTG 的大小排列。此額外負擔僅適用於鏡映「寫入」。

僅當在完成對所有鏡映的「寫入」前系統或磁區群組損毀時，才有必要保證資料在鏡映間的一致性。磁區群組中的所有邏輯磁區都共用 MWC 日誌。MWC 日誌在每一磁碟的外邊維護。將使用「主動 MWC」的邏輯磁區定位到磁碟的外邊，以便邏輯磁區可以在磁碟上 MWC 日誌的鄰近。

當將 MWC 設為被動時，磁區群組會記載已經開啟了邏輯磁區。在轉開磁區群組時發生損毀後，會啟動邏輯磁區的自動強制同步。藉由使用讀取回復原則的副本，將正在讀取的區塊傳達到邏輯磁區中的其他鏡映，可以在進行強制同步的同時，維護一致性。僅在磁區群組類型 BIG 上支援此原則。

當停用 (OFF) MWC 時，若發生系統或磁區群組損毀，則可將鏡映邏輯磁區的鏡映保持在不一致的狀態。不會有鏡映一致性的自動保護。在損毀時未執行的「寫入」可使鏡映在下一次轉開磁區群組時具有不一致的資料。 損毀後，停用 (OFF) MWC 的任何鏡映邏輯磁區應當先執行強制同步，然後再使用邏輯磁區中的資料。例如：

```
syncvg -f -l LTV name
```

強制同步的例外情況是僅當邏輯磁區開啟時內容才有效的邏輯磁區，如分頁空間。

就「寫入」而言，鏡映邏輯磁區與非鏡映邏輯磁區相同。當 LVM 完全完成「寫入」要求時，資料已經寫入 LVM 下的所有磁碟機。在 LVM 對「寫入」發出 iodone 之前，「寫入」的結果不明。完成此項作業之後，無需在損毀後進行回復。 機器損毀時，應檢查尚未完成寫入 (iodone) 的任何區塊並重新進行寫入作業（無論 MWC 設定為何或它們是否鏡映）。

因為鏡映邏輯磁區與非鏡映邏輯磁區相同，所以並無最新資料。無論邏輯磁區是否已鏡映，所有關心資料有效性的應用程式都需要判斷：磁區群組或系統損毀前，未執行或未完成的進行中寫入之資料的有效性。

僅當磁區群組在損毀後重新轉接回線上時，主動及被動 MWC 才會挑選一個鏡映並將該資料傳達給其他鏡映，從而使鏡映保持一致。這些 MWC 原則不追蹤最新資料。 主動 MWC 僅會追蹤目前正在寫入的 LTG，因此，MWC 不保證會將最新資料傳達給所有的鏡映。損毀後，被動 MWC 會進入讀取時傳達模式，以使鏡映保持一致。LVM 上的應用程式必須在損毀後判斷資料的有效性。 從 LVM 方面而言，若應用程式總是於損毀時重新發出所有未執行的「寫入」，則於完成這些「寫入」後，可能不一致的鏡映會變得一致（只要在損毀後，寫入損毀時未執行的相同區塊）。

註：包含 JFS 日誌或檔案系統的鏡映邏輯磁區必須於損毀之後進行同步化，方法可為：使用之前強制進行同步化，或者啟用 MWC 或啟用被動 MWC。

跨磁碟間配置原則

跨磁碟間配置原則可指定邏輯磁區的實體分割區所在之磁碟的數目。

邏輯磁區的實體分割區可能位於單一磁碟，或分佈於磁區群組的所有磁碟上。 可以使用 **mklv** 及 **chlv** 指令之下列選項，以決定磁碟間原則：

- Range 選項可以決定用於邏輯磁區單一實體副本的磁碟數目。
- Strict 選項可以決定若兩個或更多個副本必須佔用同一實體磁區時， **mklv** 作業是否可以順利完成。
- Super Strict 選項指定配置給一個鏡映的分割區不可與其他鏡映的分割區共用實體磁區。
- 已分段的邏輯磁區僅可有最大的範圍及上層限制磁碟間原則。

單一邏輯磁區副本的跨磁碟設定

如果您選取最小的跨磁碟設定 (`Range = minimum`)，則指派給邏輯磁區的所有實體分割區都會位於單一磁碟內，以增強可用性。 若您選取最大的磁碟間設定 (`Range = maximum`)，則實體分割區會位於多個磁碟上，以增強效能。

對於非鏡映邏輯磁區，請使用最小設定，以提供最大的可用性（在硬體故障的情況下存取資料）。 最小設定指出若可能的話，一個實體磁區會包含此邏輯磁區的所有原始實體分割區。如果配置程式必須使用兩個或更多個實體磁區，則它會在保持與其他參數一致的同時，使用最小的數目。

使用最小數目的實體磁區，可減少因磁碟故障而遺失資料的風險。每一個用於單一實體副本的額外實體磁區都會增加該風險。因一個實體磁區故障而使資料遺失的風險，在分佈於四個實體磁區的非鏡映邏輯磁區上，很可能會是包含在一個實體磁區上之邏輯磁區的四倍。

下圖說明最小跨磁碟間配置原則。

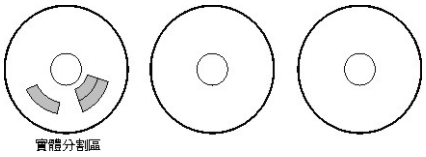


圖 *3.* 最小跨磁碟間配置原則

此圖顯示三個磁碟。一個磁碟包含三個實體分割區；其他兩個沒有實體分割區。

考慮到其他限制，最大設定會在盡可能多的實體磁區上，盡可能平均地分佈邏輯磁區的實體分割區。這種選擇是基於效能的考慮，因為將實體分割區分佈於多個磁碟，易於降低邏輯磁區的平均存取時間。為了增進可用性，最大設定僅能用於鏡映邏輯磁區。

下圖說明最大跨磁碟間配置原則。

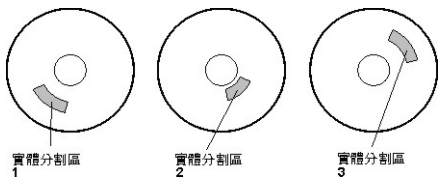


圖 *4.* 最大跨磁碟間配置原則

此圖顯示三個磁碟，每一個都包含一個實體分割區。

當擴充或複製現有邏輯磁區時，這些定義亦適用。新實體分割區的配置由您現行配置原則以及所使用之現有實體分割區的位置決定。

相關概念

邏輯磁區副本的跨磁碟設定
磁碟上邏輯磁區單一副本的配置是簡單明瞭的。

邏輯磁區副本的跨磁碟設定

磁碟上邏輯磁區單一副本的配置是簡單明瞭的。

不過，當您建立鏡映副本時，所產生的配置還是有些複雜的。 下列各圖顯示邏輯磁區第一個案例之最小、最大及磁碟間 (Range) 設定，及鏡映邏輯磁區副本的可用 Strict 設定。

例如，如果存在邏輯磁區的鏡映副本，則最小設定會導致將包含邏輯磁區第一個案例的實體分割區被配置到單一實體磁區（如果可能的話）。然後，依據 Strict 選項的設定，會將額外的一或多個副本配置到相同或個別的實體磁區。換言之，在其他參數（如 Strict 選項）強制施行的限制中，演算法會盡可能使用最少數目的實體磁區，來保留所有的實體分割區。

設定 `Strict = y` 表示邏輯分割區的每一個副本都位於不同的實體磁區。設定 `Strict = n` 表示不將副本限制到不同的實體磁區。 比較起來，`Super Strict` 選項不允許一個鏡映的任何實體分割區與同一邏輯磁區之另一個鏡映的實體分割區在相同的磁碟上。

註： 如果磁區群組中的實體磁區數目少於您選取的每一邏輯分割區副本的數目，請將 Strict 設為 **n**。如果 Strict 設為 **y**，當嘗試建立邏輯磁區時，會傳回錯誤訊息。

下圖說明具有不同 Strict 設定的最小跨磁碟間配置原則：

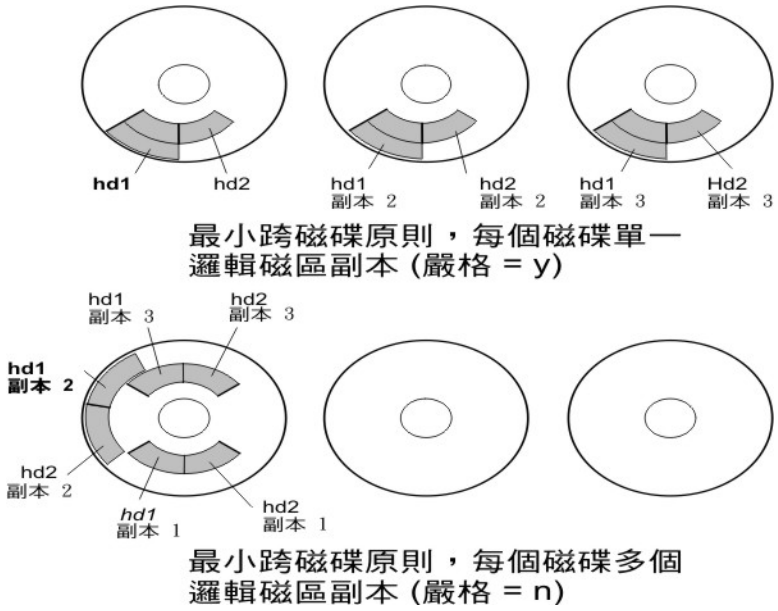


圖 *5.* 最小跨磁碟間原則/*Strict*

下圖說明具有不同 *Strict* 設定的最大跨磁碟間配置原則：

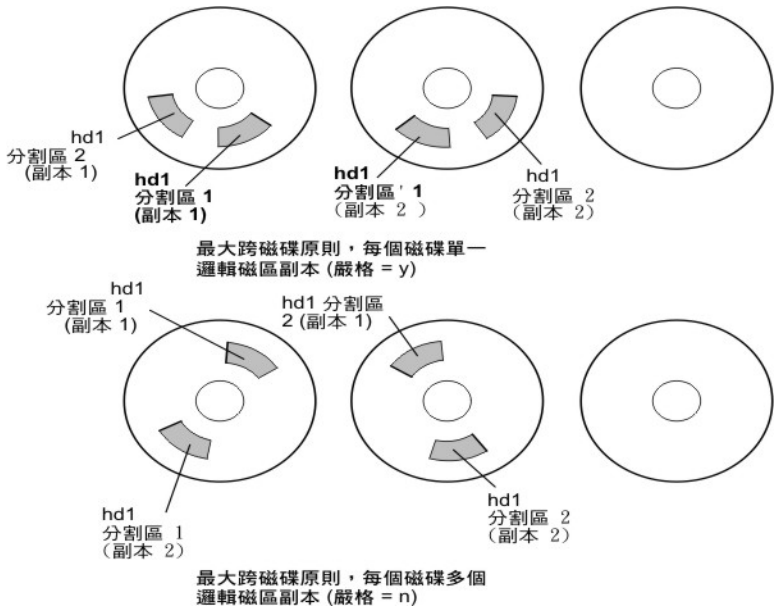


圖 *6.* 最大跨磁碟間原則/*Strict*

相關概念

單一邏輯磁區副本的跨磁碟設定

如果您選取最小的跨磁碟設定 (`Range = minimum`)，則指派給邏輯磁區的所有實體分割區都會位於單一磁碟內，以增強可用性。 若您選取最大的磁碟間設定 (`Range = maximum`)，則實體分割區會位於多個磁碟上，以增強效能。

每一個邏輯磁區的磁碟內部配置原則

磁碟內部配置原則的選擇基於磁碟上實體分割區可位於的五個範圍。

給定的實體分割區離實體磁區的中央越近，則平均探查時間越短，因為中央與磁碟的其他任何部分的平均探查距離最短。

由於作業系統經常使用檔案系統日誌，所以最好選擇在實體磁區的中央配置檔案系統日誌。相反地，開機邏輯磁區很少用到，因此最好將它配置在實體磁區的邊緣或中間。

一般規則是 I/O 越多（絕對地或在執行重要應用程式期間），越需要將邏輯磁區的實體分割區配置得靠近實體磁區的中央。

此規則有一個很重要的例外情況：啟用 (On)「鏡映寫入一致性 (MWC)」的鏡映邏輯磁區位於外邊，因為系統在此處寫入 MWC 資料。如果鏡映未起作用，則 MWC 不會應用，且不會影響效能。

實體分割區所在的五個區域如下：

1. 外邊
2. 內邊
3. 中間外部
4. 中間內部
5. 中央

邊緣分割區的平均探查時間最慢，這通常會導致使用它們之所有應用程式的回應時間較長。中央分割區的平均探查時間最快，這通常會使得使用它們之所有應用程式有最佳回應時間。不過，在中央實體磁區上的分割區少於其他範圍。

合併配置原則

如果您選擇的跨磁碟間及磁碟內部原則不相容，則可能會得到無法預期的結果。

系統會讓一個原則優先於另一個原則，來指派實體分割區。例如，如果您選擇中央的磁碟內部原則及最小的跨磁碟間原則，則跨磁碟間原則會優先於磁碟內部原則。系統會盡可能在一個磁碟上放置邏輯磁區的所有分割區，即使這些分割區不是都適合中央範圍。在施行所選原則之前，請確定您瞭解它們之間的相互作用。

使用對映檔進行精確的配置

如果中間或磁碟內部原則提供的預設選項不足以滿足您的需求，請考慮建立對映檔，來為邏輯磁區指定實體分割區的精確順序及位置。

您可以使用 SMIT 或 **mklv -m** 指令來建立對映檔。

例如，若要在 hdisk1 的分割區 1 至 3、41 至 45 及 50 至 60 的 rootvg 中建立具有 10 個分割區的邏輯磁區 lv06，您應從指令行使用下列程序。

1. 若要驗證您計劃使用的實體分割區確實可以自由配置，請鍵入：

   ```
   lspv -p hdisk1
   ```

2. 建立檔案（如 /tmp/mymap1），包含：

   ```
   hdisk1:1-3
   hdisk1:41-45
   hdisk1:50-60
   ```

   mklv 指令會以實體分割區在對映檔中出現的次序來配置它們。請確定在對映檔中確實有足夠的實體分割區可以配置您使用 **mklv** 指令所指定的整個邏輯磁區。（您可以列出比需要還多的實體分割區。）

3. 鍵入指令：

   ```
   mklv -t jfs -y lv06 -m /tmp/mymap1 rootvg 10
   ```

開發分段的邏輯磁區策略

分段的邏輯磁區用於經常被存取且效能敏感的大型循序檔案系統。分段應可增進效能。

註： 無法分段傾出空間或開機邏輯磁區。開機邏輯磁區必須是連續的實體分割區。

若要在 VGName 中，於 hdisk1、hdisk2 及 hdisk3 上，以分段大小（分段大小乘以陣列中的磁碟數等於分段大小）16 KB 建立具有 12 個分割區的分段邏輯磁區 lv07，請鍵入：

```
mklv -y lv07 -S 16K VGName 12 hdisk1 hdisk2 hdisk3
```

若要在 VGName 中，於 VGName 內的任何三個磁碟上，以分段大小 8 KB 建立具有 12 個分割區的分段邏輯磁區 lv08，請鍵入：

```
mklv -y lv08 -S 8K -u 3 VGName 12
```

請參閱效能管理，以取得有關如何使用磁碟分段增進效能的進一步資訊。

寫入驗證原則

使用寫入驗證選項，會使所有「寫入」作業由緊隨的「讀取」作業進行驗證，以檢查「寫入」作業是否順利完成。

如果「寫入」作業未順利完成，則您會收到一則錯誤訊息。此原則可以增強可用性，但是由於需要額外的時間進行「讀取」，所以會降低效能。 您可使用 **mklv** 指令建立邏輯磁區時，或於稍後使用 **chlv** 指令變更邏輯磁區時，在邏輯磁區上指定寫入驗證原則的用法。

緊急備用磁碟原則

您可以為包含鏡映邏輯磁區的磁區群組，指定磁碟作為緊急備用磁碟。

當您指定將哪些磁碟作為緊急備用磁碟時，若有一或多個磁碟開始失敗，則您可以指定使用的原則，還可以指定同步化性質。

若您向磁區群組新增實體磁區（將其標示為緊急備用磁碟），則該磁碟的容量必須至少與磁區群組中現有的最小磁碟相同。 施行此特性時，若「鏡映寫入一致性 (MWC)」寫入失敗標示為實體磁區遺失，則資料將移轉至緊急備用磁碟。

啟用緊急備用磁碟支援的指令 **chvg** 及 **chpv**， 提供了如何在您的站台實作此功能的數個選項，如下面的語法所示：

```
chvg -hhotsparepolicy -ssyncpolicy VolumeGroup
```

其中 *hotsparepolicy* 判斷磁碟失敗時，您要使用下列這些原則中的哪一個：

y
自動地將分割區從失敗的磁碟移轉到備用磁碟。於緊急備用磁碟的儲存區中， 將使用大小足夠替代失敗磁碟的最小磁碟。

Y
自動地從失敗的磁碟移轉分割區，但可能使用整個儲存區的緊急備用磁碟。

n
不自動地移轉（預設）。

r
從此磁區群組的緊急備用磁碟儲存區移除所有磁碟。

syncpolicy 引數判斷您是否想要自動同步化任何陳舊的分割區：

y
自動嘗試同步化陳舊的分割區。

n
不自動嘗試同步化陳舊的分割區。（此選項為預設選項。）

VolumeGroup 引數指定相關鏡映磁區群組的名稱。

邏輯磁區中的熱點管理
您可以識別邏輯磁區的熱點問題，而且無需岔斷系統的使用，就可以補救那些問題。

當您磁碟上的部分邏輯分割區具有過多的磁碟 I/O，而使系統效能顯著降低時，會發生熱點 問題。

解決問題的第一步為識別問題。根據預設值，系統不會收集邏輯磁區使用的統計資料。啟用這些統計資料的收集後，您第一次輸入 **lvmstat** 指令時，系統會顯示上次系統重新開機以來的計數器值。此後，每當您輸入 **lvmstat** 指令時，系統都會顯示自上次輸入 **lvmstat** 指令以來的差異。

藉由解譯 **lvmstat** 指令的輸出，您可識別流量最大的邏輯分割區。若一個實體磁碟上有數個邏輯分割區使用頻繁，而您想要在可用的磁碟之間維持平衡，可以使用 **migratelp** 指令，將這些邏輯分割區移至其他實體磁碟。

在於下列範例中，已啟用收集統計資料，並重複使用 **lvmstat** 指令來收集統計資料的基線：

```
# lvmstat -v rootvg -e
# lvmstat -v rootvg -C
# lvmstat -v rootvg
```

輸出類似下列：

```
Logical Volume            iocnt      Kb_read     Kb_wrtn      Kbps
    hd8                       4            0          16      0.00
    paging01                  0            0           0      0.00
    lv01                      0            0           0      0.00
    hd1                       0            0           0      0.00
    hd3                       0            0           0      0.00
    hd9var                    0            0           0      0.00
    hd2                       0            0           0      0.00
    hd4                       0            0           0      0.00
    hd6                       0            0           0      0.00
    hd5                       0            0           0      0.00
```

前面的輸出顯示所有計數器皆已重設為零。於下列範例中， 先將資料從 /unix 目錄複製到 /tmp 目錄。**lvmstat** 指令輸出反映 rootvg 的活動：

```
# cp -p /unix /tmp
# lvmstat -v rootvg

Logical Volume            iocnt      Kb_read     Kb_wrtn      Kbps
```

```
    hd3                     296           0        6916      0.04
    hd8                      47           0         188      0.00
    hd4                      29           0         128      0.00
    hd2                      16           0          72      0.00
    paging01                  0           0           0      0.00
    lv01                      0           0           0      0.00
    hd1                       0           0           0      0.00
    hd9var                    0           0           0      0.00
    hd6                       0           0           0      0.00
    hd5                       0           0           0      0.00
```

該輸出顯示了 **hd3** 邏輯磁區（裝載於 /tmp 目錄中）、**hd8**（JFS 日誌邏輯磁區）、**hd4** (/（根目錄）)、**hd2**（/usr 目錄）及 **hd9var**（/var 目錄）上的活動。下列輸出提供 **hd3** 及 **hd2** 的詳細資料：

```
# lvmstat -l hd3

Log_part    mirror#    iocnt    Kb_read    Kb_wrtn     Kbps
       1          1      299          0       6896     0.04
       3          1        4          0         52     0.00
       2          1        0          0          0     0.00
       4          1        0          0          0     0.00
# lvmstat -l hd2
Log_part    mirror#    iocnt    Kb_read    Kb_wrtn     Kbps
       2          1        9          0         52     0.00
       3          1        9          0         36     0.00
       7          1        9          0         36     0.00
       4          1        4          0         16     0.00
       9          1        1          0          4     0.00
      14          1        1          0          4     0.00
       1          1        0          0          0     0.00
```

磁區群組的輸出提供邏輯磁區所有 I/O 活動的摘要。其分為 I/O 要求的次數 (iocnt)、 讀取及寫入的 KB（分別是 Kb_read 及 Kb_wrtn) 及每秒轉送的資料 KB (Kbps）。若您要求邏輯磁區資訊，則會接收到該資訊，但對每個邏輯分割區為分別提供的。 若您有鏡映的邏輯磁區，則會接收到每個鏡映磁區的統計資料。於先前的範例輸出中，省略了幾行無任何活動的邏輯分割區。 輸出總是依照 iocnt 直欄的遞減次序排序。

migratelp 指令使用的參數有：邏輯磁區的名稱、 邏輯分割區的號碼（如 **lvmstat** 輸出中顯示）及特定鏡映副本的選用號碼。 若省略資訊，則會使用第一個鏡映副本。您必須指定移動的目標實體磁區；除此之外，您還可指定目標實體分割區號碼。若順利完成，則輸出類似下列：

```
# migratelp hd3/1 hdisk1/109
    migratelp: Mirror copy 1 of logical partition 1 of logical volume
                hd3 migrated to physical partition 109 of hdisk1.
```

啟用熱點特性之後，可針對邏輯磁區或磁區群組執行下列動作：定義報告及統計資料、顯示統計資料、選取要移轉的邏輯分割區、指定目的地實體分割區，以及在確定變更之前驗證資訊。

實作磁區群組原則

決定要使用哪些磁區群組原則之後，藉由在指令行上鍵入 **lspv** 指令來分析現行配置。

標準配置提供單一磁區群組，它包括連接到相同磁碟配接卡及其他支援硬體的多個實體磁區。在標準配置中，組成 Quorum 磁區群組的磁碟越多，當磁碟發生故障時，Quorum 剩餘的機會就越大。在非 Quorum 群組中，兩個磁碟中最小的一個必須組成磁區群組。 若要施行磁區群組原則變更，請執行下列動作：

1. 使用 **lspv** 指令輸出可檢查已配置的及可用的實體磁區。
2. 藉由新增一或多個實體磁區來確保 Quorum。
3. 將磁區群組變更為非 Quorum 狀態
4. 僅當必要時才重新配置硬體，以確保高可用性。請參閱系統的服務手冊，以取得指示。

分頁空間及虛擬記憶體

AIX 使用虛擬記憶體以獲得比系統中實際可用之記憶體更多的記憶體。

RAM 中或磁碟上的記憶體頁由「虛擬記憶體管理程式 (VMM)」管理。 虛擬記憶體區段分割為若干單位（稱為頁面）。分頁空間為一種邏輯磁區，其具有已配置的磁碟空間， 儲存常駐於虛擬記憶體而目前未存取的資

訊。此邏輯磁區具有與分頁相同的屬性類型， 通常簡稱為分頁空間或交換空間。當系統中可用的 RAM 數量很低時，會將最近未使用的程式或資料從記憶體移至分頁空間，以釋放記憶體供其他活動使用。

分頁空間概念

分頁空間 是一種邏輯磁區，其具有已配置的磁碟空間，用來儲存位於虛擬記憶體但目前未存取的資訊。

此邏輯磁區具有與分頁相同的屬性類型， 通常簡稱為分頁空間或交換空間。當系統中可用的 RAM 數量很低時，會將最近未使用的程式或資料從記憶體移至分頁空間，以釋放記憶體供其他活動使用。

可使用其他類型的分頁空間，透過將 NFS 伺服器用於分頁空間儲存體的裝置來加以存取。對於存取此分頁空間的 NFS 用戶端，NFS 伺服器必須已經建立了一個檔案並將其匯出到該用戶端。檔案大小代表該用戶端的分頁空間大小。

所需要的分頁空間數量視系統上執行的活動類型而定。如果分頁空間不足，則可能會遺失處理程序，如果分頁空間耗盡，則可能會使系統發生緊急情況。當偵測到分頁空間不足的狀況時，請定義額外的分頁空間。

可藉由製作新分頁空間邏輯磁區或藉由增加現有分頁空間邏輯磁區的大小來定義邏輯磁區分頁空間。若要增加 NFS 分頁空間的大小，則必須在伺服器上使用正確的動作來增加位於伺服器上的檔案。

系統可使用的總分頁空間是所有作用中分頁空間邏輯磁區的大小總和。

分頁空間配置原則

PSALLOC 環境變數可判定使用哪種分頁空間配置演算法：延遲或早期。

AIX 使用兩種模式的分頁空間配置。預設值為延遲。藉由變更 *PSALLOC* 環境變數值，您可以切換到早期分頁空間配置模式， 但於進行變更之前，有數個因素需要考量。使用早期配置演算法時（在最壞的情況中），用完所有可用的分頁空間可能會損毀系統。

比較延遲與早期分頁空間配置

作業系統使用 *PSALLOC* 環境變數來決定用於記憶體及分頁空間配置的機制。

如果 *PSALLOC* 環境變數未設定、設為 `null` 或設為 `early` 以外的任何其他值，則系統會使用預設的 `deferred` 配置演算法。

`deferred` 配置演算法可協助有效使用磁碟資源，並支援喜好資源管理稀疏配置演算法的應用程式。於提出記憶體要求時，此演算法不會保留分頁空間；將延遲分頁空間的磁碟區塊配置，直到必須輸出所要求的頁面為止。 部分程式會配置大量的虛擬記憶體，然後僅使用小部分的記憶體。這類程式的範例是將稀疏向量或矩陣用作資料結構的技術應用程式。對即時的需求分頁核心（例如作業系統中的核心）而言，延遲配置演算法亦更加有效。

可能永遠不會使用此分頁空間，尤其是在具有大型實際記憶體而分頁極少的系統上。延遲演算法會延遲分頁空間的配置，直到必須輸出頁面為止，這樣才不會浪費分頁空間配置。 此延遲配置可能會導致下列狀況：延遲演算法可嘗試配置的分頁空間大於系統可用的空間。這種狀況稱為分頁空間的過度確認。

在過度確認情境中，分頁空間即將耗盡，但卻嘗試配置分頁空間磁碟區塊以輸出某個頁面，因而導致失敗。作業系統會藉由結束受分頁空間過度確認影響的處理程序，來嘗試避免整個系統失敗。會傳送 **SIGDANGER** 信號通知處理程序可用的分頁空間數量不足。如果分頁空間狀況達到更加嚴重的狀態，則會向未接收到 **SIGDANGER** 信號的所選處理程序傳送 **SIGKILL** 信號。

您可使用 *PSALLOC* 環境變數來切換到 `early` 配置演算法， 這樣會在要求記憶體時為執行處理程序配置分頁空間。如果在要求時可用的分頁空間不足，則早期配置機制會使記憶體要求失敗。

如果 *PSALLOC* 環境變數設為 `early`，則從那時起，在該環境中啟動的每個程式（不包括目前執行中的處理程序）都會在早期配置環境中執行。 於早期配置環境中，若於提出要求時無法保留足夠的分頁空間，則諸如 **malloc** 子常式及 **brk** 子常式的介面會失敗。

如果發生分頁空間不足的狀況，則不會向在早期配置環境模式中執行的處理程序傳送 **SIGKILL** 信號。

有數種不同方法，可以將 *PSALLOC* 環境變數變更為 `early`，視您所要應用變更的範圍而定

切換到早期配置環境會影響下列記憶體配置介面子常式：

- **malloc**
- **free**

- **calloc**
- **realloc**
- **brk**
- **sbrk**
- **shmget**
- **shmctl**

相關工作

為早期配置模式配置 PSALLOC 環境變數

作業系統使用 *PSALLOC* 環境變數來決定用於記憶體及分頁空間配置的機制。

早期配置模式

早期配置演算法可保證與記憶體配置要求所要求的分頁空間一樣多。這樣，系統磁碟上的正確分頁空間配置對有效作業來說很重要。

當可用的分頁空間降至某個臨界值之下時，將無法啟動新處理程序，且目前在執行中的處理程序可能無法獲得更多的記憶體。在預設延遲配置模式下執行的任何處理程序會變得非常容易被 **SIGKILL** 信號機制損壞。此外，因為作業系統核心有時需要記憶體配置，所以用完所有可用的分頁空間可能會損毀系統。

在整個系統中使用早期配置模式之前，為系統定義足夠數量的分頁空間非常重要。早期配置模式所需要的分頁空間幾乎總是大於預設延遲配置模式所需要的分頁空間。要定義多少分頁空間要視如何使用系統及執行何種程式而定。判斷系統之恰當混合的良好開端是定義比實體記憶體數量大四倍的分頁空間。

如果某些應用程式在早期配置模式下執行，則可使用極度數量的分頁空間。當應用程式在早期配置模式下執行時，AIXwindows 伺服器目前需要大於 250 MB 的分頁空間。任何應用程式所需要的分頁空間要視應用程式的寫入方式及執行方式而定。

顯示分頁空間及處理程序記憶體使用情形的所有指令及子常式都會包括早期配置模式下配置的分頁空間。**lsps** 指令使用 **-s** 旗標來顯示總分頁空間配置，包括早期配置模式下配置的分頁空間。

分頁空間預設大小

AIX 安裝架構的系統自訂階段期間，會根據下列標準判斷預設分頁空間大小。

- 分頁空間可使用超過 16 MB，但 hd6 除外，可使用超過 64 MB。
- 分頁空間可使用的空間不超過總磁碟空間的 20%。
- 若實際記憶體小於 256 MB，則分頁空間是實際記憶體的兩倍。
- 若實際記憶體大於或等於 256 MB，則分頁空間是 512 MB。

分頁空間檔案、指令及選項

/etc/swapspaces 檔案會指定分頁空間及分頁空間的屬性。

當 **mkps** 指令建立分頁空間時，分頁空間會新增至 /etc/swapspaces 檔案，而當 **rmps** 指令刪除該分頁空間時，該分頁空間即會從 /etc/swapspaces 檔案中移除。檔案中的分頁空間屬性可由 **chps -a** 指令或 **chps -c** 指令修改。繼續支援使用先前格式的檔案（其中不具有段落總和檢查大小及自動交換的屬性）。如果分頁空間大小過大，您可以使用 **chps -d** 指令，從分頁空間扣除邏輯分割區，而無需重新開機。

使用下列指令來管理分頁空間：

項目	說明
chps	變更分頁空間的屬性。
lsps	顯示分頁空間的性質。
mkps	新增其他分頁空間。建立分頁空間邏輯磁區時，**mkps** 指令使用帶有一組特定選項的 **mklv** 指令。若要建立 NFS 分頁空間，**mkps** 指令使用帶有另一組選項的 **mkdev** 指令。對於 NFS 分頁空間，**mkps** 指令需要 NFS 伺服器的主機名稱及從伺服器匯出之檔案的路徑名稱。
rmps	移除非作用中的分頁空間。

項目	說明
swapoff	取消啟動一個以上分頁空間，而不重新啟動系統。分頁空間中的資訊會移至其他作用中的分頁空間區域。這樣，使用 **rmps** 指令即可移除取消啟動的分頁空間。
swapon	啟動分頁空間。系統起始設定前期，會使用 **swapon** 指令來啟動起始分頁空間裝置。 起始設定的後期，當其他裝置可用時，會使用 **swapon** 指令來啟動額外分頁空間，以便在數個裝置之間發生分頁活動。

所有邏輯磁區分頁空間都需要使用 `paging type` 選項。

下列選項用來將邏輯磁區的分頁效能增至最大：

- 在磁碟中間配置以減少磁碟讀寫臂的行程
- 使用多個分頁空間，每個都從個別的實體磁區配置。

配置分頁空間

許多配置作業可以使用 SMIT 來執行。分頁空間及記憶體配置是由 **PSALLOC** 環境變數來控制。

新增及啟動分頁空間

若要讓系統可以使用分頁空間，您必須新增並啟動分頁空間。

分頁空間的總數量通常是由嘗試錯誤法來判斷。一個常用的準則是將 RAM 大小加倍，並使用該數字作為分頁空間目標。

使用 SMIT 介面，在指令行上鍵入下列其中一個捷徑：

- 若要列出現行分頁空間，請鍵入：`smit lsps`
- 若要新增分頁空間，請鍵入：`smit mkps`
- 若要啟動分頁空間，請鍵入：`smit swapon`

增進分頁效能

若要增進分頁效能，請使用多重分頁空間，且如果可能，將它們放在個別的實體磁區上。

然而，在相同的實體磁區上可以有多個分頁空間。雖然您可以使用多重實體磁區， 但是除非您完全瞭解系統，不然只選 rootvg 磁區群組中的那些磁碟是個不錯的辦法。

為早期配置模式配置 PSALLOC 環境變數

作業系統使用 *PSALLOC* 環境變數來決定用於記憶體及分頁空間配置的機制。

預設設定是 `late`。下列範例顯示將 *PSALLOC* 環境變數變更為 `early` 的不同方法。您選擇的方法取決於您要在多大範圍內應用變更。

- 請在 shell 指令行上鍵入下列指令：

  ```
  PSALLOC=early;export PSALLOC
  ```

  此指令會導致從該 shell 階段作業執行的所有後續指令均以早期配置模式來執行。
- 在 shell 資源檔中新增下列指令（`.shrc` 或 `.kshrc`）：

  ```
  PSALLOC=early;export PSALLOC
  ```

  此項目會導致登入階段作業中的所有處理程序（登入 shell 除外）均在早期配置模式下執行。此方法亦會使得處理程序免於受到 **SIGKILL** 信號機制的影響。
- 將 **putenv** 子常式插入程式中，以便將 *PSALLOC* 環境變數設為 `early`。使用此方法，在下一次呼叫 **exec** 子常式時，早期配置行為就會生效。

相關概念

比較延遲與早期分頁空間配置

作業系統使用 *PSALLOC* 環境變數來決定用於記憶體及分頁空間配置的機制。

變更或移除分頁空間

使用 SMIT 可以很輕易地變更分頁空間，但是移除分頁空間會有很大的風險。

可以使用下列指令行上的 SMIT 捷徑來變更分頁空間的性質：`smit chps`。

移除分頁空間的程序是較冒險的，特別是當要移除的分頁空間為預設分頁空間（如 hd6）時。移除預設分頁空間需要特殊的程序，因為開機期間，配置系統的 shell Script 已啟動這些分頁空間。若要移除其中一個預設的分頁空間，必須變更這些 Script，且必須建立新的開機映像檔。

小心： 不正確地移除預設分頁空間，會造成系統無法重新啟動。下列程序只適用於有經驗的系統管理員。

若要移除現有的分頁空間，請使用下列程序：

1. 使用 root 權限，在指令行上鍵入下列 SMIT 捷徑，來停用分頁空間：

   ```
   smit swapoff
   ```

2. 如果您要移除的分頁空間是預設傾出裝置，則必須將預設傾出裝置變更為另一個分頁空間或邏輯磁區後，才能移除分頁空間。

   若要變更預設傾出裝置，請鍵入下列指令：

   ```
   sysdumpdev -P -p /dev/new_dump_device
   ```

3. 鍵入下列捷徑來移除分頁空間：

   ```
   smit rmps
   ```

使用分頁空間配置模式程式設計介面

控制分頁空間配置模式的程式設計介面會使用 *PSALLOC* 環境變數。

若要確保應用程式總是在所希望的模式下執行（不管有沒有早期分頁空間配置），請執行下列動作：

1. 使用 **getenv** 子常式檢查 *PSALLOC* 環境變數的現行狀態。
2. 如果 *PSALLOC* 環境變數的值不是應用程式所需要的值，請使用 **setenv** 子常式改變環境變數的值。

   因為只有 **execve** 子常式才會檢查 *PSALLOC* 環境變數的狀態，所以請使用應用程式接收的同一組參數及環境來呼叫 **execve** 子常式。 當應用程式重新檢查 *PSALLOC* 環境變數的狀態並尋找正確值時， 該應用程式會正常繼續。
3. 如果 **getenv** 子常式顯示 *PSALLOC* 環境變數的現行狀態正確，就不需要修改。

   應用程式會正常繼續。

重新定位及減少 hd6 分頁空間

您可能想要減少或移動預設分頁空間，以便藉由分頁及移動到系統中較不忙碌的其他磁碟，來加強儲存體系統效能。減少或移動預設分頁也會節省 hdisk0 上的磁碟空間。

不論是移動分頁空間或減少其大小， 其原理是相同的：將分頁空間活動移至較不忙碌的磁碟。安裝預設值會在磁碟機 hdisk0 上建立分頁邏輯磁區 (hd6)，其中含有部分或全部工作中的 /（根）及 `/usr` 檔案系統。如果選擇最小的跨磁碟間配置原則，則表示所有 / 及大量的 `/usr` 都在 hdisk0 上，且將分頁空間移至較不忙碌的磁碟可大幅增進效能。即使施行了跨磁碟間配置原則最大值，且 / 及 `/usr` 已分散在多個實體磁區上，您的 hdisk2（假設三個磁碟）可能還是會包含屬於最忙碌的檔案系統的少數邏輯分割區

下列程序說明如何將 hd6 分頁空間變得更小，以及如何在相同的磁區群組中移動 hd6 分頁空間。

使 *hd6* 分頁空間變小

下列程序會使用 **chps** 指令來收縮現有的分頁空間，包括主要分頁空間及主要與次要傾出裝置。

chps 指令會呼叫 **shrinkps** Script，此 Script 可安全地收縮分頁空間，而不會導致系統處於無法開機的狀態。特別是，該 Script 會執行下列動作：

1. 在相同的磁區中建立暫時分頁空間
2. 將資訊移到該暫時空間
3. 在相同磁區中建立新的、較小的分頁空間

4. 移除舊的分頁空間

若要順利完成 **chps** 指令，必須有足夠的可用磁碟空間（未配置給任何邏輯磁區的空間）以建立暫時分頁空間。暫時分頁空間的大小等於保留頁面輸出所有舊分頁空間中頁面所需的空間數量。主要分頁空間的大小下限是 32 MB。任何其他分頁空間的大小下限是 16 MB。

註： 如果下列程序遇到 I/O 錯誤，則系統可能需要立即關機並重新開機。

1. 檢查您的邏輯磁區與分散在實體磁區上的檔案系統，請鍵入下列指令：

   ```
   lspv -l hdiskX
   ```

   其中 *hdiskX* 是實體磁區名稱。
2. 若要收縮分頁空間大小，請在指令行上鍵入下列指令：

   ```
   smit chps
   ```

註： 主要分頁空間是寫在開機記錄的程式內的。因此，當系統重新啟動時，恆會啟動主要分頁空間。**chps** 指令無法取消啟動主要分頁空間。

已指定維護作業配置的優先順序。系統檢查會造成立即拒絕收縮分頁空間。如果在建立暫時分頁空間時發生錯誤，則會導致程序結束，且系統會恢復成原始設定。其他問題可能會引起需要系統管理者介入的情況，或可能需要立即重新開機。部分錯誤可能會造成無法移除暫時分頁空間。這通常需要管理者進行非緊急處理。

小心： 如果 **shrinkps** Script 中的 **swapoff** 指令在系統後端頁面或使用者後端頁面上偵測到 I/O 錯誤，就會建議您立即關機，以避免可能發生的系統損毀。重新開機時，即會啟動暫時分頁空間，並嘗試停止並重新啟動發生 I/O 錯誤的應用程式。如果嘗試順利成功，且可以完成取消啟動 **swapoff** 指令，則可以使用 **mkps**、**swapoff** 及 **rmps** 指令手動完成壓縮程序，以建立所需大小的分頁空間及移除暫時的分頁空間。

請勿嘗試移除（使用 **rmps**）或重新啟動（使用 **chps**）在系統重新啟動前因處於 I/O 錯誤狀態而取消啟動的分頁空間。重複使用磁碟空間是有風險的，且可能會造成其他問題。

在相同的磁區群組內移動 *hd6* 分頁空間

在相同的磁區群組中，將預設分頁空間從 hdisk0 移至不同的磁碟不需要關閉及重新啟動系統。

請以 root 權限，鍵入下列指令以將預設 (hd6) 分頁空間從 hdisk0 移至 hdisk2：

```
migratepv -l hd6 hdisk0 hdisk2
```

小心： 不建議您將名稱為 hd6 的分頁空間從 rootvg 移至另一個磁區群組，因為該名稱在數個地方是寫在程式內的，包括開機處理程序的第二階段，以及從抽取式媒體開機時存取 root 磁區群組的處理程序。在開機處理程序的第二階段中，只會啟動 rootvg 中的分頁空間，且沒有任何 rootvg 中的分頁空間可以嚴重影響系統開機效能。如果您想要大部分的分頁空間是在其他磁區群組上，則 hd6 的大小愈小愈好（與實體記憶體大小相同），然後在其他磁區群組上建立較大的分頁空間。

疑難排解分頁空間

分頁空間的最常見問題是由於配置空間用完所造成的。

分頁空間的總數量通常是由嘗試錯誤法來判斷。一個常用的準則是將 RAM 大小加倍，並使用該數字作為分頁空間目標。如果分頁空間不足，則可能會遺失處理程序，如果分頁空間耗盡，則可能會使系統發生緊急情況。下列信號及錯誤資訊可以協助您監視及解決或防止分頁空間問題。

作業系統會監視可用分頁空間的區塊數，並偵測分頁空間何時不足。當可用分頁空間區塊數低於臨界值時，這稱為分頁空間警告層次，系統會藉由傳送 **SIGDANGER** 信號，將此狀況通知所有處理程序（**kprocs** 除外）。如果不足狀況繼續存在，且低於第二個臨界值，這稱為分頁空間結束層次，系統會將 **SIGKILL** 信號傳送給那些為分頁空間之主要使用者的處理程序，以及那些沒有 **SIGDANGER** 信號之信號處理程式的處理程序 （**SIGDANGER** 信號的預設動作是忽略信號）。 系統會繼續傳送 **SIGKILL** 信號，直到可用分頁空間區塊數高於分頁空間結束層次為止。

註： 若將 `low_ps_handling` 參數設為 2（在 **vmo** 指令中），並且找不到要結束 (kill) 的處理程序（沒有 **SIGDANGER** 處理程式），則系統將會傳送 **SIGKILL** 信號給最新的處理程序，此處理程序含有 **SIGDANGER** 信號的信號處理程式。

動態配置記憶體的處理程序可藉由使用 **psdanger** 子常式來監視分頁空間層次，或藉由使用特殊的配置常式，來確保有足夠的分頁空間。您可以使用 **disclaim** 子常式，來防止達到分頁空間結束層次時結束處理程序。若要執行此動作，請定義 **SIGDANGER** 信號的信號處理程式，並釋出在它們的資料與堆疊區域及共用記憶體區段中配置的記憶體及分頁空間資源。

如果您收到類似下面的錯誤訊息，請增加分頁空間：

```
INIT：分頁空間偏低！
```

或

```
您的分頁空間即將用完。也許您要儲存文件，因為
此程式（且可能是作業系統）會在分頁空間已滿時終止，
而且不會發出警告。
```

虛擬記憶體管理程式

「虛擬記憶體管理程式 (VMM)」會管理系統及其應用程式所送出的記憶體要求。

虛擬記憶體區段是被稱為頁的單位所分割； 每一頁或是位於實際實體記憶體 (RAM) 中，或是儲存在磁碟上，直到需要它為止。AIX 使用虛擬記憶體以獲得比系統中實際可用之記憶體更多的記憶體。 RAM 中或磁碟上的記憶體頁由 VMM 管理。

虛擬記憶體管理程式中的實際記憶體管理

於 AIX 中，會將虛擬記憶體區段分割為 4096 位元組的單位，稱為頁。 將實際記憶體分成 4096 位元組的頁框。

VMM 具有兩個主要功能：

· 管理頁框的配置

· 解析對目前未在 RAM（儲存於分頁空間）或不存在之虛擬記憶體頁的參照。

若要完成這些功能，VMM 維護可用頁框的可用清單。 VMM 亦使用頁置換演算法，來判斷目前位於 RAM 中的哪些虛擬記憶體頁會將其頁框重新指派給可用清單。 頁置換演算法會考量持續與工作區段的存在、重新分頁及 VMM 臨界值。

虛擬記憶體管理程式可用清單

VMM 維護用來滿足分頁錯誤的可用（未配置）頁框清單。

AIX 嘗試全時地 使用所有 RAM，除了其在可用清單上維護的少量未配置頁外。若要維護此少量的未配置頁， VMM 使用頁面輸出及頁挪用來 釋放空間並將那些頁框重新指派給可用清單。使用 VMM 的頁置換演算法，來選擇要重新指派其頁框的虛擬記憶體頁。

虛擬記憶體管理程式中的持續或工作記憶體區段

AIX 區分不同類型的記憶體區段。若要瞭解 VMM， 瞭解工作與持續區段之間的差異很重要。

持續區段在磁碟上 具有永久儲存體位置。包含資料或可執行檔程式的檔案，會對映到持續區段。 當開啟並存取 JFS 或 JFS2 檔案時，會將檔案資料複製到 RAM。 VMM 參數控制何時應將配置到持續頁之實體記憶體頁框改寫並用於儲存其他資料。

工作區段是暫時的，僅在處理程序使用它們時存在。 工作區段無永久的磁碟儲存體位置。處理程序堆疊及資料範圍會對映到工作區段及 共用程式庫文字區段。工作區段頁無法保留在實際記憶體時，亦必須佔用磁碟儲存體位置。 磁碟分頁空間用於此目的。當程式結束時，其所有的工作頁會立即被放回可用清單中。

虛擬記憶體管理程式中的工作區段及分頁空間

會在分頁空間中為 RAM 中可修改及進行頁面輸出的工作頁指定對應的介面槽。

僅當某頁需要進行頁面輸出時才使用配置的分頁空間。不過，其他頁無法使用分頁空間中的配置頁。 只要特定頁存在於虛擬記憶體中，配置頁就會保留給該特定頁使用。因為會對持續頁進行頁面輸出 至其所來自的磁碟上相同位置，所以對於位於 RAM 的持續頁，不需要配置分頁空間。

VMM 有兩種配置分頁空間的模式：早期及後期。 每當提出對工作頁的記憶體要求時，早期配置原則會保留分頁空間。僅當實際從記憶體輸出工作頁時， 後期配置原則才會指派分頁空間（這會明顯地減少系統的分頁空間基本需求）。

虛擬記憶體管理程式記憶體載入控制機能
當處理程序參照磁碟上的虛擬記憶體頁時，由於其或是已經頁面輸出，或是尚未被讀取，因此必須對參照的頁進行頁面輸入，且若可用的頁框數量較少，則這可能會導致頁面輸出一或多個頁。 使用頁置換演算法，VMM 嘗試挪用近來尚未被參考到，因而近期不可能被參考到的頁框。

順利完成的頁置換會將目前所有作用中處理程序的記憶體頁保留在 RAM 中，而頁面輸出非作用中處理程序的記憶體頁。不過，若過度確認 RAM，則會很難選擇用於頁面輸出的頁，這是因為這些頁可能會在近期由目前執行的處理程序所參考。 結果可能很快會被參考的頁仍可能得到頁面輸出，然後在實際被參考時再進行頁面輸入。 若過度確認 RAM，則會發生連續的頁面輸入及頁面輸出，稱為猛移。 當系統處於猛移時，系統花費大部分時間進行頁面輸入及輸出，而不是執行有用的指示，且所有作用中處理程序均未取得任何有意義的進展。VMM 具有記憶體載入控制演算法， 偵測何時系統處於猛移，然後嘗試更正該狀況。

檔案系統

檔案系統是檔案以及目錄的階層式結構（檔案樹）。

這種類型的結構類似顛倒的樹，根在頂端而分支在底端。這種檔案樹使用目錄來將資料及程式組織成群組，以便同時管理數個目錄及檔案。

檔案系統位於單一邏輯磁區上。每個檔案及目錄均屬於邏輯磁區內的一個檔案系統。由於其結構，所以部分作業在檔案系統上執行，比在檔案系統內每個目錄上執行更有效率。例如，可備份、移動整個檔案系統或確保其安全。您可以製作 JFS 檔案系統或 JFS2 檔案系統的時間點映像檔， 稱為 *Snapshot*。

註：每個邏輯磁區的邏輯分割區數目上限是 32,512。如需檔案系統邏輯磁區性質的詳細資訊，請參閱 **chlv** 指令。

mkfs（製作檔案系統）指令或「系統管理介面工具」(**smit** 指令）可在邏輯磁區上建立檔案系統。

必須將檔案系統裝載到目錄裝載點，才能夠存取它。裝載多個檔案系統時，會建立呈現單一檔案系統影像的目錄結構。這是一個具有單一根的階層式結構。此結構包括基本檔案系統及您建立的所有檔案系統。可使用 **mount** 指令存取本端及遠端檔案系統。這可讓您從您的系統讀取及寫入檔案系統。裝載或卸載檔案系統通常需要系統群組的成員資格。如果在 /etc/filesystems 檔案中定義了檔案系統，檔案系統可自動被裝載。可使用 **umount** 指令卸載本端或遠端檔案系統，除非使用者或處理程序正在存取該檔案系統。請參閱第 81 頁的『裝載』，以取得有關裝載檔案系統的進一步資訊。

AIX 使用的檔案系統之基本類型稱為日誌型檔案系統 *(JFS)*。此檔案系統使用資料庫日誌登載技術維護其結構一致性。這可防止在系統異常停止時損壞檔案系統。

AIX 作業系統支援多個檔案系統類型，包括日誌型檔案系統 (JFS) 及加強型日誌型檔案系統 (JFS2)。請參閱第 85 頁的『檔案系統類型』，以取得有關檔案系統類型及每一種類型性質的進一步資訊。

一些最重要的系統管理作業必須與檔案系統一起執行，特別是：

- 為邏輯磁區上的檔案系統配置空間
- 建立檔案系統
- 使檔案系統空間可供系統使用者使用
- 監視檔案系統空間使用情況
- 備份檔案系統，以防系統失效時流失資料
- 維護檔案系統在一致的狀態

這些作業應該由您的系統管理者執行。

檔案系統概念

在您可以管理及配置檔案系統之前，您需要瞭解檔案樹的基本組織及內容。

檔案樹的組織及內容
檔案樹將檔案組織成包含類似資訊的若干目錄。這種組織方式有助於遠端裝載目錄及檔案。

系統管理者可將這些目錄用作建置要素，從一或多個伺服器裝載個別目錄，為每個用戶端建構唯一的檔案樹。遠端裝載檔案及目錄，而不是將所有資訊保存在本端，這種作法有下列優點：

· 節省磁碟空間

· 允許簡單、集中的系統管理

· 提供較安全的環境

檔案樹有下列性質：

· 可以被相同硬體架構的機器共用的檔案位於 /usr 檔案系統中。

· 可變動的每個用戶端檔案（如排存檔及郵件檔）位於 /var 檔案系統中。

· 架構獨立的可共用文字檔（如援助說明頁）位於 /usr/share 目錄中。

· /（根）檔案系統包含了系統作業所需的重要檔案及目錄。例如，它包含了裝置目錄、用於系統啟動的程式，以及用於將檔案系統裝載到根檔案系統的裝載點。

· /home 檔案系統是使用者起始目錄的裝載點。

檔案系統結構

瞭解檔案系統與目錄之間的差異很重要。檔案系統是硬碟的一個區段，被配置來包含檔案。藉由在目錄上裝載檔案系統，可以存取硬碟的這一區段。裝載後的檔案系統之外觀在一般使用者看來就像任何目錄一樣。

不過，因為檔案系統與目錄在結構上有差異，所以可分別管理這些實體中的資料。

首次安裝作業系統時，會將它載入目錄結構，如下圖所示。

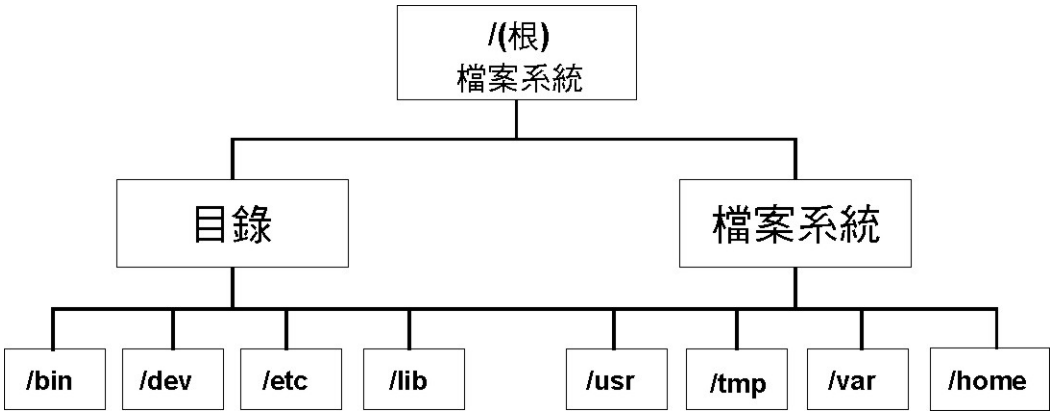


圖 7. */ (root)* 檔案系統樹

右側的目錄（/usr、/tmp、/var 及 /home）均為檔案系統，因此，會配置硬碟的個別區段供它們使用。因為這些檔案系統在系統啟動時會自動裝載，所以一般使用者看不出它們與左側所列出之目錄 （/bin、/dev、/etc 及 /lib）之間的差異。

在獨立式機器上，下列檔案系統預設是常駐於相關裝置上：

/裝置	**/檔案系統**
/dev/hd1	/home
/dev/hd2	/usr
/dev/hd3	/tmp
/dev/hd4	/ (根)
/dev/hd9var	/var
/proc	/proc
/dev/hd10opt	/opt

檔案樹有下列性質：

· 可以被相同硬體架構的機器共用的檔案位於 /usr 檔案系統中。

· 各種用戶端專屬的檔案（例如：排存和郵件檔案）位於 /var 檔案系統。

· / (根) 檔案系統包含對於系統作業而言非常重要的檔案及目錄。例如，它包含

  – 裝置目錄 (/dev)

  – 可將檔案系統裝載到根檔案系統的裝載點，例如：/mnt

· /home 檔案系統是使用者起始目錄的裝載點。

· 對伺服器而言，/export 目錄包含分頁空間檔案、用戶端專屬（不共用）根檔案系統、 傾出、起始位置和無磁碟之用戶端的 /usr/share 目錄，以及匯出的 /usr 目錄。

· /proc 檔案系統包含系統中之處理程序及執行緒狀態的相關資訊。

· /opt 檔案系統包含選用性軟體，例如：應用程式。

以下清單提供 / (根) 檔案系統一些子目錄內容的相關資訊。

項目	說明
/bin	符號鏈結至 /usr/bin 目錄。
/dev	包含本端裝置之特殊檔案的裝置節點。/dev 目錄包含磁帶機、印表機、磁碟分割區及終端機的特殊檔案。
/etc	包含隨每一部機器而不同的配置檔。範例包括： · /etc/hosts · /etc/passwd
/export	包含伺服器上給遠端用戶端之目錄以及檔案。
/home	作為包含使用者起始目錄之檔案系統的裝載點。/home 檔案系統包含每一個使用者的檔案及目錄。 在獨立式機器中，個別的本端檔案系統是裝載在 /home 目錄下。在網路中，伺服器可能會包含應該自數個機器存取的使用者檔案。 在此情形，/home 目錄的伺服器副本是經由遠端裝載至本端 /home 檔案系統。
/lib	通往 /usr/lib 目錄的符號鏈結，其包含結構獨立的檔案庫，其名稱格式為 lib*.a。
/sbin	包含啟動機器及裝載 /usr 檔案系統所需的檔案。啟動期間使用的大部分指令是來自啟動影像之 RAM 磁碟檔案系統； 因此，只有一些指令位於 /sbin 目錄。
/tmp	作為包含系統產生之暫用檔的檔案系統裝載點。
/u	通往 /home 目錄的符號鏈結。
/usr	用來作為檔案系統裝載點，包含不變更且可讓機器共用的檔案（如：可執行檔和 ASCII 文件）。 獨立式機器裝載個別的本端檔案系統於 /usr 目錄。無磁碟以及磁碟容量小的機器，都是自遠端伺服器透過 /usr 檔案系統裝載目錄。
/var	作為隨每一個機器而不同之檔案的裝載點。/var 檔案系統之所以會配置成檔案系統， 是因為它所包含的檔案很有可能會變大。例如，其為至包含暫時工作檔之 /usr/tmp 目錄的符號鏈結。

根檔案系統

根檔案系統是階層性檔案樹的頂端。其包含對系統作業而言很重要的檔案及目錄，包括啟動系統的裝置目錄及程式。根檔案系統亦包含裝載點，在那裡可裝載檔案系統以連接到根檔案系統階層。

下列圖解顯示根檔案系統的許多子目錄。

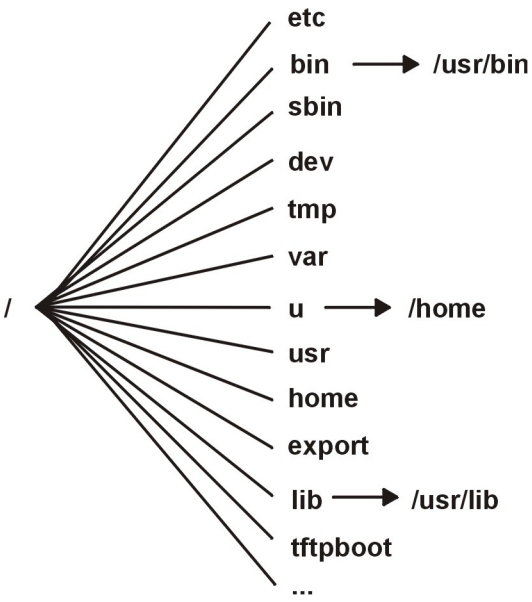


圖 *8.* 根檔案系統

下列清單提供 /（根）檔案系統之部分子目錄內容的相關資訊。

項目	說明
/etc	包含隨每一部機器而不同的配置檔。範例包括： · /etc/hosts · /etc/passwd /etc 目錄包含通常用於系統管理的檔案。 大部分先前位於 /etc 目錄中的指令目前位於 /usr/sbin 目錄。然而，基於相容性的考量， /usr/sbin 目錄包含部分可執行檔位置的符號鏈結。範例包括： · /etc/chown 是到 /usr/bin/chown 的符號鏈結。 · /etc/exportvg 是到 /usr/sbin/exportvg 的符號鏈結。
/bin	到 /usr/bin 目錄的符號鏈結。在先前的 UNIX 檔案系統中，/bin 目錄所含的使用者指令，現在位於 /usr/bin 目錄中。
/sbin	包含啟動機器及裝載 /usr 檔案系統所需的檔案。啟動期間使用的大部分指令是來自啟動影像之 RAM 磁碟檔案系統； 因此，只有一些指令位於 /sbin 目錄。
/dev	包含本端裝置之特殊檔案的裝置節點。/dev 目錄包含磁帶機、印表機、磁碟分割區及終端機的特殊檔案。
/tmp	作為包含系統產生之暫用檔的檔案系統裝載點。/tmp 檔案系統是空目錄。
/var	作為隨每一個機器而不同之檔案的裝載點。會將 /var 檔案系統配置為某個檔案系統，因為其包含的檔案似乎會一直增加。請參閱 第 63 頁的『/var 檔案系統』，以取得進一步資訊。
/u	通往 /home 目錄的符號鏈結。
/usr	包含不會變更並可由機器共用的檔案，如可執行檔及 ASCII 文件。 單機會將個別本端檔案系統的根目錄裝載到 /usr 目錄上。 無磁碟機器及具有有限磁碟資源的機器會將遠端伺服器目錄裝載到 /usr 檔案系統上。請參閱 第 61 頁的『/usr 檔案系統』，以取得裝載到 /usr 目錄上之檔案樹的進一步資訊。

項目	說明
/home	作為包含使用者起始目錄之檔案系統的裝載點。/home 檔案系統包含每一個使用者的檔案及目錄。 於單機中，/home 目錄包含於根目錄裝載到 /home 目錄根檔案系統的個別檔案系統中。在網路中，伺服器可能包含可從數台機器存取的使用者檔案。在此狀況下，/home 目錄的伺服器副本會被遠端裝載在本端 /home 檔案系統上。
/export	包含伺服器上給遠端用戶端之目錄以及檔案。 請參閱 第 63 頁的『/export 目錄』，以取得位於 /export 目錄下之檔案樹的進一步資訊。
/lib	到 /usr/lib 目錄的符號鏈結。請參閱 第 61 頁的『/usr 檔案系統』，以取得進一步資訊。
/tftpboot	包含無磁碟型用戶端的開機映像檔及開機資訊。

/usr 檔案系統

/usr 檔案系統包含可在機器之間共用的可執行檔。

/usr 目錄的主要子目錄顯示於下列圖解中。

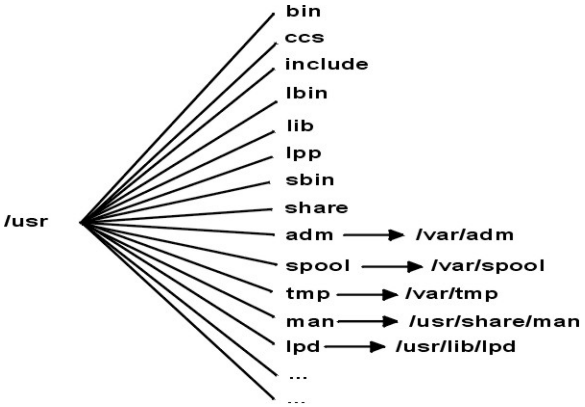


圖 *9. /usr* 檔案系統

於單機上，/usr 檔案系統為個別檔案系統（位於 /dev/hd2 邏輯磁區中）。在無磁碟機器或具有有限磁碟資源的機器上，遠端伺服器的目錄是使用唯讀許可權裝載到本端 /usr 檔案系統上。 /usr 檔案系統包含唯讀指令、程式庫及資料。

除了 /usr/share 目錄的內容之外，相同硬體架構的所有機器均可共用 /usr 檔案系統中的檔案及目錄。

/usr 檔案系統包括下列目錄：

項目	說明
/usr/bin	包含一般指令及 shell Script。例如，/usr/bin 目錄中包含 **ls**、**cat** 及 **mkdir** 指令。
/usr/ccs	包含未組合的開發套件二進位程式。
/usr/include	包含併入檔（或標頭檔）。
/usr/lbin	包含可執行檔，其為指令的後端。
/usr/lib	包含具有格式為 lib*.a 之名稱的架構獨立程式庫。/ (root) 中的 /lib 目錄為 /usr/lib 目錄的符號鏈結，因此曾經位於 /lib 目錄中的所有檔案現在都位於 /usr/lib 目錄中。出於相容性的考量，其中會包含少數非程式庫檔案。
/usr/lpp	包含選用性安裝的產品。

項目	說明
/usr/sbin	包含用於系統管理的公用程式，其中包括「系統管理介面工具 (SMIT)」指令。曾經位於 /etc 目錄中的大部分指令現在都位於 /usr/sbin 目錄中。
/usr/share	包含可在具有不同架構的機器之間共用的檔案。請參閱 第 62 頁的『/usr/share 目錄』，以取得進一步資訊。

下列是 /var 目錄的符號鏈結：

項目	說明
/usr/adm	/var/adm 目錄的符號鏈結
/usr/mail	/var/spool/mail 目錄的符號鏈結
/usr/news	/var/news 目錄的符號鏈結
/usr/preserve	/var/preserve 目錄的符號鏈結
/usr/spool	/var/spool 目錄的符號鏈結
/usr/tmp	/var/tmp 目錄的符號鏈結，因為 /usr 目錄可能是許多節點共用的，且是唯讀的

下列是 /usr/share 及 /usr/lib 目錄的符號鏈結：

項目	說明
/usr/dict	/usr/share/dict 目錄的符號鏈結
/usr/man	/usr/share/man 目錄的符號鏈結
/usr/lpd	/usr/lib/lpd 目錄的符號鏈結

/usr/share 目錄

/usr/share 目錄包含架構獨立的可共用文字檔。所有的機器都可共用此目錄的內容，無論其硬體架構為何。

於混合架構環境中，典型無磁碟型用戶端會將一個伺服器目錄裝載在其自己的 /usr 目錄上，然後將一個不同的目錄裝載在 /usr/share 目錄上。/usr/share 目錄下的檔案包含於一或多個可個別安裝的套件。這樣，當使用伺服器提供 /usr/share 目錄時，節點可能會在本端安裝它所依賴之 /usr 目錄的其他部分。

/usr/share 目錄中的部分檔案包括下列圖解中所示的目錄及檔案。

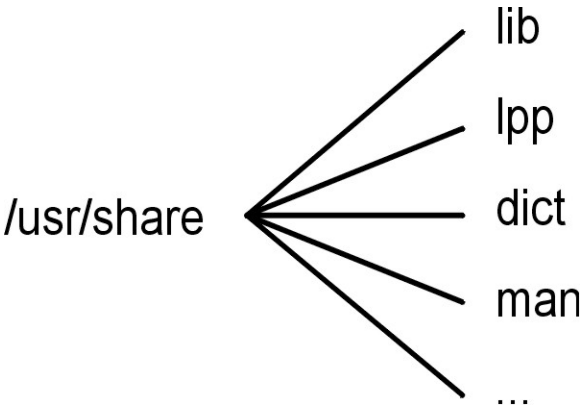


圖 *10. /usr/share* 目錄

此圖解顯示 /usr/share 目錄下的數個目錄， 包括 /lib、/lpp、/dict 及 /man。

/usr/share 目錄包括下列內容：

項目	說明
/usr/share/man	包含援助說明頁（如果已載入）
/usr/share/dict	包含拼字輔助字典及其索引
/usr/share/lib	包含架構獨立的資料檔，包括 terminfo、learn、tmac、me 及 macros
/usr/share/lpp	包含系統上可選用性安裝之產品的資料及資訊

/var 檔案系統

/var 檔案系統會逐漸變大，因為其包含工作中應用程式（如統計作業、郵件及列印排存器）所使用的子目錄及資料檔。

小心：若系統上的應用程式廣泛地使用 /var 檔案系統，請例行地執行 **skulker** 指令，或增加檔案系統的大小，使其超出 4MB /var 預設值。

確保週期性監視的特定 /var 檔案是 /var/adm/wtmp 及 /var/adm/ras/errlog。

要監視的其他 /var 檔案有：

項目	說明
/var/adm/ras/trcfile	如果啟用追蹤機能
/var/tmp/snmpd.log	若系統上正在執行 **snmpd** 指令

/var 目錄圖解顯示 /var 檔案系統中的部分目錄。

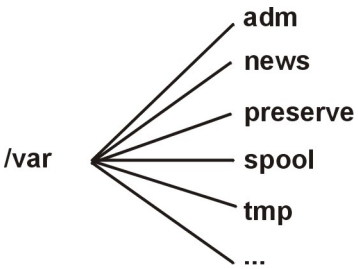


圖 *11. /var* 目錄

項目	說明
/var/adm	包含系統日誌及統計檔
/var/news	包含系統新聞
/var/preserve	包含來自岔斷編輯階段作業的保留資料；與先前版次中的 /usr/preserve 目錄類似
/var/spool	包含程式所處理的檔案（如電子郵件）；與先前版次中的 /usr/spool 目錄類似
/var/tmp	包含暫存檔；與先前版次中的 /usr/tmp 目錄類似。 /usr/tmp 目錄現為 /var/tmp 的符號鏈結。

/export 目錄

/export 目錄包含匯出到用戶端（如無磁碟、無資料或磁碟不足的機器）的伺服器檔案。

伺服器可匯出數種類型的磁碟空間，包括可執行程式套件、無磁碟型用戶端的分頁空間、無磁碟型用戶端或磁碟資源較少之用戶端的根檔案系統。檔案樹中這類磁碟空間的標準位置是 /export 目錄。/export 目錄的部分子目錄顯示在以下清單中：

/exec
包含無磁碟型用戶端裝載至其 /usr 檔案系統的目錄

/swap
包含無磁碟型用戶端之遠端分頁的檔案

/share
包含無磁碟型用戶端裝載至其 /usr/share 目錄的目錄

/root
包含無磁碟型用戶端裝載至其 / (根) 檔案系統的目錄

/home
包含無磁碟型用戶端裝載至其 /home 檔案系統的目錄

對於無磁碟型指令而言，/export 目錄是用戶端資源的預設位置。 /export 目錄僅為伺服器上用戶端資源的位置。因為用戶端將這些資源裝載到其自身的檔案樹，所以在用戶端看來，這些資源是位於檔案樹中的正常位置。 /export 目錄之主要子目錄，及用戶端檔案樹上其對應的裝載點，其中包括：

/export/root
會將此目錄裝載到用戶端根 (/) 檔案系統。用戶端根目錄預設位於 /export/root 目錄中，並以用戶端的主機名稱命名。

/export/exec
亦稱為「共用產品物件樹 (SPOT)」目錄。 會將此目錄裝載到用戶端 /usr 檔案系統。SPOT 是儲存於 /export/exec 目錄中的 /usr 檔案系統版本， 其名稱可反映其版次。根據預設值，名稱為 RISCAIX。

/export/share
會將此目錄裝載到用戶端 /usr/share 目錄。此目錄包含可由許多架構共用的資料。預設位置是 /export/share/AIX/usr/share。

/export/home
會將此目錄裝載到用戶端 /home 檔案系統。它包含依用戶端主機名稱分組的使用者目錄。 用戶端起始目錄的預設位置為 /export/home。

/export/swap
亦稱為分頁目錄。在單機或無資料型系統中，分頁由本端磁碟提供；對於無磁碟型用戶端，此服務由伺服器上的檔案提供。此檔案與用戶端的主機名稱同名。根據預設值，可在 /export/swap 目錄中找到它。

/export/dump
單機系統使用本端磁碟作為傾出裝置；無磁碟型用戶端使用伺服器上的檔案。 該檔案位於與用戶端主機名稱同名的目錄中。根據預設值，可於 /export/dump 目錄中找到它。

microcode
此目錄包含實體裝置的微碼。預設位置是 /export/exec/RISCAIX/usr/lib/microcode。

加密 JFS2 檔案系統

從 AIX 6.1 版開始，JFS2 檔案系統上就支援「加密檔案系統 (EFS)」。EFS 可讓您透過加密保護來加密資料及控制對資料的存取。

金鑰與每個使用者相關聯，並儲存在加密保護的金鑰儲存庫中。順利登入時，使用者的金鑰會載入核心，並與處理程序認證相關聯。若要開啟 EFS 保護的檔案，會測試處理程序認證。如果處理程序找到符合檔案保護的金鑰，則該處理程序會解密檔案金鑰及檔案內容。

根據預設值，JFS2 檔案系統不會啟用 EFS。在加密任何資料之前，JFS2 檔案系統必須啟用 EFS。如需如何啟用 EFS 的相關資訊，請參閱 *Commands Reference, Volume 2* 中的 **efsenable** 指令。

配置檔案系統

新增或配置檔案系統時，您可以選取 SMIT 捷徑的「檔案系統」儲存器中的選項。

SMIT 捷徑提供於下表中：

表 *3.* 管理邏輯磁區與檔案系統作業

作業	***SMIT* 捷徑**
新增 JFS 或 JFS2	**smit crfs**

表 3. 管理邏輯磁區與檔案系統作業 (繼續)	
作業	***SMIT* 捷徑**
將 JFS2 新增至現有的邏輯磁區	**smit crjfs2lvstd**
將 JFS 新增至先前定義的邏輯磁區功能表	建立邏輯磁區，然後 **smit crjfslv**
變更 JFS 或 JFS2 的屬性[附註 1]	**smit chfs**
檢查檔案系統的大小	**smit fs**
增加檔案系統的大小	JFS：**smit chjfs** JFS2：**smit chjfs2**
減少檔案系統的大小	JFS2：**smit chjfs2**

註 : 減少檔案系統大小的「SMIT 捷徑」只適用於 JFS2。

管理檔案系統

檔案系統是一個完整的目錄結構，包括根目錄及其下的任何子目錄及檔案。

檔案系統被限制在單一邏輯磁區。一些最重要的系統管理作業是有關檔案系統的，特別是：

- 為邏輯磁區上的檔案系統配置空間
- 建立檔案系統
- 使檔案系統空間可供系統使用者使用
- 監視檔案系統空間使用情況
- 備份檔案系統，以防止系統故障時遺失資料
- 製作 Snapshot，在給定的時間點擷取檔案系統的一致區塊層次映像檔
- 將檔案系統維持在一致狀態。

以下是協助管理檔案系統之系統管理指令的清單：

項目	**說明**
backup	執行檔案系統的完整或漸進式備份
chfs -a splitcopy	建立裝載 JFS 檔案系統的連線備份
dd	將資料直接從一個裝置複製到另一個裝置，以製作檔案系統備份
df	報告檔案系統上已使用及可用的空間大小
fsck	檢查檔案系統及修復不一致
mkfs	在指定邏輯磁區上製作指定大小的檔案系統
mount	將檔案系統連接至全系統命名結構，以便存取該檔案系統中的檔案及目錄
restore	從備份還原檔案
snapshot	建立 JFS2 檔案系統的 Snapshot
umount	從全系統命名結構移除檔案系統，使檔案系統內的檔案及目錄不可存取。

顯示檔案系統上的可用空間（df 指令）

使用 **df** 指令， 顯示有關檔案系統之總空間以及可用空間的相關資訊。**FileSystem** 參數指定檔案系統常駐的裝置名稱，裝載檔案系統的目錄，或檔案系統的相對路徑名稱。

如果沒有指定 **FileSystem** 參數，**df** 指令會顯示目前已裝載的所有檔案系統資訊。若已指定檔案或目錄，那麼 **df** 指令將顯示其常駐之檔案系統的相關資訊。

一般而言，**df** 指令會使用超級區塊中所包含的可用數目。在某些例外狀況下，這些計數可能錯誤。例如，如果檔案系統在 **df** 指令執行時被修改，可用數目可能就不正確。

請參閱 *Commands Reference, Volume 2* 中的 **df** 指令，以取得完整語法。

註： 在某些遠端檔案系統上（如「網路檔案系統 (NFS)」），若伺服器不提供資訊，則會將表示可用空間的直欄留白。

以下是 **df** 指令的用法範例：

· 若要顯示所有已裝載檔案系統的資訊，請鍵入：

```
df
```

如果您的系統已配置成 /、/usr、/site 和 /usr/venus 目錄常駐於個別的檔案系統中，則 則 **df** 指令的輸出會類似如下：

```
Filesystem 512-blocks  free   %used   Iused %Iused   Mounted on
/dev/hd4    20480     13780    32%      805   13%     /
/dev/hd2   385024     15772    95%    27715   28%     /usr
/dev/hd9var 40960     38988     4%      115    1%     /var
/dev/hd3    20480     18972     7%       81    1%     /tmp
/dev/hd1     4096      3724     9%       44    4%     /home
```

· 若要顯示現行目錄所在之檔案系統的可用空間，請鍵入：

```
df .
```

檔案系統指令
有一些指令是設計用來操作檔案系統（不管類型為何）。

/etc/filesystems 檔案控制著下列指令可操作之檔案系統的清單：

項目	說明
chfs	變更檔案系統的性質
crfs	新增檔案系統
lsfs	顯示檔案系統的性質
rmfs	移除檔案系統
mount	使檔案系統可供使用

有四個指令可操作虛擬檔案系統類型。/etc/vfs 檔案包含下列指令可操作之檔案系統類型的相關資訊：

項目	說明
chvfs	變更檔案系統類型的性質
crvfs	新增檔案系統類型
lsvfs	列出檔案系統類型的性質
rmvfs	移除檔案系統類型

比較不同機器上的檔案系統
如果不同機器上的檔案系統應該相同、但您懷疑其中之一已損毀，就可以比較檔案系統。

下列程序說明如何比較常駐於現行主機（在這個範例情節中，稱為 *orig_host*）與遠端主機上之相同檔案系統的屬性。

此作法範例情節中的資訊已使用特定版本的 AIX 進行測試。依據您的 AIX 版本及層次，所得到的結果可能大不相同。

1. 以 root 使用者身分登入遠端主機。
   例如：

   ```
   tn juniper.mycompany.com
   ```

```
AIX 6.1 版
(C) Copyrights by IBM and by others 1982, 2002.
login: root
root's Password:
```

2. 使用您喜好的編輯器來編輯遠端主機的 .rhosts 檔案，以新增允許使用者執行安全遠端指令的段落。新段落請使用下列格式：

```
orig_host root
```

所產生的 .rhosts 檔案可能如下：

```
NIM.mycompany.com root
nim.mycompany.com root
host.othernetwork.com root
orig_host.mycompany.com root
```

3. 儲存變更並結束遠端連接。
4. 使用 *orig_host* 上的 root 權限，以您喜好的編輯器來建立另一個檔案。對於此範例情節而言，新檔案名為 compareFS。例如：

```
vi compareFS
```

5. 在這個檔案中插入下列文字，其中 *FSname* 是您要比較的檔案系統名稱，而 *remote_host* 是所比較之檔案系統常駐的主機名稱：

```
FSname -> remote_host
install -v ;
```

註：在這個檔案的 **install** 指令行中，-v 參數與分號 (;) 之間必須有一個空格。

例如：

```
/home/jane/* -> juniper.mycompany.com
install -v ;
```

6. 儲存檔案並結束編輯器。在下列步驟中，compareFS 檔案是作為 **rdist** 指令的 *distfile*。
7. 在指令提示中鍵入下列指令：

```
/usr/bin/rdist -f compareFS
```

或者，如果您預期某個比較動作會有大量輸出，則可指定將輸出傳送到檔案。例如：

```
/usr/bin/rdist -f compareFS > compareFS_output
```

此輸出會列出檔案系統之間的任何差異。

維護檔案系統

下表對維護檔案系統時，可能需要進行的最簡單作業分組。

表 *4.* 維護檔案系統作業

作業	***SMIT* 捷徑**	**指令或檔案**
依檔案名稱或目錄進行備份	**smit backfile**	**backup** 附註 1
建立及備份 JFS2 Snapshot 映像檔	**smit backsnap**	**backsnap** 附註 1
列出磁碟上的所有檔案系統	**smit lsmntdsk**	
列出抽取式磁碟上的檔案系統	**smit lsmntdsk**	
列出裝載的檔案系統	**smit fs**	

表 4. 維護檔案系統作業 (繼續)		
作業	***SMIT* 捷徑**	**指令或檔案**
裝載一組檔案系統[附註 5]	**smit mountg**	**mount -t** *GroupName*
裝載 JFS 或 JFS2 [附註 3]	**smit mountfs**	**mount**
裝載 JFS2 Snapshot	**smit mntsnap**	**mount -v jfs2 -o snapshot** Device *MountPoint*
移除 JFS 或 JFS2	**smit rmfs**	
移除 JFS2 Snapshot	**smit rmsnap**	**snapshot -d** *SnapshotDevice*
將 JFS2 檔案系統回復至某個時間點 Snapshot	**smit rollbacksnap**	**rollback** [-s] [-v] [-c] *snappedFS snapshotObject*
卸載檔案系統[附註 4]	**smit umountfs**	
卸載抽取式磁碟上的檔案系統[附註 4]	**smit umntdsk**	
卸載一組檔案系統[附註 5]	**smit umountg**	**umount -t** *GroupName*
管理「已強化的日誌型檔案系統」限額	**smit j2fsquotas**	
啟用或停用限額管理	**smit j2enablequotas**	
停止/重新啟動強制執行限額限制	**smit j2enforcequotas**	**quotaon|off -v**
列出限額使用情況	**smit j2repquota**	**repquota -v**
重新計算現行磁碟區塊及檔案使用情況統計資料	**smit j2quotacheck**	**quotacheck -v**
新增限制類別	**smit j2addlimit**	**j2edlimit -e**
變更/顯示限制類別的性質	**smit j2changelimit**	
將限制類別設成檔案系統的預設限制	**smit j2defaultlimit**	
指派限制類別的使用者或群組	**smit j2assignlimit**	
列出檔案系統的限制類別	**smit j2listlimits**	**j2edlimit -l '-u'**
移除限制類別	**smit j2removelimit**	

註：

1. 如需選項，請參閱個別指令。
2. 請勿變更系統重要檔案系統的名稱，它們是邏輯磁區 4 (hd4) 上的 /（根）、hd2 上的 /usr、hd9var 上的 /var、hd3 上的 /tmp 及 hd5 上的 /blv。如果您使用 hd*n* 慣例，請在 hd10 上啟動。
3. 使用程序第 71 頁的『檔案系統驗證』或執行 **fsck** 指令，以在裝載之前檢查檔案系統。
4. 如果卸載失敗，可能是因為使用者或處理程序在卸載的檔案系統中開啟了某一檔案。**fuser** 指令可讓您找到是哪一個使用者或處理程序可能造成此失敗。
5. 檔案系統群組是在 /etc/filesystems 檔案中具有相同 **type=** ID 值的檔案系統集合。

從線上外部 JFS2 Snapshot 回復一個以上檔案

如果您在線上外部 JFS2 Snapshot 中有正確的副本，則可以取代損毀的檔案。

使用下列程序，可以從線上外部 JFS2 Snapshot 映像檔回復一個以上檔案。

對於此範例，假設 /home/aaa/myfile 是 /home 檔案系統中損毀的檔案。

1. 使用類似於下面所示的指令來裝載 Snapshot：

   ```
   mount -v jfs2 -o snapshot /dev/mysnaplv /tmp/mysnap
   ```

2. 使用類似於下面所示的指令，變更為包含 Snapshot 的目錄：

   ```
   cd /tmp/mysnap
   ```

3. 從 Snapshot 複製正確的檔案，以使用類似於下面所示的指令來改寫損毀的檔案：

   ```
   cp aaa/myfile /home/aaa/myfile
   ```

   前一個範例僅會複製名為 myfile 的檔案。如果您要從 Snapshot 將所有檔案都複製到 aaa 目錄，請使用類似於下面所示的指令：

   ```
   cp -R aaa /home/aaa
   ```

   如需使用 Snapshot 映像檔取代損毀檔案的更多範例，請參閱 *Commands Reference, Volume 1* 中的 **cp** 或 **cpio** 指令說明。

從線上內部 JFS2 Snapshot 回復一個以上檔案

如果您在線上內部 JFS2 Snapshot 中有正確的副本，則可以取代損毀的檔案。

使用下列程序，可以從線上內部 JFS2 Snapshot 映像檔回復一個以上檔案。

對於此範例，假設 /home/aaa/myfile 是 /home 檔案系統中損毀的檔案。

1. 使用類似於下面所示的指令，變更為包含 Snapshot 的目錄：

   ```
   cd /home/.snapshot/mysnap
   ```

2. 從 Snapshot 複製正確的檔案，以使用類似於下面所示的指令來改寫損毀的檔案：

   ```
   cp aaa/myfile /home/aaa/myfile
   ```

   前一個範例僅會複製名為 myfile 的檔案。如果您要從 Snapshot 將所有檔案都複製到 aaa 目錄，請使用類似於下面所示的指令：

   ```
   cp -R aaa /home/aaa
   ```

   如需使用 Snapshot 映像檔取代損毀檔案的更多範例，請參閱 *Commands Reference, Volume 1* 中的 **cp** 或 **cpio** 指令說明。

光碟及影碟的檔案系統

CD 及 DVD 並不會自動裝載，但可啟用此特性。

若要啟用此特性，請使用 **cdmount** 指令來裝載 CDRFS 或 UDFS 檔案系統，例如：

```
cdmount cd0
```

您可以使用下列指令手動裝載讀取/寫入 UDFS：

```
mount -V udfs DevName MtPt
```

其中 *DevName* 是 DVD 磁碟機名稱，*MtPt* 是檔案系統的裝載點。

在讀取/寫入光學媒體上使用檔案系統

您可以在讀取/寫入光學媒體上，使用 CDRFS 及 JFS 檔案系統。

如果光學媒體受到寫入保護，則光碟裝置檔案系統 (CDRFS) 可儲存在讀取/寫入光學媒體以及光碟裝置上。下表會告知您如何新增、裝載或卸載讀取/寫入光學媒體上的 CDRFS。在裝載檔案系統時，您必須指定下列資訊：

項目	說明
裝置名稱	定義含有媒體的裝置名稱。
裝載點	指定將裝載檔案系統的目錄。
自動裝載	指定是否於系統重新啟動時自動裝載檔案系統。

光學媒體作業上的 CDRFS		
作業	*SMIT* 捷徑	指令或檔案
新增 CDRFS [1]	**smit crcdrfs**	1. 新增檔案系統：**crfs -v cdrfs -p ro -d***DeviceName* **-m** *MountPoint* **-A** *AutomaticMount* 2. 裝載檔案系統：**mount** *MountPoint*
移除 CDRFS[2]	1. 卸載檔案系統：**smit umountfs** 2. 移除檔案系統：**smit rmcdrfs**	1. 卸載檔案系統：**umount** *FileSystem* 2. 移除檔案系統：**rmfs** *MountPoint*

註：

· 請確定讀取/寫入光學媒體受到寫入保護。

· 必須先卸載 CDRFS 檔案系統，然後才能移除讀取/寫入光學媒體。

JFS 提供之光學媒體上的讀取/寫入檔案系統，類似於硬碟上的那些檔案系統。您必須具有系統權限，才能在讀取/寫入光學媒體上建立或匯入讀取/寫入檔案系統（亦即，您的登入必須屬於系統群組），且您必須具有下列資訊：

磁區群組名稱
指定磁區群組的名稱

裝置名稱
指定可讀寫光碟裝置的邏輯名稱

裝載點
指定要裝載檔案系統的目錄

自動裝載
指定是否要在系統重新啟動時自動裝載檔案系統

註：

· 建立在讀取/寫入光學媒體上的任何磁區群組，本身必須內含於該媒體中。磁區群組不能超出一個讀取/寫入光碟。

· 在存取之前建立的日誌型檔案系統時，磁區群組名稱不必符合建立磁區群組時所使用的名稱。

光學媒體作業上的 JFS		
作業	*SMIT* 捷徑	指令或檔案

光學媒體作業上的 JFS		
新增 JFS	1. 將光碟插入磁碟機中。 2. 建立磁區群組（必要的話）：**smit mkvg** 3. 建立日誌型檔案系統：**smit crfs**	1. 將光碟插入磁碟機中。 2. 建立磁區群組（必要的話）：**mkvg -f -y** *VGName* **-d 1** *DeviceName* 3. 建立日誌型檔案系統：**crfs -v jfs -g** *VGName* **-a size=***SizeFileSystem* **-m** *MountPoint* **-A** *AutomaticMount* **-p rw** 4. 裝載檔案系統：**mount** *MountPoint*
存取先前建立的 JFS [附註 1]	1. 將光碟插入磁碟機中。 2. 匯入磁區群組：**smit importvg**	1. 將光碟插入磁碟機中。 2. 匯入磁區群組：**importvg -y** *VGName DeviceName* 3. 裝載檔案系統：**mount** *MountPoint*
移除 JFS [附註 2]	1. 卸載檔案系統：**smit umountfs** 2. 移除檔案系統：**smit rmjfs**	1. 卸載檔案系統：**umount** *FileSystem* 2. 移除檔案系統：**rmfs** *MountPoint*

註：

· 只要是插入內含日誌型檔案系統的媒體， 就需要使用此程序。

· 移除日誌型檔案系統會損毀內含於該檔案系統及讀取/寫入光學媒體上的所有資料。

檔案系統驗證

系統停止而檔案系統仍在裝載，或磁碟損壞時，檔案系統中可能發生不一致。在這種情況下，裝載之前先驗證檔案系統是很重要的。

在下列情況下，也要驗證您的檔案系統：

· 發生故障之後；例如，如果使用者無法將目錄變更為具有該使用者許可權 (uid) 的目錄

· 備份檔案系統之前，防止錯誤及可能的還原問題

· 在安裝或系統開機時，確定沒有任何作業系統檔錯誤

檢查使用者定義的檔案系統

若要檢查使用者定義的檔案系統，請執行下列步驟：

1. 卸載正在檢查之使用者定義的檔案系統。
2. 請確定您擁有檔案系統中檔案的寫入許可權。否則，即使您在修復提示中回答是， **fsck** 也不會修復損壞的檔案。
3. 使用 **smit fsck** 捷徑，以存取**驗證檔案系統**功能表。
4. 請執行下列其中一項：
   - 在**檔案系統名稱**欄位中指定要檢查的個別檔案系統名稱，或
   - 在**檔案系統類型**欄位中選取要檢查的一般檔案系統類型，如日誌型檔案系統 (JFS)。
5. 如果您要將檢查限制為針對最可能的檔案系統進行，請在**快速檢查？**欄位中指定是。
   快速檢查選項僅會檢查可能有不一致的那些檔案系統，如在過去某一點停止系統時裝載的那些檔案系統。

6. 在**暫用檔案**欄位中，指定檔案系統上未在檢查的暫存檔名稱。
7. 啟動檔案系統檢查。

檢查 /（根）及 */usr* 檔案系統
若要在 / 或 /usr 檔案系統上執行 **fsck** 指令，您必須關閉系統並從抽取式媒體重新開機，因為系統在執行中時， 不能卸載 /（根）及 /usr 檔案系統。

下列程序說明如何從維護 shell，在 / 及 /usr 檔案系統上執行 **fsck**。

1. 關閉系統。
   （需要 root 存取權）
2. 從安裝媒體開機。
3. 從**歡迎使用**功能表中選擇**維護**選項。
4. 從**維護**功能表中選擇選項來存取磁區群組。
5. 選擇 rootvg 磁區群組。即會顯示屬於您所選取的磁區群組的邏輯磁區清單。
6. 選擇 **2** 以存取磁區群組，並在裝載檔案系統前先啟動 shell。
   在下列步驟中，您將使用適當的選項與檔案系統裝置名稱來執行 **fsck** 指令。**fsck** 指令會檢查檔案系統一致性，並以互動方式修復檔案系統。/（根）檔案系統裝置是 /dev/hd4，且 /usr 檔案系統裝置是 /dev/hd2。
7. 若要檢查 / 檔案系統，請鍵入下列指令：

   ```
   $ fsck -y /dev/hd4
   ```

   建議較資淺的使用者使用 **-y** 旗標（請參閱 **fsck** 指令）。
8. 若要檢查 /usr 檔案系統，請鍵入：

   ```
   $ fsck -y /dev/hd2
   ```

9. 若要檢查 rootvg 中的其他檔案系統，請使用適當的裝置名稱鍵入 **fsck** 指令。/tmp 的裝置是 /dev/hd3，且 /var 的裝置是 /dev/hd9var。
10. 當您完成檔案系統的檢查後，請重新啟動系統。

減少 root 磁區群組中檔案系統的大小
將所有檔案系統大小減至最小的最簡單方法是，從備份還原基本作業系統時， 將 **SHRINK** 選項設為 **yes**。

將所有檔案系統大小減至最小的最簡單方法是，從備份還原基本作業系統時， 將 **SHRINK** 選項設為 **yes**。**SHRINK** 選項與下面的範例情節不能連貫使用。如果您在執行下面的程序後將 **SHRINK** 選項設為 **yes**，則安裝作業會置換您對 /image.data 檔案所做的變更。

此範例情節會引導您完成減少所選 rootvg 檔案系統大小的手動處理程序。您需先識別未使用其全部配置磁碟空間的檔案系統，然後根據檔案系統實際使用的空間數量重新配置，以釋出更多空間供 root 磁區群組使用。在此程序中，您將使用修改的配置備份您的磁區群組，並重新安裝作業系統。

小心：此程序需要關機並重新安裝基本作業系統。每當您重新安裝任何作業系統時，都請將停機時間排定在對工作量影響最小的期間，以免發生資料或功能遺失。在重新安裝作業系統之前，請確定您擁有資料及任何自訂應用程式或磁區群組的可靠備份。

此作法範例情節中的資訊已使用特定版本的 AIX 進行測試。依據您的 AIX 版本及層次，所得到的結果可能大不相同。

1. 建立不包含在 rootvg 中的所有檔案系統之個別備份。個別備份會協助確保所有檔案系統的完整性。
2. 以 root 權限，鍵入下列指令來檢查 root 磁區群組中未使用其配置磁碟空間的檔案系統：

   ```
   df -k
   ```

   -k 旗標會以 KB 為單位顯示檔案系統大小。結果將與下列所示類似：

   ```
   Filesystem    1024-blocks      Free %Used    Iused %Iused Mounted on
   /dev/hd4           196608      4976   98%     1944     2% /
   /dev/hd2          1769472    623988   65%    36984     9% /usr
   ```

```
/dev/hd9var        163840     65116   61%      676     2% /var
/dev/hd3            65536     63024    4%      115     1% /tmp
/dev/hd1            49152      8536   83%      832     7% /home
/proc                   -         -    -        -     -  /proc
/dev/hd10opt        32768     26340   20%      293     4% /opt
```

查看這些結果時，您會注意到有很多可用的區塊，而且對於 /usr 上裝載的檔案系統，相關聯的使用百分比相當地低。 您決定可以藉由減少配置給 /usr 檔案系統的分割區數來釋出大量區塊。

3. 檢查 /etc/filesystems 檔案的內容，以確保能夠裝載 rootvg 中的所有檔案系統。如果無法容納，它們將不會併入重新安裝的系統中。
4. 鍵入下列指令建立 /image.data 檔案，該檔案會列出內含於安裝程序中的所有 rootvg 作用中檔案系統：

```
mkszfile
```

5. 在您喜好的編輯器中開啟 /image.data 檔案。
6. 搜尋 usr 字串以找出專屬於 /usr 檔案系統的 lv_data 段落。

   以此段落中的數字為基礎來判斷您可以減少的 /usr 檔案系統邏輯分割區數。在 /image.data 檔案的 PP_SIZE 項目中定義每個額外邏輯分割區的預設大小。您的 /image.data 檔案將如下：

```
lv_data:
    VOLUME_GROUP= rootvg
    LV_SOURCE_DISK_LIST= hdisk0
    LV_IDENTIFIER= 00042345d300bf15.5
    LOGICAL_VOLUME= hd2
    VG_STAT= active/complete
    TYPE= jfs
    MAX_LPS= 32512
    COPIES= 1
    LPs= 108
    STALE_PPs= 0
    INTER_POLICY= minimum
    INTRA_POLICY= center
    MOUNT_POINT= /usr
    MIRROR_WRITE_CONSISTENCY= on/ACTIVE
    LV_SEPARATE_PV= yes
    PERMISSION= read/write
    LV_STATE= opened/syncd
    WRITE_VERIFY= off
    PP_SIZE= 16
    SCHED_POLICY= parallel
    PP= 108
    BB_POLICY= relocatable
    RELOCATABLE= yes
    UPPER_BOUND= 32
    LABEL= /usr
    MAPFILE=
    LV_MIN_LPS= 70
    STRIPE_WIDTH=
    STRIP_SIZE=
```

   此邏輯磁區專用的邏輯分割區數為 108 (LPs=108)。

7. 使用步驟 2 的結果，來判斷 /usr 檔案系統中現有資料所需的邏輯分割區數。

   您可以使用下列指令，特別為 /usr 檔案系統顯示現有檔案大小：

```
df -k /usr
```

   結果與步驟 2 中針對 /usr 檔案系統得到的數字（以 KB 為單位）相同。例如：

```
Filesystem    1024-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd2          1769472    623988   65%    36984     9% /usr
```

   a) 從配置的 1024 區塊總數中扣除可用空間數量：

```
1769472 - 623988 = 1145484
```

   b) 新增您可能需要的空間估計值，以容納此檔案系統的所有未來預期的成長。

在此例中，將結果加上 200000。

```
1145484 + 200000 = 1345484
```

c) 將結果除以邏輯分割區大小（以位元組計）(16*1024)，以判斷您需要之邏輯分割區的最小數目。

```
1345484 / 16384 = 82.121826171875
```

使用此結果（四捨五入）來重新定義所需的邏輯分割區數 (LPs=83)。

8. 在 image.data 檔案中，將 LPs 欄位從 108 變更為 83。
9. 尋找專屬於 /usr 檔案系統的 fs_data 段落。
   您的 fs_data 段落將如下：

```
fs_data:
    FS_NAME= /usr
    FS_SIZE= 3538944
    FS_MIN_SIZE= 2290968
    FS_LV= /dev/hd2
    FS_FS= 4096
    FS_NBPI= 4096
    FS_COMPRESS= no
    FS_BF= false
    FS_AGSIZE= 8
```

10. 將實體分割區大小 (PP_SIZE) 乘以 2（實體分割區使用的 512 位元組區塊的數目），再乘以邏輯分割區數 (LPs) 來計算檔案系統大小 (FS_SIZE)。
    使用此範例中給定的值計算如下：

```
PP_SIZE * 512 區塊 * LPs = FS_SIZE
16384 * 2 * 83 = 2719744
```

11. 在 image.data 檔案中，將 FS_SIZE 欄位從 3538944 變更為 2719744。
12. 根據 /usr 檔案系統所使用的現行資料實際大小，計算最小檔案系統大小 (FS_MIN_SIZE)，如下所示：
    a) 計算所需的分割區最小數目。
    使用此範例中給定的值計算如下：

    size_in_use（請參閱步驟 7a） / PP_SIZE = 分割區數
    1145484 / 16384 = 69.914794921875

    b) 計算分割區數目所需的最小大小。
    將先前的計算結果四捨五入到 70，計算如下：

```
PP_SIZE * 512 區塊 * 分割區數 = FS_MIN_SIZE
16384 * 2 * 70 = 2293760
```

13. 在 image.data 檔案中，將 FS_MIN_SIZE 欄位從 2290968 變更為 2293760。
14. 儲存您的編輯並結束編輯器。
15. 卸載不在 rootvg 磁區群組中的所有檔案系統。
16. 如果您具有任何使用者定義的磁區群組，請鍵入下列指令以轉斷並匯出它們：

```
varyoffvg VGName

exportvg VGName
```

17. 使用磁帶機中的磁帶，鍵入下列指令來起始完整的系統備份：

```
mksysb /dev/rmt0
```

此類型備份包括您在 /image.data 檔案中指定的檔案系統大小資訊，這項資訊稍後會用來以新的檔案系統大小重新安裝系統。

註：若要起始此備份，您必須從指令行執行 **mksysb** 指令。如果您使用系統管理工具（如 SMIT），則備份會建立新的 image.data 檔案，並改寫您所做的變更。

18. 使用此備份，利用**使用現行系統設定安裝**選項重新安裝作業系統。

在安裝期間，請檢查是否已適當設定下列選項：

- **使用對映**必須設為**否**
- **縮小檔案系統**必須設為**否**

如需安裝程序的相關資訊，請參閱安裝系統備份。

19. 在安裝完作業系統後，請以「標準」模式重新啟動系統。
    此時，會調整 /usr 檔案系統的大小，但無法使用您的使用者定義檔案系統。
20. 鍵入下列指令以裝載所有檔案系統：

    ```
    mount all
    ```

    如果您收到有關已裝載之檔案系統的裝置忙碌訊息，可以忽略這些訊息。

此時，會調整 /usr 檔案系統的大小，而 root 磁區群組將擁有更多可用空間，且可以使用您的檔案系統。

相關概念
邏輯磁區儲存體
邏輯磁區是位於實體磁區上的資訊群組。

相關資訊
將 Root 磁區群組備份建立到磁帶或檔案中
/image.data 檔案說明
mkszfile 指令
mksysb 指令

疑難排解檔案系統

使用這些疑難排解方法，可以解決檔案系統可能發生的部分基本問題。如果疑難排解資訊未解決您的問題，請聯絡客戶服務代表。

修正使用者定義的檔案系統溢位

使用此程序來修正使用者定義的檔案系統溢位。

1. 移除舊的備份檔與核心檔。下面範例會移除所有 *.bak、.*.bak、a.out、core、* 或 ed.hup 檔案。

   ```
   find / \( -name "*.bak" -o -name core -o -name a.out -o \
           -name "...*" -o -name ".*.bak" -o -name ed.hup \) \
           -atime +1 -mtime +1 -type f -print | xargs -e rm -f
   ```

2. 為了避免檔案在磁碟中定期發生溢位，請執行 **skulker** 指令作為 **cron** 處理程序的一部分，並移除不必要或暫時的檔案。

   skulker 指令會清除 /tmp 目錄中的檔案、 比指定的經歷時間還要舊的檔案、a.out 檔案、核心檔案或 ed.hup 檔案。它會每天在離峰期間內執行，作為 **cron** 指令所執行的統計作業程序的一部分（假設您已啟用統計作業）。

   cron 常駐程式會在指定的日期及時間執行 shell 指令。您可以依據 crontab 檔案中所含的指示，指定定期排定的指令，如 **skulker**。使用 **crontab** 指令提交 crontab 檔案。若要編輯 crontab 檔案，您必須具有 root 使用者權限。

修正損壞的檔案系統

當檔案系統目錄結構的 i-node 或超級區塊資訊受損時， 檔案系統可能也受損。

這個毀損可能是因為硬體相關問題，或直接存取 i-node 或超級區塊資訊的程式受損。（以組譯器及 C 撰寫的程式可以略過作業系統，並直接寫入硬體。）檔案系統毀損的一個徵兆是系統找不到、無法讀取或寫入位於特定檔案系統中的資料。

若要修正受損的檔案系統，您必須診斷出問題， 然後修復它。**fsck** 指令會執行低層次的診斷與修復。

下列是修正損壞之檔案系統的程序：

1. 以 root 權限，使用下列其中一個 SMIT 捷徑來卸載損壞的檔案系統：**smit unmountfs**（若為硬式磁碟機上的檔案系統）或 **smit unmntdsk**（若為抽取式磁碟上的檔案系統）。
2. 執行 **fsck** 指令，來評估檔案系統的損壞。在下列範例中，**fsck** 指令會檢查位於 /dev/myfilelv 裝置的已卸載檔案系統：

   ```
   fsck /dev/myfilelv
   ```

   fsck 指令會檢查並以互動方式修復不一致的檔案系統。一般而言，檔案系統應該是一致的，且 **fsck** 指令僅報告檔案系統中關於檔案、已使用區塊與未使用區塊資訊。 如果檔案系統不一致，**fsck** 指令將顯示所發現不一致的相關資訊，並提示您要求同意修復。**fsck** 指令在其修復作業中是很保守的，並避免可能會導致有效的資料流失的動作。然而，在某些情況下，**fsck** 指令將建議摧毀受損的檔案。請參閱 *Commands Reference, Volume 2* 中的 **fsck** 指令說明，以取得此指令會檢查的不一致清單。
3. 如果無法修復檔案系統，請從備份中還原它。

   小心：從備份還原檔案系統會損毀並置換先前儲存在磁碟上的任何檔案系統。

   若要從備份還原檔案系統，請使用 SMIT 捷徑 **smit restfilesys** 或下面範例中顯示的一系列指令：

   ```
   mkfs /dev/myfilelv
   mount /dev/myfilelv /myfilesys
   cd /myfilesys
   restore -r
   ```

   在此範例中，**mkfs** 指令會在名為 /dev/myfilelv 的裝置上建立新的檔案系統，並起始設定磁碟機標籤、檔案系統標籤，以及啟動區塊。**mount** 指令會建立 /dev/myfilelv 作為 **myfilesys** 的裝載點，且 **restore** 指令會從備份中擷取檔案系統。

   如果是利用遞增式檔案系統備份來製作備份，您必須以漸進式備份層次順序（例如：0、1、2）來還原備份。使用 **smit restfilesy** 來還原整個檔案系統時，請鍵入目標目錄、還原裝置 （/dev/rfd0 以外的裝置）及單一輸入作業中讀取的區塊數。

修正檔案系統超級區塊中毀損的識別碼

如果檔案系統的超級區塊損壞，則將無法存取該檔案系統。您可以修正檔案系統超級區塊中毀損的識別碼。

大部分的超級區塊損壞都無法修復。下面的程序說明當問題是由損毀的識別碼所導致時，要如何修復 JFS 檔案系統中的超級區塊。如果 JFS2 檔案系統中的主要超級區塊損毀，請使用 **fsck** 指令自動複製次要超級區塊並修復主要超級區塊。

在下面的範例情節中，假設 /home/myfs 是實體磁區 /dev/lv02 上的 JFS 檔案系統。

此作法範例情節中的資訊已使用特定版本的 AIX 進行測試。依據您的 AIX 版本及層次，所得到的結果可能大不相同。

1. 使用下列指令卸載您懷疑可能已損壞的 /home/myfs 檔案系統：

   ```
   umount /home/myfs
   ```
2. 若要確認檔案系統是否損壞，請對該檔案系統執行 **fsck** 指令。例如：

   ```
   fsck -p /dev/lv02
   ```

   如果問題是超級區塊損壞，則 **fsck** 指令會傳回下列其中一則訊息：

   ```
   fsck：不是 AIXV5 檔案系統
   ```

   或

   ```
   無法辨識的檔案系統類型
   ```
3. 以 root 權限使用 **od** 指令來顯示檔案系統的超級區塊，如下列範例所示：

   ```
   od -x -N 64 /dev/lv02 +0x1000
   ```

其中 -x 旗標會以十六進位格式顯示輸出，-N 旗標會指示系統從偏移參數 (+) 格式化不超過 64 個輸入位元組，此偏移參數 (+) 可指定檔案中開始輸出檔案的那一點。下面是範例輸出：

```
0001000  1234 0234 0000 0000 0000 4000 0000 000a
0001010  0001 8000 1000 0000 2f6c 7633 0000 6c76
0001020  3300 0000 000a 0003 0100 0000 2f28 0383
0001030  0000 0001 0000 0200 0000 2000 0000 0000
0001040
```

在上面的輸出中，請記下 0x1000 處的損毀識別值 (1234 0234)。如果在建立檔案系統時採用了所有預設值，則識別碼應為 0x43218765。如果置換了任何預設值，則識別碼應為 0x65872143。

4. 請使用 **od** 指令檢查次要超級區塊以尋找正確的識別碼。下面是範例指令及其輸出：

```
$ od -x -N 64 /dev/lv02 +0x1f000
001f000  6587 2143 0000 0000 0000 4000 0000 000a
001f010  0001 8000 1000 0000 2f6c 7633 0000 6c76
001f020  3300 0000 000a 0003 0100 0000 2f28 0383
001f030  0000 0001 0000 0200 0000 2000 0000 0000
001f040
```

請記下 0x1f000 處的正確識別值。

5. 將次要超級區塊複製到主要超級區塊。下面是範例指令及輸出：

```
$ dd count=1 bs=4k skip=31 seek=1 if=/dev/lv02 of=/dev/lv02

dd: 1+0 records in.
dd: 1+0 records out.
```

6. 使用 **fsck** 指令清除因使用次要超級區塊而導致的不一致檔案。例如：

```
fsck /dev/lv02 2>&1 | tee /tmp/fsck.errs
```

相關資訊

fsck 指令
od 指令

磁碟溢位

當太多檔案填滿了所配置的空間時，即會發生磁碟溢位。這可能是因為失控的處理程序建立了太多不必要的檔案。

您可以使用下列程序來更正問題：

註 : 您必須具有 root 使用者權限，才能移除不是您擁有的處理程序。

識別問題處理程序

使用下列程序來隔離問題處理程序。

1. 若要檢查處理程序狀態並識別可能造成問題的處理程序，請鍵入：

```
ps -ef | pg
```

ps 指令會顯示處理程序狀態。**-e** 旗標會寫入所有處理程序（核心處理程序除外）的相關資訊，而 **-f** 旗標會產生處理程序的完整清單，包括建立處理程序時所使用的指令名稱與參數。**pg** 指令會限制每次輸出一頁，所以資訊不會捲動過快而離開畫面。

檢查正在使用過量系統資源（如 CPU 時間）的系統或使用者處理程序。系統處理程序，如 **sendmail**、**routed** 及 **lpd**，似乎是最容易失控的系統處理程序。

2. 若要檢查使用的 CPU 超出預期的使用者處理程序，請鍵入：

```
ps -u
```

3. 請記下每一個問題處理程序的處理程序 ID (PID)。

終止處理程序

您可以終止有問題的處理程序。

使用下列程序來終止問題處理程序：

1. 鍵入下列指令來終止造成問題的處理程序：

   ```
   kill -9 PID
   ```

   其中的 *PID* 是問題處理程序的 ID。
2. 鍵入下列指令來移除處理程序生成的檔案：

   ```
   rm file1 file2 file3
   ```

   其中 *file1 file2 file3* 代表處理程序相關檔案的名稱。

收回檔案空間但不終止處理程序

若要收回配置給作用中檔案的區塊，但不要終止處理程序，請將另一個指令的輸出重新導向至檔案。資料重新導向會截斷檔案，並收回記憶體區塊。

從檔案系統中移除作用中的檔案時，配置給檔案的區塊仍維持配置， 直到移除最後一次開啟的參照為止，這是因為處理程序關閉檔案，或因為開啟檔案的處理程序已終止。如果失控的處理程序正在寫入檔案，而檔案已移除，則在處理程序終止前，無法釋出配置給檔案的區塊。

例如：

```
$ ls -l
total 1248
-rwxrwxr-x      1 web   staff   1274770 Jul 20 11:19 datafile
$ date > datafile
$ ls -l
total 4
-rwxrwxr-x      1 web   staff        29 Jul 20 11:20 datafile
```

date 指令的輸出會取代之前的 datafile 檔案內容。截斷檔案的報告區塊反映出大小差異，從 1248> 到 4。如果失控的處理程序繼續附加資訊到此新截斷檔案，則下一個 **ls** 指令會產生下列結果：

```
$ ls -l
total 8
-rxrwxr-x       1 web   staff   1278866 Jul 20 11:21 datefile
```

datafile 檔案的大小反映出失控處理程序所執行的附加， 但配置的區塊數很小。現在，datafile 檔案中有一個孔。檔案孔是沒有配置磁碟區塊的檔案範圍。

/（根）溢位

當根檔案系統 (/) 填滿時，請檢查下列內容。

- 使用下列指令來讀取 /etc/security/failedlogin 檔案的內容：

  ```
  who /etc/security/failedlogin
  ```

  TTY 重建太快的狀況可能會建立失敗的登入項目。若要在讀取或儲存輸出之後清除檔案，請執行下列指令：

  ```
  cp /dev/null /etc/security/failedlogin
  ```

- 檢查 /dev 目錄以找出未正確鍵入的裝置名稱。如果未正確鍵入裝置名稱（如 rmto 而不是 rmt0），則會在 /dev 中建立稱為 rmto 的檔案。指令在失敗之前通常會繼續，直到整個根檔案系統填滿為止。 /dev 是根 (**/**) 檔案系統的一部分。尋找不是裝置（沒有主要或次要號碼）的項目。若要檢查此狀況，請使用下列指令：

  ```
  cd /dev
  ls -l | pg
  ```

  在指出一般檔案之檔案大小的同一位置上，裝置檔案有兩個以逗點隔開的數字。例如：

  ```
  crw-rw-rw-   1 root     system    12,0 Oct 25 10:19 rmt0
  ```

如果檔名或大小位置指出無效的裝置（如下列範例所示），請移除相關的檔案：

```
crw-rw-rw-   1 root     system   9375473 Oct 25 10:19 rmto
```

註：

– 請勿移除 /dev 目錄中的有效裝置名稱。無效裝置的一個指示器是大於 500 個位元組的相關檔案大小。

– 如果系統審核在執行中，則預設 /audit 目錄可迅速填滿且需要特別注意。

· 使用 **find** 指令，檢查可被移除的特大型檔案。例如，若要尋找根 (/) 目錄中大於 1 MB 的所有檔案，請使用下列指令：

```
find / -xdev -size  +2048 -ls |sort -r -n +6
```

此指令會尋找所有大於 1 MB 的檔案，並以反向順序將其排序（最大的檔案最先排）。find 指令的其他旗標（如 **-newer**）可能在此搜尋中很有用。如需相關資訊，請參閱 **find** 指令的指令說明。

註：檢查根目錄時，會在 /dev 目錄中裝置的主要及次要號碼之中穿插實際的檔案及檔案大小。可忽略以逗點區隔的主要及次要號碼。

移除任何檔案之前，請使用下列指令來確定檔案目前未被使用者處理程序使用：

```
fuser filename
```

其中 *filename* 是猜想的大型檔案名稱。如果在移除時檔案處於開啟狀態，則僅會從目錄清單移除它。直到使檔案保持開啟的處理程序結束後，才會釋放配置給該檔案的區塊。

解決 /var 檔案系統中的溢位

當 /var 檔案系統填滿時，請檢查下列內容。

· 您可以使用 find 指令在 /var 目錄中尋找大型檔案。例如：

```
find /var -xdev -size  +2048 -ls| sort -r  +6
```

如需相關資訊，請參閱 **find** 指令的指令說明。

· 檢查 /var/tmp 中已作廢或剩餘的檔案。

· 檢查 /var/adm/wtmp 檔案的大小，該檔案記載了所有 logins、rlogins 及 telnet 階段作業。除非系統統計作業在執行中，否則日誌將無限增大。系統統計作業每晚都會清除日誌。可清除或編輯 /var/adm/wtmp 檔案，以移除舊的及不想要的資訊。若要清除它，請使用下列指令：

```
cp /dev/null  /var/adm/wtmp
```

若要編輯 /var/adm/wtmp 檔案，請先使用下列指令暫時複製檔案：

```
/usr/sbin/acct/fwtmp < /var/adm/wtmp >/tmp/out
```

編輯 /tmp/out 檔案以移除不想要的項目，然後使用下列指令來置換原始檔案：

```
/usr/sbin/acct/fwtmp -ic < /tmp/out > /var/adm/wtmp
```

· 使用下列程序，清除 /var/adm/ras 目錄中的錯誤日誌。除非手動清除，否則永遠不會清除錯誤日誌。

註：絕對不要使用 **cp /dev/null** 指令來清除錯誤日誌。零長度的 errlog 檔案會停用作業系統的錯誤記載功能，且必須從備份進行置換。

1. 使用下列指令停止錯誤常駐程式：

   ```
   /usr/lib/errstop
   ```

2. 使用下列其中一個指令，移除錯誤日誌檔或將它移至不同的檔案系統：

   ```
   rm /var/adm/ras/errlog
   ```

或

```
mv /var/adm/ras/errlog filename
```

其中 *filename* 是移動的 errlog 檔案名稱。

註： 如果您移除錯誤日誌檔，則會刪除歷程錯誤資料。

3. 使用下列指令來重新啟動錯誤常駐程式：

```
/usr/lib/errdemon
```

註： 藉由執行 **cron** 中的下列項目，來考量限制 errlog：

```
0 11 * * * /usr/bin/errclear -d S,O 30
0 12 * * * /usr/bin/errclear -d H 90
```

- 檢查此目錄中的 `trcfile` 檔案是否為大型檔案。如果它是大型的，且目前未在執行追蹤，則可使用下列指令來移除檔案：

```
rm /var/adm/ras/trcfile
```

- 如果您的傾出裝置設為 hd6（這是預設值），則 `/var/adm/ras` 目錄中可能有大量 `vmcore*` 檔案。如果它們的檔案日期是舊的，或您不想保留它們，則可使用 **rm** 指令將它們移除。
- 檢查包含佇列子系統檔案的 `/var/spool` 目錄。使用下列指令來清除佇列子系統：

```
stopsrc -s qdaemon
rm /var/spool/lpd/qdir/*
rm /var/spool/lpd/stat/*
rm /var/spool/qdaemon/*
startsrc -s qdaemon
```

- 檢查包含統計記錄的 `/var/adm/acct` 目錄。如果統計作業在執行中，則此目錄可能包含數個大型檔案。
- 檢查 `/var/preserve` 目錄，找出終止的 **vi** 階段作業。一般來說，移除這些檔案是安全的。如果使用者要回復階段作業，您可以使用 **vi -r** 指令來列出所有可回復的階段作業。若要回復特定的階段作業，請使用 **vi -r** *filename*。
- 修改 `/var/adm/sulog` 檔案，該檔案會記錄嘗試使用 **su** 指令的次數以及每一次是否順利完成。此為純文字檔，可使用喜好的編輯器來檢視及修改它。如果檔案已移除，則下一次嘗試的 **su** 指令會重建它。修改 `/var/tmp/snmpd.log`，其中會記錄 **snmpd** 常駐程式的事件。如果檔案已移除，則 **snmpd** 常駐程式將重建它。

  註： 可限制 `/var/tmp/snmpd.log` 檔案的大小，這樣它便不會無限增大。編輯 `/etc/snmpd.conf` 檔案，以在適當的大小區段中變更數目（以位元組為單位）。

修正其他檔案系統及一般搜尋技術

將 **find** 指令與 **-size** 旗標搭配使用時，可以尋找大型檔案，或如果檔案系統最近溢位了，請使用 **-newer** 旗標來尋找最近修改過的檔案。

若要產生 **-newer** 旗標尋找所依據的檔案，請使用下列 touch 指令：

```
touch mmddhhmm filename
```

其中的 *mm* 是月份，*dd* 是日期，*hh* 是 24 小時時間格式中的小時，*mm* 是分鐘，*filename* 是您使用 **touch** 指令建立的檔案名稱。

在您以 touch 指令建立檔案之後，可使用下列指令來尋找更新的大型檔案：

```
find /filesystem_name -xdev -newer touch_filename -ls
```

您還可以使用 **find** 指令來尋找在最近 24 小時內變更過的檔案，如下列範例所示：

```
find /filesystem_name -xdev -mtime 0 -ls
```

裝載

裝載使檔案系統、檔案、目錄、裝置及特殊檔案可在特定位置使用。它是使檔案系統可存取的唯一方法。

mount 指令可指示作業系統連接指定目錄上的檔案系統。

如果您具有對裝載之檔案或目錄的存取權限及對裝載點的寫入許可權，則可裝載此檔案或目錄。 系統群組成員亦可執行裝置裝載（在其中會將裝置或檔案系統裝載到若干目錄上）及 /etc/filesystems 檔案中所說明的裝載。以 root 使用者權限來操作的使用者可任意裝載檔案系統，方法是在指令行上同時命名裝置及目錄。 /etc/filesystems 檔案用來定義系統起始設定時自動進行的裝載。**mount** 指令用於系統啟動後的裝載。

裝載點

裝載點是新檔案系統、目錄或檔案在其上可存取的目錄或檔案。若要裝載檔案系統或目錄，則裝載點必須是目錄；若要裝載檔案，則裝載點必須是檔案。

通常，檔案系統、目錄或檔案會裝載到空的裝載點上，但不是必要的。如果當作裝載點的檔案或目錄有包含任何資料，則其他檔案或目錄裝載到該檔案或目錄上時，將無法存取該資料。實際上，裝載的檔案或目錄涵蓋了該目錄中先前的內容。當取消原始目錄或檔案上的裝載時，該目錄或檔案即重新變得可存取。

當檔案系統裝載到某個目錄上時，已裝載之檔案系統根目錄的許可權優先於裝載點的許可權。裝載的目標目錄的 ..（點點）上層目錄項目是一個例外。裝載點的上層目錄資訊必須可使用，作業系統才可存取新檔案系統。

比方說，如果現行工作目錄為 /home/frank，則指令 **cd ..** 會將工作目錄變更為 /home。如果 /home/frank 目錄為已裝載之檔案系統的根目錄，則作業系統必須尋找 /home/frank 目錄的上層目錄資訊，**cd ..** 指令才能順利完成。

對於需要上層目錄資訊才會順利完成的任何指令，使用者必須在裝載的目標目錄中具有搜尋許可權。授與裝載的目標目錄搜尋許可權的失敗可能會導致無法預期的結果，特別是當看不見裝載的目標目錄許可權時。 常見問題為 **pwd** 指令的失敗。若於裝載的目標目錄中無搜尋許可權，則 **pwd** 指令會傳回此訊息：

```
pwd：拒絕許可權
```

一律將裝載的目標目錄許可權至少設為 111，即可避免此問題。

裝載檔案系統、目錄及檔案

有兩種類型的裝載，遠端裝載及本端裝載。遠端裝載於透過電信線路以傳輸資料的遠端系統上執行。遠端檔案系統（如「網路檔案系統 (NFS)」）需要在裝載檔案之前先將其匯出。本端裝載是在本端系統上執行的裝載。

每個檔案系統都與不同的裝置（邏輯磁區）相關聯。使用檔案系統之前，該檔案系統必須連接到現有的目錄結構（根檔案系統或已連接的另一個檔案系統）。**mount** 指令會建立此連線。

可透過多個路徑來存取相同的檔案系統、目錄或檔案。例如，如果您有一個資料庫，且有數位使用者使用此資料庫，則同一資料庫具有數個裝載會非常有用。每個裝載應有其自己的名稱及密碼以用於追蹤及分隔工作。這可藉由在不同的裝載點上裝載同一檔案系統來完成。例如，您可從 /home/server/database 裝載到指定為 /home/user1、/home/user2 及 /home/user3 的裝載點：

```
/home/server/database        /home/user1
/home/server/database        /home/user2
/home/server/database        /home/user3
```

透過符號鏈結的使用，檔案系統、目錄或檔案可供各種使用者使用。符號鏈結是使用 **ln -s** 指令建立的。將多位使用者鏈結到一個中央檔案會確保每次使用者存取該檔案時，都會反映對該檔案的所有變更。

自動裝載控制

可將裝載設為在系統起始設定期間自動發生。

有兩種類型的自動裝載。第一種類型由需要開機並執行系統的那些裝載組成。這些檔案系統由開機處理程序明確裝載。/etc/filesystems 檔案中此類檔案系統的段落具有 mount = automatic。第二種類型的自動裝載是使用者控制的。當 /etc/rc Script 發出 **mount all** 指令時，就會裝載這些檔案系統。在 /etc/filesystems 中，由使用者控制之自動裝載段落具有 mount = true。

/etc/filesystems 檔案會控制自動裝載； 它們會按階層架構裝載，每次一個裝載點。亦可以特定次序（可變更及重新排列）來裝載它們。如需 /etc/filesystems 檔案的相關資訊，請參閱 /etc/filesystems。

/etc/filesystems 檔案組織成段落， 每個裝載一個段落。段落說明對應檔案系統的屬性及如何裝載它。系統以 /etc/filesystems 檔案中顯示的次序來裝載檔案系統。以下是 /etc/filesystems 檔案中的段落範例：

```
/:
  dev=/dev/hd4
  vol="root"
  mount=automatic
  check=false
  free=true
  vfs=jfs
  log=/dev/hd8
  type-bootfs

/home:
  dev=/dev/hd1
  vfs=jfs
  log=/dev/hd8
  mount=true
  check=true
  vol="/home"
  free=false

/usr:
  /dev=/dev/hd2
  vfs=jfs
  log=/dev/hd8
  mount=automatic
  check=false
  type=bootfs
  vol="/usr"
  free=false
```

您可編輯 /etc/filesystems 檔案來控制裝載發生的次序。 若裝載未順利完成，則 /etc/filesystems 檔案中定義的任何隨後裝載會繼續進行裝載。 例如，若 /home 檔案系統的裝載未順利完成，則 /usr 檔案系統的裝載會繼續並會被裝載。可能因為像是印刷錯誤、相依關係或系統問題的原因而導致裝載未順利完成。

無磁碟工作站的裝載安全

無磁碟工作站必須具有在遠端機器上建立並存取裝置特殊檔案的能力，以從伺服器裝載它們的 /dev 目錄。因為伺服器無法區分用於用戶端的裝置特殊檔案與用於伺服器的裝置特殊檔案，所以伺服器上的使用者可能能夠使用用戶端裝置的特殊檔案來存取伺服器的實體裝置。

例如，使用 **tty** 可自動將 **tty** 的所有權設定給使用者。如果用戶端及伺服器上的使用者 ID 不同，則伺服器上的非特許使用者可存取正在由伺服器上不同使用者所使用的 **tty**。

用戶端上特許的使用者可建立裝置特殊檔案以符合伺服器上的實體裝置，並使它們不需要專用權就可存取。然後使用者可使用伺服器上非特許的帳戶來存取正常受保護的裝置（使用新裝置特殊檔案）。

類似安全問題與用戶端及伺服器上 **setuid** 及 **setgid** 程式的使用有關。 無磁碟型用戶端必須能建立於伺服器上並執行 **setuid** 及 **setgid** 程式，以用於正常作業。相同地，伺服器將無法區分用於伺服器的那些程式與用於用戶端的那些程式。

此外，在伺服器及用戶端之間的使用者 ID 及群組 ID 可能不相符，因此伺服器上的使用者可能能夠以本來不是其所屬的能力來執行程式。

此問題存在的原因，是因為 **setuid** 及 **setgid** 程式及 裝置特殊檔案僅應於建立它們的機器上使用。

解決方案是使用 **mount** 指令的安全選項，這些安全選項可以限制使用這些物件的能力。 這些選項亦可使用於 /etc/filesystems 檔案的段落中。

mount 指令中的 nosuid 選項可防止 **setuid** 及 **setgid** 程式的執行，這些程式是透過產生的裝載檔案系統存取。此選項用於特定主機上正被裝載的任何檔案系統，僅供不同的主機使用（例如：為無磁碟型用戶端匯出）。

mount 指令中的 nodev 選項可防止使用透過產生的裝載檔案系統存取的裝置特殊檔案，來開啟裝置。此選項亦用於正被裝載的任何檔案系統，僅供不同的主機使用（例如為無磁碟型用戶端匯出）。

一般而言，伺服器上的使用者沒有對 /export 目錄的任何存取權限。

匯出 /export/root 目錄

必須使用讀取/寫入許可權匯出 /export/root 目錄，且伺服器上的 root 使用者必須具有存取權。不過，您可能想要使用 **mount** 指令的下列選項裝載此目錄：

項目	說明
nosuid	防止伺服器上的使用者執行用戶端的 **setuid** 程式
nodev	防止使用者使用用戶端的裝置特殊檔案存取伺服器裝置。

若不使用這些選項裝載 /export/root 目錄，則可使用另一種方式， 即避免向在伺服器上執行的使用者授與對 /export/root 目錄的任何存取權限。

匯出 /export/exec 目錄

使用唯讀許可權匯出 /export/exec 目錄，且必須提供 root 存取權。不過，您可能想要使用 **mount** 指令的下列選項裝載此目錄：

項目	說明
nosuid	防止伺服器上的使用者執行用戶端的 **setuid** 程式。如果您正在匯出伺服器 /usr 目錄，您將無法使用 **nousid** 選項。
nodev	防止使用者使用用戶端的裝置特殊檔案存取伺服器裝置。

匯出 /export/share 目錄

使用唯讀許可權匯出 /export/share 目錄，且必須提供 root 存取權。因為此目錄一般僅包含資料（無可執行檔或裝置），所以您無需使用裝載安全選項。

匯出 /export/home 目錄

裝載使用者 /home 目錄有數種方法：

- 您可以將 /export/home/*Clienthostname* 目錄裝載至用戶端 /home 目錄上。在此狀況下，用戶端具有讀取/寫入許可權，且 root 使用者具有存取權。若要確保系統安全，請在 **mount** 指令使用下列選項，來裝載 /export/home 目錄：

項目	說明
nosuid	防止伺服器上的使用者執行用戶端的 **setuid** 程式。
nodev	防止使用者使用用戶端的裝置特殊檔案存取伺服器裝置。

- 可將伺服器上的 /home 目錄裝載至用戶端的 /home 目錄上。在此狀況下，可使用讀取/寫入許可權匯出 /home 目錄（無需 root 存取權）。若要確保系統安全，請使用 **mount** 指令的 nosuid 及 nodev 選項， 同時裝載伺服器及用戶端上的 /home 目錄。
- 另外，您可於用戶端上透過用戶端上的 /home/*Username* 目錄來 裝載伺服器上每個 /home/*UserName* 目錄，以便使用者可登入不同的機器，而仍具有對其起始目錄的存取權。在此情況下， 伺服器及用戶端上的 /home/*Username* 目錄都是使用 **mount** 指令的 nousid 及 nodev 選項裝載。

匯出 /export/swap 目錄

使用讀取/寫入許可權及 root 存取權匯出 /export/swap/*Clienthostname* 檔案。不必採取安全措施。伺服器上的使用者沒有對 /export/swap/*Clienthostname* 檔案的任何存取權限。

無磁碟裝載

雖然無磁碟工作站的檔案系統是從伺服器 /exports 目錄裝載到無磁碟機器上，但是檔案系統看起來就像是獨立式機器上的檔案系統。

下列顯示伺服器匯出與無磁碟工作站裝載點之間的關係：

伺服器匯出	無磁碟匯入
/export/root/*HostName*	/（根）
/export/exec/*SPOTName*	/usr
/export/home/*HostName*	/home
/export/share	/usr/share
/export/dump	無磁碟型用戶端當作傾出空間使用
/export/swap	由無磁碟型用戶端作為遠端分頁空間使用

如需 /export 目錄的相關資訊，請參閱 第 63 頁的『/export 目錄』。

一般而言，伺服器上的使用者沒有對 /export 目錄的任何存取權限。

匯出 /export/root 目錄

必須使用讀取/寫入許可權匯出 /export/root 目錄，且伺服器上的 root 使用者必須具有存取權。不過，您可能想要使用 **mount** 指令的下列選項裝載此目錄：

項目	說明
nosuid	防止伺服器上的使用者執行用戶端的 **setuid** 程式
nodev	防止使用者使用用戶端的裝置特殊檔案存取伺服器裝置。

若不使用這些選項裝載 /export/root 目錄，則可使用另一種方式， 即避免向在伺服器上執行的使用者授與對 /export/root 目錄的任何存取權限。

匯出 /export/exec 目錄

使用唯讀許可權匯出 /export/exec 目錄，且必須提供 root 存取權。不過，您可能想要使用 **mount** 指令的下列選項裝載此目錄：

項目	說明
nosuid	防止伺服器上的使用者執行用戶端的 **setuid** 程式。如果您正在匯出伺服器 /usr 目錄，您將無法使用 **nousid** 選項。
nodev	防止使用者使用用戶端的裝置特殊檔案存取伺服器裝置。

匯出 /export/share 目錄

使用唯讀許可權匯出 /export/share 目錄，且必須提供 root 存取權。因為此目錄一般僅包含資料（無可執行檔或裝置），所以您無需使用裝載安全選項。

匯出 /export/home 目錄

裝載使用者 /home 目錄有數種方法：

- 您可以將 /export/home/*Clienthostname* 目錄裝載至用戶端 /home 目錄上。在此狀況下，用戶端具有讀取/寫入許可權，且 root 使用者具有存取權。若要確保系統安全，請在 **mount** 指令使用下列選項，來裝載 /export/home 目錄：

項目	說明
nosuid	防止伺服器上的使用者執行用戶端的 **setuid** 程式。
nodev	防止使用者使用用戶端的裝置特殊檔案存取伺服器裝置。

- 可將伺服器上的 /home 目錄裝載至用戶端的 /home 目錄上。在此狀況下，可使用讀取/寫入許可權匯出 /home 目錄（無需 root 存取權）。若要確保系統安全，請在伺服器及用戶端上使用 **mount** 指令的 **nosuid** 及 **nodev** 選項，裝載 /home 目錄。
- 另外，您可於用戶端上透過用戶端上的 /home/*Username* 目錄來 裝載伺服器上每個 /home/*UserName* 目錄，以便使用者可登入不同的機器，而仍具有對其起始目錄的存取權。在此情況下，使用 **mount** 指令的 **nousid** 及 **nodev** 選項，在伺服器及用戶端上裝載 /home/*Username* 目錄。

匯出 /export/dump 目錄

使用讀取/寫入許可權及 root 存取權匯出 /export/dump/*Clienthostname* 目錄。 伺服器上的使用者沒有對 /export/dump/*Clienthostname* 檔案的任何存取權限。

匯出 /export/swap 目錄

使用讀取/寫入許可權及 root 存取權匯出 /export/swap/*Clienthostname* 檔案。不必採取安全措施。伺服器上的使用者沒有對 /export/swap/*Clienthostname* 檔案的任何存取權限。

檔案系統類型

AIX 支援多種檔案系統類型。

這些裝置包括：

日誌型檔案系統 (JFS) 或加強型日誌型檔案系統 (JFS2)

支援整個檔案系統語意集。這些檔案系統使用資料庫日誌登載技術來保持其結構的一致性。這可防止在系統異常停止時損壞檔案系統。

每個 JFS 或 JFS2 均位於個別的邏輯磁區上。作業系統於起始設定期間裝載檔案系統。此多重檔案系統配置有助於系統管理功能（如備份、還原及修復），因為它可以隔離一部分檔案樹以便您使用。

JFS 是支援整個檔案系統指令的基本檔案系統類型。

JFSJFS2 是支援整個檔案系統指令的基本檔案系統類型。

JFS 與 JFS2 的差異在於 JFS2 設計用來支援大型檔案及大型檔案系統。

網路檔案系統 (NFS)

為一種分散式檔案系統，可讓使用者存取位於遠端電腦上的檔案及目錄， 及使用那些檔案與目錄，如同它們在本端一樣。例如，使用者可使用作業系統指令，建立、移除、讀取及寫入遠端檔案及目錄，以及為其設定檔案屬性。

CD-ROM 檔案系統 (CDRFS)
允許經由一般檔案系統介面來存取 CD-ROM 內容（開啟、讀取及關閉）。

DVD-ROM 檔案系統 (UDFS)
可讓您透過一般檔案系統介面存取 DVD 的內容。

JFS 及 JFS2

日誌型檔案系統 (JFS) 及加強型日誌型檔案系統 (JFS2) 內建於基本作業系統之中。這兩種檔案系統類型均將它們的檔案及目錄資料鏈結至 AIX「邏輯磁區管理程式」 所使用的結構，進行儲存及擷取。

差異為 JFS2 是設計用來容納 64 位元核心及較大型檔案。

下列各節說明這些檔案系統。除非另有說明，否則下列各節對 JFS 及 JFS2 均適用。

JFS 與 *JFS2* 功能

「加強型日誌型檔案系統 (JFS2)」是一種檔案系統，它能儲存的檔案，遠大於現有的「日誌型檔案系統 (JFS)」。

您可以選擇實作 JFS 或 JFS2。JFS2 是 AIX 6.1 中的預設檔案系統。

註 : JFS2 檔案系統不像 JFS 檔案系統，它不允許在目錄類型的檔案上使用 **link()** API。 此一限制可能造成能在 JFS 檔案系統正常工作的應用程式到了 JFS2 檔案系統上卻會失敗。

下表提供 JFS 和 JFS2 功能的摘要：

功能	JFS2	JFS
片段及區塊大小	區塊大小（位元組）：512, 1024, 2048, 4096 最大檔案系統大小 (TB)：4, 8, 16, 32	片段大小（位元組）：512, 1024, 2048, 4096 最大檔案系統大小 (GB)：128, 256, 512, 1024
最大檔案系統大小	32 TB	1 TB
最小檔案系統大小	16 MB	不適用
最大檔案大小	16 TB	約 63.876 GB
i-node 的數目	動態，只受磁碟空間限制	固定，於檔案系統建立時設定
目錄組織	B 樹	線性
壓縮	否	是
限額	是	是
錯誤記載	是	是

註：

1. 搭配 32 位元核心使用時，最大檔案大小及最大檔案系統大小限制為 (1 TB -（實體分割區大小）)。比方說，如果磁區群組的實體分割區大小是 64 MB，則最大檔案系統大小為 (1 TB - 64 MB) = (1048576 MB - 64 MB) = 1048512 MB。這是因為使用 32 位元核心時，會對邏輯磁區的大小上限採取基本限制。
2. JFS2 支援標準 AIX 錯誤記載方法。如需 AIX 錯誤記載的相關資訊，請參閱 *General Programming Concepts: Writing and Debugging Programs* 中的錯誤記載概觀。

***JFS* 及 *JFS2* 磁碟空間區段**

許多 UNIX 檔案系統僅會配置連續的磁碟空間，這種磁碟空間單位數等於檔案及目錄邏輯分割所用的邏輯區塊大小。這些配置單位通常稱為磁碟區塊，且單一磁碟區塊是專門用來儲存包含於檔案或 目錄之單一邏輯區塊中的資料。

使用相對較大的邏輯區塊大小（如 4096 個位元組）及維護大小等於邏輯區塊的磁碟區塊配置，有助於減少單一檔案系統作業必須執行的磁碟 I/O 作業數。磁碟上的檔案或目錄資料是儲存在少量的大型磁碟區塊中，而不是大量的小型磁碟區塊中。例如，如果邏輯區塊大小是 4096 個位元組，則會為大小等於或小於 4096 個位元組的檔案，配置單一 4096 位元組磁碟區塊。因此，讀取或寫入作業僅需要執行單一磁碟 I/O 作業，以存取磁碟上的資料。如果邏輯區塊較小，就需要為相同數量的資料配置多個磁碟區塊，則存取資料時即需要多次磁碟 I/O 作業。因為大型磁碟區塊可容納較多資料，所以大型邏輯區塊及相等的磁碟區塊大小，亦有助於減少新增資料到檔案及目錄時必須執行的磁碟空間配置活動量。

不過，如果檔案系統包含許多小型檔案及目錄，將磁碟空間配置單位限制為邏輯區塊大小可能會浪費磁碟空間。在將等於邏輯區塊大小的磁碟空間配置給檔案或目錄的部分邏輯區塊時，會浪費磁碟空間。因為部分邏輯區塊總是包含小於邏輯區塊大小的資料，所以部分邏輯區塊僅會使用配置給它的磁碟空間的一部分。剩餘部分仍舊沒有使用，因為其他任何檔案或目錄均無法將其內容寫入已配置的磁碟空間。對於含有大量小型檔案及目錄的檔案系統而言，所浪費的磁碟空間總量可能會很大。

日誌型檔案系統 (JFS) 會將磁碟空間分成稱為片段的配置單位。 加強型日誌型檔案系統 (JFS2) 將磁碟空間分成區塊。 目標是相同的：即有效地儲存資料。

JFS 片段小於 4096 個位元組的預設磁碟配置大小。藉由將資料更有效地儲存於檔案或目錄部分邏輯區塊中， 片段可使浪費的磁碟空間減至最小。JFS 片段支援的功能行為是基於 Berkeley Software Distribution (BSD) 片段支援所提供的功能行為。

JFS2 支援多種檔案系統區塊大小：512、1024、2048 及 4096。藉由將資料更有效地儲存在檔案或目錄的部分邏輯區塊中，較小的區塊大小可使浪費的磁碟空間減至最小。較小的區塊大小亦會導致作業額外負擔。JFS2 的區塊大小在其建立期間指定。 不同的檔案系統可具有不同的區塊大小，但單一檔案系統內只能使用一個區塊大小。

JFS 片段
於 JFS 中，磁碟空間配置單位稱為片段，其為可小於 4096 個位元組的邏輯區塊大小。

使用小於 4096 個位元組的片段時，僅使用容納資料所需的片段數，可更有效地儲存部分邏輯區塊內所包含的資料。例如，可為僅有 500 個位元組的部分邏輯區塊配置一個 512 個位元組的片段（假設片段大小為 512 個位元組），從而大大地減少浪費的磁碟空間量。如果部分邏輯區塊的儲存需求增加，則會配置一或多個額外的片段。

在建立檔案系統時會指定其片段大小。日誌型檔案系統 (JFS) 可允許的片段大小為 512、1024、2048 及 4096 個位元組。不同的檔案系統可具有不同的片段大小，但單一檔案系統內只能使用一個片段大小。單一系統（機器）上亦可同時存在不同的片段大小，以便使用者可針對每個檔案系統選取最適合的片段大小。

JFS 片段支援使檔案系統可被看成是一系列連續的片段，而不是一系列連續的磁碟區塊。不過，為保持磁碟作業的有效性，常常以 4096 個位元組的單位來配置磁碟空間，以使磁碟區塊或配置單位大小保持等於邏輯區塊。這種狀況下的磁碟區塊配置可視為配置 4096 個位元組的連續片段。

隨著檔案系統片段大小的減小，作業額外負擔（額外的磁碟探查、資料轉送及配置活動）及磁碟空間的較佳使用率均會增加。為保持增加的額外負擔與增加的可用磁碟空間之間的最佳平衡，下列因素適用於 JFS 片段支援：

· 若可能的話，為檔案或目錄邏輯區塊保持使用 4096 個位元組片段的磁碟空間配置。

· 僅大小小於 32KB 之檔案或目錄的部分邏輯區塊可配置小於 4096 個位元組的片段。

當檔案系統中檔案及目錄大小超過 32 KB，保持部分邏輯區塊之小於 4096 個位元組磁碟空間配置的益處會減少。節省之磁碟空間佔總檔案系統空間的百分比會減小，而保持小型磁碟空間配置的額外效能損失仍不變。因為小於 4096 個位元組的磁碟空間配置用於小型檔案及目錄時，對磁碟空間的使用率會最有效益，所以針對等於或大於 32 KB 之檔案及目錄的邏輯區塊，一律會配置 4096 個位元組片段。針對與這類大型檔案或目錄相關的任何部分邏輯區塊，亦配置 4096 個位元組片段。

JFS2 區塊
加強型日誌型檔案系統將磁碟空間分成區塊。JFS2 支援多種檔案系統區塊大小：512、1024、2048 及 4096。

不同的檔案系統可具有不同的區塊大小，但單一檔案系統內只能使用一個區塊大小。

藉由將資料更有效地儲存在檔案或目錄的部分邏輯區塊中，較小的區塊大小可使浪費的磁碟空間減至最小。較小的區塊大小亦可能導致作業額外負擔。此外，裝置驅動程式必須提供對等於或小於檔案系統區塊大小之磁碟區塊的定址能力。

因為對於區塊大小並非 4096 個位元組的檔案系統，會以較小單位配置磁碟空間，所以當檔案或目錄大小重複擴充時，會更頻繁地發生配置活動。例如，使零長度檔案之大小擴充 512 個位元組的寫入作業，會使得一個區塊被配置到該檔案（假設區塊大小為 512 個位元組）。如果藉由再次寫入 512 個位元組進一步擴充該檔案的大小，則必須再配置一個區塊到該檔案。將此範例應用到具有 4096 個位元組區塊的檔案系統時，僅會發生一次磁碟空間配置，作為首次寫入作業的一部分。不會執行其他配置活動來作為第二次寫入作業的一部分，因為最初的 4096 位元組區塊配置已夠大，可以容納第二次寫入作業所新增的資料。

使用「系統管理介面工具 (SMIT)」或 **crfs** 及 **mkfs** 指令建立檔案系統期間，會指定檔案系統區塊大小。選擇檔案系統區塊大小的原則應依據檔案系統所包含之檔案的預計大小。

可透過「系統管理介面工具 (SMIT)」或 **lsfs** 指令，來識別檔案系統區塊大小值。針對應用程式，可使用 **statfs** 子常式來指定檔案系統區塊大小。

區塊是磁碟空間配置的基本單位，檔案系統中每個區塊的配置狀態均記錄在檔案系統區塊配置對映內。可能需要更多的虛擬記憶體及檔案系統磁碟空間，以容納區塊大小小於 4096 個位元組之檔案系統的區塊配置對映。

可變的 *i-node* 數目
將磁碟空間分成小於 4096 個位元組的大小可使磁碟空間使用率最佳化， 但它會增加在檔案系統內儲存之小型檔案及目錄的數目。

不過，磁碟空間只是檔案及目錄所需的其中一種檔案系統的資源，每個檔案或目錄亦需要一個磁碟 i-node。

JFS 及 *i-node*
JFS 可讓您在想要大於或小於預設的磁碟 i-node 數時，指定在檔案系統內所建立的磁碟 i-node 數。

在建立檔案系統時，會將磁碟 i-node 數指定為稱為每 *i-node* 位元組數或 *NBPI* 的值。例如，如果 NBPI 值為 1024，就會為每 1024 個位元組的檔案系統磁碟空間建立一個磁碟 i-node。換言之，NBPI 值越小（如 512），i-node 數就會越多；而 NBPI 值越大（如 16,384），i-node 數就會越少。

對於 JFS 檔案系統，會針對配置到檔案系統之每 NBPI 個位元組的配置群組空間均建立一個 i-node。檔案系統中的 i-node 總數限制了檔案總數及檔案系統的總計大小。儘管仍配置每配置群組的完整 i-node 數目，還是可以部分配置一個配置群組。NBPI 與檔案系統中 i-node 的總數成反比。

JFS 會將所有檔案系統限制為 16M (2^{24}) i-node。

根據配置群組大小 (*agsize*) 的不同，可允許的 NBPI 值集亦不同。預設值是 8 MB。對於 8 MB 的 *agsize*，可容許的 NBPI 值為 512、1024、2048、4096、8192 及 16,384。可使用較大的 *agsize*。*agsize* 可容許的值為 8、16、32 及 64。隨著 *agsize* 的增加，可容許的 NBPI 值範圍也會按比例增加。如果 *agsize* 增加一倍到 16 MB，則 NBPI 值範圍也會加倍：1024、2048、4096、8193、16384 及 32768。

使用「系統管理介面工具 (SMIT)」或 **crfs** 及 **mkfs** 指令建立檔案系統期間，會指定片段大小及 NBPI 值。片段大小及為檔案系統所建立之 i-node 數目的原則， 是以檔案系統所預計包含的檔案數及其大小為基礎。

您可以使用「系統管理介面工具 (SMIT)」或 **lsfs** 指令，來識別片段大小及 NBPI 值。 請針對應用程式，使用 `statfs` 子常式來識別檔案系統片段大小。

JFS2 及 i-node
JFS2 依需要配置 i-node。

若檔案系統中有供其他 i-node 使用的空間，則會自動配置它們。因此，可用的 i-node 數目受檔案系統自身大小限制。

JFS 及 JFS2 大小限制
您於建立檔案系統時定義 JFS 的最大大小。定義 JFS 大小的原則基於數個重要的議題。

建議的 JFS2 最大大小是 16 TB。JFS2 的最小檔案系統大小為 16 MB。

儘管與使用預設配置單位 4096 個位元組相比，使用小於 4096 個位元組配置單位的檔案系統 需要的磁碟空間會小很多，但使用較小片段會引起效能損失。

檔案系統中每個片段 (JFS) 或區塊 (JFS2) 的配置狀態均記錄於檔案系統配置對映表中。 可能需要更多的虛擬記憶體及檔案系統磁碟空間，以容納片段或區塊大小小於 4096 個 位元組之檔案系統的配置對映表。

因為對於片段 (JFS) 或區塊 (JFS2) 大小並非 4096 個位元組的檔案系統，會以較小單位配置磁碟空間，所以當檔案或目錄大小重複擴充時，會更頻繁地發生配置活動。例如，使零長度檔案之大小擴充 512 個位元組的寫入作業，會導致一個 512 位元組片段或 區塊被配置到該檔案（取決於檔案系統類型）。若藉由再次寫入 512 個位元組進一步擴充 該檔案的大小，則必須再配置一個片段或區塊到該檔案。將此範例應用至具有 4096 位元組片段或 區塊的檔案系統時，作為首次寫入作業的一部分，僅會發生一次磁碟空間配置。 無需執行其他配置活動來作為第二次寫入作業的一部分，因為最初的 4096 位元組配置已夠大，可容納第二次寫入作業所新增的資料。若檔案一次擴充 4096 個位元組，則配置活動可減少到最少。

一個與大小相關的議題為檔案系統日誌的大小。

對於 JFS，於大部分案例中，多個檔案系統皆使用一個大小配置為 4 MB 的共用日誌。例如，初次安裝後，root 磁區群組中的所有檔案系統均使用邏輯磁區 hd8 作為共用 JFS 日誌。預設邏輯磁區分割區大小為 4 MB，且預設日誌大小為一個分割區，因此，root 磁區群組通常包含一個 4 MB JFS 日誌。當檔案系統超過 2 GB 或當使用單一日誌的檔案系統空間總量超過 2 GB 時，預設日誌大小可能不夠。在任一狀況下，日誌大小都會隨檔案系統大小的增加而按比例增加。日誌邏輯磁區的大小變更時，必須先執行 **logform** 指令來重新起始設定日誌，才能使用新的空間。JFS 日誌的大小最大值限制為 256 MB。

單一 JFS 日誌可支援之合併檔案系統的大小有實際限制。作為指導方針， 整個檔案系統容量一兆個位元組為單一 JFS 日誌的建議限制。 當超過或快要超過此指導方針，或 **logredo** 指令（由 **fsck** 指令呼叫）招致記憶體不足的錯誤時，請新增其他 JFS 日誌，然後在兩個 JFS 日誌檔之間共用負荷。

對於 JFS2，於大部分案例中，多個檔案系統亦皆使用一個共用日誌。當檔案系統超過 2 GB 或當使用單一日誌的檔案系統空間總量超過 2 GB 時，預設日誌大小可能不夠。 於此兩種情況下，您均可隨檔案系統大小的增加而按比例增加日誌大小，或新增其他 JFS2 日誌，然後在兩個 JFS2 日誌檔之間共用負荷。

JFS 大小限制
在建立 JFS 時，會定義該檔案系統大小的最大值。 NBPI、片段大小及配置群組大小均為決定原則的因素。

檔案系統大小限制為下列項目中的最小值：

$$NBPI * 2^{24}$$

或

$$FragmentSize * 2^{28}$$

比方說，如果您選取 NBPI 比例 512，則檔案系統大小會限制為 8 GB ($512 * 2^{24} = 8$ GB)。JFS 支援以下 NBPI 值：512、1024、2048、4096、8192、16384、32768、65536 及 131072。

JFS 會將所有檔案系統限制為 16M (2^{24}) i-node。

會針對配置至檔案系統之每 *NBPI* 個位元組的配置群組空間均建立一個 i-node。儘管仍配置每配置群組的完整 i-node 數目，還是可以部分配置一個配置群組。NBPI 與檔案系統中 i-node 的總數成反比。

JFS 會將檔案系統空間分離成 i-node 及磁碟區塊分組（用於使用者資料）。這些分組稱為配置群組。配置群組大小可在建立檔案系統時指定。配置群組大小為 8M、16M、32M 及 64M。每個配置群組大小均有一個相關的 NBPI 範圍。範圍由以下表格定義：

配置群組 大小 (以 MB 為單位)	**允許的 NBPI 值**
8	512, 1024, 2048, 4096, 8192, 16384
16	1024, 2048, 4096, 8192, 16384, 32768
32	2048, 4096, 8192, 16384, 32768, 65536
64	4096, 8192, 16384, 32768, 65536, 131072

JFS 支援四種連續磁碟空間的片段大小單位：512、1024、2048 及 4096 位元組。JFS 在 i-node 及間接區塊中以 28 位元數字維護片段位址。每個片段必須可使用從 0 到 (2^{28}) 的數字定址。

JFS2 大小限制
測試顯示維護包含較大型檔案的特大型 JFS2 檔案系統， 要比維護包含大量小型檔案的檔案系統來得實際。若大型檔案系統包含許多小型檔案，則 **fsck** 指令及其他檔案系統維護作業要花很長時間執行。

建議使用下列大小限制：

項目	**說明**
最大 JFS2 檔案系統大小：	32TB
最大 JFS2 檔案大小：	16TB
最小 JFS2 檔案系統大小：	16MB

JFS 可用空間片段化
對於 JFS 檔案系統，使用小於 4096 個位元組的片段會導致磁碟上可用空間中有較多碎塊。

例如，假設有一個磁碟區域被分成 8 個片段，每個片段 512 個位元組。假設不同檔案（每個需要 512 個位元組）已寫入此磁碟區域的第一、第四、第五及第七個片段，而保留第二、第三、第六及第八個片段可用。儘管代表 2048 個位元組磁碟空間的四個片段是可用的，卻不會對這些可用的片段配置部分邏輯區塊（需要四個片段，即 2048 個位元組），因為在單一配置中片段必須是連續的。

因為配置給檔案或目錄邏輯區塊的片段必須是連續的，所以即使可用空間總量足夠要求新磁碟空間的檔案系統作業使用，可用空間有碎塊亦會導致該作業失敗。例如，將零長度檔案擴充一個邏輯區塊的寫入作業，需要配置 4096 個位元組的連續磁碟空間。如果檔案系統可用空間有碎塊，且它是由 32 個非連續的 512 位元組碎塊（即總計 16 KB 的可用磁碟空間）組成，則寫入作業會失敗，因為 8 個連續的片段（即 4096 個位元組的連續磁碟空間）不可用於滿足寫入作業。

若 JFS 檔案系統之可用空間碎塊量達到難以管理的程度，可使用 **defragfs** 指令進行磁碟重整。執行 **defrags** 指令有助於增進效能。

稀疏檔案
檔案為一系列索引區塊。區塊從 i-node 對映至其代表之檔案的邏輯偏移。

具有未對映到資料區塊之一或多個索引的檔案稱為稀疏配置的或稀疏檔案。稀疏檔案具有與其相關聯的大小，但它不會使所有的資料區塊都被配置來滿足大小的基本要求。若要識別檔案是否為稀疏配置，請使用 **fileplace** 指令。其會指出檔案中目前未配置的所有區塊。

註： 於大部分狀況中，**du** 亦可用來 判斷配置至檔案的資料區塊數目是否與保留其大小之檔案所需的數目不相符。 壓縮的檔案系統對未稀疏配置的檔案顯示的行為可能是一樣的。

應用程式藉由探查目前配置索引之外的位置來擴充檔案時，會建立稀疏檔案，但寫入的資料不會佔用所有新指派的索引。新檔案的大小可反映檔案的最大寫入限度。

讀取具有未配置資料區塊之檔案的區段會導致傳回零緩衝區。 寫入具有未配置資料區塊之檔案的區段會導致配置必要的資料區塊並寫入資料。

此行為可影響檔案操作或保存指令。例如，下列指令不保留檔案的稀疏配置：

- **cp**
- **mv**
- **tar**
- **cpio**

註： 於使用 **mv** 的狀況下，此僅適用於將檔案移至另一個檔案系統。若在相同的檔案系統內移動檔案，則其仍會保持稀疏。

從之前指令複製或還原之檔案的結果是使每個資料區塊都被配置，因此即不再具有稀疏性質。不過，下列保存指令可保留稀疏性質，或主動讓檔案變稀疏：

- **backup**
- **restore**
- **pax**

因為可能會過量使用具有稀疏檔案之檔案系統的資源，所以使用及維護此類型的檔案時要特別注意。

***JFS* 及大型檔案**
您可以建立具有 JFS 檔案系統類型的大型檔案。

所有 JFS2 檔案系統都支援大型檔案。

為大型檔案所啟用的檔案系統可使用 **crfs** 及 **mkfs** 指令來建立。 這兩個指令皆具有選項 (bf=true)，用於指定為大型檔案啟用的檔案系統。您也可以使用 SMIT 來建立這些檔案系統。

在為大型檔案所啟用的檔案系統中，是以 4096 位元組區塊配置儲存於 4 MB 檔案偏移之前的檔案資料。以大小為 128 KB 的大型磁碟區塊配置儲存在 4 MB 檔案偏移以外的檔案資料。 大型磁碟區塊實際上是 32 個連續的 4096 位元組區塊。

例如，在一般檔案系統中，132 MB 檔案需要 33K 4 KB 磁碟區塊（33 個單一間接區塊， 每一個都以 1024 個 4 KB 磁碟位址填入）。為大型檔案所啟用之檔案系統中之一個 132 MB 檔案具有 1024 個 4-KB 磁碟區塊及 1024 個 128 KB 磁碟區塊。 對於 132 MB 檔案，大型檔案幾何佈置僅需要兩個單一間接區塊。 大型檔案及一般檔案類型均需要一個雙重間接區塊。

大型磁碟區塊需要 32 個連續的 4 KB 區塊。當您寫入超過 4 MB 的大型檔案時， 若檔案系統不包含 32 個未用的連續 4 KB 區塊，則檔案偏移會失敗並產生 ENOSPC。

註： 檔案系統可能具有成千上萬的可用區塊，但若其中有 32 個不連續，則配置會失敗。

defragfs 指令可重組磁碟區塊，以提供較大的連續可用區塊區域。

必須要有 JFS，才能起始設定所有新的磁碟配置。在系統上裝載第一個大型檔案啟用的檔案系統時，JFS 會啟動用來將起始檔案配置調整為零的核心 kproc 程序。為大型檔案啟用的檔案系統順利卸載後，kproc 程序仍會保留下來。

JFS 資料壓縮

JFS 支援分段及壓縮檔案系統，這些檔案系統藉由允許以小於完整區塊大小 4096 個位元組的單位或 「片段」，將邏輯區塊儲存在磁碟上，以節省磁碟空間。

JFS2 不支援資料壓縮。

在分段的檔案系統中，僅不大於 32KB 之檔案的最後邏輯區塊是以這種方式儲存，如此一來，片段支援僅有助於含有許多小型檔案的檔案系統。不過，資料壓縮允許將任意大小之檔案的所有邏輯區塊儲存成一或多個連續的片段。資料壓縮平均可以節省一半的磁碟空間。

不過，使用片段及資料壓縮確實會增加磁碟可用空間產生片段的可能性。配置到邏輯區塊的片段在磁碟上必須是連續的。其可用空間有碎塊的檔案系統可能難以找到足夠的連續片段用於邏輯區塊配置（即使可用片段的總數可能超過邏輯區塊需求）。JFS 提供了可藉由增大連續可用空間，對檔案系統進行「磁碟重整」的 **defragfs** 程式，以減少可用空間碎塊。此公用程式可用於分段及壓縮的檔案系統。片段及資料壓縮可節省大量的磁碟空間，而可用空間碎塊問題也得到了控制。

現行 JFS 中的資料壓縮與此作業系統的先前版本相容。在 JFS 的兩個版本中， 由所有系統呼叫組成的 API 保持不變。

JFS 資料壓縮實作

資料壓縮是檔案系統的一個屬性，在使用 **crfs** 或 **mkfs** 指令建立檔案系統時所指定。您可以使用 SMIT 來指定資料壓縮。

小心：不可壓縮根檔案系統 (/)。因為 **installp** 必須能準確計算更新資料及新安裝架構的大小，所以不建議壓縮 /usr 檔案系統。

壓縮僅適用於這類檔案系統中的一般檔案及長符號鏈結。片段支援仍繼續適用於未壓縮的目錄及中繼資料。檔案的每個邏輯區塊在被寫入磁碟之前會自行壓縮。這種方式的壓縮有助於隨機探查及更新，並且與以較大單位壓縮資料相比，僅會損失少量可用磁碟空間。

壓縮後，一個邏輯區塊通常需要小於 4096 個位元組的磁碟空間。會將壓縮的邏輯區塊寫入磁碟，且僅會配置儲存它所需的連續片段數目。如果某個邏輯區塊未壓縮，則會以未壓縮的格式將它寫入磁碟，並為它配置 4096 個位元組的連續片段。

lsfs -q 指令顯示壓縮的現行值。您也可以使用 SMIT 來識別資料壓縮。

JFS 資料壓縮的隱含行為

因為寫入檔案的程式並未預期到在順利寫入（或順利儲存對映檔）後，會出現空間不足 (ENOSPC) 狀況，所以有必要保證在將邏輯區塊寫入磁碟時有可用的空間。

可採用這樣的方法：在首次修改某個邏輯區塊時，為它配置 4096 個位元組，以便即使不壓縮該區塊也會有可用的磁碟空間。如果沒有 4096 位元組配置，系統會傳回 ENOSPC 或 EDQUOT 錯誤狀況，即使可能有足夠的磁碟空間來容納壓縮的邏輯區塊。如果在接近磁碟限額限制或檔案系統幾乎已滿時進行作業，最有可能發生過早報告空間不足的狀況。

壓縮檔案系統還可能表現下列行為：

- 因為一開始對邏輯區塊配置了 4096 個位元組，所以某些系統呼叫可能會收到 ENOSPC 或 EDQUOT 錯誤。例如，可能使用 mmap 系統 呼叫對映了某個舊檔案，且在先前寫入的位置執行儲存作業會導致 ENOSPC 錯誤。
- 進行了資料壓縮後，會保持對修改過的區塊配置完整磁碟區塊，直到將它寫入磁碟為止。如果對該區塊先前確認的配置小於完整區塊，則該區塊所佔用的磁碟空間量是兩者的總和，且直到已確認檔案 (i-node) 之後，才會釋放先前的配置。這種情況是針對正常片段。 於先前確認配置之檔案內邏輯區塊的數目，對於正常片段而言最多是一個，但壓縮後可與檔案內區塊的數目相同。
- 在應用程式執行 fsync 或 sync 系統呼叫之後，才會釋放邏輯區塊的任何先前確認資源。
- stat 系統呼叫指出配置給某個檔案的片段數目。所報告的數目是基於配置給已修改但未寫入之區塊的 4096 個位元組及未修改區塊的已壓縮大小。stat 系統呼叫不會對先前確認的資源計數。 若未壓縮任何修改過的區塊，於 i-node 確認作業後，stat 系統呼叫會報告正確的已配置片段數。同樣地，現行配置會佔

用磁碟限額。由於檔案的邏輯區塊會寫入磁碟，如果進行壓縮，配置給它們的片段數會減少，從而會變更磁碟限額及 stat 的結果。

JFS 資料壓縮演算法
壓縮演算法是 LZ 的 IBM® 版本。一般而言，LZ 演算法壓縮資料的方式是，以識別給定字串首次出現位置及其長度的指標，來代表該字串的第二次及以後的出現。

壓縮處理程序一開始時，尚未識別任何字串，因此必須至少以「原始」字元（需要 9 位元）來代表資料的第一個位元組（0、位元組）。 壓縮給定數量的資料（如 *N* 個位元組）後，壓縮程式會在 *N* 個位元組中 搜尋與下一個未處理位元組開始處字串相符的最長字串。如果最長的相符字串長度為 0 或 1，則會將下一個位元組編碼為「原始」字元。否則，會以（指標、長度）配對來代表該字串，且所處理的位元組數會按長度遞增。在架構上，IBM LZ 可支援 *N* 值為 512、1024 或 2048。IBM LZ 會指定（指標、長度）配對及原始字元的編碼。指標是大小為 log2 *N* 的固定長度欄位，而其長度被編碼為可變長度欄位。

JFS 資料壓縮的效能成本
因為資料壓縮是片段支援的擴充，所以與片段相關的效能亦適用於資料壓縮。

壓縮的檔案系統亦會以下列方式影響效能：

- 壓縮及解壓縮資料需要大量的時間，如此一來，對於部分使用者環境而言，壓縮檔案系統的可用性可能受到限制。
- 大部分的 UNIX 一般檔案只會寫入一次， 不過有些檔案會適時地更新。對於後者，資料壓縮會導致額外的效能損失，因為必須在首次修改邏輯區塊時配置 4096 個位元組的磁碟空間，然後在將該邏輯區塊寫入磁碟後重新配置磁碟空間。 對於非壓縮檔案系統中的一般檔案而言，不需要此額外配置活動。
- 資料壓縮會增加處理器循環的次數。對於軟體壓縮程式而言，壓縮循環次數約為每位元組 50 次循環，解壓縮循環次數約為每位元組 10 次循環。

JFS* 線上備份及 *JFS2 Snapshot
您可以製作 JFS 檔案系統或 JFS2 檔案系統的時間點映像檔，然後作為備份使用。然而，針對每一個檔案系統類型，此映像檔的需求及行為存在一些差異。

對於 JFS 檔案系統，您可分割檔案系統之鏡映副本的唯讀靜態副本。一般而言， 鏡映副本會隨原始檔案系統的更新而更新，但此時間點副本則保持不變。它仍為製作副本所處時間點的穩定映像檔。若將此映像檔用於備份，則備份副本中可能沒有於您開始建立映像檔程序之後開始的所有修改。因此在執行分割時，建議檔案系統活動應減至最低。 備份副本中將沒有分割後所發生的任何變更。

若為 JFS2 檔案系統，時間點映像檔稱為 *Snapshot*。Snapshot 會保持靜態，並且會保留製作 Snapshot 時與原始檔案系統（稱為 *snappedFS*）相同的安全許可權。同樣地，您也可以建立 JFS2 Snapshot，而無需卸載或停止檔案系統。您可以使用 JFS2 Snapshot，將它用作檔案系統的線上備份，以存取檔案或目錄（如果在製作 Snapshot 時檔案或目錄存在），或備份抽取式媒體。請注意 JFS2 Snapshot 的下列相關內容：

- 當系統重新開機時，會改寫 root (/) 或 /usr 檔案系統。在重新開機之前卸載檔案系統，即可保留其他檔案系統的 Snapshot。AIX 5.2（含 5200-01） 中所建立的 Snapshot 是可回復的。當 **fsck** 或 **logredo** 在具有 AIX 5.2（含 5200-01） 上所建立之 Snapshot 的 JFS2 檔案系統上執行時，將保留 Snapshot。具有 AIX 5.2 建立之 Snapshot 的徹底卸載檔案系統裝載到 AIX 5.2（含 5200-01） 系統之後也可回復。
- 不建議您對具有 Snapshot 的檔案系統執行 **defragfs** 指令。在重組期間移動的每個區塊都必須複製到 Snapshot，這既浪費時間又浪費 Snapshot 邏輯磁區的空間。
- 如果 Snapshot 耗盡空間，則會刪除該 snappedFS 的所有 Snapshot。此失敗會將項目寫入錯誤日誌。
- 如果寫入 Snapshot 失敗，則會刪除該 snappedFS 的所有 Snapshot。此失敗會將項目寫入錯誤日誌。
- 無法在 AIX 5.2 系統上存取於 AIX 5.2（含 5200-01） 系統上建立或存取的 Snapshot。您必須先刪除這些 Snapshot，才能裝載檔案系統。
- 無法從 AIX 5.2（含 5200-01） 之前的任何版本存取 Snapshot 位在 AIX 5.3 上的 JFS2 檔案系統。若要將系統移回，必須先刪除 Snapshot，讓檔案系統能被存取。

***JFS* 線上備份**
您可以製作 JFS 檔案系統的時間點映像檔，然後作為備份使用。

對於 JFS 檔案系統，您可分割檔案系統之鏡映副本的唯讀靜態副本。一般而言， 鏡映副本會隨原始檔案系統的更新而更新，但此時間點副本則保持不變。它仍為製作副本所處時間點的穩定映像檔。若將此映像檔用於

備份，則備份副本中可能沒有於您開始建立映像檔程序之後開始的所有修改。因此在執行分割時，建議檔案系統活動應減至最低。 備份副本中將沒有分割後所發生的任何變更。

JFS2 Snapshot

您可以製作 JFS2 檔案系統的時間點映像檔，然後作為備份使用。

JFS2 檔案系統的時間點映像檔稱為 *Snapshot*。Snapshot 保持靜態且保留製作 Snapshot 時與原始檔案系統（稱為 *snappedFS*）相同的安全許可權。同樣，您亦可以建立 JFS2 Snapshot，而無需卸載或停止檔案系統。您可以使用 JFS2 Snapshot，以：

- 依照檔案或目錄在製作 Snapshot 時的原樣來存取它們。
- 備份到抽取式媒體。

JFS2 Snapshot 類型有兩種類型：內部及外部。JFS2 外部 Snapshot 建立於檔案系統的不同邏輯磁區中。外部 Snapshot 可以從位於其唯一裝載點的檔案系統進行獨立裝載。

JFS2 內部 Snapshot 與檔案系統建立於相同的邏輯磁區中，並會從檔案系統配置區塊。內部 Snapshot 可以從含有 Snapshot 之 JFS2 檔案系統根目錄的隱藏 `.snapshot` 目錄中存取。建立檔案系統時，必須啟用 JFS2 檔案系統以支援內部 Snapshot。

JFS2 Snapshot 不支援檢查檔案系統限額。您無法使用 Snapshot 上的 **repquota** 指令來判定限額的狀態。如果您將檔案系統映像檔回復為 Snapshot 映像檔，則會保留時間點限額資訊。請注意下列 JFS2 外部 Snapshot 及 JFS2 內部 Snapshot 特定的注意事項：

- 無法在 AIX 5.2 系統上存取於 AIX 5.2（含 5200-01） 系統上建立或存取的外部 Snapshot。您必須先刪除這些 Snapshot，才能裝載檔案系統。
- 無法從 AIX 5.2（含 5200-01） 之前的任何版本存取 Snapshot 位在 AIX 5.3 上的 JFS2 檔案系統。若要將系統移回，必須先刪除 Snapshot，讓檔案系統能被存取。
- 因為於磁碟重整期間還必須將移動的每個區塊都複製到 Snapshot，這既浪費時間又浪費 Snapshot 邏輯磁區的空間，所以不建議您針對含有外部 Snapshot 的 JFS2 檔案系統執行 **defragfs** 指令。
- 如果某個外部 Snapshot 耗盡空間，或者某個外部 Snapshot 失敗，則該 snappedFS 的所有外部 Snapshot 都會標記為無效。對 Snapshot 的進一步存取將失敗。這些失敗會將項目寫入錯誤日誌。

內部 JFS2 Snapshot 注意事項：

- 當在含有內部 Snapshot 的 JFS2 檔案系統上執行 **logredo** 指令時，會保留內部 Snapshot。
- 如果 **fsck** 指令必須修改 JFS2 檔案系統來修復內部 Snapshot，則會移除該內部 Snapshot。
- 如果某個內部 Snapshot 耗盡空間，或者寫入某個內部 Snapshot 失敗，則該 snappedFS 的所有內部 Snapshot 都會標記為無效。對內部 Snapshot 的進一步存取將失敗。這些失敗會將項目寫入錯誤日誌。
- 內部 Snapshot 不可個別裝載。建立內部 Snapshot 之後，您可以立即在檔案系統根目錄的 `.snapshot` 目錄中存取它們。因此，您可以透過 NFS 伺服器存取內部 Snapshot，而無需匯出 Snapshot 的個別裝載點。
- 內部 Snapshot 與 AIX 6.1 之前的 AIX 版本不相容。建立以支援內部 Snapshot 的 JFS2 檔案系統無法在舊版 AIX 中進行修改。
- 建立以支援內部 Snapshot 的 JFS2 檔案系統還可以支援「延伸屬性第 2 版」。
- 含有內部 Snapshot 的 JFS2 檔案系統無法與「資料管理應用程式設計介面 (DMAPI)」一起使用。
- 您無法針對含內部 Snapshot 的 JFS2 檔案系統使用 **defragfs** 指令。
- **readdir()** 系統呼叫不會傳回 `.snapshot` 目錄。這可防止對 Snapshot 的非預期造訪。任何系統呼叫或根據 **readdir()** 系統呼叫的指令都不會傳回 `.snapshot` 目錄（例如，**/bin/pwd** 指令及 `.snapshot` 目錄的 **getcwd()** 系統呼叫都找不到上層目錄）。

相容性及移轉

JFS 檔案系統在 AIX 5.1 及 AIX 5.2 中完全相容。 此作業系統之先前受支援版本與現行 JFS 相容，儘管具有非預設片段大小、NBPI 值或配置群組大小的檔案系統，於移轉至先前版本時可能需要特別注意。

註： 磁區大小為 4 KB 的磁碟不支援 JFS 檔案系統。因此，當您建立檔案系統或執行備份作業時，請確定磁碟的磁區大小不是 4 KB。

JFS2 檔案系統（Snapshot 除外）在 AIX 5.1 及 AIX 5.2 中相容， 但與作業系統的先前版本不相容。AIX 5.1 不支援包含 Snapshot 的 JFS2 檔案系統。在恢復為舊版 AIX 之前，一定要將所有 JFS2 檔案系統完全卸載，因為 **logredo** 指令不一定能在為較新版次建立的檔案系統上執行。

註 : 在 AIX 之前的版本上， 不能存取建立或轉換為 v2 格式的 JFS2 檔案系統。

下列清單說明可能導致在先前版本作業系統下所建立之檔案系統出現問題的各個因素：

JFS 檔案系統映像檔
以預設片段大小及 NBPI 值（4096 個位元組），及預設配置群組大小 (agsize) 8 所建立的 任一 JFS 檔案系統映像檔，均可與在此作業系統之 AIX 4.3 及更新版本下所建立的 JFS 檔案系統映像檔交換，而無需任何特殊移轉活動。

附註：JFS2 Snapshot : 無法在舊版中存取在 AIX 5L 5.2 版（含 5200-01 建議維護套件） 中建立或存取的 JFS2 Snapshot。您必須先刪除這些 Snapshot，才能裝載檔案系統。

JFS 檔案系統之間的備份及還原
可於含不同區塊大小的 JFS 檔案系統間執行備份及還原順序。不過，由於磁碟使用率增加，若來源檔案系統的區塊大小小於目標檔案系統的區塊大小， 還原作業可能會因缺少可用的區塊而失敗。這種情況在執行完整檔案系統備份及還原順序時尤其可能發生，甚至在目標檔案系統的總檔案系統大小大於來源檔案系統總大小時也會發生。

由於壓縮檔案系統會增強磁碟使用率，而從已壓縮到非壓縮檔案系統、或在含不同片段大小之已壓縮檔案系統間，可執行備份及還原的一連串動作，但還原作業可能會因磁碟空間不足而失敗。這種情況在執行完整檔案系統備份及還原順序時尤其可能發生，甚至在目標檔案系統的總檔案系統大小大於來源檔案系統的總大小時也可能發生。

JFS 及 JFS2 裝置驅動程式限制
裝置驅動程式必須提供對等於或小於 JFS 檔案系統片段大小或 JFS2 區塊大小之磁碟區塊的定址能力。例如，如果在使用者提供的 RAM 磁碟裝置驅動程式上建立了 JFS 檔案系統，則該驅動程式必須讓 512 位元組區塊包含具有 512 位元組片段的檔案系統。如果該驅動程式僅允許頁面層次的定址能力，則只能使用片段大小為 4096 個位元組的 JFS。

將 JFS 複製至另一個實體磁區
您可以將 JFS 檔案系統複製至另一個實體磁區，同時仍保持檔案系統的完整性。

下面的範例情節說明如何將 JFS 檔案系統複製到不同的實體磁區，同時還保持檔案系統的完整性。

此作法範例情節中的資訊已使用特定版本的 AIX 進行測試。依據您的 AIX 版本及層次，所得到的結果可能大不相同。

若要將 JFS 檔案系統複製至另一個實體磁區，同時還保持檔案系統的完整性，請執行下列動作：

1. 停止您要複製之檔案系統的活動。除非正在使用檔案系統的應用程式處於靜止狀態，或檔案系統處於您已知的狀態，否則您無法知道備份的資料中有什麼。
2. 鏡映邏輯磁區，方法是在指令行上鍵入下列 SMIT 捷徑：

   ```
   smit mklvcopy
   ```

3. 使用下列指令複製檔案系統：

   ```
   chfs -a splitcopy=/backup -a copy=2 /testfs
   ```

   -a 旗標的 **splitcopy** 參數會導致該指令分割檔案系統的鏡映副本，並在新裝載點以唯讀形式裝載之。此動作會提供檔案系統的副本，以及可用於備份的一致日誌型 meta 資料。
4. 如果您要將鏡映副本移至其他裝載點，請使用下列 SMIT 捷徑：

   ```
   smit cplv
   ```

此時，便可以使用檔案系統副本了。

CD-ROM 檔案系統和 UDF 檔案系統
CD-ROM 檔案系統 (CDRFS) 是唯讀的本端檔案系統實作，它可以儲存在 CD-ROM 媒體、CD-RW 媒體（如果是防寫保護的話）及 DVD-ROM 媒體上。不論使用的媒體為何，CDRFS 檔案大小上限都是 2 GB。「通用磁

碟格式 (UDF)」檔案系統是可寫入的本端檔案系統實作，它能以唯讀的形式儲存在 DVD-ROM 媒體上，或以讀寫的形式儲存在 DVD-RAM 媒體中。

預設會自動裝載 CD，但可停用此特性。若停用了此特性，請使用 **cdmount** 指令來裝載 CDRFS 檔案系統。

AIX 支援下列 CDRFS 磁區及檔案結構格式：

類型	**說明**
ISO 9660:1988(E) 標準	CDRFS 支援 ISO 9660 層次 3 交換及層次 1 施行。
High Sierra Group 規格	在 ISO 9660 之前，並提供與先前 CD-ROM 之間的向後相容性。
Rock Ridge Group Protocol	指定 ISO 9660 的擴充，這些擴充完全符合 ISO 9660 標準，並根據 System Use Sharing Protocol (SUSP) 及 Rock Ridge Interchange Protocol (RRIP) 提供完整的 POSIX 檔案系統語意，讓您可以裝載/存取 CD-ROM，如同在其他 UNIX 檔案系統中使用一樣。
CD-ROM eXtended Architecture 檔案格式（僅有 Mode 2 Form 1 磁區格式）	「CD-ROM eXtended Architecture (XA)」檔案格式會指定用於 CD-ROM 型多媒體應用程式（例如，照片 CD）的 ISO 9660 擴充。

所有磁區及檔案結構格式適用下列限制：

· 僅單一磁區的磁區集

· 僅非交錯的檔案

CDRFS 與基礎 CD-ROM 裝置驅動程式相依，以提供實體磁區格式（CD-ROM Mode 1 及 CD-ROM XA Mode 2 Form 1）及磁碟多階段作業格式（對映最後階段作業之磁區辨識區域的磁區描述子集）的透通性。

註 : 必須先將 CDRFS 從系統卸載，您才能移除 CD-ROM 媒體。

有另一個支援的檔案系統類型（稱為 UDFS），它是儲存在 DVD-ROM 媒體上的唯讀檔案系統。必須先將 UDFS 從系統卸載，您才能移除該媒體。AIX 支援 UDFS 格式 1.50、2.00 及 2.01 版。

必須在唯讀模式中使用 NFS 來匯出 UDFS。 不支援寫入 NFS 裝載的 UDFS。

若要使用 **cdmount** 指令來自動裝載讀取/寫入 UDFS，請編輯 `cdromd.conf` 檔案。您還可以使用 **mount** 指令手動裝載讀取/寫入 UDFS。

目錄

目錄 為一獨特的檔案類型，其中只包含存取檔案或其他目錄所需的資訊。其結果是，目錄比其他檔案類型佔用較少的空間。

檔案系統 包含目錄群組與目錄中的檔案。檔案系統一般都是以顛倒樹狀表示。根目錄是以斜線(/) 符號來表示，它定義了檔案系統，並且出現在檔案系統樹狀圖的頂端。

在樹狀圖中，目錄從根目錄開始向下分支，並包含檔案及子目錄。分支經由目錄結構建立一些唯一的路徑，來指向檔案系統中的每一個物件。

檔案集合儲存在目錄中。這些檔案集合通常彼此相關； 將它們儲存在同一目錄結構中，以方便組織管理。

檔案 是可讀取或寫入的資料的集成。檔案可以是一您所建立的程式、您所寫入的文字、您所要求的資料或您所使用的裝置。指令、印表機、終端機、通訊與應用程式都儲存在檔案中。如此可讓使用者以統一的方法來存取各種系統元素，並提供檔案系統很大的彈性。

目錄可讓您將檔案與其他目錄分組，將檔案系統組織成模組化階層，這會為檔案系統結構提供彈性及深度。

目錄包含目錄項目。每一項目包含檔案或子目錄名稱，與一索引節點參考號碼（*inode* 號碼）。若要提高速度，並加強磁碟空間的使用，可將檔案中的資料儲存在電腦記憶體中不同的位置上。inode 號碼包含用以尋找與檔案有關的所有被分散的資料區塊。inode 號碼也會記錄檔案的其他相關資訊，包括修改與存取時間、存取模式、鏈結數目、檔案擁有者和檔案類型。

一組特殊的指令可控制目錄。例如，您可以使用 **ln** 指令來建立目錄項目， 將許多檔案名稱鏈結至同一個 inode 號碼。

由於目錄經常包含應該無法讓所有系統使用 者使用的資訊，因此目錄的存取應可受到保護。藉由設定目錄許可權，您可控制 存取目錄人員，與決定可修改目錄內資訊的使用者名單。

目錄類型
目錄可由作業系統、系統管理者或使用者來定義。

系統定義的目錄包含特定的系統檔案種類，如指令等。檔案系統階層頂端 為系統定義的 / (根) 目錄。/ (根) 目錄通常包含下列標準系統相關目錄：

項目	說明
/dev	包含特殊的輸入／輸出 (I/O) 裝置檔案。
/etc	包含系統起始設定與系統管理檔案。
/home	包含登入系統使用者目錄。
/tmp	包含會在指定的天數後自動刪除的暫存檔。
/usr	包含 lpp、inclu de 和其他系統目錄。
/usr/bin	包含使用者可執行的程式。

某些目錄，如您的登入或起始目錄 ($HOME)，可由系統管理者定義與自訂。當您登入至作業系統，登 入目錄為現行目錄。

您建立的目錄稱為使用者定義的目錄。這些目錄可讓您組織與維護您的檔案。

目錄組織
目錄包含檔案、子目錄或此二者的組合。子目錄為目錄中的目錄。包含子目錄的目錄稱為母目錄。

每一個目錄都有一個代表了建立它的母目錄之項目，.. （點點），也會有一個用來表示這個目錄本身的項目，. （點）。大部分目錄清單中，這些檔案都已隱藏。

目錄樹
目錄的檔案系統結構很輕易的就會變得複雜。請嘗試將檔案與目錄結構盡可能簡化的保存。以容易識別的名稱來建立檔案與目錄。者將可使檔案能更輕易的使用。

母目錄
除了 / (根) 目錄以外，每一個目錄都有一個母目錄，並且可能會有子目錄。

起始目錄
當您登入時，系統將會將您放置在稱為起始目錄或登入目錄的目錄中。系統管理者會為每一個使用者設定此目錄。起始目錄是您個人檔案的儲存庫。一般而言，您所建立來供自己使用的目錄，會是您的起始目錄的子目錄。若要隨時回到您的起始目錄，請在提示下輸入 **cd** 指令然後按 Enter 鍵。

工作目錄
您一定是在目錄中工作。不論您目前工作的目錄為何，均稱為現行或工作目錄。**pwd**（表示工作目錄）指令會報告您工作目錄的名稱。請使用 **cd** 指令來變更工作目錄。

目錄命名慣例
每一目錄名稱必須為所儲存目錄中唯一的。這將確保目錄在檔案系統中會有唯一的路徑名稱。

目錄會遵循與檔案相同的命名慣例，如檔案命名慣例所述。

目錄縮寫
縮寫提供一種方便的方法來指定特定目錄。

以下是縮寫清單：

縮寫	意義
.	現行工作目錄。

縮寫	意義
..	現行工作目錄之上的目錄（現行目錄的母項）。
~	您的起始目錄（對 Bourne shell 來說則不是如此。如需相關資訊，請參閱 Bourne Shell。）
$HOME	您的起始目錄（對所有的 shell 來說都是如此。）

目錄路徑名稱
每一檔案與目錄可經由檔案系統樹狀結構 中的唯一路徑進入，稱為路徑名稱。路徑名稱將指定目錄或檔案在檔案 系統中的位置。

註 : 路徑名稱的長度不能超出 1023 個字元。

檔案系統使用下列兩種路徑名稱：

項目	說明
絕對路徑名稱	從 / (根) 目錄追蹤路徑。絕對路徑名稱一定要以斜線 (/) 符號開頭。
相對路徑名稱	從現行目錄經由其母項，或其子目錄與檔案追蹤路徑。

絕對路徑名稱代表從 / (根) 目錄以下的目錄完整名稱或檔案。不論您在檔案系統中的工作位置為何，您都可以 藉由指定其絕對路徑名稱，尋找到目錄或檔案。絕對路徑名稱以斜線 (/) 開始，此符號代表根目錄。路徑名稱 /A/D/9 為 9 的絕對路徑名稱。第一個斜線 (/) 代表 / (根) 目錄， 為搜尋工作開始的位置。路徑名稱其餘部分的搜尋方向依序為 A、D， 最後是 9。

可存在兩個名稱都是 9 的檔案，因為這兩個檔案的絕對路徑名稱在檔案系統中都是唯一的。路徑名稱 /A/D/9 與 /C/E/G/9 是指定兩個名為 9 的唯一檔案。

不像完整路徑名稱一般，相對路徑名稱將依 據現行工作目錄，指定一目錄或檔案。若為相對路徑名稱，您可以使用表示法點點 (..) 在檔案系統階層中向上移動。點點 (..) 代表上層目錄。由於相對路徑名稱指定開始於現行目錄的路徑，而不是以斜線 (/) 開始。 相對路徑名稱是用來指定現行目錄中的檔案名稱，或檔案或目錄在檔案系統中的 現行目錄層次之上或之下的路徑名稱。如果 D 是現行目錄，則存取 10 的相對路徑名稱為 F/10。但是，絕對路徑名稱一律為 /A/D/F/10。同時，存取 3 的相對路徑名稱為 ../../B/3。

您也可以使用表示法點 (.) 來代表現行目錄的名稱。點 (.) 表示法在執行一些會讀取現行目錄名稱的程式時普遍使用。

建立目錄（mkdir 指令）
使用 **mkdir** 指令，可以建立 *Directory* 參數所指定的一或多個目錄。

每一個新的目錄都會包含標準的項目點 (.) 及點點 (..)。您可以用 **-m** 模式 旗標來指定新目錄的許可權。

當您建立目錄時，會建立在現行或工作目錄中，除非您將絕對路徑名稱指定成檔案系統中的其他位置。

下列範例說明如何使用 **mkdir** 指令：

- 若要在現行工作目錄下建立名為 Test 的新目錄，並使用預設許可權，請鍵入：

```
mkdir Test
```

- 若要在先前建立的 /home/demo/sub1 目錄下，建立名為 Test 的目錄，並且許可權為 rwxr-xr-x，請鍵入：

```
mkdir -m 755 /home/demo/sub1/Test
```

- 若要在 /home/demo/sub2 目錄下建立名為 Test 的目錄，並使用預設許可權，請鍵入：

```
mkdir -p /home/demo/sub2/Test
```

-p 旗標會建立 /home、/home/demo 及 /home/demo/sub2 目錄（如果這些目錄不存在的話）。

請參閱 *Commands Reference, Volume 3* 中的 **mkdir** 指令，以取得完整語法。

移動或重新命名目錄（mvdir 指令）

使用 **mvdir** 指令來移動或更改目錄的名稱。

以下是如何使用 **mvdir** 指令的範例：

- 若要移動目錄，請鍵入：

```
mvdir book manual
```

  如此會將 book 目錄移至 manual 目錄底下（如果 manual 目錄存在）。否則，book 目錄將重新命名為 manual。

- 若要移動並重新命名目錄，請鍵入：

```
mvdir book3 proj4/manual
```

  如果名稱為 manual 的目錄已經存在，則會將 book3 及其內容移至 proj4/manual。換言之，book3 會變成 proj4/manual 的子目錄。如果 manual 不存在，則會將 book3 目錄重新命名為 proj4/manual。

請參閱 *Commands Reference, Volume 3* 中的 **mvdir** 指令，以取得完整語法。

顯示現行目錄（pwd 指令）

使用 **pwd** 指令，可以將現行目錄的完整路徑名稱（從 / (根) 目錄開始）寫入標準輸出。

所有目錄都會以一個斜線 (/) 隔開。/ (根) 目錄是以第一個斜線 (/) 來表示， 最後一個指名的目錄則是您的現行目錄。

例如，若要顯示現行目錄，請鍵入：

```
pwd
```

您現行目錄的完整路徑名稱類似下列：

```
/home/thomas
```

變更至其他目錄（cd 指令）

使用 **cd** 指令，從您目前的目錄移動至其他目錄。您必須具有指定目錄中 的執行（搜尋）許可。

如果未指定 *Directory* 參數，則 **cd** 指令會將您移至您的登入目錄（**ksh** 及 **bsh** 環境中的 $HOME，或 **csh** 環境中的 $home）。如果指定的目錄為一完整路徑名稱，則將成為現行目錄。完整路徑名稱若以斜線 (/) 開始，是表示 / (根) 目錄，若以點 (.) 開始， 是表示現行目錄，或若以點點 (..) 開始，則表示母目錄。如果目錄名稱不是完整路徑名稱，則 **cd** 指令將在由 *$CDPATH* shell 變數（或 *$cdpath* **csh** 變數）所指定的路徑中搜尋相關項目。此變數的語法與 *$PATH* shell 變數（或 *$path* **csh** 變數）相同，且語意相似。

以下是 **cd** 指令的用法範例：

- 若要變更為起始目錄，請鍵入：

```
cd
```

- 若要變更為 /usr/include 目錄，請鍵入：

```
cd /usr/include
```

- 若要往下移一層目錄樹至 sys 目錄，請鍵入：

```
cd sys
```

  如果現行目錄為 /usr/include，且包含一個名為 sys 的子目錄，則 /usr/include/sys 將成為現行目錄。

- 若要往上移一層目錄樹，請鍵入：

```
cd ..
```

特殊檔案名稱，點點 (..)，是指現行目錄的上一層目錄，亦即它的母目錄。

請參閱 *Commands Reference, Volume 1* 中的 **cd** 指令，以取得完整語法。

複製目錄（cp 指令）

使用 **cp** 指令，把 *SourceFile* 或 *SourceDirectory* 參數所指定的檔案或目錄內容副本，建立到 *TargetFile* 或 *TargetDirectory* 參數所指定的檔案或目錄。

如果指定成 *TargetFile* 的檔案已存在，則副本將會改寫檔案的原始內容。如果您正在複製一個以上的 *SourceFile*，目標端必須為一個目錄。

若要把 *SourceFile* 的副本放置在一個目錄內，請對 *TargetDirectory* 參數指定一個已存在目錄的路徑。除非您在路徑的尾端指定好一個新檔名，否則在複製檔案到一個目錄時，會維持它們個別的名稱。如果您指定了 **-r** 或 **-R** 旗標，**cp** 指令也可以把整個目錄複製到其他目錄內。

以下是 **cp** 指令的用法範例：

- 若要將 /home/accounts/customers/orders 目錄中的所有檔案複製到 /home/accounts/customers/shipments 目錄，請鍵入：

```
cp /home/accounts/customers/orders/* /home/accounts/customers/shipments
```

  這會將檔案（而不是目錄）從 orders 目錄複製到 shipments 目錄。

- 若要將目錄，包括其所有檔案和子目錄，複製到另一個目錄，請鍵入：

```
cp -R /home/accounts/customers /home/accounts/vendors
```

  這樣會將 customers 目錄（包括其所有檔案、 子目錄及子目錄中的所有檔案）複製到 vendors 目錄中。

請參閱 *Commands Reference, Volume 1* 中的 **cp** 指令，以取得完整語法。

顯示目錄內容（ls 指令）

使用 **ls** 指令來顯示目錄的內容。

ls 指令會將每一個指定的 *Directory* 內容，或每一個指定的 *File* 名稱，以及任何使用旗標所要求的資訊，一起寫入標準輸出。如果您沒有指定 *File* 或 *Directory*，**ls** 指令將會顯示現行目錄內容。

就預設值而言，**ls** 指令將依檔名字母順序，顯示所有資訊。如果是以具有 root 授權的使用者身分來執行指令， 根據預設值，它會使用 **-A** 旗標來列出所有項目，除了點 (.) 以及點點 (..) 以外。若要顯示檔案的所有項目（包括那些以點 (.) 來開頭的項目），請使用 **ls -a** 指令。

您可以用下列方法將輸出格式化：

- 使用 **-l** 旗標，每一行列出一個項目。
- 指定 **-C** 或 **-x** 旗標，以多欄方式列出項目。當輸出至 tty 時，**-C** 旗標為預設格式。
- 指定 **-m** 旗標，列出以逗號隔開的一系列項目。

要判斷輸出行中的字元位置數目時，**ls** 指令會使用 *$COLUMNS* 環境變數。如果此變數未設定，則指令將讀取 terminfo 檔案。如 果 **ls** 指令無法以這些方法判斷字元位置數字，將使用預設值 80。

以 **-e** 與 **-l** 旗標顯示的資訊將解譯如下：

每個項目的第一個字元可能會是下列其中一項：

項目	說明
d	項目為一目錄。
b	項目為一區塊特殊檔案。
c	項目為一字元特殊檔案。
l	項目為一符號鏈結。

項目	說明
p	項目為一先進先出法 (FIFO) 管線特殊檔案。
s	項目為一區域 Socket。
-	項目為一一般檔案。

下九個字元將區分為三組，每組三個字元。前三個字元顯示檔案或目錄擁有者的許可權。接下來的三個字元顯示群組中其他使用者的許可權。最後三個字元顯示可存取檔案的任何其他人之許可權。每一組的三個字元都顯示檔案的讀取、寫入與執行許 可。目錄的執行許可可讓您搜尋目錄中指定的檔案。

許可權表示如下：

項目	說明
r	授與讀取許可
t	只有目錄擁有者或檔案擁有者可以刪除或更名該目錄內的檔案，即使他 人具有該目錄的寫入許可。
w	授與寫入（編輯）許可
x	授與執行（搜尋）許可
-	未授與相對應許可

以 **-e** 旗標顯示的資訊和以 **-l** 旗標顯示的相同， 但新增的第 11 個字元解譯如下：

項目	說明
+	表示檔案具有延伸安全保護資訊。例如，檔案可能具有模式中延伸的 ACL、TCB，或 TP 屬性。
-	表示檔案沒有延伸的安全保護資訊。

在列出目錄中檔案大小時， **ls** 指令將顯示區塊的總數，包含間接 的區塊。

請參閱下列範例：

· 若要列出現行目錄下的所有檔案，請鍵入：

```
ls -a
```

此舉將列出所有檔案，包括

– 點 (.)
– 點點 (..)
– 其他以點 (.) 或不以點為檔名開頭的檔案

· 若要顯示詳細資訊，請鍵入：

```
ls -l chap1 .profile
```

如此會顯示一個長清單，內含有關 chap1 及 .profile 檔案的詳細資訊。

· 若要顯示目錄的相關詳細資訊，請鍵入：

```
ls -d -l . manual manual/chap1
```

這會針對目錄 . 及 manual，以及檔案 manual/chap1 顯示長清單。若沒有 **-d** 旗標，這會列出 . 及 manual 目錄中的檔案，而非關於目錄本身的詳細資訊。

請參閱 *Commands Reference, Volume 3* 中的 **ls** 指令，以取得完整語法。

刪除或移除目錄（rmdir 指令）

使用 **rmdir** 指令，可以自系統中移除由 *Directory* 參數所指定的目錄。

目錄必須空白（只能包含 **.** 及 **..**）才能移除，而且您必須對其母目錄具有寫入權。使用 **ls -a** *Directory* 指令，以檢查該目錄是否是空的。

以下是 **rmdir** 指令的用法範例：

· 若要清空並移除目錄，請鍵入：

```
rm mydir/* mydir/.*
rmdir mydir
```

此舉將移除 mydir 中的內容，然後移除空的目錄。**rm** 指令會顯示有關嘗試移除目錄點 (**.**) 及點點 (**..**) 的錯誤訊息，而 **rmdir** 指令會移除它們及目錄本身。

註：rm mydir/* mydir/.* 會先移除檔名不是以點 (.) 開頭的檔案，然後再移除那些檔名以點 (.) 開頭的檔案。**ls** 指令不會列出以點 (.) 開頭的檔名，除非您使用 **-a** 旗標。

· 若要移除 /tmp/jones/demo/mydir 目錄及其底下的所有目錄，請鍵入：

```
cd /tmp
rmdir -p jones/demo/mydir
```

此舉將自 /tmp 目錄中移除 jones/demo/mydir 目錄。如果在移除目錄時，目錄不是空的，或您沒有寫入許可，指令將終止，並顯示 適當的錯誤訊息。

請參閱 *Commands Reference, Volume 4* 中的 **rmdir** 指令，以取得完整語法。

比較目錄內容（dircmp 指令）

使用 **dircmp** 指令，可以比較由 *Directory1* 與 *Directory2* 參數所指定的兩個目錄，並將其內容的相關資訊寫入標準輸出。

首先 **dircmp** 指令將比較每一目錄中的檔名。如果兩個目錄中都包含相同的檔案名稱，**dircmp** 指令將比較這兩個檔案的內容。

在輸出中，**dircmp** 指令將列出每一目錄中的唯一檔案。然後將列出 在兩個目錄中名稱相同，但內容不同的檔案。如果沒有指定旗標，也將會列出 在兩目錄中具有相同內容以及相同名稱的檔案。

以下是 **dircmp** 指令的用法範例：

· 若要彙總 proj.ver1 和 proj.ver2 目錄中的檔案差異，請鍵入：

```
dircmp proj.ver1 proj.ver2
```

此舉將顯示 proj.ver1 與 proj.ver2 目錄之間的差異的摘要。該摘要將個別清單只在一個目錄中發現的檔案，與在這兩個目錄 中發現的檔案。如果檔案在這兩個目錄中發現，則 **dircmp** 指令將註明這 兩個檔案是否相同。

· 若要顯示 proj.ver1 和 proj.ver2 目錄中的檔案差異詳細資訊，請鍵入：

```
dircmp -d -s proj.ver1 proj.ver2
```

-s 旗標會抑制相同檔案的相關資訊。**-d** 旗標會顯示在這兩個目錄中發現的每一個不同檔案之 **diff** 清單。

請參閱 *Commands Reference, Volume 2* 中的 **dircmp** 指令，以取得完整語法。

檔案系統及目錄的指令摘要

下列是檔案系統與目錄的指令、目錄處理程序的指令，以及目錄縮寫的清單。

表 *5.* 檔案系統的指令摘要

項目	說明
df	報告檔案系統空間的相關資訊。

表 6. 目錄縮寫

項目	說明
.	現行工作目錄。
..	現行工作目錄之上的目錄（母目錄）。
~	您的起始目錄（對 Bourne shell 來說則不是如此。如需相關資訊，請參閱 Bourne Shell。）
$HOME	您的起始目錄（對所有的 shell 來說都是如此。）

表 7. 目錄處理程序的指令摘要

項目	說明
cd	變更現行目錄。
cp	複製檔案或目錄。
dircmp	比較兩個目錄與其共同檔案內容。
ls	顯示目錄內容。
mkdir	建立一或多個新的目錄。
mvdir	移動（更名）目錄。
pwd	顯示工作目錄的路徑名稱。
rmdir	移除目錄。

工作量管理程式

工作量管理程式 (WLM) 是設計用來為系統管理者提供對排程程式虛擬記憶體管理程式 (VMM)， 及磁碟 I/O 子系統如何將資源配置給處理程序之增加的控制。

您可使用 WLM 來防止不同的工作類別互相干擾，並根據不同使用者群組的基本要求來配置資源。

小心： 有效使用 WLM 必須對現有系統處理程序及效能有廣泛瞭解。如果系統管理者使用極限或不正確的值來配置 WLM，則會大大降低效能。

WLM 主要與大型系統搭配使用。大型系統常用於伺服器合併，其中許多不同伺服器系統（如印表機、資料庫、一般使用者及交易處理系統）的工作量會合併到單一大型系統，以減少系統維護的成本。這些工作量常常會互相干擾，且具有不同的目標及服務協議。

WLM 亦在具有完全不同系統行為的使用者群集之間提供隔離。這可防止其他行為（例如，批次或高記憶體使用率的工作）的工作量導致某些行為（例如，互動式或低 CPU 使用率工作）的工作量不被處理。

WLM 亦會嵌入統計作業子系統，這可讓使用者除了每一使用者或群組的標準統計作業之外，還會針對每一個 WLM 類別進行資源使用率統計作業。

工作量管理概念

使用 WLM，您可以建立工作的不同服務程式類別，並為那些類別指定屬性。

這些屬性可指定要配置給類別之 CPU、實體記憶體及磁碟 I/O 的最小及最大產能。然後，WLM 會使用系統管理者提供的類別指派規則，自動指派工作類別。這些指派規則基於處理程序的一組屬性值。系統管理者或特許使用者亦可手動指派工作類別，以置換自動指派的類別。

工作量管理的術語

下表會列出並說明與工作量管理相關的常用術語。

項目	說明
類別	類別是處理程序及其相關執行緒的集合。類別有單一一組資源限制值及目標持有率。類別 用於說明子類別及超類別。
超類別	超類別是一種有與其相關之子類別的類別。處理程序必須屬於某個子類別，才能屬於某個超類別。超類別有一組類別指派規則，可判斷將哪些處理程序指派給超類別。超類別亦有一組資源限制值及資源目標共用，可判斷屬於超類別的處理程序可使用的資源數量。這些資源是根據子類別的資源限制值及資源目標持有率，而在子類別中進行分割。
子類別	子類別是正好與一個超類別相關的一個類別。子類別中的每一個處理程序亦是其超類別的成員。子類別僅可存取超類別可用的資源。子類別有一組類別指派規則，可判斷哪些指派給超類別的處理程序屬於子類別。子類別亦有一組資源限制值及資源目標持有率，可判斷子類別中處理程序可使用的資源。 這些資源限制值及資源目標持有率指出，子類別中的處理程序可使用多少超類別可用的資源（超類別的目標）。 可使用 SMIT 或者 WLM 指令行介面來執行 WLM 管理。
分類機制	分類機制 是一組類別指派規則，可判斷要將哪些處理程序指派給哪些類別（超類別或超類別內的子類別）。
類別指派規則	類別指派規則指出，一組處理程序屬性中的哪些值會導致將處理程序指派給某個特定的類別（超類別或超類別中的子類別）。
處理程序屬性值	處理程序屬性值是處理程序具有之處理程序屬性的值。處理程序屬性可包括諸如使用者 ID、群組 ID 及應用程式路徑名稱的屬性。
資源限制值	資源限制值是 WLM 針對一組資源使用率值所維護的一組值。這些限制完全獨立於使用 **setrlimit** 子常式指定的資源限制之外。
資源目標持有率	資源目標持有率是類別（子類別或超類別）可用的資源持有率。這些持有率與其他類別持有率（子類別或超類別）在同一層次及層級上搭配使用，以判斷在該層次及層級類別之間想要的資源配送。
資源使用率值	資源使用率值是一個處理程序或一組處理程序目前在系統中使用的資源數量。它是一個處理程序還是一組處理程序是由處理程序資源收集的範圍來進行判斷。
資源收集範圍	資源收集範圍是收集資源使用率所在的層次，以及應用資源限制值所在的層次。這有可能在類別中每一處理程序的層次上、每一位使用者所擁有之類別中每一處理程序的總和層次，或者類別中每一處理程序的總和層次。目前支援的唯一範圍是後者。
處理程序類別特性	處理程序類別特性是根據已對某個處理程序指派的類別（子類別及超類別）而提供給該處理程序的一組特性。
類別授權	類別授權是一組規則，指出允許哪些使用者和群組執行某個類別上的作業，或某個類別中的處理程序及執行緒。這包括將處理程序手動指派給某個類別或者建立某個超類別之子類別授權。
類別層級	類別層級值是在全部類別的資源限制期望之階層中的類別位置。在將任何資源提供給較低層級類別之前，會先滿足層級中全部類別的資源限制（包括資源目標）。層級是在超類別及子類別層次上提供。會根據超類別的層級，將資源提供給超類別。在某個超類別內，會根據此超類別內子類別的層級值，將資源提供給子類別。因此，超類別層級是資源配送中主要的區分器；子類別層級提供了超類別內其他較小的區分器。

工作量管理的類別

WLM 可讓系統管理者定義類別，並為每一個類別定義一組屬性及資源限制。

會根據系統管理者提供的基準，將處理程序指派給類別。會在類別層次上強制執行資源權利及限制。這種定義服務程式類別及調節應用程式每一類別之資源使用率的方法，可防止有完全不同資源的應用程式使用型樣在共用單一伺服器時相互干擾。

WLM 支援兩個層次的類別階層：

- 根據每一個超類別的資源權利，將系統的資源在超類別中進行配送。系統管理者可定義資源權利。
- 而每一個超類別又可有子類別。會根據提供給每一個子類別的資源權利，將配置給超類別的資源在子類別中進行配送。
- 系統管理者可將每一個超類別之子類別的管理委託給超類別管理者或超類別管理者群組。
- WLM 最多支援 69 個超類別（64 個使用者定義的），及每個超類別支援 64 個子類別（61 個使用者定義的）。
- 根據組織的需要，系統管理者可決定是僅使用超類別還是使用超類別及子類別。

註： 在這部分的整個 WLM 討論中，術語類別同時適用於超類別及子類別。如果討論僅適用於特定的類別類型，則會明確提及該類型。

指派給類別進行工作量管理的處理程序
使用系統管理者提供的類別指派規則，將處理程序指派給類別。分類基準基於處理程序的一組屬性值，如使用者 ID、群組 ID、應用程式檔案名稱、處理程序類型及應用程式標記。

會使用定義的一組規則來判斷已指派處理程序的超類別。如果此超類別的子類別已定義，則此超類別會使用另一組規則來判斷要將哪個子類別指派給哪個處理程序。 此自動指派處理程序亦會同時考量超類別及子類別的繼承屬性。

自動類別分派是在處理程序呼叫 **exec** 子常式時執行的。類別指派是在處理程序使用可改變用於分類之處理程序屬性的子常式時重新評估的。範例為 **setuid**、**setgid**、**setpri** 及 **plock** 子常式。

除此自動類別指派之外，擁有適當授權的使用者可將處理程序或處理程序群組手動指派給特定的超類別或子類別。

相關概念
類別屬性
列出 WLM 類別的所有屬性。

資源控制
WLM 允許以兩種方式管理資源：以可用資源百分比的方式，或以總計資源使用情形的方式。

可根據百分比控制的資源包括下列內容：

- 類別中執行緒 SCHED_OTHER 類別的處理器用量。這是該類別中所有執行緒耗用之所有處理器循環之總和。固定優先順序執行緒是無法調整的。因此，無法予以變更，且這些執行緒可以超過處理器用量目標。
- 類別中處理程序的實體記憶體使用率。這是屬於類別中處理程序之所有記憶體頁的總和。
- 類別的磁碟 I/O 頻寬。這是在類別存取的每一個磁碟裝置上，由類別中執行緒啟動之所有 I/O 的頻寬（以每秒 512 位元組區塊為單位）。

可根據總計使用情形控制的資源屬於兩個種類之一：類別總計或處理程序總計。 類別總計種類包括：

類別中處理程序的數目
此為類別中同時處於作用中的處理程序數目。

類別中執行緒的數目
此為類別中同時處於作用中的執行緒數目。

類別中登入的數目
此為類別中同時處於作用中的登入階段作業數目。

處理程序總計種類包括：

總計 CPU 時間
此為單一處理程序的累計 CPU 時間總量。

總計磁碟 I/O
此為單一處理程序的累計磁碟 I/O 區塊總數。

總計連接時間

此為登入階段作業可處於作用中的時間總量。

資源授權

WLM 可讓系統管理者為每一種資源類型獨立地指定各類別的資源權利。

可藉由指出下列內容來指定這些權利：

- 不同類型資源的使用目標。此目標是以持有率指定的。會將持有率指定為不同類別之間的相對使用數量。例如，如果兩個類別分別有 CPU 的 1 及 3 的持有率，且它們此時是唯一的作用中類別，則 WLM 用於 CPU 調節的這兩個類別百分比目標將分別為 25% 及 75%。 每一層級中類別之目標百分比的計算是依據層級中作用中的持有率數目及層次可用的資源數量。
- 最小及最大限制。會將這些限制指定為可用的總資源百分比。WLM 支援兩種最大限制：
  - 軟性限制使用型樣值指出當存在資源競爭時可用資源的最大數量。如果沒有競爭（即沒有其他人需要資源），則可超出此最大值。
  - 強迫最大限制表示不論有無資源爭用情況，可用的最大資源量。不過，固定優先順序執行緒並不受限於這些規則，因此可超出限制。
- 總計限制。嚴格地強制執行總計限制。若處理程序超出其總計用量限制中的一個，則會終止該處理程序。若某類別處於其總計限制中的一個，則會導致建立該資源其他案例的任何作業都將失敗。

於大多數狀況下，軟性最大限制足以確保符合及強制執行資源權利。 使用硬性最大限制可能會導致未用的系統資源，因為即使沒有對資源的競爭，也會嚴格地強制執行這些限制。使用硬性最大限制時必須謹慎考量，因為若將這些限制設定得太低，就會大大影響系統或應用程式效能。 還應謹慎使用總計限制，因為這些限制可能導致處理程序終止或無法如預期那樣起作用。

WLM 在處於主動模式時，會嘗試使作用中類別接近其目標。因為對不同限制值很少有限制，所以對全部類別的任一限制總和可能遠超出 100%。於此狀況下，若所有類別皆處於作用中，則全部類別都達到該限制是不可能的。WLM 會根據系統中非固定優先順序執行緒所屬類別的執行情況，及其相對的限制及目標，調整這些執行緒的排程優先順序，藉此來調整處理器的用量。此方法可確保特定時間內保持處理器用量平均，而不是極短的間隔（如 10 毫秒）內有處理器用量。

例如，如果類別 A 是唯一作用中的類別，處理器最小值為 0% 而處理器目標是 60 持有率，那麼此類別可 100% 利用處理器。如果類別 B（處理器最小值為 0% 而處理器目標為 40 持有率）啟用時，類別 A 處理器利用率會逐漸降低至 60%，而類別 B 處理器利用率會從 0% 增至 40%。系統會在幾秒時間內分別保持 60% 及 40% 處理器利用率情況。

此範例假設類別之間無記憶體競爭。在一般工作條件下，處理器及記憶體所設定的限制是彼此相關的。例如，如果某類別記憶體用量最大限制，低於其工作設定，則該類別可能無法達到其目標，甚至達不到其最小處理器配置。

為協助修正給定一組應用程式的類別定義及類別限制，WLM 提供 **wlmstat** 報告工具，其可顯示每一類別目前正在使用的資源數量。亦提供系統監視一圖形顯示工具 **wlmmon**。

工作量管理程式虛擬記憶體限制

「工作量管理程式 (WLM)」虛擬記憶體限制提供管理者一種方法，讓他們對類別或處理程序提供虛擬記憶體限制，來預防系統由於分頁過多而降低效能或失敗。

當超出限制時，WLM 會執行下列其中一項來採取動作：

- 結束 WLM 類別下所有超出其限制的處理程序
- 只結束導致 WLM 類別用量超出其限制的處理程序
- 結束超出其處理程序限制的處理程序

您可以針對任何使用者定義的類別、使用者定義的超類別下的任何預設子類別，以及預設超類別指定虛擬記憶體限制。

為了統計作業，當判斷 WLM 總類別或處理程序用量時，WLM 只會考慮下列項目作為虛擬記憶體：

- 資料堆
- 載入器起始設定的資料、BSS、共用程式庫，以及私人載入的區段
- UBLOCK 及 mmap 區域

· 大型及固定的使用者空間頁面

管理者可以針對類別或針對類別中的每一個處理程序，指定 WLM 虛擬記憶體限制。當超出類別限制時，WLM 可以結束所有已指派給類別的處理程序，或只結束導致超出限制的處理程序，視 **vmenforce** 類別屬性分別設為 `class` 或 `proc` 而定。預設行為是只結束導致超出限制的處理程序。如果處理程序的虛擬記憶體用量大於限制，就會結束處理程序限制。

工作量管理程式的作業模式

可使用 WLM 來調節資源用量，以每一類別百分比、每一類別總計或每一處理程序總計表示。藉由以主動模式執行 WLM，可啟用對所有資源類型的調節。

您可以選擇啟動 WLM 的一種模式，將新的及現有處理程序分類並監視各種類別的資源使用情況，而不嘗試調節此使用情況。此模式稱為被動模式。

在新系統上配置 WLM 時，可使用被動模式來驗證分類及指派規則，以及在 WLM 不調整處理器及記憶體配置時，建立各種類別的資源利用基線。此應提供基準，供系統管理者判定如何應用資源持有率及資源限制（若必要），讓重要的應用程式優先，而限制較不重要的工作，以符合業務目標。

如果處理器時間是唯一您有興趣調整的資源，則您可以選擇，對於處理器以主動模式執行 WLM；所有其他資源則以被動模式執行。此模式稱為僅 *CPU* 模式。 若您要調節每一類別百分比，就無法為每一類別總計、每一處理程序總計或這兩者停用總計 資源類型與總計資源統計及調節。於所有模式下，您均可選擇停用資源組連結。

工作量管理程式的動態控制

當 WLM 處於作用中時，可以在任何時間修改現行配置的任何參數，包括類別屬性、其資源持有率及限制、指派規則，以及新增類別或刪除現有類別。

有數個方法可執行此動作，如：

· 修改目前作用中配置（由符號鏈結 `/etc/wlm/current` 指向的目錄）的內容檔案，以及藉由使用 **wlmcntrl** 指令重新整理 WLM，來使用新的參數。

· 使用另一組參數建立其他配置，並更新 WLM 以載入新配置的參數，從而使其成為現行配置。

· 修改目前作用中配置的部分參數，方法是使用 WLM 指令行介面（**mkclass**、**chclass** 及 **rmclass** 指令）。

· 修改應用程式目前作用中配置的部分參數，方法是使用 WLM API。

使用配置集，可在一天的指定時間自動切換到新配置。 配置集可讓管理者指定所要使用的一組配置，及每一配置將處於作用中的時間範圍。

監視工具

使用這些 WLM 指令，可以顯示 WLM 統計資料並監視 WLM 的作業。

· **wlmstat** 指令是文字導向的，並可將統計資料顯示為文字（由 WLM 管理的所有資源類型之每一類別的資源使用率百分比）。

· **wlmmon** 指令提供每一類別資源使用率及 WLM 調節情形的圖形式檢視畫面。

· **wlmperf** 指令是「效能工具箱」所提供之可選用的工具，它提供更多的功能（如長期記錄及重映）。

工作量管理程式中的舊版相容性

wlmstat 指令的預設輸出僅列出超類別，且與先前版本的那些超類別類似。

例如：

```
# wlmstat
                CLASS  CPU  MEM  DKIO
         Unclassified    0    0   0
            Unmanaged    0    0   0
              Default    0    0   0
               Shared    0    2   0
               System    2   12   0
               class1    0    0   0
               class2    0    0   0
#
```

如果部分超類別由 WLM 管理者來定義其子類別，則會列出這些子類別。例如：

```
# wlmstat
                CLASS  CPU  MEM  DKIO
        Unclassified    0    0   0
           Unmanaged    0    0   0
             Default    0    0   0
              Shared    0    2   0
              System    3   11   7
              class1   46    0   0
    class1.Default     28    0   0
     class1.Shared      0    0   0
        class1.sub1    18    0   0
              class2   48    0   0
#
```

該輸出與您使用 **ps** 指令時相同。對於無任何子類別之超類別中的處理程序，**ps** 指令會將超類別名稱列出為處理程序類別名稱。

每一類別統計作業

AIX 統計作業系統公用程式可讓您收集及報告使用者、群組或 WLM 類別對各種系統資源的使用。

當處理程序統計作業啟用時，作業系統會記錄當處理程序存在時，統計檔中處理程序資源使用情形的相關統計資料。此統計記錄會包括代表處理程序所屬 WLM 類別名稱的 64 位數字鍵值。

統計作業子系統使用 64 位元鍵值而不是完整 34 個字元的類別名稱，以便節省空間（否則，變更實際上將會是統計資料記錄大小的兩倍）。當執行統計指令來擷取各個處理程序的資料時，會使用上述常式將此鍵值轉換回為類別名稱。此轉換使用目前在 WLM 配置檔中的類別名稱。因此，如果在寫入統計記錄（當處理程序終止時）的時間與執行統計報告的時間當中已將某個類別刪除，則將找不到對應該鍵值的類別名稱，且類別會顯示為不明。

若要保留在統計期間刪除之類別資源使用情形的準確記錄，請執行下列其中一項：

- 不刪除類別，而是保留類別檔案中的類別名稱，並從規則檔案移除類別，使得無處理程序可指派給它。然後，您可在統計期間末，在統計報告已產生之後刪除該類別。
- 或者，從該類別所屬的配置將其刪除，並將類別檔案中的類別名稱保存在「虛擬」配置（永不啟動的配置）中，直到該期間的統計記錄已產生之後為止。

管理工作量管理程式

工作量管理程式 (WLM) 可讓系統管理者更能控制排程程式與虛擬記憶體管理程式 (VMM) 配置處理程序的資源之方式。您可使用 WLM 來防止不同的工作類別互相干擾，並根據不同使用者群組的基本需求來配置資源。

使用 WLM，您可以建立工作的不同服務程式類別，並為那些類別指定屬性。這些屬性可指定要配置給類別之 CPU、實體記憶體及磁碟 I/O 的最小及最大產能。然後，WLM 會使用系統管理者提供的類別指派規則，自動指派工作類別。這些指派規則基於處理程序的一組屬性值。系統管理者或特許使用者亦可手動指派工作類別，以置換自動指派的類別。

WLM 是基本作業系統的一部分，它會隨基本作業系統一同安裝，是一種可選用的服務。必須對它進行配置使其符合您的系統環境，並且必須在您想要使用時啟動它，而在您想要暫停或結束 WLM 服務時停止它。

本章節包含使用適合於您站台的類別及規則來配置 WLM 的程序，以及為非預期資源用量行為疑難排解的建議。

小心： 本節中的作業假設您熟悉 WLM 概念。有效使用 WLM 必須對現有系統處理程序及效能有廣泛瞭解。如果系統管理者使用極限或不正確的值來配置 WLM，則會大大降低效能。

啟動及關閉工作量管理程式

WLM 是一種必須啟動及停止的選用服務。

建議您使用其中一種系統管理介面 (SMIT) 來啟動或停止 WLM。

- 若要使用 SMIT 啟動或停止 WLM，請使用 `smit wlmmanage` 捷徑。

這兩種選項的主要差異在於永久性。您可以在 SMIT 中使用三種方式來啟動或停止 WLM：

現行階段作業
如果您使用此選項來要求停止 WLM，WLM 將僅在此階段作業停止，而會在下一次重新開機時重新啟動。如果您使用此選項來要求啟動，WLM 將僅在此階段作業啟動，而不會在下一次重新開機時重新啟動。

下一次重新開機
如果您使用此選項來要求停止 WLM，WLM 將僅在此階段作業保持執行中，而將不會在下一次重新開機時重新啟動。如果您使用此選項來要求啟動，WLM 在此階段作業會無法使用，而將在下一次重新開機時啟動。

兩者
如果您使用此選項來要求停止 WLM，WLM 將僅在此階段作業停止，而將不會在下一次重新開機時重新啟動。如果您使用此選項來要求啟動，WLM 將僅在此階段作業啟動且會在下一次重新開機時重新啟動。

您亦可使用 **wlmcntrl** 指令，但 **wlmcntrl** 指令僅可讓您在目前階段作業啟動或停止 WLM。如果您想要使用指令行介面，並且想要在機器重新啟動時讓變更維持有效，則必須編輯 /etc/inittab 檔案。

可使用 WLM 來調節資源用量，以每一類別百分比、每一類別總計或每一處理程序總計表示。藉由以主動模式執行 WLM，可啟用對所有資源類型的調節。您可以選擇啟動 WLM 的一種模式，將新的及現有處理程序分類並監視各種類別的資源使用情況，而不嘗試調節此使用情況。此模式稱為被動模式。如果 CPU 時間是您唯一有興趣進行調節的資源，則可以為 CPU 選擇以主動模式執行 WLM，為所有其他資源選擇以被動模式執行。此模式稱為僅 *CPU* 模式。

在 WLM 啟動前即已存在於系統中的所有處理程序，均會依據新載入的指派規則加以分類，並受 WLM 監視。

工作量管理程式內容

您可以使用 SMIT、WLM 指令行介面，或建立純 ASCII 檔，來指定 WLM 配置的內容。SMIT 介面會使用 WLM 指令來記錄相同純 ASCII 檔（稱為內容檔）中的資訊。

一組 WLM 特性檔可定義 WLM 配置。您可以建立多組特性檔，以定義不同的工作量管理配置。這些配置位於 **/etc/wlm** 的子目錄中。說明 *Config* 配置超類別的 WLM 內容檔是 /etc/wlm/Config 中檔案的 *classes*、*description*、*limits*、*shares* 及 *rules*。然後，說明此配置的超類別 *Super* 之子類別的特性檔是目錄 /etc/wlm/Config/Super 中檔案的 *classes*、*limits*、*shares* 及 *rules*。僅 root 使用者可以啟動或停止 WLM，或從一個配置切換至另一個配置。

這些特性檔命名如下：

項目	說明
classes	類別定義
description	配置說明文字
groupings	屬性值分組
limits	類別限制
shares	類別目標共用
rules	類別指派規則

提交 WLM 內容檔的指令 **wlmcntrl** 及其他 WLM 指令，可讓使用者指定 WLM 內容檔的替代目錄名稱。這可讓您變更 WLM 特性，而不用變更預設的 WLM 特性檔。

符號鏈結 /etc/wlm/current 指向含有現行配置檔的目錄。當您使用指定的配置或配置集啟動 WLM 時，請使用 **wlmcntrl** 指令更新此鏈結。作業系統隨附的範例配置檔位於 /etc/wlm/standard 中。

建立屬性值分組

您可以將屬性值分組，並在 rules 檔案中以單一值代表它們。這些屬性值分組定義在 WLM 配置目錄內的 groupings 檔案中。

根據預設值，配置沒有 groupings 檔案。沒有可用來建立這種檔案的指令或管理介面。若要建立及使用屬性值分組，請使用下列程序：

1. 以 root 權限變更到適當的配置目錄，如以下範例所示：

```
cd /etc/wlm/MyConfig
```

2. 使用您喜好的編輯器來建立及編輯名為 groupings 的檔案。
例如：

```
vi groupings
```

3. 使用下列格式定義屬性及其相關值：

attribute =
value, *value*, ...

所有值均必須以逗點區隔， 空格無意義。允許使用範圍及萬用字元。例如：

```
trusted = user[0-9][0-9], admin*
nottrusted = user23, user45
shell = /bin/?sh, \
                /bin/sh, \
                /bin/tcsh
rootgroup=system,bin,sys,security,cron,audit
```

4. 儲存檔案。
5. 若要在某類別的選取準則內使用屬性分組，請編輯 rules 檔案。

屬性分組名稱之前必須有貨幣符號 ($) 以包含對應值，或者必須有驚嘆號 (!) 以排除對應值。不能在群組成員中使用驚嘆號（步驟 第 109 頁的『3』），且它是唯一可用於此規則檔案中分組之前的修飾元。在以下範例中，星號 (*) 表示註解行：

```
*class  resvd  user                  group       application       type  tag
classA  -      $trusted,!$nottrusted  -           -                 -     -
classB  -      -                      -           $shell,!/bin/zsh  -     -
classC  -      -                      $rootgroup  -                 -     -
```

6. 儲存檔案。

此時，您的分類規則包括了屬性值分組。解析規則時，如果某元素以 $ 開頭，則系統會在 groupings 檔案內尋找該元素。如果某元素在語法上無效，或者 groupings 檔案不存在，則系統會顯示警告訊息並繼續處理其他規則。

建立時間型配置集

您可以建立一組專門的配置，並將該組配置中的每個配置指派到您想要特定配置生效的日期及時間。

這些配置組合（稱為時間型配置集）完全獨立於一般配置之外，但卻相容。您可以使用 **wlmcntrl -u** 指令，視需要在配置集與一般配置之間切換。

使用配置集時，您通常將現有的已命名配置與特定時間範圍產生關聯。因為在任一給定時間只能使用一個配置，所以每個指定的時間範圍必須是唯一的；時間範圍不能重疊或重複。

當指定的配置超出時間範圍，且需要使用另一個配置時，**wlmd** 常駐程式會對 WLM 提出警示。僅 root 使用者可以管理這些時間範圍，時間範圍在配置集目錄內稱為 .times 的 ASCII 檔中加以指定。

使用下列程序建立時間型配置集：

1. 以 root 權限建立配置集目錄，然後變更到該目錄。
例如：

```
mkdir /etc/wlm/MyConfigSet
cd /etc/wlm/MyConfigSet
```

2. 使用您喜好的編輯器建立配置集的 .times 檔案，並以下列格式指定配置及時間範圍：

```
ConfigurationName:
            time = "N-N,HH:MM-HH:MM"
```

或

```
ConfigurationName:
            time = -
```

（未指定時間值）

其中，*N* 是代表星期幾的數字，範圍是從 0（星期日）到 6（星期六），*HH* 代表小時，範圍是從 00（午夜）到 23（晚上 11 點），而 *MM* 代表分鐘，範圍是從 00 到 59。您可以僅指定星期幾，或根本不指定。只要分鐘值為 00，則對於一天中的最後一個小時而言，24 是有效的小時值。如果為特定配置鍵入一個連字號（-) 而不是時間範圍，則將在其他配置的時間範圍未生效時使用該配置。僅可以指定一個沒有時間範圍的配置。

例如：

```
conf1:
        time =
conf2:
        time = "1-5,8:00-17:00"
conf2
        time = "6-0,14:00-17:00"
conf3
        time = "22:00-6:00"
```

3. 使用 **wlmcntrl -u** 指令，可以新的配置集更新 WLM。
   例如：

```
wlmcntrl -u /etc/wlm/MyConfigSet
```

此時，WLM 的現行配置是新的時間型配置集。

您亦可使用 **confsetcntrl** 及 **lswlmconf** 指令來建立及操控配置集。例如：

若要以預設配置 conf1 建立 confset1 配置集，請使用下列指令：

```
confsetcntrl -C confset1 conf1
```

若要將 conf2 新增至 confset1，並使它在每天的上午 8 點到下午 5 點為作用中配置，請使用下列指令：

```
confsetcntrl -d confset1 -a conf2 "0-6,08:00-17:00"
```

若要讓此配置集成為作用中配置，請使用下列指令：

```
wlmcntrl -d confset1
```

建立資源集

就 CPU 而論，使用資源集 (rset) 是隔離工作量的有效方法。將兩個不同的工作量分隔成兩個類別，並提供每一個類別不同的 CPU 子集，則可以確定即使這兩個工作量仍競爭實體記憶體及 I/O 頻寬，它們也絕不會彼此競爭 CPU 資源。

建立資源集最簡單的方法是使用 SMIT 介面（**smit addrsetcntl** 捷徑）或 **mkrset** 指令。

為提供指示，以下範例說明了在 4 向系統上建立及命名資源集的每個步驟。其目標是建立含有處理器 0 到 2 的資源集，並在 WLM 配置中使用它，將超類別的所有處理程序限制在這 3 個處理器中。

1. 利用 root 權限，使用下列指令檢視可用的構成要素（用來建立資源集）：

```
lsrset -av
```

此範例的輸出如下：

```
T  Name              Owner   Group   Mode    CPU  Memory  Resources
r  sys/sys0          root    system  r-----    4   98298  sys/sys0
r  sys/node.00000    root    system  r-----    4   98298  sys/sys0
r  sys/mem.00000     root    system  r-----    0   98298  sys/mem.00000
r  sys/cpu.00003     root    system  r-----    1       0  sys/cpu.00003
r  sys/cpu.00002     root    system  r-----    1       0  sys/cpu.00002
r  sys/cpu.00001     root    system  r-----    1       0  sys/cpu.00001
r  sys/cpu.00000     root    system  r-----    1       0  sys/cpu.00000
```

在該輸出中，**sys/sys0** 代表整個系統（在此情況下，是 4 向 SMP）。當 WLM 類別未指定 `rset` 屬性時，這就是其處理程序可能存取的預設集。

2. 使用下列 SMIT 捷徑建立及命名資源集：

```
smit addrsetcntl
```

在此範例中，按如下方式填寫欄位：

名稱空間
admin

資源集名稱
proc0_2

資源
從清單中選取對應於記憶體及 CPU 0 到 2（sys/cpu.00000 到 sys.cpu.00002）的那些行。

所有其他欄位
從清單中選取。

完成輸入欄位並結束 SMIT 後，會在 `/etc/rsets` 中建立 `admin/proc0_2` rset。

3. 若要使用新 rset，請使用下列 SMIT 捷徑將它新增至核心資料結構：

```
smit reloadrsetcntl
```

此功能表可讓您選擇要以立即、下一次開機時或兩者的方式重新載入資料庫。因為這是您第一次使用新資源集，請選取兩者，以便立即及在每次重新開機後載入此 rset。（如果您已變更現有的 rset，則可能已選取立即。）

4. 使用下列 SMIT 捷徑將新的 rset 新增至 WLM 類別：

```
smit wlmclass_gal
```

選取類別（在此範例中，為 **super1**），然後從**資源集**欄位的可用清單中， 選取 **admin/proc0_2**。作出選擇並結束 SMIT 後，磁碟上的 `classes` 檔案會變更。

5. 請執行下列其中一項：

· 如果 WLM 在執行中，請使用下列 SMIT 捷徑來更新配置：

```
smit wlmupdate
```

· 如果 WLM 不在執行中，請使用下列 SMIT 捷徑啟動它：

```
smit wlmstart
```

6. 監視該類別上新資源集的效果。例如：

a) 在類別 **super1** 中啟動 90 個 CPU 迴圈（執行無限迴圈的程式）。

b) 在指令行上鍵入 `wlmstat`。此範例的輸出如下：

```
                CLASS CPU MEM BIO
        Unclassified   0   0   0
           Unmanaged   0   0   0
             Default   8   0   0
              Shared   0   0   0
              System   0   0   0
              super1  75   0   0
              super2   0   0   0
    super2.Default     0   0   0
     super2.Shared     0   0   0
       super2.sub1     0   0   0
       super2.sub2     0   0   0
```

此輸出顯示 90 個 CPU 限制處理程序， 由於資源集限制它們在 CPU 0 到 2 上執行，所以目前僅使用 75% 的 CPU（如果沒有限制，它們會佔用 100% 的 CPU）。

c) 使用下列 SMIT 捷徑來驗證處理程序（按 PID 識別）有權存取的資源集：

```
smit lsrsetproc
```

輸入您感興趣的處理程序 PID，或從清單中選取。以下是一個迴圈處理程序的輸出：

```
    CPU  Memory  Resources
           3   98298  sys/mem.00000 sys/cpu.00002 sys/cpu.00001 sys/cpu.00000
```

不過，類別中沒有指定 `rset` 屬性的處理程序會使用 `Default` 資源集，其中包括除了屬於專用資源集的那些處理器之外的所有處理器。 不屬於任何類別的處理程序會使用 `System` 類別（如果它是 root 處理程序）或 `Default` 類別（如果它是非 root 處理程序）。 任何這些類別都可能有已為它們定義的資源集。

以下是位於未指定資源集之類別中 **init** 處理程序的輸出：

```
    CPU  Memory  Resources
           4   98298  sys/sys0
```

此時，您的資源集已存在，且至少正由 WLM 中的一個類別使用。

註： WLM 不會為目前具有 **bindprocessor** 子常式連結或另一個 rset 連接的處理程序設定其 rset 連接。當其他連接不復存在時，WLM 將自動指派其 rset。

註： 可為任何 WLM 類別建立資源集，這些類別包括 `Default` 和 `System` 類別。

配置工作量管理程式以合併工作量

工作量管理程式 (WLM) 可讓您控制系統上的工作使用的資源。

每個安裝的 AIX 作業系統上均有預設的 WLM 配置範本。下面的程序會更新此 WLM 配置範本，以便在共用伺服器上施行資源管理原則。所產生的配置可作為測試的開始。如何正確配置 WLM 取決於您對環境的工作量及原則基本需求。

註：

1. 有效使用 WLM 必須對現有系統處理程序及效能有廣泛瞭解。您可能需要進行重複測試及調整，才能開發出適合您工作量的配置。如果您使用極值或不正確的值來配置 WLM，可能會大大降低系統效能。
2. 如果您已經知道處理程序的一個以上分類屬性（如使用者、群組或應用程式名稱），則配置 WLM 的處理程序會更為簡單。如果您不熟悉資源的現行使用狀況，請使用工具（如 **topas**），來識別主要資源使用者的處理程序，並利用所產生的資訊開始定義類別及規則。
3. 下面的範例情節假設您熟悉第 102 頁的『工作量管理概念』中說明的基本工作量管理程式概念。

WLM 配置檔存在於 `/etc/wlm/`*ConfigurationName* 目錄中。每個超類別的所有子類別均定義在名為 `/etc/wlm/`*ConfigurationName/SuperClassName* 的配置檔中。如需這些檔案的詳細資訊，請參閱 *Files Reference*。

在下面的程序中，您可將兩個不同的部門伺服器工作量合併到一台更大的伺服器上。此範例會編輯配置檔，但是您還可以使用 SMIT（使用 **smit wlmconfig_create** 捷徑）來建立配置。簡而言之，在此程序中，您將執行下列動作：

1. 指定您要合併之應用程式的資源基本需求。這會協助您判斷可以將多少個應用程式移至更大的伺服器上。
2. 定義層級以及資源持有率和限制，以便開始測試合併後的工作量。
3. 微調配置，直到達到期望結果為止。

此作法範例情節中的資訊已使用特定版本的 AIX 進行測試。依據您的 AIX 版本及層次，所得到的結果可能大不相同。

步驟 1. 識別應用程式需求

在此範例情節中，工作量是您可能在資料庫伺服器上看到的典型工作量。假設工作屬於下列一般種類：

接聽程式
它們大部分時間處於休眠狀態，但會定期甦醒以回應要求的處理程序。雖然這些處理程序不會耗用很多資源，但回應時間可能極為重要。

工作者
它們是代表某個要求（無論是本端或遠端的要求）來執行工作的處理程序。這些處理程序可能會佔用大量 CPU 時間及記憶體。

報告程式
它們是執行自動化作業的處理程序。它們可能需要佔用大量 CPU 時間或記憶體，但可以忍受較慢的回應時間。

監視程式
它們是通常定期執行來驗證系統或應用程式狀態的處理程序。這些處理程序可能會使用大量資源，但使用時間很短。

指令
它們是系統使用者可能隨時執行的指令或其他應用程式。它們的資源基本需求並無法預測。

除了此項工作，已排程的工作可能屬於下列其中一個種類：

SysTools
它們是執行自動化作業的處理程序。這些工作對系統作業不是很重要，但需要在特定時間限制內定期執行。

SysBatch
它們是很少執行、對系統作業不很重要、且無需及時完成的處理程序。

建立配置首先要定義類別及規則。在下列步驟中，您可使用上列的一般工作種類來定義類別。請使用下列程序：

1. 使用下列指令，在 /etc/wlm 目錄中建立稱為 MyConfig 的新配置：

   ```
   mkdir /etc/wlm/MyConfig
   ```

2. 使用下列指令，將範本檔案複製到 /etc/wlm/MyConfig 目錄中：

   ```
   cp -pr /etc/wlm/template/* /etc/wlm/MyConfig
   ```

3. 若要建立超類別，請使用您偏好使用的編輯器來修改 /etc/wlm/MyConfig/classes 檔案，以加入下列內容：

   ```
   System:
   Default:
   DeptA:
   DeptB:
   SysTools:
   SysBatch:
   ```

   一開始時，您需為每個部門定義一個超類別（因為兩個部門將共用伺服器）。SysTool 及 SysBatch 超類別將處理上述一般種類中概述的已排程工作。System 及 Default 超類別一律都已定義。

4. 在 MyConfig 目錄內，為每一個 DeptA 及 DeptB 超類別建立目錄。（建立配置時，您必須為擁有子類別的每個超類別建立一個目錄。）在接下來的步驟中，您必須為每個部門的超類別定義五個子類別（每個工作種類都需定義一個子類別）。

5. 若要為工作的每個一般種類建立子類別，請編輯 /etc/wlm/MyConfig/DeptA/classes 及 /etc/wlm/MyConfig/DeptB/classes 檔案，需包含下列內容：

   ```
   Listen:
   Work:
   Monitor:
   ```

```
Report:

Command:
```

註：對每個超類別而言，classes 檔案的內容可能會不同。

指定完類別後，在接下來的步驟中，您需建立分類規則，以在超類別及子類別層次中分類處理程序。為求簡化，假設所有應用程式都是從已知位置執行，來自某一部門的所有處理程序都在 deptA UNIX 群組下執行，而來自其他部門的所有處理程序是在 deptB UNIX 群組下執行。

6. 若要建立超類別指派規則，請修改 /etc/wlm/MyConfig/rules 檔案來包含下列內容：

```
DeptA - - deptA - -
DeptB - - deptB - -
SysTools - root,bin - /usr/sbin/tools/* -
SysBatch - root,bin - /usr/sbin/batch/* -
System - root - - -
Default - - - - -
```

註：如果相同應用程式的多個案例可能在執行，且所有分類屬性（標記除外）均相同，請使用 **wlmassign** 指令或 **wlm_set_tag** 子常式，藉由將它們指派給不同的類別來區分它們。

7. 若要建立多個特定的子類別規則，請建立含有下列內容的 /etc/wlm/MyConfig/DeptA/rules 及 /etc/wlm/MyConfig/DeptB/rules 檔案：

```
Listen - - - /opt/myapp/bin/listen* -
Work - - - /opt/myapp/bin/work* -
Monitor - - - /opt/bin/myapp/bin/monitor -
Report - - - /opt/bin/myapp/report* -
Command - - - /opt/commands/* -
```

8. 若要判斷每個類別的資源使用行為，請使用下列指令以被動模式啟動 WLM：

```
wlmcntrl -p -d MyConfig
```

在以被動模式啟動 WLM 後，您可以先個別執行每個應用程式，以更細微的觀點來瞭解其資源基本需求。然後，您可以同時執行所有應用程式，以便更適當地判斷所有類別之間的互動。

識別應用程式資源基本需求的一個替代方法是，先在您從中合併應用程式的個別伺服器上以被動模式執行 WLM。此方法的缺點是您必須在較大的系統上重建配置，且所需的資源百分比在較大的系統上可能會不同。

步驟 2. 定義層級、持有率及限制

WLM 配置被用來施行資源管理原則。以被動模式執行 WLM 所提供的資訊可協助您判斷資源管理原則對於給定的工作量是否合理。您現在可以根據資源管理原則來定義層級、持有率及限制，以調節工作量。

對於此範例情節，假設您有下列基本需求：

- System 類別必須擁有最高優先順序，且必須保證它總是能夠存取某個百分比的系統資源。
- SysTools 類別必須總是能夠存取特定百分比的資源，但存取的資源量不至於嚴重影響在 DeptA 及 DeptB 中執行的應用程式。
- SysBatch 類別不能干擾系統上的任何其他工作。
- DeptA 將得到 60% 的可用資源（即提供給具有持有率之類別所使用的資源），而 DeptB 將得到 40%。在 DeptA 及 DeptB 中：
  - Listen 類別中的處理程序必須回應具有低潛伏期的要求，但不可以耗用太多資源。
  - 必須允許 Work 類別耗用大部分資源。而 Monitor 及 Command 類別也須耗用部分資源，但必須少於 Work 類別。
  - Report 類別不能干擾任何其他工作。

在下面的程序中，您可定義層級、持有率及限制：

1. 若要建立超類別層級，請使用您喜好的編輯器修改 /etc/wlm/MyConfig/classes 檔案，在其中包含下列內容：

```
System:

Default:

DeptA:
                localshm = yes
                adminuser = adminA
                authuser = adminA
                inheritance = yes

DeptB:
                localshm = yes
                adminuser = adminB
                authuser = adminB
                inheritance = yes

SysTools:
                localshm = yes

SysBatch:
                tier = 1
                localshm = yes
```

   將 SysBatch 超類別放置在層級 1 中，因為此類別包含優先順序非常低的工作，且您不想讓這些工作干擾系統上的其他工作。（如果未指定層級，類別會預設為層級 0。）對每個部門超類別的管理是由 adminuser 及 authuser 屬性來定義。已為 DeptA 及 DeptB 啟用了 inheritance 屬性。所有在具有繼承的類別中啟動的新處理程序，都將繼續被分類於該類別中。

2. 若要為工作的每個群組建立子類別層級，請修改 /etc/wlm/MyConfig/DeptA/classes 及 /etc/wlm/MyConfig/DeptB/classes 檔案，以包含下列內容：

```
Listen:

Work:

Monitor:

Report:
                tier = 1
Command:
```

3. 若要為超類別指派起始持有率，請編輯 /etc/wlm/MyConfig/shares 檔案，以包含下列內容：

```
DeptA:
                CPU = 3
                memory = 3

DeptB:
                CPU = 2
                memory = 2
```

   因為您已指派 CPU 總計 5 個持有率，所以 DeptA 處理程式將可存取總計 CPU 資源 5 個持有率中的 3 個持有率（或 60%），而 DeptB 處理程序將可存取 5 個中的 2 個持有率（或 40%）。因為您未給 SysTools、System 及 Default 類別指派持有率，所以它們的用量目標與作用中持有率數量互為獨立，因而具有高於 DeptA 及 DeptB 的資源存取優先順序（直到達到它們的限制為止）。您未給 SysBatch 類別指派任何持有率，因為它是層級 1 中唯一的超類別，因此任何持有率分派都是不恰當的。SysBatch 類別中的工作只能耗用層級 0 中所有類別未使用的資源。

4. 若要為子類別指派起始持有率，請編輯 /etc/wlm/MyConfig/DeptA/shares 及 /etc/wlm/MyConfig/DeptB/shares 檔案，以包含下列內容：

```
Work:
                CPU = 5
                memory = 5

Monitor:
                CPU = 4
                memory = 1
```

```
Command:
                CPU = 1
                memory = 1
```

因為您未給 Listen 類別指派持有率，所以當它需要資源時，將擁有對資源的最高優先順序存取權（在超類別中）。您為 Work 類別指派最大數目的持有率，因為它的資源基本需求最大。因此，您根據 Monitor 及 Command 類別的觀察行為及相對重要性，為它們指派持有率。您未給 Report 類別指派持有率，因為它是層級 1 中唯一的子類別，因此任何持有率指派都是不恰當的。Report 類別中的工作僅耗用層級 0 中子類別未使用的資源。

在此範例接下來的步驟中，您將指派限制給無指派持有率的類別。（您還可以為具有持有率的類別指派限制。如需相關資訊，請參閱使用 WLM 管理資源）。

5. 若要為超類別指派限制，請編輯 /etc/wlm/MyConfig/limits 檔案，以包含下列內容：

```
Default:
                CPU = 0%-10%;100%
                memory = 0%-10%;100%

SysTools:
                CPU = 0%-10%;100%
                memory = 0%-5%;100%

System:
                CPU = 5%-50%;100%
                memory = 5%-50%;100%
```

您為 System、SysTools 及 Default 類別指派軟性最大限制，以防止它們對系統上的其他工作有重大干擾。您為 System 類別指派 CPU 及記憶體的最小限制，因為此類別包含系統作業所需的基本處理程序，且它必須能夠使用保證數量的資源。

6. 若要為子類別指派限制，請編輯 /etc/wlm/MyConfig/DeptA/limits 及 /etc/wlm/MyConfig/DeptB/limits 檔案，以包含下列內容：

```
Listen:
                CPU = 10%-30%;100%
                memory = 10%-20%;100%

Monitor:
                CPU = 0%-30%;100%
                memory = 0%-30%;100%
```

註：每個子類別檔案的限制可能會不同。

您為 Listen 及 Monitor 類別指派軟性最大限制，以防止它們對相同超類別中的其他子類別有重大干擾。特別是，您不想要系統繼續接受 Work 類別中工作的要求（如果 Work 類別無法存取資源來處理它們）。您也為 Listen 類別指派最小限制以確保有快速的回應時間。記憶體的最小限制會確保此類別使用的分頁不會被分頁置換所挪用，從而加快執行時期。CPU 的最小限制則確保在可執行這些處理程序時，它們會擁有對 CPU 資源的最高優先順序存取權（在超類別中）。

步驟 3. 細部調整工作量管理程式配置

1. 使用 **wlmstat** 指令監視系統，並驗證由 WLM 進行的調節符合您的目標，以及部分應用程式未被過度地剝奪資源（而其他應用程式取得多於它們應得的資源）。如果是這樣的狀況，則調整持有率並重新整理 WLM。
2. 您監視並調整持有率、限制及層級號碼時，決定您是否要針對部分或所有超類別委託子類別的管理。然後管理者可監視並設定子類別持有率、限制及層級號碼。

每一超類別的管理者可針對每一超類別的子類別重複此處理程序。唯一的差異是 WLM 無法以被動模式僅在子類別層次執行。必須使用主動模式的 WLM 來進行子類別配置及調整。不影響超類別中使用者及應用程式的一個方法是，啟動層級號碼，並以子類別的預設值（'-'（連字號）代表持有率，0% 代表最小值，100% 代表軟性及強迫最大值）啟動子類別的持有率及限制。使用這些設定，WLM 將不會調節子類別之間的資源配置。

如需相關資訊

· 工作量管理程式。

· 工作量管理。

· 效能管理中的工作量管理診斷。

· *Files Reference* 中的 **classes**、**limits**、**rules** 及 **shares** 檔案的說明。

· **topas**、**wlmassign**、**wlmcheck**、**wlmcntrl** 及 **wlmstat**。

· WLM 子常式說明，尤其是 **wlm_set_tag**。

類別

工作量管理程式可藉由定義服務類別，及將資源配置到所有這些類別，來協助您控制系統資源的配置。

每個類別皆具有一組屬性，可判斷其資源權利是什麼，及其他行為。系統上的每個處理程序皆被分類到一個服務程式類別，因而受制於該類別之資源權利及行為的實施。可使用手動指派或自動根據使用者定義的分類規則，將處理程序指派給類別。

WLM 支援兩種層次的類別：超類別及子類別。根據可用的系統資源將資源授權提供給超類別，而將相對於子類別的相關聯超類別授權的資源授權提供給子類別。 您可選擇性地定義子類別，以允許對超類別中處理程序進行更精密地控制。也可以藉由指定超類別的 admin 使用者或 admin 群組，來委託定義子類別的責任。

對於超類別及子類別層次，可以使用 SMIT 或指令行介面，來定義類別、資源共用和限制，以及規則。應用程式可使用 WLM API。 配置定義保留於稱為 WLM 特性檔案的一組文字檔中。

類別名稱最長可為 16 個字元，僅可包含大寫與小寫字母、數字及底線 (_)。 對於給定的 WLM 配置，每個超類別名稱都必須唯一。每個子類別名稱在該超類別內必須是唯一的，但它可以符合其他超類別中的子類別名稱。若要獨一無二地識別每個子類別，完整的子類別名稱由超類別名稱及子類別名稱所組成，以點區隔；例如：*Super.Sub*。

超類別

系統管理者最多可定義 64 個超類別。

此外，還會自動建立下列五個超類別：

***Default* 超類別**
是預設超類別，且一定會定義。所有未被自動指派給特定超類別的非 root 處理程序會被指派給 Default 超類別。藉由提供特定的指派規則，亦可將其他處理程序指派給 *Default* 超類別。

***System* 超類別**
如果所有特許 (root) 處理程序未由規則指派給特定的類別，則會將那些處理程序指派給該超類別。 此超類別亦會收集屬於核心記憶體區段及核心處理程序的記憶體頁。藉由為此超類別提供特定的指派規則，亦可將其他處理程序指派給 System 超類別。此超類別的預設記憶體最小限制為 1%。

***Shared* 超類別**
接收多個超類別中處理程序所共用的記憶體頁。這包括共用記憶體區域中的頁，以及多個超類別（或不同超類別的子類別中）中處理程序使用之檔案中的頁。由完全屬於單一超類別（或同一超類別的子類別）之多個處理程序所使用的共用記憶體及檔案與該超類別有關聯。僅當源自不同超類別的處理程序存取共用記憶體區域或檔案時，才會將頁置於 Shared 超類別中。此超類別僅可應用實體記憶體持有率及限制。它不可有其他資源類型的持有率或限制、子類別或指定的指派規則。由同一超類別之不同子類別中的處理程序共用的記憶體區段，是要被分類到 *Shared* 子類別，還是要保留在其原始子類別中，是取決於原始子類別的 **localshm** 屬性值。

***Unclassified* 超類別**
是針對未分類處理程序的記憶體配置。在啟動 WLM 時存在的處理程序是根據正在載入之 WLM 配置的指派規則進行分類的。在此起始分類期間，與每一個處理程序連接的所有記憶體頁會被「記入」處理程序所屬的超類別（當不共用或由同一超類別中的處理程序所共用時），或被「記入」*Shared* 超類別（當由不同超類別中的處理程序所共用時）。

不過，在進行此分類時，有數頁無法直接連繫到任何處理程序（並因此無法連繫到任何類別），且此記憶體會被記入 *Unclassified* 超類別。此記憶體的大部分會隨著時間的過去而被正確地重新分類（當它是由處理程序存取，或者在啟動 WLM 之後被釋放並重新配置到處理程序時）。*Unclassified* 超類別中沒有處理程序。此超類別可應用實體記憶體持有率及限制。它不可有其他資源類型的持有率或限制、子類別或指定的指派規則。

***Unmanaged* 超類別**

一律定義名為 *Unmanaged* 的特定超類別。不會對此類別指派處理程序。此類別用於累計不受 WLM 管理之系統中所有已固定頁的記憶體用量。不會在任何類別中累計 `waitproc` 處理程序的 CPU 使用率，以防止系統出現 100% 使用率。這個超類別無法具有任何其他資源類型的持有率或限制、子類別或指定的指派規則。

子類別

系統管理者或超類別管理者最多可定義 61 個子類別。

此外，兩個特殊子類別 *Default* 及 *Shared* 都一定會定義。

***Default* 子類別**

是預設子類別，且一定會定義。未被自動指派給超類別之特定子類別的所有處理程序會被指派給 *Default* 子類別。您亦可藉由提供特定的指派規則，將其他處理程序指派給 *Default* 子類別。

***Shared* 子類別**

接收超類別之多個子類別中處理程序所使用的所有記憶體頁。會包含共用記憶體區域中的頁，以及同一超類別的多個子類別中處理程序所使用之檔案中的頁。由完全屬於單一子類別之多個處理程序使用的共用記憶體及檔案與該子類別有關聯。僅當源自同一超類別之不同子類別的處理程序存取共用記憶體區域或檔案時，才會將頁置於超類別的 *Shared* 子類別中。*Shared* 子類別中沒有處理程序。此子類別僅可應用實體記憶體持有率及限制，而不可有其他資源類型的持有率或限制，或指定的指派規則。由同一超類別之不同子類別中處理程序所共用的記憶體區段是要被分類到 *Shared* 子類別，還是保留在其原始子類別中，取決於原始子類別的 `localshm` 屬性值。

類別屬性

列出 WLM 類別的所有屬性。

類別名稱

最長可為 16 個字元，僅可包含大寫與小寫字母、數字及底線 (_)。

層級

0 與 9 之間的數字，用於安排類別之間資源配置的優先順序。

繼承

指定子處理程序是否從其母項繼承類別指派。

localshm

防止屬於一個類別的記憶體區段移轉到 Shared 類別。

管理者（adminuser、admingroup、authgroup）（僅超類別）

代表超類別的管理。

授權（authuser、authgroup）

代表手動將處理程序指派給類別的權利。

資源組 (rset)

就 CPU 而論（處理器組），限制給定類別有權存取的一組資源。

delshm

如果最後一個參照處理程序是因為虛擬記憶體限制而遭到結束，請刪除共用記憶體區段。

vmeforce

指出當類別達到其虛擬記憶體限制時，是要結束類別中的所有處理程序，或只結束不當的處理程序。

io_priority

指定指派給由分類至類別之執行緒所發出的 I/O 要求的優先順序。此優先順序用於在裝置層次安排 I/O 緩衝區的優先順序。如果儲存裝置不支援 I/O 優先順序，則會忽略優先順序。有效的 I/O 優先順序範圍是從 0 到 15。

相關概念

指派給類別進行工作量管理的處理程序

使用系統管理者提供的類別指派規則，將處理程序指派給類別。分類基準基於處理程序的一組屬性值，如使用者 ID、群組 ID、應用程式檔案名稱、處理程序類型及應用程式標記。

層級屬性

層級代表系統資源配置到 WLM 類別的次序。

管理者可定義最多 10 個層級（從 0 到 9），0 是最高或最重要的層級。層級 0 可用的資源數量為所有可用的系統資源。較低（較大數字）層級可用的資源數量是所有較高層級不使用的資源數量。 類別的目標用量百分比為基於其層級中作用中持有率的數目，及該層級可用的資源數量。由於層級 0 是唯一得到保證一直有可用資源的層級，所以建議將系統作業必要的處理程序分類到此層級的類別中。 若沒有為類別指定的層級值，則會將其置於層級 0。

可於超類別及子類別層次上指定層級。超類別層級用於指定超類別之間的資源配置優先順序。子類別層級用於指定同一超類別之子類別之間的資源配置優先順序。 不同超類別之次層級間沒有任何關係。

Inheritance 屬性

類別的 inheritance 屬性指出類別中的處理程序是否應在處理程序的其中一個分類屬性變更時，自動重新分類。

當使用 fork 子常式建立新處理程序時，不管是否啟用繼承， 該處理程序都會自動繼承其母項的類別。一個例外為當上代程序具有標記， 其**在 fork 時繼承標記**設為關閉，且母項類別的類別繼承為關閉時。 於此狀況下，會根據分類規則重新分類子處理程序。

當沒有為類別啟動繼承時，呼叫會變更分類規則中使用之處理程序屬性的任何服務程式後， 類別中的任何處理程序都會根據分類規則自動分類。這些呼叫中最常見的是 exec 子常式， 但可變更分類的其他子常式包括 setuid、setgid、plock、setpri 及 wlm_set_tag。 當啟用繼承時，不會根據分類規則對處理程序進行重新分類，它仍在其現行類別中。 手動指派優先於繼承，並可用於將已啟用繼承之類別中的處理程序重新分類。

inheritance 屬性的指定值可以為 yes 或 no。如果未指定，類別就不會啟用繼承。

此屬性可同時在超類別及子類別層次指定。對於給定超類別的子類別：

- 如果同時在超類別及子類別層次上將 inheritance 屬性設為 yes，則子類別中的子處理程序將保留在同一子類別中。
- 如果超類別的 inheritance 屬性設為 yes，子類別的 inheritance 屬性設為 no（或未指定），則子類別中的子處理程序將保留在同一超類別中，並將根據超類別的指派規則，分類至其一個子類別中。
- 如果超類別的 inheritance 屬性為 no（或未指定）， 子類別的 inheritance 屬性設為 yes，則子類別中的子處理程序將服從超類別的自動指派規則。
  - 如果處理程序由同一超類別中的規則所分類，則它將保留在子類別中（它不會服從子類別的指派規則）。
  - 如果處理程序由不同超類別中的超類別規則所分類，則會應用新超類別的子類別指派規則，以判斷將被指派處理程序之新超類別的子類別。
- 如果超類別及子類別的 inheritance 屬性皆設為 no（或未指定），則子類別中的子處理程序會服從標準自動指派。

localshm 屬性

可在超類別及子類別層次指定 localshm 屬性。

localshm 屬性可以防止屬於一個類別的記憶體區段在由其他類別中的處理程序存取時，被移轉到 *Shared* 超類別或子類別。屬性的可能值為 yes 或 no。值 yes 表示此類別中的共用記憶體區段必須保留在類別端，而不轉移到適當的 *Shared* 類別。如果未指定屬性，則值 no 是預設值。

記憶體區段在分頁錯誤時分類。區段在建立時被標示為屬於 *Unclassified* 超類別。在區段上第一個分頁錯誤時，此區段會被分類到與錯誤處理程序相同的類別中。如果稍後，處理程序屬於不同於此區段上區段分頁錯誤的類別，則 WLM 會考慮是否需要將區段重新分類到適當的 *Shared* 類別（超類別或子類別）。如果錯誤處理程序及區段屬於不同的超類別，則會發生下列其中一項狀況：

- 如果區段超類別的 localshm 屬性設為 yes，則該區段會保留於其現行超類別中。如果區段子類別的 localshm 屬性設為 yes，則該區段會保留於其現行子類別中。如果超類別 localshm 屬性設為 yes，但其子類別屬性設為 no，則會轉到現行超類別的 *Shared* 子類別中。
- 如果區段超類別的 localshm 屬性設為 no，則區段會轉到 *Shared* 超類別。此為預設動作。

如果錯誤處理程序及區段屬於同一超類別的不同子類別，且該區段子類別的 localshm 屬性設為 yes，則該區段會保留在現行類別（超類別及子類別）中。否則，區段會轉到超類別的 *Shared* 子類別。

當然，如果錯誤處理程序及區段屬於同一類別（同一超類別及同一子類別），則不論 localshm 屬性值為何，都不會將區段重新分類。

管理者屬性
adminuser 及 admingroup 屬性用於將超類別管理委派給使用者或使用者群組。

註：這些屬性僅對超類別有效。

adminuser 屬性指定獲授權可在超類別上執行管理作業的使用者名稱（列於 /etc/passwd）。
admingroup 屬性指定獲授權可在超類別上執行管理作業的使用者群組名稱（列於 /etc/group）。

每一個屬性僅允許一個值（使用者或群組）。可指定它們中的任意一個、不指定，或指定它們兩個。使用者或群組將有權執行下列動作：

- 建立及刪除子類別。
- 變更子類別的屬性及資源的持有率及限制。
- 定義、移除或修改子類別指派規則。
- 重新整理（更新）超類別的作用中 WLM 配置。

授權屬性
authuser 及 authgroup 屬性對全部類別都有效。它們用於指定授權手動將處理程序指派給類別（超類別或子類別）的使用者或群組。

當手動將處理程序（或處理程序群組）指派給超類別時，超類別的指派規則可用於判斷將每一處理程序指派給超類別的哪個子類別。

每一個屬性僅允許一個值（使用者或群組）。可指定它們中的任意一個、不指定，或指定它們兩個。

資源集屬性
可對任何類別指定資源組屬性（稱為 *rset*）。它的值是由系統管理者定義的資源組名稱。

rset 屬性代表系統上可用 CPU 資源的子集（處理器組）。預設值是 "system"，它提供對系統上所有可用 CPU 資源的存取權。 唯一的限制為，若對子類別指定了 *rset*，則該組中的 CPU 組必須為超類別可用之 CPU 的子集 （如需相關資訊，請參閱 **mkrset** 指令）。

註：仔細考量將資源組指派給不在層級 0 中的任何類別。 因為較低層級僅具有對較高層級不使用之資源的存取權，所以若那些 CPU 上沒有可用的 CPU 時間，則將非層級 0 類別限制在系統上 CPU 子集會導致資源缺乏。

工作量管理程式中的處理程序分類

在 WLM 中，可使用以下兩種方式中的任意一種來將處理程序分類。

- 處理程序分類屬性變更時，會使用指派規則自動指派處理程序。當 WLM 以主動模式執行時，此自動指派（無法停用）一律有效。這是分類處理程序最常用的方法。
- 具有在處理程序及目標類別上必要權限的使用者，可將所選取的處理程序或處理程序群組手動指派給某個類別。手動指派可由 WLM 指令（可直接呼叫或者透過 SMIT），或由使用「WLM 應用程式設計介面」功能的應用程式來執行。 此手動指派會置換自動指派及繼承。

工作量管理程式中的自動類別指派
將處理程序自動指派給類別時，會使用由 WLM 管理者指定的一組類別指派規則。

有兩個層次的指派規則：

- WLM 配置層次的一組指派規則，用於判斷將給定的處理程序指派給哪個超類別。
- 而每一個已定義子類別的超類別又擁有一組指派規則，用於判斷將處理程序指派給超類別的哪個子類別。

這兩個層次上的指派規則基於一組處理程序屬性的值。這些屬性如下所示：

- 處理程序使用者 ID
- 處理程序群組 ID
- 所執行應用程式（程式）的路徑名稱

· 處理程序類型（例如 32bit 或 64bit）

· 處理程序標記。

標記是定義為字串的一種處理程序屬性，應用程式可使用 WLM API 以程式來對其進行設定。

每當屬性變更時，就會執行分類，方法是針對類別指派規則檔案（稱為 rules）中給定的可能值清單，比較這些處理程序屬性值。進行比較則可以判斷符合處理程序屬性現行值的規則。

類別指派規則是包括下列欄位（以一個以上空格區隔）的字串：

項目	說明
名稱	必須包含類別檔案（對應 rules 檔案的層次）中定義的類別（超類別或子類別）名稱。類別名稱僅可包含大寫與小寫字母、數字及底線，名稱最長可為 16 個字元。沒有可為系統定義類別 Unclassified、Unmanaged 及 Shared 指定的指派規則。
保留	保留用於將來擴充。它的值必須是連字號 (-)，且必須存在。
使用者	可包含連字號 (-) 或至少一個有效的使用者名稱（定義於 /etc/passwd 檔案）。清單是由一個以上名稱（以逗點 (,) 區隔）組成。驚嘆號 (!) 可用在名稱前，以將給定的使用者從類別排除。可使用完整 Korn shell 型樣相符語法，指定型樣以符合一組使用者名稱。 若無有效的使用者名稱，則會忽略該規則。
群組	可包含連字號 (-) 或至少一個有效群組名稱（定義於 /etc/group 檔案）。清單是由一個以上名稱（以逗點 (,) 區隔）組成。驚嘆號 (!) 可用在名稱前，以將給定的群組從類別排除。可使用完整的 Korn shell 型樣相符語法來指定型樣，以使其與一組群組名稱相符。 若無有效的群組名稱，則會忽略該規則。
應用程式	可包含連字號 (-) 或應用程式路徑名稱清單。此為由內含在類別中之處理程序執行的應用程式（程式）路徑名稱。應用程式名稱將是完整路徑名稱或與路徑名稱相符的 Korn shell 型樣。清單是由一個以上路徑名稱（以逗點 (,) 區隔）組成。驚嘆號 (!) 可用在名稱前，以排除給定的應用程式。 在載入時，必須至少有一個清單中的應用程式，否則將會忽略規則。若所裝載的檔案系統包含清單中的一個以上應用程式，則一開始因這個原因而忽略的規則將在稍後變成有效。
類型	可包含連字號 (-) 或處理程序屬性清單。可能的屬性值有： · **32bit**：32 位元的處理程序 · **64bit**：64 位元的處理程序 · **plock**：呼叫 **plock** 子常式的處理程序 pin 記憶體 · **fixed**：處理程序是固定優先順序處理程序（SCHED_FIFO 或 SCHED_RR） fixed 類型僅作為分類用途。WLM 不會調整固定優先順序的處理程序或執行緒的處理器使用情況。因為固定優先順序處理程序有可能造成某一類別其他處理器之間 的撤銷，所以提供此分類屬性來釐清這些作業。此屬性也可用來報告這類處理器的耗用量。 type 欄位的值可以是一個或多個以上屬性（以加號 + 分隔屬性）的組合。32bit 與 64bit 值互斥。
標記	可能包含連字號 (-) 或應用程式標記清單。應用程式標記是最多 30 個英數字元的字串。清單是由以逗點區隔的一個以上應用程式標記值組成。

「使用者」、「群組」、「應用程式」及「標記」屬性可為屬性值分組。

當建立 (fork) 處理程序時，處理程序會維持在與其母項相同的類別中。重新分類發生的情況是，新的處理程序發出的系統呼叫可修改用於分類之處理程序的屬性之其中一個；例如，*exec*、*setuid*（及相關呼叫）、*setgid*（及相關呼叫）、*setpri* 及 *plock*。

為分類處理程序，WLM 會檢查作用中配置的頂層 rules 檔案，以判斷處理程序屬於哪個超類別。對於檔案中的每一個規則，WLM 都會針對規則中指定的值及值清單，來檢查處理程序屬性的現行值。會依照規則在

檔案中的顯示次序來檢查規則。如果找到相符項，則會將處理程序指派給於規則第一個欄位中命名的超類別。然後會以相同的方法檢查超類別的規則檔案，以判斷應將處理程序指派給哪個子類別。

對於符合規則之其中一個的處理程序，它的每一個屬性都必須符合規則中對應的欄位。下列基準清單用於判斷屬性值是否符合 `rules` 檔案欄位中的值：

- 若規則檔案中的欄位其值為連字號 (-)，則對應處理程序屬性的任何值皆是相符項。
- 對於除 *type* 外的所有屬性，如果處理程序屬性值與未排除之規則檔案（之前有 "!"）清單中的其中一個值相符，則相符項已發生。
- 對於 *type* 屬性，如果規則中其中一個值由兩個或兩個以上的值（以加號 (+) 區隔）組成，則僅當處理程序性質與所有值相符時，該處理程序才是相符項。

在超類別及子類別層次上，WLM 以它們在 `rules` 檔案中的顯示次序來查看規則，並將對應第一個規則（處理程序為該規則的相符項）之類別中的處理程序分類。因此規則檔案中規則的次序是非常重要的。請謹慎建立或修改規則檔案。

工作量管理程式中的手動類別指派

藉由使用 SMIT 或 **wlmassign** 指令，可以將處理程序或處理程序群組手動指派給超類別及（或）子類別。

請參閱 **wlmassign**，以取得相關資訊。應用程式可透過 **wlm_assign** API 函數指派處理程序。

若要將處理程序手動指派給類別或取消現有的手動指派，使用者必須具有適當的專用權層次。手動指派可分別在超類別層次、子類別層次，或者在超類別及子類別層次上執行或取消。此指派是由程式設計介面的旗標及 WLM 管理工具使用之指令行介面的一組選項所指定的。因此，可手動將處理程序僅指派給超類別、僅指派給子類別，或者指派給超類別及該超類別的子類別。在後者的狀況下，雙重指派可由不同的使用者同時（使用單一指令或 API 呼叫）或在不同的時間執行。

指派非常有彈性，但可能會讓人混淆。下列是可能案例的兩個範例。

範例 1：處理程序的首次指派

系統管理者將 *Process1* 從 *superclassA* 手動指派給 *superclassB*（僅超類別層次指派）。WLM 使用 *superclassB* 之子類別的自動指派規則，來判斷最終會將處理程序指派給哪個子類別。*Process1* 已指派給 *superclassB.subclassA*，並標示為有「僅超類別」指派。

具有適當專用權的使用者會將 *Process2* 從其現行類別 *superclassA.subclassA* 指派給同一超類別的新子類別 *superclassA.subclassB*。*Process2* 已指派給其新子類別，並標示為有「僅子類別」指派。

superclassB 之子類別的 WLM 管理者會手動將 *Process1* 重新指派給 *subclassC*（*superclassB* 的另一個子類別）。*Process1* 已重新分類到 *superclassB.subclassC* 中，且現在標示為同時有超類別及子類別層次指派。

範例 2：手動指派的重新指派或取消

在子類別層次重新指派及取消手動指派較不複雜，其僅影響子類別層次指派。

假設系統管理者要讓 *Process2* 處於具有更多資源的超類別中，並決定將 *Process2* 手動指派給 *superclassC*。在範例 1 中，已使用「僅子類別」指派將 *Process2* 手動指派給 *superclassA* 的 *subclassB*。因為 *Process2* 是指派給不同的超類別，所以先前的手動指派會變得無意義並被取消。*Process2* 現有對 *superclassC* 的「僅超類別」手動指派，並於無繼承的情況下，使用自動指派規則指派給 *superclassC* 的子類別。

現在，系統管理者決定終止從 *Process1* 到 *superclassB* 的手動指派。 *Process1* 的「超類別層次」手動指派已取消，並於無繼承的情況下，使用頂層自動指派規則將 *Process1* 指派給超類別。

如果規則未變更，則 *Process1* 會指派給 *superclassA*，且它對 *superclassB.subclassC* 的子類別層次手動指派會變得無意義並被取消。

如果因為某個原因，而使頂層規則將 *Process1* 指派給 *superclassB*，則對 *superclassB.subclassC* 的子類別層次指派仍然有效並仍然起作用。*Process1* 現在有「僅子類別」手動指派。

工作量管理程式的更新

更新 WLM 時（使用 **wlmcntrl -u** 指令），更新的配置可載入一組新的分類規則。

當發生此狀況時，經常會使用新規則重新分類處理程序。WLM 不重新分類已手動指派或位於啟動繼承之類別中的處理程序，除非它們的類別不存在於新配置中。

工作量管理程式的安全注意事項

若要將處理程序指派給類別或取消先前的手動指派，使用者必須同時具有處理程序及目標類別上的權限。

這些限制轉換成下列規則：

- root 使用者可將任何處理程序指派給任何類別。
- 具有在給定超類別之子類別上管理專用權的使用者（換言之，使用者或群組名稱與超類別屬性 `adminuser` 及 `admingroup` 中指定的使用者或群組名稱相符）可手動將任何處理程序從此超類別的子類別之其中一個重新指派給超類別的其他子類別。
- 使用者可手動將他們自己的處理程序（與同樣真實或有效之使用者 ID 相關的處理程序）指派給使用者具有手動指派專用權的子類別（換言之，使用者或群組名稱與超類別或子類別之屬性 `authuser` 及 `authgroup` 中指定的使用者或群組名稱相符）。

若要修改或終止手動指派，使用者必須至少具有與發出上次手動指派之人員相同的專用權層次。

使用工作量管理程式進行資源管理

WLM 會按每一類別來監視及調節系統上作用中執行緒及處理程序的資源使用率。您可針對 WLM 管理的每一資源類型，來設定每一類別的最小或最大限制，並為每一資源設定每一類別的目標值。

此目標代表對類別中工作的最佳資源數量。超類別層次上的持有率及限制是指系統上每一種可用資源的總數量。在子類別層次上，持有率及限制是指子類別所在的超類別可用之每一種資源的數量（超類別目標）。類別階層是一種方法，用於在使用者群組（超類別）之間劃分系統資源，並將這一份資源的共用的管理委託給超類別管理者。之後，每一位超類別管理者都可藉由建立子類別並為這些子類別定義資源權利，而在群組中的使用者之間重新配送這個數量的資源。

工作量管理程式中的資源類型

WLM 按百分比用量來管理三種類型的資源。

項目	說明
類別中執行緒的 CPU 使用率	這是類別中每一個執行緒使用之所有 CPU 循環的總和。
類別中處理程序的實體記憶體使用率	此為屬於類別中處理程序之所有記憶體頁的總和。
類別的磁碟 I/O 頻寬	這是在類別存取的每一個磁碟裝置上，由類別中執行緒啟動之所有 I/O 的頻寬（以每秒 512 位元組區塊為單位）。

WLM 每秒都為每一資源計算上一秒期間的每一類別使用率，以總可用資源的百分比來表示，如下所示：

- 對於 CPU，每秒可用的 CPU 時間總量等於 1 秒乘以系統上的 CPU 數目。例如，在八向 SMP 上，如果某個類別的所有執行緒結合起來會在上一秒期間耗用掉 2 秒的 CPU 時間，則這表示百分比為 2/8 = 25%。WLM 用於調節的百分比是此「瞬間」每秒資源使用率的數秒衰減平均值。
- 對於實體記憶體，處理程序在任何給定時間可用的實體記憶體總量等於系統上實際存在的記憶體總頁數減去已定位的頁數。已定位的頁不是由 WLM 管理，因為無法從類別中挪用這些頁，亦無法將其提供給其他類別以調節記憶體使用率。類別的記憶體使用率是類別中所有處理程序所擁有之未定位記憶體頁數與系統上可用頁數的比例，以百分比表示。
- 對於磁碟 I/O，主要問題是為裝置判斷有意義的可用頻寬。當磁碟 100% 處於工作中時，一個應用程式正在執行循序 I/O 的狀況與數個應用程式正在產生隨機 I/O 的狀況相比，磁碟的產量（以每秒區塊數計）完全不同。如果您僅使用了為循序 I/O 狀況測量的最大產量（作為裝置可用的 I/O 頻寬值），來計算在隨機 I/O 狀況下裝置使用率的百分比，則您可能會被誤導，認為該裝置處於 20% 使用率（而實際上它是處於 100% 使用率）。

為取得每一類別磁碟使用率更精確可靠的百分比，WLM 會使用磁碟驅動程式提供的統計資料（使用 **iostat** 指令顯示），該統計資料為每一磁碟裝置提供在上一秒期間，該裝置處於工作中時間的百分比。WLM 會計算在上一秒期間由存取此裝置之所有類別在裝置上已讀取或寫入的總區塊數目、每一個類別已讀取或寫入多少區塊，以及該裝置使用率的百分比為何。 然後，WLM 會計算每一類別所使用的磁碟產量百分比。

例如，如果在上一秒期間讀取或寫入的總區塊數是 1000，且該裝置曾經 70% 處於工作中，則這表示讀取或寫入 100 個區塊的某個類別使用了磁碟頻寬的 7%。與 CPU 時間（另一種可重複使用的資源）類似，WLM 用於其磁碟 I/O 調節的值亦是這些每秒百分比在幾秒當中的衰減平均值。

對於磁碟 I/O 資源，持有率及限制適用於由類別個別存取的每一磁碟裝置。調節是針對每一裝置獨立完成的。這表示某個類別可能在某一個裝置上高於其權利，並將調節此裝置的 I/O（雖然此類別在其他磁碟上低於其權利），但不會限制此其他裝置的 I/O。

WLM 支援按總計用量的資源統計及調節。 有兩種類型的資源可以此方式調節：類別總計及處理程序總計。

類別總計
可為類別中處理程序、執行緒及登入階段作業的數目指定每一類別限制。 將這些限制指定為可在任意時刻存在於類別中之每一資源的絕對數目。 嚴格地強制執行這些限制；當類別已達到其對這些資源之一的限制時，任何建立該資源之另一個案例的嘗試都將失敗。 類別中任何處理程序的作業都將持續失敗，直到該類別低於其對該資源的指定限制為止。

處理程序總計
可為 CPU 時間總量、磁碟 I/O 區塊總數，及登入階段作業的連接時間指定每一處理程序限制。 這些限制指定於類別層次，但可個別應用到類別中的每一處理程序（每一處理程序均可使用此數量）。 這些用量值為累加的，因此代表處理程序在其使用期限內所使用之每一特定資源的總量。 一旦處理程序超出其對任何資源的總計限制，將被終止。將向處理程序傳送 SIGTERM 信號，若處理程序捕捉到此信號且在 5 秒寬限期後不結束，將向處理程序傳送 SIGKILL 信號。 若登入階段作業已達到其連接時間限制的 90%，則會將警告訊息寫入控制終端機，以警告使用者不久將終止階段作業。

工作量管理程式中的目標持有率
類別的目標（或想要的）資源用量百分比由其對特定資源擁有的持有率數目決定。

持有率代表類別應取得多少特定資源（相對於類別層級中的其他類別）。 對於特定資源，類別的目標百分比為其持有率數目除以其層級中作用中持有率數目。 若同時還在使用限制，則目標會限制於範圍 [最小值, 軟性最大值]。 若所計算的目標超出此範圍，則會將其設為適當的上界/下界（請參閱「資源限制」）。 作用中持有率數目為其中至少有一個作用中處理程序之所有類別的持有率總數。因為作用中持有率數目是動態的，所以目標也是動態的。 若某類別為層級中唯一的作用中類別，則其目標將為 100% 的層級可用資源數量。

例如，假設層級 0 中有三個超類別 A、B 和 C，它們對特定資源的持有率分別是 15、10 及 5，其目標將會是：

target(A) = 15/30 = 50%
target(B) = 10/30 = 33%
target(C) = 5/30 = 17%

若於一段時間後，類別 B 變為非作用中（沒有作用中的處理程序），則將自動調整類別 A 及 C 的目標：

target(A) = 15/20 = 75%
target(C) = 5/20 = 25%

如您可看到的，持有率代表自我調適百分比，其可讓配置給某類別的資源平均配送，或於其變成作用中/非作用中時由其他類別取得資源。

若要允許高度彈性，類別的持有率數目可為介於 1 與 65535 之間的任何數字。 可為超類別及子類別指定持有率。對於超類別，持有率相對於同一層級中所有其他作用中超類別。 對於子類別，持有率相對於同一層級內同一超類別中所有其他作用中子類別。 一個超類別之一個子類別的持有率與其他超類別之子類別的持有率沒有關係。

於某些狀況下，可能應使類別的目標獨立於作用中持有率的數目。 若要這樣做，可為持有率數目指定值 "-"。於此狀況下， 將不為該資源調整類別，此表示其無持有率，且其目標與作用中持有率數目無關。 會將其目標設為（層級可用的資源 - 層級中所有其他類別的最小值總和）。 然後從同一層級中其他類別可用的資源得出此目標或實際用量（取其中較低的一個）。

例如，假設類別 A、B、C 及 D 針對特定資源分別具有持有率 "-"、200、150 及 100。 全部類別皆處於作用中，而類別 A 正在使用 50% 的資源：

target(A) = unregulated = 100%
target(B) = 200/450 * available = 44% * 50% = 22%

target(C) = 150/450 * available = 33% * 50% = 17%
target(D) = 100/450 * available = 22% * 50% = 11%

因為未調整類別 A 且類別 A 正在使用 50% 之可用資源，所以其他類別僅有 50% 的資源可供使用，它們的目標是依據此百分比來計算的。因為類別 A 將恆低於其目標 (100%)，所以它將一直擁有高於 同一層級中所有其他類別（處於或高於其目標）之優先順序（請參閱第 127 頁的『工作量管理程式中的類別優先順序』，以取得詳細資訊）。

註：使資源不調節類別，與將類別置於較高層級不同。此處所列之下列行為對於未調節的類別（在同一層級中）是正確的，而若將該類別置於較高層級則下列行為是不正確的：

- 因為持有率是依每一資源所定義的，所以可不為一或多個資源調整類別，而為其他資源進行調整。
- 接受同一層級中其他類別的最小限制。較高層級不接受較低層級中所指定的最小值。
- 即使缺少具有持有率之類別的最小限制，未調節類別的用量也在某種程度上相依於具有持有率的類別，因為它們正在為層級可用的部分資源而競爭。必須做實驗以查看給定工作量下的行為。

若未指定持有率數目，則會使用預設值 "-"，而不會為該資源調整類別。 請注意在 WLM 第一版中，預設持有率值為 1（若未指定）。

持有率是對每一類別、為所有資源類型指定的。持有率在 **shares** 檔案的段落中有指定。例如：

```
shares
classname:
   CPU      =   2
   memory   =   4
   diskIO   =   3
```

工作量管理程式中的資源限制規格
除了使用持有率來定義相對資源權利之外，WLM 還提供為類別指定資源限制的能力。 資源限制可使管理者對資源配置有更多的控制。以百分比指定的這些限制為相對於 類別所在層級可用的資源數量。

對基於百分比的調節，有三種類型的限制：

最小值
此為應提供給類別之資源的最小數量。若實際的類別用量低於此值，則會將對資源的最高優先順序存取權提供給類別。可能的值為 0 到 100，0 是預設值（若未指定）。

軟性最大值
此為當存在對該資源的競爭時，類別可使用的資源最大數量。如果類別用量超出此值，則會將其層級中最低優先順序提供給類別。若資源無競爭（來自同一層級中的其他類別），則會允許類別隨意使用它想要的資源數量。可能的值為 1 到 100，100 是預設值（若未指定）。

硬性最大值
此為類別可使用的資源最大數量（即使在沒有競爭時）。若類別達到此限制，則不會允許它繼續使用更多的資源，直到它的用量百分比低於該限制時為止。 可能的值為 1 到 100，100 是預設值（若未指定）。

資源限制值是由每一類別的段落內之資源類型在資源限制檔案內所指定的。會將限制指定為軟性最大值到最小值之間的範圍（以連字號 (-) 區隔並忽略空格）。 指定硬性最大值時，它會接在軟性最大值之後（以分號 (;) 區隔）。每一個限制值後面都接著百分比符號 (%)。

下列是使用規則檔案的範例：

- 如果來自群組 acct3 的使用者 joe 執行 `/bin/vi`，則會將處理程序置於超類別 acctg 中。
- 如果來自群組 dev 的使用者 sue 執行 `/bin/emacs`，則處理程序會在超類別 devlt（群組 ID 相符）中，但不會被分類到 editors 子類別中，因為使用者 sue 被排除在該類別之外。此處理程序將前往 devlt。（預設值。）
- 當 DB 管理者使用 `oracle` 使用者 ID 及 `dbm` 群組 ID 啟動 `/usr/sbin/oracle` 時，會將處理程序分類到 Default 超類別中，以便為資料庫 `DB1` 服務。僅當處理程序將其標記設為 `_DB1` 時，才會將它指派給超類別 `db1`。

限制是針對每一類別為所有資源類型在 limits 檔案的段落中指定的。例如：

```
shares
classname:
      CPU     =   0%-50%;80%
      memory  =   10%-30%;50%
```

在此範例中，沒有為磁碟 I/O 設定限制。使用系統預設值，會轉換成下列內容：

```
diskIO   =   0%-100%;100%
```

所有前述範例都假設所說明的超類別及子類別都未將繼承屬性設為 yes。 否則，新的處理程序將僅從其母項繼承超類別或子類別。

下列是 WLM 對資源限制值的僅有限制：

· 最小限制必須小於或等於軟性最大限制。

· 軟性最大限制必須小於或等於強迫最大限制。

· 層級內所有超類別的最小值總和不可超出 100。

· 層級內給定超類別之所有子類別的最小值總和不可超出 100。

當有強迫記憶體限制的類別已達到此限制並要求更多頁時，會起始 VMM 頁置換演算法 (LRU) 並從限制類別「挪用」頁，藉而在分送新頁之前會先將其頁數減至強迫最大值以下。此行為是正確的，但額外分頁活動（甚至在有大量的可用頁時亦可能發生）會影響系統的一般效能。在為任何類別強加強迫記憶體最大值之前，建議使用其他類別的最小記憶體限制。

因為低於其最小值的類別在其層級中有最高優先順序，所以應根據同一層級中其他類別的資源需求，使最小值總和保持在合理的層次。

如果層級內最小限制總和的限制小於或等於 100，則表示一律允許最高優先順序層級中的類別取得最多到其最小限制的資源。WLM 不保證類別將實際達到它的最小限制。這取決於類別中的處理程序如何使用它們的資源以及起作用的其他限制。例如，因為類別無法取得足夠的記憶體，所以它可能不會達到其最小 CPU 權利。

對於實體記憶體，設定最小記憶體限制可為類別（至少針對在最高優先順序層級中的那些類別）處理程序的記憶體頁提供部分保護。如果類別低於其最小限制，則不應挪用其頁，除非所有作用中類別都低於它們的最小限制，且它們中的一個類別有要求更多的頁。最高層級中類別的頁永遠不應被挪用（即使該類別低於其最小值）。為互動式工作類別設定記憶體最小限制，有助於確保它們的頁不會在連續啟動之間被全部挪用（甚至當記憶體用量緊湊時），並會改進回應時間。

小心： 硬性最大限制若不適當使用，會對系統或應用程式效能產生很大影響。由於強制使用硬性限制會導致閒置系統資源，因此在大部分情況下，軟性最大限制更適當。 硬性最大限制的一種用途是限制較高層級的用量，以使某資源可供較低層級使用， 儘管將需要資源的應用程式置於較高層級可能是更好的解決方案。

可於限制檔案中指定總計限制，在下表中彙總各值及單位：

表 *8.* 工作量管理程式的資源限制

資源	允許的單位	預設單位	最大值	最小值
totalCPU	s、m、h、d、w	s	$2^{30} - 1$s	10 s
totalDiskIO	KB、MB、TB、PB、 EB	KB	$(2^{63} - 1)$ * 512/1024 KB	1 MB
totalConnectTime	s、m、h、d、w	s	$2^{63} - 1$ s	5 m
totalProcesses	–	–	$2^{63} - 1$	2
totalThreads	–	–	$2^{63} - 1$	2
totalLogins	–	–	$2^{63} - 1$	1

註：單位指定元不區分大小寫。s = 秒，m = 分鐘，h = 小時，d = 天，w = 週，KB = KB， MK = MB，...等等。

範例限制段落如下：

```
BadUserClass:
          totalCPU = 1m
          totalConnectTime = 1h
```

可使用以上表格中的任一值來指定總計限制，但有下列限制：

· 若指定，則 totalThreads 的值必須至少是 totalProcesses 的值。

· 若已指定 totalThreads 而未指定 totalProcesses，則會將 totalProcesses 的限制設為 totalThreads 的值。

可在超類別及子類別層次指定總計限制。檢查限制時，會先檢查子類別限制，再檢查超類別限制。 若已指定這兩個限制，則會強制執行這兩個限制中較低的一個。若指定的子類別限制高於其相關聯的 超類別限制，則載入配置時會發出警告，但仍將載入配置。這對於類別總計限制是很有意義的，因為限制是絕對的（而不是相對於超類別），一個子類別可使用超類別可用的所有資源。 若未指定，則所有總計限制的預設值為 "-"，表示無限制。根據預設值，執行 WLM 時，將啟用類別、處理程序總計統計及調節。**wlmcntrl** 指令的 **-T [class|proc]** 選項可用來停用總計統計及調節。

工作量管理程式中的類別優先順序

WLM 藉由將優先順序提供給每一資源的每一類別，將資源配置給類別。

類別的優先順序是動態的，並以層級、持有率、限制及類別的現行用量為基礎。 於任何給定時間，會將對資源的優先存取權提供給最高優先順序類別。 在最高層次上，層級代表類別優先順序之非重疊範圍。 將一律保證層級 0 中之類別具有高於層級 1 中類別的優先順序（除非超出硬性最大值）。

判定類別優先順序時，WLM 會以下列優先順序（從最高到最低）強制執行其限制：

硬性限制
若類別用量超出資源的硬性最大值，則會提供類別資源可能之最低優先順序，並拒絕存取類別，直到其用量低於此限制為止。

層級
於缺少硬性限制的狀況下，類別的優先順序將受其層級所允許最小及最大優先順序限制。

軟性限制
若類別用量低於資源軟性最大限制的最小值，則會將層級中最高優先順序提供給類別。 若類別用量高於軟性最大值，則會將層級中最低優先順序提供給類別。

shares
這些持有率用來為每一資源計算類別用量目標。類別用量低於目標時，類別優先順序會提高；類別用量高於目標時，類別優先順序會降低。請注意，軟性限制有較高優先順序，在適用時，會依據它們來判斷類別的優先順序。

即使持有率與限制可用於每一類別及每一資源，但若每一類別僅使用其中一個，則結果會更容易預測。

專用處理器資源集

專用處理器資源集 (XRSETs) 可讓管理者確保提供重要作業所需的資源。XRSET 是一種具名的資源集，可變更其所含的所有 CPU 的行為。一旦 CPU 設成專用，便只能執行明確導向它的程式。

建立 XRSET

您必須是 root 使用者，才能建立 XRSET。使用 **mkrset** 指令，可以在 sysxrset 名稱空間內建立資源集。例如，指令 **mkrset -c 1-3 sysxrset/set1** 會針對 CPU 1、2、3 建立 XRSET。rs_registername() 子常式也可以用來建立 XRSET。

判斷系統是否已定義 XRSET

lsrset -v -n sysxrset 指令會顯示系統上所有定義的 XRSET。（目前並無程式設計 API 來執行這個功能。）

刪除 XRSET

您必須是 root 使用者，才能刪除 XRSET。**rmrset** 指令可用來刪除 XRSET。rs_discardname() 子常式也可以用來刪除 XRSET。

重新啟動系統

重新啟動系統時，任何先前設定的 XRSET 會從登錄中移除，也不再有效。

指定 XRSET 的作業

有多種方式，可將作業標示為可以專門使用處理器。您可以使用 **attachrset** 和 **execrset** 指令，來指定包含專用處理器的資源集合。含專用處理器的資源會與 WLM 類別關聯。歸類為這個 WLM 類別的作業將會使用資源集中指定的專用處理器。

使用 XRSET 搭配 Bindprocessor 及 _system_configuration.ncpus

您不能使用 bindprocessor 讓作業於專用處理器上執行。只有以資源集為基礎的連接裝置可以使作業執行於專用處理器上。

建立 XRSET 時，並不會變更系統配置中的 CPU 數目（**_system_configuration.ncpus** 欄位）。系統中仍然有 NCPU。

當程式使用 bindprocessor 系統呼叫 NCPU 時，XRSET 中的 CPU 會失效，並顯示 EINVAL 錯誤。您可以連結至 **bindprocessor** 指令的查詢選項所傳回的任何 ID。查詢選項 (bindprocessor -q) 只會傳回有效的連結 ID，不含那些與專用 CPU 關聯的 ID。

例如，如果系統中有 10 個連線的 CPU，其中三個屬於 XRSET，那麼連結 ID 範圍為 0 至 6 的 bindprocessor 至 CPU 便會成功。使用連結 ID 範圍 7 至 9 的 bindprocessor 至 CPU，會收到 EINVAL 錯誤。

使用 XRSET 搭配動態 CPU 重新配置作業

通常，「動態 CPU 重新配置」並不受專用處理器影響。不過，建立 XRSET 並指派作業至這些專用處理器，可能會防止移除 CPU。動態新增至系統的 CPU 可能會以一般使用或專用處理器形式進入系統。如果有一個含邏輯 CPU ID 的 XRSET 進入系統，則這些 CPU 會以專用形式進入系統。

工作量管理程式中的資源集

WLM 使用資源組（或 *rset*），將給定類別中的處理程序限制為系統實體資源的子集。於 WLM 中，受管理的實體資源為記憶體及處理器。 有效的資源組由記憶體及至少一個處理器組成。

藉由使用 SMIT，系統管理者可以定義及命名包含系統上可用資源子集的資源集。 然後，使用 WLM 管理介面，root 使用者或指定的超類別管理者可使用資源組的名稱，作為 WLM 類別的 rset 屬性。從那時起，僅在資源組中的一個處理器上指派指定給此 WLM 類別的每一個處理程序，為 CPU 資源有效地配送工作量。

所有現行系統僅有一個由所有資源組共用的記憶體網域，因此這種方法並未實際配送記憶體中的工作量。

工作量管理程式中的資源集登錄

rset 登錄服務可讓系統管理者定義及命名資源組，這樣其他使用者或應用程式就可以使用它們。

為降低名稱衝突的風險，登錄支援兩層命名架構。資源組的名稱為 *name_space*/*rset_name* 的形式。*name_space* 及 *rset_name* 兩者的長度均可為 255 個字元、區分大小寫，且僅可包含大寫及小寫字母、數字、底線及句點 (.)。**sys** 的 *name_space* 由作業系統所保留， 用於代表系統資源的 rset 定義。

rset 定義名稱在登錄名稱空間內是唯一的。 使用與現有 rset 定義相同的名稱，將新的 rset 定義新增至登錄會導致新的定義取代現有定義（若具有適當的許可權及專用權）。只有 root 可以使用 SMIT 來建立、修改及刪除資源集，並更新核心內 rset 資料庫。

每一個 rset 定義皆有擁有者（使用者 ID）、 群組（群組 ID），及與其相關的存取許可權。這些是於建立 rset 定義時指定， 其為存取控制而存在。如同檔案一樣，針對擁有者、群組及其他人存在個別存取許可權， 定義是否已授與讀取及/或寫入許可權。讀取許可權允許擷取 rset 定義，而寫入許可權允許修改或移除 rset 定義。

系統管理者所定義的 rset 定義，會保留在 /etc/rsets 段落檔中。此檔案的格式沒有說明，而且使用者必須透過 SMIT 介面來操作 rset，以避免將來因修改檔案格式，而產生潛在的相容性問題。如同 WLM 類別定義一樣，必須先將 rset 定義載入核心資料結構，WLM 才能使用它們。

設定工作量管理程式

可使用 SMIT 或 WLM 指令行介面來輸入類別定義、類別屬性、共用及限制，以及自動類別指派規則。這些定義及規則保存在純文字檔中，亦可使用文字編輯器來建立或修改它們。

這些檔案（稱為 *WLM* 特性檔案）是保留在 /etc/wlm 的子目錄中。說明超類別及其相關聯子類別的一組檔案會定義 WLM 配置。 WLM **Config** 配置的檔案位於 /etc/wlm/Config。此目錄包含超類別的 WLM 參數的定義。檔案名稱是 description、classes、shares、limits 及 rules。此目錄可能也包含具有超類別名稱的子目錄，會在其中保存子類別定義。例如，對於 WLM 配置的 **Config** 的超類別 *Super*，目錄 /etc/wlm/Config/Super 中會包含超類別 *Super* 的子類別內容檔。檔案名稱是 description、classes、shares、limits 及 rules。

在系統管理者已定義 WLM 配置之後，可以使用 **smit wlmmanage** 捷徑或 **wlmcntrl** 指令，使該配置成為作用中配置。

您可定義多組特性檔案，定義工作量管理的不同配置。這些配置通常位於 /etc/wlm 的子目錄中。符號鏈結 /etc/wlm/current 會指向包含現行配置檔的目錄。當以一組指定配置檔啟動 WLM 時，此鏈結會由 **wlmcntrl** 指令更新。

工作量管理程式配置的應用程式需求

定義配置的第一個階段需要瞭解使用者及其計算需要，及系統上的應用程式、其資源需求及業務的基本需求（例如，哪些作業是重要的，可對哪些給定較低的優先順序）。您可以根據此瞭解，先定義超類別，然後定義子類別。

設定優先順序相依於 WLM 在您組織中提供何種功能服務。在伺服器合併的狀況下，您可能已經瞭解應用程式、使用者及其資源基本要求，您也許能夠跳過或縮短部分步驟。

WLM 可讓您按使用者或群組、應用程式、類型、標記，或這些屬性的組合，來分類處理程序。因為 WLM 會調節類別之間的資源使用率，所以系統管理者應將具有相同資源使用率型樣的應用程式及使用者分組到同一類別。例如，您可能想要將互動式工作（通常使用很少的 CPU 時間，但需要快速回應時間）與批次類型工作（通常是 CPU 及記憶體非常密集的）分隔。這與您需要在資料庫環境中將 OLTP 型資料流量與資料開採的重負荷查詢分隔是相同的。

此步驟是使用 SMIT 或指令行介面完成的。對於最初幾次，使用 SMIT 來引導您完成建立首次 WLM 配置（包括定義超類別並設定其屬性）的步驟可能是好辦法。第一次時，您可設定部分屬性，並讓其他屬性保留為其預設值。對於資源持有率及限制也是同樣。所有這些類別性質可在稍後進行動態修改。

然後您可以被動模式啟動 WLM、檢查分類，並開始複查應用程式的資源使用率型樣。

使用 **wlmcheck** 指令或相對應的 SMIT 功能表來驗證配置。然後在新定義的配置上以被動模式啟動 WLM。WLM 將分類所有現有的處理程序（及從該時起建立的所有處理程序），並開始編譯有關各種類別之 CPU、記憶體及磁碟 I/O 使用率的統計資料。WLM 將不會嘗試調節此資源使用率。

驗證各種處理程序在適當的類別中，如系統管理者預期的方式進行分類（使用 **ps** 指令的 **-o** 旗標）。如果部分處理程序未如預期那樣分類，則您可修改指派規則，或為部分類別設定繼承位元（如果您想讓新的處理程序繼續保留在與其母項相同的類別中），並更新 WLM。您可重複處理程序，直到您對分類的這個第一個層次（超類別）感到滿意為止。

以被動模式執行 WLM 及重新整理 WLM（一律以被動模式）涉及低風險，並有低額外負擔作業，且可在正式作業系統上安全進行而無需干擾正常系統作業。若要啟動並重新整理 WLM，請使用 **wlmcntrl** 指令（從指令行或從 SMIT 呼叫）。

藉由使用 **wlmstat** 指令，可以被動模式執行 WLM 來收集統計資料。針對超類別，可定期使用 **wlmstat** 指令，將每個類別資源使用率顯示為可用資源總計的百分比。這可讓您長期監視系統，以複查主要應用程式的資源使用率。

工作量管理程式中的層級、持有率及限制

使用從以被動模式執行的 WLM 所收集的資料以及您的業務目標，來決定將哪個層級號碼提供給每一超類別，及應將每個資源的何種持有率提供給不同的類別。

對於部分類別，您可能想要定義最小或最大限制。調整持有率及層級號碼，以達到您的資源配置目標。對於無法僅使用持有率解決的狀況，請保留限制。 您亦可決定需要新增子類別。

- 針對一般有低資源使用率、但當由外部事件啟動時需要快速回應時間的應用程式，使用最小限制。在記憶體用量緊湊的狀況下，互動式工作面臨的問題之一是，它們的頁在非作用中期間被挪用。如果類別處於層級 0，則記憶體最小限制可用於保護互動式工作的部分頁。
- 使用最大限制來包含部分資源密集、低優先順序的工作。除非因為其他原因而分割系統資源，否則強迫最大值通常對非重複使用資源（如記憶體）有意義。這是由於它將資料輸出寫入分頁空間所花費的時間（如果較高優先順序類別需要第一個類別可能已使用的頁）。對於 CPU 使用情形，您可使用層級或軟性最大值，以確定是否立即將 CPU 時間指派給較高優先順序類別。

建立及調整子類別的參數時，您可以只針對給定超類別的子類別（不影響其他超類別中的使用者及應用程式）重新整理 WLM，直到您對系統行為滿意為止。

您亦可根據業務需要，使用不同參數來定義其他配置。這樣做可藉由複製及修改現有配置來節省時間。

細部調整工作量管理程式配置

使用 **wlmstat** 指令監視系統，並驗證由 WLM 進行的調節符合您的目標，以及部分應用程式未被過度地剝奪資源（而其他應用程式取得多於它們應得的資源）。如果是這樣的狀況，則調整持有率並重新整理 WLM。

當您監視並調整共用、限制及層級號碼時，請決定您是否要針對部分或所有超類別委託子類別的管理。然後管理者可監視並設定子類別持有率、限制及層級號碼。

每一超類別的管理者可針對每一超類別的子類別重複此處理程序。唯一的差異是 WLM 無法以被動模式僅在子類別層次執行。必須使用主動模式的 WLM 來進行子類別配置及調整。不影響超類別中使用者及應用程式的一個方法是，啟動層級號碼，並以子類別的預設值（'-'（連字號）代表持有率，0% 代表最小值，100% 代表軟性及強迫最大值）啟動子類別的持有率及限制。使用這些設定，WLM 將不會調節子類別之間的資源配置。

疑難排解工作量管理程式

如果使用現行配置未得到想要的行為，則可能需要調整 WLM 配置。

每一個類別的用量值都可使用 **wlmstat** 指令予以監視。收集這些資料並加以分析可協助判斷可能需要對配置進行哪些變更。更新配置之後，請使用 **wlmcntrl -u** 指令來更新作用中的 WLM 配置。

下列指引可協助您決定如何變更配置：

- 如果層級中的作用中持有率數目隨著時間而變化很大，則可不為類別提供資源持有率，如此它可擁有獨立於作用中持有率數目之外的用量目標。此技術對需要資源高優先順序存取權的重要類別非常有用。
- 如果您需要保證某些數量資源的存取權，請指定最小限制。此技術對不耗用大量資源但必須快速回應外部事件的互動式工作非常有用。
- 如果您需要限制資源存取權，但持有率未提供足夠的控制，請指定最大限制。在大部分情況下，軟性最大限制就足夠了，但硬性最大限制可用於嚴格強迫施行。因為硬性最大限制可導致系統資源浪費，且可增加記憶體調節時所使用的分頁活動，所以您應在強制任何硬性限制之前，先強制其他類別的最小限制。
- 如果較不重要的工作正在干擾較重要的工作，請將較不重要的工作置於較低層級中。此技術可確保較不重要的工作擁有較低優先順序，且無法在較重要的工作執行時與其爭奪可用的資源。
- 如果類別無法達到其資源用量目標，請檢查此狀況是否是由競爭另一個資源而引起的。如果是，請變更所競爭資源的類別配置。
- 如果類別中的處理程序在它們的行為或資源用量方面大不相同，請建立更多的類別來獲得更精密的控制。此外，可能需要為每一個重要的應用程式建立個別類別。
- 如果您的分析顯示一個類別所需的資源相依於另一個類別的用量，請相對應地重新配置資源。例如，如果 ClassZ 所需的資源數量相依於 ClassA 可處理的工作要求數目，則必須保證 ClassA 可存取足夠的資源，以供 ClassZ 所需。
- 如果一或多個應用程式始終未收到足夠的資源以充分執行，則您的唯一選擇可能是減少系統上的工作量。

註： 您可以為超類別定義一個 *adminuser*，來減少 WLM 管理者需要執行的工作量。測試並調整完頂層配置之後，超類別的管理使用者可進行後續變更（包括建立及配置子類別），以滿足他們的特定需要。

工作量管理程式 API

應用程式可以使用 WLM API，這是 /usr/lib/libwlm.a 程式庫中的一組常式，來執行 WLM 管理者使用 WLM 指令行介面執行的所有作業。

這些功能包括：

· 建立、修改或刪除類別

· 變更類別屬性或資源持有率及限制

· 移除類別

· 手動指派處理程序給類別

· 擷取 WLM 統計資料

API 可讓應用程式設定一個由應用程式定義的分類屬性，稱為標記。使用系統管理者（透過應用程式使用者文件）提供的一組值來設定此標記，可允許同一應用程式的數個案例之間有區別。因此可以不同的資源權利將不同的類別加以分類。

此外，**wlm_set_tag** 常式還可以讓應用程式設定應用程式標記，及指定此標記是否應在 **fork** 或 **exec** 時由子處理程序繼承。也可以使用 **wlm_set_thread** 標記，對執行緒指派應用程式標記。執行緒的應用程式標記可以跨 **fork**、**exec** 或 **pthread_create** 子常式來繼承。程式庫提供對多執行緒 32 位元或 64 位元應用程式的支援。

應用程式標記

應用程式標記是一個字串，用作處理程序或執行緒自動分類的分類基準之一（使用 `rules` 檔案）。除了系統定義的基準（如 *user*、*group*、*application* 及 *type*）外，此標記基本上還提供了應用程式定義的分類基準。

當應用程式處理程序或執行緒有設定其標記時，就會使用對目前作用中 WLM 配置有效的超類別及子類別規則，立即重新分類。然後 WLM 會使用包括新標記的所有處理程序屬性，來複查指派規則，同時尋找相符項。

此標記必須出現在一個以上指派規則中才有效。每一個應用程式對各種標記的格式及用法必須在應用程式管理文件中明確指定，且必須讓 WLM 管理者熟知，以便 WLM 管理者可使用指派規則中標記的各種值，來區分同一應用程式的不同案例。

因為關於使用者要使用應用程式處理程序的何組性質來分類那些處理程序，不同的使用者可能有不同的基本要求，所以建議應用程式最好提供可用於建置標記的一組配置或執行時期屬性。應用程式管理者可將此標記的格式指定給應用程式。可用於讓標記及語法指定 WLM 標記格式的屬性取決於應用程式，且是應用程式提供者的責任。

例如，資料庫伺服器案例可判斷它正在哪個資料庫上工作 (*db_name*)，以及給定的使用者是透過哪個 TCP 埠連接的 (*port_num*)。管理者可能有下列任一優先順序：

· 為存取不同資料庫的處理程序建立不同的類別，將不同的資源權利提供給每一個類別

· 分隔服務於不同來源之遠端要求的處理程序，並使用埠號作為分類屬性

· 為每一個資料庫建立一個超類別，並在每一個超類別中建立每一埠號的子類別。

滿足這些不同需要的一個方法是指定標記的內容及格式。於此範例中，假設可將標記傳遞給配置檔中的應用程式，或傳遞給執行時期參數（如 `WLM_TAG=$db_name` 或 `WLM_TAG=$db_name_$port_num`）。

當設定它的標記時，應用程式可指定此標記是否將由其子項繼承，以便可在同一類別中分類由特定應用程式實例建立的所有處理程序。設定標記繼承是使用應用程式標記的最常見方法。

下列是如何使用應用程式標記的範例。在此範例中，資料庫標記與資料庫名稱相同。因此，在兩個不同資料庫上工作之伺服器的兩個案例將設定兩個不同的標記，例如 db1 及 db2。

系統管理者可建立兩個不同的類別 dbserv1 及 dbserv2，並使用標記在這些類別中分類這兩個資料庫伺服器（如果使用標記繼承，則還分類其所有子項）。之後，將可能根據特定的業務目標，將不同的資源權利提供給每一個類別。

對應的指派規則可能與下列類似：

```
* class   resvd  user  group     application     type  tag
*
dbserv1     -    -     dbadm     /usr/sbin/dbserv  -   db1
dbserv2     -    -     dbadm     /usr/sbin/dbserv  -   db2
```

應用程式設計介面類型

下列是工作量管理程式 (WLM) 應用程式設計介面類型。

類別管理 API

WLM API 將下列能力提供給應用程式：

- 查詢給定 WLM 配置現有類別的名稱及性質 (`wlm_read_classes`)。
- 為給定的 WLM 配置建立新類別，定義類別各種屬性（如層級及繼承）的值，以及由 WLM 管理之資源（如 CPU、實體記憶體及區塊 I/O）的持有率及限制 (`wlm_create_class`)。
- 變更給定 WLM 配置現有類別的性質，包括類別屬性及資源持有率與限制 (`wlm_change_class`)。
- 刪除給定配置的現有類別 (`wlm_delete_class`)。

這些變更將僅應用於指定 WLM 配置的特性檔。選用性地，藉由將空字串指定為配置名稱，可能將變更僅應用到核心內類別，這會導致作用中配置狀態的立即更新。

API 呼叫需要與呼叫者相同的專用權層次（指令行或者 SMIT 介面將需要該專用權層次），如下所示：

- 任何使用者都可讀取類別名稱及性質
- 僅 root 使用者可建立、修改或刪除超類別
- 僅 root 使用者或指定的超類別管理者（超類別屬性 `adminuser` 或 `admingroup`）可建立、修改或刪除給定超類別的子類別。

在完成 WLM 管理的狀況下（方法為 WLM 管理者透過指令行及管理工具來完成，以及應用程式透過 API 來完成），必須採用部分注意事項。兩個介面共用超類別及子類別名稱的同一名稱空間，以及超類別及子類別總數。

此外，如果 API 直接修改核心內 WLM 資料（例如，建立新的類別），則直到 WLM 管理者未建立的類別出現在指令（如 **wlmstat** 指令）的輸出上後，他們才會瞭解此狀況。為避免在系統管理者更新 WLM 時讓使用此 API 的應用程式混淆而產生衝突，透過 API 建立之未在 WLM 特性檔中定義的類別不會自動從核心內資料移除。它們在透過 `wlm_delete_class` 常式或透過呼叫 **rmclass** 指令（系統管理者直接呼叫或透過 SMIT 呼叫）將其明確移除之前，都維持有效。

WLM API 亦將下列能力提供給應用程式：

- 使用 `wlm_set` 函數查詢或變更 WLM 作業模式
- 查詢 WLM 的現行狀態
- 停止 WLM
- 在主動模式與被動模式之間輪換
- 啟用及停用 **rset** 連結
- 藉由使用 `wlm_load` 常式，來以現行或替代配置更新或啟動 WLM
- 藉由使用 `wlm_assign` 常式，將處理程序或處理程序群組指派給類別。

API 需要與對應 **wlmcntrl** 及 **wlmassign** 指令相同的專用權層次：

- 任何使用者都可查詢 WLM 的狀態
- 僅 root 使用者可變更 WLM 的作業模式
- 僅 root 使用者可更新或重新整理整個配置
- root 或授權的超類別管理者（`adminuser` 或 `admingroup`）可針對給定超類別的子類別更新 WLM
- root、授權使用者（由 `authuser` 或 `authgroup` 所指定）或授權的超類別管理者（`adminuser` 或 `admingroup`）可將處理程序指派給超類別或子類別。

WLM 統計資料 API

WLM API 常式及 `wlm_get_bio_stats` 提供對 **wlmstat** 指令所顯示之 WLM 統計資料的應用程式存取權。

WLM 分類 API

wlm_check 常式可讓使用者驗證給定 WLM 配置的指派規則及類別定義。API 常式 wlm_classify 可讓應用程式判斷具有指定屬性集的處理程序將分類到何類別。

二進位相容性

如果資料結構將來有變更，則為提供二進位相容性，會將版本號碼當作參數傳遞給每一個 API 呼叫。

這樣可讓程式庫判斷應用程式是使用哪個資料結構版本建置的。

工作量管理程式分類、規則及限制的範例

分類處理程序有數個方法，且所有方法可並行操作。

自上而下最精確首次相符演算法用於提供最大配置彈性。針對有某些名稱之程式的特殊狀況，您可按使用者分組以組織處理程序；針對某些使用者的特殊狀況，您可按路徑名稱分組以組織處理程序；或執行任何其他安排。

工作量管理程式指派規則的範例

這個範例顯示配置 *Config* 的頂層 rules 檔案（/etc/wlm/Config/rules 檔案）。

```
* This file contains the rules used by WLM to
* assign a process to a superclass
*
* class resvd user    group   application     type  tag
db1      -      -       -    /usr/bin/oracle*      _DB1
db2      -      -       -    /usr/bin/oracle*      _DB2
devlt    -      -      dev        -            -    -
VPs      -   bob,ted    -         -            -    -
acctg    -      -     acct*       -            -    -
System   -    root      -         -            -    -
Default   -       -        -       -                 -          -
```

下列是 /etc/wlm/Config/devlt/rules 檔案中 devlt 超類別的 rules 檔案範例：

```
* This file contains the rules used by WLM to
* assign a process to a subclass of the
* superclass devlt
*
* class  resvd   user    group  application         type       tag
hackers   -    jim,liz     -       -                -          -
hogs      -       -        -       -            64bit+plock    -
editors   -     !sue       -   /bin/vi,/bin/emacs    -          -
build     -       -        -   /bin/make,/bin/cc     -          -
Default   -       -        -       -                -          -
```

註： 星號 (*) 為 rules 檔案中所使用的註解字元。

下列為使用此規則檔案的範例。下列範例假設所說明的超類別及子類別皆未將繼承屬性設為 yes。 若已啟用繼承，則新的處理程序將從其上代處理程序繼承超類別或子類別。

- 如果來自群組 acct3 的使用者 joe 執行 **/bin/vi**，則會將處理程序置於超類別 acctg 中。
- 如果來自群組 dev 的使用者 sue 執行 **/bin/emacs**，則會將處理程序置於超類別 devlt（群組 ID 相符）中，但不會被分類到 editors 子類別中，因為該使用者被排除在該類別之外。根據預設值，該處理程序將前往 devlt。
- 當資料庫管理者使用 oracle 使用者 ID 及 dbm 群組 ID 啟動 **/usr/sbin/oracle** 來為 DB1 資料庫服務時，會將處理程序分類到 Default 超類別中。僅當處理程序將其標記設為 _DB1 時，才會將它指派給超類別 db1。

含持有率及限制的工作量管理程式類別範例

在此範例中，假設類別 A、B、C 及 D 分別有 3、2、1 及 1 持有率。

若類別 A、C 及 D 在作用中，則計算的目標將是：

target(A) = 3/5 = 60%
target(C) = 1/5 = 20%
target(D) = 1/5 = 20%

若測試期間發現，當允許類別 A 中的應用程式使用 50% 的資源時它們可充分執行，則應當讓其他類別使用另外 50% 的資源。此可藉由為類別 A 提供軟性最大值（此資源的 50%）來完成。 因為 60% 的現行計算目標超過此限制，所以將它下調至軟性最大值。此狀況發生時，會從可用的資源量中扣除類別 A 的目標或實際用量（取其中較低的那個）。因為此類別目前有一個目標是由其限制（不是其持有率）所限制，所以還要從作用中持有率數目中扣除類別的持有率。假設類別 A 的現行用量是 48%，則現在目標將是：

target(A) = 3/5 = 60%, softmax = 50, = 50%
target(C) = 1/2 * (100 - 48) = 26%
target(D) = 1/2 * (100 - 48) = 26%

一段時間之後，所有類別皆可能變成作用中，目標將自動重新調整：

target(A) = 3/7 = 42%
target(B) = 2/7 = 28%
target(C) = 1/7 = 14%
target(D) = 1/7 = 14%

含 CPU 限制的工作量管理程式範例
這個範例會檢查 CPU 配置，假設每一個類別會耗用提供給它的所有 CPU。

兩個類別 A 及 B 位於同一層級。A 的 CPU 限制是 [30% - 100%]。B 的 CPU 限制是 [20% - 100%]。當兩個類別都在執行且都使用足夠的 CPU 時，WLM 會先確定它們是否都達到每秒最小的 CPU 百分比（數秒的平均值）。然後 WLM 會根據任何 CPU 目標持有率值分送剩餘的 CPU 循環。

如果 A 及 B 的 CPU 目標持有率分別是 60% 及 40%，則 A 及 B 的 CPU 使用率會分別穩定在 60% 及 40%。

新增第三個類別 C。此類別是一組與 CPU 頻繁的工作，且應以大約一半（或更多）可用 CPU 執行。類別 C 的限制為 [20% - 100%]，CPU 目標持有率為 100%。如果 C 與 A 及 B 位於同一層級，則當 C 啟動時，A 及 B 的 CPU 配置將急遽減少，而三個類別會分別穩定在 30%、20% 及 50%。在此狀況下，它們的目標亦是 A 與 B 的最小值。

如果可能有更高優先順序的其他工作亦在執行，則系統管理者可能不想讓批次作業使用 50% 的 CPU。在如同先前範例的狀況下，會將 C 置於較低的優先順序層級。在 A 及 B 接收它們所需要的 CPU 之後，C 會接收剩餘的全部 CPU。在上述範例中，C 不接收 CPU 時間，因為 A 與 B 每一個都可接收 100% 的 CPU。不過在大部分狀況下，處於高優先順序層級的 A 及 B 是由互動式或交易導向的工作所組成的，它們不會一直使用所有 CPU。則 C 會接收部分持有率的 CPU，因此與同一或較低層級中的其他類別競爭。

含記憶體限制的工作量管理程式範例
這個範例會檢查對具有各種記憶體目標之處理程序群組的記憶體配置。

三組處理程序必須執行：需要在每次使用時執行的一組互動式處理程序 (PEOPLE)、一律在背景執行的批次作業 (BATCH1)，以及每晚執行之一個更重要的批次作業 (BATCH0)。

PEOPLE 的指定記憶體下限為 20%、記憶體目標為 50 持有率、類別層級值為 1。20% 最小限制會確保當使用者觸控他們的鍵盤時，此類別中的桌面應用程式可相當快速地進行回復。

BATCH1 的記憶體下限為 50%、記憶體目標為 50 持有率、類別層級值為 3。

BATCH0 的記憶體下限為 80%、記憶體目標為 50 持有率、類別層級值為 2。

類別 PEOPLE 及 BATCH1 的總記憶體下限為 70。在正常作業下（BATCH0 未處於執行中），允許這兩個類別取得其所有保留的記憶體。即使它們處於不同的層級中，它們還是會共用機器中剩下的記憶體（大約各用一半）。當在午夜啟動 BATCH0 時，最小記憶體總計達到 150。WLM 會忽略最低層級的最低基本要求，直到上層中的處理程序結束為止。BATCH0 是從 BATCH1 50% 儲存處而不是從 PEOPLE 20% 儲存處取得記憶體。在執行 BATCH0 之後，會再次接受層級 3 處理程序的記憶體儲存處，且系統會回到它正常的記憶體平衡狀態。

工作量管理程式指令
WLM 提供的指令可讓系統管理者執行各種功能。

這些功能包括：

- 使用 **mkclass**、**chclass** 及 **rmclass** 指令，建立、修改及刪除超類別及子類別。 這些指令更新檔案的類別、持有率及限制。

- 使用 **wlmcntrl** 指令，啟動、停止及更新 WLM。
- 使用 **wlmcheck** 指令檢查 WLM 特性檔的給定配置，並判定將具有給定一組屬性的處理程序指派給哪個類別（超類別及子類別）。
- 使用 **wlmstat** (ASCII) 指令，監視每一類別資源使用率。 大部分效能工具（如由 **svmon** 及 **topas** 指令啟動的那些效能工具）具有擴充，可使用新的指令行選項來考量 WLM 類別並提供每一類別及每一層級統計資料。
- **ps** 指令中的旗標可讓使用者顯示處理程序及其應用程式標記位於哪個類別中。 **ps** 指令還可讓使用者清單屬於給定超類別或子類別的全部處理程序。
- 沒有管理指派規則的指令行介面。您必須使用 SMIT 管理工具或文字編輯器。

裝置節點

裝置被組織成若干叢集，稱作節點。每個節點是裝置的一個邏輯子系統，其中，下層裝置以上下代關係與上層裝置相依。

例如，系統節點是所有節點中最高的節點，它由系統中所有的實體裝置所組成。系統裝置位於節點的頂端，其下是匯流排及配接卡，它們與上層的系統裝置相依。在該階層的底端是未連接其他任何裝置的裝置。這些裝置與階層中位於其上的所有裝置相依。

啟動時，會使用上下代相依關係來配置組成節點的所有裝置。配置是自頂端節點開始向下執行，在配置完上層裝置之後，才會配置與它相依的裝置。

AIX 作業系統支援多路徑 I/O (MPIO) 功能。若裝置有 MPIO 功能的裝置驅動程式，則其在此階層中可能有多個母項，這允許在該裝置與給定機器或機器中的邏輯分割區間有多個同時通訊路徑。

裝置類別

對裝置進行管理時，作業系統需要瞭解允許哪些裝置的連接。作業系統將裝置依階層分成三個群組：

這些群組為：

- 功能類別
- 功能子類別
- 裝置類型

功能類別包含執行相同功能的裝置。例如，印表機可以組成一個功能類別。根據某些特定的裝置相似性，可將功能類別分組成子類別。例如，印表機有序列或平行介面。序列式印表機可組成一個子類別，而並列式印表機可組成另一個子類別。可根據裝置的機型及製造商將裝置類型分類。

裝置類別為作業系統定義有效的上下代連接。階層為每個可能的子項連接位置定義可連接的可能子類別。例如，術語 RS-232 8 埠配接卡指定，只有屬於 RS-232 子類別的裝置才能連接到配接卡 8 個埠中的任一個。

裝置類別及其階層性相依關係在「物件資料管理程式 (ODM) 裝置配置」資料庫中加以維護。

裝置配置資料庫及裝置管理

裝置資訊包含在組成裝置配置資料庫的預先定義資料庫或自訂的資料庫中。

預先定義的資料庫包含所有可能受系統支援之裝置的配置資料。階層性裝置類別資訊包含在此資料庫中。自訂的資料庫包含系統中所有目前已定義及配置之裝置的配置資料。目前連接到系統的每個裝置均會保留一筆記錄。

「配置管理程式」是一種可在系統啟動及執行時期內於系統上自動配置裝置的程式。「配置管理程式」在此處理程序期間使用預先定義及自訂資料庫中的資訊，並在以後更新自訂的資料庫。

多路徑 I/O (MPIO) 功能會使用唯一裝置 ID (UDID) 來識別每一個具有 MPIO 功能的裝置，而不管探索到該裝置的路徑為何。UDID 儲存於裝置配置資料庫。 當發現裝置時，會檢查資料庫中的 UDID 來判斷裝置是否是新的，或發現的是否是現有裝置的另一個路徑。當偵測到裝置的多個路徑時，裝置驅動程式或「路徑控制管理程式」核心擴充會判斷將哪條路徑用於特殊要求。

您可以使用 SMIT 或作業系統指令，來執行諸如刪除、新增或配置裝置等裝置管理作業。

裝置狀態

連接到系統的裝置可處於下列四種狀態中的一種。

連接到系統的裝置可處於下列其中一種狀態：

項目	說明
未定義	系統無法識別該裝置。
已定義	自訂的資料庫中記錄了有關該裝置的特定資訊，但此項資訊無法供系統使用。
可用	定義的裝置已與作業系統聯結，或已配置定義的裝置。
已停止	裝置不可用，但其裝置驅動程式仍可識別它。

如果一個 tty 裝置及一台印表機交替使用同一 tty 連接器，則在裝置配置資料庫中會在同一母項及埠上定義這兩個裝置。一次只能配置其中一個裝置。配置 tty 連接器時，印表機特定的設定資訊會保留下來，直到重新配置它為止。該裝置不會被移除；它處於已定義狀態。在目前未使用的裝置首次成為可用之前或被暫時從系統中移除時，使裝置保持已定義狀態，就能夠保留該裝置的自訂資訊。

如果某裝置有裝置驅動程式，可透過裝置驅動程式使該裝置成為可用。

部分裝置（特別是 TCP/IP 虛擬裝置）需要處於已停止狀態。

裝置位置碼

位置碼是從 CPU 匣或主機開始，經由配接卡、信號電纜及非同步分送盒（如果有的話），最後到裝置或工作站的路徑。此碼是另一種識別實體裝置的方法。

位置碼至多由四個欄位的資訊組成，視裝置類型而定。這些欄位代表匣、介面槽、連接器及埠。這些欄位中的每一個都由兩個字元組成。

匣的位置碼僅由匣欄位組成，且僅是兩個字元的字碼。配接卡的位置碼由其位置匣及介面槽欄位組成，格式為 *AA-BB*，其中的 *AA* 對應於匣位置，*BB* 表示包含配接卡的匯流排及介面槽。其他裝置的位置碼格式為 *AA-BB-CC* 或 *AA-BB-CC-DD*，其中的 *AA-BB* 是連接裝置之配接卡的位置碼，*CC* 對應於連接裝置之配接卡上的連接器，而 *DD* 則對應於埠號或 SCSI 裝置位址。

配接卡位置碼

配接卡的位置碼由兩對數字組成，格式為 *AA-BB*，其中的 *AA* 識別包含配接卡的匣位置碼，*BB* 則識別包含該卡的 I/O 匯流排及介面槽。

AA 欄位的值 **00** 表示配接卡位於 CPU 匣或主機中，視系統類型而定。**AA** 欄位的任何其他值都指出該卡位於 I/O 擴充匣中。在此狀況下，*AA* 值可識別包含非同步擴充配接卡之 CPU 匣中的 I/O 匯流排及介面槽號碼。第一個數字可識別 I/O 匯流排，0 與標準 I/O 匯流排對應，1 與可選用的 I/O 匯流排對應。第二個數字可識別所指出 I/O 匯流排上的介面槽號碼。

BB 欄位的第一個數字可識別包含配接卡的 I/O 主機板。如果該卡位於 CPU 匣或主機中，則此數字是 0（表示標準 I/O 匯流排）及 1（表示可選用的 I/O 匯流排）。如果該卡位於 I/O 擴充匣中，則此數字是 0。第二個數字可識別包含該卡之所指出 I/O 匯流排上的介面槽號碼（或 I/O 擴充匣中的介面槽號碼）。

位置碼 00-00 可用來識別標準 I/O 主機板。

範例：

項目	說明
00-05	識別標準 I/O 主機板之介面槽 5 中的配接卡，該卡位於 CPU 匣或主機中（視系統類型而定）。
00-12	識別可選用 I/O 匯流排之介面槽 2 中的配接卡，該卡位於 CPU 匣中。
18-05	識別位於 I/O 擴充匣之介面槽 5 中的配接卡。該匣會連接至非同步擴充配接卡，此配接卡位於 CPU 匣中選用 I/O 匯流排的介面槽 8 中。

印表機及繪圖機位置碼

位置碼 00-00-S1-00 或 00-00-S2-00 指出印表機、繪圖機或 tty 裝置與標準 I/O 主機板序列埠 s1 或 s2 連接。位置碼 00-00-0P-00 指出並列埠印表機與標準 I/O 主機板並列埠連接。

任何其他位置碼都指出印表機、繪圖機或 tty 裝置與配接卡連接，而不是與標準 I/O 主機板連接。這些印表機、繪圖機及 tty 裝置的位置碼格式為 *AA-BB-CC-DD*，其中的 *AA-BB* 指出控制配接卡的位置碼。

項目	說明
AA	AA 欄位值若為 **00**，表示配接卡位於 CPU 匣或主機中，視系統類型而定。**AA** 欄位的任何其他值都指出該卡位於 I/O 擴充匣中；而於此情況下， 第一個及第二個數字分別識別 CPU 匣中包含連接 I/O 擴充匣之非同步擴充配接卡的 I/O 匯流排及匯流排上的介面槽號碼。
BB	**BB** 欄位的第一個數字可識別包含配接卡的 I/O 匯流排。如果該卡位於 CPU 匣或主機中，則此數字是 0（表示標準 I/O 匯流排）及 1（表示可選用的 I/O 匯流排）。如果該卡位於 I/O 擴充匣中，則此數字是 0。第二個數字可識別包含該卡之 I/O 匯流排上的介面槽號碼（或 I/O 擴充匣中的介面槽號碼）。
CC	**CC** 欄位可識別連接非同步分送盒之配接卡上的連接器。可能的值有 01、02、03 及 04。
DD	**DD** 欄位可識別連接印表機、繪圖機或 tty 裝置之非同步分送盒上的埠號。

tty 位置碼

位置碼 00-00-S1-00 或 00-00-S2-00 指出 tty 裝置與標準 I/O 序列埠 s1 或 s2 連接。

任何其他位置碼都指出 tty 裝置與配接卡連接，而不是與標準 I/O 主機板連接。這些裝置的位置碼格式為 *AA-BB-CC-DD*，其中的 *AA-BB* 指出控制配接卡的位置碼。

項目	說明
AA	AA 欄位值若為 **00**，表示配接卡位於 CPU 匣或主機中，視系統類型而定。**AA** 欄位的任何其他值都指出該卡位於 I/O 擴充匣中。在此狀況下，第一個及第二個數字分別識別 CPU 匣中包含連接 I/O 擴充匣之非同步擴充配接卡的 I/O 匯流排及匯流排上的介面槽號碼。
BB	**BB** 欄位的第一個數字可識別包含配接卡的 I/O 匯流排。如果該卡位於 CPU 匣或主機中，則此數字將是 0（表示標準 I/O 匯流排）及 1（表示可選用的 I/O 匯流排）。如果該卡位於 I/O 擴充匣中，則此數字是 0。第二個數字可識別包含該卡之 I/O 匯流排上的介面槽號碼（或 I/O 擴充匣中的介面槽號碼）。
CC	**CC** 欄位可識別連接非同步分送盒之配接卡上的連接器。可能的值有 01、02、03 及 04。
DD	**DD** 欄位可識別連接 tty 裝置之非同步分送盒上的埠號。

SCSI 裝置位置碼

下列是 SCSI 裝置的位置碼。

這些位置碼適用於所有的 SCSI 裝置，包括：

- CD-ROM
- 磁碟
- 起始器裝置
- 可讀寫光碟裝置
- 磁帶
- 目標模式

位置碼格式為 AA-BB-CC-S,L。**AA-BB** 欄位識別控制 SCSI 裝置之 SCSI 配接卡的位置碼。

項目	說明
AA	AA 欄位值若為 **00**，表示控制配接卡位於 CPU 匣或主機中，視系統類型而定。

項目	說明
BB	**BB** 欄位可識別包含配接卡的 I/O 匯流排及介面槽。第一個數字可識別 I/O 匯流排。若為 0，則表示標準 I/O 匯流排；若為 1，則表示可選用的 I/O 匯流排。第二個數字是包含配接卡之所指出 I/O 匯流排上的介面槽。**BB** 欄位值若為 00，表示是標準 SCSI 控制器。
CC	**CC** 欄位可識別連接裝置之配接卡的 SCSI 匯流排。對於僅提供單一 SCSI 匯流排的配接卡，此欄位設為 00。否則，值 00 指出連接到卡內部 SCSI 匯流排的裝置，值 01 指出連接到卡外部 SCSI 匯流排的裝置。
S,L	**S,L** 欄位可識別 SCSI 裝置的 SCSI ID 及邏輯單元號碼 (LUN)。S 值指出 SCSI ID，L 值指出 LUN。

Dials/LPFKeys 位置碼

對於連接至圖形輸入配接卡的 Dials/LPFKeys 裝置，其位置碼格式為 *AA-BB-CC*。

個別欄位的解譯如下：

項目	說明
AA	AA 欄位值若為 **00**，表示控制配接卡位於 CPU 匣或主機中，視系統類型而定。
BB	**BB** 欄位可識別包含配接卡的 I/O 匯流排及介面槽。第一個數字可識別 I/O 匯流排。若為 0，則表示標準 I/O 匯流排；若為 1，則表示可選用的 I/O 匯流排。第二個數字是包含配接卡之所指出 I/O 匯流排上的介面槽。
CC	**CC** 欄位指出連接裝置的配接卡連接器。該值是 01 或 02（視連接的裝置是卡上的埠 1 還是埠 2 而定）。

註： 序列連接的 Dials/LPFKeys 裝置不會指出位置碼。這是因為這些裝置被認為與 tty 裝置連接。tty 裝置由使用者於 Dials/LPFKeys 定義期間指定。

多重通訊協定埠位置碼

多重通訊協定埠的位置碼格式為 *AA-BB-CC-DD*，其中的 *AA-BB* 指出多重通訊協定配接卡的位置碼。

個別欄位的解譯如下：

項目	說明
AA	AA 欄位值若為 **00**，表示多重通訊協定配接卡位於 CPU 匣或主機中，視系統類型而定。
BB	**BB** 欄位可識別包含配接卡的 I/O 匯流排及介面槽。第一個數字可識別 I/O 匯流排。若為 0，則表示標準 I/O 匯流排；若為 1，則表示可選用的 I/O 匯流排。第二個數字是包含配接卡之所指出 I/O 匯流排上的介面槽。
CC	**CC** 欄位可識別連接多重通訊協定分送盒之配接卡上的連接器。該值一律為 01。
DD	**DD** 欄位可識別多重通訊協定分送盒上的實體埠號。可能的值有 00、01、 02 及 03。

設定 iSCSI 卸載配接卡

設定 iSCSI 卸載配接卡包括配置配接卡，以及新增或更新目標。這些指示只適用於 iSCSI 卸載配接卡。如需配置 iSCSI 軟體起始程式的相關資訊，請參閱 iSCSI 軟體起始程式及軟體目標。

配置 AIX 中的 iSCSI 配接卡

iSCSI 配接器配置是簡單且直接明確的作業。

1. 請在 AIX 指令提示中，輸入 **smit iscsi**。
   即會顯示 iSCSI 畫面。

2. 從 iSCSI 畫面選取 **iSCSI 配接卡**。
   即會顯示「iSCSI 配接卡」畫面。
3. 從「iSCSI 配接卡」畫面選取**變更/顯示 iSCSI 配接卡的性質**。
   即會顯示「變更/顯示 iSCSI 配接卡的性質」畫面。
4. 從清單中選取要配置的 iSCSI 配接卡。
   即會顯示配置畫面，類似下列範例。

```
                                                    [輸入欄位]

iSCSI 配接卡                                       ics0

說明                                                iSCSI 配接卡

狀態                                                可用的

位置                                                10-60

要放入配接卡佇列的指令數上限                      [200]      +                          +#


轉送大小上限                                       [0x100000]     +

探索檔名稱                                          [/etc/iscsi/targetshw]
探索原則                                            [file]
自動探索密碼檔名稱                                 [/etc/iscsi/autosecret]
配接卡 IP 位址                                      [10.1.4.187]
配接卡子網路遮罩                                   [255.255.255.0]

配接卡閘道位址                                      [10.1.4.1]

變更僅應用至資料庫                                 no         +
```

註：如果您對特定欄位的用途有疑問，請將游標置於欄位中，再按 **F1** 即可取得說明。

若要使用純文字檔探索，請在**探索原則**欄位中輸入 `file`。若要使用 ODM 探索，請在**探索原則**欄位中輸入 `odm`。若為 DHCP 探索的 iSCSI 目標，請在**探索原則**欄位中輸入 `slp`。

更新 iSCSI 目標的純文字檔

純文字檔是用來配置 iSCSI 目標的靜態配置檔。其預設檔名為 `/etc/iscsi/targetshw`。

您必須在純文字檔中，明確地指定所有相關 iSCSI 目標探索特性。

相關資訊

targets 檔

將利用統計方式探索到的 iSCSI 目標新增至 ODM

對 iSCSI 配接卡或 iSCSI 軟體起始程式使用 ODM 探索原則時，iSCSI 驅動程式從 ODM 取得 iSCSI 目標說明。

您可以使用 AIX 指令或 SMIT，來操作 ODM 中的 iSCSI 目標資訊。**chiscsi**、**lsiscsi**、**mkiscsi** 及 **rmiscsi** 指令會變更、顯示、新增及移除 ODM 中的 iSCSI 目標資訊。

若要使用 SMIT 將一個利用統計方式探索到的 iSCSI 目標新增至 ODM，請執行下列動作：

1. 請在 AIX 指令提示中，輸入 **smit iscsi**。
   即會顯示 iSCSI 畫面。
2. 從 iSCS 畫面選取 **ODM 中的 iSCSI 目標裝置參數**。
   即會顯示「ODM 中的 iSCSI 目標裝置參數」畫面。
3. 從 iSCS 畫面選取**在 ODM 中新增 iSCSI 目標裝置**。
   即會顯示「在 ODM 中新增 iSCSI 目標裝置」畫面。
4. 從清單中選取要配置的 iSCSI 配接裝置。清單可能包括 iSCSI 卸載配接卡（開頭為 "ics" 的裝置）及 iSCSI 軟體起始程式（開頭為 "iscsi" 的裝置）。
5. 從清單中選取要配置的 iSCSI 配接裝置。清單可能包括 iSCSI 卸載配接卡（開頭為 "ics" 的裝置）及 iSCSI 軟體起始程式（開頭為 "iscsi" 的裝置）。
6. 請在欄位中輸入適當的資訊。

以下是範例。

```
[輸入欄位]

iSCSI 配接卡                                          ics0
iSCSI 目標名稱                                        [iqn.com.ibm.tgt:0]
iSCSI 目標的 IP 位址                                  [10.1.4.25]
iSCSI 目標的埠號                                      [3260]
密碼                                                   [我的密碼]
使用者名稱                                            [使用者名稱]
```

註：如果您對特定欄位的用途有疑問，請將游標置於欄位中，再按 **F1** 即可取得說明。

將利用統計方式探索到的 iSCSI 目標從純文字檔新增至 ODM

您可以使用 SMIT，將純文字檔的資訊匯入 ODM。

1. 請在 AIX 指令提示中，輸入 **smit iscsi**。
   即會顯示 iSCSI 畫面。
2. 從 iSCS 畫面選取 **ODM 中的 iSCSI 目標裝置參數**。
   即會顯示「ODM 中的 iSCSI 目標裝置參數」畫面。
3. 從 iSCS 畫面選取**在 ODM 中新增 iSCSI 目標裝置**。
   即會顯示「在 ODM 中新增 iSCSI 目標裝置」畫面。
4. 從 iSCS 畫面選取**在 ODM 中新增來自檔案的 iSCSI 目標裝置**。
   即會顯示「在 ODM 中新增來自檔案的 iSCSI 目標裝置」畫面。
5. 從清單中選取要配置的 iSCSI 配接卡。
   即會針對您選取的 iSCSI 配接卡，顯示「在 ODM 中新增來自檔案的 iSCSI 目標裝置」畫面。
6. 請在欄位中輸入適當的資訊。
   以下是範例。

```
                                                    [輸入欄位]

iSCSI 通訊協定裝置                                  iscsi3

iSCSI 群組                                           [靜態]       +

iSCSI 目標的檔名                                    [/etc/iscsi/targetshw]       +
```

註：如果您對特定欄位的用途有疑問，請將游標置於欄位中，再按 **F1** 即可取得說明。

PCI 熱插拔管理

可於作業系統處於執行中時，將新的 PCI 熱插拔配接卡插入可用的 PCI 介面槽。

新 PCI 配接卡類型可與目前所安裝之 PCI 配接卡相同，也可以不同。無需重新啟動作業系統，作業系統及應用程式即可使用新資源。 新增熱插拔配接卡的部分理由可能包括：

- 新增額外的功能或容量到現有硬體及韌體。
- 從不再需要 PCI 配接卡所提供之功能的系統移轉 PCI 配接卡。
- 安裝新系統，在起始配置可選用硬體子系統（包括 PCI 配接卡），以及安裝及啟動作業系統後，配接卡即成為可用。

註：若您使用 PCI 熱插拔的更換或新增作業，或是使用「動態邏輯分割」來新增配接卡，則在使用 **bootlist** 指令將該配接卡及其子項裝置指定為開機裝置時，可能無法使用它們。您可能必須重新啟動機器，才能使作業系統識別所有潛在的開機裝置。已列在開機清單中的配接卡仍然是有效的開機裝置，它會在稍後被確定類型的配接卡所取代。

無需關機或關閉系統電源，亦可移除毀損或故障的 PCI 熱插拔配接卡或用同類型的另一張配接卡來交換。 當您交換配接卡時，現有的裝置驅動程式會支援該配接卡，因為其類型相同。 會為置換裝置保留有關該配接卡之子項裝置的裝置配置及配置資訊。更換某個配接卡的部分理由可能包括：

- 暫時更換該卡，以利於判斷問題或隔離故障的 FRU。
- 用功能良好的卡替換有缺陷、有故障或間歇地發生故障的配接卡。

· 更換 HACMP 或多路徑 I/O 配置中發生故障的多餘配接卡。

移除熱插拔配接卡時，作業系統及應用程式會變得無法使用該配接卡提供的資源。移除配接卡的部分理由可能包括：

· 移除現有 I/O 子系統。

· 當不再需要某配接卡，或配接卡有故障而沒有替換的卡時，移除配接卡。

· 當配接卡所在之系統不再需要其功能時，將配接卡移轉到另一個系統上。

必須先取消配置熱插拔裝置，才能將它移除或更換。相關裝置驅動程式必須釋放它為該裝置配置的所有系統資源。這包括取消定位及釋放記憶體、取消定義岔斷及 EPOW 處理程式、釋放 DMA 及計時器資源，以及任何其他必要步驟。驅動程式亦必須確保裝置上的岔斷、匯流排記憶體及匯流排 I/O 已停用。

於移除配接卡之前後，系統管理者必須執行下列作業：

· 終止並還原使用裝置的應用程式、常駐程式或處理程序。

· 卸載並重新裝載檔案系統。

· 移除並重建裝置定義，以及執行其他必要的作業以釋放使用中的裝置。

· 將要服務的系統置於安全狀態。

· 取得並安裝所有必要的裝置驅動程式。

除非已取消配置連接到所識別介面槽的裝置，且該裝置處於已定義狀態，否則，移除及更換作業會失敗。可使用 **rmdev** 指令來這樣做。將配接卡置於已定義狀態之前，請關閉所有正在使用配接卡的應用程式，否則，該指令將無法順利執行。如需 **rmdev** 指令的相關資訊，請參閱 **rmdev**。

於部分情況下，您亦可執行下列作業：

· 準備要插入、移除或更換的 PCI 熱插拔配接卡。

· 識別熱插拔作業所涉及的介面槽或 PCI 配接卡。

· 移除或插入 PCI 熱插拔配接卡。

註： 熱插拔作業期間，「物件資料管理程式 (ODM)」仍是已鎖定。因此，其他需要 ODM 的作業可能懸置或失敗。在叢集中透過其他節點起始的整個叢集配置變更可能懸置或失敗。因此，在完成熱插拔作業之前，確定不會執行這類作業。

小心： 儘管 PCI 熱插拔管理提供了無需關閉系統電源或重新啟動作業系統，即可新增、 移除及更換 PCI 配接卡的能力，但並非熱插拔槽中的所有裝置均可以這種方式來管理。例如，如果不關閉系統電源，就無法移除或更換組成 rootvg 磁區群組的硬碟或其連接的 I/O 控制器，因為它是執行作業系統所必需的。如果已對 rootvg 磁區群組執行鏡映，您可以使用 **chpv** 指令讓磁碟離線。如果 rootvg 磁區群組所在的一或多個磁碟上已啟用多路徑 I/O (MPIO)，且已連接至多重 I/O 控制器，則可移除（或置換）其中之一的 I/O 控制器，而不需要重新啟動系統。在此狀況下，應該使用配接卡上的 **rmdev -R** 指令來取消配置要從此 I/O 控制器中移除（或置換）的所有路徑。這樣將會取消配置路徑及配接卡。然後您可以使用「熱插拔管理」繼續作業。嘗試移除或插入 PCI 熱插拔配接卡之前，請參閱 *PCI* 配接卡放置參照（隨附於支援熱插拔的主機），以判斷您的配接卡是否可熱交換。請參閱主機文件，以取得安裝或移除配接卡的指示。

顯示 PCI 熱插拔槽資訊

新增、移除或取代熱插拔配接卡之前，您可以顯示機器中 PCI 熱插拔槽的相關資訊：

您可以顯示下列資訊：

· 機器中所有 PCI 熱插拔槽的清單

· 介面槽是否可用或是空的

· 目前使用中的介面槽

· 特定介面槽的性質，如槽名、 說明、連接器類型與連接的裝置名稱。

您可以使用 SMIT 或系統指令。若要執行這些作業，您必須以 root 使用者的身分登入。

SMIT 捷徑程序

1. 在系統提示中鍵入 `smit devdrpci`，然後按 Enter 鍵。
2. 使用 SMIT 對話框來完成作業。

若要取得完成作業的其他資訊，您可以在 SMIT 對話框中選取「F1 輔助說明鍵」。

指令程序

您可以使用下列指令來顯示熱插拔槽與連接裝置的相關資訊：

- lsslot 指令顯示所有 PCI 熱插拔槽及其性質的清單。
- **lsdev** 指令顯示安裝在系統中的所有裝置之現行狀態。

取消配置 PCI 通訊配接卡

下列是取消配置 PCI 通訊配接卡的處理程序概觀。這包括乙太網路、記號環、FDDI 及 ATM 配接卡。

如果應用程式使用的是 TCP/IP 通訊協定，則必須先從網路介面清單移除該配接卡的 TCP/IP 介面，才能將配接卡置於已定義狀態。使用 **netstat** 指令，來判斷是否針對 TCP/IP 配置了配接卡，以及檢查配接卡上的作用中網路介面。如需 **netstat** 指令的相關資訊，請參閱 **netstat**。

乙太網路配接卡可有兩個介面：「標準乙太網路」(en*X*) 或 IEEE 802.3 (et*X*)。*X* 與 ent*X* 配接卡名稱中的數字相同。一次只有這些介面中的一個能夠使用 TCP/IP。例如，乙太網路配接卡 ent0 可有 en0 及 et0 介面。

記號環配接卡只能有一個介面：記號環 (tr*X*)。*X* 與 tok*X* 配接卡名稱中的數字相同。例如，記號環配接卡 tok0 有一個 tr0 介面。

ATM 配接卡只能有一個 atm 介面：ATM (at*X*)。*X* 與 atm*X* 配接卡名稱中的數字相同。例如，ATM 配接卡 atm0 有一個 at0 介面。不過，ATM 配接卡允許在單一配接卡上執行多個模擬用戶端。

ifconfig 指令可從網路移除介面。**rmdev** 指令可取消配置 PCI 裝置，但在「自訂的裝置物件類別」中保留其裝置定義。一旦配接卡處於已定義狀態，即可使用 **drslot** 指令移除之。

如果要取消配置 PCI 匯流排 pci1 的子項，以及其下其他所有的裝置，並將其裝置定義保留在自訂的裝置物件類別中，請鍵入：

```
rmdev  -p pci1
```

系統會顯示一則類似下面所示的訊息：

```
rmt0 已經定義
hdisk1 已經定義
scsi1 已經定義
ent0 已經定義
```

移除或取代 PCI 熱插拔配接卡

您可以在主機中移除或取代 PCI 熱插拔配接卡，而不用關閉作業系統或關閉系統電源。移除配接卡會使作業系統與應用程式無法使用該配接卡所提供的資源。

您必須先取消配置配接卡，然後才能移除它。

下列是移除 PCI 熱插拔配接卡的程序。您可以使用 SMIT 或系統指令來完成這些作業。若要執行這些作業，您必須以 root 使用者的身分登入。

以另一個同類型的配接卡取代配接卡，可保留被取代的配接卡之配置資訊，並將該資訊與取代的卡相比較。被取代的配接卡之現有裝置驅動程式必須能夠支援置換配接卡。

SMIT 捷徑程序

1. 在系統提示中鍵入 `smit devdrpci`，然後按 Enter 鍵。
2. 使用 SMIT 對話框來完成作業。

若要取得完成作業的其他資訊，您可以在 SMIT 對話框中選取「F1 輔助說明鍵」。

指令程序

您可以使用下列指令以顯示熱插拔槽與連接裝置的相關資訊，並移除 PCI 熱插拔配接卡：

- **lsslot** 指令會顯示所有 PCI 熱插拔槽及其性質的清單。如需使用此指令的相關資訊，請參閱 *Commands Reference, Volume 3* 中的 **lsslot**。
- **lsdev** 指令顯示安裝在系統中的所有裝置之現行狀態。如需使用此指令的相關資訊，請參閱 *Commands Reference, Volume 3* 中的 **lsdev**。
- **drslot** 指令會準備熱插拔槽，以便移除熱插拔配接卡。如需使用此指令的相關資訊，請參閱 *Commands Reference, Volume 2* 中的 **drslot**。

新增 PCI 熱插拔配接卡

您可以在主機的可用介面槽中，新增 PCI 熱插拔配接卡，使作業系統與應用程式可以使用新的資源，而無需重新啟動作業系統。配接卡可以是目前安裝的另一種配接卡類型，或是不同的配接卡類型。

以下是新增 PCI 熱插拔配接卡的程序。

小心： 在您嘗試新增 PCI 熱插拔配接卡之前，請參照支援熱插拔的主機所隨附的 *PCI Adapter Placement Reference*，以判斷您的配接卡是否可以熱插拔。請參閱主機文件，以取得安裝或移除配接卡的指示。

新增 PCI 熱插拔配接卡含有下列作業：

- 尋找並識別機器中的可用介面槽
- 準備介面槽以配置配接卡
- 必要時，安裝裝置驅動程式
- 配置新配接卡

您可以使用 SMIT 或系統指令。若要執行這些作業，您必須以 root 使用者的身分登入。

註： 當您將熱插拔配接卡新增至系統時，如果您使用 **bootlist** 指令，將該配接卡及其子項裝置指定為開機裝置，它們可能無法使用。您可能需要重新啟動系統，使作業系統知道所有可能的開機裝置。

SMIT 捷徑程序

1. 在系統提示中鍵入 `smit devdrpci`，然後按 Enter 鍵。
2. 使用 SMIT 對話框來完成作業。

若要取得完成作業的其他資訊，您可以在 SMIT 對話框中選取「F1 輔助說明鍵」。

指令程序

您可以使用下列指令來顯示 PCI 熱插拔槽及連接裝置的相關資訊，並新增 PCI 熱插拔配接卡：

- **lsslot** 指令會顯示所有熱插拔槽及其性質的清單。如需使用此指令的相關資訊，請參閱 *Commands Reference, Volume 3* 中的 **lsslot**。
- **lsdev** 指令顯示安裝在系統中的所有裝置之現行狀態。如需使用此指令的相關資訊，請參閱 *Commands Reference, Volume 3* 中的 **lsdev**。
- **drslot** 指令準備新增或移除熱插拔配接卡的熱插拔槽。如需使用此指令的相關資訊，請參閱 *Commands Reference, Volume 2* 中的 **drslot**。

多路徑 I/O

使用多路徑 I/O (MPIO)，您可以透過一或多個實體連線或路徑，專門偵測裝置。

路徑控制模組 (PCM) 可提供路徑管理功能。

具 MPIO 功能的裝置驅動程式，可以控制不只一種類型的目標裝置。PCM 可支援一或多種特定的裝置。因此，一個裝置驅動程式可作為多個 PCM 的介面，透過通往每一個目標裝置的路徑來控制 I/O。

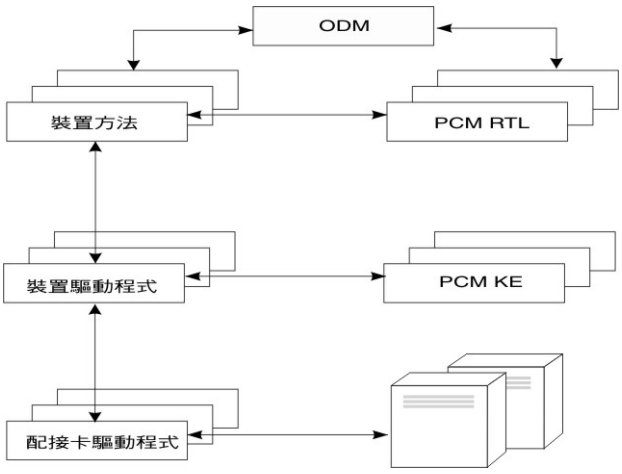


圖 *12. MPIO* 元件互動

您必須先在「物件資料管理程式 (ODM)」中修改裝置的驅動程式、方法和預先定義的屬性，讓它支援多路徑的偵測、配置和管理，該裝置才能利用 MPIO。平行的 SCSI 和「光纖通道」磁碟裝置驅動程式及其裝置方法，都支援 MPIO 磁碟裝置。支援 iSCSI 磁碟裝置作為 MPIO 裝置。「光纖通道」磁帶機驅動程式及其裝置方法支援 MPIO 磁帶機。 同時，ODM 中某些裝置的預先定義屬性，也針對 MPIO 而加以修改。

AIX PCM 是由下列各項所組成：PCM RTL 配置模組和 PCM KE 核心擴充。PCM RTL 是執行時期可載入模組，可讓裝置方法偵測 PCM KE 所需的其他 PCM KE 裝置專用或路徑 ODM 屬性。PCM RTL 是由裝置方法載入。然後存取 PCM RTL 中的一或多個常式，來起始設定或修改 PM KE 變數的特定作業。

PCM KE 提供路徑控制管理功能，給任何支援 MPIO 介面的裝置驅動程式使用。PCM KE 依據裝置配置來偵測路徑，並且將這些資訊傳送給裝置驅動程式。每一個具 MPIO 功能的裝置驅動程式，都會從它的母項（一或多個）將路徑加入裝置當中。透過不同路徑來維護和排程 I/O 的作業是由 PCM KE 提供的，具 MPIO 功能的裝置驅動程式不會察覺它。

PCM KE 可以提供不只一個遞送演算法，供使用者選擇。PCM KE 也可以幫助您收集資訊，用來判斷和選取任何 I/O 要求的最佳路徑。PCM KE 可以根據各種準則來選擇最佳路徑，包括負載平衡、連線速度以及連線失敗等等。

AIX PCM 具有健檢功能，可用來執行下列作業：

- 檢查路徑，並判斷目前有哪些路徑可以用來傳送 I/O
- 啟用之前因暫時路徑錯誤（例如，移除連接裝置的纜線，然後再重新連線）而被標示為失敗的路徑
- 檢查目前不用，但在發生失效接手（例如，當 algorithm 屬性值為 `failover` 時，健檢可以測試其他的替代路徑）之後仍會使用的路徑

不是所有磁碟裝置和磁帶機，都可以使用 AIX 預設 PCM 來偵測和配置。AIX 預設 PCM 包含兩個路徑控制模組，一個用來管理磁碟裝置，另一個用來管理磁帶機。如果沒有偵測到您的裝置，請詢問該裝置的供應商，看看是否有適合您裝置使用的 PCM。

管理具備 MPIO 功能的裝置

「多路徑 I/O (MPIO)」特性可用來定義通往失效接手裝置的另一個路徑。

失效接手 是一種路徑管理演算法，它可以提高裝置的可靠性與可用性，因為系統會自動偵測 I/O 路徑失敗，並透過替代路徑重新遞送 I/O。所有 SCSI SCSD 磁碟機自動配置成 MPIO 裝置，且「光纖通道」磁碟機的選取號碼可配置成「MPIO 其他」磁碟。對其他裝置也予以支援，只要裝置驅動程式與 AIX 中的 MPIO 實作相容即可。

MPIO 已安裝並配置為 BOS 安裝的一部分。雖然不需要進一步的配置，但是您可以使用 SMIT 或指令行介面來新增、移除、重新配置、啟用及停用裝置（或裝置路徑）。下列指令可協助您管理 MPIO 路徑：

mkpath
新增目標裝置的路徑。

rmpath
移除目標裝置的路徑。

chpath
變更目標裝置路徑的屬性或作業狀態。

lsmpio
顯示關於 multiPath I/O (MPIO) 儲存裝置的資訊，例如狀態、配置及統計資料等等。

lspath
顯示目標裝置之路徑的相關資訊。

如果要在具有多個埠（例如：多路徑 I/O (MPIO)）的磁碟上執行 BOS 安裝或 mksysb 安裝，則埠必須在安裝期間保持作用中。

將 SCSI 裝置以纜線安裝為 MPIO 裝置
當 SCSI 裝置被配置為 MPIO 相容的裝置時，最多會有兩個配接卡支援它。

如果您要將平行 SCSI 裝置以纜線安裝為 MPIO 裝置，請參考下面這個簡單配置作為範例。以下是必須完成的最小配置；您的裝置可能還需要額外的配置。

1. 關閉電源，然後安裝兩個 SCSI 配接卡。
2. 將該裝置以纜線安裝到這兩個 SCSI 配接卡。
3. 開啟系統的電源。
4. 將其中一個配接卡的設定，變更為專屬的 SCSI ID。SCSI 配接卡的 SCSI ID 是預設為 7，但因為每個 ID 都必須是專屬的，所以請將其中一個配接卡改成另外的號碼，例如 6。
5. 執行 **cfgmgr** 指令。
6. 如果要驗證配置，請在指令行輸入下述指令：

   ```
   lspath -l hdiskX
   ```

   其中 *X* 是新配置裝置的邏輯號碼。指令輸出應該會顯示兩個路徑及其狀態。

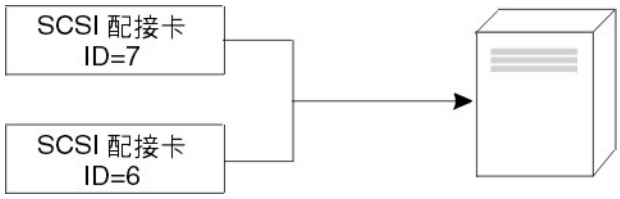


圖 *13. MPIO SCSI* 裝置的纜線配置

這個圖例顯示，兩個 SCSI 配接卡以纜線安裝到同一個裝置。

將光纖通道裝置以纜線安裝為 MPIO 裝置
您可以將「光纖通道」裝置，以纜線安裝到多個配接卡。軟體內沒有限制。

如果您要將「光纖通道」裝置以纜線安裝為 MPIO 裝置，請參考下面這個簡單配置作為範例。以下是必須完成的最小配置；您的裝置可能還需要額外的配置。

1. 關閉電源，然後安裝兩個「光纖通道」配接卡。
2. 將配接卡以纜線安裝到交換器或集線器。
3. 將裝置以纜線安裝到交換器或集線器。
4. 開啟系統電源。
5. 如果要驗證配置，請在指令行輸入下述指令：

   ```
   lspath -l hdiskX
   ```

   其中 *X* 是新配置裝置的邏輯號碼。指令輸出應該會針對您所安裝的每個配接卡，分別顯示一個路徑及其狀態。

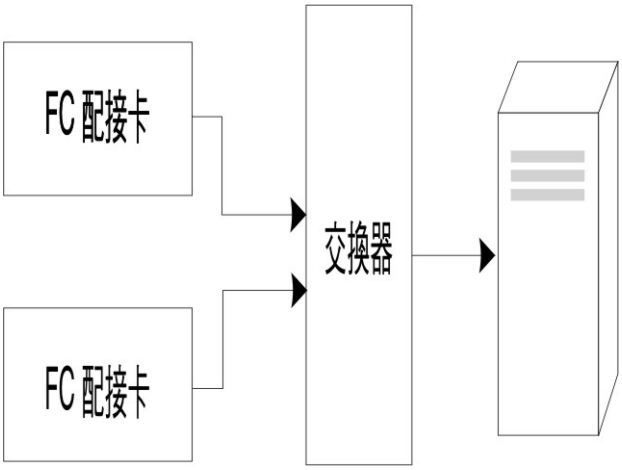


圖 *14. MPIO* 光纖通道裝置的纜線配置

配置 MPIO 裝置

配置具 MPIO 功能的裝置時，所用與非 MPIO 裝置一樣的指令。

cfgmgr、**mkdev**、 **chdev**、 **rmdev** 及 **lsdev** 指令，都支援管理 MPIO 裝置實例，並顯示它們的屬性。MPIO 裝置案例也有與裝置案例相關聯的路徑案例。**mkpath** 、**chpath**、**rmpath** 及 **lspath** 指令負責管理路徑案例，並顯示它們的屬性。

您可以從 MPIO 裝置新增或移除路徑實例，而無需取消配置該裝置。

MPIO 裝置可以有其他屬性，但所有的 MPIO 裝置都必須支援 `reserve_policy` 和 `algorithm` 這兩個必要的屬性。`reserve_policy` 屬性會判斷在裝置開啟時，裝置驅動程式要實作的保留方法類型，而且不管是在同一個系統還是在另一個系統上，都可以使用它來限制其他配接卡的裝置存取權。`algorithm` 屬性會定義 PCM 在透過為裝置而配置的路徑來管理 I/O 時所用的方法。

支援的多路徑裝置

AIX 預設 PCM 支援 `devices.common.IBM.mpio.rte` 檔案集中定義的一組磁碟裝置和磁帶機。

AIX 磁碟 PCM 或磁帶 PCM 不支援的裝置需要裝置供應商提供 ODM 中預先定義的屬性、PCM，以及任何其他將裝置辨識為具備 MPIO 功能之裝置所需的支援碼。

若要判斷 AIX 磁碟 PCM 支援哪些磁碟裝置，請執行下列 Script：

```
          odmget -qDvDr=aixdiskpcmke PdDv | grep uniquetype | while read line
                    do
                              utype=`echo $line | cut -d'"' -f2`
                              dvc=`odmget -q"uniquetype=$utype AND attribute=dvc_support"
PdAt`
                              echo $dvc | grep values | cut -d'"' -f2
          done
```

若要判斷 AIX 磁碟 PCM 支援那些磁帶機，請執行下列 Script：

```
          odmget -qDvDr=aixtapepcmke PdDv | grep uniquetype | while read line
                    do
                              utype=`echo $line | cut -d'"' -f2`
                              dvc=`odmget -q"uniquetype=$utype AND attribute=dvc_support"
PdAt`
                              echo $dvc | grep values | cut -d'"' -f2
          done
```

此 Script 輸出會顯示 AIX 預設 PCM 所支援的專屬裝置類型清單。「AIX 磁碟 PCM」支援的三種裝置類型為自我配置並列式 *SCSI* 磁碟 (`disk/scsi/scsd`) 及 *MPIO* 其他 *FC* 磁碟 (`disk/fcp/mpioosdisk`)，以及 *MPIO* 其他 *iSCI* (`disk/iscsi/mpioosdisk`)。AIX 磁帶 PCM 支援的裝置類型為 *MPIO* 其他 *FC* 磁帶 (`tape/fcp/mpioost`)。

MPIO 其他 *FC* 磁碟及 *MPIO* 其他 *FC* 磁帶裝置分別是其他光纖通道磁碟與其他光纖通道磁帶的子集。只有在供應商未提供 ODM 預先定義屬性，且裝置經過認證可與其中一個 AIX 預設 PCM 搭配運作時，才會支援裝置作為 *MPIO* 其他 *FC* 裝置。認證並不保證當裝置被配置為 *MPIO* 其他 *FC* 裝置時，便支援所有的裝置功能。

MPIO 其他 *iSCSI* 磁碟是其他 iSCSI 磁碟的子集。只有在供應商未提供 ODM 預先定義屬性，且裝置經過認證可以與 AIX PCM 搭配運作時，才會支援裝置作為 *MPIO* 其他 *iSCSI* 磁碟。認證並不保證，裝置被配置為 *MPIO* 其他 *iSCSI* 磁碟時，便支援它的所有功能。

若要判斷以 *MPIO* 其他 *FC* 磁碟方式支援哪些裝置，請執行下例 Script：

```
odmget –quniquetype=disk/fcp/mpioosdisk PdAt | grep deflt | cut -d'"' -f2
```

若要判斷以 *MPIO* 其他 *FC* 磁帶方式支援那些裝置，請執行下列 Script：

```
odmget -q "uniquetype=tape/fcp/mpioosdisk AND attribute=mpio_model_map PdAt | grep deflt | cut -d'"' -f2
```

若要判斷以 *MPIO* 其他 *iSCSI* 磁碟方式支援那些裝置，請執行下列 Script：

```
odmget –quniquetype=disk/iscsi/mpioosdisk PdAt | grep deflt | cut -d'"' -f2
```

此 Script 輸出會顯示含有裝置供應商和機型的查詢資料清單。

若要顯示所有安裝在系統上且具 MPIO 功能的裝置，請執行下列 Script：

```
odmget -qattribute=unique_id PdAt | grep uniquetype | cut -d'"' -f2
```

此 Script 輸出會顯示唯一具備 MPIO 功能的裝置類型清單，AIX 預設 PCM 及供應商供應的 PCM 均支援這些裝置。

MPIO 裝置屬性

只有多路徑裝置才支援下列屬性。可以使用 SMIT 或指令（尤其是 **lsattr** 及 **chdev** 指令）來顯示或變更屬性。

針對屬性的並行更新，已啟用某些「多路徑 I/O (MPIO)」裝置屬性。您可以在磁碟開啟及使用中時更新屬性值，新值會立即生效。 針對某些屬性，尤其是 `reserve_policy` 屬性，對於何時可以變更屬性或屬性可以接受哪些新值可能有所限制。比方說，如果磁碟已開啟且目前是作為叢集儲存庫磁碟使用，則 AIX 作業系統會封鎖任何想要對該磁碟設定保留原則的嘗試，因為它會造成其他叢集節點失去該儲存庫的存取權。

所有的 MPIO 裝置都必須支援 `reserve_policy` 這個必要的裝置屬性。通常，多路徑裝置也具有 `PR_key_value` 裝置屬性。多路徑裝置可以有其他裝置特定屬性。 其他裝置專用的屬性如下：

FC3_REC
指出裝置是否必須開啟使用「光纖通道」類別 3 的錯誤回復。啟用此特性，可改進與「光纖通道」相關的某些光纖錯誤類型之錯誤偵測和錯誤回復效能。 此屬性只能用於一組有限的裝置。此屬性可含有下列值：

true
啟用使用「光纖通道」類別 3 的錯誤回復。

FS_BF= false
停用使用「光纖通道」類別 3 的錯誤回復。

reserve_policy
可定義在裝置開啟時，是否要採用保留方法。其值如下：

no_reserve
不在裝置上應用保留方法。其他發起者可能會存取該裝置，而這些發起者可能位於其他主機系統上。

single_path_reserve
在裝置上應用 SCSI2 保留方法，這表示只有發出保留的發起者才能存取該裝置。這個原則可以防止同一部主機或其他主機上的其他發起者，存取該裝置。這個原則使用 SCSI2 保留，將裝置鎖定給單一發起者（路徑），經由其他路徑所遞送的指令，都會導致保留衝突。

當您選用 single_path_reserve 值時，在多個路徑之間交替使用指令的路徑選擇演算法，可能會產生混亂。舉例來說，假設裝置特定的 PCM 有一個必要屬性，該屬性所設定的值，會透過多個路徑分散 I/O。當 single_path_reserve 生效時，磁碟驅動程式必須發出匯流排裝置重設 (BDR)，然後使用用來傳送下一個指令的新路徑來發出保留，以中斷前一個保留。每次選擇不同的路徑時，便會產生混亂，而且由於傳送 BDR 及發出保留給目標裝置所產生的額外負荷，會降低效能。(AIX PCM 不允許您選取會導致混亂的演算法）。

PR_exclusive
在裝置開啟時，應用 SCSI3 persistent-reserve、exclusive-host 方法。在每一個主機系統上，PR_key_value 屬性值都必須是專屬的。PR_key_value 屬性可用來避免其他主機系統上的發起者存取該裝置。

PR_shared
在裝置開啟時，應用 SCSI3 persistent-reserve、shared-host 方法。在每一個主機系統上，PR_key_value 值都必須是專屬的。其他主機系統上的發起者必須先登錄才能存取該裝置。

PR_key_value
只有當裝置支援任何一個持續性保留原則（PR_exclusive 或 PR_shared）時才需要它。

路徑控制模組屬性

除了預設的 AIX 預設路徑控制模組 (PCM) 之外，裝置供應商可能也會提供裝置特有的 PCM。使用者可以變更的屬性集，是由裝置供應商來定義的。裝置特定的 PCM 可以具有裝置和路徑屬性。

下面是 AIX 預設 PCM 的裝置屬性：

algorithm
判斷透過裝置路徑來分散 I/O 所用的方法。algorithm 屬性具有下列值：

註： 有些裝置只支援這些值的子集。

failover
將所有 I/O 作業傳送至單一路徑。如果該路徑是標示為失敗或已停用，則會選取下一個可用的路徑來傳送所有 I/O 作業。此演算法會根據 path_priority 屬性的遞增值，來維護依序清單中所有已啟用的路徑。會針對每一個 I/O 作業選取具有最低路徑優先順序值的有效路徑。

round_robin
透過多個已啟用的路徑配送 I/O 作業。如果裝置有主動及被動路徑，或有偏好及非偏好的路徑，則對 I/O 作業只使用一個路徑子集。 如果該路徑是標示為失敗或已停用，則不再使用它來傳送 I/O 作業。I/O 作業是根據 path_priority 屬性進行配送。有較高路徑優先順序值的路徑會接收到較大量的 I/O 作業。

shortest_queue
透過多個已啟用的路徑配送 I/O 作業。如果裝置有主動及被動路徑，或有偏好及非偏好的路徑，則對 I/O 作業只使用一個路徑子集。 此演算法類似於 round_robin 演算法。不過，shortest_queue 演算法會根據每一個路徑上擱置的 I/O 作業數來配送 I/O 作業。 會選取目前有最少擱置 I/O 作業數的路徑進行下一個作業。當演算法是設為 shortest_queue 時，會忽略 path_priority 屬性。

hcheck_mode
判斷在使用性能檢查功能時，必須檢查哪些路徑。此屬性支援下列模式：

啟用
透過其狀態為已啟用的路徑來傳送 **healthcheck** 指令。此模式不會透過其狀態為已停用或遺漏的路徑來傳送 **healthcheck** 指令。

failed
透過其狀態為失敗的路徑來傳送 **healthcheck** 指令。此模式不會透過其狀態為已啟用、已停用或遺漏的路徑來傳送 **healthcheck** 指令。

非作用中
（預設值）透過裝置上沒有作用中 I/O 的路徑（包括其狀態為失敗或已啟用的路徑）來傳送 **healthcheck** 指令。 此模式不會透過其狀態為已停用或遺漏的路徑來傳送 **healthcheck** 指令。

hcheck_interval
可定義在裝置路徑上執行性能檢查的頻率。此屬性支援的範圍從 0 到 3600 秒。如果您選取 0 值，就表示停用性能檢查。

註：只有在磁碟已由部分處理程序開啟且尚未關閉時，才會執行性能檢查。如果沒有已開啟磁碟的實體，則即使該裝置的 **hcheck_interval** 屬性已設為非零值，「路徑控制」模組依然不會檢查路徑。

dist_tw_width
定義「時間視窗」的持續期間。在這段時間範圍內，分散式錯誤偵測演算法會累積傳回錯誤的 I/O。dist_tw_width 屬性的測量單位是毫秒。降低此屬性值，會減少每個範例所採用的期間，並降低對小型的突發 I/O 錯誤的演算法靈敏度。增加此屬性值，會增加對小型突發錯誤的演算法靈敏度，並增加路徑失敗的可能性。

dist_err_percent
定義具有錯誤（路徑上因效能不佳而失敗之前所允許發生的錯誤）的「時間視窗」之百分比。dist_err_percent 的範圍是從 0 到 100。當此屬性設為零 (0) 時，會停用分散式錯誤偵測演算法。預設設定為零。分散式錯誤偵測演算法會取樣測試連接裝置與配接卡的光纖是否有錯誤。此演算法會針對發生錯誤的樣本計算出百分比，並且會在計算出來的值大於 dist_err_percent 屬性值時使路徑失敗。

以下是 AIX PCM 的路徑屬性：

path priority
可修改路徑清單中的演算法方法的行為。

如果 algorithm 屬性值是 failover，路徑會保留在清單中。此清單中的順序，可決定要先選擇哪一個路徑，如果其中一個路徑失敗了，接下來要選擇哪一個路徑。順序是由 path priority 屬性值來決定的。優先順序 1 代表最高優先順序。多個路徑可以有相同的優先順序值，但如果所有的路徑都具有相同值，則會根據配置每一個路徑時的先後時間來選擇。

當演算法屬性值為 round_robin 時，path priority 演算法會將優先順序值指派給每一個路徑。將依照路徑優先順序的比例選取路徑以進行 I/O 作業。因此，會選取具有較高優先順序值的路徑進行更多 I/O 作業。如果所有路徑優先順序都一樣，則會平均選取路徑。

cntl_hcheck_int

控制器性能檢查順序會在偵測到儲存體光纖傳輸失敗之後啟動。**cntl_delay_time** 屬性可決定持續時間上限（以秒為單位），在該時間內控制器性能檢查順序為作用中。如果控制器性能檢查指令順利完成，並偵測到可用的路徑，則控制器性能檢查順序會結束，以允許 I/O 進行回復。控制器性能檢查順序結束時如果未偵測到良好的路徑，則裝置的所有擱置及後續的 I/O 都會失敗，直到裝置性能檢查器偵測到已傳回失敗的路徑為止。

如果控制器性能檢查順序為作用中，則 **cntl_hcheck_interval** 屬性為時間量（以秒為單位），在該時間內會發出下一個設定控制器性能檢查指令。**cntl_hcheck_interval** 屬性必須小於 **cntl_delay_time**，除非已設為 0 或是已停用。如果 **cntl_delay_time** 或 **cntl_hcheck_interval** 已設為 0，則會停用此特性。

cntl_delay_time

控制器性能檢查順序會在偵測到儲存體光纖傳輸失敗之後啟動。**cntl_delay_time** 屬性可決定持續時間上限（以秒為單位），在該時間內控制器性能檢查序列為作用中。如果控制器性能檢查指令順利完成，並偵測到可用的路徑，則控制器性能檢查順序會結束，並允許 I/O 進行回復。控制器性能檢查順序結束時如果未偵測到良好的路徑，則裝置的所有擱置及後續的 I/O 都會失敗，直到裝置性能檢查器偵測到已傳回失敗的路徑為止。

如果控制器性能檢查順序為作用中，則 **cntl_hcheck_interval** 屬性為時間量（以秒為單位），在該時間內會發出下一個設定控制器性能檢查指令。**cntl_hcheck_interval** 屬性必須小於 **cntl_delay_time**，除非已設為 0 或是已停用。如果 **cntl_delay_time** 或 **cntl_hcheck_interval** 已設為 0，則會停用此特性。

timeout_policy
調整 PCM 對於指令逾時和傳輸錯誤的行為。當「多路徑 I/O (MPIO)」裝置在裝置的部分路徑上遇到間歇性的儲存區域網路 (SAN) 光纖問題時，如果 **timeout_policy** 已設為 **fail_path** 或 **disable_path**，則可能會改善效能降低的情形。**timeout_policy** 屬性具有下列值：

retry_path
在路徑上第一次發生指令逾時時不會立即導致路徑失敗。如果性能檢查回復了因傳輸問題而導致失敗的路徑，就能立即使用所回復的路徑。

fail_path
路徑在第一次發生指令逾時時失敗，並假設它不是路徑群組中的最後一個路徑。如果因傳輸問題而導致失敗的路徑回復了，則必須等到在該路徑上沒有發生任何失敗的期間到期之後，才會使用該路徑進行讀取或寫入 I/O 作業。如果啟用此特性，在將讀取或寫入 I/O 遞送至從傳輸錯誤中回復的路徑之前，可能會發生延遲。

>|

fail_ctrl
此設定會導致 MPIO 更快速地從偏好的控制器切換至非偏好的控制器。如果您未啟用此設定，則在 MPIO 從偏好的控制器切換至非偏好的控制器之前，偏好的控制器的所有路徑將失敗。當您啟用此設定時，若偏好的控制器的兩個路徑發生錯誤，MPIO 即會切換。 此設定與 **fail_path** 設定類似。

|<

disable_path
路徑在第一次發生指令逾時時失敗，並假設它不是路徑群組中的最後一個路徑。如果因傳輸問題而導致失敗的路徑回復了，則必須等到在該路徑上沒有發生任何失敗的期間到期之後，才會使用該路徑進行讀取或寫入 I/O。如果此路徑在某段期間內繼續發生多次指令逾時，則可能會停用此路徑。在您執行下列其中一個動作之前，已停用的路徑會保留停用狀態（而且無法使用）：執行 **chpath** 指令以啟用已停用的路徑、重新配置受影響的磁碟，或重新啟動系統。

SAN 抄寫屬性

必須安裝「AIX 多路徑 I/O (MPIO)」，而且裝置必須使用 AIX 路徑控制模組 (PCM)。這些屬性對於儲存體子系統所提供的設定和特性具有相依關係。

下列 AIX 屬性與邏輯單元號碼 (LUN) 的行為相關，而這些號碼是透過儲存體子系統來抄寫。所有儲存體子系統或微碼層次可能不支援這些屬性。將安裝叢集作業或高可用性軟體（例如，PowerHA® SystemMirror®），在叢集的各個節點之中協調管理儲存區域網路 (SAN) 抄寫。在「虛擬 I/O 伺服器 (VIOS)」上無法變更下列屬性。

san_rep_cfg
決定如何在 AIX 作業系統中定義及配置正在使用「對等式遠端複製 (PPRC)」的同步裝置。此屬性會影響磁碟實例的 unique_id 值，因此如果變更此屬性值，此磁碟實例值也會變更。**san_rep_cfg** 屬性不會變更儲存體子系統上的 PPRC 裝置的狀態。此屬性具有下列值：

none [Default]
將正在使用 PPRC 的同步裝置上的 LUN 配置為不同邏輯磁碟實例。

new
將正在使用 PPRC 的同步裝置定義及配置為單一邏輯實例。只有在沒有任何現有的磁碟實例符合 PPRC 裝置中的任何 LUN 時，才會定義及配置此裝置。

new_and_existing
將正在使用 PPRC 的同步裝置定義及配置為單一邏輯實例。如果沒有任何邏輯磁碟實例代表 PPRC 裝置，則會定義新的磁碟實例。

migrate_disk
將正在使用 PPRC 的同步裝置定義及配置為單一邏輯實例，並針對該裝置使用選取的邏輯磁碟實例。只有其 **san_rep_device** 屬性已設為 supported 或 detected 的裝置才支援此作業。如果目標裝置將 **san_rep_device** 屬性設為 supported，則只有在前次配置磁碟後已在儲存體上設定 SAN 抄寫時，此作業才能運作。如果不使用此裝置作為叢集內的儲存庫磁碟，則 AIX 作業系統所開啟及使用中的磁碟可支援此作業。會更新受影響的磁碟的 unique_id 值來反映 PPRC 裝置的 ID。

revert_disk
將正在使用 PPRC 邏輯磁碟實例的現有同步裝置配置為非 PPRC 裝置硬碟實例。只有其 **san_rep_device** 屬性已設為 yes 的裝置才支援此作業。目標磁碟實例的邏輯裝置名稱和特殊檔案會維持不變。將 SAN 抄寫裝置的主要（來源）LUN 用於回復的硬碟實例。如果找不到主要（來源）LUN 或此 LUN 對主機而言是不明的，則會使用次要（目的地）LUN 作為回復的硬碟實例。如果不使

用此裝置作為叢集內的儲存庫磁碟，則 AIX 作業系統所開啟及使用中的磁碟可支援此作業。會更新受影響的磁碟的 unique_id 值來反映 LUN ID。

none、**new** 及 **new_and_existing** 值表示要更新相同「物件資料管理程式 (ODM)」唯一類型的所有裝置的行為。**chdef** 指令是用來設定 **none**、**new** 及 **new_and_existing** 值。**chdef** 指令會更新所指定的 ODM 唯一類型的所有裝置的屬性預設值。**chdef** 指令還會更新已定義的現有裝置實例的屬性值。對於執行 **chdef** 指令時已配置的裝置而言，只有在重新啟動系統或解除配置並重新配置那些裝置之後，變更才會生效。**chdef** 指令需要屬性的類別、子類別及類型。若要判定 **san_rep_cfg** 屬性的 ODM 唯一類型，請使用 lsattr –l hdisk# -F class,subclass,type 指令。例如：

```
lsdev –l hdisk0 –F class,subclass,type
disk,fcp,aixmpiods8k
chdef –a san_rep_cfg=none –c disk –s fcp –t aixmpiods8k
chdef –a san_rep_cfg=new –c disk –s fcp –t aixmpiods8k
chdef –a san_rep_cfg=new_and_existing –c disk –s fcp –t aixmpiods8k
```

migrate_disk 和 **revert_disk** 值表示要更新單一及指定的裝置實例的行為。必須使用 **chdev** 指令來設定所指定的裝置的 **migrate_disk** 或 **revert_disk** 值。**chdev** 指令只會更新所指定的裝置的值。例如：

```
chdev -a san_rep_cfg=migrate_disk -l hdisk0
chdev -a san_rep_cfg=revert_disk -l hdisk0
```

san_rep_device
指出邏輯磁碟實例已定義為 SAN 抄寫裝置。此屬性是在磁碟配置期間所設定的，因此如果裝置的狀態在配置磁碟後已發生變更，那麼此屬性就是舊的。此屬性具有下列值：

否
裝置未在 AIX 作業系統中配置為 SAN 抄寫裝置。此裝置不符合要配置為 SAN 抄寫裝置的需求。

supported
裝置未在 AIX 作業系統中配置為 SAN 抄寫裝置。此裝置符合要配置為 SAN 抄寫裝置的需求。不過，它目前未在儲存體子系統上設定為 SAN 抄寫裝置。

detected
裝置未在 AIX 作業系統中配置為 SAN 抄寫裝置。AIX 作業系統偵測到此裝置符合要配置為 SAN 抄寫裝置的需求，而且目前已在儲存體子系統上設定為 SAN 抄寫裝置。

是
裝置已在 AIX 作業系統中配置為 SAN 抄寫裝置。

移除通訊配接卡

您必須先取消配置熱插拔配接卡，然後才能移除或取代該配接卡。

取消配置通訊配接卡包含下列作業：

- 關閉正在使用您要移除或取代的配接卡的所有應用程式
- 確定連接至配接卡的所有裝置均已識別及停止
- 列出目前正在使用中的所有介面槽，以及由某一特定配接卡使用的介面槽
- 識別配接卡的介面槽位置
- 顯示並移除網路介面清單中的介面資訊
- 使配接卡無法使用

若要使用下列程序來取消配置通訊配接卡，您必須以 **root** 身分登入：

取消配置乙太網路、記號環、FDDI 及 ATM 配接卡

若要取消配置乙太網路、記號環、FDDI 或 ATM 配接卡，請執行下列步驟：

1. 鍵入 lsslot -c pci，列出主機中的所有熱插拔槽並顯示其性質。
2. 鍵入適當的 SMIT 指令（如下面範例所示），以列出安裝的配接卡並顯示主機中所有裝置的現行狀態：

項目	說明
smit lsdenet	可以列出乙太網路配接卡
smit lsdtok	可以列出記號環配接卡
smit ls_atm	可以列出 ATM 配接卡

下列命名慣例用於不同的配接卡類型：

項目	說明
名稱	**配接卡類型**
atm0, atm1, ...	ATM 配接卡
ent0, ent1, ...	乙太網路配接卡
tok0, tok1, ...	記號環配接卡

3. 關閉正在使用您要取消配置的配接卡的所有應用程式。

   如果要繼續進行這個程序，系統必須停用網路傾出位置。如果要尋找和停用網路傾出位置，請執行下列步驟：

   a) 從指令行輸入下述指令：

   ```
   smit dump
   ```

   b) 選取**顯示現行傾出裝置**。

   c) 檢查是否有任何配置的傾出裝置顯示了網路位置。

   如果沒有，請結束 SMIT，跳至步驟 4。如果要將傾出裝置改為本端位置，請選取**取消**，或按 F3，繼續進行下一步驟。

   d) 如果主要傾出裝置顯示了網路位置，請選取**變更主要傾出裝置**，改為本端位置，然後在**主要傾出裝置**欄位中，輸入本端位置。

   e) 如果次要傾出裝置顯示了網路位置，請選取**變更次要傾出裝置**，改為本端位置，然後在**次要傾出裝置**欄位中，輸入本端位置。

   f) 完成時，請按一下**確定**，或者按 Enter 鍵。

4. 鍵入 `netstat -i`， 顯示所有已配置的介面清單，並判斷您的配接卡是否已配置 TCP/IP。會顯示如下的輸出：

```
Name  Mtu    Network   Address        Ipkts  Ierrs  Opkts Oerrs Coll
lo0   16896  link#1                     076      0    118     0    0
lo0   16896  127       127.0.0.1        076      0    118     0    0
lo0   16896  ::1                        076      0    118     0    0
tr0    1492  link#2    8.0.5a.b8.b.ec   151      0    405    11    0
tr0    1492  19.13.97  19.13.97.106     151      0    405    11    0
at0    9180  link#3    0.4.ac.ad.e0.ad    0      0      0     0    0
at0    9180  6.6.6     6.6.6.5            0      0      0     0    0
en0    1500  link#5    0.11.0.66.11.1   212      0      1     0    0
en0    1500  8.8.8     8.8.8.106        212      0      1     0    0
```

記號環配接卡只能有一個介面。乙太網路配接卡可以有兩個介面。ATM 配接卡可以有多個介面。

5. 鍵入適當的 **ifconfig** 指令（如下面範例所示），以移除網路介面清單中的介面。

項目	說明
`ifconfig en0 detach`	可以移除標準乙太網路介面
`ifconfig et0 detach`	可以移除 IEEE 802.3 乙太網路介面
`ifconfig tr0 detach`	可以移除記號環介面
`ifconfig at0 detach`	可以移除 ATM 介面

6. 鍵入適當的 **rmdev** 指令（如下面範例所示），取消配置配接卡並在「自訂的裝置物件類別」中保留其裝置定義：

項目	說明
`rmdev -l ent0`	可以取消配置乙太網路配接卡
`rmdev -l tok1`	可以取消配置記號環配接卡
`rmdev -l atm1`	可以取消配置 ATM 配接卡
`rmdev -p pci1`	取消配置 PCI 匯流排的子項，以及其下其他所有的裝置，並將其裝置定義保留在自訂的裝置物件類別中。

註： 若要取消配置配接卡並在「自訂的裝置」物件類別中移除 裝置定義，您可以將 **rmdev** 指令與 **-d** 旗標一起搭配使用。

小心： 請勿 在熱插拔作業中將 **-d** 旗標與 **rmdev** 指令一起搭配使用，除非您想要移除配接卡而不替換它。

取消配置 WAN 配接卡

您可以取消配置 WAN 配接卡。

若要取消配置 WAN 配接卡：

1. 鍵入 `lsslot -c pci`，列出主機中的所有熱插拔槽並顯示其性質。
2. 鍵入適當的 SMIT 指令（如下面範例所示），以列出安裝的配接卡並顯示主機中所有裝置的現行狀態：

項目	說明
`smit 331121b9_ls`	列出 2 埠多重通訊協定 WAN 配接卡
`smit riciophx_ls`	列出 ARTIC WAN 配接卡

下列命名慣例用於不同的配接卡類型：

名稱	配接卡類型
dpmpa	2 埠多重通訊協定配接卡
riciop	ARTIC960 配接卡

3. 關閉正在使用您要取消配置的配接卡的所有應用程式。
   如果要繼續進行這個程序，系統必須停用網路傾出位置。如果要尋找和停用網路傾出位置，請執行下列步驟：
   a) 從指令行輸入下述指令：
   ```
   smit dump
   ```
   b) 選取**顯示現行傾出裝置**。
   c) 檢查是否有任何配置的傾出裝置顯示了網路位置。
   如果沒有，請結束 SMIT，您就可以準備進行底下的步驟 5。如果要將傾出裝置改為本端位置，請選取**取消**，或按 F3，繼續進行下一步驟。
   d) 如果主要傾出裝置顯示了網路位置，請選取**變更主要傾出裝置**，改為本端位置，然後在**主要傾出裝置**欄位中，輸入本端位置。
   e) 如果次要傾出裝置顯示了網路位置，請選取**變更次要傾出裝置**，改為本端位置，然後在**次要傾出裝置**欄位中，輸入本端位置。
   f) 完成時，請按一下**確定**，或者按 Enter 鍵。
4. 使用下表中的指令，取消配置及移除這些配接卡的裝置驅動程式與模擬程式埠：

名稱	「2 埠多重通訊協定」配接卡
smit rmhdlcdpmpdd	取消配置裝置
smit rmsdlcscied	取消配置 SDLC COMIO 模擬程式

名稱	ARTIC960Hx PCI 配接卡
smit rmtsdd	取消配置裝置驅動程式
smit rmtsdports	移除 MPQP COMIO 模擬埠

取消配置 IBM 4-Port 10/100 Base-TX Ethernet PCI 配接卡

4-Port 10/100 Base-TX Ethernet PCI 配接卡有四個乙太網路埠，且您必須先取消配置每一個埠，然後才能移除配接卡。

1. 鍵入 lsslot -c pci，列出主機中的所有熱插拔槽並顯示其性質。
2. 鍵入 smit lsdenet，列出 PCI 子類別中的所有裝置。
   會顯示類似下面的訊息：

```
ent1  Available 1N-00 IBM 4-Port 10/100 Base-TX Ethernet PCI Adapter (23100020) (Port 1)
ent2  Available 1N-08 IBM 4-Port 10/100 Base-TX Ethernet PCI Adapter (23100020) (Port 2)
ent3  Available 1N-10 IBM 4-Port 10/100 Base-TX Ethernet PCI Adapter (23100020) (Port 3)
ent4  Available 1N-18 IBM 4-Port 10/100 Base-TX Ethernet PCI Adapter (23100020) (Port 4)
```

3. 關閉正在使用您要取消配置的配接卡的所有應用程式。
   如果要繼續進行這個程序，系統必須停用網路傾出位置。如果要尋找和停用網路傾出位置，請執行下列步驟：
   a) 從指令行輸入下述指令：

```
smit dump
```

   b) 選取**顯示現行傾出裝置**。
   c) 檢查是否有任何配置的傾出裝置顯示了網路位置。
   如果沒有，請結束 SMIT，跳至步驟 4。如果要將傾出裝置改為本端位置，請選取**取消**，或按 F3，繼續進行下一步驟。
   d) 如果主要傾出裝置顯示了網路位置，請選取**變更主要傾出裝置**，改為本端位置，然後在**主要傾出裝置**欄位中，輸入本端位置。
   e) 如果次要傾出裝置顯示了網路位置，請選取**變更次要傾出裝置**，改為本端位置，然後在**次要傾出裝置**欄位中，輸入本端位置。
   f) 完成時，請按一下**確定**，或者按 Enter 鍵。
4. 鍵入 netstat -i， 顯示所有已配置的介面清單，並判斷您的配接卡是否已配置 TCP/IP。
   會顯示如下的輸出：

```
Name  Mtu    Network   Address        Ipkts  Ierrs  Opkts Oerrs Coll
lo0   16896  link#1                     076      0    118     0    0
lo0   16896  127       127.0.0.1        076      0    118     0    0
lo0   16896  ::1                        076      0    118     0    0
tr0    1492  link#2    8.0.5a.b8.b.ec   151      0    405    11    0
tr0    1492  19.13.97  19.13.97.106     151      0    405    11    0
at0    9180  link#3    0.4.ac.ad.e0.ad    0      0      0     0    0
at0    9180  6.6.6     6.6.6.5            0      0      0     0    0
en0    1500  link#5    0.11.0.66.11.1   212      0      1     0    0
en0    1500  8.8.8     8.8.8.106        212      0      1     0    0
```

   乙太網路配接卡可以有兩個介面，例如 et0 及 en0。
5. 使用 **ifconfig** 指令以移除網路介面清單中的每一個介面。
   例如，鍵入 iconfig en0 detach 以移除標準乙太網路介面，並鍵入 iconfig et0 以移除 IEEE 802.3 介面。
6. 使用 **rmdev** 指令以取消配置配接卡，並在「自訂的裝置物件類別」中保留其裝置定義。

例如，rmdev -l ent0。

註：若要取消配置配接卡並在「自訂的裝置」物件類別中移除 裝置定義，您可以將 **rmdev** 指令與 **-d** 旗標一起搭配使用。

小心：請勿 在熱插拔作業中將 **-d** 旗標與 **rmdev** 指令一起搭配使用，除非您想要移除配接卡而不替換它。

取消配置 ATM 配接卡

您必須先取消配置每一個 LAN 模擬裝置，然後才能移除配接卡。

在 ATM 配接卡上可以執行標準 IP 與 LAN 模擬通訊協定。LAN 模擬通訊協定可以在 ATM 網路上施行模擬的 LAN。模擬的 LAN 可以是乙太網路/IEEE 802.3、記號環/IEEE 802.5 及 MPOA (MultiProtocol Over ATM)。

若要移除 LAN 介面，請執行下列動作：

1. 鍵入 lsslot -c pci，列出主機中的所有熱插拔槽並顯示其性質。
2. 鍵入 smit ls_atm 以列出所有 ATM 配接卡。
   會顯示類似下面的訊息：

```
.
.
atm0 Available 04-04 IBM PCI 155 Mbps ATM Adapter (14107c00)
atm1 Available 04-06 IBM PCI 155 Mbps ATM Adapter (14104e00)
```

3. 鍵入 smit listall_atmle 以列出配接卡上的所有 LAN 模擬用戶端。
   會顯示類似下面的訊息：

```
ent1 Available  ATM LAN Emulation Client (Ethernet)
ent2 Available  ATM LAN Emulation Client (Ethernet)
ent3 Available  ATM LAN Emulation Client (Ethernet)
tok1 Available  ATM LAN Emulation Client (Token Ring)
tok2 Available  ATM LAN Emulation Client (Token Ring)
```

   所有 ATM 配接卡上可以執行多個模擬的用戶端。
4. 鍵入 smit listall_mpoa 以列出配接卡上所有 LAN 模擬用戶端。
   會顯示類似下面的訊息：

```
mpc0 Available     ATM LAN Emulation MPOA Client
```

   atm0 及 atm1 是實體 ATM 配接卡。mpc0 是 MPOA 模擬用戶端。ent1、ent2、ent3、tok1 及 tok2 都是 LAN 模擬用戶端。
5. 鍵入 entstat 以判斷執行用戶端的配接卡。
   會顯示類似下面的訊息：

```
-------------------------------------------------------------
ETHERNET STATISTICS (ent1) :
Device Type: ATM LAN EmulationATM Hardware Address: 00:04:ac:ad:e0:ad
.
.
.
ATM LAN Emulation Specific Statistics:
------------------------------------
Emulated LAN Name: ETHelan3
Local ATM Device Name: atm0
Local LAN MAC Address:
.
.
```

6. 關閉正在使用您要取消配置的配接卡的所有應用程式。
   如果要繼續進行這個程序，系統必須停用網路傾出位置。如果要尋找和停用網路傾出位置，請執行下列步驟：
   a) 從指令行輸入下述指令：

```
smit dump
```

b) 選取**顯示現行傾出裝置**。

c) 檢查是否有任何配置的傾出裝置顯示了網路位置。

如果沒有，請結束 SMIT，跳至步驟 7。如果要將傾出裝置改為本端位置，請選取**取消**，或按 F3，繼續進行下一步驟。

d) 如果主要傾出裝置顯示了網路位置，請選取**變更主要傾出裝置**，改為本端位置，然後在**主要傾出裝置**欄位中，輸入本端位置。

e) 如果次要傾出裝置顯示了網路位置，請選取**變更次要傾出裝置**，改為本端位置，然後在**次要傾出裝置**欄位中，輸入本端位置。

f) 完成時，請按一下**確定**，或者按 Enter 鍵。

7. 使用 **rmdev -l** *device* 指令，依照下列次序來取消配置介面：

· 模擬介面 = en1、et1、en2、et2、tr1、tr2 ...

· 模擬介面 = ent1、ent2、tok1、tok2 ...

· Multiprotocol Over ATM (MPOA) = mpc0

· ATM 配接卡 = atm0

8. 如果要取消配置 SCSI 配接卡 `scsi1` 及其所有的子項，並將其裝置定義保留在自訂的裝置物件類別中，請鍵入：

```
rmdev  -R scsi1
```

系統會顯示一則類似下面所示的訊息：

```
        rmt0 已經定義
        hdisk1 已經定義
        scsi1 已經定義
```

9. 如果只要取消配置 SCSI 配接卡 `scsi1` 的子項，但不取消配置配接卡本身，並將其裝置定義保留在自訂的裝置物件類別中，請鍵入：

```
rmdev  -p scsi1
```

系統會顯示一則類似下面所示的訊息：

```
        rmt0 已經定義
        hdisk1 已經定義
```

10. 如果要取消配置 PCI 匯流排 `pci1` 的子項，以及其下其他所有的裝置，並將其裝置定義保留在自訂的裝置物件類別中，請鍵入：

```
rmdev  -p pci1
```

系統會顯示一則類似下面所示的訊息：

```
         rmt0 已經定義
         hdisk1 已經定義
         scsi1 已經定義
         ent0 已經定義
```

取消配置儲存體配接卡

您必須先取消配置儲存體配接卡，然後才能移除或取代該配接卡。

若要執行這些作業，您必須以 root 使用者的身分登入。

下列步驟取消配置 SCSI 及「光纖通道」儲存體配接卡。

取消配置儲存體配接卡包含下列作業：

· 關閉正在使用您要移除、取代或移動的配接卡的所有應用程式

· 卸載檔案系統

· 確定連接至配接卡的所有裝置均已識別及停止
· 列出目前正在使用中的所有介面槽，以及由某一特定配接卡使用的介面槽
· 識別配接卡的介面槽位置
· 使母項與子項裝置無法使用
· 使配接卡無法使用

取消配置 SAS、SCSI、NVMe 及「光纖通道」配接卡

儲存體配接卡通常是媒體裝置（如磁碟機或磁帶機）的母項裝置。移除母項必須先移除所有連接的子項裝置，或將子項裝置至於定義狀態。

若要取消配置 SCSI 及「光纖通道」配接卡，請執行下列步驟：

1. 關閉正在使用您要取消配置的配接卡的所有應用程式。
2. 鍵入 `lsslot-c pci`，以列出主機中的所有熱插拔槽並顯示其性質。
3. 鍵入 `lsdev -C`，列出主機中所有裝置的現行狀態。
4. 鍵入 `umount`， 卸載之前使用此配接卡裝載的檔案系統、目錄或檔案。如需相關資訊，請參閱裝載 JFS 或 JFS2。
5. 鍵入 `rmdev -l adapter -R`，使配接卡無法使用。

小心：對於熱插拔作業，請不要 在 **rmdev** 指令使用 **-d** 旗標，因為這會導致配置遭到移除。

取消配置非同步配接卡

您可以取消配置非同步配接卡。

若要執行這些作業，您必須以 root 使用者的身分登入。

下列是取消配置非同步配接卡的步驟。

您必須先取消配置非同步配接卡，然後才能移除或取代該配接卡。取消配置非同步配接卡包括下列作業：

· 關閉正在使用您要移除、取代或移動的配接卡的所有應用程式
· 確定連接至配接卡的所有裝置均已識別及停止
· 列出目前正在使用中的所有介面槽，以及由某一特定配接卡使用的介面槽
· 識別配接卡的介面槽位置
· 使母項與子項裝置無法使用
· 使配接卡無法使用

程序

您必須先取消配置配接卡及由該配接卡所控制的所有裝置，然後才能取代或移除該非同步配接卡。若要取消配置裝置，您必須終止使用那些裝置的所有處理程序。請使用下列步驟：

1. 關閉正在使用您要取消配置的配接卡的所有應用程式。
2. 鍵入 `lsslot-c pci`，以列出主機中的所有熱插拔槽並顯示其性質。
3. 鍵入 `lsdev -C -c tty`，以列出所有可用的 tty 裝置，以及主機中所有裝置的現行狀態。
4. 鍵入 `lsdev -C -c printer`，以列出連接至該配接卡的所有印表機與繪圖機裝置。
5. 使用 **rmdev** 指令，使配接卡無法使用。

小心：對於熱插拔作業，請不要在 **rmdev** 指令使用 **-d** 旗標，因為這會導致配置遭到移除。

疑難排解 I/O 裝置

您可以判斷裝置問題的原因。

裝置軟體檢查

您可以執行下列動作，以更正裝置軟體問題：

- 檢查錯誤日誌
- 列出所有裝置
- 檢查裝置的狀態
- 檢查裝置的屬性
- 變更裝置的屬性
- 以另一個應用程式使用裝置
- 定義新裝置

錯誤日誌檢查

檢查錯誤日誌以察看是否記錄了裝置、配接卡或使用裝置的應用程式錯誤。請前往錯誤記載機能，以取得執行此檢查的相關資訊。完成程序後，請返回此步驟。

您是否已更正了裝置問題？

如果無法使用先前的方法更正問題，請前往下一個步驟（**裝置清單**），以列出所有裝置。

裝置清單

使用 **lsdev -C** 指令，列出所有已定義或可用的裝置。此指令會顯示系統中所有裝置的性質。

如果裝置出現在裝置清單中，請前往下一個步驟（**裝置狀態檢查**），檢查裝置的狀態。

如果裝置未出現在裝置清單中，請定義新的裝置（請參閱底下的**新建裝置定義**）。

裝置狀態檢查

在 **lsdev -C** 指令產生的清單中找出裝置。檢查裝置是否處於 Available 狀態。

如果裝置處於 Available 狀態，請前往下一個步驟（**裝置屬性檢查**），檢查裝置屬性。

如果裝置不處於 Available 狀態，請定義新的裝置（請參閱底下的**新裝置定義**）。

裝置屬性檢查

使用 **lsattr -E -l** *DeviceName* 指令，列出裝置的屬性。

lsattr 指令會顯示系統中裝置的屬性性質與可能的屬性值。請參照特定裝置的文件，以取得正確的設定。

如果裝置屬性設定正確，請參閱底下的**將裝置與其他應用程式搭配使用**。

如果裝置屬性設定不正確，請前往下一個步驟（**裝置屬性變更**）。

裝置屬性變更

使用 **chdev-l** *Name* **-a** *Attribute*=*Value* 指令，變更裝置屬性。在您執行此指令之前，請參閱 *Commands Reference, Volume 1*。

chdev 指令會變更您使用 **-l** *Name* 旗標指定的裝置性質。

如果變更屬性沒有更正裝置的問題，請前往下一個步驟（**將裝置與其他應用程式搭配使用**）。

將裝置與其他應用程式搭配使用

嘗試以另一個應用程式使用裝置。如果裝置能與另一個應用程式一起正確地運作，則可能是第一個應用程式的問題。

如果裝置能與另一個應用程式一起正確地運作，則您的第一個應用程式可能有問題。請向軟體客戶服務代表報告此問題。

如果裝置無法與其他應用程式正確運作，請前往下一個步驟（**新建裝置定義**）。

新建裝置定義

註 : 您必須具有 root 使用者權限或是為安全群組的成員，才能使用 **mkdev** 指令。

使用 **mkdev** 指令以將裝置新增至系統。

mkdev 指令可以定義並使新裝置可供使用，或使已經定義的裝置可使用。您可以使用 **-c**、**-s** 及 **-t** 旗標的任何組合，獨一無二地識別預先定義的裝置。執行此指令之前，請參照 *Commands Reference, Volume 3*。

如果定義裝置未能更正問題，您可以停止並向客戶服務代表報告問題，或使用診斷程式來測試裝置。

檢查裝置連線

若要檢查裝置連線，請執行下列步驟：

1. 檢查電器插座上是否有電源可用。
2. 檢查裝置電源纜線是否已正確地連接裝置與電器插座。
3. 檢查裝置信號纜線是否已正確地連接裝置與主機上的正確連線。
4. 若為 SCSI 裝置，請檢查 SCSI 終止器是否已正確地連接，以及 SCSI 位址設定是否正確。
5. 若為通訊裝置，請檢查裝置是否已正確地連接通訊線路。
6. 檢查裝置是否已開啟。

請參照特定裝置的文件，以取得纜線安裝與配置程序，以及進一步的疑難排解資訊。

如果此主題中的步驟未更正問題，請前往下一步。

配接卡移除問題解決方法

當使用 **rmdev** 指令來取消配置配接卡時，如果開啟裝置，可能會收到錯誤訊息。

在使用 **rmdev** 指令來取消配置配接卡時，如果出現下列類型的訊息，表示裝置處於開啟狀態，可能是因為應用程式仍在嘗試存取您要移除或取代的配接卡。

```
#rmdev -l ent0
方法錯誤 (/usr/lib/methods/ucfgent) :
          0514-062
        由於指定的裝置忙碌中，所以無法執行
        所要求的功能。
```

若要解決問題，您必須識別仍在使用配接卡的所有應用程式並關閉它們。這些應用程式可能包括下列各項：

- TCP/IP
- SNA
- OSI
- IPX/SPX
- Novell NetWare
- Streams
- 同屬的資料鏈結控制 (GDLC)
  - IEEE 乙太網路 DLC
  - 記號環 DLC
  - FDDI DLC

系統網路架構應用程式

部分可能正在使用配接卡的 SNA 應用程式包括：

- DB2®
- TXSeries® (CICS® & Encina)
- DirectTalk

· MQSeries®

· HCON

· ADSM

Streams 應用程式

部分可能正在使用配接卡的 streams 型應用程式包括：

· IPX/SPX

· Novell NetWare V4 與 Novell NetWare Services 4.1

· 此作業系統的連線與 NetBios

在 WAN 配接卡上執行的應用程式

可能正在使用 WAN 配接卡的應用程式包括：

· SDLC

· Bisync

· ISDN

TCP/IP 應用程式

您可以使用 **ifconfig** 指令，分離正在使用介面層的所有 TCP/IP 應用程式。這會造成使用 TCP/IP 的應用程式逾時，並警告使用者介面關閉。在您新增或取代配接卡，並執行 **ifconfig** 指令以連接介面之後， 應用程式即會回復。

檢查裝置的備妥狀態

您可以檢查裝置是否處於備妥狀態。

欲判斷裝置是否已處於備妥狀態，請執行下列動作：

1. 檢查裝置的「備妥」指示器是否已開啟。
2. 檢查抽取式媒體（如磁帶、磁片或光碟裝置）是否正確地插入。
3. 檢查印表機與繪圖機的色帶、紙張供應及碳粉供應。
4. 如果您嘗試寫入裝置，請檢查寫入媒體是否已啟用寫入。

您的檢查是否已更正裝置的問題？如果裝置的備妥狀態檢查未更正問題，請前往下一步。

裝置診斷程式

若要判斷裝置是否毀損，請執行硬體診斷程式。

如果執行中硬體診斷程式無法找出裝置的問題，請檢查裝置軟體。如果裝置通過診斷測試，則可能是裝置使用系統軟體的方式有問題。如果可能是之前存在的問題，請向軟體服務組織報告此問題。

目標裝置配置

cfgmgr 指令可以與 **-c** 旗標一起使用，作為 I/O 裝置的目標配置有限範圍的連線選項。

FC 和 FCoE 裝置的目標配置

cfgmgr -c 選項與「光纖通道 (FC)」和「透過乙太網路的光纖通道 (FCoE)」配接卡一起使用，以進行目標配置。

cfgmgr 指令可以與 **-c** 旗標一起使用，作為裝置配置有限範圍的連線選項。對於 FC 和 FCoE 配接卡，其語法如下所示：

```
cfgmgr -l fscsi0 -c "parameter=val[,parameter=val,...]"
```

利用連線過濾器字串，您可以使用下列其中一個以上的參數來限制裝置探索的範圍：

表 9. ***cfgmgr -c*** 旗標的參數	
參數名稱	**說明**
ww_name	目標裝置全球埠名稱
node_name	目標裝置全球節點名稱
scsi_id	目標裝置 N_Port ID，其對映到「光纖通道通訊協定 (FCP)」儲存裝置的「小型電腦系統介面 (SCSI)」ID
lun_id	邏輯單元號碼 (LUN)

例如，下列指令會在具有全球埠名稱 0x5001738000330191 的儲存體目標埠上配置 **lun_id** 0x1000000000000 的單一 LUN：

```
# cfgmgr -l fscsi0 -c "ww_name=0x5001738000330191,lun_id=0x1000000000000"
```

此掃描只會針對 `fscsi0` 主機配接卡埠進行。

附註：

· 參數值中的前導字元 `0x` 是選用項目。

· 所有參數都必須以十六進位數代表。

在下列範例中，只指定一個參數：

```
# cfgmgr -l fscsi0 -c "lun_id=0x1000000000000"
```

此指令會掃描儲存區網路 (SAN) 上的所有儲存裝置埠，並為此 LUN 所在的每一個 SAN 目標埠配置這個單一邏輯單元。

連線過濾器參數的準則和規則

當您使用連線過濾器參數時，請考量下列要點：

· FC 和 FCoE 裝置的目標配置僅適用於切換-連接環境。如果您指定的連線字串是直接連接到目標埠，則連線會失敗，並出現訊息指出找不到子項裝置。

· 唯有隨附 **cfgmgr** 指令的 **-l** 旗標時，才支援 **-c** 旗標，該指令將指令範圍限制為一次一個 `fscsiX` 裝置。

· 如果您指定 **-?** 作為 **cfgmgr** 指令的 **-c** 旗標的連線字串，則用法資訊會連同 **-v** 旗標一起顯示出來。

· 如果您指定重複的參數（例如，**lun_id** 列出兩次），則它會產生錯誤。未偵測到裝置。

· 除了重複項之外，容許 **lun_id**、**scsi_id**、**ww_name** 和 **node_name** 參數的任何組合。 若要專門識別要配置的 LUN、目標或儲存節點，您必須指定一個（最好兩個）參數，不過也可以容許更多參數。 下列清單指定要專門識別 LUN、目標或儲存節點時所需的參數或參數組合：

  – **ww_name** 和 **lun_id** 參數專門識別目標埠上要配置的 LUN。

  – **scsi_id** 和 **lun_id** 參數專門識別目標埠上要配置的 LUN。

  – **node_name** 和 **lun_id** 參數為特定儲存節點的所有目標埠配置 LUN。 唯有當所有目標埠都具有相同的 **node_name** 參數（這種情況對某些儲存裝置可能屬實），這些參數才能配置目標埠。

  – **ww_name** 參數為特定目標配置所有 LUN。

  – 唯有當所有目標埠都具有相同的 **node_name** 參數（這種情況對某些儲存裝置可能屬實），**node_name** 參數才會為特定儲存節點配置所有目標埠。

  – **lun_id** 參數對 `fscsi` 裝置上可見的所有目標埠配置 LUN。

· 如果指定的參數超過兩個，裝置配置程式碼會使用此額外資訊來驗證裝置位置。如果任何指定的參數值與 SAN 上所報告的值相衝突，則指令會失敗，而且不會配置裝置。

磁帶機

這裡說明與磁帶機相關的系統管理功能。

在這些功能中，有很多會變更，或從包含系統裝置相關資訊的裝置配置資料庫中取得資訊。該裝置配置資料庫由預先定義的配置資料庫（包含系統上支援之所有可能類型裝置的相關資訊）及自訂配置資料庫（包含目前系統上之特殊裝置的相關資訊）所組成。若要讓作業系統能夠使用磁帶機或任何其他裝置，該裝置必須定義於自訂的配置資料庫中，且必須在預先定義的配置資料庫中定義裝置類型。

磁帶機屬性

您可以調整這些磁帶機屬性，以符合系統的需求。

可以使用 SMIT 或指令（尤其是 **lsattr** 及 **chdev** 指令）來顯示或變更屬性。

每一種類型的磁帶機只使用所有屬性的子集。

每一種屬性的一般相關資訊

區塊大小

區塊大小屬性表示在讀取或寫入磁帶時，所使用的區塊大小。資料會寫入磁帶的資料區塊，且在區塊間具有記錄之間的間隙。寫入未格式化的磁帶時，較大的記錄是很有用的，因為減少磁帶上記錄之間的間隙數就可以寫入更多資料。值 **0** 表示可變長度區塊。可允許的值與預設值會隨著磁帶機而變更。

裝置緩衝區

將「裝置緩衝區」屬性設為（使用 **chdev** 指令）mode=yes，指出將資料轉送至磁帶機的資料緩衝區之後（但不必在資料實際寫入磁帶之後），通知應用程式寫入完成。如果指定 mode=no，則僅當資料實際寫入磁帶之後，才會通知應用程式寫入完成。如果將此屬性設為 mode=no 值，則無法維護讀取或寫入的串流模式。預設值是 mode=yes。

使用 mode=no 值，磁帶機會較慢，但會在停電或系統故障發生時保留更完整的資料，並讓您更有技巧地處理媒體尾端狀況。

延伸檔案標示

如果將「擴充檔案標示」屬性（對於 **chdev** 指令是 extfm 屬性）設為 no 值，則每當寫入檔案標示時，都會將一般檔案標示寫入磁帶。將此屬性設為 yes 值會寫入擴充檔案標示。若為磁帶機，此屬性可以設定為此值。預設值為 no。例如，8 mm 磁帶機上的擴充檔案標示會使用 2.2 MB 的磁帶，並且最多會花費 8.5 秒時間來寫入。一般檔案標示使用 184 KB，並大約要花費 1.5 秒來寫入。

若要在以附加模式使用 8 mm 的磁帶時減少錯誤，請使用擴充檔案標示，以在檔案標示處反向作業之後進行更佳的定位。

重新拉緊

如果將 *Retensioning* 屬性（對於 **chdev** 指令是 ret 屬性）設為 ret=yes，則每當插入磁帶或重設磁帶機時，會指示磁帶機自動重新拉緊磁帶。重新拉緊磁帶表示轉動至磁帶末端，然後倒轉至磁帶開頭以使磁帶上的張力相等。重新拉緊 磁帶可以減少錯誤，但此動作會需要數分鐘的時間。如果指定 ret=no 值，則磁帶機不會自動重新拉緊磁帶。預設值是 yes。

密度設定 #1 及密度設定 #2

「密度設定 #1」(對於 **chdev** 指令是 density_set_1 屬性）會設定當磁帶機使用特殊檔案 /dev/rmt*、/dev/rmt*.1、/dev/rmt*.2 及 /dev/rmt*.3 時，寫入的密度值。「密度設定 #2」(對於 **chdev** 是 density_set_2 屬性）會設定當磁帶機使用特殊檔案 /dev/rmt*.4、/dev/rmt*.5、/dev/rmt*.6 及 /dev/rmt*.7 時，寫入的密度值。請參閱第 170 頁的『磁帶機的特殊檔案』，以取得進一步資訊。

密度設定是以範圍 **0** 到 **255** 之間的十進位數來表示。零 (**0**) 的設定值會選取磁帶機的預設密度，其通常是磁帶機的高密度設定。特定的允許值與其意義會隨著不同的磁帶機類型而變更。如果磁帶是以磁帶機所支援的密度寫入，則這些屬性不會影響磁帶機讀取磁帶的能力。慣例是將「密度設定 #1」設為磁帶機上最高的可能密度，並將「密度設定 #2」設為磁帶機上次高的可能密度。

保留支援

對於使用「保留」屬性（對於 **chdev** 指令是 res_support 屬性）的磁帶機，指定值 res_support=yes 會導致磁帶機在開啟時，保留在 SCSI 匯流排上。如果有多個 SCSI 配接卡共用磁帶機，這可確保在裝置開啟時由單一配接卡存取。部分 SCSI 磁帶機不支援保留或放開指令。部分 SCSI 磁帶機具有此屬性的預先定義值，所以會一直支援保留或釋放指令。

可變長度區塊大小

「可變長度區塊大小」屬性（對於 **chdev** 指令是 var_block_size 屬性）指定當寫入可變長度記錄時，磁帶機所需要的區塊大小。即使在寫入可變長度記錄時，部分 SCSI 磁帶機仍需在其「模式選取」資料中指定非零區塊大小。區塊大小屬性設為 **0**，表示可變長度記錄。請參閱特定的磁帶機 SCSI 規格以判斷這是否是必要的。

資料壓縮

如果磁帶機能夠壓縮資料，則將「資料壓縮」屬性（對於 **chdev** 指令是 compress 屬性）設為 compress=yes，會導致磁帶機處於壓縮模式。如果如此，則磁碟機會以壓縮格式將資料寫入磁帶，以便在單一磁帶中能儲存更多資料。將此屬性設為 no 會強制磁帶機以原生模式（非壓縮）寫入資料。此屬性的設定不會影響讀取作業。預設設定是 yes。

自動載入器

將「自動載入器」屬性（對於 **chdev** 指令是 autoload 屬性）設為 autoload=yes，會使「自動載入器」處於作用中（如果磁帶機裝配有「自動載入器」）。如果如此，且可以在載入器中使用另一個磁帶，則當任何讀取或寫入作業將磁帶推進至尾端時，即會自動在下一個磁帶上繼續。限制在單一匣式磁帶的磁帶機指令不受影響。預設設定是 yes。

重試延遲

「重試延遲」屬性會設定指令失敗後、重新發出指令前，系統所等待的秒數。系統最多可以重新發出 4 次失敗的指令。此屬性僅適用於類型 OST 磁帶機。預設設定是 45。

讀取/寫入逾時

READ/WRITE 屬性的「讀取/寫入逾時」或「最大延遲」可設定系統允許完成讀取或寫入指令的最大秒數。此屬性僅適用於類型 OST 磁帶機。預設設定是 144。

磁帶變更時傳回錯誤

若設定「磁帶變更或重設時傳回錯誤」屬性，則如果磁帶機已重設或磁帶已變更，即會在開啟時傳回錯誤。磁帶機上的前作業結束時，磁帶定位必須是在磁帶開頭之後。傳回的錯誤是 -1，且將 errno 設為 EIO。一旦顯示給應用程式，則會清除錯誤狀況。而且，重新配置磁帶機本身也將清除錯誤狀況。

2.0 GB 4 公釐磁帶機（類型 4mm2gb）的屬性

下列是 2.0 GB 4 公釐磁帶機（類型 4mm2gb）的屬性。

區塊大小

預設值是 1024。

裝置緩衝區

此屬性的一般資訊適用於此磁帶機類型。

具有固定值的屬性

如果將磁帶機配置為 2.0 GB 4 mm 磁帶機，則「重新拉緊」、「保留支援」、「可變長度區塊大小」、「密度設定 #1」及「密度設定 #2」屬性均具有無法變更的預定值。密度設定已預先定義，因為磁帶機一律是以 2.0 GB 模式寫入。

4.0 GB 4 公釐磁帶機（類型 4mm4gb）的屬性

下列是 4.0 GB 4 公釐磁帶機（類型 4mm4gb）的屬性。

區塊大小

預設值是 1024。

裝置緩衝區

此屬性的一般資訊適用於此磁帶機類型。

密度設定 #1 及密度設定 #2

使用者不能變更此磁碟機的密度設定；裝置會依照安裝的「數位資料儲存體 (DDS)」媒體類型， 自動重新配置裝置本身，如下所示：

媒體類型	裝置配置
DDS	唯讀。
DDS ||||	只以 2.0 GB 模式讀取/寫入。
DDS2	以任一密度讀取；只以 4.0 GB 模式寫入。
非 DDS	不支援；卡匣會跳出。

資料壓縮
此屬性的一般資訊適用於此磁帶機類型。

具有固定值的屬性
如果磁帶機是配置為 4.0 GB 4 公釐磁帶機，則「重新拉緊」、「保留支援」、「可變長度區塊大小」、「密度設定 #1」及「密度設定 #2」屬性具有無法變更的預定值。

2.3 GB 8 公釐磁帶機（類型 8mm）的屬性
下列是 2.3 GB 8 公釐磁帶機（類型 8mm）的屬性。

區塊大小
預設值是 1024。較小的值會減少儲存在磁帶上的資料量。

裝置緩衝區
此屬性的一般資訊適用於此磁帶機類型。

延伸檔案標示
此屬性的一般資訊適用於此磁帶機類型。

具有固定值的屬性
如果將磁帶機配置為 2.3 GB 8mm 磁帶機，則「重新拉緊」、「保留支援」、「可變長度區塊大小」、「資料壓縮」、「密度設定 #1」及「密度設定 #2」屬性均具有無法變更的預定值。密度設定已預先定義，因為磁帶機一律是以 2.3 GB 模式寫入。

5.0GB 8 公釐磁帶機（類型 8mm5gb）的屬性
下列是 5.0GB 8 公釐磁帶機（類型 8mm5gb）的屬性。

區塊大小
預設值是 1024。如果磁帶是以 2.3 GB 模式寫入，則較小的值會減少儲存在磁帶上的資料量。

裝置緩衝區
此屬性的一般資訊適用於此磁帶機類型。

延伸檔案標示
此屬性的一般資訊適用於此磁帶機類型。

密度設定 #1 及密度設定 #2
下列設定適用：

設定	意義
140	5 GB 模式（可以壓縮）
21	5 GB 模式未壓縮的磁帶
20	2.3 GB 模式
0	預設值（5.0 GB 模式）

「密度設定 #1」的預設值是 140，且「密度設定 #2」的預設值是 20。「密度設定 #1」或「密度設定 #2」的值 21 會讓使用者以 5 GB 模式讀取或寫入未壓縮的磁帶。

資料壓縮
此屬性的一般資訊適用於此磁帶機類型。

具有固定值的屬性
如果磁帶機是配置為 5.0 GB 8 公釐磁帶機，則「重新拉緊」、「保留支援」及「可變長度區塊大小」屬性具有無法變更的預定值。

20000 MB 8 公釐磁帶機（自行配置）的屬性

下列是 20000 MB 8 公釐磁帶機（自行配置）的屬性。

區塊大小

預設值是 1024。

裝置緩衝區

此屬性的一般資訊適用於此磁帶機類型。

延伸檔案標示

此屬性的一般資訊適用於此磁帶機類型。

密度設定 #1 及密度設定 #2

磁帶機可以以 20.0 GB 格式讀取並寫入資料卡匣。「讀取」指令期間，磁帶機會自動判斷磁帶上寫入何種格式。在「寫入」期間，「密度設定」會判斷寫入磁帶的資料格式。

下列設定適用：

設定	意義
39	20 GB 模式（可以壓縮）
0	預設值（20.0 GB 模式）

「密度設定 #1」及「密度設定 #2」的預設值是 39。

資料壓縮

此屬性的一般資訊適用於此磁帶機類型。

具有固定值的屬性

如果磁帶機是配置為 20.0 GB 8 公釐磁帶機，則「重新拉緊」、「保留支援」及「可變長度區塊大小」屬性具有無法變更的預定值。

35 GB 磁帶機（類型 35gb）的屬性

下列是 35 GB 磁帶機（類型 35gb）的屬性。

區塊大小

IBM 7205 Model 311 傳輸量與區塊大小密切相關。此磁帶機的最小建議區塊大小是 32 KB。小於 32 KB 的任何區塊大小均會限制資料傳送速率（備份或還原時間）。下表按指令列出建議的區塊大小：

支援的指令	預設區塊大小（位元組）	建議
BACKUP	32 K 或 51.2 K（預設值）	依據是否按名稱執行備份，使用 32 K 或 51.2 K。不需要任何變更。
TAR	10 K	手冊中，512 KB 區塊大小的說明有誤。將 **Blocking** 參數設為 **-N64**。
MKSYSB	請參閱 **BACKUP**	**MKSYSB** 指令使用 **BACKUP** 指令。不需要任何變更。
DD	n/a	將「**Blocking** 參數」設為 **bs=32K**。
CPIO	n/a	將「**Blocking** 參數」設為 **-C64**。

註： 當您選取區塊大小時，必須瞭解容量與傳輸量。小型區塊大小對效能有重大影響，但對容量的影響最小。當您使用的區塊大小小於建議的區塊大小時，會影響 2.6 GB 格式（密度）及 6.0 GB 格式（密度）的容量。例如，使用 1024 位元組的區塊大小來備份 32 GB 的資料，大約需要 22 小時。使用 32 KB 區塊大小來備份相同的 32 GB 資料，大約需要 2 小時。

裝置緩衝區

此屬性的一般資訊適用於此磁帶機類型。

延伸檔案標示
此屬性的一般資訊適用於此磁帶機類型。

密度設定 #1 及密度設定 #2
下表顯示 IBM 7205-311 磁帶機 的「支援資料卡匣」類型及「密度設定」（十進位及十六進位）。當您執行還原（讀取）作業時，磁帶機會自動設定密度以符合寫入的密度。當您執行備份（寫入）作業時，您必須設定密度，以符合您正在使用的「資料卡匣」。

支援的資料卡匣	原有的容量	壓縮的資料容量	SMIT 密度設定	十六進位密度設定
DLTtape III	2.6 GB	2.6 GB（無壓縮）	23	17h
	6.0 GB	6.0 GB（無壓縮）	24	18h
	10.0 GB	20.0 GB（磁碟機的預設值）	25	19h
DLTtapeIIIxt	15.0 GB	30.6 GB（磁碟機的預設值）	25	19h
DLTtapeIV	20.0 GB	40.0 GB	26	1Ah
	35.0 GB	70.0 GB（磁碟機的預設值）	27	1Bh

註 : 如果您要求未支援的資料卡匣之原有的容量，則磁碟機會預設為載入磁碟機之資料卡匣的最高支援容量。

資料壓縮
實際壓縮需視要寫入的資料類型而定（請參閱前一個表格）。假設此壓縮資料容量的壓縮比例是 2:1。

具有固定值的屬性
此屬性的一般資訊適用於此磁帶機類型。

150 MB 1/4 英吋磁帶機（類型 150mb）的屬性
下列是 150 MB 1/4 英吋磁帶機（類型 150mb）的屬性。

區塊大小
預設區塊大小是 512。其他唯一可變長度區塊的有效區塊大小是 0。

裝置緩衝區
此屬性的一般資訊適用於此磁帶機類型。

延伸檔案標示
只有在磁帶開頭 (BOT) 或在偵測到空白磁帶後，才能寫入 1/4 吋磁帶。如果資料存在於磁帶上，您不能改寫資料，但在 BOT 上除外。如果您想要在已寫入資料且倒轉的磁帶中新增資料，您必須往前尋找空間，直到偵測到下一個檔案標示，這會造成系統傳回錯誤。然後，您才可以重新開始寫入。

重新拉緊
此屬性的一般資訊適用於此磁帶機類型。

密度設定 #1 及密度設定 #2
下列設定適用：

設定	意義
16	QIC-150
15	QIC-120
0	預設值 (QIC-150)，或上一次使用系統時所設定的密度。

「密度設定 #1」的預設值是 16，且「密度設定 #2」的預設值是 15。

具有固定值的屬性
如果將磁帶機配置為 150 MB 1/4 英吋磁帶機，則「擴充檔案標示」、「保留支援」、「可變長度區塊大小」及「資料壓縮」屬性均具有無法變更的預定值。

525 MB 1/4 英吋磁帶機（類型 525mb）的屬性

下列是 525 MB 1/4 英吋磁帶機（類型 525mb）的屬性。

區塊大小

預設區塊大小是 512。其他有效區塊大小是 0（可變長度區塊）及 1024。

裝置緩衝區

此屬性的一般資訊適用於此磁帶機類型。

延伸檔案標示

只有在磁帶開頭 (BOT) 或在偵測到空白磁帶後，才能寫入 1/4 吋磁帶。如果資料存在於磁帶上，您不能改寫資料，但在 BOT 上除外。如果您想要在已寫入資料且倒轉的磁帶中新增資料，您必須往前尋找空間，直到偵測到下一個檔案標示，這會造成系統傳回錯誤。然後，您才可以重新開始寫入。

重新拉緊

只有在磁帶開頭 (BOT) 或在偵測到空白磁帶後，才能寫入 1/4 吋磁帶。如果資料存在於磁帶上，您不能改寫資料，但在 BOT 上除外。如果您想要在已寫入資料且倒轉的磁帶中新增資料，您必須往前尋找空間，直到偵測到下一個檔案標示，這會造成系統傳回錯誤。然後，您才可以重新開始寫入。

密度設定 #1 及密度設定 #2

下列設定適用：

設定	意義
17	QIC-525*
16	QIC-150
15	QIC-120
0	預設值 (QIC-525)，或上一次使用系統時所設定的密度。

* QIC-525 是唯一支援 1024 區塊大小的模式。

「密度設定 #1」的預設值是 17，且「密度設定 #2」的預設值是 16。

具有固定值的屬性

如果磁帶機是配置為 525 MB 1/4 吋磁帶機，則「延伸檔案標示」、「保留支援」、「可變長度區塊大小」及「資料壓縮」屬性具有不能變更的預定值。

1200 MB 1/4 英吋磁帶機（類型 1200mb-c）的屬性

下列是 1200 MB 1/4 英吋磁帶機（類型 1200mb-c）的屬性。

區塊大小

預設區塊大小是 512。其他有效區塊大小是 0（可變長度區塊）及 1024。

裝置緩衝區

此屬性的一般資訊適用於此磁帶機類型。

延伸檔案標示

只有在磁帶開頭 (BOT) 或在偵測到空白磁帶後，才能寫入 1/4 吋磁帶。如果資料存在於磁帶上，您不能改寫資料，但在 BOT 上除外。如果您想要在已寫入資料且倒轉的磁帶中新增資料，您必須往前尋找空間，直到偵測到下一個檔案標示，這會造成系統傳回錯誤。然後，您才可以重新開始寫入。

重新拉緊

此屬性的一般資訊適用於此磁帶機類型。

密度設定 #1 及密度設定 #2

下列設定適用：

設定	意義
21	QIC-1000*
17	QIC-525*
16	QIC-150
15	QIC-120

設定	意義
0	預設值 (QIC-1000)，或上一次使用系統時所設定的密度。

* QIC-525 及 QIC-1000 是支援 1024 區塊大小的僅有模式。

「密度設定 #1」的預設值是 21，且「密度設定 #2」的預設值是 17。

具有固定值的屬性
如果磁帶機是配置為 1200 MB 1/4 吋磁帶機，則「延伸檔案標示」、「保留支援」、「可變長度區塊大小」及「資料壓縮」屬性具有不能變更的預定值。

12000 MB 4 公釐磁帶機（自行配置）的屬性
下列是 12000 MB 4 公釐磁帶機（自行配置）的屬性。

區塊大小
IBM 12000 MB 4 公釐磁帶機的傳輸量與區塊大小相關。此磁帶機的最小建議區塊大小是 32 KB。小於 32 KB 的任何區塊大小均會限制資料傳送速率（備份或還原時間）。下表按指令列出建議的區塊大小：

支援的指令	預設區塊大小（位元組）	建議
BACKUP	32 K 或 51.2 K（預設值）	依據是否按名稱執行備份，使用 32 K 或 51.2 K。不需要任何變更。
TAR	10 K	手冊中，512 KB 區塊大小的說明有誤。將 **Blocking** 參數設為 **-N64**。
MKSYSB	請參閱 BACKUP	**MKSYSB** 指令使用 **BACKUP** 指令。不需要任何變更。
DD		將 **Blocking** 參數設為 bs=32K。
CPIO		將 **Blocking** 參數設為 -C64。

註： 當您選取區塊大小時，必須瞭解容量與傳輸量。小型區塊大小對效能有重大影響，但對容量的影響最小。

裝置緩衝區
此屬性的一般資訊適用於此磁帶機類型。

延伸檔案標示
此屬性的一般資訊適用於此磁帶機類型。

密度設定 #1 及密度設定 #2
下表顯示 IBM 12000 MB 4 公釐磁帶機支援的資料卡匣類型及密度設定（十進位及十六進位）。當您執行還原（讀取）作業時，磁帶機會自動設定密度以符合寫入的密度。當您執行備份作業（寫入）時，您必須設定密度，以符合您正在使用的資料卡匣。

支援的資料卡匣	原有的容量	壓縮的資料容量	SMIT 密度設定	十六進位密度設定
DDS III	2.0 GB	4.0 GB	19	13h
DDS2	4.0 GB	8.0 GB	36	24h
DDS3	12.0 GB	24.0 GB	37	25h

註： 如果您要求未支援的資料卡匣之原有的容量，則磁碟機會預設為載入磁碟機之資料卡匣的最高支援容量。

資料壓縮
實際壓縮需視要寫入的資料類型而定（請參閱前一個表格）。假設此壓縮資料容量的壓縮比例是 2:1。

具有固定值的屬性
此屬性的一般資訊適用於此磁帶機類型。

13000 MB 1/4 英吋磁帶機（自行配置）的屬性

下列是 13000 MB 1/4 英吋磁帶機（自行配置）的屬性。

區塊大小

預設區塊大小是 512。其他有效區塊大小是 0（可變長度區塊）及 1024。

裝置緩衝區

此屬性的一般資訊適用於此磁帶機類型。

延伸檔案標示

只有在磁帶開頭 (BOT) 或在偵測到空白磁帶後，才能寫入 1/4 吋磁帶。如果資料存在於磁帶上，您不能改寫資料，但在 BOT 上除外。如果您想要在已寫入資料且倒轉的磁帶中新增資料，您必須往前尋找空間，直到偵測到下一個檔案標示，這會造成系統傳回錯誤。然後，您才可以重新開始寫入。

重新拉緊

此屬性的一般資訊適用於此磁帶機類型。

密度設定 #1 及密度設定 #2

下列設定適用：

設定	意義
33	QIC-5010-DC*
34	QIC-2GB*
21	QIC-1000*
17	QIC-525*
16	QIC-150
15	QIC-120
0	預設值 (QIC-5010-DC)*

* QIC-525、QIC-1000、QIC-5010-DC 及 QIC-2GB 是支援 1024 區塊大小的僅有模式。

「密度設定 #1」的預設值是 33，且「密度設定 #2」的預設值是 34。

具有固定值的屬性

如果磁帶機配置為 13000 MB 1/4 英吋磁帶機，則擴充檔案標示、保留支援 及可變長度區塊大小屬性都會具有無法變更的預定值。

1/2 英吋 9 磁軌磁帶機（類型 9trk）的屬性

下列是 1/2 英吋 9 磁軌磁帶機（類型 9trk）的屬性。

區塊大小

預設區塊大小是 1024。

裝置緩衝區

此屬性的一般資訊適用於此磁帶機類型。

密度設定 #1 及密度設定 #2

下列設定適用：

設定	意義
3	每英寸 6250 位元 (bpi)
2	1600 bpi
0	先前使用的任何寫入密度

「密度設定 #1」的預設值是 3，「密度設定 #2」的預設值是 2。

具有固定值的屬性

如果磁帶機配置為 1/2 英吋 9 磁軌磁帶機，則「擴充檔案標示」、「重新拉緊」、「保留支援」、「可變長度區塊大小」及「資料壓縮」等屬性都會具有無法變更的預定值。

3490e 1/2 英吋卡匣（類型 3490e）的屬性

下列是 3490e 1/2 英吋卡匣（類型 3490e）的屬性。

區塊大小

預設區塊大小是 1024。此磁帶機的特性是高資料轉送速率，且區塊大小對於作業效率很重要。較大的區塊大小可以大幅增進作業速度，且一般而言，應使用最大的可能區塊大小。

註： 增加區塊值會造成與系統上的其他程式不相容。如果發生此狀況，則在執行那些程式時，您會收到下列錯誤訊息：

```
系統呼叫收到無效的參數。
```

裝置緩衝區

此屬性的一般資訊適用於此磁帶機類型。

壓縮

此屬性的一般資訊適用於此磁帶機類型。

自動載入器

此磁碟機的特性是磁帶排序機，這是一種自動載入器，可以在卡匣載入器中循序載入及跳出一系列的匣式磁帶。若要正確操作此功能，前面板的開關應處於 AUTO 位置，且「自動載入器」屬性必須設為 `yes`。

其他 SCSI 磁帶（類型 ost）的屬性

下列是其他 SCSI 磁帶（類型 ost）的屬性。

區塊大小

系統預設值是 512，但這應調整為您磁帶機的預設區塊大小。一般值是 512 與 1024。如果將區塊大小屬性保持為 512，則 8 mm 及 4 mm 磁帶機通常會使用 1024，從而浪費了磁帶上的空間。值 0 指出部分磁帶機上的可變區塊大小。

裝置緩衝區

此屬性的一般資訊適用於此磁帶機類型。

延伸檔案標示

此屬性的一般資訊適用於此磁帶機類型。

密度設定 #1 及密度設定 #2

這兩個設定的預設值都是 0。其他值及其意義會隨著不同的磁帶機而改變。

保留支援

預設值是 no。如果磁帶機支援保留/釋放指令，則可將其設為 yes。如果不確定，則 no 是較安全的值。

可變長度區塊大小

預設可變長度區塊大小值為 0。非零值主要用於 1/4 英吋卡匣 (QIC) 磁帶機。請參閱特定磁帶機的 SCSI 規格，以取得建議。

重試延遲

此屬性只適用於類型 ost 磁帶機。

讀取/寫入逾時

此屬性只適用於類型 ost 磁帶機。

具有固定值的屬性

如果將磁帶機配置為其他 SCSI 磁帶機，則「擴充檔案標示」、「重新拉緊」及「資料壓縮」屬性均具有無法變更的預定值。

MPIO 磁帶屬性

MPIO 支援磁帶機會有一些額外的屬性列在 MPIO 裝置屬性下。

磁帶機的特殊檔案

有數個特殊檔案是與作業系統已知的每一個磁帶機相關聯。

使用 `rmt` 特殊檔案，在磁帶的檔案中執行寫入與讀取。 這些特殊檔案為 `/dev/rmt*`、`/dev/rmt*.1`、`/dev/rmt*.2` 一直到 `/dev/rmt*.7`。`rmt*` 是磁帶機的邏輯名稱，如 `rmt0`、`rmt1` 等等。

選取其中一個與磁帶機相關的特殊檔案，您可以選擇如何執行與磁帶機相關的 I/O 作業。

項目	說明
密度	您可以選取是否要使用磁帶機「密度設定 #1」或使用磁帶機「密度設定 #2」執行寫入。這些密度設定的值是磁帶機屬性的一部分。因為通常會將「密度設定 #1」設為磁帶機可能的最高密度，而將「密度設定 #2」設為磁帶機可能的次高密度，所以使用「密度設定 #1」的特殊檔案有時稱為高密度，而使用「密度設定 #2」的特殊檔案有時稱為低密度，但此觀點並非永遠正確。從磁帶中讀取時，密度設定會被忽略。
結束時倒轉	您可以選取是否要在參照磁帶機的特殊檔案結束時倒轉磁帶。如果已選取結束時倒轉，則當檔案結束時，磁帶會定位於磁帶的開頭。
開啟時重新拉緊	您可以選取是否要在檔案開啟時重新拉緊磁帶。重新拉緊表示轉到磁帶的結尾，然後再倒轉到磁帶的開頭，以減少錯誤。如果選取開啟時重新拉緊，則磁帶會定位於磁帶的開頭作為開啟處理程序的一部分。

下表顯示 `rmt` 特殊檔案的名稱及其性質。

特殊檔案	關閉時倒轉	開啟時重新拉緊	密度設定
/dev/rmt*	是	否	#1
/dev/rmt*.1	否	否	#1
/dev/rmt*.2	是	是	#1
/dev/rmt*.3	否	是	#1
/dev/rmt*.4	是	否	#2
/dev/rmt*.5	否	否	#2
/dev/rmt*.6	是	是	#2
/dev/rmt*.7	否	是	#2

假設您要在磁帶機 rmt2 中的磁帶上寫入三個檔案。第一個檔案位於磁帶的開頭，第二個檔案在第一個檔案之後，第三個檔案在第二個檔案之後。然後，假設您想要磁帶機的「密度設定 #1」。依下列給定的順序所列出的特殊檔案可用來寫入磁帶。

1. /dev/rmt2.3
2. /dev/rmt2.1
3. /dev/rmt2

選擇這些特定的特殊檔案的原因是：

- 選擇 /dev/rmt2.3 作為第一個檔案，因為此檔案具有確保第一個檔案位於磁帶開頭的「開啟時重新拉緊」。不選擇「關閉時倒轉」，因為下一個 I/O 作業會從此檔案結束的地方開始。如果第一個檔案開啟時磁帶已位於開頭，則將 /dev/rmt2.1 檔案用作第一個檔案會更快，因為免除了重新拉緊磁帶的時間。
- 選擇 /dev/rmt2.1 作為第二個檔案，因為此檔案既未選擇「開啟時重新拉緊」，亦未選擇「關閉時倒轉」。無論當檔案開啟時還是關閉時，都不會跳至磁帶的開頭。
- 選擇 /dev/rmt2 作為第三個檔案（亦為最後一個檔案），因為不需要「開啟時重新拉緊」，原因是第三個檔案要跟在第二個檔案之後。選取「關閉時倒轉」，因為未計劃在磁帶上的第三個檔案之後再寫入任何內容。下次使用磁帶時將從磁帶的開頭開始。

除了藉由選擇特定的 `rmt` 特殊檔案來控制磁帶作業外，您亦可使用 **`tctl`** 指令來控制磁帶作業。

USB 裝置支援

AIX 作業系統支援「通用序列匯流排 (USB)」裝置。

AIX 作業系統支援下列類型的協力廠商 USB 裝置及 IBM USB 裝置：

- 快閃記憶體隨身碟
- 磁碟機
- 光碟裝置（藍光、DVD 及 CD-ROM）
- 磁帶機
- 鍵盤
- 滑鼠
- 喇叭

註： AIX 作業系統不支援 USB 印表機。

AIX 作業系統可能無法辨識部分協力廠商 USB 裝置。例如，來自 Power Systems USB 埠的電流可能不足。因此，AIX 作業系統無法支援所有可能的協力廠商 USB 裝置。

USB 快閃記憶體隨身碟支援

從具有「技術層次」5300-09 的 AIX 5.3 和具有「技術層次」6100-02 的 AIX 6.1 開始，即支援「通用序列匯流排 (USB)」快閃記憶體隨身碟。

對這些裝置的支援包括在下列裝置套件中：

```
devices.usbif.08025002
```

會根據產業標準 OEM USB 快閃記憶體隨身碟的樣本，驗證 AIX 對 USB 快閃記憶體隨身碟的支援。 AIX USB 主機控制器的裝置驅動程式支援 USB 2.0。USB 快閃記憶體隨身碟是以邏輯名稱（例如，usbms0 和 usbms1）配置，因此它們會呈現原始和區塊特殊檔案。例如，usbms0 的原始特殊檔案是 /dev/rusbms0，而區塊特殊檔案是 /dev/usbms0。在具有 5300-11「技術層次」的 AIX 5.3 版及具有 6100-04「技術層次」的 AIX 6.1 版之前，USB 快閃記憶體隨身碟是配置為 /dev/flashdrive0。

在這些磁碟機上支援「國際標準組織 (ISO)」檔案系統（唯讀 ISO 9660）。您可以使用 **tar** 指令、**cpio** 指令或使用備份或還原保存檔，在磁碟機上建立系統備份。您也可以使用 **dd** 指令，將 ISO 映像檔新增至磁碟機。

AIX 作業系統不支援 USB 快閃記憶體隨身碟隨插即用功能。若要使快閃記憶體隨身碟可供 AIX 使用者使用，root 使用者必須將隨身碟連接至系統 USB 埠，並執行下列指令：

```
cfgmgr -l usb0
```

小心： 當您從埠中移除快閃記憶體隨身碟時要小心。如果在您移除隨身碟之前未適當地關閉或卸載它們，則隨身碟上的資料可能會毀損。

移除隨身碟之後，它們在「物件資料管理程式 (ODM)」中仍維持可用狀態，直到 root 使用者執行下列指令為止：

```
rmdev -l usbmsn
```

當隨身碟處於可用狀態時，您可以將它重新連接至系統，就可以重新裝載或重新開啟它。如果隨身碟與系統 USB 埠斷線時，仍然對使用者開啟，則要等到使用者關閉再重新開啟隨身碟之後，才能重複使用該隨身碟。

USB 藍光光碟機唯讀支援

AIX 6.1 版（含 6100-06 技術層次） 以及更新版本可辨識及配置以 USB 連接的藍光光碟機。

此特性包括在下列裝置套件中：

```
devices.usbif.08025002
```

會根據產業標準原始設備製造商 (OEM) USB 藍光光碟機樣本，來驗證 AIX 作業系統讀取藍光媒體的功能。

USB 藍光光碟機是使用邏輯名稱（例如，cd0 和 cd1）加以配置。這些光碟機代表原始和區塊特殊檔案。例如，cd0 的原始特殊檔案是 /dev/rcd0，而區塊特殊檔案則是 /dev/cd0。

對於「國際標準組織 (ISO)」檔案系統（唯讀 ISO 9660）、「通用磁碟格式 (UDF)」檔案系統（2.01 版或更舊版本）及標準光學媒體存取指令（例如，**dd** 和 **tar**），均提供唯讀功能。

AIX 作業系統不支援對 CD、DVD 或 USB 藍光光碟機所呈現之藍光媒體的寫入作業。雖然不會防止寫入作業（如果磁碟機是可寫入的），但 IBM 對於寫入作業期間發現的任何問題並不提供支援。

AIX 作業系統不支援 USB 藍光光碟機隨插即用功能。若要使 USB 藍光光碟機可供 AIX 使用者使用，root 使用者必須將光碟機連接至系統的 USB 埠，並執行下列指令：

```
cfgmgr -l usb0
```

移除光碟機之後，該光碟機在「物件資料管理程式 (ODM)」資料庫中仍維持可用狀態，直到 root 使用者執行下列指令為止：

```
rmdev -l cd n
```

當光碟機處於可用狀態時，您可以將它重新連接至系統。如果光碟機與系統 USB 埠斷線時，仍然對使用者開啟，則要等到您關閉再重新開啟光碟機之後，才能使用該光碟機。

快取儲存體資料

在快取裝置上支援儲存體資料的伺服器端快取。

快取裝置可以是下列其中一種類型的裝置：

- 伺服器連接的快閃記憶體裝置，例如伺服器中的內建固態硬碟 (SSD)。
- 使用「序列連接 SCSI (SAS)」控制器直接連接至伺服器的快閃記憶體裝置。
- 儲存區域網路 (SAN) 中的快閃記憶體資源。

AIX 作業系統必須將裝置識別為快閃裝置，它才能用作快取裝置。AIX 作業系統會使用「SCSI 區塊裝置性質重要產品資料 (VPD)」頁面中的**媒體旋轉率**欄位，來判斷裝置是否為快閃裝置。 如果裝置不支援頁面或在**媒體旋轉率**欄位中放置非 0001h 非旋轉媒體的值，則裝置無法作為快取裝置使用。

儲存體資料快取概念

當工作量正在執行中時，您可以動態地快取儲存體資料（啟動或停止快取）。 不需要卸下工作量使其成為非作用中狀態，就能執行快取作業。

下列術語用於說明快取概念：

快取裝置
快取裝置是用於快取的固態硬碟 (SSD) 或快閃記憶體磁碟。

快取儲存區
快取儲存區是僅用於儲存體快取的一個快取裝置群組。

快取分割區
快取分割區是從快取儲存區建立的一個邏輯快取裝置。

目標裝置
目標裝置是要進行快取的儲存裝置。

單一快取分割區可用來快取一個以上的目標裝置。快取目標裝置時，裝置區塊的所有讀取要求都會遞送至快取軟體。 如果在快取中發現特定的區塊，則會從快取裝置中處理 I/O 要求。如果在快取中找不到所要求的區塊，或者這是一個寫入要求，則 I/O 要求會傳回目標裝置。

儲存體資料快取的優點

儲存體資料的伺服器端快取可以提高虛擬化密度，尤其是在儲存體子系統處於壅塞狀態時。

儲存體資料快取具有下列優點：

延遲時間
分析性和交易式工作量減少了查詢/回應時間，因為固態硬碟 (SSD) 儲存體的延遲時間較低。 如果您使用伺服器端快取，則交易式工作量的平均延遲時間可以減少一半。

傳輸量
線上交易處理 (OLTP) 工作量的交易速率較高，因為 SSD 儲存體提供了更大的傳輸量。

寫入傳輸量
在儲存區域網路 (SAN) 處於壅塞狀態的環境中，用作快取的快閃記憶體裝置可以卸載絕大部分的讀取要求百分比。 卸載讀取要求時，SAN 可以具有更大的寫入傳輸量，並且可以有效地為大量用戶端與主機提供服務。

較小的記憶體覆蓋區
如果配置了快閃記憶體快取裝置，則即使利用較低的記憶體覆蓋區，也可以執行某些工作量。

儲存體資料快取的限制

請確保您瞭解使用快取功能時的限制及其他配置需求。 您還必須考量必須快取之目標裝置的應用程式限制。

請考量下列快取儲存體資料的限制：

- 快取軟體會配置為唯讀快取，這表示只會從快閃記憶體固態硬碟 (SSD) 中處理讀取要求。 所有寫入要求都由原始儲存裝置進行處理。
- 寫入儲存裝置的資料不會自動移入快取中。如果對位於快取中的區塊執行寫入作業，則快取中的現有資料將標示為無效。 該區塊會根據區塊的存取頻率和新近程度重新出現在快取中。
- 每一個 AIX 邏輯分割區 (LPAR) 都需要額外的記憶體，因為快取軟體會管理每一個讀取區塊上的 meta 資料。 對於已啟用快取的任何 LPAR 而言，至少需要 4 GB 的記憶體。
- 快取軟體會根據本端讀取型樣將資料載入到快取中，並在本端使快取項目失效。 目標裝置不得由一個以上的 LPAR 同時共用。目標裝置不能是任何叢集儲存體（例如 Oracle Real Application Clusters (RAC)、DB2 pureScale® 及 General Parallel File System (GPFS)）的一部分。只有當存取指定一次只有單一主機從目標裝置讀取資料或寫入資料，而且只在作用中節點上啟用快取時，才可以快取屬於高可用性叢集一部分的目標裝置。
- 快取磁碟可以供應給 AIX LPAR 或 Virtual I/O Server (VIOS) LPAR。快取裝置不可共用。
- 快取軟體必須開啟目標裝置，才能截取對目標裝置的任何 I/O 要求。 如果工作量需要在啟動快取之後專門開啟目標裝置，則專用開啟作業會失敗。 在這些情況下，必須停止快取，然後在工作量啟動之後重新啟動。 專用開啟作業的範例是設定目標磁碟的實體磁區 ID (PVID)。
- 如果將磁碟用作目標裝置，則不得將該磁碟的 **reserve_policy** 屬性設為 `single_path`。
- 對目標裝置啟動快取作業時，快取引擎邏輯會將資料的升級延遲到快取中。 若要確保先完成目標裝置上所有未完成的 I/O 作業（其在啟動快取作業之前發出），然後再啟動快取作業，此延遲是必要項目。 延遲的確切時間是在內部根據目標磁碟的可用路徑數及 **rw_timeout** 屬性（如果有的話）進行計算。 如果使用者定義的時間必須置換內部計算的時間，則您可以將位於 `/etc/environment` 檔案中的 *DEFAULT_IO_DRAIN_TIMEOUT_PD* 環境變數設定為自訂逾時值（秒）。

儲存體資料快取的元件

快取軟體包含快取管理元件及快取引擎元件。

快取管理

您可以使用 **cache_mgt** 指令來管理儲存體資料快取，該指令可在 AIX 作業系統及 Virtual I/O Server (VIOS) 上使用。您可以使用 **cache_mgt** 指令來執行下列作業：

- 建立快取儲存區並對其進行分割。
- 將快取分割區指派給目標裝置或 AIX 邏輯分割區 (LPAR)，作為「虛擬小型電腦系統介面 (vSCSI)」裝置。
- 啟動和停止快取作業。

快取引擎

快取引擎是快取軟體中最重要的組件。快取引擎可決定儲存體中必須快取的區塊，以及必須從快取還是主要儲存體擷取該資料。

快取演算法是根據「讀取時移入」機制，以資料填入具有空間地區（接近其他最近讀取的區塊）的快取。當快取是空的時，快取演算法在快取中填入資料的速度更快。

將監視快取中的所有區塊以檢查它們的讀取頻率，並產生熱圖。 熱圖會考量存取的頻率和最近狀態。完全移入快取時，只有在新區塊的熱度高於快取中最冷的區塊時，才會在快取中新增項目。 會從快取中移除最冷的區塊，並新增項目。

主動移入可確保暖機時間較短，進而讓快取只要啟用就可立即生效。 根據熱圖的移除原則會確保快取是動態的，而且會調整以變更工作量型樣。

配置儲存體資料快取

在 AIX 作業系統中，幾種不同配置都支援快閃記憶體裝置的伺服器端快取。這些配置的不同之處在於向 AIX 邏輯分割區 (LPAR) 供應快取裝置的方式。

在 AIX 作業系統中，伺服器端快取支援下列模式：

· 專用模式

· 虛擬模式

· N_Port ID 虛擬化 (NPIV) 模式

在專用模式中快取儲存體資料

在專用模式中，快取裝置會直接供應給 AIX 邏輯分割區 (LPAR)。

您必須建立快取儲存區，然後在快取裝置上建立快取分割區。一個專用的快取裝置只能建立一個快取分割區。您可以使用快取分割區在 AIX LPAR 上快取任意數目的目標裝置。因為快取裝置專用於此 LPAR，所以 LPAR 不是行動式分割區。如果需要將 LPAR 移轉到另一個伺服器，您必須先手動停止快取並取消配置快取裝置，然後再進行移轉。

下圖顯示 AIX LPAR 上專用快取裝置的快取配置範例。

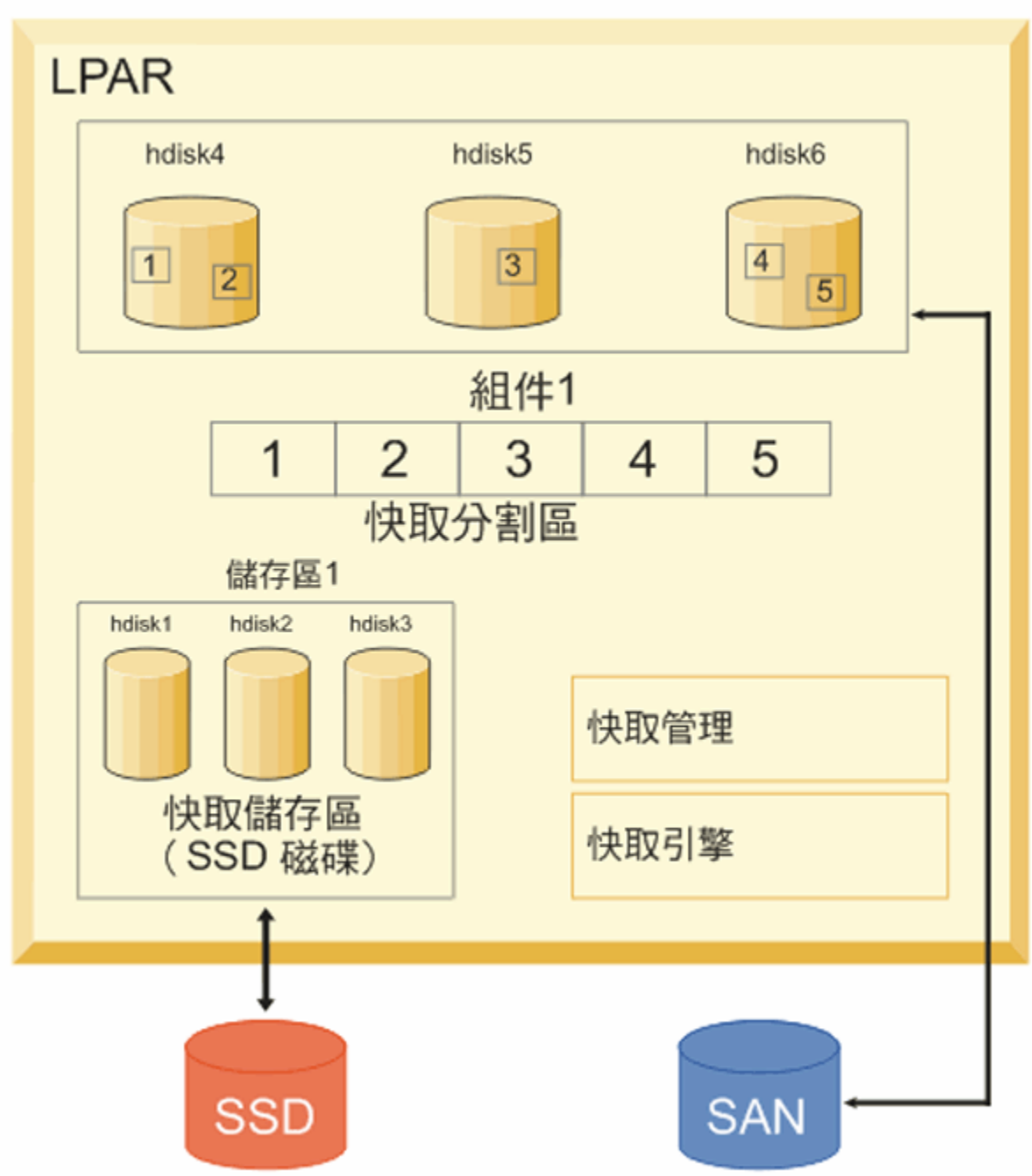


圖 *15.* 儲存體資料快取：專用快取裝置的配置

假定快取裝置為 hdisk1、hdisk2 及 hdisk3，而目標裝置為 hdisk4、hdisk5 及 hdisk6。 若要啟動及監視目標裝置的快取，請完成下列步驟：

1. 在 SSD 儲存體上建立快取儲存區。

```
# cache_mgt pool create -d hdisk1,hdisk2,hdisk3 -p cmpool0
```

2. 從快取分割區中建立大小為 80 MB 的快取分割區。

```
# cache_mgt partition create -p cmpool0 -s 80M -P part1
```

3. 將快取分割區指派給您要快取的目標磁碟。

```
# cache_mgt partition assign -t hdisk4 -P part1
# cache_mgt partition assign -t hdisk5 -P part1
# cache_mgt partition assign -t hdisk6 -P part1
```

4. 開始快取目標裝置。

```
# cache_mgt cache start -t hdisk4
# cache_mgt cache start -t hdisk5
# cache_mgt cache start -t hdisk6
```

5. 監視器快取命中的相關統計資料。

```
# cache_mgt monitor get -h -s
```

在虛擬模式中快取儲存體資料

在虛擬模式中，快取裝置會指派給 Virtual I/O Server (VIOS)。

在虛擬模式中，將於 VIOS 上建立快取儲存區。然後，會在 VIOS 上將快取儲存區分割成多個分割區。 每一個快取分割區都可以指派給一個虛擬主機 (vhost) 配接器。在 AIX 邏輯分割區 (LPAR) 上探索快取分割區時，快取分割區可用於快取目標裝置。 因為快取裝置是虛擬裝置，所以快取分割區可以移轉到另一個伺服器。在移轉之前，將於來源 LPAR 上自動停止快取。 如果目標 VIOS 上已安裝快取軟體且有快取儲存區可供使用，則作為移轉作業的一部分，將在目標 VIOS 上動態建立一個相同大小的快取分割區。 在移轉期間，快取分割區可供 LPAR 使用。當移轉完成時，將在目的地 LPAR 上自動啟動快取。 在此情況下，快取會以空白（未移入）狀態啟動。

下圖顯示虛擬模式中的快取配置範例，其中快取裝置位在 VIOS LPAR 上，而目標裝置位在 AIX LPAR 上。

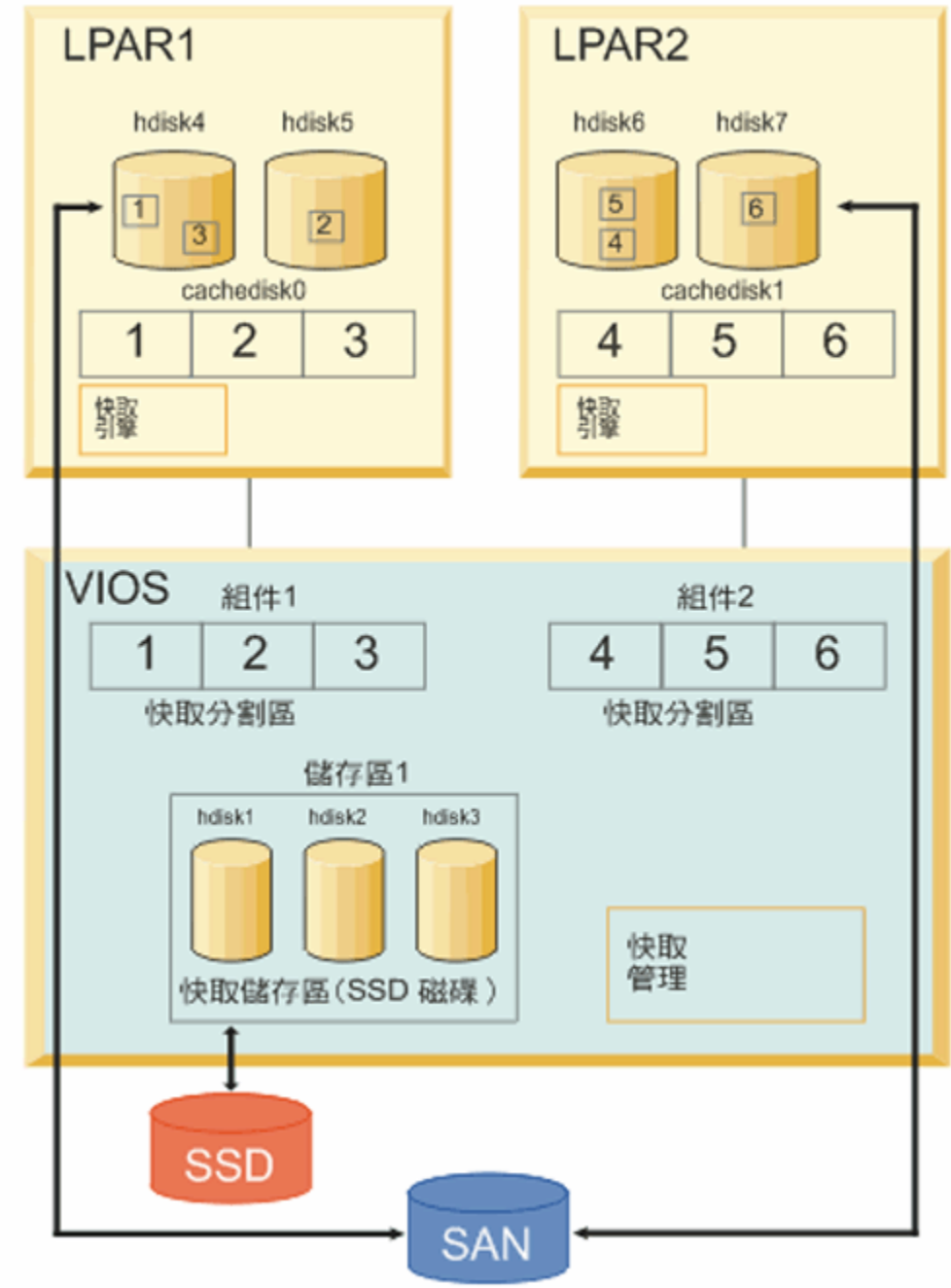


圖 16. 儲存體資料快取：虛擬快取裝置的配置

假定快取裝置為 hdisk1、hdisk2 及 hdisk3（在 VIOS LPAR 上），而目標裝置為 hdisk4 及 hdisk5（在 AIX LPAR 上）。若要啟動及監視目標裝置的快取，請完成下列程序：

1. 在 VIOS LPAR 中，使用 SSD 儲存體建立一個快取儲存區。

```
# cache_mgt pool create -d hdisk1,hdisk2,hdisk3 -p cmpool0
```

2. 在 VIOS LPAR 中，從快取儲存區建立一個大小為 80 MB 的快取分割區。

```
# cache_mgt partition create -p cmpool0 -s 80M -P part1
```

3. 在 VIOS LPAR 中，將該分割區指派給虛擬主機配接器。

```
# cache_mgt partition assign -P part1 -v vhost0
```

4. 在 AIX LPAR 中，將快取分割區指派給目標裝置。

```
# cache_mgt partition assign -t hdisk4 -P cachedisk0
# cache_mgt partition assign -t hdisk5 -P cachedisk0
```

5. 在 AIX LPAR 中，開始快取目標裝置。

```
# cache_mgt cache start -t hdisk4
# cache_mgt cache start -t hdisk5
```

6. 在 AIX LPAR 中，監視快取命中的相關統計資料。

```
# cache_mgt monitor get -h -s
```

在 NPIV 模式中快取儲存體資料

在此模式中，快取裝置可作為 AIX 邏輯分割區 (LPAR) 上的虛擬光纖通道（N_Port ID 虛擬化）裝置使用。

您必須建立快取儲存區，然後在 AIX LPAR 上建立快取分割區。只能在 AIX LPAR 上建立一個快取分割區。您可以使用快取分割區在 AIX LPAR 上快取任意數目的目標裝置。因為可以從儲存區域網路 (SAN) 取得快取裝置，所以 LPAR 可以移轉到另一個伺服器。在目的地系統上，必須可以看見快取裝置。 快取作業可以在移轉處理程序期間繼續執行，而快取將在移轉完成之後移入。

在 NPIV 模式中快取目標裝置與在專用模式中快取儲存體資料相同，不同處在於可以從 SAN 取得快取裝置。不過，快取目標裝置的程序與在專用模式中快取儲存體資料的程序相同。

管理儲存體資料快取

儘管已配置快取，但是快取需求可能會隨著時間而變更。您可能想要新增要快取的工作量。 為了滿足變更需求，可以使用其他快取裝置來擴充快取儲存區、可以在現有儲存區中建立新的快取分割區，也可以增加現有分割區的大小。

您可以使用下列範例來管理快取配置：

· 若要將快取裝置新增至儲存區，請輸入下列指令：

```
# cache_mgt pool extend -p pool1 -d hdisk4 -f
```

如果磁碟 (hdisk4) 包含現有磁區群組，則 **-f** 旗標會置換該磁碟的任何現有用法。

· 若要針對大小為 100 MB 的新工作量建立分割區，請輸入下列指令：

```
# cache_mgt partition create -p pool1 -s 100M -P part2
```

· 若要將現有分割區的大小增加 20 MB，請輸入下列指令：

```
# cache_mgt  partition extend -p part1 -s 20M
```

高可用性考量

如果已快取的目標裝置是高可用性叢集中受管理之資源群組的一部分，則必須適當地規劃失效接手作業。

一次只能在一個節點上啟動快取。在起始任何失效接手事件之前，您必須先確保已在原始系統上停用了快取作業。 在完成失效接手到替代系統的作業之後，您必須手動啟用快取軟體。

若要啟用快取軟體，請完成下列步驟：

1. 在原始系統上停止快取。

```
# cache_mgt cache stop -t hdisk2
```

2. 在完成故障回復之後，於新系統上啟動快取。

```
# cache_mgt cache start -t hdisk2
```

監視快取統計資料

每個目標裝置的快取統計資料可以使用 **cache_mgt monitor get** 指令來顯示。

例如，如果 hdisk1 是快取的唯一目標裝置，則 **cache_mgt** 指令的輸出可能會類似於下列範例：

```
# cache_mgt monitor get -h -s
ETS Device I/O Statistics -- hdisk1
Start time of Statistics -- Mon Mar 27 07:10:41 2017
------------------------------------------------
 Read Count:                              152125803
 Write Count:                              79353626
 Read Hit Count:                                871
 Partial Read Hit Count:                         63
 Read Bytes Xfer:                     10963365477376
 Write Bytes Xfer:                     4506245999616
 Read Hit Bytes Xfer:                        48398336
 Partial Read Hit Bytes Xfer:                 5768192
 Promote Read Count:                       2033078104
 Promote Read Bytes Xfer:            532959226494976
```

cache_mgt 指令提供下列快取度量值：

Read Count
所有應用程式發出給目標裝置的所有讀取作業總數。若資料在快取裝置中可用，此值也包括對於快取裝置的讀取作業。此值是個別讀取要求的總數，並不表示讀取要求的大小（位元組計數）。

Write Count
發出給目標裝置的寫入作業總數。此值是個別寫入要求的總數，並不表示寫入要求的大小（位元組計數）。

Read Hit Count
發出給目標裝置且是完整讀取命中 的讀取作業總數。完整讀取命中 是讀取要求完全由快取裝置實現的讀取作業實例。**Read Hit Count** 值是個別讀取命中要求的總數，並不表示讀取要求的大小（位元組計數）。此值包含在 **Read Count** 值中。

Partial Read Hit Count
發出給目標裝置且是局部讀取命中 的讀取作業總數。局部讀取命中 是讀取要求部分由快取裝置實現的讀取作業實例。剩餘的資料（在快取裝置中無法使用）必須從目標裝置獲得。**Partial Read Hit Count** 值是個別讀取要求的總計數，並不表示讀取要求的大小（位元組計數）。此值包含在 **Read Count** 值中。

Read Bytes Xfer
從應用程式發出給目標裝置的所有讀取要求中，傳輸的位元組總數。此值代表完整讀取命中資料、局部讀取命中 資料，以及從目標裝置取得的任何資料，三者的總位元組計數。

Write Bytes Xfer
從應用程式發出給目標裝置的所有寫入要求中，傳輸的位元組總數。

Read Hit Bytes Xfer
在完整讀取命中 實例期間讀取的位元組總數。

Partial Read Hit Bytes Xfer
在局部讀取命中 實例期間讀取的位元組總數。

Promote Read Count
當資料移入到快取裝置時，發出給目標裝置的讀取指令總數。此值並不指出資料移入到快取裝置時的實例數，因為如果對目標磁碟的最大資料傳輸大小小於要求大小，將資料移入到快取裝置的單一要求可能會分成多個讀取作業。

Promote Read Bytes Xfer
當資料移入到快取裝置時，從目標裝置讀取的位元組總數。

登入名稱、系統 ID 及密碼

作業系統必須知道您的身分，以提供正確的環境給您。

若要向作業系統識別您自己，請鍵入您的登入名稱（亦稱為您的使用者 ID 或使用者名稱）和密碼來登入。密碼是安全保護的一種方式。只知道您的登入名稱並不能登入您的系統，除非也知道您的密碼。

如果您系統設定成多使用者系統，則每一位授權使用者在系統上皆具有一個帳戶、密碼及登入名稱。作業系統會追蹤每一個使用者已使用的資源。這稱為系統帳戶。 每一個使用者在系統的儲存體空間內將擁有一個專用區域，稱為檔案系統。雖然系統上有成千上萬個檔案，當您登入時，檔案系統只呈現您的檔案。

在一個系統上可以具有一個以上有效的登入名稱。如果您要改變登入名稱，不必登出系統。您可同時在不同 shell 中使用不同的登入名稱，或在相同的 shell 中連續使用不同的登入名稱，但不必登出系統。此外如果您的系統與其他系統連線，是網路的一部分，則您可登入擁有登入名稱的其他系統。此動作稱為遠端登入。

當您在作業系統上工作完成時，請登出以確保檔案和資料的安全。

登入作業系統

若要使用作業系統，您的系統必須是在執行中，而且您必須登入系統中。當您登入作業系統時，您要向系統識別您的身分，讓系統設定您的環境。

您的系統可設定成讓您僅可於特定的某些天及某些時間才能登入。如果您嘗試在不允許的時間登入，將會拒絕您存取。您的系統管理者可以驗證您的登入時間。

請在登入提示下登入。當您登入作業系統時，您會自動被置於您的起始目錄中（亦稱為您的登入目錄）。

在開啟您的系統之後，請登入系統來啟動階段作業。

1. 請在 **login:** 提示之後輸入您的登入名稱：

   ```
   login: LoginName
   ```

   例如，如果您的登入名稱是 denise：

   ```
   login: denise
   ```

2. 如果顯示 **password:** 提示，請鍵入您的密碼。
   (當您鍵入密碼時，畫面上不會出現您的密碼。)

   ```
   password: [your password]
   ```

如果沒有顯示密碼提示，表示您尚未定義密碼；您可以開始在作業系統中工作。

如果您的機器沒有打開，請在登入之前先執行下列動作：

1. 開啟每一個附屬裝置的電源開關。
2. 開啟電源開關來啟動主機 (**I**)。
3. 查看三位數顯示畫面。當自我測試完成無誤時，三位數顯示畫面呈現空白。

若發生錯誤，則三位數字碼會持續出現，而且主機隨即會停止。請洽您的系統管理者，以取得關於錯誤碼及復原的相關資訊。

當自我測試順利完成時，您的畫面上會顯示類似下列的登入提示：

```
login:
```

在您登入之後，根據您作業系統的設定方式，系統會以指令行介面 (shell) 或圖形介面（例如，AIXwindows 或「一般桌上管理系統環境 (CDE)」) 啟動。

若您對於密碼或使用者名稱的配置有任何問題，請洽詢您的系統管理者。

登入一次以上（login 指令）

如果您手上有好幾個專案要處理，並且想要個別維護帳戶，您可以同時擁有好幾個登入帳戶。作法是使用相同或不同的登入名稱來登入您的系統。

註 : 每一個系統中可同時作用的登入名稱，皆有一定的限制數量。此數目決定於您的授權合約，並且隨著各種安裝而異。

例如，如果您已經以 denise1 身分登入，而您的另一個登入名稱是 denise2，請在提示時鍵入：

```
login denise2
```

如果顯示 **password:** 提示，請鍵入您的密碼。（當您鍵入密碼時，畫面上不會出現您的密碼。）您現在有兩個登入在您的系統上執行。

請參閱 *Commands Reference, Volume 3* 中的 **login** 指令，以取得完整語法。

成為系統上的另一個使用者（su 指令）

若要變更與階段作業關聯的使用者 ID（如果您知道該使用者的登入名稱），可使用 **su**（切換使用者）指令。

例如，如果您想要切換成為使用者 joyce，請在提示時鍵入：

```
su joyce
```

如果顯示 **password:** 提示，請鍵入使用者 joyce 的密碼。現在您的使用者 ID 是 joyce。如果您不知道密碼，則要求會遭拒絕。

若要驗證您的使用者 ID 是 joyce，請使用 **id** 指令。

抑制登入訊息

在順利完成登入之後，**login** 指令會顯示此使用者上次成功和失敗登入嘗試的日期與時間訊息，以及自從上次變更鑑別資訊之後（通常是密碼）的失敗登入嘗試總數。您可以將 .hushlogin 檔案併入您的起始目錄，以抑制這些訊息的出現。

在起始目錄的提示下，請鍵入下列指令：

```
touch .hushlogin
```

如果名為 .hushlogin 的檔案不存在，則 **touch** 指令會建立該檔案（空的）。您下次登入時，將會抑制所有的登入訊息。您可指示系統只保留當天的訊息，並暫停其他登入訊息。

登出作業系統（exit 及 logout 指令）

若要登出作業系統，請在系統提示下執行下列其中一項。

按下檔案結尾控制鍵順序（Ctrl-D 鍵）。

或

輸入 exit。

或

輸入 logout。

在您登出之後，系統會顯示 **login:** 提示。

顯示使用者 ID

若要顯示指定使用者的系統識別 (ID)，請使用 **id** 指令。系統 ID 是用來向系統識別使用者和使用者群組的號碼。

id 指令將顯示下列資訊（若有的話）：

- 使用者名稱與實際的使用者 ID
- 使用者群組的名稱與實際的群組 ID
- 使用者增補群組的名稱及增補群組 ID（若有的話）

例如，在提示時鍵入：

```
id
```

系統會顯示如下的資訊：

```
uid=1544(sah) gid=300(build) euid=0(root) egid=9(printq) groups=0(system),10(audit)
```

在本例中，使用者具有下列名稱和 ID 號碼： 使用者名稱是 sah，其 ID 號碼是 1544； 主群組名稱是 build，其 ID 號碼是 300； 有效的使用者名稱是 root，其 ID 號碼是 0； 有效的群組名稱是 printq，其 ID 號碼是 9； 兩個增補群組名稱是 system 及 audit，其 ID 號碼分別是 0 及 10。

例如，在提示時鍵入：

```
id denise
```

系統會顯示如下的資訊：

```
uid=2988(denise) gid=1(staff)
```

在本例中，使用者 denise 所擁有的 ID 號碼是 2988，而且僅擁有一個主群組，名稱是 staff，其 ID 號碼是 1。

請參閱 *Commands Reference, Volume 3* 中的 **id** 指令，以取得完整語法。

顯示您的登入名稱（whoami 及 logname 指令）

當您有一個以上的並行登入時，很容易迷失登入名稱的路徑，尤其是您正在使用的登入名稱。這時您可以使用 **whoami** 及 **logname** 指令，來顯示此資訊。

使用 whoami 指令

若要判斷正在使用的登入名稱，請在提示時鍵入：

```
whoami
```

系統會顯示如下的資訊：

```
denise
```

在本例中，所使用的登入名稱是 denise。

請參閱 *Commands Reference, Volume 6* 中的 **whoami** 指令，以取得完整語法。

使用 who am i 指令

who am i 指令是 **who** 指令的變體，可讓您顯示登入名稱、終端機名稱及登入時間。在提示時鍵入：

```
who am i
```

系統會顯示如下的資訊：

```
denise   pts/0   Jun 21 07:53
```

在本例中，登入名稱是 denise，終端機名稱是 pts/0，而使用者是在 6 月 21 日的 7:53 a.m. 登入。

請參閱 *Commands Reference, Volume 6* 中的 **who** 指令，以取得完整語法。

使用 logname 指令

who 指令的另一個變體，**logname** 指令會顯示與 **who** 指令相同的資訊。

在提示時鍵入：

```
logname
```

系統會顯示如下的資訊：

```
denise
```

在本例中，登入名稱是 denise。

顯示作業系統名稱（uname 指令）

若要顯示作業系統的名稱，請使用 **uname** 指令。

例如，在提示時鍵入：

```
uname
```

系統會顯示如下的資訊：

```
AIX
```

在本例中，作業系統名稱是 AIX。

請參閱 *Commands Reference, Volume 5* 中的 **uname** 指令，以取得完整語法。

顯示您的系統名稱（uname 指令）

當您在網路上，若要顯示系統的名稱，請在 **uname** 指令使用 **-n** 旗標。您的系統名稱可讓網路識別您的系統，與您的登入 ID 並不相同。

例如，在提示時鍵入：

```
uname -n
```

系統會顯示如下的資訊：

```
barnard
```

在此範例中，系統名稱是 barnard。

請參閱 *Commands Reference, Volume 5* 中的 **uname** 指令，以取得完整語法。

顯示所有已登入的使用者

若要顯示目前登入本端系統的所有使用者之資訊，請使用 **who** 指令。

將會顯示下列資訊：登入名稱、系統名稱及登入的日期和時間。

註： 此指令只識別本端節點上已登入的使用者。

若要顯示正在使用本端系統節點的使用者資訊，請鍵入：

```
who
```

系統會顯示如下的資訊：

```
joe   lft/0 Jun 8 08:34
denise   pts/1 Jun 8 07:07
```

在本例中，使用者 joe，位於終端機 lft/0 中，登入時間為 6 月 8 日的 8:34 a.m.。

請參閱 **who** 指令。

密碼

唯一密碼可為檔案提供某個程度的系統安全。

您的系統會在每一個帳戶上建立一個相關的密碼。在電腦系統中，安全機制是很重要的一個部分，因為它擋住了未授權的使用者，讓他們無法進入系統，擅改其他使用者的檔案。安全機制也讓某些使用者，對於他們能夠使用的指令及能夠存取的檔案，擁有專用權。基於保護的目的，有些系統管理者只會讓使用者存取某些特定的指令或檔案。

密碼準則

您應該只有一個唯一的密碼。密碼不可共用。 密碼應該和任何其他公司資產一樣，受到嚴密保護。當建立密碼時，請務必讓它們很不容易猜中，但也不能太困難，造成必須把它們寫下才能記住的情況。

使用難解的密碼，可以保護使用者 ID 的安全。以個人資訊（如名稱或生日）為根據的密碼是不好的密碼。即便是一般用字，往往也很容易被猜中。

好的密碼至少有 6 個字元，其中包括不是字母的字元。奇怪的文字組合和故意的拼錯的字也是很好的選擇。

註： 如果您的密碼不好記，而必須將它寫下來，就不是好密碼。

選取密碼時，請遵循下列指引：

- 請勿使用您的使用者 ID 來作為密碼。不要以反轉、重複或任何其他修改方式來使用它。
- 請勿重覆使用密碼。系統可設定成拒絕重覆使用相同的密碼。
- 請勿使用任何人的名字來作為您的密碼。
- 請勿使用可在線上併字檢查字典中找到的文字來作為密碼。
- 請勿使用小於六個字元的密碼。
- 請勿使用猥褻的文字，如果人家要猜測密碼，可能會最先想到它們。
- 使用容易記得的密碼，這樣您就不用將它寫下來。
- 使用含有字母和數字的密碼，且字母含有大寫和小寫。
- 使用兩個字，中間夾著一個數字，來作為密碼。
- 使用唸得出來的密碼， 這樣比較容易記得。
- 請勿寫下密碼。不過如果您必須要把它們寫下時，請將它們放在安全的地方，例如上鎖的櫃子。

變更密碼（passwd 指令）

若要變更您的密碼，請使用 **passwd** 指令。

1. 在提示時鍵入：

   ```
   passwd
   ```

   如果您還沒有密碼，請略過步驟 2。
2. 會顯示下列提示：

   ```
   Changing password for
   UserID
   UserID's Old password:
   ```

   這個要求會讓未獲授權的使用者，無法在您離開系統時，擅改您的密碼。輸入您的現行密碼，然後按 Enter 鍵。
3. 會顯示下列提示：

   ```
   UserID's New password:
   ```

   輸入您想要的新密碼，然後按 Enter 鍵。
4. 會顯示下列提示，要求您重新輸入您的新密碼。

   ```
   Enter the new password again:
   ```

   這個要求可防止您將密碼設定成輸入錯誤且無法重建的字串。

請參閱 *Commands Reference, Volume 4* 中的 **passwd** 指令，以取得完整語法。

密碼設為空值（passwd 指令）

如果您不要在每次登入時鍵入密碼，您可以將密碼設為空值（空白）。

若要設定密碼為空值，請鍵入：+

```
passwd
```

提示您輸入新密碼時，請按 Enter 鍵或 Ctrl-D。

passwd 指令不會重新顯示密碼項目的提示。畫面中會出現驗證空密碼的訊息。

請參閱 **passwd** 指令，以取得相關資訊及完整語法。

登入名稱、系統 ID 及密碼的指令摘要

這些指令可用於使用登入名稱、系統 ID 及密碼。

登入及登出指令

項目	說明
login	起始您的階段作業
logout	停止您所有的處理程序
shutdown	結束系統作業
su	變更與階段作業相關的使用者 ID
touch	更新檔案的存取和修正次數，或建立空的檔案

使用者及系統識別指令

項目	說明
id	顯示指定使用者的系統 ID
logname	顯示登入名稱。
uname	顯示現行作業系統的名稱
who	識別目前登入的使用者
whoami	顯示您的登入名稱

密碼指令

項目	說明
passwd	變更使用者密碼

一般桌上管理系統環境

使用一般桌上管理系統環境 (CDE)可讓您存取網路上的裝置及工具，而不需知道它們的位置在哪裡。您只要以拖放物件的方式，即可透過應用程式來交換資料。

有許多作業，在以前需要使用複雜的指令行語法，現在可以更容易地達成，而且各平台之間的操作也都更為相似。例如，您可以集中配置及分散應用程式給使用者。您也可以針對您支援的使用者，集中管理應用程式的安全性、可用性及互動性。

註： 一般桌上管理系統環境 (CDE) 1.0 說明冊、Web 型文件及硬本手冊可能會將桌面稱為「一般桌上管理系統環境」、AIXwindows 桌面、CDE 1.0 或桌面。

啟用及停用桌面自動啟動

您會發現，將您的系統設定成在系統啟動時，自動啟動「一般桌上管理系統環境」會比較方便。

您可以透過「系統管理介面工具 (SMIT)」或從指令行執行此動作。

必要條件

您必須具備 root 使用者權限，才能啟用或停用桌面自動啟動。

請參閱下表，以判定如何啟用或停用桌面自動啟動。

作業	SMIT 捷徑	指令或檔案
啟用桌面自動啟動 [1]	**smit dtconfig**	**/usr/dt/bin/dtconfig -e**

作業	SMIT 捷徑	指令或檔案
停用桌面自動啟動 [1]	**smit dtconfig**	**/usr/dt/bin/dtconfig -d**

註 : 完成此作業後，請重新啟動系統。

手動啟動一般桌上管理系統環境

使用此程序可以手動啟動「一般桌上管理系統環境」。

1. 以 root 身分登入您的系統。
2. 在指令行中，輸入：

```
/usr/dt/bin/dtlogin -daemon
```

即顯示**桌面登入**畫面。當您登入時，會啟動桌面階段作業。

手動停止一般桌上管理系統環境

當您手動停止「登入管理程式」時，「登入管理程式」所啟動的所有 X 伺服器及桌面階段作業都會停止。

1. 開啟終端機模擬程式視窗，然後以 root 身分登入。
2. 取得「登入管理程式」的處理程序 ID，請鍵入：

```
cat /var/dt/Xpid
```

3. 停止「登入管理程式」，請鍵入：

```
kill -term process_id
```

修改桌面設定檔

當您登入桌面時，不會自動讀取 shell 環境檔（.profile 或 .login）。桌面會在您登入之前執行 X 伺服器，所以桌面的「登入管理程式」必須提供 .profile 檔案或 .login 檔案所提供的函數。

特定使用者的環境變數設定在 */Home Directory/* .dtprofile 中。此檔案的範本位於 /usr/dt/config/sys.dtprofile。請將變數及 shell 指令放在只套用於桌面的 .dtprofile。在 .dtprofile 結尾新增指令行，以納入 shell 環境檔。

全系統環境變數可能設定在「登入管理程式」配置檔中。有關配置環境變數的詳細資訊，請參閱 *Common Desktop Environment 1.0: Advanced User's and System Administrator's Guide*。

新增及移除一般桌上管理系統環境的顯示器及終端機

您可以新增及移除一般桌上管理系統環境的顯示器及終端機。

「登入管理程式」可從具有單一本端點陣圖或圖形主控台的系統來啟動。但也有可能發生許多其他狀況（請參閱下列圖表）。您可以從下列位置來啟動 一般桌上管理系統環境：

- 本端主控台
- 遠端主控台
- 在網路上的主機系統中執行的點陣圖及字元顯示器 X 終端機系統

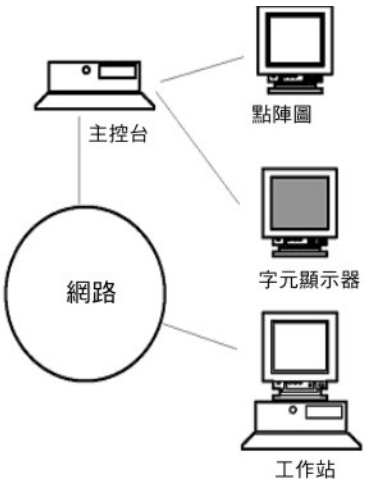


圖 *17. CDE* 介面點

X 終端機系統是由顯示裝置、鍵盤和只執行 X 伺服器的滑鼠所組成。用戶端（包括一般桌上管理系統環境）是在網路中的一或數個主機系統上執行。用戶端的輸出會被導向 X 終端機顯示器。

下列「登入管理程式」配置作業支援許多可能的配置。

· 移除本端顯示器

· 新增 ASCII 或字元顯示器終端機

若要使用工作站作為 X 終端機，請在指令行中鍵入下列指令：

```
/usr/bin/X11/X -query hostname
```

用來作為 X 終端機的 X 伺服器工作站必須：

· 支援 XDMCP 及 **-query** 指令行選項。

· 提供 xhost 許可權（在 /etc/X*.hosts 中）給終端機主機。

若要移除本端顯示器，請將其項目從 /usr/dt/config 目錄中的 Xservers 檔案中移除。

字元顯示器終端機或 *ASCII* 終端機是一種配置，在此配置中，終端機不是點陣圖裝置。

沒有點陣圖顯示器時，若要新增 ASCII 或字元顯示器主控台，請執行下列步驟：

1. 如果 /etc/dt/config/Xservers 檔案不存在，請將 /usr/dt/config/Xservers 檔案複製到 /etc/dt/config 目錄。
2. 如果必須將 Xservers 檔案複製到 /etc/dt/config，請將 /etc/dt/config/Xconfig 中的 Dtlogin.servers: 那一行變更或新增為：

   ```
   Dtlogin*servers: /etc/dt/config/Xservers
   ```

3. 將用來啟動 X 伺服器之 /etc/dt/config/Xservers 中的指令行變成備註。

   ```
   # * Local local@console /path/X :0
   ```

   此動作會停用**登入選項功能表**。
4. 讀取「登入管理程式」配置檔。

有點陣圖顯示器存在時，若要新增字元顯示器主控台，請執行下列步驟：

1. 如果 /etc/dt/config/Xservers 檔案不存在，請將 /usr/dt/config/Xservers 檔案複製到 /etc/dt/config 目錄。

2. 如果必須將 Xservers 檔案複製到 /etc/dt/config，請將 /etc/dt/config/Xconfig 中的 Dtlogin.servers: 那一行變更或新增為：

```
Dtlogin*servers: /etc/dt/config/Xservers
```

3. 編輯用來啟動 X 伺服器之 /etc/dt/config/Xservers 中的指令行，使其變為：

```
* Local local@none /path/X :0
```

4. 讀取「登入管理程式」配置檔。

顯示一般桌上管理系統環境的裝置自訂

您可以配置「一般桌上管理系統環境登入管理程式」，以在具有兩個以上之顯示裝置的系統上執行。

若系統具有多重顯示器，就必須符合下列配置需求：

· 伺服器必須在每一個顯示器上啟動。

· 每個顯示器必須配置「無 Windows 模式」。

可能必須或需要針對各顯示器來使用不同的 dtlogin 資源。

也可能必須或需要針對各顯示裝置使用不同的全系統環境變數。

在每個顯示裝置上啟動伺服器

使用這個程序，可以在每一個顯示裝置上啟動伺服器。

1. 如果 /etc/dt/config/Xservers 檔案不存在，請將 /usr/dt/config/Xservers 檔案複製到 /etc/dt/config 目錄。
2. 如果必須將 Xservers 檔案複製到 /etc/dt/config，請將 /etc/dt/config/Xconfig 中的 Dtlogin.servers: 那一行變更為：

```
Dtlogin*servers: /etc/dt/config/Xservers
```

3. 編輯 /etc/dt/config/Xservers，以在每一個顯示裝置上啟動 X 伺服器。

啟動伺服器的一般語法為：

```
DisplayName DisplayClass DisplayType [ @ite ] Command
```

只有包含相關聯的「內部終端機模擬程式 (ITE)」的顯示器，才能在「無 *Windows* 模式」下運作。「無 Windows 模式」會暫時停用顯示器的桌面，並執行 *getty* 處理程序（如果它尚未啟動的話）。getty 處理程序是 UNIX 程式，它會設定終端機類型，並且可以用於登入處理程序。

這可讓您登入並執行不可能在一般桌上管理系統環境下執行的作業。當您登出時，便會為該顯示裝置重新啟動桌面。如果 getty 處理程序尚未在顯示裝置上執行，則起始「無 Windows 模式」時，「登入管理程式」便會啟動一個 getty 處理程序。

在預設配置中，當 ite 被省略掉時，display:0 就會與 ITE (/dev/console) 產生關聯。

將別的顯示器設為 ITE

使用這個程序，可以將別的顯示器設為 ITE。

若要指定其他顯示器來作為 ITE：

· 在 ITE 顯示器上，將 ITE 設定成字元裝置。

· 在所有其他的顯示器上，將 ITE 設定成「無」。

下列範例顯示 Xserver 檔案中的項目，它們會在 sysaaa:0 上的三個本端顯示器上啟動伺服器。顯示器 :0 將是主控台 (ITE)。

```
sysaaa:0 Local local /usr/bin/X11/X :0
sysaaa:1 Local local /usr/bin/X11/X :1
sysaaa:2 Local local /usr/bin/X11/X :2
```

在主機 sysbbb 上，點陣圖顯示器 :0 不是 ITE；ITE 會與裝置 /dev/ttyi1 相關聯。Xserver 檔案中的下列項目會在兩個點陣圖顯示器上啟動伺服器，而且在 :1 上已啟用「無 Windows 模式」。

```
sysaaa:0 Local local@none /usr/bin/X11/X :0
sysaaa:1 Local local@ttyi1 /usr/bin/X11/X :1
```

在 Xconfig 設定顯示器名稱
您無法對 /etc/dt/config/Xconfig 中的顯示器名稱使用一般 hostname:0 語法。

若要在 Xconfig 中指定顯示器名稱：

· 使用底線來代替冒號。

· 在完整的主機名稱中，使用底線來代替句點。

下列範例顯示如何在 Xconfig 中設定顯示器名稱：

```
Dtlogin.claaa_0.resource: value
Dtlogin.sysaaa_prsm_ld_edu_0.resource: value
```

針對每個顯示器來使用不同的登入管理程式資源
若要針對每個顯示器使用不同的「登入管理程式」資源，請執行下列步驟：

1. 如果 /etc/dt/config/Xconfig 檔案不存在，請將 /usr/dt/config/Xconfig 檔案複製到 /etc/dt/config 目錄。
2. 編輯 /etc/dt/config/Xconfig 檔案，指定每一個顯示器的不同資源檔。
   例如：

   ```
   Dtlogin.DisplayName.resources: path/file
   ```

   其中的 *path* 是要使用的 Xresource 檔案之路徑名稱，*file* 則是要使用的 Xresource 檔案之檔名。
3. 建立 Xconfig 檔案中所指定的每個資源檔。
   語言特有的 Xresources 檔案會安裝在 /usr/dt/config/<LANG>。
4. 在每個檔案中，針對該顯示器來放置 dtlogin 資源。

下列範例顯示 Xconfig 檔案中，針對三個顯示器指定不同資源檔的指令行：

```
Dtlogin.sysaaa_0.resources: /etc/dt/config/Xresources0
Dtlogin.sysaaa_1.resources: /etc/dt/config/Xresources1
Dtlogin.sysaaa_2.resources: /etc/dt/config/Xresources2
```

針對每個顯示器來執行不同的 Script
使用這個程序，可以針對特定顯示器執行特定 Script。

1. 如果 /etc/dt/config/Xconfig 檔案不存在，請將 /usr/dt/config/Xconfig 檔案複製到 /etc/dt/config 目錄。
2. 使用 /etc/dt/config/Xconfig 中的 startup、reset 及 setup 資源，針對每一個顯示器指定不同的 Script（因此會執行這些檔案，而不執行 Xstartup、Xreset 及 Xsetup 檔案）：

   ```
   Dtlogin*DisplayName*startup: /
   path/file
   Dtlogin*DisplayName*reset: /path/
   file
   Dtlogin*DisplayName*setup: /path/file
   ```

其中 *path* 是要使用的檔案之路徑名稱，*file* 則是要使用的檔案之檔名。使用者登入之後，在啟動一般桌上管理系統環境階段作業之前，會以 root 執行 startup script。

Script /usr/dt/config/Xreset 可用來回復 Xstartup 檔案中的設定值。當使用者登出時，便會執行 Xreset 檔案。

下列範例顯示 Xconfig 檔案中，針對兩個顯示器指定不同的 Script 之指令行：

```
Dtlogin.sysaaa_0*startup:   /etc/dt/config/Xstartup0
Dtlogin.sysaaa_1*startup:    /etc/dt/config/Xstartup1
Dtlogin.sysaaa_0*setup:      /etc/dt/config/Xsetup0
Dtlogin.sysaaa_1*setup:      /etc/dt/config/Xsetup1
Dtlogin.sysaaa_0*reset:      /etc/dt/config/Xreset0
Dtlogin.sysaaa_1*reset:      /etc/dt/config/Xreset1
```

針對每個顯示器來設定不同的全系統環境變數

使用這個程序，可以自訂每一個顯示器的全系統環境變數。

若要針對每個顯示器來設定不同的全系統環境變數：

1. 如果 /etc/dt/config/Xconfig 檔案不存在，請將 /usr/dt/config/Xconfig 檔案複製到 /etc/dt/config 目錄。
2. 針對每個顯示器，在 /etc/dt/config/Xconfig 中個別設定環境資源：

   ```
   Dtlogin*DisplayName*environment: value
   ```

下列規則適用於每個顯示器的環境變數：

· 以空格或跳格字元來隔開所指派的變數。

· 請勿使用環境資源來設定 TZ 及 LANG。

· Xconfig 檔案中沒有 shell 處理程序。

下列範例顯示 Xconfig 檔案中，針對兩個顯示器設定變數的指令行：

```
Dtlogin*syshere_0*environment:EDITOR=vi SB_DISPLAY_ADDR=0xB00000
Dtlogin*syshere_1*environment: EDITOR=emacs \
         SB_DISPLAY_ADDR=0xB00000
```

搭配主機乙太網路配接卡的即時分割區行動性

使用「即時分割區行動性 (LPM)」與 IBM PowerVM® 軟體的「主機乙太網路配接卡 (HEA)」特性搭配時，您可以將 AIX LPAR 及管理的應用程式從某個實體分割區移轉至另一個實體分割區，同時將 HEA 指派給移轉分割區。

移轉期間，將從移轉分割區移除 HEA，而且移轉完成時，將不會在分割區上還原 HEA。 不過，您的網路連線將不受任何影響。

搭配 HEA 的即時分割區行動性的需求

在開始使用 LPM 與 HEA 搭配之前，您必須確定系統環境符合配置及存取需求。

分割區需求

· CEC 來源與 CEC 目標必須能夠進行分割區移轉。

· 來源 AIX LPAR 不得具有任何其設定檔中有「必要」設定的實體資源。

· 除了 HEA 以外，來源 AIX LPAR 不得具有任何實體資源。

存取需求

· 您必須對想要移轉的分割區具有 root 權限。

· 您必須對來源 HMC 與目標 HMC 具有移轉分割區所需的 hscroot 權限或相等權限。

配置需求

· HEA 不得在分割區設定檔中具有「必要」設定，但是它可以在設定檔中具有「想要」設定。

· 所有 HEA 都必須在「乙太網通道」下配置為主要配接卡。

· 「乙太網通道」中的所有主要配接卡都必須是 HEA。

· 「乙太網通道」的備份配接卡必須是「虛擬乙太網路」配接卡。

- 至少有一個「乙太網通道」必須與 HEA 一起配置為主要配接卡，以及與「虛擬乙太網路」配接卡一起配置為備份配接卡。
- 最多支援四個「乙太網通道」。
- 「乙太網通道」失效接手必須可以運作。
- 您必須驗證是否已同時設定來源系統及目標系統來進行分割區移轉。
- 如果是在兩個 HMC 之間進行移轉，則您必須在來源 HMC 與遠端 HMC 之間設定 SSH 鑑別。 在開始移轉之前，您必須在來源 HMC 上執行 **mkauthkeys** 指令。

執行搭配 HEA 的即時分割區行動性

您可以使用 SMIT 介面，執行搭配 HEA 的 LPM。

在試圖使用 LPM 與 HEA 搭配之前，請先檢閱第 191 頁的『搭配 HEA 的即時分割區行動性的需求』主題。

若要完成搭配 HEA 的 LPM 分割區移轉，請完成下列步驟：

1. 從命令提示字元中，輸入下列 SMIT 捷徑：**smitty migration**，以顯示「搭配主機乙太網路配接卡 (HEA) 的即時分割區主機」功能表。
2. 指定來源 HMC 主機名稱或 IP 位址。
3. 指定來源 HMC 使用者名稱。
4. 如果來源與目標系統是由相同 HMC 管理，請輸入 **no**，並前往步驟 5。 如果來源與目標系統是由不同 HMC 管理，請輸入 **yes**，並完成下列步驟：

   註： 在輸入 **yes** 之前，您必須在來源 HMC 上執行 **mkauthkeys** 指令。

   a) 指定遠端 HMC 主機名稱或 IP 位址。

   b) 指定來源 HMC 使用者名稱。
5. 指定來源系統的名稱。
6. 指定目標系統的名稱。
7. 指定您要移轉的分割區名稱。
8. 如果想要執行移轉，但不執行驗證，請輸入 **no**。如果只想要執行移轉驗證，請輸入 **yes**。 如果指定 **yes**，則不會執行移轉，而且只會執行驗證。

   註： 在執行分割區移轉之前，您應該執行分割區移轉驗證。
9. 請驗證所有欄位是否具有正確資訊，然後按 **Enter** 鍵以執行移轉。

   註： 將提示您兩次輸入密碼。請輸入您先前在步驟 4b 中指定之來源 HMC 使用者名稱的密碼。

在此範例中，分割區 X 是從由 HMC A 管理的 CEC C 移轉至由 HMC B 管理的 CEC D。 在 HMC A 上執行 **mkauthkeys** 指令，以在 HMC A 與 HMC B 之間進行鑑別。

對於此移轉處理程序，會在 SMIT 中指定下列值：

```
來源 HMC 主機名稱或 IP 位址：A
So 來源 HMC 使用者名稱：hscroot
在兩個 HMC 之間移轉：yes
         遠端 HMC 主機名稱或 IP 位址：B
         遠端 HMC 使用者名稱：hscroot
來源系統：C
目標系統：D
移轉分割區名稱：X
僅移轉驗證：no
```

另一個範例將是，如果分割區 X 是從由 HMC A 管理的 CEC C 移轉至也由 HMC A 管理的 CEC D。

對於此移轉處理程序，會在 SMIT 中指定下列值：

```
來源 HMC 主機名稱或 IP 位址：A
So 來源 HMC 使用者名稱：hscroot
在兩個 HMC 之間移轉：no
         遠端 HMC 主機名稱或 IP 位址：
         遠端 HMC 使用者名稱：
來源系統：C
```

```
目標系統：D
移轉分割區名稱：X
僅移轉驗證：no
```

如果移轉失敗，請遵循程序，從來源 HMC 執行「即時分割區行動性」。

使用 LPM 移轉 NIM 用戶端

若使用「即時分割區行動性 (LPM)」將一部機器從某個實體伺服器移至另一個實體伺服器，而且該機器是定義為「網路安裝管理程式 (NIM)」用戶端，則 NIM 管理者必須更新 NIM 用戶端的 *cpuid* 屬性，以便在 LPM 移轉完成之後反映新的硬體值。

若要更新 *cpuid* 屬性，請完成下列步驟：

1. 在 NIM 用戶端上，執行下列指令以獲得新的 cpuid ID：

   ```
   uname -a
   ```

2. 在 NIM 主要伺服器上，執行下列指令：

   ```
   nim -o change -a cpuid=cpuid client
   ```

   註：由於已移除 AIX 作業系統中的「叢集系統管理 (CSM)」支援，因此 `OS_install` 網路安裝程式不再支援 Linux 作業系統的安裝。

重新定位配接卡以進行 DLPAR

在重新定位配接卡以進行動態邏輯分割 (DLPAR) 作業之前，您必須先配置圖形配接卡。

請使用下列指示來動態地重新定位圖形配接卡（例如，FC 5748）：

1. 確保沒有任何現行處理程序（Desktop 和 Xserver）正在使用圖形配接卡（例如，/dev/lft0）。
2. 驗證主控台未設為 lft0。若要識別已定義或可用的相依介面（lft 或 rcm），請輸入下列指令：

   ```
   lsdev -C | grep lft
   lsdev -C | grep rcm
   ```

   若要在定義圖形配接卡之後取得母項「週邊元件交互連接 (PCI)」配接卡，請輸入下列指令：

   ```
   odmget -q name=<cortina adapter name. for instance cor0> CuDv
   ```

   此指令會提供母項和 Cortina 的相關資訊。
3. 若要移除所有的相依介面，使配接卡可以用於 DLPAR 作業，請輸入下列指令：

   ```
   # pdisable lft0

   # rmdev -l rcm0
   rcm0 Defined

   # rmdev -l lft0
   lft0 Defined

   # rmdev -Rdl pci23
   cor0 deleted
   pci23 deleted
   ```

管理主控台介面可以用於圖形配接卡上的 DLPAR 作業。

迴圈裝置

迴圈裝置是一種可以用作區塊裝置來存取檔案的裝置。

迴圈檔案可包含 ISO 映像檔、磁碟映像檔、檔案系統或邏輯磁區映像檔。例如，將 CD-ROM ISO 映像檔連接至迴圈裝置，並裝載該裝置，即可使用存取 CD-ROM 裝置的相同方式來存取映像檔。

使用 **loopmount** 指令，可以建立迴圈裝置、將指定的檔案連結至迴圈裝置，以及裝載迴圈裝置。使用 **loopumount** 指令，可以卸載迴圈裝置上先前已裝載的映像檔，以及移除裝置。在 AIX 中，並未限制迴圈裝置的數目。預設絕不會建立迴圈裝置；您必須明確地建立該裝置。迴圈裝置的區塊大小一律為 512 位元組。

使用 **mkdev** 指令也可以建立新的裝置、使用 **chdev** 指令則可以進行變更，而使用 **rmdev** 指令則可以進行移除。建立裝置之後，即可裝載它，以存取基礎映像檔或用作原始 I/O 的區塊裝置。使用 **ioctl (IOCINFO)** 指令，則可以擷取基礎映像檔的相關資訊。

下列是 AIX 中的迴圈裝置限制：

- 不支援磁碟映像檔上的 **varyonvg** 指令。
- 只支援唯讀格式的 CD ISO、DVD UDF+ISO 及其他 CD/DVD 映像檔。
- 一個映像檔只可以與一個迴圈裝置相關聯。
- 在工作量分割區中，不支援迴圈裝置。

注意事項

本資訊係針對 IBM 在美國所提供之產品與服務所開發。

在其他國家，IBM 不見得有提供本文件所提及之各項產品、服務或功能。請洽詢當地的 IBM 業務代表，以取得當地目前提供的產品和服務之相關資訊。本文件在提及 IBM 的產品、程式或服務時，不表示或暗示只能使用 IBM 的產品、程式或服務。只要未侵犯 IBM 之智慧財產權，任何功能相當之產品、程式或服務皆可取代 IBM 之產品、程式或服務。不過，任何非 IBM 之產品、程式或服務，使用者必須自行負責作業之評估和驗證責任。

本文件所說明之主題內容，IBM 可能擁有其專利或專利申請案。使用者不得享有本書內容之專利權。 您可以用書面方式來查詢特許權限，來函請寄到：

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785US

若要查詢有關雙位元組字元 (DBCS) 資訊的授權查詢事宜，請聯絡您國家的 IBM 智慧財產部門，或書面提出授權查詢，來函請寄到：

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

International Business Machines Corporation 只依「現況」提供本出版品，不提供任何明示或默示之保證，其中包括且不限於未涉侵權、可商用性或特定目的之適用性的隱含保證。有些司法管轄區在特定交易上，不允許排除明示或暗示的保證，因此，這項聲明不一定適合 貴客戶。

本書中可能會有技術上或排版印刷上的訛誤。因此，IBM 會定期修訂； 並將修訂後的內容納入新版中。IBM 隨時會改進及/或變更本出版品所提及的產品及/或程式，不另行通知。

本資訊中任何對非 IBM 網站的敘述僅供參考，IBM 對該網站並不提供任何保證。該網站上的資料，並非本 IBM 產品所用資料的一部份，如因使用該網站而造成損害，其責任由 貴客戶自行負責。

IBM 得以各種 IBM 認為適當的方式使用或散佈由 貴客戶提供的任何資訊，而無需對 貴客戶負責。

如果本程式之獲授權人為了 (i) 在個別建立的程式和其他程式（包括本程式）之間交換資訊，以及 (ii) 相互使用所交換的資訊，因而需要相關的資訊，請洽詢：

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785US

上述資料之取得有其特殊要件，在某些情況下必須付費方得使用。

IBM 基於 IBM 客戶合約、IBM 國際程式授權合約或雙方之任何同等合約的條款，提供本文件所提及的授權程式與其所有適用的授權資料。

引用的效能資料與客戶範例僅供舉例說明之用。實際的效能結果可能會因為特定的配置與運作條件而有差異。

本書所提及之非 IBM 產品資訊，係一由產品的供應商，或其出版的聲明或其他公開管道取得。IBM 並未測試過這些產品，也無法確認這些非 IBM 產品的執行效能、相容性或任何對產品的其他主張是否完全無誤。如果您對非 IBM 產品的性能有任何的疑問，請逕向該產品的供應商查詢。

關於 IBM 未來方針或意向之聲明，僅代表 IBM 的目標與目的，隨時可能變動或撤消，而不另行通知。

本出版品中所顯示的所有 IBM 價格皆為 IBM 的現行建議零售價，隨時可能變更，恕不另行通知。經銷商的價格可能與此不同。

本資訊僅供規劃之用。在所述的產品上市之前，本文之資訊隨時可能更改。

本資訊中包含日常商業活動使用的資料與報告範例。為了盡可能完整地說明，範例中包括了個人、公司行號、品牌以及產品等的名稱。此等名稱皆屬虛構，若與實際的人員或企業有任何雷同之處，純屬巧合。

著作權：

本資訊含有原始語言之範例應用程式，用以說明各作業平台中之程式設計技術。貴客戶可以為了研發、使用、銷售或散布符合範例應用程式所適用的作業平台之應用程式介面的應用程式，以任何形式複製、修改及散布這些範例程式，不必向 IBM 付費。該等範例並未在一切情況下完整測試。因此，IBM 不保證或默示保證這些樣本程式之可靠性、服務性或功能。這些程式範例以「現狀」提供，且無任何保證。 IBM 對因使用這些程式範例而產生的任何損害概不負責。

這些程式範例或是任何衍生著作的每一份副本或任何部份，都必須包括如下所示的版權聲明：

©（貴公司名稱）（年）。

本程式碼之若干部分係衍生自 IBM 公司的範例程式。

© Copyright IBM Corp. _enter the year or years_.

隱私權條款考量

IBM 軟體產品（包括軟體即服務解決方案，下面簡稱「軟體供應項目」）可能會使用 Cookie 或其他技術來收集產品使用情況資訊，以協助改良一般使用者體驗、修正與一般使用者的互動，或用於其他用途。在許多情況下，「軟體供應項目」並不會收集個人識別資訊。本公司的部分「軟體供應項目」可協助讓您收集個人識別資訊。 如果本「軟體供應項目」使用 Cookie 來收集個人識別資訊，以下將規定關於這類供應項目使用 Cookie 的特定資訊。

本「軟體供應項目」不會使用 Cookie 或其他技術來收集個人識別資訊。

如果本「軟體供應項目」所部署的配置向 貴客戶提供了通過 Cookie 及其他技術來收集一般使用者個人可識別資訊的能力， 貴客戶應該自行尋求關於此類資料收集所適用的任何法律建議，其中包括對於注意事項及同意的任何需求。

如需將各種技術（包括 Cookie）用於這些目的的相關資訊，請參閱《IBM 隱私權條款》(http://www.ibm.com/privacy) 和《IBM 線上隱私權聲明》(http://www.ibm.com/privacy/details) 以及 http://www.ibm.com/software/info/product-privacy 中標題為「Cookie、Web 訊號指標及其他技術」和「IBM 軟體產品及軟體即服務 (SaaS) 的隱私權聲明」的章節。

商標

IBM、IBM 標誌及 ibm.com 是 International Business Machines Corp. 在世界許多管轄區註冊的商標或註冊商標。其他產品及服務名稱可能是 IBM 或其他公司的商標。如需最新的 IBM 商標清單，請造訪位於 www.ibm.com/legal/copytrade.shtml 的 Copyright and trademark information 網頁。

Linux 是 Linus Torvalds 在美國及（或）其他國家的註冊商標。

UNIX 是 The Open Group 在美國及（或）其他國家的商標。

Windows 是 Microsoft Corporation 在美國及（或）其他國家/地區的商標。