



Sun StorEdge™ SAM-FS 儲存與歸檔管理指南

版本 4 Update 4

Sun Microsystems, Inc.
www.sun.com

文件編號 819-4783-10
2005 年 12 月，修訂版 A

請將您對本文件的意見提交至：<http://www.sun.com/hwdocs/feedback>

Copyright 2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 版權所有。

Sun Microsystems, Inc. 對於本文件所述技術擁有智慧財產權。這些智慧財產權包含 <http://www.sun.com/patents> 上所列的一項或多項美國專利，以及在美國與其他國家/地區擁有的一項或多項其他專利或申請中專利，但並不以此為限。

本文件及相關產品在限制其使用、複製、發行及反編譯的授權下發行。未經 Sun 及其授權人(如果有)事先的書面許可，不得使用任何方法、任何形式來複製本產品或文件的任何部分。

協力廠商軟體，包含字型技術，其版權歸 Sun 供應商所有，經授權後使用。

本產品中的某些部分可能源自加州大學授權的 Berkeley BSD 系統的開發成果。UNIX 是在美國及其他國家/地區的註冊商標，已獲得 X/Open Company, Ltd. 專屬授權。

Sun、Sun Microsystems、Sun 標誌、AnswerBook2、docs.sun.com、Solaris 和 Sun StorEdge 是 Sun Microsystems, Inc. 在美國及其他國家/地區的商標或註冊商標。

所有 SPARC 商標都是 SPARC International, Inc. 在美國及其他國家/地區的商標或註冊商標，經授權後使用。凡具有 SPARC 商標的產品都是採用 Sun Microsystems, Inc. 所開發的架構。

Mozilla 是 Netscape Communications Corporation 在美國和其他國家/地區的商標或註冊商標。

OPEN LOOK 和 Sun™ Graphical User Interface (Sun 圖形化使用者介面) 都是由 Sun Microsystems, Inc. 為其使用者及授權者所開發的技術。Sun 感謝 Xerox 公司在研究和開發視覺化或圖形化使用者介面之概念上，為電腦工業所做的開拓性貢獻。Sun 已向 Xerox 公司取得 Xerox 圖形化使用者介面之非獨占性授權，該授權亦適用於使用 OPEN LOOK GUI 並遵守 Sun 書面授權合約的 Sun 公司授權者。

美國政府權利 — 商業軟體。政府使用者均應遵守 Sun Microsystems, Inc. 的標準授權合約和 FAR 及其增補文件中的適用條款。

本文件以其「原狀」提供，對任何明示或暗示的條件、陳述或擔保，包括對適銷性、特殊用途的適用性或非侵權性的暗示保證，均不承擔任何責任，除非此免責聲明的適用範圍在法律上無效。



目錄

前言 xvii

本書架構 xvii

使用 UNIX 指令 xviii

Shell 提示符號 xviii

印刷排版慣例 xix

相關文件 xx

線上存取 Sun 文件資料 xx

協力廠商網站 xxi

連絡 Sun 技術支援 xxi

授權 xxi

安裝協助 xxi

Sun 歡迎您提出寶貴意見 xxii

1. 簡介 1

功能 1

儲存裝置 2

File System Manager 3

▼ 要呼叫 File System Manager 4

建立其他管理者和使用者帳號 5

▼ 建立其他管理員帳號 5

▼ 建立其他 Guest 帳號	5
從 File System Manager 管理其他伺服器	6
2. 使用自動程式庫和手動載入磁碟機	7
慣例	8
指令引數	8
專有名詞	9
自動程式庫作業	9
▼ 停止可移除式媒體作業	10
▼ 啓動可移除式媒體作業	11
▼ 開啓自動程式庫	11
▼ 關閉自動程式庫	11
▼ 將卡匣載入自動程式庫	12
▼ 從磁碟機中卸載卡匣	13
製作卡匣標籤	13
▼ 製作或重新製作磁帶標籤	13
▼ 製作或重新製作光碟標籤	14
▼ 稽核磁碟區	15
▼ 稽核自動程式庫 (僅限於直接連結式)	16
使用清理卡匣	16
▼ 重設清理週期數目	16
▼ 搭配條碼使用清理卡匣	17
▼ 不搭配條碼使用清理卡匣	17
▼ 清理磁帶機	18
磁帶機自動清理	19
▼ 清除媒體錯誤	19
▼ 從磁碟機中移除卡住的卡匣	20
目錄檔作業，匯入和匯出卡匣	22
追蹤匯出的媒體 — 記錄器	22

從自動程式庫匯入與匯出	23
▼ 從使用信箱的程式庫匯入卡匣	23
▼ 從使用信箱的程式庫匯出卡匣	24
▼ 自不使用信箱的程式庫匯入卡匣	24
▼ 自不使用信箱的程式庫匯出卡匣	25
▼ 啓動載入通知	25
手動載入磁碟機作業	26
▼ 載入卡匣	26
▼ 卸載卡匣	26
▼ 檢視程式庫目錄檔	27
3. 歸檔	29
歸檔程序簡介	29
歸檔組	30
歸檔操作	31
步驟 1：辨識要歸檔的檔案	31
步驟 2：組合歸檔請求	33
步驟 3：排定歸檔請求	34
步驟 4：爲歸檔請求中的檔案進行歸檔	36
範例預設執行結果	36
歸檔器常駐程式	37
歸檔記錄檔案與事件記錄	37
關於 archiver.cmd 檔案	39
▼ 建立或修改 archiver.cmd 檔案及傳遞您的變更	40
archiver.cmd 檔案	41
archiver.cmd 檔案範例	41
使用歸檔器指令	43
全域歸檔指令	43
archivemeta 指令：控制是否歸檔中介資料	43

- archmax 指令：控制歸檔檔案的大小 44
- bufsize 指令：設定歸檔器的緩衝區大小 44
- drives 指令：控制用於歸檔的磁碟機數量 45
- examine 指令：控制歸檔掃描 46
- interval 指令：指定歸檔間隔 47
- logfile 指令：指定歸檔器記錄檔 47
 - ▼ 備份歸檔器記錄檔 48
- notify 指令：重新命名事件通知程序檔 48
- ovflmin 指令：控制磁碟區溢位 48
- wait 指令：延遲歸檔器啟動 50
- 檔案系統指令 50
 - fs 指令：指定檔案系統 51
 - 其他檔案系統指令 51
- 歸檔組指定指令 51
 - 指定歸檔組 52
 - 檔案大小搜尋條件：-access 和 -nftv 53
 - 檔案大小搜尋條件：-minsize 和 -maxsize 53
 - 所有者和群組搜尋條件：-user 和 -group 54
 - 使用模式比對的檔案名稱搜尋條件：-name *regex* 54
 - 釋放和分段備份檔案屬性：-release 和 -stage 56
 - 歸檔組成員衝突 57
- 歸檔備份指令 58
 - 歸檔後釋放磁碟空間：-release 59
 - 延遲釋放磁碟空間：-norelease 59
 - 同時使用 -release 和 -norelease 60
 - 設定歸檔時間 60
 - 自動取消歸檔 60
 - 為中介資料指定多個備份 60

歸檔組備份參數	61
控制歸檔檔案的大小：-archmax	62
設定歸檔器的緩衝區大小：-bufsize	62
指定用於歸檔請求的磁帶機數量：-drivemax、-drivemin 及 -drives	63
最大化磁碟區上的空間：-fillvsns	64
指定歸檔緩衝區鎖定：-lock	64
製作離線檔案的歸檔備份：-offline_copy	65
指定回收	66
關聯歸檔：-join path	66
控制取消歸檔	67
控制如何寫入歸檔檔案：-tapenonstop	68
預留磁碟區：-reserve	69
設定歸檔優先權：-priority	72
排定歸檔：-startage、-startcount 及 -startsize	74
VSN 關聯指令	75
VSN 池指令	77
關於磁碟歸檔	78
配置設定指引	79
磁碟歸檔指令	79
▼ 啟動磁碟歸檔	81
磁碟歸檔範例	82
範例 1	82
範例 2	83
範例 3	84
規劃歸檔作業	84
預覽佇列	85
歸檔器範例	86
範例 1	86

範例 2	88
範例 3	91
範例 4	95

4. 釋放 99

釋放程序簡介	99
作業理論	100
定義	100
時間	100
候選者	101
優先權	101
權數	101
部分釋放	102
關於部分釋放和部分分段備份	102
系統管理員選項摘要	103
使用者選項摘要	104
關於 <code>releaser.cmd</code> 檔案	105
指定與時間和大小相關的釋放優先權指令	105
檔案時間	105
檔案大小	107
為個別檔案系統指定指令： <code>fs</code>	107
指定除錯指令： <code>no_release</code> 和 <code>display_all_candidates</code>	108
指定最短常駐時間： <code>min_residence_age</code>	108
指定記錄檔： <code>logfile</code>	109
限制釋放重新歸檔的檔案： <code>rearch_no_release</code>	110
調整釋放器候選者清單的大小： <code>list_size</code>	111
<code>archiver.cmd</code> 檔案在釋放中的角色	111
規畫釋放器作業	112
手動執行釋放器	114

5. 分段備份 115

- 關於 `stager.cmd` 檔案 115
 - ▼ 建立或修改 `stager.cmd` 檔案及傳遞您的變更 116
 - 指定磁碟機數目 117
 - 設定分段備份緩衝區大小 117
 - 指定記錄檔 118
 - 指定分段備份請求數目 121
 - `stager.cmd` 檔案範例 121
 - `archiver.cmd` 檔案在分段備份中的角色 121
- 設定預覽請求的優先權 122
 - 全域 VSN 與時間指令 123
 - 全域或檔案系統特定參數指令 124
- 計算預覽請求的總優先權 125
- 設定預覽請求優先機制 125
 - 範例 1：強制分段備份請求 126
 - 範例 2：強制歸檔請求 126
 - 範例 3：依媒體排定請求的優先權 127
 - 範例 4：複合排定優先權 127

6. 回收 129

- 回收程序簡介 129
- 使用回收指令 131
 - 指定記錄檔：`logfile` 指令 131
 - 防止回收：`no_recycle` 指令 131
 - 指定回收整個自動程式庫：程式庫指令 132
- 規劃回收作業 133
 - ▼ 步驟 1：建立 `recycler.cmd` 檔案 134
 - `recycler.cmd` 檔案範例 135
 - ▼ 步驟 2：編輯 `archiver.cmd` 檔案 136

- ▼ 步驟 3：執行回收器 137
- ▼ 步驟 4：建立回收器的 crontab 檔案 139
- ▼ 步驟 5：移除 -recycle_ignore 和 ignore 參數 139
- ▼ 步驟 6：建立 recycler.sh 檔案 139

7. 使用 Sun SAM-Remote 軟體 141

Sun SAM-Remote 軟體簡介 141

功能 141

需求 143

限制 143

技術簡介 143

Sun SAM-Remote 伺服器簡介 144

Sun SAM-Remote 用戶端簡介 144

Sun SAM-Remote 伺服器與 Sun SAM-Remote
用戶端之間的互動 145

程式庫目錄 145

歸檔 146

設定 Sun SAM-Remote 軟體 146

配置範例 146

配置軟體 147

- ▼ 登入潛在的伺服器與用戶端主機 147
- ▼ 確認用戶端與伺服器配置 147
- ▼ 編輯 mcf 檔案 149
- ▼ 定義 Sun SAM-Remote 用戶端 151
- ▼ 在伺服器的 mcf 檔案中定義 Sun SAM-Remote 伺服器 152
- ▼ 建立 Sun SAM-Remote 伺服器配置檔案 153
- ▼ 啟用歸檔 155

使用 Sun SAM-Remote 軟體進行回收 159

於 Sun SAM-Remote 環境中進行回收 — 方法 1 160

- 伺服器 sky 的配置檔案 160
- 用戶端 zeke 的配置檔案 162
 - ▼ 配置回收 — 方法 1 162
 - ▼ 回收 no-data VSN 179
 - ▼ 回收 partially full VSN 181
- 於 Sun SAM-Remote 環境進行回收 — 方法 2 183
 - ▼ 設定回收 — 方法 2 183
- 8. 進階主題 185**
 - 使用裝置記錄 185
 - 何時使用裝置記錄檔 186
 - 啟動裝置記錄檔 186
 - ▼ 使用 samset(1M) 指令以啟動裝置記錄檔 187
 - ▼ 編輯 defaults.conf 檔案以啟動裝置記錄檔 187
 - 使用可移除式媒體檔案 188
 - ▼ 建立可移除式媒體或磁碟區溢位檔案 188
 - 使用區段檔案 190
 - 歸檔 190
 - 災難復原 191
 - 使用系統錯誤功能報告 191
 - ▼ 啟動 SEF 報告 191
 - SEF 報告輸出 192
 - ▼ 產生 SEF 輸出 192
 - 管理 SEF 記錄檔 195
 - SEF sysevent 功能 195
 - ▼ 建立 SEF sysevent 處理程式 195
- A. 使用供應商指定作業程序的程式庫基本作業 197**
 - ADIC/Grau 自動程式庫 197

▼ 匯入卡匣	198
▼ 匯出卡匣	198
Fujitsu LMF 自動程式庫	199
▼ 匯入卡匣	199
▼ 匯出卡匣	200
IBM 3584 UltraScalable 程式庫	201
匯入卡匣	201
清潔磁碟機	201
分割	201
▼ 移除卡匣	201
IBM 3494 程式庫	202
▼ 匯入卡匣	202
▼ 匯出卡匣	202
Sony 直接連結式 8400 PetaSite 自動程式庫	203
▼ 匯入磁帶	203
匯出磁帶	204
▼ 在不將信箱插槽當做儲存插槽的情況下匯出磁帶	204
▼ 將信箱插槽用作儲存插槽以匯出磁帶	204
▼ 如何將卡匣移至不同插槽	205
Sony 網路連結式自動程式庫	206
▼ 匯入卡匣	206
▼ 匯出卡匣	207
StorageTek ACSLS 連結式自動程式庫	207
▼ 匯入磁帶	208
▼ 使用信箱匯出磁帶	208
字彙表	211
索引	221

表

表 P-1	Shell 提示符號 xviii
表 P-2	印刷排版慣例 xix
表 P-3	相關文件 xx
表 1-1	自動程式庫常駐程式 3
表 2-1	指令引數 8
表 2-2	專有名詞 9
表 2-3	samcmd(1M) load 的引數 12
表 2-4	tplabel(1M) 的引數 13
表 2-5	odlabel(1M) 的引數 14
表 2-6	auditslot(1M) 的引數 15
表 2-7	chmed(1M) 的引數 17
表 2-8	chmed(1M) 的引數 20
表 2-9	auditslot(1M) 的引數 20
表 2-10	chmed(1M) 的引數 21
表 2-11	samexport(1M) 的引數 24
表 3-1	歸檔器記錄檔欄位 38
表 3-2	archiver.cmd 檔案指令單位 41
表 3-3	archmax 指令的引數 44
表 3-4	bufsize 指令的引數 45
表 3-6	examine 指令的方法引數的值 46

表 3-5	<code>drives</code> 指令的引數	46
表 3-7	<code>ovflmin</code> 指令的引數	49
表 3-8	歸檔組指定指令的引數	52
表 3-9	<code>-access</code> 時間後綴	53
表 3-10	<code>-minsize</code> 和 <code>-maxsize</code> 大小後綴	53
表 3-11	<code>-release</code> 選項	57
表 3-12	<code>-stage</code> 指令的屬性	57
表 3-13	歸檔組備份參數的引數	61
表 3-14	<code>-drivemax</code> 、 <code>-drivemin</code> 及 <code>-drives</code> 參數的引數	63
表 3-15	歸檔組範例分割	64
表 3-16	<code>-offline_copy</code> 指令的方法引數的值	66
表 3-17	歸檔組表單範例	71
表 3-18	擁有者組表單範例	71
表 3-19	檔案系統表單範例	72
表 3-20	歸檔優先權	73
表 3-21	<code>-startage</code> 、 <code>-startcount</code> 及 <code>-startsize</code> 指令格式	74
表 3-22	VSN 關聯指令的引數	75
表 3-23	VSN Pool 指令的引數	77
表 3-24	目錄結構範例	86
表 4-1	部分釋放的掛載選項	103
表 4-2	使用者釋放選項	104
表 4-3	歸檔設定引數檔案屬性	111
表 5-1	<code>drives</code> 指令的引數	117
表 5-2	<code>bufsize</code> 指令的引數	118
表 5-3	事件引數的關鍵字	119
表 5-4	分段備份程式記錄檔欄位	120
表 5-5	可出現在 <code>archiver.cmd</code> 檔案中的分段備份 <code>archiver.cmd</code>	122
表 5-6	參數優先權指令	124
表 5-7	請求優先權範例	126
表 6-1	回收方法和媒體類型	130

表 6-3	程式庫指令的參數值	132
表 6-2	no_recycle 指令的引數	132
表 6-4	歸檔組回收指令	136
表 7-1	samu(1M) R 畫面旗標	158
表 8-1	request(1) 指令的引數	189
表 8-2	import(1M) 指令的引數	198
表 8-3	samexport(1M) 指令的引數	199
表 8-5	samexport(1M) 指令的引數	200
表 8-4	import(1M) 指令的引數	200
表 8-6	move(1M) 指令的引數	204
表 8-7	move(1M) 指令的引數	205
表 8-8	import(1M) 指令的引數	206
表 8-9	samexport(1M) 指令的引數	207
表 8-10	import(1M) 指令的引數	208
表 8-11	samexport(1M) 指令的引數	209

前言

本手冊「Sun StorEdge™ SAM-FS 儲存與歸檔管理指南」說明 Sun StorEdge SAM-FS 版本 4 Update 4 (4U4) 發行版本中所支援的儲存與歸檔管理軟體。Sun StorEdge SAM-FS 軟體會自動將檔案從線上磁碟複製到歸檔媒體。歸檔媒體可以由線上磁碟或可移除式媒體卡匣組成。

以下 Sun Solaris™ 作業系統 (OS) 平台上可支援 Sun StorEdge SAM-FS 4U4 發行版本：

- Solaris 9 Update 3 4/03
- Solaris 10

本手冊是針對負責配置和維護 Sun StorEdge SAM-FS 軟體的系統管理員所撰寫的。我們假設您是系統管理員，並知道 Solaris 作業系統管理程序，包含建立帳號、執行系統備份及其他基本的 Solaris 系統管理工作。

備註 – 您可以購買 Sun StorEdge QFS 和 Sun StorEdge SAM-FS 軟體的授權，以將 Sun StorEdge QFS 檔案系統和 Sun StorEdge SAM-FS 軟體中所找到的儲存與歸檔管理合併起來。這種系統即所謂的 *SAM-QFS*。

除非為了清楚明瞭起見，本手冊不針對 SAM-QFS 配置提供圖解。關於本手冊中的儲存與歸檔管理部分，您可以假設 Sun StorEdge SAM-FS 的參照也適用於 SAM-QFS 配置。同樣地，關於檔案系統設計和功能的部分，您可以假設 Sun StorEdge QFS 的參照也適用於 SAM-QFS 配置。

本書架構

本手冊包含以下各章：

- 第 1 章提供簡介資訊。
- 第 2 章說明基本操作。本章中的資訊適用於大多數自動程式庫與手動載入裝置。

- 第 3 章說明歸檔程序。
- 第 4 章說明釋放程序。
- 第 5 章說明分段備份程序。
- 第 6 章說明回收程序。
- 第 7 章說明如何使用 Sun SAM-Remote 軟體。
- 第 8 章說明 Sun StorEdge SAM-FS 作業中的進階選項。
- 附錄 A 說明如何管理程式庫中的卡匣，並提供了僅適用於該類型程式庫的操作說明。本章將說明這些程式庫及其特定的基本操作程序。

詞彙表提供本手冊及其他 Sun StorEdge QFS 和 Sun StorEdge SAM-FS 文件資料中所用的專有名詞之定義。

使用 UNIX 指令

本文件不會介紹基本 UNIX® 指令和操作程序，如關閉系統、啓動系統及配置裝置。若需此類資訊，請參閱以下文件：

- 系統隨附的軟體文件
- Solaris™ 作業系統文件，其 URL 為：
<http://docs.sun.com>

Shell 提示符號

表 P-1 顯示本手冊所使用的 shell 提示。

表 P-1 Shell 提示符號

Shell	提示符號
C shell	電腦名稱%
C shell 超級使用者	電腦名稱#
Bourne shell 和 Korn shell	\$
Bourne shell 和 Korn shell 超級使用者	#

印刷排版慣例

表 P-2 列出本手冊所使用的印刷排版慣例。

表 P-2 印刷排版慣例

字體或符號	意義	範例
AaBbCc123	指令、檔案及目錄的名稱；螢幕畫面輸出。	請編輯您的 <code>.login</code> 檔案。 請使用 <code>ls -a</code> 列出所有檔案。 % You have mail.
AaBbCc123	您所鍵入的內容 (與螢幕畫面輸出相區別)。	% su Password:
術語強調變數	新的字彙或術語、要強調的詞。將用實際的名稱或數值取代的指令行變數。	這些被稱為類別選項。 您必須是超級使用者才能執行此操作。 要刪除檔案，請鍵入 rm 檔案名稱。
「AaBbCc123」	用於書名及章節名稱。	「Solaris 10 使用者指南」 請參閱第 6 章「資料管理」。
[]	在語法表示上，括號表示該引數是選用引數。	<code>scmadm [-d sec] [-r n[:n][,n]...][-z]</code>
{ arg arg }	在語法表示上，大括號與直線表示必須指定其中一個引數。	<code>sndradm -b { phost shost }</code>
\	指令行結尾的反斜線 (\) 表示該指令繼續到下一行。	<code>atm90 /dev/md/rdisk/d5 \ /dev/md/rdisk/d1</code>

相關文件

本手冊屬於文件集的一部份，該文件集說明 Sun StorEdge QFS 和 Sun StorEdge SAM-FS 軟體產品之作業。表 P-3 顯示這些產品的完整發行版本 4U4 文件集。

表 P-3 相關文件

書名	文件號碼
「Sun StorEdge SAM-FS 檔案系統配置與管理指南」	819-4808-10
「Sun StorEdge SAM-FS 安裝與升級指南」	819-4778-10
「Sun StorEdge SAM-FS 疑難排解指南」	819-4788-10
「Sun StorEdge QFS 配置與管理指南」	819-4798-10
「Sun QFS, Sun SAM-FS, and Sun SAM-QFS Man Page Reference Manual」	819-2757-10
「Sun StorEdge QFS and Sun StorEdge SAM-FS 4.4 版本說明」	819-4803-10

線上存取 Sun 文件資料

Sun StorEdge QFS 和 Sun StorEdge SAM-FS 軟體發行包括這些產品的相關文件之 PDF 檔案。這些 PDF 檔案可透過下列位置進行檢視：

- Sun 的 Network Storage 文件網站。

本網站包含許多有關儲存軟體產品的文件。

- a. 若要存取此網站，請至下列 URL：

http://www.sun.com/products-n-solutions/hardware/docs/Software/Storage_Software

將會顯示 [Storage Software] 頁面。

- b. 在下列清單中按一下合適的連結：

- Sun StorEdge QFS 軟體
- Sun StorEdge SAM-FS 軟體
- docs.sun.com。

本網站包含 Solaris 及其他許多 Sun 軟體產品的文件。

- a. 若要存取此網站，請至下列 URL：

<http://docs.sun.com>

將出現 docs.sun.com 頁面。

- b. 在搜尋方塊中，搜尋下列其中一項產品，找出您所使用的產品文件資料：
- Sun StorEdge QFS
 - Sun StorEdge SAM-FS

協力廠商網站

Sun 對於本文件中所提及之協力廠商網站的使用不承擔任何責任。Sun 對於此類網站或資源中的 (或透過它們所取得的) 任何內容、廣告、產品或其材料不做背書，也不承擔任何責任。對於因使用或依靠此類網站或資源中的 (或透過它們所取得的) 任何內容、產品或服務而造成的或連帶產生的實際或名義上之損壞或損失，Sun 概不負責，也不承擔任何責任。

連絡 Sun 技術支援

如果您在本文件中找不到所需之本產品相關技術問題的解答，請至：

<http://www.sun.com/service/contacting>

授權

如需有關取得 Sun StorEdge SAM-FS 軟體授權的資訊，請連絡您的 Sun 銷售代表或您的授權服務供應商 (ASP)。

安裝協助

要取得安裝與配置方面的服務，請連絡 Sun 企業服務 (美國：1-800-USA4SUN) 或連絡當地的企業服務業務代表。

Sun 歡迎您提出寶貴意見

Sun 致力於提高文件品質，因此誠心歡迎您提出意見和建議。請至下列網址提出您對本文件的意見：

<http://www.sun.com/hwdocs/feedback>

請在您的意見中註明本文件的書名和文件號碼：「Sun StorEdge SAM-FS 儲存與歸檔管理指南」，文件號碼：819-4783-10。

第1章

簡介

Sun StorEdge SAM-FS 環境提供搭配儲存、歸檔管理和擷取功能的可配置檔案系統。Sun StorEdge SAM-FS 軟體會將線上磁碟快取記憶體中的檔案複製到歸檔媒體，以便歸檔檔案。歸檔媒體可以由其他檔案系統中的磁碟片段組成，或者由自動或手動載入儲存裝置的可移除式磁帶或磁光碟機匣組成。此外，Sun StorEdge SAM-FS 軟體會使用站點指定的使用率臨界值，自動維護線上磁碟空間。其會釋放與歸檔檔案資料相關的磁碟空間，並在需要時將檔案復原至線上磁碟。

本章提供 Sun StorEdge SAM-FS 元件的技術簡介。介紹下列主題：

- 第 1 頁「功能」
- 第 2 頁「儲存裝置」
- 第 3 頁「File System Manager」

功能

Sun StorEdge SAM-FS 環境儲存與歸檔管理軟體要和檔案系統 (例如：Sun StorEdge QFS) 一起運作。檔案系統是高效能 UNIX 檔案系統，位於伺服器磁碟快取記憶體中。如需更多有關此檔案系統的資訊，請參閱「Sun StorEdge QFS 配置與管理指南」。

位於 Sun StorEdge SAM-FS 環境的其他元件如下所示：

- 歸檔器會將線上磁碟快取記憶體檔案自動複製到歸檔媒體。歸檔媒體可以由線上磁碟檔案或可移除式媒體匣組成。依照預設，歸檔器會於 Sun StorEdge SAM-FS 檔案系統中自動建立一份所有檔案的歸檔備份，並將歸檔備份寫入歸檔媒體。您可以設定歸檔器的組態，以在各種歸檔媒體上建立最多四份歸檔備份。如果檔案已分段，每個區段將視為一個檔案，且每個區段將獨立歸檔。在磁碟檔案與站點可定義的選擇條件組相符後，即初始化歸檔程序。

如需有關歸檔器的詳細資訊，請參閱第 29 頁「歸檔」。如需有關分段檔案的詳細資訊，請參閱第 190 頁「使用區段檔案」。

- 釋放器可透過釋放合格的歸檔檔案所佔用之磁碟區塊，以站點指定的百分比使用率臨界值，自動維護檔案系統的線上磁碟快取記憶體。

釋放是釋放歸檔檔案資料使用的主要 (磁碟) 儲存空間的程序。兩個臨界值 (均表示為總磁碟空間的百分比) 可用來管理線上磁碟快取記憶體的未使用空間。這些臨界值是高值參數與低值參數。線上磁碟消耗量超過高值參數時，系統會自動開始釋放合格的歸檔檔案佔用的磁碟空間。達到低值參數之前，會釋放歸檔檔案資料佔用的磁碟空間。選擇釋放的檔案取決於檔案大小與存在時間。也可以在磁碟上保留檔案的第一部份，以快速存取與隱藏分段備份延遲。如果檔案已歸檔在區段中，則可以個別地釋放檔案的部份。如需有關釋放器的詳細資訊，請參閱第 99 頁「釋放」。

- 分段備份程式可將檔案資料回復至磁碟快取記憶體。使用者或程序要求已從磁碟快取記憶體釋放的檔案資料時，分段備份程式會自動將檔案資料複製回線上磁碟快取記憶體。

存取已釋放資料區塊的檔案時，分段備份程式會自動將檔案或檔案區段資料分段備份回線上磁碟快取記憶體。讀取操作會直接在分段備份作業後進行追蹤，以便在完全分段備份整個檔案之前，應用程式可立即使用該檔案。

Sun StorEdge SAM-FS 軟體程序會自動分段備份請求錯誤。如果傳回分段備份錯誤，系統會嘗試尋找下一個可用的檔案歸檔備份。可以自動處理的分段備份錯誤包括媒體錯誤、無法使用媒體、無法使用自動程式庫及其他錯誤。如需有關分段備份的詳細資訊，請參閱第 115 頁「分段備份」。

- 回收器可清除已過期的歸檔備份之歸檔磁碟區，以使磁碟區可重複使用。

當使用者修改檔案時，與這些舊版檔案相關聯的歸檔備份，在其歸檔媒體上會被視為已過期。系統不再需要這種備份所以能夠將它清除。回收器會識別已過期歸檔備份所佔比例最大的歸檔磁碟區，並會將未到期的備份移至獨立磁碟區進行保存。

如果可移除式媒體磁碟區只包含過期的備份，您可以執行以下動作之一：

- 您可以重新製作磁碟區標籤以便立即重新使用。
- 您可以將磁碟區匯出至離線儲存裝置，當做檔案變更的歷史記錄。您可以使用標準 UNIX 公用程式，從過期的歸檔備份復原之前的檔案版本。

回收程序對最終使用者來說是不需設定的，因為它與使用者的資料檔案相關。如需有關回收的詳細資訊，請參閱第 129 頁「回收」。

儲存裝置

Sun StorEdge SAM-FS 環境支援各種磁帶儲存與磁光碟機裝置。Sun StorEdge SAM-FS 支援的自動程式庫，可以根據其連接至環境的方式分為以下群組：

- 直接連結。直接連結式程式庫是使用小型電腦系統介面 (SCSI) 直接連接至主機系統。此方式可稱為直接連結或光纖通道連結。例如：Sun StorEdge 程式庫使用直接連結。Sun StorEdge SAM-FS 系統使用 SCSI 標準直接控制這些自動程式庫。

- 網路連結。Sun StorEdge SAM-FS 軟體可以配置為程式庫主機系統的用戶端。網路連結式程式庫包含某些 StorageTek、ADIC/Grau、IBM 及 Sony 程式庫。這些程式庫使用廠商提供的套裝軟體。在這些案例中，Sun StorEdge SAM-FS 軟體使用特別針對自動程式庫設計的常駐程式，當做與供應商軟體之間的介面。

表 1-1 將顯示各種自動程式庫特定的常駐程式。

表 1-1 自動程式庫常駐程式

常駐程式	說明
sam-robotd	監控自動裝置常駐程式的執行。sam-amld 常駐程式會自動啟動 sam-robotd 常駐程式。
sam-genericd	控制直接連結式程式庫與媒體抽換裝置。同時透過 DAS 介面控制 ADIC 程式庫。
sam-stkd	在 ACSAPI 介面中控制 StorageTek 媒體抽換裝置。
sam-ibm3494d	透過 lmcpsd 介面控制 IBM 3494 程式庫。
sam-sonyd	在 DZC-8000S 介面中控制 Sony 網路連結式自動程式庫。

有關支援的儲存裝置清單，請與 Sun Microsystems 銷售代表或授權的服務供應商 (ASP) 連絡。

Sun StorEdge SAM-FS 環境內管理裝置之間的關係定義在主要配置檔案 /etc/opt/SUNWsamfs/mcf 中。mcf 檔案指定 Sun StorEdge SAM-FS 環境中包含的可移除式媒體裝置、程式庫，以及檔案系統。mcf 檔案中為每台設備指定了唯一的設備識別碼。mcf 中的項目也定義了手動掛載的歸檔裝置與自動程式庫目錄檔。

如有必要，系統會使用標準 Solaris 磁碟與磁帶裝置驅動程式。針對 Solaris 作業系統 (OS) 並未直接支援的裝置，例如某些程式庫與光碟裝置，Sun StorEdge SAM-FS 套裝軟體中包含特殊的裝置驅動程式。

File System Manager

File System Manager 軟體為瀏覽器基礎的圖形化使用者介面，讓您能夠配置和控制一個或多個 Sun StorEdge QFS 或從中央位置而來的 Sun StorEdge SAM-QFS 伺服器。要存取這個中央位置，您可以使用您的網路上任何主機上的網頁瀏覽器。

軟體的目標是提供較不複雜的方式以執行大多與 Sun StorEdge QFS 或 Sun StorEdge SAM-QFS 伺服器相關的一般作業。為了進一步配置或管理伺服器，請使用伺服器的指令行介面、程序檔、配置檔等等。若需更多資訊和安裝檔案系統管理之指示，請參閱「Sun StorEdge SAM-FS 安裝與升級指南」。

安裝 File System Manager 之後，您便可以使用兩個可能的使用者名稱 (samadmin 和 samuser) 以及兩個不同的角色 (SAMadmin 或 no role) 登入到軟體。您可以使用 File System Manager 執行的作業會依您在登入時採用的使用者名稱和角色而有所不同。以下是其中差異之處：

- 若您使用 samadmin 登入，您便可以從兩個角色之中選擇其中一個角色。
 - SAMadmin 角色將賦予您完整的管理員權限，可以配置、監視、控制和重新配置在您的 Sun StorEdge QFS 環境中之裝置。

僅有 Sun StorEdge QFS 管理者應該使用 SAMadmin 角色登入。所有其他的使用者應該以 samuser 的身份登入。
 - no role 角色只能讓您監視環境。您無法使用任何方法變更或重新配置環境。
- 若您登入為 samuser，您僅可以監視環境。您無法使用任何方法變更或重新配置環境。

關於系統管理，請注意在有安裝 File System Manager 伺服器上的 Solaris root 使用者並不一定是 File System Manager 的管理員。只有 samadmin 具有對 File System Manager 應用程式的管理員權限。root 使用者是管理工作站的管理員。

▼ 要呼叫 File System Manager

執行此程序以將 File System Manager 載入您的瀏覽器。

1. 登入到管理工作站的 Web 伺服器。
2. 從 Web 瀏覽器呼叫 File System Manager 軟體。

URL 如下所示：

`https://主機名稱:6789`

在主機名稱鍵入主機名稱。若您要在主機名稱以外指定網域名稱，請以下列格式指定主機名稱：主機名稱.網域名稱。

注意此 URL 以 https 開頭，而非 http。之後會出現 Sun Java Web Console 登入畫面。

3. 在 [User Name] 提示符號下輸入 samadmin。
4. 在 [Password] 提示符號輸入您安裝檔案系統軟體時選取的密碼。
5. 按一下 [SAMadmin] 角色。

只有 Sun StorEdge SAM-FS 管理員應使用 SAMadmin 角色登入。
6. 在 [Role Password] 提示符號下輸入您在步驟 4 所輸入的密碼。
7. 按一下 [Log In]。
8. 按一下 [File System Manager 2.0]。

您現在已登入到 File System Manager。

建立其他管理者和使用者帳號

在初始 File System Manager 配置之後，您可以隨時建立其他的管理員和 Guest 帳號。這些是屬於管理工作站本端的 Guest 帳號。

在安裝 File System Manager 之後，File System Manager 會建立以下兩種 Solaris 作業系統 (OS) 登入帳號及以下角色：

- 帳號：samadmin、samuser
- 角色：SAMadmin

指定 SAMadmin 角色給使用者帳號 samadmin。此使用者具有管理員權限 (讀取和寫入) 以管理 File System Manager、Sun StorEdge QFS 軟體及 Sun StorEdge SAM-FS 軟體。

使用者帳號 samuser 僅指定給具有 Guest 權限者。此使用者僅具有 Sun StorEdge QFS 和 Sun StorEdge SAM-FS 作業的讀取權限。

如果您移除了 File System Manager 軟體，系統則會移除 samadmin 和 samuser Solaris 帳號及 SAMadmin 角色。不過，移除程序檔不會移除任何手動建立的其他帳號。請使用以下一種或兩種程序來管理任何手動新增的帳號。

▼ 建立其他管理員帳號

管理員帳號所有者具有管理員權限 (讀取和寫入) 以管理 File System Manager、Sun StorEdge QFS 軟體及 Sun StorEdge SAM-FS 軟體。

1. 登入管理工作站。
2. 鍵入 useradd 使用者名稱。
3. 鍵入 passwd 使用者名稱。
4. 按照畫面上的指示鍵入密碼。
5. 鍵入 usermod -R SAMadmin 使用者名稱。

備註 – 請勿使用 root 做為使用者名稱。

▼ 建立其他 Guest 帳號

Guest 帳號所有者僅具有 Sun StorEdge QFS 和 Sun StorEdge SAM-FS 作業的讀取權限。

1. 登入管理工作站。
2. 鍵入 useradd 帳號名稱。
3. 鍵入 passwd 帳號名稱。

4. 按照畫面上的指示鍵入密碼。

從 File System Manager 管理其他伺服器

根據預設，設定 File System Manager 是爲了管理安裝此程式的伺服器。您也可以使用此軟體來管理執行 Sun StorEdge QFS 或 Sun StorEdge SAM-FS 軟體的其他伺服器，但是那些其他伺服器必須先配置成爲允許檔案系統管理員存取。新增其他的伺服器可讓您透過瀏覽器介面管理其檔案系統、歸檔程序和媒體。

新增伺服器：

1. 在瀏覽器介面以外，請使用 **telnet** 來連接到您要新增以 **root** 身份登入的伺服器。
2. 使用 **fsmadm(1M) add** 指令新增管理工作站 (安裝了 **File System Manager** 的系統) 到可遠端管理此伺服器的主機清單中。

所有使用此指令新增到清單中的主機都可以遠端管理伺服器。所有其他主機都不可以遠端管理伺服器。

例如：

```
# fsmadm add management-station-name.domain-name
```

要確認管理工作站已成功新增，請使用 **fsmadm(1M) list** 指令並驗證您的管理工作站已列示在輸出中。

3. 以管理者使用者的身份登入至檔案系統管理員瀏覽器介面。
4. 從 **[Servers]** 頁面，按一下 **[Add]**。
接著便顯示 **[Add Server]** 視窗。
5. 在 **[Server Name]** 或 **[IP Address]** 欄位，請鍵入伺服器的名稱或輸入伺服器的 **IP** 位址。
6. 按一下 **[OK]**。

第2章

使用自動程式庫和手動載入磁碟機

自動程式庫是一種自動裝置，無需操作者手動操作即可載入或卸載可移除式卡匣。卡匣可以匯入程式庫並從其中匯出。它們均可自動載入與卸載。歸檔與分段備份程序使用站點定義的配置以分配使用的磁碟機數目。自動程式庫也可稱為媒體抽換裝置、光碟櫃、自動裝置、程式庫或媒體程式庫。

以下章節說明在 Sun StorEdge SAM-FS 環境中使用程式庫的各方面資訊。Sun StorEdge SAM-FS 安裝與升級指南 提供了初始配置指示，而本章則提供自動程式庫與手動載入裝置的作業指示。此外，本章還將說明以操作者為導向的載入通知工具，此工具在程式庫中沒有所需的磁碟區時向操作者發出警示。

備註 – Sun StorEdge SAM-FS 軟體可以與許多製造商的自動程式庫互通。請連絡 Sun 客戶支援以獲得有關程式庫型號、韌體等級，以及其他相容性資訊。

某些自動程式庫中的功能可能會使特定操作與本章中的說明有所不同。若要判斷當在 Sun StorEdge SAM-FS 環境中使用您的自動程式庫時，是否有供應商特定的作業指示，請檢查第 197 頁「使用供應商指定作業程序的程式庫基本作業」。

本章包含下列主題：

- 第 8 頁「慣例」
- 第 9 頁「自動程式庫作業」
- 第 26 頁「手動載入磁碟機作業」

慣例

本章中說明執行基本作業的程序，通常會顯示如何使用 `samcmd(1M)` 指令、`samu(1M)` 操作者公用程式，以及以下指令：

- `tplabel(1M)`
- `odlabel(1M)`
- `auditslot(1M)`
- `cleandrive(1M)`
- `chmed(1M)`
- `import(1M)`
- `set_state(1M)`
- `samexport(1M)`

在許多情況下，可以使用多種方法來執行說明的工作。您可以在 File System Manager 中執行這些工作，其為 Sun StorEdge QFS 和 Sun StorEdge SAM-FS 軟體的網頁圖形化使用者介面 (GUI)。您可以使用此介面來配置、控制、監視及重新配置 Sun StorEdge QFS 和 Sun StorEdge SAM-FS 環境的元件。如需安裝 File System Manager 的資訊，請參閱「Sun StorEdge SAM-FS 安裝與升級指南」。如需使用 File System Manager 的資訊，請參閱其線上說明。有關使用 `samu(1M)` 公用程式的資訊，請參閱「Sun StorEdge QFS 配置與管理指南」。

指令引數

許多指令接受普通的引數集。表 2-1 顯示這些引數。

表 2-1 指令引數

引數	意義
<code>eq</code>	要被定址的裝置之「設備序號」，如 <code>mcf</code> 檔案所定義。識別的裝置可以是自動程式庫、磁碟機或檔案系統。
插槽	表示在程式庫目錄檔中識別的自動程式庫中儲存插槽的號碼。
分割區	磁光碟的一面。分割區必須是 1 或 2。
媒體類型	表示媒體類型。如需有效媒體類型清單，請參閱 <code>mcf(4)</code> 線上手冊。
<code>vsn</code>	表示指定給磁碟區的磁碟區序列名稱。

視情況而定，某些指令可接受各種引數的組合。例如，在 `samu(1M)` 操作者公用程式中，`load` 指令具有以下兩種格式：

```
:load eq:slot  
:load media_type.vsn
```

請注意以下內容：

- 第一種形式使用冒號 (:) 分隔 *eq* 和插槽。
- 第二種形式使用小數點 (.) 分隔媒體類型和 *vsn*。

專有名詞

本章中使用的某些專有名詞對您來說可能是新的專有名詞。表 2-2 顯示最常用的某些專有名詞及其意義。

表 2-2 專有名詞

專有名詞	意義
自動程式庫	用於儲存磁帶與光學卡匣的自動裝置。
卡匣	磁帶或磁光碟匣。磁光碟匣可以包含一個或多個磁碟區或分割區。
分割區	整個磁帶或磁光碟的一面。一個分割區只包含一個磁碟區。
磁碟區	卡匣上用於儲存資料的命名區域。一個卡匣有一個或多個磁碟區。雙面卡匣有兩個磁碟區，一面一個。磁碟區序列名稱 (VSN) 可以識別磁碟區。

自動程式庫作業

一些基本操作在所有自動程式庫上都是基本相同的。本章節將解釋以下基本作業：

- 第 11 頁 「啓動可移除式媒體作業」
- 第 10 頁 「停止可移除式媒體作業」
- 第 11 頁 「開啓自動程式庫」
- 第 11 頁 「關閉自動程式庫」
- 第 12 頁 「將卡匣載入自動程式庫」
- 第 13 頁 「從磁碟機中卸載卡匣」
- 第 13 頁 「製作卡匣標籤」

- 第 15 頁 「稽核磁碟區」
- 第 16 頁 「稽核自動程式庫 (僅限於直接連結式)」
- 第 16 頁 「使用清理卡匣」
- 第 18 頁 「清理磁帶機」
- 第 19 頁 「清除媒體錯誤」
- 第 20 頁 「從磁碟機中移除卡住的卡匣」
- 第 22 頁 「目錄檔作業，匯入和匯出卡匣」
- 第 25 頁 「啓動載入通知」

▼ 停止可移除式媒體作業

您可以停止可移除式媒體作業，並將 Sun StorEdge SAM-FS 系統維持在掛載狀態。例如，如果要手動操作程式庫中的卡匣，您即可執行此操作。在繼續執行作業時，待處理的分段備份將重新發出且歸檔將繼續執行。

- 若要停止可移除式媒體作業，請使用 `samcmd(1M) idle` 和 `samd(1M) stop` 指令。以下列格式使用這些指令：

```
samcmd idle eq  
samd stop
```

關於 `eq` 值，請輸入要被定址的設備之「設備序號」，如 `mcf` 檔案所定義。若要閒置磁碟機，請為 `mcf` 檔案中配置的每個 `eq` 輸入 `samcmd idle eq` 指令。

您也可以使用 `samu(1M)` 操作者公用程式或 File System Manager 閒置磁碟機。

備註 – 在執行 `samd(1M) stop` 指令之前，應該先閒置 Sun StorEdge SAM-FS 環境中的磁碟機。這將允許歸檔器、分段備份程式及其他程序完成目前工作。當繼續歸檔、分段備份和其他作業時，`samd(1M) stop` 指令執行失敗會造成未預期的結果。

▼ 啓動可移除式媒體作業

通常當掛載 Sun StorEdge SAM-FS 檔案系統時，會開始可移除式媒體作業。

- 要手動啟動可移除式媒體作業而無需掛載任何檔案系統，請輸入 `samd(1M) start` 指令。請用以下格式使用此指令：

```
# samd start
```

如果已在輸入上述指令之前執行可移除式媒體操作，則將產生下列訊息：

```
SAM-FS sam-amld 常駐程式已經在執行。
```

如需有關 `samd(1M)` 指令的詳細資訊，請參閱 `samd(1M)` 線上手冊。

▼ 開啓自動程式庫

當程式庫處於 `on` 狀態時，會受到 Sun StorEdge SAM-FS 系統的控制，並且可以使用一般作業繼續。當開啓程式庫時，Sun StorEdge SAM-FS 軟體會執行以下動作：

- 查詢裝置內部狀態。它將尋找磁帶位置、是否使用條碼等等。
- 更新目錄及其他內部結構。
- 使用 `samcmd(1M) on` 指令開啟自動程式庫。

請用以下格式使用此指令：

```
samcmd on eq
```

關於 `eq` 值，請指定要被定址的自動程式庫之「設備序號」，如 `mcf` 檔案所定義。

您也可以使用 `samu(1M)` 或 File System Manager 執行此工作。

▼ 關閉自動程式庫

將程式庫置於 `off` 狀態會停止 I/O 作業，並且從 Sun StorEdge SAM-FS 控制中移除自動程式庫。卡匣不會自動移動。請注意，自動程式庫中的磁碟機將保持在 `on` 狀態。您可能要關閉自動程式庫以執行以下工作：

- 僅針對此自動程式庫停止 Sun StorEdge SAM-FS 作業。
- 關閉自動程式庫的電源。

- 使用 `samcmd(1M) off` 指令關閉自動程式庫。

請用以下格式使用此指令：

```
samcmd off eq
```

關於 `eq` 值，請指定要被定址的自動程式庫之「設備序號」，如 `mcf` 檔案所定義。

您也可以使用 `samu(1M)` 或 File System Manager 執行此工作。

▼ 將卡匣載入自動程式庫

將一個卡匣載入在歸檔或分段備份請求 `VSN` 時會自動出現的磁碟機中。載入是指將卡匣從儲存插槽移至磁碟機並使其就緒。

- 使用 `samcmd(1M) load` 指令手動載入卡匣。

即使磁碟機處於 `unavail` 狀態，您也可以使用此指令。此指令具有以下兩種可能的格式：

```
samcmd load eq:插槽[:分割區]
samcmd load 媒體類型.vsn
```

表 2-3 `samcmd(1M) load` 的引數

引數	意義
<code>eq</code>	要被定址的磁帶之「設備序號」，如 <code>mcf</code> 檔案所定義。
插槽	表示在程式庫目錄檔中識別的自動程式庫中儲存插槽的號碼。
媒體類型	表示媒體類型。如需有效媒體類型清單，請參閱 <code>mcf(4)</code> 線上手冊。
分割區	磁光碟的一面。分割區必須是 1 或 2。此引數不適用在磁帶卡匣。
<code>vsn</code>	表示指定給磁碟區的磁碟區序列名稱。

您也可以使用 `samu(1M)` 或 File System Manager 執行此工作。

在手動載入卡匣時，它一般會載入程式庫中的下一個可用磁碟機中。若要使磁碟機無法用於此目的，請使用 `samu(1M)` 公用程式的 `:unavail` 指令或變更使用 File System Manager 的裝置狀態。例如，您可以在災難復原作業或分析磁帶時執行此作業。

▼ 從磁碟機中卸載卡匣

當不再需要磁碟區時會自動卸載卡匣。您也可以手動卸載磁碟機。卸載是指從磁碟機中取出卡匣。

- 使用 `samcmd(1M) unload` 指令手動卸載卡匣。

即使磁碟機處於 `unavail` 狀態，您也可以使用此指令。請用以下格式使用此指令：

```
samcmd unload eq
```

關於 `eq` 值，請指定要被定址的自動程式庫之「設備序號」，如 `mcf` 檔案所定義。

您也可以使用 `samu(1M)` 或 File System Manager 執行此工作。

製作卡匣標籤

視爲磁帶卡匣或光學卡匣製作標籤而定，製作卡匣標籤的程序會有所不同。以下兩個章節將說明這些程序。



注意 – 製作和重新製作卡匣標籤會使任何軟體無法存取目前卡匣上的資料。只有在確定不需要卡匣中儲存資料的情況下再重新製作卡匣標籤。

▼ 製作或重新製作磁帶標籤

以下 `tplabel(1M)` 指令行格式顯示了製作或重新製作磁帶標籤時最常用的選項：

```
tplabel [ -new | -old vsn ] -vsn vsn eq:插槽
```

表 2-4 `tplabel(1M)` 的引數

引數	意義
<code>vsn</code>	磁碟區序列名稱。如果重新製作標籤，新的 VSN 名稱可以與舊的 VSN 名稱相同。
<code>eq</code>	要被定址的自動程式庫或手動載入的磁碟機之「設備序號」，如 <code>mcf</code> 檔案所定義。
插槽	表示在程式庫目錄檔中識別的自動程式庫中儲存插槽的號碼。此引數不適用於手動載入磁碟機。

- 若要製作新磁帶標籤，請使用 `tplabel(1M)` 指令。
請用以下格式使用此指令：

```
tplabel -new -vsn vsn eq:插槽
```

- 若要重新製作現有磁帶標籤，請使用 `tplabel(1M)` 指令。
請用以下格式使用此指令：

```
tplabel -old vsn -vsn vsn eq:插槽
```

在發出指令以製作或重新製作磁帶標籤後，磁帶將會載入並定位，然後寫入磁帶標籤。如需有關 `tplabel(1M)` 指令的詳細資訊，請參閱 `tplabel(1M)` 線上手冊。

您也可以使用 File System Manager 執行此工作。

▼ 製作或重新製作光碟標籤

以下 `odlabel(1M)` 指令行格式顯示了製作或重新製作光碟標籤時最常用的選項：

```
odlabel [ -new | -old vsn ] -vsn vsn eq:插槽:分割區
```

表 2-5 `odlabel(1M)` 的引數

引數	意義
<code>vsn</code>	磁碟區序列名稱。如果重新製作標籤，新的 VSN 名稱可以與舊的 VSN 名稱相同。
<code>eq</code>	要被定址的自動程式庫或手動載入的磁碟機之「設備序號」，如 <code>mcf</code> 檔案所定義。
插槽	表示在程式庫目錄檔中識別的自動程式庫中儲存插槽的號碼。此引數不適用於手動載入磁碟機。
分割區	磁光碟的一面。分割區必須是 1 或 2。此引數不適用於磁帶卡匣。

- 若要製作新光碟標籤，請使用 `odlabel(1M)` 指令。
請用以下格式使用此指令：

```
odlabel -new -vsn vsn eq:插槽:分割區
```

- 若要重新製作現有光碟標籤，請使用 `odlabel(1M)` 指令。
請用以下格式使用此指令：

```
odlabel -old vsn -vsn vsn eq:插槽:分割區
```

在發出指令以製作或重新製作光碟標籤後，光碟將會載入並定位，然後寫入光碟標籤。如需有關 `odlabel(1M)` 指令的詳細資訊，請參閱 `odlabel(1M)` 線上手冊。

您也可以使用 File System Manager 執行此工作。

▼ 稽核磁碟區

磁帶或光學卡匣上剩餘的報告空間偶爾可能需要在程式庫目錄中進行更新。`auditslot(1M)` 指令將載入包含磁碟區的卡匣、讀取標籤以及更新插槽的程式庫目錄項目。

- 使用 `auditslot(1M)` 指令稽核磁碟區。
請用以下格式使用此指令：

```
auditslot [-e] eq:插槽[:分割區]
```

表 2-6 `auditslot(1M)` 的引數

引數	意義
<code>-e</code>	如果指定 <code>-e</code> 選項並且媒體是磁帶，便會更新剩餘空間。否則，其不會變更。
<code>eq</code>	要被定址的自動程式庫或手動載入的磁碟機之「設備序號」，如 <code>mcf</code> 檔案所定義。
插槽	表示在程式庫目錄檔中識別的自動程式庫中儲存插槽的號碼。此引數不適用於手動載入磁碟機。
分割區	磁光碟的一面。分割區必須是 1 或 2。此引數不適用於磁帶卡匣。

如需有關 `auditslot(1M)` 指令的詳細資訊，請參閱 `auditslot(1M)` 線上手冊。

您也可以使用 `samu(1M)` 公用程式的 `:audit` 指令或使用 File System Manager 執行此工作。

▼ 稽核自動程式庫 (僅限於直接連結式)

備註 – 此工作無法在網路連結式自動程式庫中執行。

一個完整的稽核會將每個卡匣載入磁碟機、讀取標籤並更新程式庫目錄。在以下情況下應該稽核程式庫：

- 在不使用 Sun StorEdge SAM-FS 指令的情況下移動自動程式庫中的卡匣後。
 - 如果您懷疑程式庫目錄的狀態並想更新它 (例如：停電之後)。
 - 在沒有信箱的自動程式庫中新增、移除或移動了卡匣。
- 使用 `samcmd(1M) audit` 指令在自動程式庫上執行完整稽核。
請用以下格式使用此指令：

```
samcmd audit eq
```

關於 `eq` 值，請指定要被定址的自動程式庫之「設備序號」，如 `mcf` 檔案所定義。

您也可以使用 `samu(1M)` 公用程式的 `:audit` 指令或使用 File System Manager 執行此工作。

使用清理卡匣

Sun StorEdge SAM-FS 系統允許您匯入清理卡匣以清理磁碟機。視清理卡匣是否已製作條碼而定，此程序會有所不同。以下章節將解釋使用清理卡匣的各個方面。

清理的實際操作會因製造商而異。如果對此不清楚的話，請參閱第 197 頁「使用供應商指定作業程序的程式庫基本作業」以確定是否具有建議用於您的設備的特定程序。

備註 – 此工作無法在網路連結式自動程式庫中執行。

▼ 重設清理週期數目

僅對於有限的清理週期數目，清理卡匣才非常有用。您可以使用 `samu(1M)` 公用程式的 `:v` 顯示，或從 File System Manager 檢視剩餘週期數。

Sun StorEdge SAM-FS 系統將追蹤用於每個清理卡匣的清理週期數目，並在剩餘週期為零時退出磁帶。例如，DLT 清理卡匣具有 20 個週期，而 Exabyte 清理卡匣具有 10 個週期。每次匯入清理卡匣後，清理週期將重設為該類磁帶的最高週期數目。

如果您的系統可進行自動清理，但自動程式庫中所有清理卡匣的週期數目均為零，則磁碟機將設定為關閉且在 Sun StorEdge SAM-FS 錄檔中發出一則訊息。

- 請使用 `chmed(1M)` 指令將清理卡匣計數重設為零。
請用以下格式使用此指令：

```
chmed -count 計數 媒體類型.vsn
```

表 2-7 `chmed(1M)` 的引數

引數	意義
計數	要重設的清理卡匣之清理週期數目。
媒體類型	表示媒體類型。如需有效媒體類型清單，請參閱 <code>mcf(4)</code> 線上手冊。
<i>vsn</i>	表示指定給磁碟區的磁碟區序列名稱。

▼ 搭配條碼使用清理卡匣

如果清理卡匣已製作條碼，您可以使用 `import(1M)` 指令將其匯入。

1. 請確定清理卡匣有 CLEAN 條碼，或是以 CLN 字母開頭。
2. 使用 `import(1M)` 指令匯入清理卡匣。

請用以下格式使用此指令：

```
import eq
```

關於 `eq` 值，請指定要被定址的自動程式庫之「設備序號」，如 `mcf` 檔案所定義。

Sun StorEdge SAM-FS 系統會將卡匣從信箱移至儲存插槽，並更新每個卡匣的程式庫目錄。此外，在發出此指令後會設定清理媒體旗標，並且存取數目將根據媒體類型設定為適當的清理週期數目。每次使用媒體清理磁碟機，存取數目就會減少一次。

例如，以下指令可將清理卡匣匯入 `mcf` 檔案中編號為 50 的自動程式庫：

```
# import 50
```

您也可以使用 `samu(1M)` 或 File System Manager 執行此工作。

▼ 不搭配條碼使用清理卡匣

如果卡匣未製作條碼，則必須先將其匯入。它不會標記為清理卡匣。請執行以下步驟：

1. 使用 `import(1M)` 指令匯入卡匣。

請用以下格式使用此指令：

```
import eq
```

關於 *eq* 值，請指定要被定址的自動程式庫之「設備序號」，如 *mcf* 檔案所定義。

2. 使用 **chmed(1M)** 指令將類型變更為清理卡匣。

您必須知道自動程式庫與要載入清理卡匣的插槽之「設備序號」。

在以下指令行範例中，自動程式庫的「設備序號」為 50，清理卡匣位於插槽 77：

```
# chmed +C 50:77
```

以上指令會將卡匣類型變更為清理卡匣的類型。

3. 再次使用 **chmed(1M)** 指令以設定清理週期數目。

以下指令範例將設定上述步驟中使用的卡匣上的數目：

```
# chmed -count 20 50:77
```

如需有關 **chmed(1M)** 指令的詳細資訊，請參閱 **chmed(1M)** 線上手冊。

▼ 清理磁帶機

備註 – Sun StorEdge SAM-FS 系統並不支援自動清理網路連結式程式庫。您必須使用供應商的程式庫管理員軟體進行自動清理。

如果硬體支援清理卡匣，Sun StorEdge SAM-FS 環境便支援使用清理卡匣。如果磁碟機要求進行清理，系統將自動載入清理卡匣。

如果系統使用已製作條碼的標籤，清理卡匣條碼標籤中的 **VSN** 必須為 **CLEAN** 或以 **CLN** 字母開頭。另外，您也可以使用 **chmed(1M)** 指令將 **VSN** 標記為清理卡匣，並設定清理週期數目。一個系統中可以載入多個清理卡匣。

備註 – 某些磁碟機錯誤可能會導致系統重複載入清理卡匣，直至所有清理週期均耗盡。您可以使用 **chmed(1M)** 指令來限制清理卡匣上的清理週期數目，以防止發生此情況。例如：

```
# chmed -count 20 50:77
```

當無法使用自動清理並且系統使用條碼時，請執行以下程序手動要求清理磁碟機：

- 使用 `cleandrive(1M)` 指令。

請用以下格式使用此指令：

```
cleandrive eq
```

關於 `eq` 值，請指定要被定址的自動程式庫之「設備序號」，如 `mcf` 檔案所定義。此磁碟機是要與清理卡匣一起載入的磁碟機。

磁帶機自動清理

以 Sun StorEdge SAM-FS 4U4 開始，由軟體啟動的磁帶機清理之預設設定為 `off`，強制使用者選擇由硬體或軟體啟動的磁帶機清理策略。

硬體啟動的清理策略使用媒體抽換裝置的內建自動清理功能。為了使用此功能，清理卡匣可能需要放入到特定的插槽。請參閱製造商的文件以取得相關指示。

軟體啟動的清理策略使用 Sun StorEdge SAM-FS 自動清理功能。`logsense` 選項是現有功能的增強功能，可避免磁碟機使用過期的清理媒體。要啓用 Sun StorEdge SAM-FS 自動清理功能，應該停用硬體啟動的清理且下行應顯示在 `defaults.conf` 檔案中：

```
tapeclean = all autoclean on logsense on
```

要呼叫舊的僅依賴感應資料取得磁碟機清理狀態的 Sun StorEdge SAM-FS 自動清理功能，下行應顯示在 `defaults.conf` 檔案中：

```
tapeclean = all autoclean on logsense off
```

備註 – 當具有兩個以上磁碟機的程式庫使用自動清理功能，建議您在每個 Sun StorEdge SAM-FS 目錄中都至少有兩個清理卡匣。如果有需要清理的磁碟機，但是沒有足夠的清理卡匣，那些磁碟機會被放置到 `DOWN` 狀態。

▼ 清除媒體錯誤

當卡匣發生硬體或軟體錯誤時，Sun StorEdge SAM-FS 系統會在 `VSN` 目錄檔中設定 `media error` 旗標。在任何產生 `media error` 信號的指定卡匣上，您可以使用 `chmed(1M)` 指令清除錯誤，然後可以嘗試使用該卡匣。`media error` 旗標會顯示在 `samu(1M)` 公用程式的 `v` 畫面以及 File System Manager 中。

1. 使用 **chmed(1M)** 指令清除 **media error** 旗標。
以下列格式使用此指令清除 **media error** 旗標：

```
chmed -E 媒體類型.usn
```

表 2-8 chmed(1M) 的引數

引數	意義
媒體類型	表示媒體類型。如需有效媒體類型清單，請參閱 mcf(4) 線上手冊。
usn	表示指定給磁碟區的磁碟區序列名稱。

2. 執行 **auditslot(1M)** 指令更新剩餘空間資訊。
請用以下格式使用此指令：

```
auditslot -e eq:插槽[:分割區]
```

表 2-9 auditslot(1M) 的引數

引數	意義
-e	如果指定 -e 選項並且媒體是磁帶，便會更新剩餘空間。否則，其不會變更。
eq	要被定址的自動程式庫或手動載入的磁碟機之「設備序號」，如 mcf 檔案所定義。
插槽	表示在程式庫目錄檔中識別的自動程式庫中儲存插槽的號碼。此引數不適用於手動載入磁碟機。
分割區	磁光碟的一面。分割區必須是 1 或 2。此引數不適用於磁帶卡匣。

如需有關 **auditslot(1M)** 指令的詳細資訊，請參閱 **auditslot(1M)** 線上手冊。

您也可以使用 **samu(1M)** 公用程式的 **:audit** 指令或使用 **File System Manager** 執行此工作。

▼ 從磁碟機中移除卡住的卡匣

如果卡匣被卡在磁碟機中，請遵循以下步驟。

1. 使用 **samcmd(1M)** **off** 指令關閉自動程式庫中的磁碟機。
請用以下格式使用此指令：

```
samcmd off eq
```

關於 *eq* 值，請指定要被定址的磁碟機之「設備序號」，如 *mcf* 檔案所定義。您也可以使用 *samu(1M)* 或 File System Manager 執行此工作。

2. 使用 *samcmd(1M) off* 指令關閉自動程式庫。

請用以下格式使用此指令：

```
samcmd off eq
```

關於 *eq* 值，請指定要被定址的程式庫之「設備序號」，如 *mcf* 檔案所定義。您也可以使用 *samu(1M)* 或 File System Manager 執行此工作。

3. 從磁碟機中實際移除卡匣。

確定卡匣或磁碟機未損毀。

4. 使用 *samcmd(1M) on* 指令開啟自動程式庫以及磁碟機。

針對磁碟機以及程式庫各執行一次此指令。請用以下格式使用此指令：

```
samcmd on eq
```

關於 *eq* 值，請指定要被定址的程式庫或磁碟機之「設備序號」，如 *mcf* 檔案所定義。如果自動程式庫在開啓時執行稽核，則您已完成所有步驟。否則，請執行下一步驟。

5. 如果將卡匣裝回其儲存插槽，請使用 *chmed(1M)* 指令調整程式庫目錄，以設定損毀磁帶的被佔用旗標。

請用以下格式使用此指令：

```
chmed +o eq:插槽
```

表 2-10 *chmed(1M)* 的引數

引數	意義
<i>eq</i>	要被定址的自動程式庫或磁碟機之「設備序號」，如 <i>mcf</i> 檔案所定義。
插槽	表示在程式庫目錄檔中識別的程式庫中儲存插槽的數量。此引數不適用於手動載入磁碟機。

如需有關 *chmed(1M)* 指令的詳細資訊，請參閱 *chmed(1M)* 線上手冊。

如果將卡匣留在外面，並想稍後將其裝回，則必須將卡匣匯入自動程式庫。

目錄檔作業，匯入和匯出卡匣

實際從自動程式庫新增 (匯入) 或移除 (匯出) 卡匣可以讓您執行幾項功能，其中包含以下功能：

- 您可以替換卡匣。
- 您可以將卡匣重新配置到離站儲存裝置中，以便於未來災難復原時使用。如果這是您的工作，可以使用 `chmed(1M)` 指令的 `-I` 選項指定像是卡匣儲存位置的額外資訊。

當匯入和匯出卡匣時，您也可以更新程式庫目錄檔。在 Sun StorEdge SAM-FS 系統中，您可以使用 `import(1M)` 和 `samexport(1M)` 指令完成這些作業。您也可以使用 File System Manager 執行這些工作。

程式庫目錄是 Sun StorEdge SAM-FS 環境尋找自動程式庫中的卡匣時，所需要的所有資訊集中儲存庫。程式庫目錄檔是包含自動程式庫中每個插槽資訊的 UFS 常駐二進位檔案。此檔案中的資訊包括：與插槽中儲存的卡匣相關的一個或多個磁碟區序列名稱 (VSN)、卡匣中的剩餘容量與空間、表示唯讀、防寫保護、回收的旗標，以及有關卡匣的其他狀態資訊。

Sun StorEdge SAM-FS 環境會根據自動程式庫連接伺服器的方式，以如下所示的不同方法處理目錄：

- 如果是直接連結式自動程式庫，則程式庫目錄在程式庫目錄項目與自動程式庫中的實體插槽之間是一對一對映的。程式庫目錄中的第一條項目對應於自動程式庫中的第一個插槽。在需要卡匣時，系統將查閱程式庫目錄檔以確定包含 VSN 的插槽，並發出指令以將卡匣從該插槽載入磁碟機。
- 如果是網路連結式自動程式庫，則程式庫目錄並非直接對映至插槽。而是已知安裝在自動程式庫中的 VSN 之清單。在需要卡匣時，系統會傳送請求至供應商軟體以將 VSN 下載至磁碟機。供應商軟體將找出 VSN 儲存插槽所在位置。

由於系統特性與供應商提供的軟體不同，每個自動程式庫處理卡匣匯入與匯出的方式也會不同。例如，在 ACL 4/52 程式庫中，您需要在從自動程式庫中匯出卡匣之前，執行 `move` 指令以便將卡匣移至匯入/匯出單元中。

備註 – 網路連結式自動程式庫使用專屬公用程式匯入與匯出卡匣，因此 `import(1M)` 和 `samexport(1M)` 指令只會更新 Sun StorEdge SAM-FS 系統使用的程式庫目錄項目。如果您有網路連結式程式庫，請參閱第 197 頁「使用供應商指定作業程序的程式庫基本作業」以取得有關匯入與匯出卡匣的資訊。

追蹤匯出的媒體 — 記錄器

Sun StorEdge SAM-FS 記錄器會持續追蹤自動程式庫匯出的卡匣，或手動掛載的裝置。記錄器的功能與虛擬程式庫相同，但它沒有定義的硬體裝置。與自動程式庫相同，它也是在 `mcf` 檔案中配置、具有記錄與其相關的所有卡匣項目的目錄、可以匯入與匯出卡匣，並可以在 File System Manager 中顯示為另一個自動程式庫。

記錄器可以使用 `hy` 裝置類型在 `mcf` 檔案中配置。如果並未在 `mcf` 檔案中配置記錄器，便會使用如下方式建立：

```
historian n+1 hy - on /var/opt/SUNWsamfs/catalog/historian
```

在之前的項目中，`n+1` 是 `mcf` 檔案中的最後一個「設備序號」加 1。如果想要使用不同的目錄檔「設備序號」或路徑名稱，只需要在 `mcf` 中定義記錄器。

記錄器程式庫目錄檔在記錄器第一次啟動時初始化並具有 32 個項目。確定目錄檔位於足夠大的檔案系統（可以容納整個目錄檔）中。您的站點可能想要追蹤程式庫匯出的現有 Sun StorEdge SAM-FS 卡匣。在此情況下，您需要按照 `build_cat(1M)` 線上手冊中所述，從現有卡匣建立記錄器目錄。

`defaults.conf` 檔案中的以下兩種配置指令可以影響記錄器的運作方式：

- 如果出現 `exported_media = unavailable` 指令，則所有從自動程式庫匯出的卡匣均會標記為記錄器無法使用。標記為無法使用的卡匣之請求將產生 EIO 錯誤。
- 如果出現 `attended = no` 指令，其將告訴記錄器沒有處理載入請求的操作者。記錄器已知的載入卡匣與尚未載入的卡匣之請求將產生 EIO 錯誤。

如需配置詳細資訊，請參閱 `historian(7)` 和 `defaults.conf(4)` 線上手冊。

從自動程式庫匯入與匯出

信箱是自動程式庫中用於新增或從中移除卡匣的區域。`import(1M)` 指令可將卡匣從信箱移至儲存插槽。`samexport(1M)` 指令可將卡匣從儲存插槽移至信箱。對於大多數程式庫，如果卡匣在 Sun StorEdge SAM-FS 軟體啟動時即存在於信箱中，則軟體將在啟動後自動匯入卡匣。

匯入與匯出實際操作會因製造商而異。如果對此不清楚的話，請參閱第 197 頁「使用供應商指定作業程序的程式庫基本作業」以確定是否具有建議用於您的設備的特定程序。

以下章節說明匯入與匯出卡匣：

- 第 23 頁「從使用信箱的程式庫匯入卡匣」
- 第 24 頁「從使用信箱的程式庫匯出卡匣」
- 第 24 頁「自不使用信箱的程式庫匯入卡匣」
- 第 25 頁「自不使用信箱的程式庫匯出卡匣」

▼ 從使用信箱的程式庫匯入卡匣

要將卡匣匯入使用信箱的自動程式庫，請遵循以下步驟。

1. 使用製造商建議的作業來開啟信箱。

信箱旁邊通常會有一個按鈕。信箱有時是單插槽信箱，在供應商的文件資料中被稱為郵件插槽。

2. 將卡匣手動放入信箱。
3. 關閉信箱。
4. 使用 `import(1M)` 指令匯入卡匣。

請用以下格式使用此指令：

```
import eq
```

關於 `eq` 值，請指定要被定址的程式庫之「設備序號」，如 `mcf` 檔案所定義。系統會將卡匣從信箱移至儲存插槽，並更新每個卡匣的程式庫目錄檔。

您也可以使用 `samu(1M)` 或 `File System Manager` 執行此工作。

▼ 從使用信箱的程式庫匯出卡匣

此程序可將卡匣從儲存插槽移至信箱或郵件插槽。要從使用信箱的程式庫匯出 (退出) 卡匣，請按照以下步驟執行。

1. 使用 `samexport(1M)` 指令將卡匣從儲存插槽移至信箱。

以下列格式之一使用此指令：

```
samexport eq:插槽  
samexport 媒體類型.vsn
```

表 2-11 `samexport(1M)` 的引數

引數	意義
<code>eq</code>	要被定址的自動程式庫之「設備序號」，如 <code>mcf</code> 檔案所定義。
插槽	表示在程式庫目錄檔中識別的自動程式庫中儲存插槽的號碼。
媒體類型	表示媒體類型。如需有效媒體類型清單，請參閱 <code>mcf(4)</code> 線上手冊。
<code>vsn</code>	表示指定給磁碟區的磁碟區序列名稱。

您也可以使用 `samu(1M)` 或 `File System Manager` 執行此步驟。

2. 使用製造商建議的作業來開啟信箱或郵件插槽。

信箱旁邊通常會有一個按鈕。

▼ 自不使用信箱的程式庫匯入卡匣

1. 使用 `samcmd(1M) unload` 指令。

請用以下格式使用此指令：

```
samcmd unload eq
```

關於 *eq* 值，請指定要被定址的程式庫之「設備序號」，如 *mcf* 檔案所定義。

待系統完成其目前工作後，將其狀態設定為關閉，然後將目前已啟動的目錄檔傳輸至記錄器。

2. 解除鎖定並打開自動程式庫的門蓋。
3. 將卡匣載入可用插槽。
4. 合上並鎖定自動程式庫的門蓋。

自動程式庫將重新初始化並掃描程式庫中的卡匣。Sun StorEdge SAM-FS 軟體會更新程式庫目錄檔，將匯入卡匣的 VSN 新增至目錄。自動程式庫的狀態將設定為 on。

▼ 自不使用信箱的程式庫匯出卡匣

1. 使用 **samcmd(1M) unload** 指令。

請用以下格式使用此指令：

```
samcmd unload eq
```

關於 *eq* 值，請指定要被定址的程式庫之「設備序號」，如 *mcf* 檔案所定義。

待系統完成其目前工作後，將其狀態設定為關閉，然後將目前已啟動的目錄檔傳輸至記錄器。

2. 解除鎖定並打開自動程式庫的門蓋。
3. 將卡匣從各自的插槽中移除。
4. 合上並鎖定自動程式庫的門蓋。

自動程式庫將重新初始化並掃描自動程式庫中的卡匣。系統將使用目前程式庫插槽中卡匣的 VSN 來更新程式庫目錄檔。被移除卡匣的 VSN 已從程式庫目錄檔中移除，且現在只記錄於記錄器檔案中。自動程式庫的狀態將設定為 on。

▼ 啟動載入通知

Sun StorEdge SAM-FS 軟體會請求經常載入卡匣以滿足歸檔與分段備份請求。如果請求載入位於程式庫內部的卡匣，則系統會自動處理請求。如果請求載入位於程式庫外部的卡匣，則需要操作者進行操作。如果已啟動，當需要從程式庫外取得卡匣時，`load_notify.sh(1M)` 程序檔會傳送電子郵件。

1. 成為超級使用者。
2. 使用 `cp(1)` 指令將載入通知程序檔從其安裝位置複製到可操作位置。
例如：

```
# cp /opt/SUNWsamfs/examples/load_notify.sh
/etc/opt/SUNWsamfs/scripts/load_notify.sh
```

3. 使用 `more(1)` 或另一個指令來檢查 `defaults.conf` 檔案。
請確定檔案中存在以下指令並且如下所示：
 - `exported_media=available`
 - `attended=yes`這些指令是依據預設而設定。如果載入通知功能已啟動，請確定這些指令未作任何變更。
4. 修改 `load_notify.sh` 程序檔以便將通知傳送給操作者。
依據預設，程序檔會將電子郵件傳送至 `root`，但是也可編輯它以將電子郵件傳送至其他人員，或撥打呼叫器，或提供某些其他通知方式。

手動載入磁碟機作業

如果您有手動載入的獨立磁碟機且並非自動程式庫，本章節將說明其不同於其他情況的操作。每個手動載入的磁碟機均有自己的單插槽程式庫目錄檔。

▼ 載入卡匣

- 若要將卡匣載入手動載入裝置，請根據製造商的指示將卡匣放入磁碟機。
Sun StorEdge SAM-FS 系統會識別已載入卡匣、讀取標籤，然後更新說明手冊與一個插槽的目錄檔。無需採取其他動作。

▼ 卸載卡匣

- 使用 `samcmd(1M) idle` 指令以閒置磁碟機。
本指令可確保沒有已啟動的歸檔或分段備份程序。請用以下格式使用此指令：

```
samcmd idle eq
```

關於 *eq* 值，請指定要被定址的磁碟機之「設備序號」，如 *mcf* 檔案所定義。

在所有 I/O 作業完成時，磁碟機將從 *idle* 切換至 *off*，然後退出磁帶。

如果是磁帶的話，可以倒帶且移除卡匣。光學卡匣會自動退出。請參閱製造商對有關移除特定卡匣的指示。

您也可以使用 *samu(1M)* 或 *File System Manager* 執行此工作。

▼ 檢視程式庫目錄檔

- 使用 *samu(1M)* 公用程式的 *:v* 指令。

請用以下格式使用此指令：

```
:v eq
```

關於 *eq* 值，請指定要被定址的程式庫之「設備序號」，如 *mcf* 檔案所定義。

第3章

歸檔

歸檔是將檔案從 Sun StorEdge SAM-FS 檔案系統複製到磁碟區的過程，該磁碟區位於可移除的媒體匣或其他系統的磁碟分割區中。在本章中，**歸檔媒體**這個專有名詞指的是歸檔磁碟區寫入的各種卡匣或磁碟區塊。Sun StorEdge SAM-FS 歸檔功能包括許多功能，如那些可用於指定某個檔案立即歸檔、永不歸檔及執行其他工作等功能。

本章將說明歸檔器的作業理論，提供對開發站點歸檔規則的一般指引，及解釋如何透過建立 `archiver.cmd` 檔案以執行規則。

介紹下列主題：

- 第 29 頁 「歸檔程序簡介」
- 第 39 頁 「關於 `archiver.cmd` 檔案」
- 第 43 頁 「使用歸檔器指令」
- 第 78 頁 「關於磁碟歸檔」
- 第 84 頁 「規劃歸檔作業」
- 第 86 頁 「歸檔器範例」

歸檔程序簡介

歸檔器會自動將 Sun StorEdge SAM-FS 檔案寫入到歸檔媒體。無需操作者手動操作即可歸檔與分段備份檔案。檔案將歸檔至歸檔媒體上的磁碟區，每個磁碟區由稱為**磁碟區序列名稱 (VSN)**的唯一識別標記識別。歸檔媒體可包含一個或多個磁碟區。要識別個別磁碟區，必須指定媒體類型與 VSN。

在掛載 Sun StorEdge SAM-FS 檔案系統時，歸檔器就會自動開始執行。您可將歸檔指令加入下列檔案，以自訂站點的歸檔器操作。

```
/etc/opt/SUNWsamfs/archiver.cmd
```

archiver.cmd 檔案無需為要進行的歸檔保留。在沒有這個檔案的時候，歸檔器會採用以下預設：

- 所有檔案將歸檔至可用的磁碟區。
- 所有檔案的歸檔時間是 4 分鐘。歸檔時間是自檔案上次修改後的時間。
- 歸檔間隔是 10 分鐘。歸檔間隔是完成歸檔處理所需的時間。

下列章節將說明歸檔組概念，並解釋在歸檔程序時執行的操作。

歸檔組

歸檔組可識別一組要歸檔的檔案。歸檔組可透過任何檔案系統群組定義。歸檔組中的檔案將共用與大小、所有權、群組或目錄位置有關的一般條件。歸檔組可控制歸檔備份的目標、保留歸檔備份多久及在歸檔資料之前要等待多久。歸檔組中的所有檔案將複製到與此歸檔組相關的磁碟區。檔案系統中的檔案可以是一個且僅是一個歸檔組的成員。

在建立與修改檔案時，歸檔器會將它們複製到歸檔媒體。歸檔檔案與標準的 UNIX tar(1) 格式相容。這可確定資料與 Sun Solaris 作業系統 (OS) 及其他 UNIX 系統相容。此格式包括檔案存取資料 (inode 資訊) 與檔案路徑。如果您的 Sun StorEdge SAM-FS 環境發生完全消失的狀況，tar(1) 格式會允許使用標準 UNIX 工具和指令來修復檔案。此歸檔程序也會複製 Sun StorEdge SAM-FS 檔案系統作業所需的資料。此資料包含目錄、符號連結、區段檔案索引及歸檔媒體資訊。

在本章節的其他部分，檔案這個專有名詞指的是檔案資料與中介資料 (metadata)。僅在需要區分時使用檔案資料與中介資料等專有名詞。專有名詞檔案系統指的是一個掛載的 Sun StorEdge SAM-FS 檔案系統。

歸檔組名稱由管理員決定且實際上不受限制，除了下列例外情況：

- 有兩個預留的歸檔組名稱：no_archive 和 allsets。
 - 依據預設，系統將定義 no_archive 歸檔組。在此歸檔組中選擇的檔案永不歸檔。例如，暫存目錄 (如 /sam1/tmp) 中的檔案可能包含在 no_archive 歸檔組中。
 - allsets 歸檔組可用於定義套用於所有歸檔組的參數。
- 為每個 Sun StorEdge SAM-FS 檔案系統所命名的歸檔組會預留用於控制結構資訊。Sun StorEdge SAM-FS 檔案系統會為每個檔案系統提供預設歸檔組。對於每個檔案系統，中介資料與資料檔將歸檔。檔案系統歸檔組將包含目錄與連結資訊及其他歸檔組中不包含的任何檔案。系統給予預設歸檔組的名稱就是與其關聯的檔案系統的名稱，且無法變更。例如，對於配置與命名為 samfs1 的檔案系統，samfs1 即為歸檔組名稱。
- 歸檔組名稱限制為 29 個字元。字元限制為 26 個大寫與小寫字母、數字 0 到 9 及底線字元 (_)

歸檔操作

根據預設值，歸檔器僅為每個歸檔組製作一個備份，但您可請求為每個歸檔組最多製作四個歸檔備份。歸檔組與備份數目將成為磁碟區集合的同義字。歸檔備份提供對獨立磁碟區上的檔案進行複製。

要確定在歸檔之前已完成檔案，歸檔器將在檔案歸檔前修改之後等待指定的時間。正如先前提到的，此段時間稱為**歸檔時間**。

在將檔案視為歸檔或重新歸檔的候選者之前，檔案中的資料必須修改。僅存取檔案不會將其歸檔。例如，在檔案上執行 `touch(1)` 或 `mv(1)` 指令不會導致檔案歸檔或重新歸檔。執行 `mv(1)` 指令將變更檔案名稱，但不會變更檔案資料；同時如果從 `tar(1)` 檔案中回復，這將具有災難復原狀況的分支。若需更多有關災難復原的資訊，請參閱「Sun StorEdge SAM-FS 疑難排解指南」。

選擇歸檔的檔案以歸檔時間為基礎。歸檔時間可為每個歸檔備份進行定義。

使用者可使用 `touch(1)` 指令將其檔案的預設時間參照變更為過去或將來的值。但是，這樣會導致非預期的歸檔結果。為避免此類問題，歸檔器將調整參照，以便它們總是處於以下範圍內：

建立時間 < 時間參照 < 此刻時間

下列章節說明歸檔器從初始的檔案掃描到檔案備份處理所執行的步驟。

步驟 1：辨識要歸檔的檔案

每個掛載的檔案系統都有單獨的 `sam-arfind` 程序。`sam-arfind` 程序會監視每個檔案系統，以決定需要歸檔的檔案。每當檔案受到變更且此變更會影響其歸檔狀態時，檔案系統就會通知其 `sam-arfind` 程序。這種變更的範例包括檔案修改、重新歸檔、取消歸檔及重新命名。收到通知時，`sam-arfind` 程序會檢查檔案以決定所需執行的歸檔動作。

`sam-arfind` 程序可使用檔案屬性說明以決定檔案所屬的歸檔組。用於決定檔案的歸檔組之特性包含如下：

- 使用常規表示式的檔案名稱的目錄路徑部分及完整檔案名稱 (選用)
- 檔案所有者的使用者名稱
- 檔案所有者的群組名稱
- 最小檔案大小
- 最大檔案大小

如果已達到或超出一個或多個備份的檔案歸檔時間，`sam-arfind` 會將檔案新增至歸檔組的一或多個歸檔請求。**歸檔請求**是所有屬於相同歸檔組的檔案集合。單獨的歸檔請求是用於重新歸檔的檔案。這樣可允許尚未歸檔和重新歸檔的檔案單獨控制排程。歸檔請求是在以下目錄中的檔案：

```
/var/opt/SUNWsamfs/archiver/file_sys/ArchReq
```

此目錄中的檔案為二進位檔案，而且您可以藉由使用 `showqueue(1M)` 指令來顯示這些檔案。

歸檔請求有時稱為 *ArchReq*。

如果還沒達到一或多個備份的檔案歸檔時間，檔案位於的目錄和達到歸檔時間的時間會新增到掃描清單中。當到達掃描清單的時間，就會對目錄進行掃描。到達其歸檔時間的檔案會新增到歸檔請求中。

如果檔案已離線，`sam-arfind` 程序將為歸檔備份選擇用作來源的磁碟區。如果檔案備份正在重新歸檔，`sam-arfind` 程序將選擇包含正在重新歸檔的歸檔備份之磁碟區。

如果檔案已分段，則僅選擇歸檔那些已變更的區段。區段檔案的索引不包含使用者資料，因此它將被視為檔案系統歸檔組的成員並單獨歸檔。

歸檔優先權可從檔案屬性特性，及從與歸檔組相關的檔案屬性倍增器計算。基本上，計算如下：

歸檔優先權 = (檔案屬性值 * 屬性倍增器) 的總合

大多數檔案屬性值數是 1 或 0，而屬性是 TRUE 或 FALSE。例如，如果製作了歸檔備份 1，則屬性備份 1 的值是 1。因此，備份 2、備份 3 及備份 4 的值都是 0。

其他如歸檔時間與檔案大小，可以是除了 0 或 1 之外的值。

可從歸檔組的 `-priority` 參數決定屬性倍增器值。可從檔案的各方面 (如時間或大小) 給予值，以便您的站點變更歸檔請求的優先權。若需更多有關 `-priority` 參數的資訊，請參閱 `archiver.cmd(4)` 線上手冊。

歸檔屬性與屬性倍增器是浮點數。所有屬性倍增器的預設值是 0.0。歸檔請求在歸檔請求中被設定為最高檔案優先權。

可使用以下兩種方法標示要進行歸檔的檔案：連續歸檔和掃描歸檔。若是連續歸檔，歸檔器會和檔案系統一起運作，決定哪些檔案需要歸檔。若是掃描歸檔，歸檔器會定期詳察檔案系統並選取要歸檔的檔案。以下章節將說明這兩種方法。

連續歸檔

連續歸檔是預設的歸檔方法 (`examine=noscan`)。藉由連續歸檔，您可以使用 `-startage`、`-startcount` 及 `-startsize` 參數，來指定歸檔組排程的開始條件。這些條件可讓您最佳化歸檔適時性與執行的歸檔工作。

- 範例 1。如果建立應歸檔在一起的檔案需要一個小時，您可以將 `-startage` 參數設為 1 小時 (`-startage 1h`)，以確保所有檔案都已在「歸檔請求」排程之前建立。
- 範例 2。您可以將 `-startsize` 指定為 150 GB (`-startsize 150g`)，以引導歸檔器等候到有 150 GB 的資料可以進行歸檔為止。

- 範例 3。如果您知道將會產生 3000 個檔案進行歸檔，則請指定 `-startcount 3000` 以確保讓檔案歸檔在一起。

在達到任何一種排程開始條件時，`sam-arfind` 程序會將每個歸檔請求傳送至歸檔器常駐程式 (`sam-archiverd`) 進行排程，以將檔案複製到歸檔媒體。

掃描歸檔

此為連續歸檔的替代方式，您可以指定 `examine=scan` 以引導 `sam-arfind` 透過掃描檢查要歸檔的檔案。需要歸檔的檔案會置入歸檔請求。`sam-arfind` 程序將定期掃描每個檔案系統，以決定需要歸檔的檔案。`sam-arfind` 首先執行的掃描是目錄掃描。在進行此掃描時，`sam-arfind` 將透過樹狀目錄遞迴遞減。檢查每個檔案，如果檔案不需要歸檔，將設定檔案狀態旗標 `archdone`。在連續掃描時，`.inodes` 檔案將被掃描。僅有那些具有未設定 `archdone` 旗標的 `inodes` 將被檢查。

在完成檔案系統掃描時，`sam-arfind` 程序會將每個歸檔請求傳送至歸檔器常駐程式 (`sam-archiverd`) 進行排程，以將檔案複製到歸檔媒體。接著，在 `interval=time` 指令所指定的持續時間內，`sam-arfind` 程序將會靜止。在間隔結束時，`sam-arfind` 程序將繼續掃描。

步驟 2：組合歸檔請求

在 `sam-archiverd` 常駐程式收到歸檔請求時，這些歸檔請求將被組合。此步驟將說明組合過程。

一個歸檔請求中的所有檔案可能無法一次歸檔。這可能是歸檔媒體容量或歸檔器程序檔中指定的控制所引起的。組合是指從歸檔請求中一次選擇歸檔檔案的過程。在完成歸檔請求的歸檔備份操作時，如果檔案仍然要歸檔，歸檔請求將重新組合。

`sam-archiverd` 常駐程式將根據特定預設與站點特定條件對歸檔請求中的檔案進行排序。預設操作是依照掃描檔案系統時檔案的出現順序，將歸檔請求中的所有檔案歸檔至相同的歸檔磁碟區。站點特定條件允許您控制檔案歸檔的順序及它們如何在磁碟區上分佈。這些條件稱為歸檔組參數，並且它們將按以下順序進行評估：`-reserve`、`-join`、`-sort`、`-rsort` (執行反向排序) 及 `-drives`。要取得更多有關這些參數的資訊，請參閱 `archiver.cmd(4)` 線上手冊。

如果歸檔請求屬於指定 `-reserve` 所有者的歸檔組，`sam-archiverd` 常駐程式將根據檔案目錄路徑、使用者名稱或群組名稱對歸檔請求中的檔案進行排序。此動作由歸檔組的 `-reserve` 參數控制。將選擇屬於第一個所有者的檔案以歸檔。剩餘的檔案以後歸檔。

如果歸檔請求屬於指定 `-join` 方法的歸檔組，`sam-archiverd` 常駐程式將根據指定的 `-join` 方法將檔案組合在一起。如果也指定 `-sort` 或 `-rsort` 方法，`sam-archiverd` 常駐程式則會根據 `-sort` 或 `-rsort` 方法，為每個群組內的檔案進行排序。歸檔請求將被合併與排序。

合併檔案的每個群組將在組合與排定程序的其他部份被視為單一檔案。

如果歸檔請求屬於指定 `-sort` 或 `-rsort` 方法的歸檔組，`sam-archiverd` 常駐程式則會根據 `-sort` 或 `-rsort` 參數上指定的排序方法，來為檔案進行排序。視排序方法而定，`sam-archiverd` 常駐程式將基於排序方法、時間、大小或目錄位置傾向於將檔案保留在一起。根據預設值，歸檔請求沒有排序，因此檔案將按檔案系統掃描時出現的順序歸檔。

`sam-archiverd` 常駐程式決定檔案在線上或離線。如果線上檔案與離線檔案均在歸檔請求中，將首先選擇線上檔案以歸檔。

如果歸檔請求不需要按排序方法合併或排序，離線檔案將按歸檔備份所在的磁碟區進行排序。這樣可確定同一個磁碟區上的所有檔案 (在每個歸檔組中)，同時按它們在媒體上儲存的順序進行分段備份。在製作離線檔案的多個歸檔備份時，離線檔案將直到製作所有需要的備份時才釋放。與第一個檔案處於相同磁碟區的所有待分段備份檔案均將被選定進行歸檔。

請注意，在歸檔離線檔案時，使用 `-join`、`-sort` 或 `-rsort` 參數可能會對效能產生負面影響。這是因為要歸檔的檔案順序與離線檔案所需的磁碟區順序可能不符。建議您僅在製作第一個歸檔備份時使用 `-join`、`-sort` 或 `-rsort` 參數。如果在開始備份時有足夠的歸檔媒體可用，其他備份很可能會維持第一個備份的順序。

歸檔請求將進入 `sam-archiverd` 常駐程式的排定佇列中。

步驟 3：排定歸檔請求

當存在以下其中一種情況時，`sam-archiverd` 常駐程式中的排程器將根據需要執行：

- 歸檔請求將輸入排定佇列中。
- 已完成對歸檔請求的歸檔。
- 將從目錄伺服器接收媒體狀態的變更。
- 將接收到歸檔器狀態變更的訊息。

排定佇列中的歸檔請求將按優先權進行排序。排程器每次執行時，將檢查所有的歸檔請求，以決定它們是否可指定給 `sam-arcopy` 程序以將檔案複製到歸檔媒體。

製作檔案備份必須有可使用的磁碟機。您必須有歸檔組可用的磁碟區，並有足夠的空間以儲存歸檔請求中的檔案。

磁碟機

如果歸檔組已指定 `-drives` 參數，`sam-archiverd` 常駐程式將切分多個磁碟機中的歸檔請求之選定檔案。如果此時可用的磁碟機數目少於 `-drives` 參數指定的數目，將使用較小的數目。

如果歸檔請求的檔案總量小於 `-drivemin` 值，將僅使用一個磁碟機。`-drivemin` 值是 `-drivemin` 參數指定的值或者是 `archmax` 值。

archmax 值由 -archmax 參數指定或者是為媒體定義的值。若需更多有關 -archmax 參數與 archmax= 指令的資訊，請參閱 archiver.cmd(4) 線上手冊。

如果歸檔請求的檔案總量大於 -drivemin 值，則會運算出以下的值： $\text{磁碟機計數} = \text{總大小} / \text{磁碟機最小值}$ 。如果磁碟機計數小於 -drives 參數指定的磁碟機數量，磁碟機計數則會變成要使用的磁碟機數量。

磁碟機可用不同的時間量來為檔案進行歸檔。您可以使用 -drivemax 參數取得較佳的磁碟機使用狀況。-drivemax 參數會請求您先指定要寫入到磁碟機的最大位元組，然後再重新排定該磁碟機以取得更多資料。

磁碟區

您必須有一個或多個具有足夠空間的磁碟區，至少可以儲存歸檔請求中的一些檔案。如果有足夠的空間，則歸檔組將使用最近用過的磁碟區。而且，歸檔器必須未使用該磁碟區。

如果歸檔組可使用的磁碟區目前正在使用，則選擇其他磁碟區。除非指定 -fillvsns 參數。在此情況下，歸檔請求不可排定。

如果歸檔請求太大而無法容於一個磁碟區，則將選擇符合此磁碟區的檔案以將其歸檔至磁碟區。如果歸檔請求包含大於一個磁碟區的檔案，且沒有選擇歸檔請求的磁碟區溢位功能，則檔案將無法歸檔。適合此情況的訊息將傳送至記錄檔。

您可指定歸檔組 (使用 -ovflmin 參數) 或媒體 (使用 ovflmin= 指令) 的磁碟區溢位功能。若需更多有關 -ovflmin 參數與 ovflmin= 指令的資訊，請參閱 archiver.cmd(4) 線上手冊。指定 ovflmin 將決定溢位媒體的最小檔案大小。為歸檔組指定的 ovflmin 優先於媒體定義的 ovflmin。如果檔案大小小於 ovflmin，將無法歸檔檔案。適合此情況的訊息將傳送至記錄檔。

如果檔案大小大於 ovflmin，則將根據需要指定額外的磁碟區。按大小遞減的順序選擇額外的磁碟區，以減少檔案需要的磁碟區數目。

如果無法找到歸檔請求可使用的磁碟區，歸檔請求將等待。

在決定特定歸檔請求的排定優先權時，某些屬性 (如檔案在線上或離線) 將與歸檔優先權 (在步驟 1 中計算的) 一起使用。若需更多有關自訂優先權倍增器的資訊，請參閱 archiver.cmd(4) 線上手冊中的 -priority 參數。

對於每個歸檔請求，sam-archiverd 常駐程式會將歸檔優先權新增至與各種系統資源屬性相關的倍增器以計算排定優先權。這些屬性與佇列歸檔請求的秒數、要在歸檔程序中使用的第一個磁碟區是否載入磁碟機等有關。

使用調整過的優先權，sam-archiverd 常駐程式可指定每個要複製的就緒歸檔請求。

步驟 4：為歸檔請求中的檔案進行歸檔

在歸檔請求準備歸檔時，`sam-archiverd` 常駐程式將逐一查看每個歸檔請求以標記歸檔檔案 (tarball) 範圍，以便每個歸檔檔案的大小小於 `-archmax` 目標大小 指定。如果單一檔案超過目標大小，它將成為歸檔檔案中的唯一檔案。

對於每個歸檔請求與每個要使用的磁碟機，`sam-archiverd` 常駐程式會將歸檔請求指定給 `sam-arcopy` 程序，以便將檔案複製到歸檔媒體。如果單一檔案超過目標大小，它將成為歸檔檔案中的唯一檔案。歸檔資訊將輸入 `inode`。

如果已啟用歸檔記錄，將建立歸檔記錄。

如果檔案已分段備份，磁碟空間將被釋放。此程序將繼續，直到清單中的所有檔案已歸檔。

各種錯誤與檔案狀態變更會阻止檔案成功複製，這些錯誤可能包括從快取磁碟讀取時發生的錯誤及寫入磁碟區時發生的錯誤。狀態變更包括選擇後的修改、為寫入而開啓的檔案及移除的檔案。

在 `sam-arcopy` 程序結束後，`sam-archiverd` 常駐程式將檢查歸檔請求。如果有任何檔案尚未歸檔，歸檔請求將重新組合。

範例預設執行結果

程式碼範例 3-1 顯示一個執行 `archiver(1M) -l` 指令的輸出範例。

程式碼範例 3-1 `archiver(1M) -l` 指令的輸出

```
# archiver

Archive media:
default:mo
media:mo archmax:5000000
media:lt archmax:50000000
Archive devices:
device:mo20 drives_available:1 archive_drives:1
device:lt30 drives_available:1 archive_drives:1
Archive file selections:
Filesystem samfs1:
samfs1 Metadata
    copy:1 arch_age:240
big path:. minsize:512000
    copy:1 arch_age:240
all path:
    copy:1 arch_age:30
Archive sets:
```

程式碼範例 3-1 archiver(1M) -l 指令的輸出 (續)

```
all
    copy:1 media:mo
big
    copy:1 media:lt
samfs1
    copy:1 media:mo
```

歸檔器常駐程式

sam-archiverd 常駐程式會排定歸檔作業。sam-arfind 程序會將要歸檔的檔案指定給歸檔組。sam-arcopy 程序會將要歸檔的檔案複製到選定磁碟區。

sam-archiverd 常駐程式是在 Sun StorEdge SAM-FS 作業開始時，由 sam-fsd 啟動的。sam-archiver 常駐程式將執行 archiver(1M) 指令以讀取 archiver.cmd 檔案，並建立控制歸檔所需的表格。它將為每個掛載的檔案系統啟動 sam-arfind 程序；同樣地，如果檔案系統未掛載，相關的 sam-arfind 程序將停止。sam-archiverd 程序將監控 sam-arfind 與來自操作者或其他程序的程序訊號。

歸檔記錄檔案與事件記錄

sam-arfind 與 sam-arcopy 程序可產生包含有關每個已歸檔或自動取消歸檔的檔案之資訊的記錄檔。記錄檔是歸檔動作的連續記錄。您可使用記錄檔以找出作為傳統備份用途的較早檔案備份。

這個檔案不是依預設產生。您可使用 archiver.cmd 檔案中的 logfile= 指令，以指定要建立的記錄檔，並指定記錄檔名稱。要取得更多有關記錄檔的資訊，請參閱本章中第 43 頁「使用歸檔器指令」，並參閱 archiver.cmd(4) 線上手冊。

歸檔器可使用 syslog 功能與 archiver.sh 來記錄記錄檔中的警告與通知訊息。

程式碼範例 3-2 顯示定義每個欄位的歸檔器記錄檔之文字行範例。

程式碼範例 3-2 歸檔器記錄檔文字行

```
A 2001/03/23 18:42:06 mo 0004A arset0.1 9a089.1329 samfs1
118.51162514 t0/fdn f 0 56
A 2001/03/23 18:42:10 mo 0004A arset0.1 9aac2.1 samfs1 189.53
1515016 t0/fae f 0 56
A 2001/03/23 18:42:10 mo 0004A arset0.1 9aac2.b92 samfs1 125.53
867101 t0/fai f 0 56
A 2001/03/23 19:13:09 lt SLOT22 arset0.2 798.1 samfs1 71531.14
1841087 t0/fhh f 0 51
```

程式碼範例 3-2 歸檔器記錄檔文字行 (續)

```
A 2001/03/23 19:13:10 lt SLOT22 arset0.2 798.e0e samfs1 71532.12
543390 t0/fhg f 0 51
A 2003/10/23 13:30:24 dk DISK01/d8/d16/f216 arset4.1 810d8.1 qfs2
119571.301 1136048 t1/fileem f 0 0
A 2003/10/23 13:30:25 dk DISK01/d8/d16/f216 arset4.1 810d8.8ad
qfs2 119573.295 1849474 t1/fileud f 0 0
A 2003/10/23 13:30:25 dk DISK01/d8/d16/f216 arset4.1 810d8.16cb
qfs2 119576.301 644930 t1/fileen f 0 0
A 2003/10/23 13:30:25 dk DISK01/d8/d16/f216 arset4.1 810d8.1bb8
qfs2 119577.301 1322899 t1/fileeo f 0 0
```

從左向右讀，以上清單中的欄位內容將顯示在表 3-1 中。

表 3-1 歸檔器記錄檔欄位

欄位	內容
1	歸檔活動如下所示： <ul style="list-style-type: none"> ● A 表示已歸檔。 ● R 表示已重新歸檔。 ● U 表示取消歸檔。
2	以 <i>yyyy/mm/dd</i> 格式表示的歸檔動作日期。
3	以 <i>hh:mm:ss</i> 格式表示的歸檔作業時間。
4	歸檔媒體類型。要取得有關媒體類型的資訊，請參閱 <i>mcf(4)</i> 線上手冊。
5	VSN。對可移除的媒體卡匣而言，此為磁碟區序列名稱。對磁碟歸檔而言，此為磁碟區名稱和歸檔 <i>tar(1)</i> 檔案路徑。
6	歸檔組與備份數目。
7	歸檔檔案在媒體 (<i>tar(1)</i> 檔案) 中的實體開始位置，以及檔案偏移在十六進位歸檔檔案中的實體開始位置。
8	檔案系統名稱。
9	Inode 號碼與產生號碼。產生碼號是為確保唯一性，在 inode 號碼以外所使用的其他號碼，因為 inode 號碼會重新使用。
10	檔案長度 (如果檔案僅寫在 1 個磁碟區)。區段長度 (如果檔案寫在多個磁碟區)。
11	相對於檔案系統的掛載點之檔案路徑與名稱。

表 3-1 歸檔器記錄檔欄位 (續)

欄位	內容
12	檔案類型如下所示： <ul style="list-style-type: none">• d 代表目錄。• f 代表一般檔案。• l 代表符號連結。• R 代表可移除的媒體檔案。• I 代表區段索引。• S 代表資料區段。
13	溢位檔案或區段的部份。若此檔案為溢位檔案，此值則為非零。若是所有其他檔案類型，此值則為零。
14	檔案歸檔的磁帶機之設備序號。

關於 archiver.cmd 檔案

archiver.cmd 檔案可控制歸檔器的運作方式。依據預設，每當啟動 sam-fsd 且掛載 Sun StorEdge SAM-FS 檔案系統時，歸檔器就會執行。歸檔器採取的預設動作如下所示：

- 將所有檔案歸檔至所有可用的磁碟區。
- 所有檔案的歸檔時間是四分鐘。
- 歸檔間隔是 10 分鐘。

您將有可能自訂歸檔器動作，以符合您站點的歸檔請求。這些動作由位於歸檔器程序檔 (archiver.cmd) 中的指令控制。

備註 – 做為下列程序的替代方式，您可以使用 File System Manager 軟體建立或修改 archiver.cmd 檔案。任何您在檔案系統管理中對歸檔配置所做的變更，都會自動對 archiver.cmd 檔案做適當的變更。若需更多資訊，請參閱 File System Manager 線上手冊。

▼ 建立或修改 archiver.cmd 檔案及傳遞您的變更

1. 決定是否要編輯 archiver.cmd 檔案，或者是否要編輯暫存 archiver.cmd 檔案。(選用)。

如果您有 /etc/opt/SUNWsamfs/archiver.cmd 檔案，而且您的系統已經在進行檔案的歸檔，則請執行此步驟。考慮將您的 archiver.cmd 檔案複製到一個可以在開始製作之前，進行編輯和測試的暫存位置。

2. 使用 vi(1) 或其他編輯器來編輯您的 archiver.cmd 檔案或暫存檔案。

新增需要用來在您的站點控制歸檔的指令。若需有關您可以在此檔案中包括的指令之資訊，請參閱第 43 頁「使用歸檔器指令」和第 78 頁「關於磁碟歸檔」。

3. 儲存並關閉 archiver.cmd 檔案或暫存檔案。

4. 使用 archiver(1M) -lv 指令驗證檔案的正確性。

每當變更了 archiver.cmd 檔案，您應該使用 archiver(1M) 指令檢查語法錯誤。如同以下指定 archiver(1M) 指令可針對目前的 Sun StorEdge SAM-FS 系統，對 archiver.cmd 檔案進行評估：

```
# archiver -lv
```

上述指令將列出所有選項，並將 archiver.cmd 檔案、磁碟區、檔案系統內容及錯誤寫入標準的輸出檔案 (stdout)。若有錯誤，則歸檔器無法執行。

依據預設，archiver(1M) 指令會評估 /etc/opt/SUNWsamfs/archiver.cmd 檔案，檢查是否有錯誤。如果您目前是在開始製作前先處理 archiver.cmd 暫存檔案，您可以使用 archiver(1M) 指令的 -c 選項，並提供此暫存檔案的名稱。

5. 重複步驟 2、步驟 3 及步驟 4，直到檔案無誤為止。

您必須先更正所有錯誤，再繼續下一個步驟。如果歸檔器在 archiver.cmd 檔案中發現錯誤，則不會歸檔任何檔案。

6. 儲存並關閉 archiver.cmd 檔案。

7. 將暫存檔案移到 /etc/opt/SUNWsamfs/archiver.cmd (選用)。

只有在處理暫存檔案時才執行此步驟。

8. 使用 samd(1M) config 指令傳遞檔案變更，然後重新啟動系統。

```
# samd config
```

archiver.cmd 檔案

archiver.cmd 檔案由以下類型的指令組成：

- 一般指令
- 歸檔組指派指令
- 歸檔組指令
- VSN pool 指令
- VSN 關聯指令

這些指令由從 archiver.cmd 檔案中讀取的文字行組成。每個指令行包含一個或多個由空格或 Tab 分隔的欄位。任何出現在井號字元 (#) 後面的文字將被視為註解，並不作檢查。文字行以反斜線 (\) 結尾可延續至下一行。

archiver.cmd 檔案中的某些指令需要您指定時間單位或位元組單位。要指定這些單位，請使用第 41 頁的表 3-2 「archiver.cmd 檔案指令單位」中的其中一個字母做為代表單位的數字後綴字元。

表 3-2 archiver.cmd 檔案指令單位

單位後綴	意義
時間後綴：	
s	秒。
m	分。60 秒。
h	小時。3,600 秒。
d	天。86,400 秒。
w	週。604,800 秒。
Y	年。31,536,000 秒。
大小後綴：	
b	位元組。
k	KB。2**10 或 1,024 位元組。
M	MB。2**20 或 1,048,576 位元組。
G	GB。2**30 或 1,073,741,824 位元組。
T	TB。2**40 或 1,099,511,627,776 位元組。
P	PB。2**50 或 1,125,899,906,842,624 位元組。
E	EB。2**60 或 1,152,921,504,606,846,976 位元組。

archiver.cmd 檔案範例

程式碼範例 3-3 顯示 archiver.cmd 檔案範例。右側的註解表示各種指令類型。

程式碼範例 3-3 archiver.cmd 檔案範例

```
interval = 30m                # General directives
logfile = /var/opt/SUNWsamfs/archiver/archiver.log

fs = samfs1                    # Archive Set Assignments
no_archive tmp
work work
    1 1h
    2 3h
images images -minsize 100m
    1 1d
    2 1w
samfs1_all .
    1 1h
    2 1h

fs = samfs2                    # Archive Set Assignments
no_archive tmp
system . -group sysadmin
    1 30m
    2 1h
samfs2_all .
    1 10m
    2 2h

params                          # Archive Set Directives
allsets -drives 2
images.1 -join path -sort size
endparams

vsns                            # VSN Associations
samfs1.1 mo          optic-2A
samfs1.2 lt          TAPE01
work.1 mo            optic-[3-9] [A-Z]
work.2 lt            .*
images.1 lt          TAPE2 [0-9]
images.2 lt          TAPE3 [0-9]
samfs1_all.1 mo.*
samfs1_all.2 lt.*
samfs2.1 mo          optic-2A
samfs2.2 lt          TAPE01
system.1 mo          optic08a optic08b
system.2 lt          ^TAPE4 [0-1]
samfs2_all.1 mo.*
samfs2_all.2 lt.*
endvsns
```

使用歸檔器指令

以下章節將解釋 `archiver.cmd` 指令。這些指令如下：

- 第 43 頁 「全域歸檔指令」
- 第 50 頁 「檔案系統指令」
- 第 51 頁 「歸檔組指定指令」
- 第 58 頁 「歸檔備份指令」
- 第 61 頁 「歸檔組備份參數」
- 第 75 頁 「VSN 關聯指令」
- 第 77 頁 「VSN 池指令」

全域歸檔指令

全域指令控制全部的歸檔器作業，並讓您最佳化站點配置的歸檔器作業。全域指令可直接新增至 `archiver.cmd` 檔案或使用 File System Manager 軟體指定。若需更多關於使用 File System Manager 設定全域指令的資訊，請參閱檔案系統管理線上手冊。

`archiver.cmd` 檔案中的全域指令可透過第二欄位的等號 (=) 或未出現的其他欄位進行識別。

必須先指定全域指令，然後再指定 `archiver.cmd` 檔案中的任何 `fs=` 指令。`fs=` 是與特定檔案系統有關的指令。如果歸檔器在 `fs=` 指令後偵測到全域指令，它將會發出一則訊息。

archivemeta 指令：控制是否歸檔中介資料

`archivemeta` 指令可控制是否要對檔案系統中介資料進行歸檔。如果檔案經常四處移動，而且檔案系統中的目錄結構通常有很多變更時，您應該會想要對中介資料進行歸檔。不過，如果目錄結構十分穩定，您則可停用中介資料歸檔，並減少在卡匣載入到歸檔中介資料或從其中卸載時，可移除的媒體磁碟機所執行的動作。依據預設，會對中介資料進行歸檔。

請用以下格式使用此指令：

```
archivemeta = 狀態
```

關於狀態，您可以指定 `on` 或 `off`。預設值為 `on`。

中介資料歸檔會依您使用的是「版本 1」或「版本 2」超級區塊而有所不同，詳情如下：

- 若是「版本 1」檔案系統，歸檔器會將目錄、可移除的媒體檔案、區段索引 `inode` 及符號連結歸檔為中介資料。
- 若是「版本 2」檔案系統，可移除的媒體檔案和符號連結則會儲存在 `inode`，而不是儲存在資料區塊，而且不會被歸檔。只有目錄和區段索引 `inode` 會被歸檔為中介資料。符號連結會歸檔為資料。

archmax 指令：控制歸檔檔案的大小

`archmax` 指令將指定歸檔檔案大小的最大值。使用者檔案組合在一起即形成歸檔檔案。在達到目標大小時，無法再將更多的使用者檔案新增至歸檔檔案。較大的使用者檔案將被寫入單一的歸檔檔案。

要變更預設值，請使用下列指令：

```
archmax=媒體 目標大小
```

表 3-3 `archmax` 指令的引數

引數	意義
媒體	表示媒體類型。如需有效媒體類型清單，請參閱 <code>mcf(4)</code> 線上手冊。
目標大小	指定歸檔檔案的最大大小。歸檔檔案大小的最大值取決於媒體。根據預設值，寫入光碟的歸檔檔案不大於 5 MB。磁帶的最大預設歸檔檔案大小是 512 MB。

歸檔檔案大小設定為大或小，是各有利弊之處。例如，如果歸檔至磁帶，且將 `archmax` 設定為大，磁帶機將很少停止與啟動。但是，在寫入較大的歸檔檔案時，有可能會提前到達磁帶末端，從而會浪費較多的磁帶。一般而言，不應將 `archmax` 設定為超過媒體容量的 5%。例如，您可以對 20 GB 的磁帶使用以下 `archmax` 指令：

```
archmax=sg 1G
```

`archmax` 指令亦可為個別歸檔組設定。

bufsize 指令：設定歸檔器的緩衝區大小

根據預設值，歸檔的檔案將被複製到使用記憶體緩衝區的歸檔媒體。您可以使用 `bufsize` 指令指定非預設的緩衝區大小 (選用)，也可以鎖定緩衝區。這些動作可以提高效能，並且您可嘗試不同的緩衝區大小值。

請用以下格式使用此指令：

```
bufsize=媒體 緩衝區大小 [ lock ]
```

表 3-4 bufsize 指令的引數

引數	意義
媒體	表示媒體類型。如需有效媒體類型清單，請參閱 <code>mcf(4)</code> 線上手冊。
緩衝區大小	指定 2 至 32 的數字。預設值為 4。此值會與該媒體類型的 <code>dev_blksize</code> 值相乘，而使用產生的緩衝區大小。 <code>dev_blksize</code> 可在 <code>defaults.conf</code> 檔案中指定。若需更多有關此檔案的資訊，請參閱 <code>defaults.conf(4)</code> 線上手冊。
lock	<p><code>lock</code> 引數表示在製作歸檔備份時，歸檔器是否應該使用鎖定的緩衝區。如果指定了 <code>lock</code>，歸檔器在 <code>sam-arcopy(1M)</code> 作業過程中會將檔案鎖定在記憶體的歸檔緩衝區內。這樣即可免除根據每個 I/O 請求來鎖定與解除鎖定緩衝區的操作，從而減少了系統 CPU 時間。</p> <p><code>lock</code> 引數應該僅在具有大容量記憶體的大系統中指定。記憶體不足會導致出現記憶體不夠用的情況。</p> <p>僅在為要歸檔的檔案啟動了直接 I/O 時，<code>lock</code> 引數才會生效。依據預設，<code>lock</code> 參數並未指定，且檔案系統會鎖定所有直接 I/O 緩衝區，包括要歸檔的緩衝區。若需更多有關啟動直接 I/O 的資訊，請參閱 <code>setfa(1)</code> 線上手冊、<code>sam_setfa(3)</code> 程式庫常式線上手冊，或 <code>mount_samfs(1M)</code> 線上手冊上的 <code>-O forcedirectio</code> 選項。</p>

例如，此指令可在 `archiver.cmd` 檔案的文字行中指定，如下所示：

```
bufsize=od 7 lock
```

您可以使用 `-bufsize` 和 `-lock` 歸檔組複製參數，以歸檔組為基礎來指定緩衝區大小和鎖定。如需更多資訊，請參閱第 61 頁「歸檔組備份參數」。

drives 指令：控制用於歸檔的磁碟機數量

依據預設，歸檔器將使用自動程式庫中的所有磁碟機進行歸檔。要限制歸檔器使用的自動程式庫中之磁碟機數目，請使用 `drives` 指令。

請用以下格式使用此指令：

```
drives=自動程式庫 計數
```

表 3-5 drives 指令的引數

引數	意義
自動程式庫	在 mcf 檔案中定義的自動程式庫之「系列組」名稱。
計數	用於歸檔活動的磁碟機數目。

另請參閱第 63 頁「指定用於歸檔請求的磁帶機數量：-drivemax、-drivemin 及 -drives」中所述的 -drivemax、-drivemin 及 -drives 歸檔組複製參數。

examine 指令：控制歸檔掃描

新檔案和已變更的檔案為歸檔的候選者。歸檔器會實行以下其中一種方法來尋找這類檔案：

- 連續歸檔。當實行連續歸檔時，歸檔器會與檔案系統一起運作，以在檔案發生變更後立即偵測。
- 掃描型歸檔。若是掃描型歸檔，歸檔器則會定期掃描檔案系統，尋找需要歸檔的檔案。

examine 指令可控制要讓歸檔器執行連續或掃描型歸檔，詳情如下：

examine=方法

關於方法，請指定表 3-6 顯示的其中一個關鍵字。

表 3-6 examine 指令的方法引數的值

方法值	意義
noscan	指定連續歸檔。在初始掃描後，只會在內容變更和歸檔需要時才掃描目錄。不會掃描目錄和 inode 資訊。這種歸檔方法提供比掃描型歸檔更好的效能，特別是對有 1,000,000 檔案以上的檔案系統。預設值。
scan	指定掃描型歸檔。初始檔案系統掃描為目錄掃描。後續掃描為 inode 掃描。
scandirs	只在目錄上指定掃描型歸檔。指定時，如果歸檔器發現一個具有 no_archive 屬性組的目錄，則不會掃描該目錄。不會變更的檔案可置於此目錄，這種方式可以動態減低耗費於歸檔掃描的時間量。
scaninodes	只在 inode 上指定掃描型歸檔。

interval 指令：指定歸檔間隔

歸檔器會定期執行，檢查所有掛載 Sun StorEdge SAM-FS 檔案系統的狀態。計時由歸檔間隔控制。**歸檔間隔**是介於對每個檔案系統的掃描作業之間的時間。要變更時間，請使用 `interval` 指令。

備註 – `interval` 指令僅在連續歸檔未設定時才會初始完整掃描。若未設定連續歸檔且沒有指定 `startage`、`startsize` 或 `startcount` 參數，歸檔器會使用 `interval` 指令排程掃描。若已設定連續歸檔 (`examine=noscan`)，間隔指令會以預設的 `startage` 值動作。

請用以下格式使用此指令：

```
interval=時間
```

關於**時間**，請指定檔案系統上的掃描作業之間的時間 (以秒為單位)。依據預設，**時間**是以秒為解讀的單位。依據預設，`interval=600`，其為 10 分。您可以指定時間單位，例如：分、小時等等。要取得有關指定時間單位的資訊，請參閱第 41 頁的表 3-2「`archiver.cmd` 檔案指令單位」。

如果歸檔器接收到 `samu(1M)` 公用程式的 `:arrun` 指令，它將立即開始掃描所有檔案系統。如果也在 `archiver.cmd` 檔案中指定 `examine=scan` 指令，掃描則會在發出 `:arrun` 或 `:arscan` 指令之後執行。

如果設定了檔案系統的 `hwm_archive` 掛載選項，歸檔間隔可自動減短。此掛載選項將指定，在檔案系統裝滿且超過高值參數時，歸檔器將開始掃描。`high=百分比掛載選項`會設定檔案系統的高值參數。

若需更多有關指定歸檔間隔的資訊，請參閱 `archiver.cmd(4)` 線上手冊。若需更多有關設定掛載選項的資訊，請參閱 `mount_samfs(1M)` 線上手冊。

logfile 指令：指定歸檔器記錄檔

歸檔器可產生包含有關每個已歸檔、重新歸檔或自動取消歸檔的資訊之記錄檔。記錄檔是歸檔動作的連續記錄。要指定記錄檔，請使用 `logfile` 指令。請用以下格式使用此指令：

```
logfile=路徑名稱
```

關於**路徑名稱**，請指定絕對路徑和記錄檔名稱。根據預設值，將不產生此檔案。

`logfile` 指令亦可為個別檔案系統設定。

▼ 備份歸檔器記錄檔

假設您要將前一天的記錄檔複製到替代位置，以便每天備份歸檔器記錄檔。如果您確定在關閉歸檔器記錄檔時執行備份，才可完成此操作。也就是說，您無需在歸檔器開啓以進行寫入操作時執行備份操作。您需執行的步驟如下：

1. 使用 `mv(1)` 指令可在 **UFS** 中移動歸檔器記錄檔。

這將會給予任何 `sam-arfind(1M)` 或 `sam-arcopy(1M)` 完成寫入歸檔器記錄檔的作業時間。

2. 使用 `mv(1)` 指令將前一天的歸檔器記錄檔移動到 **Sun StorEdge SAM-FS** 檔案系統。

notify 指令：重新命名事件通知程序檔

`notify` 指令會將歸檔器的事件通知程序檔檔案的名稱設定為檔案名稱。請用以下格式使用此指令：

```
notify=檔案名稱
```

關於**檔案名稱**，請指定含有歸檔器事件通知程序檔或此檔案的完整路徑的檔案之名稱。

以下為預設的檔案名稱：

```
/etc/opt/SUNWsamfs/scripts/archiver.sh
```

歸檔器會以特定站點的方式執行此程序檔來處理各種事件。可使用關鍵字為第一個引數呼叫指令碼。關鍵字如下所示：`emerg`、`alert`、`crit`、`err`、`warning`、`notice`、`info` 及 `debug`。

額外的引數在預設的指令碼中有說明。要取得更多相關資訊，請參閱 `archiver.sh(1M)` 線上手冊。

ovflmin 指令：控制磁碟區溢位

磁碟區溢位是允許歸檔檔案跨越多個磁碟區的程序。在 `archiver.cmd` 檔案中使用 `ovflmin` 指令時，磁碟區溢位將會啓動。在檔案大小超過 `ovflmin` 指令的**最小檔案大小**引數時，歸檔器會將此檔案的其他部分寫入相同類型的其他可用磁碟區 (如有需要)。寫入每個磁碟區的部分檔案稱為一個**區段**。

備註 – 在使用磁碟區溢位之前，確定瞭解此概念。請僅在完全評估對站點的影響之後再謹慎使用磁碟區溢位。跨越磁碟區的檔案要進行災難復原與回收較為困難。

歸檔器使用 `ovflmin` 指令控制磁碟區溢位。`ovflmin` 指令可指定允許磁碟區溢位的最小檔案大小。磁碟區溢位依預設為停用。

請用以下格式使用此指令：

```
ovflmin = 媒體 最小檔案大小
```

表 3-7 ovflmin 指令的引數

引數	意義
媒體	表示媒體類型。如需有效媒體類型清單，請參閱 mcf(4) 線上手冊。
最小檔案大小	指定要溢位的最小檔案大小。

範例 1。 假設許多檔案存在一個佔 mo 媒體重要部分 (如 25%) 的長度。這些檔案會部份填充磁碟區，並在每個磁碟區上保留未使用的空間。要獲得較好的磁碟區組裝，請將 mo 媒體的 ovflmin 設定為比最小檔案稍小的大小。下列指令將其設定為 150 MB：

```
ovflmin=mo 150m
```

請注意，在此範例中啟用溢位磁碟亦會導致載入兩個磁碟區以歸檔與分段備份檔案。

ovflmin 指令亦可為個別歸檔組設定。

範例 2。 sls(1) 指令列出顯示每個 VSN 上每個檔案的區段之歸檔備份。程式碼範例 3-4 顯示歸檔器記錄檔，以及一個橫跨多個磁碟區的大型檔案 file50 之 sls -D 指令輸出。

程式碼範例 3-4 歸檔器記錄檔範例

```
A 97/01/13 16:03:29 lt DLT000 big.1 7eed4.1 samfs1 13.7
477609472 00 big/file50 0 0

A 97/01/13 16:03:29 lt DLT001 big.1 7fb80.0 samfs1 13.7
516407296 01 big/file50 0 1

A 97/01/13 16:03:29 lt DLT005 big.1 7eb05.0 samfs1 13.7
505983404 02 big/file50 0 2
```

程式碼範例 3-4 顯示 file50 以 DLT000、DLT001 及 DLT005 的 VSN，橫跨三個磁碟區。檔案在磁碟區中的位置與每個區段的大小分別在第七個欄位與第十個欄位中顯示，並且也顯示與 sls -D 輸出相符。有關歸檔器記錄項目的完整說明，請參閱 archiver(1M) 線上手冊。

程式碼範例 3-5 顯示 `sls -D` 指令和輸出。

程式碼範例 3-5 `sls(1M) -D` 指令和輸出

```
# sls -D file50
file50:
mode: -rw-rw---- links: 1 owner: gmm group: sam
length: 1500000172 admin id: 7 inode: 1407.5
offline; archdone; stage -n
copy1: ---- Jan 13 15:55 lt
section 0: 477609472 7eed4.1 DLT000
section 1: 516407296 7fb80.0 DLT001
section 2: 505983404 7eb05.0 DLT005
access: Jan 13 17:08 modification: Jan 10 18:03
changed: Jan 10 18:12 attributes: Jan 13 16:34
creation: Jan 10 18:03 residence: Jan 13 17:08
```

磁碟區溢位檔案不會產生總和檢查。要取得更多有關使用總和檢查的資訊，請參閱 `ssum(1)` 線上手冊。

備註 – 當您使用磁碟區溢位功能，請記住萬一發生重大災難時，將會難以擷取磁碟區溢位資料。若需有關如何擷取這種檔案的資訊，請參閱「Sun StorEdge SAM-FS 疑難排解指南」中的範例。更多相關資訊，請參閱 `request(1)` 線上手冊。

wait 指令：延遲歸檔器啟動

`wait` 指令會導致歸檔器等待來自 `samu(1M)` 或 File System Manager 的啟動訊號。在接收到此訊號時，將開始一般的歸檔器操作。歸檔器依預設由 `sam-fsd(1M)` 啟動後便開始歸檔。要延遲歸檔，請使用 `wait` 指令。請用以下格式使用此指令：

```
wait
```

`wait` 指令亦可為個別檔案系統設定。

檔案系統指令

在 `archiver.cmd` 檔案中，您可以在一般指令之後使用 `fs=` 指令，以包括特定檔案系統特定的指令。在遇到 `fs=` 指令之後，歸檔器會假設所有後續指令指定的動作都是僅限於對個別的檔案系統執行。

您可以藉由編輯 `archiver.cmd` 檔案指定 `fs=` 指令 (如下節所述)，或者您可以使用 File System Manager 軟體指定。請參閱 File System Manager 線上手冊以取得更多資訊。

fs 指令：指定檔案系統

根據預設值，歸檔控制可套用於所有檔案系統。但是，您可對個別檔案系統進行某些限制。要指定個別檔案系統，請使用 `fs` 指令。請用以下格式使用此指令：

```
fs=fs名稱
```

關於 `fs` 名稱，請指定在 `mc.f` 檔案中定義的檔案系統名稱。

在這些指令後出現的一般指令與歸檔組關聯指令將僅套用於指定的檔案系統，直到出現其他的 `fs=` 指令。例如，您可使用此指令以便為每個檔案系統指定不同的記錄檔。

其他檔案系統指令

幾個指令可被同時指定為所有檔案系統的全域指令，與特定於一個檔案系統的指令。無論在何處指定，它們的作用都是一樣的。這些指令如下所示：

- `interval` 指令。若需更多有關此指令的資訊，請參閱第 47 頁「`interval` 指令：指定歸檔間隔」。
- `logfile` 指令。要取得更多有關此指令的資訊，請參閱第 47 頁「`logfile` 指令：指定歸檔器記錄檔」。
- `wait` 指令。若需更多有關此指令的資訊，請參閱第 50 頁「`wait` 指令：延遲歸檔器啟動」。

歸檔組指定指令

依據預設，檔案會歸檔為針對該檔案系統所命名的歸檔組之一部分。但是，您可指定歸檔組以包括共用類似特性的檔案。如果檔案不符合其中一個指定的歸檔組，它將作為為檔案系統命名的預設歸檔組之部份進行歸檔。

您可以直接編輯 `archiver.cmd` 檔案建立歸檔組 (如下節所述)，或者您可以使用 File System Manager 軟體建立歸檔組。在 File System Manager 中，歸檔規則會定義歸檔組。若需更多資訊，請參閱 File System Manager 線上手冊。

指定歸檔組

歸檔組成員指令會將具有類似特性的檔案指派給歸檔組。這些指令的語法將在 `find(1)` 指令後被模仿。每個歸檔組指派指令具有下列格式：

```
歸檔組名稱 路徑 [搜尋條件 1 搜尋條件 2 ... ] [檔案屬性]
```

表 3-8 歸檔組指定指令的引數

引數	意義
歸檔組名稱	歸檔組的站點定義名稱。必須是歸檔組指派指令中的第一個欄位。歸檔組名稱通常表示屬於歸檔組的檔案特性。歸檔組名稱限制為字母、數字及底線字元 (<code>_</code>)。不允許使用其他特殊字元或空格。歸檔組名稱的第一個字元必須是字母。 為防止歸檔各種檔案，請將 <code>no_archive</code> 指定為歸檔組名稱。
路徑	相對於檔案系統的掛載點之路徑這可允許讓歸檔組成員指令套用到多個 Sun StorEdge SAM-FS 檔案系統。如果路徑要包括檔案系統中的所有檔案，可在路徑欄位中使用小數點 (<code>.</code>)。不允許在路徑中使用正斜線 (<code>/</code>)。路徑指定目錄中的檔案及其子目錄，將被視為包括在此歸檔組中。
搜尋條件 1 搜尋條件 2	可指定零、一個或多個搜尋條件引數。根據檔案大小、檔案所有權及其他因素，可指定搜尋條件以限制歸檔組。要取得有關可用搜尋條件引數的資訊，請參閱以下章節。
檔案屬性	可指定零、一個或多個檔案屬性。 <code>sam-arfind</code> 程序在歸檔過程中掃描檔案系統時，系統將為檔案設定這些檔案屬性。

範例 1。程式碼範例 3-6 顯示常見的歸檔組成員指令。

程式碼範例 3-6 歸檔組成員指令

```
hmk_files    net/home/hmk    -user hmk
datafiles    xray_group/data -size 1M
system      .
```

範例 2。您可以藉由將檔案包括於一個名為 `no_archive` 的歸檔組中，來抑制歸檔器。程式碼範例 3-7 顯示阻止進行 `tmp` 目錄 (任何層級) 中的檔案歸檔之文字行，而且無論 `tmp` 目錄位於該檔案系統的哪個目錄中。

程式碼範例 3-7 阻止歸檔的歸檔指令

```
fs = samfs1
no_archive tmp
no_archive . -name */tmp/
```

以下章節將說明可指定的搜尋條件。

檔案大小搜尋條件：-access 和 -nftv

您可以使用 `-access` 時間特性來指定一個檔案用來決定歸檔組成員的時間。當您使用搜尋條件時，存取時間早於時間的檔案會重新歸檔到其他媒體。關於時間，請指定一個整數，後面接著表 3-9 顯示的其中一個後綴。

表 3-9 -access 時間後綴

字母	意義
s	秒
m	分
h	小時
d	天
w	週
ly	年

例如，您可以使用此指令來指定長期未存取的檔案，將其重新歸檔到較廉價的媒體。

決定時間時，會驗證檔案的存取和修改時間以確認這些時間是大於或等於檔案建立時間，且小於或等於檢查檔案的時間。這是要提供適當的歸檔與取消歸檔。但是，對於已遷移到目錄的檔案，此驗證無法以預想的運作方式產生。`-nftv` (沒有檔案時間驗證) 參數可用於這些情形，以防止檔案存取與修改時間的驗證。

檔案大小搜尋條件：-minsize 和 -maxsize

透過 `-minsize size` 和 `-maxsize size` 大小特性，檔案的大小可以用來決定歸檔組成員。關於大小，請指定一個整數，後面接著表 3-10 顯示的其中一個字母。

表 3-10 -minsize 和 -maxsize 大小後綴

字母	意義
b	位元組
k	KB
M	MB
G	GB
T	TB
P	PB
E	EB

範例。此範例中的文字行可指定，所有至少 500 KB、小於 100 MB 的檔案屬於歸檔組 `big_files`。大於 100 MB 的檔案屬於歸檔組 `huge_files`。程式碼範例 3-8 顯示相關文字行。

程式碼範例 3-8 使用 `-minsize` 和 `-maxsize` 指令範例

```
big_files . -minsize 500k -maxsize 100M
huge_files . -minsize 100M
```

所有者和群組搜尋條件：`-user` 和 `-group`

所有權和群組的合併使用可決定歸檔組成員，亦即透過 `-user` 名稱和 `-group` 名稱特性。程式碼範例 3-9 顯示其指令範例。

程式碼範例 3-9 使用 `-user` 和 `-group` 指令的範例

```
adm_set . -user sysadmin
mktng_set . -group marketing
```

屬於使用者 `sysadmin` 的所有檔案都屬於歸檔組 `adm_set`，並且所有具有 `marketing` 群組名稱的檔案都在歸檔組 `mktng_set` 中。

使用模式比對的檔案名稱搜尋條件：`-name regex`

要包括在歸檔組中的檔案名稱，可透過使用一般表示式進行指定。做為搜尋條件的 `-name regex` 指定可指定，任何符合常規表示式 `regex` 的完整路徑都是歸檔組成員。

`regex` 引數遵循 `regexp(5)` 線上手冊中列出的規則。請注意，一般表示式與 UNIX 萬用字元遵循不同的規則。

在內部，選定目錄下的所有檔案 (與相對於檔案系統的掛載點之指定路徑一起) 將為模式比對列出與傳遞。這將允許您在 `-name regex` 欄位中建立模式，以符合檔案名稱與路徑名稱。

範例

1. 以下指令會將歸檔組 `images` 中的檔案限制為以 `.gif` 結尾的那些檔案：

```
images . -name \.gif$
```

2. 以下指令將選擇以字元 GEO 開頭的檔案。

```
satellite . -name /GEO
```

3. 您可使用具有 no_archive 歸檔組的常規表示式。以下指定可防止歸檔任何以 .o 結尾的檔案：

```
no_archive . -name \.o$
```

4. 假設您的 archiver.cmd 檔案含有程式碼範例 3-10 中所示的文字行。

程式碼範例 3-10 常規表示式檔案

```
# File selections.
fs = samfs1
    1 ls
    2 ls
no_archive share/marketing -name fred\.
```

使用此 archiver.cmd 檔案，歸檔器將不歸檔使用者目錄或子目錄中的 fred.*。檔案的歸檔情況如下所示：

- 程式碼範例 3-11 顯示不歸檔的檔案 (如果您指定程式碼範例 3-10 中所示的指令)。

程式碼範例 3-11 不歸檔的檔案 (假設程式碼範例 3-10 中所示的指令)

```
/sam1/share/marketing/fred.anything
/sam1/share/marketing/first_user/fred.anything
/sam1/share/marketing/first_user/first_user_sub/fred.anything
```

- 程式碼範例 3-12 顯示已歸檔的檔案 (如果您指定程式碼範例 3-10 中所示的指令)。

程式碼範例 3-12 已歸檔的檔案 (假設程式碼範例 3-10 中所示的指令)

```
/sam1/fred.anything
/sam1/share/fred.anything
/sam1/testdir/fred.anything
/sam1/testdir/share/fred.anything
/sam1/testdir/share/marketing/fred.anything
/sam1/testdir/share/marketing/second_user/fred.anything
```

5. 假設您的 archiver.cmd 檔案含有程式碼範例 3-13 中所示的文字行。

程式碼範例 3-13 archiver.cmd 檔案範例

```
# File selections.
fs = samfs1
    1 ls
    2 ls
no_archive share/marketing -name ^share/marketing/[^/]*fred\.
```

程式碼範例 3-13 中的 archiver.cmd 檔案將不歸檔使用者主目錄中的 fred.*，但將歸檔使用者子目錄與目錄 share/marketing 中的 fred.*。在此情況下，使用者主目錄將成爲 first_user。此範例會將從 share/marketing/ 至下一個斜線字元 (/) 的任何目錄視爲使用者主目錄。檔案的歸檔情況如下所示：

- 下列檔案將不歸檔：

```
/sam1/share/marketing/first_user/fred.anything
```

- 程式碼範例 3-14 顯示已歸檔的檔案 (如果您指定程式碼範例 3-13 中所示的指令)。

程式碼範例 3-14 已歸檔的檔案 (假設程式碼範例 3-13 中所示的指令)

```
/sam1/share/fred.anything
/sam1/share/marketing/fred.anything
/sam1/share/marketing/first_user/first_user_sub/fred.anything
/sam1/fred.anything
/sam1/testdir/fred.anything
/sam1/testdir/share/fred.anything
/sam1/testdir/share/marketing/fred.anything
/sam1/testdir/share/marketing/second_user/fred.anything
/sam1/testdir/share/marketing/second_user/sec_user_sub/fred.any
```

釋放和分段備份檔案屬性：-release 和 -stage

您可以使用 -release 和 -stage 選項，個別設定與歸檔組內的檔案相關之釋放和分段備份屬性。這些設定值將覆寫使用者之前可能已設定的分段備份或釋放屬性。

-release 選項具有以下格式：

```
-release 屬性
```

用於 `-release` 指令的屬性與 `release(1)` 指令遵循相同的規則，如表 3-11 所示。

表 3-11 `-release` 選項

屬性	意義
a	釋放完成第一個歸檔備份後的檔案。
d	重設回預設值。
n	永不釋放檔案。
p	部份釋放檔案的磁碟空間。

`-stage` 選項具有以下格式：

```
-stage 屬性
```

用於 `-stage` 指令的屬性與 `stage(1)` 指令遵循相同的規則，如表 3-12 所示。

表 3-12 `-stage` 指令的屬性

屬性	意義
a	結合分段備份歸檔組中的檔案。
d	重設回預設值。
n	永不分段備份歸檔組中的檔案。

下列範例顯示如何可使用檔案名稱指定與檔案屬性，以部份釋放 Macintosh 資源目錄：

```
MACS . -name .*/\.\rscs/ -release p
```

歸檔組成員衝突

有時為包括在歸檔組中的檔案選擇路徑與其他檔案特性，會導致不明確的歸檔組成員。這些情況可按下列方式解決：

1. 選擇首先出現在歸檔組中的成員定義。
2. 在任何全域定義之前，首先選擇檔案系統本機的成員定義。
3. 完全重複前一個定義的成員定義被記錄為錯誤。

作為這些規則的結果，更具限制性的成員定義應該置於程序檔之前。

在控制某特定檔案系統 (使用 `fs=fs名稱` 指令) 的歸檔時，歸檔器會先評估檔案系統的特定指令，再評估全域指令。因此，檔案可被指定給本機歸檔組 (包括 `no_archive` 歸檔組)，而不是被指定給全域歸檔。這在設定全域歸檔組指定，如 `no_archive` 時有其含義。

程式碼範例 3-15 顯示 `archiver.cmd` 檔案。

程式碼範例 3-15 可能具有成員衝突的 `archiver.cmd` 檔案

```
no_archive . -name *.*\.*$
fs = samfs1
    allfiles .
fs = samfs2
    allfiles .
```

從程式碼範例 3-15 的讀取結果看來，管理員似乎不想透過兩個檔案系統歸檔任何 `.o` 檔案。但是，因為在全域歸檔組指定 `no_archive` 之前，本機歸檔組指定 `allfiles` 已被評估，`samfs1` 與 `samfs2` 檔案系統中的 `.o` 檔案將被歸檔。

程式碼範例 3-16 顯示用來確保兩個檔案系統中都沒有歸檔任何 `.o` 檔案的指令。

程式碼範例 3-16 更正的 `archiver.cmd` 檔案

```
fs = samfs1
    no_archive .-name *.*\.*$
    allfiles .
fs = samfs2
    no_archive .-name *.*\.*$
    allfiles .
```

歸檔備份指令

若不指定歸檔備份，歸檔器將為歸檔組中的檔案寫入單一歸檔備份。根據預設值，在檔案的歸檔時間為四分鐘時製作此備份。如果需要多個歸檔備份，必須使用歸檔備份指令指定所有備份，包括第一個備份。

歸檔備份指令是以一個整數的備份數目做為開頭。此數字 (1、2、3 或 4) 是備份數目。數字後面是為此備份指定歸檔特性的一個或多個引數。

歸檔備份指令必須在與它們有關的歸檔組指派指令之後立即出現。每個歸檔備份指令具有下列格式：

```
備份數目 [ -release | -norelease ] [歸檔時間] [取消歸檔時間]
```

歸檔複製可藉由編輯 `archiver.cmd` 檔案指定 (如此處所述)，或使用 File System Manager 軟體指定。若需更多資訊，請參閱 File System Manager 線上手冊。

下列章節將描述歸檔備份指令引數。

歸檔後釋放磁碟空間：`-release`

您可指定，使用備份數目後的 `-release` 指令為歸檔備份後自動釋放的檔案釋放磁碟空間。此選項具有下列格式：

```
-release
```

在程式碼範例 3-17 中，具有群組 `images` 的檔案在其歸檔時間達到 10 分鐘時將被歸檔。在製作歸檔備份 1 後，磁碟快取記憶體空間將釋放。

程式碼範例 3-17 使用 `-release` 指令的 `archiver.cmd` 檔案

```
ex_set . -group images  
1 -release 10m
```

延遲釋放磁碟空間：`-norelease`

您可能直到完成多個歸檔備份才想釋放磁碟空間。`-norelease` 選項可防止自動釋放磁碟空間，直到製作所有標記 `-norelease` 的備份。此選項具有下列格式：

```
-norelease
```

`-norelease` 選項使歸檔組在所有副本都已歸檔後釋放，但是檔案直到呼叫釋放器且將其選為釋放候選者前都不會釋放。

程式碼範例 3-18 指定名為 `vault_tapes` 的歸檔組。將建立兩個備份，但是與此歸檔組相關的磁碟快取記憶體，將直到製作兩個備份才被釋放。

程式碼範例 3-18 使用 `-norelease` 指令的 `archiver.cmd` 檔案

```
vault_tapes  
1 -norelease 10m  
2 -norelease 30d
```

請注意，單一備份上的 `-norelease` 指定對自動釋放沒有影響，因為直到檔案至少具有一個歸檔備份才可被釋放。

同時使用 -release 和 -norelease

如果您想確認在所有歸檔組副本都已歸檔後立即釋放磁碟空間，您可以同時使用 -release 和 -norelease 選項。-release 和 -norelease 的組合會使歸檔器在所有副本都有此組合時釋放。有了此用法，磁碟空間即會立即釋放，而不用等候呼叫釋放器，單獨使用 -norelease 選項的情況也是如此。

設定歸檔時間

您可在指令的第二欄位中指定歸檔時間，以設定檔案的歸檔時間。歸檔時間可用具有如 h (代表小時) 或 m (代表分) 等後綴字元來指定。第 41 頁的表 3-2 「archiver.cmd 檔案指令單位」顯示後綴字元及其意義的完整清單。

在程式碼範例 3-19 中，data 目錄中的檔案在其歸檔時間達到一小時的時候將被歸檔。

程式碼範例 3-19 指定歸檔時間的 archiver.cmd 檔案

```
ex_set data
  1 1h
```

自動取消歸檔

如果您指定檔案的多個歸檔備份，則可能自動取消歸檔除了一個之外的所有備份。使用不同歸檔時間將檔案歸檔至各種媒體時，可能會發生此情況。

程式碼範例 3-20 顯示指定取消歸檔時間的指令。

程式碼範例 3-20 指定取消歸檔時間的 archiver.cmd 檔案

```
ex_set home/users
  1 6m 10w
  2 10w
  3 10w
```

路徑 home/users 中檔案的第一個備份，將在修改後六分鐘被歸檔。在檔案達到 10 週時，將製作第二與第三個歸檔備份。然後將取消歸檔第一個備份。

若需有關控制取消歸檔的更多方法，請參閱第 67 頁「控制取消歸檔」。

為中介資料指定多個備份

如果需要一個以上的中介資料備份，您可以在 fs= 指令之後，立即將備份定義置於程序檔案中。

程式碼範例 3-21 顯示需要指定多個中介資料備份的 archiver.cmd 檔案。

程式碼範例 3-21 指定多個中介資料備份的 archiver.cmd 檔案

```
fs = samfs7
  1 4h
  2 12h
```

在此範例中，四小時後將製作 samfs7 檔案系統的中介資料之備份 1，12 小時後將製作第二個備份。

檔案系統中介資料包括對檔案系統中路徑名稱的變更。由於此原因，如果您經常變更目錄，將建立新歸檔備份。這將導致經常載入為中介資料指定的磁碟區。

歸檔組備份參數

archiver.cmd 檔案的歸檔組參數區段是以 params 指令做為開頭，以 endparams 指令做為結尾。程式碼範例 3-22 顯示歸檔組指令的格式。

程式碼範例 3-22 歸檔組備份參數格式

```
params
歸檔組名稱.備份數目[R] [ -參數 1 -參數 2 ...]
.
.
.
endparams
```

表 3-13 歸檔組備份參數的引數

引數	意義
歸檔組名稱	歸檔組的站點定義名稱。通常是表示屬於該歸檔組的檔案特性。可以是 allsets。歸檔組名稱限制為字母、數字及底線字元 ()。不允許使用其他特殊字元或空格。歸檔組名稱的第一個字元必須是字母。
.	句號 (.) 字元。用來從備份數目隔開歸檔組名稱。
備份數目	定義歸檔備份數目的整數。可以是 1、2、3 或 4。
R	指定被定義的參數是用於此歸檔組的重新歸檔備份。例如，您可以使用 R 並指定 -參數 1 引數中的 VSN，以將重新歸檔備份導向特定磁碟區。
-參數 1	一或多個參數。以下子區段描述可在 params 和 endparams 指令之間指定的參數。
-參數 2	

歸檔組複製參數可藉由編輯 archiver.cmd 檔案指定 (如此處所示)，或使用 File System Manager 軟體指定。若需更多資訊，請參閱 File System Manager 線上手冊。

虛擬歸檔組 `allsets` 可提供為所有歸檔組設定預設歸檔組目錄的方法。所有 `allsets` 指令必須在那些實際的歸檔組備份指令之前出現。個別歸檔組備份的設定參數將覆寫 `allsets` 指令設定的參數。要取得更多有關 `allsets` 歸檔組的資訊，請參閱 `archiver.cmd(4)` 線上手冊。

所有歸檔組處理參數將在此章節中有說明，磁碟歸檔參數除外。若需有關磁碟歸檔參數的資訊，請參閱第 78 頁「關於磁碟歸檔」。

控制歸檔檔案的大小：-archmax

`-archmax` 指令可設定歸檔組的最大檔案大小。其格式如下所示：

```
-archmax 目標大小
```

此指令相當近似於 `archmax` 全域指令。要取得有關此指令及 `目標大小` 中可輸入的值之資訊，請參閱第 44 頁「`archmax` 指令：控制歸檔檔案的大小」。

設定歸檔器的緩衝區大小：-bufsize

依據預設，在將歸檔檔案寫入歸檔媒體之前，此檔案將儲存在緩衝區的記憶體中。您可使用 `-bufsize` 參數指定非預設的緩衝區大小。這些動作可以提高效能，並且您可嘗試各種緩衝區大小值。

此參數具有下列格式：

```
-bufsize=緩衝區大小
```

關於緩衝區大小，請指定 2 至 32 的數字。預設值為 4。此值會與該媒體類型的 `dev_blksize` 值相乘，而使用產生的緩衝區大小。`dev_blksize` 可在 `defaults.conf` 檔案中指定。要取得更多有關此檔案的資訊，請參閱 `defaults.conf(4)` 線上手冊。

例如，此參數可在 `archiver.cmd` 檔案的以下文字行中指定：

```
myset.1 -bufsize=6
```

亦可透過指定 `bufsize=媒體 緩衝區大小` 指令，以在全域基礎上指定此指令。若需更多有關此主題的資訊，請參閱第 44 頁「`bufsize` 指令：設定歸檔器的緩衝區大小」。

指定用於歸檔請求的磁帶機數量：-drivemax、-drivemin 及 -drives

依據預設，歸檔器僅使用一個磁碟機以歸檔歸檔組中的檔案。在歸檔組具有許多檔案或大檔案時，這將有利於使用多個磁碟機。此外，如果您的自動程式庫中的磁帶機以不同的速度運作，使用這些指令可促進歸檔效率。

程式碼範例 3-23 和表 3-14 顯示您可以用來分割磁帶機上的歸檔請求及平衡磁帶機傳輸速度差異的參數。

程式碼範例 3-23 -drivemax、-drivemin 及 -drives 指令的格式

```
-drivemax 最大大小  
-drivemin 最小大小  
-drives 數目
```

表 3-14 -drivemax、-drivemin 及 -drives 參數的引數

引數	意義
最大大小	要使用一個磁帶機進行歸檔的資料最大數量。
最小大小	要使用一個磁帶機進行歸檔的資料最小數量。預設為 -archmax 目標大小 值 (如有指定) 或是其媒體類型的預設值。 如果您指定 -drivemin <i>minsize</i> 參數，Sun StorEdge SAM-FS 只有在有足夠的工作能夠確保時，才會使用多個磁碟機。您可以將 <i>minsize</i> 設為大到足以讓傳輸時間明顯長於卡匣更換時間 (載入、定位、卸載)，並以此做為準則。
數目	要在歸檔此歸檔組時使用的磁碟機數量。預設為 1。

歸檔請求會針對指定的參數加以評估，詳情如下：

- 如果歸檔請求小於**最小大小**，將僅有一個磁碟機用於寫入歸檔請求。
- 如果歸檔請求大於**最小大小**，歸檔請求將由**最小大小**評估，並且磁帶機的適當數目最多可排定為指定的全部磁碟機。
- 如果**最小大小**是零，將嘗試在指定磁碟機的全部數目中進行分割。

在使用 -drives 參數時，多個磁碟機僅用於立即歸檔大於**最小大小**的資料。同時使用的磁碟機數目是**歸檔請求大小總數/最小大小**與 -drives 參數指定的磁碟機數目的較小者。

若要切分磁碟機中的歸檔請求，但要避免連結具有小歸檔請求的所有磁碟機，則可使用 -drivemin 和 -drives 參數。這可能套用於使用較大檔案的操作。

若要設定這些參數，使用者需要考量檔案建立速率、磁碟機數量、載入和卸載磁碟機所需的時間及磁碟機傳輸速率。

範例 1。 假設您要在五個磁碟機之間分割名為 `big_files` 的歸檔組。根據其大小，此歸檔組可按表 3-15 中所示進行分割。

表 3-15 歸檔組範例分割

歸檔組大小	磁帶機數目
< 20 GB	1
≥ 20 GB 至 < 30 GB	2
≥ 30 GB 至 < 40 GB	3
≥ 40 GB 至 < 50 GB	4
≥ 50 GB	5

程式碼範例 3-24 顯示 `archiver.cmd` 檔案中用來分割多個磁碟機上的歸檔請求的指令行。

程式碼範例 3-24 用來分割多個磁碟機上的歸檔請求的指令

```
params
bigfiles.1 -drives 5 -drivemin 10G
endparams
```

範例 2。 `archiver.cmd` 檔案中已指定下行：

```
huge_files.2 -drives 2
```

在歸檔組 `huge_files.2` 中的檔案總大小等於或大於媒體 `drivemin` 的兩倍時，將有兩個磁碟機可用於歸檔檔案。

最大化磁碟區上的空間：-fillvsns

依據預設，歸檔器在寫入歸檔備份時，會使用所有指定給歸檔組的磁碟區。在寫入歸檔備份時，歸檔器會選取具有空間足以容納所有檔案的磁碟區。此動作可讓磁碟區不至於到達容量上限。如果指定了 `-fillvsns`，歸檔器會將歸檔請求分隔為數個較小的群組。

指定歸檔緩衝區鎖定：-lock

依據預設，在將歸檔檔案寫入歸檔媒體之前，此檔案將儲存在緩衝區的記憶體中。如果已啟動直接 I/O，您可使用 `-lock` 參數鎖定此緩衝區。此動作可以提高效能，並且您可嘗試此參數。

此參數具有下列格式：

```
-lock
```

`-lock` 參數表示在製作歸檔備份時，歸檔器是否應該使用鎖定的緩衝區。如果指定了 `-lock`，歸檔器在 `sam-arcopy(1M)` 作業過程中會將檔案鎖定在記憶體中的歸檔緩衝區內。這樣可避免分頁緩衝區，並可提高效能。

`-lock` 參數應該僅在具有大容量記憶體的大系統中指定。記憶體不足會導致出現記憶體不夠用的情況。

僅在為要歸檔的檔案啟動了直接 I/O 時，`-lock` 參數才會生效。依據預設，`-lock` 參數並未指定，且檔案系統會鎖定所有直接 I/O 緩衝區，包括要歸檔的緩衝區。要取得更多有關啟動直接 I/O 的資訊，請參閱 `setfa(1)` 線上手冊、`sam_setfa(3)` 程式庫常式線上手冊，或 `mount_samfs(1M)` 線上手冊上的 `-O forcedirectio` 選項。

例如，此參數可在 `archiver.cmd` 檔案的以下文字行中指定：

```
yourset.3 -lock
```

您亦可透過指定 `bufsize=媒體緩衝區大小 [lock]` 指令的 `lock` 引數，以在全域基礎上指定此指令。若需更多有關此主題的資訊，請參閱第 44 頁「`bufsize` 指令：設定歸檔器的緩衝區大小」。

製作離線檔案的歸檔備份：`-offline_copy`

檔案為在一個歸檔備份製作完畢後，要被釋放的候選者。如果檔案在所有歸檔備份製作完畢之前就釋放並離線，歸檔器會使用此參數決定在製作其他歸檔備份時要使用的方法。在選擇使用方法時，請注意 Sun StorEdge SAM-FS 系統可用的磁碟機數量及可用的磁碟快取記憶體量。此參數具有下列格式：

```
-offline_copy 方法
```

關於方法，請指定顯示的其中一個關鍵字。

表 3-16 -offline_copy 指令的方法引數的值

方法	意義
none	在複製到歸檔磁碟區之前，視各個檔案的需要分段備份檔案。預設值。
direct	直接將檔案從離線磁碟區複製到歸檔磁碟區，而不使用快取記憶體。此方法假設來源磁碟區和目標磁碟區是不同的磁碟區，而且兩個磁帶機皆可用。如果指定了這種方法，請將 <code>stage_n_window</code> 掛載選項的值提高到大於其預設 256 KB 的值。要取得更多有關掛載選項的資訊，請參閱 <code>mount_samfs(1M)</code> 線上手冊。
stageahead	在歸檔另一個檔案的同時，分段備份一個檔案。如有指定，系統會在將檔案寫入其目標時，同時分段備份下一個歸檔檔案。
stageall	在進行歸檔之前，將所有檔案分段備份至磁碟快取記憶體。此方法只使用一個磁帶機，並假設磁碟快取記憶體上的空間足以容納所有檔案。

指定回收

回收程序允許您重新恢復過期的歸檔影像佔用的歸檔磁碟區空間。依據預設，不會產生回收。

如果您要進行回收，您可以在 `archiver.cmd` 檔案和 `recycler.cmd` 檔案中指定指令。要取得更多有關 `archiver.cmd` 檔案支援的回收指令之資訊，請參閱第 129 頁「回收」。

關聯歸檔：-join path

如果您指定了 `-join path` 參數，歸檔器就會使用關聯歸檔。如果要將整個目錄檔至一個磁碟區，而且您知道歸檔檔案實際僅可位於一個磁碟區上，關聯歸檔則很實用。否則，如果您要將目錄保留在一起，可使用 `-sort path` 或 `-rsort path` 參數以保持檔案的連續性。`-rsort` 會執行反向排序。

在歸檔器將歸檔檔案寫入磁碟區時，它會有效率地將使用者檔案納入磁碟區中。隨後，在存取相同目錄中的檔案時，您會遇到延遲情況，因為分段備份程序會透過磁碟區重新定位以讀取下一個檔案。要減緩延遲，您可在歸檔檔案中連續歸檔相同目錄路徑中的檔案。關聯歸檔程序將覆寫空間效率演算法，以將相同目錄中的檔案歸檔在一起。`-join path` 參數允許這些檔案在歸檔組備份中連續歸檔。

在檔案內容沒有變更、但您總要同時存取一組檔案時，關聯歸檔將很有用。例如，您可以在醫院使用關聯歸檔以存取醫療影像。與同一個患者相關的影像可保留在目錄中，醫師可能要同時存取那些影像。如果根據這些靜態影像的目錄位置進行連續歸檔，您可更有效地存取它們。例如：

```
patient_images.1 -join path
```

備註 - `-join path` 參數會將相同目錄中的資料檔寫入相同的歸檔檔案。如果有許多包含小檔案的目錄，歸檔器將建立許多小歸檔檔案。這些較小、不連續的歸檔檔案將降低系統的寫入效能，因為資料檔與每個歸檔檔案的 `tar(1)` 標頭相比相對較小。這樣，在寫入高速磁帶機時，可能會影響其效能。

而且，因為 `-join path` 參數指定相同目錄中的所有檔案將歸檔在單一磁碟區，所以有可能一組檔案不符合任何可用的磁碟區。在此情況下，檔案直到將更多的磁碟區指派給歸檔組才被歸檔。亦可能會出現要歸檔的檔案群組太大，無法符合單一磁碟區的情況。在此情況下，檔案將永不歸檔。

對於大多應用程式而言，若未請求使用較具限制性的 `-join path` 作業，則會較偏好使用 `-sort path` 或 `-join path` 參數。

亦可能按時間、大小或路徑將歸檔組備份中的檔案排序。`age` 和 `size` 引數不能同時使用。程式碼範例 3-25 顯示如何使用 `-sort` 參數加上 `age` 或 `size` 引數，來排序歸檔組。

程式碼範例 3-25 排序歸檔組的指令

```
cardiac.1 -sort path
cardiac.2 -sort age
catscans.3 -sort size
```

第一行會強制歸檔器按路徑名稱將歸檔請求排序。第二個範例行會強制歸檔器將名為 `cardiac.2` 的歸檔組備份按從檔案最舊到最近的時間進行排序。第三行會強制將名為 `catscans` 的歸檔組備份按從最小到最大的檔案大小進行排序。如果您本來打算使用反向排序，您當時可以指定 `-rsort` 以取代 `-sort`。

控制取消歸檔

取消歸檔是刪除檔案或目錄的歸檔項目之程序。根據預設值，檔案將永不取消歸檔。檔案將根據上次存取的時間被取消歸檔。所有經常存取的資料將被儲存在快速媒體上，如磁碟；所有時間較早、不經常存取的資料可被儲存在磁帶上。

範例 1。程式碼範例 3-26 顯示 archiver.cmd 檔案。

程式碼範例 3-26 控制取消歸檔的指令

```
arset1 dir1
  1    10m    60d
  2    10m
  3    10m
vsns
arset1.1    mo    OPT00 [0-9]
arset1.2    lt    DLTA0 [0-9]
arset1.3    lt    DLTB0 [0-9]
```

如果程式碼範例 3-26 中顯示的 archiver.cmd 檔案控制的檔案經常被存取，它將一直保留在磁碟中，即使它早於 60 天。只要檔案在 60 天內沒有被存取，備份 1 資訊將被移除。

如果備份 1 資訊被移除 (因為 60 天沒有存取檔案)，且某人從備份 2 中分段備份檔案，它將被從磁帶中讀取。在檔案回到線上狀態時，歸檔器將在磁碟上製作新備份 1，並且 60 天的存取循環將重新啟動。如果再次存取檔案，Sun StorEdge SAM-FS 歸檔器將重新產生備份 1。

範例

假設患者在醫院四週。在此期間，此患者的所有檔案將儲存在快速媒體上 (備份 1=mo)。四週後，患者將出院。如果在此患者出院後 60 天沒有人存取其資料，inode 中的備份 1 記錄將被取消歸檔，並且僅有備份 2 與備份 3 記錄可用。磁碟區現在即被回收，以便為目前更多的患者留出空間，而無需增加程式庫。如果患者六個月後回醫院檢查，第一次將從磁帶 (備份 2) 存取資料。現在，歸檔器將自動在磁碟上建立新備份 1，以確定在檢查期間 (需要幾天或幾週) 可從快速媒體上獲得資料。

控制如何寫入歸檔檔案：-tapenonstop

依據預設，歸檔器可在歸檔檔案之間寫入一個磁帶標記、一個 EOF 標籤及再兩個磁帶標記。在啟動下一個歸檔檔案時，磁帶機將回到導致效能喪失的第一個磁帶標記後之位置。-tapenonstop 參數將指示歸檔器僅寫入初始的磁帶標記。此外，如果指定 -tapenonstop 參數，歸檔器將在備份作業結束時輸入歸檔資訊。

若需更多有關 -tapenonstop 參數的資訊，請參閱 archiver.cmd(4) 線上手冊。

預留磁碟區：-reserve

依據預設，歸檔器會將歸檔組備份寫入任何由常規表示式指定的磁碟區，該常規表示式在 `archiver.cmd` 檔案的磁碟區關聯部分有說明。但是，歸檔組磁碟區有時會僅包含一個歸檔組中的檔案。預留磁碟區程序可用於滿足此資料儲存裝置的請求。

備註 - `-reserve` 參數可預留由一個歸檔組單獨使用的磁碟區。使用預留磁碟區的站點可能會引起更多的卡匣載入與卸載。

`-reserve` 參數可為歸檔組預留磁碟區。在設定 `-reserve` 參數且將磁碟區指定歸檔組備份之後，歸檔識別標記將不會被指定給任何其他歸檔組備份，即使有常規表示式與其相符。

在選擇歸檔組使用的磁碟區時，將有預留名稱指派給磁碟區。預留名稱是將歸檔組指派給磁碟區的唯一識別標記。

`-reserve` 參數的格式如下所示：

```
-reserve 關鍵字
```

指定的**關鍵字**視您使用的表單而定。可能的表單是歸檔組表單、擁有者表單及檔案系統表單，如下所示：

- 歸檔組表單。此表單使用 `set` **關鍵字**，如下所示：`-reserve set`
- 擁有者表單。此表單使用以下其中一個 *keyword*：`dir`、`user`、或 `group`。程式碼範例 3-27 顯示這些指令的格式。

程式碼範例 3-27 `-reserve` 參數的所有者表單

```
-reserve dir  
-reserve user  
-reserve group
```

程式碼範例 3-27 中顯示的三個所有者表單不可同時使用。換言之，僅有三個擁有者表單中的一個可用於歸檔組與備份。

- 檔案系統表單。此表單使用 `fs` **關鍵字**，如下所示：`-reserve fs`

在 `archiver.cmd` 檔案中，您可為一個、兩個或所有三個可能的表單指定 `-reserve` 參數。這三個表單可以組合，並可在歸檔組參數定義中一起使用。

例如，程式碼範例 3-28 顯示 archiver.cmd 檔案片段。以 arset.1 做為開頭的那一行會根據歸檔組、群組及檔案系統來建立預留名稱。

程式碼範例 3-28 具有預留磁碟區的 archiver.cmd 檔案

```
params
arset.1 -reserve set -reserve group -reserve fs
endparams
```

有關保留磁碟區的資訊將儲存在程式庫目錄中。程式庫目錄中的行包含媒體類型、VSN、預留資訊及預留日期與時間。預留資訊包括由正斜線 (/) 分隔的歸檔組元件、路徑名稱元件及檔案系統元件。

這些正斜線並不表示路徑名稱；它們僅是顯示預留名稱的三個元件之分隔字元。如程式碼範例 3-29 所示，描述預留磁碟區的文字行在程式庫目錄中是以 #R 字元做為開頭。

程式碼範例 3-29 顯示預留磁碟區的程式庫目錄檔

```
6 00071 00071 lt 0xe8fe 12 9971464 1352412 0x6a000000 131072 0x
# -il-o-b----- 05/24/00 13:50:02 12/31/69 18:00:00 07/13/01 14:03:00
#R lt 00071 arset0.3// 2001/03/19 18:27:31
10 ST0001 NO_BAR_CODE lt 0x2741 9 9968052 8537448 0x68000000 1310
# -il-o----- 05/07/00 15:30:29 12/31/69 18:00:00 04/13/01 13:46:54
#R lt ST0001 hgm1.1// 2001/03/20 17:53:06
16 SLOT22 NO_BAR_CODE lt 0x76ba 6 9972252 9972252 0x68000000 1310
# -il-o----- 06/06/00 16:03:05 12/31/69 18:00:00 07/12/01 11:02:05
#R lt SLOT22 arset0.2// 2001/03/02 12:11:25
```

請注意，程式碼範例 3-29 中的某些文字行已刪略以符合頁面。

視在 archiver.cmd 檔案中定義的選項而定，一個或多個預留資訊欄位可以是空的。日期與時間表示制定預留的時間。預留行可為每個在歸檔時預留給歸檔組的磁碟區附加至檔案。

您可以使用 samu(1M) 公用程式的 v 顯示，或以程式碼範例 3-30 中所示的其中一種格式使用 archiver(1M) 或 dump_cat(1M) 指令，來顯示預留資訊。

程式碼範例 3-30 用於顯示預留資訊的指令

```
archiver -lv
dump_cat -v 目錄名稱
```

下列格式將展現顯示參數、關鍵字及指派給磁碟區的預留名稱之範例的每個表單。

- 歸檔組表單。如表 3-17 所示，set 關鍵字會啓動保留名稱中的歸檔組元件。

表 3-17 歸檔組表單範例

指令與關鍵字	預留名稱範例
-reserve set	users.1// Data.1//

例如，在程式碼範例 3-31 中，archiver.cmd 檔案片段顯示以 allsets 歸檔組名稱做為開頭的那一行，將按歸檔組為所有歸檔組設定預留。

程式碼範例 3-31 按歸檔組預留磁碟區

```
params
allsets -reserve set
endparams
```

- 擁有者表單。dir、user 及 group 關鍵字將啓動預留名稱中的所有者元件。dir、user 及 group 關鍵字不可同時使用。dir 關鍵字立即使用遵循歸檔組定義路徑指定的目錄路徑元件。user 與 group 關鍵字意義明顯而無需加以說明。表 3-18 顯示相關範例。

表 3-18 擁有者組表單範例

指令與關鍵字	預留名稱範例
-reserve dir	proj.1/p105/ proj.1/p104/
-reserve user	users.1/user5/ users.1/user4/
-reserve group	data.1/engineering/

備註 — -reserve 參數計劃預留由一個歸檔組單獨使用的磁碟區。許多包含小檔案的目錄，會導致許多小歸檔檔案被寫入每個預留的磁碟區。這些較小、不連續的歸檔檔案將降低系統的效能，因為資料檔與每個歸檔檔案的 tar(1) 標頭相比相對較小。

- 檔案系統表單。fs 關鍵字會啓動「保留名稱」中的檔案系統元件。表 3-19 顯示相關範例。

表 3-19 檔案系統表單範例

指令與關鍵字	預留名稱範例
-reserve fs	proj.1/p103/samfs1
	proj.1/p104/samfs1

第 95 頁「範例 4」顯示使用預留磁碟區的完整歸檔範例。

歸檔器會記錄程式庫目錄檔中的磁碟區預留情況。磁碟區在重新製作標籤時將自動取消預留，因為歸檔資料已被有效刪除。

您也可以使用 reserve(1M) 和 unreserve(1M) 指令來預留和取消預留磁碟區。要取得更多有關這些指令的資訊，請參閱 reserve(1M) 和 unreserve(1M) 線上手冊。

設定歸檔優先權：-priority

Sun StorEdge SAM-FS 檔案系統提供用於歸檔檔案的可配置優先權系統。每個檔案將指定給可從檔案屬性，及可在 archiver.cmd 檔案中為每個歸檔組設定的優先權倍增器計算之優先權。這些屬性包括線上/離線、時間、製作的備份數目及大小。

根據預設值，歸檔請求中的檔案沒有排序，並且所有屬性倍增器都是零。這將導致檔案以第一次建立、第一次歸檔的順序進行歸檔。要取得更多有關這些優先權的資訊，請參閱 archiver(1M) 和 archiver.cmd(4) 線上手冊。

您可控制檔案按設定優先權與排序方法歸檔的順序。以下是您可設定的優先權範例。

- 選擇 priority 排序方法以便按優先權順序歸檔歸檔請求中的檔案。
- 變更 archive_loaded 優先權以減少媒體載入。
- 變更 offline 優先權以使線上檔案在離線檔案之前進行歸檔。
- 變更 copy# 優先權以便按備份順序製作歸檔備份。

表 3-20 列出歸檔優先權。

表 3-20 歸檔優先權

歸檔優先權	定義
-priority age 值	歸檔時間屬性倍增器
-priority archive_immediate 值	歸檔立即屬性倍增器
-priority archive_overflow 值	多個歸檔磁碟區屬性倍增器
-priority archive_loaded 值	歸檔磁碟區載入屬性倍增器
-priority copy1 值	備份 1 屬性倍增器
-priority copy2 值	備份 2 屬性倍增器
-priority copy3 值	備份 3 屬性倍增器
-priority copy4 值	備份 4 屬性倍增器
-priority copies 值	備份製作屬性倍增器
-priority offline 值	檔案離線屬性倍增器
-priority queuwait 值	佇列等待屬性倍增器
-priority rearchive 值	重新歸檔屬性倍增器
-priority reqrelease 值	Reqrelease 屬性倍增器
-priority size 值	檔案大小屬性倍增器
-priority stage_loaded 值	分段備份磁碟區載入屬性倍增器
-priority stage_overflow 值	多個分段備份磁碟區屬性倍增器

關於值，請在以下範圍內指定浮點數字：

$-3.400000000E+38 \leq \text{值} \leq 3.402823466E+38$

排定歸檔：-startage、-startcount 及 -startsize

在歸檔器掃描檔案系統時，其會辨識要歸檔的檔案。辨識為歸檔候選者的檔案會列入一個所謂的歸檔請求清單。在檔案系統掃描結束時，系統會排定歸檔請求以進行歸檔。-startage、-startcount 及 -startsize 歸檔組參數可控制歸檔工作量，並確保適時進行檔案歸檔。表 3-21 顯示用於這些參數的格式。

表 3-21 -startage、-startcount 及 -startsize 指令格式

指令	意義
-startage 時間	指定時間量，其為在掃描中第一個被標示為包含在歸檔請求的檔案與開始歸檔之間所經歷的時間。關於 <i>time</i> ，請以第 60 頁「設定歸檔時間」中所用的格式指定時間。若是未設定此變數，則會使用 <i>interval</i> 指令。
-startcount 計數	指定要包含在歸檔請求中的檔案數量。當歸檔請求中的檔案數量到達 <i>count</i> 時，歸檔就會開始進行。關於計數，請指定一個整數。依據預設， <i>count</i> 並未設定。
-startsize 大小	指定歸檔請求中要歸檔的所有檔案之最小的總大小 (以位元組為單位)。歸檔工作會加以累積，而當檔案總大小到達「大小」時，歸檔就會開始進行。依據預設，大小並未設定。

examine=方法 指令和 *interval=時間* 指令為與 -startage、-startcount 及 -startsize 指令具有互動關係的指令。-startage、-startcount 及 -startsize 指令會針對完成的歸檔工作最佳化歸檔適時性。這些值會覆寫 *examine=方法* 規定 (如果有的話)。要取得更多有關 *examine* 指令的資訊，請參閱第 46 頁「*examine* 指令：控制歸檔掃描」。要取得更多有關 *interval* 指令的資訊，請參閱第 47 頁「*interval* 指令：指定歸檔間隔」。

archiver.cmd 檔案中只能指定每個歸檔副本的 -startage、-startcount 及 -startsize 指令。如果指定了一個以上的上述指令，將會針對第一個遇到的情況開始進行歸檔作業。若皆未指定 -startage、-startcount 或 -startsize，則會根據 *examine=方法* 指令排定歸檔請求，如下所示：

- 如果 *examine=noscan*，歸檔請求則會在歸檔請求中遇到的第一個檔案之後，根據 *interval=時間* 指令的規定而排定。此為連續歸檔。預設為 *examine=noscan*。
- 如果 *examine=scan* | *scaninodes* | *scandirs*，歸檔請求則會在檔案系統掃描之後，排定進行歸檔的時間。

archiver.cmd(4) 線上手冊含有示範如何使用這些指令的範例。

VSN 關聯指令

archiver.cmd 檔案的 VSN 關聯區段會將磁碟區指定給歸檔組。此區段以 `vsns` 指令開頭，以 `endvsns` 指令結尾。

VSN 指令也可使用 File System Manager 軟體配置。請參閱檔案系統管理線上手冊以取得更多資訊。

磁碟區集合將被下列表單中的指令指派給歸檔組：

```
歸檔組名稱.備份數目 媒體類型 vsn_expr ... [ -pool vsn 池名稱... ]
```

表 3-22 VSN 關聯指令的引數

引數	意義
歸檔組名稱	歸檔組的站點定義名稱。必須是歸檔組指派指令中的第一個欄位。歸檔組名稱通常表示屬於歸檔組的檔案特性。歸檔組名稱限制為字母、數字及底線字元 (<code>_</code>)。不允許使用其他特殊字元或空格。歸檔組名稱的第一個字元必須是字母。
備份數目	後面是為此備份指定歸檔特性的一個或多個引數之數字。歸檔備份指令以數字開頭。此數字 (1、2、3 或 4) 是備份數目。
媒體類型	表示媒體類型。如需有效媒體類型清單，請參閱 <code>mcf(4)</code> 線上手冊。
<code>vsn</code> 表示式	表示一般表示式。請參閱 <code>regex(5)</code> 線上手冊。
<code>-pool vsn</code> 儲存池名稱	表示已命名的 VSN 集合。

關聯至少需要三個欄位：歸檔組名稱、備份數目及媒體類型，以及至少一個磁碟區。歸檔組名稱與備份數目由小數點 (`.`) 連接。

下列範例將以不同的方法指定相同的 VSN。

範例 1。程式碼範例 3-32 顯示兩行 VSN 規定。

程式碼範例 3-32 VSN 規定 — 範例 1

```
vsns
set.1 lt VSN001 VSN002 VSN003 VSN004 VSN005
set.1 lt VSN006 VSN007 VSN008 VSN009 VSN010
endvsns
```

範例 2。 程式碼範例 3-33 顯示使用可延續至下一行的反斜線字元 (\) 之 VSN 指定。

程式碼範例 3-33 VSN 規定 — 範例 2

```
vsns
set.1 lt VSN001 VSN002 VSN003 VSN004 VSN005 \
  VSN006 VSN007 VSN008 VSN009 VSN010
endvsns
```

範例 3。 程式碼範例 3-34 將以隨手的註解指定使用常規表示式的 VSN。

程式碼範例 3-34 VSN 規定 — 範例 3

```
vsns
set.1 lt VSN0[1-9] VSN10
endvsns
```

磁碟區將由一個或多個 *vsn* 表示式關鍵字註解，這些關鍵字是常規表示式，在 `regexp(5)` 線上手冊中有說明。請注意，這些一般表示式與萬用字元遵循不同的規則。除一般表示式之外，您亦可指定從其中選擇磁碟區的 VSN pool。Pool 使用具有 VSN 關聯的 `-pool vsn 儲存池名稱` 指令表示。

在歸檔組的歸檔器需要磁碟區時，系統將檢查所有自動程式庫與手動掛載磁帶機中選定媒體類型的每個磁碟區，以決定它是否滿足任何 VSN 運算式。將選擇符合包含歸檔備份作業所需足夠空間的運算式之第一個磁碟區。例如：

- 以下指令將指定屬於備份 1 的歸檔組 `ex_set` 之檔案將被複製到媒體類型 `mo`，此媒體類型使用二十個磁碟區中任何一個具有名稱 `optic20` 至 `optic39` 的磁碟區：

```
ex_set.1 mo optic[2-3][0-9]
```

- 以下指令會將屬於備份 2 的歸檔組 `ex_set` 複製到媒體類型 `lt`，此媒體類型具有任何以 `TAPE` 開頭的磁碟區：

```
ex_set.2 lt ^TAPE
```

如果您的 Sun StorEdge SAM-FS 環境是配置為按歸檔組回收，請勿將 VSN 指定給一個以上的歸檔組。

備註 — 在設定 `archiver.cmd` 檔案時，確定已為中介資料將磁碟區指定給歸檔組。每個檔案系統都具有一個與其有相同名稱的歸檔組。若需更多有關保留中介資料的資訊，請參閱 `samfsdump(1M)` 線上手冊，或參閱「Sun StorEdge SAM-FS 疑難排解指南」。

VSN 池指令

archiver.cmd 檔案的 VSN pool 區段是以 vsnpools 指令做為開頭，以 endvsnpools 指令做為結尾，或以 archiver.cmd 檔案的尾端做為結尾。此部份可命名磁碟區集合。

VSN 池也可使用 File System Manager 軟體配置。請參閱檔案系統管理線上手冊以取得更多資訊。

VSN pool 是已命名的磁碟區集合。VSN 儲存池對於定義可用於歸檔組的磁碟區很有用。同樣，VSN 儲存池可提供有用的緩衝區以便將磁碟區指派、預留給歸檔組。

您可使用 VSN 儲存池，以便按組織內的部門、群組中的使用者、資料類型及其他方便的群組定義要使用的獨立群組。系統將指派給 pool 名稱、媒體類型及一組磁碟區。臨時儲存池是在 VSN 關聯中的指定磁碟區用盡或其他 VSN 儲存池用盡時使用的一組磁碟區。若需更多有關 VSN 關聯的資訊，請參閱第 75 頁「VSN 關聯指令」。

磁碟區已被預留，它將對從其中產生它的儲存池不再可用。因此，已命名儲存池中的磁碟區數目將隨使用的磁碟區變更。您可按以下格式輸入 archiver(1M) 指令以檢視 VSN 儲存池：

```
# archiver -lv | more
```

VSN pool 定義至少需要三個欄位 (以空格隔開)：儲存池名稱、媒體類型，以及至少一個 VSN。其語法如下所示：

```
vsn 儲存池名稱 媒體類型 vsn 表示式
```

表 3-23 VSN Pool 指令的引數

引數	意義
<i>vsn</i> 儲存池名稱	指定 VSN 儲存池
媒體類型	2 個字元的媒體類型。如需有效媒體類型清單，請參閱 mcf(4) 線上手冊。
<i>vsn</i> 表示式	常規表示式。可以有一或多個 <i>vsn</i> 表示式引數。請參閱 regcmp(3G) 線上手冊。

以下範例使用四個 VSN pool：users_pool、data_pool、proj_pool 及 scratch_pool。如果三個特定 pool 的其中之一是來自於磁碟區，歸檔器則會選取 scratch pool VSN。程式碼範例 3-35 顯示使用四個 VSN pool 的 archiver.cmd 檔案。

程式碼範例 3-35 顯示 VSN Pool 的範例

```
vsnpools
users_pool    mo    ^MO[0-9][0-9]
data_pool     mo    ^DA.*
scratch_pool  mo    ^SC[5-9][0-9]
proj_pool     mo    ^PR.*
endvsnpools
vsns
users.1       mo    -pool users_pool    -pool scratch_pool
data.1        mo    -pool data_pool     -pool scratch_pool
proj.1        mo    -pool proj_pool     -pool scratch_pool
endvsns
```

關於磁碟歸檔

歸檔是將檔案從線上磁碟複製到歸檔媒體的程序。通常，歸檔備份被寫入自動程式庫中的磁光碟匣或磁帶卡匣之磁碟區，但對於磁碟歸檔，檔案系統中的線上磁碟將用作歸檔媒體。

磁碟歸檔可以被執行，以便將檔案從一個 Sun StorEdge SAM-FS 檔案系統歸檔至同一個主機電腦系統上的其他檔案系統。磁碟歸檔亦可被執行，以便將來源檔案歸檔至不同 Sun Solaris 系統上的其他檔案系統。在使用兩個主機系統執行磁碟歸檔時，涉及的系統將充當用戶端與伺服器。用戶端系統是接收來源檔案的系統。伺服器系統是接收歸檔備份的目標系統。

要寫入歸檔檔案的檔案系統可以是任何 UNIX 檔案系統，它不一定是 Sun StorEdge SAM-FS 檔案系統。如果磁碟歸檔備份寫入到其他主機，主機上則必須至少有安裝一個 Sun StorEdge SAM-FS 相容的檔案系統。

歸檔器對歸檔至磁碟區的檔案與歸檔至程式庫中磁碟區的檔案同樣處理。您仍可製作一、二、三或四個歸檔備份。如果您正在製作多個歸檔備份，其中一個歸檔備份可寫入磁碟區，而其他歸檔備份將寫入可移除的媒體磁碟區。此外，如果您通常歸檔至 Sun StorEdge SAM-FS 檔案系統的磁碟區，歸檔檔案備份將根據此檔案系統中的 archiver.cmd 檔案規則自行歸檔。

下列清單說明歸檔至線上磁碟與歸檔至可移除的媒體之間的某些相似與不同之外：

- 與寫入磁光碟或磁帶的歸檔備份不同，寫入磁碟的歸檔備份將不記錄在目錄檔中。此外，磁碟區中的歸檔檔案將不出現在記錄器中。

- 如果您正在歸檔至可移除的媒體磁碟區，您可在掛載檔案系統後開始歸檔，而無需變更 archiver.cmd 檔案中的任何預設值。但是，如果您正在歸檔至磁碟區，您必須在掛載檔案系統之前，先編輯 archiver.cmd 檔案並定義磁碟歸檔組。

- 磁碟歸檔不會依賴 mcf(4) 檔案中的項目。您需要在 archiver.cmd 檔案中指定磁碟歸檔組，並需要在 /etc/opt/SUNWsamfs/diskvols.conf 中定義磁碟區。這是額外的配置設定檔案，如果僅歸檔至可移除的媒體磁碟區，則不需要它。

diskvols.conf 檔案必須在來源檔案所在的系統中建立。視歸檔備份寫入的位置而定，此檔案亦包含下列資訊：

- 如果歸檔備份被寫入相同主機系統上的檔案系統，diskvols.conf 檔案將定義 VSN 與至每個 VSN 的路徑。
- 如果歸檔備份被寫入不同的 Sun Solaris 系統，diskvols.conf 檔案將包含此伺服器系統的主機名稱。在此情況下，伺服器系統上亦必須有 diskvols.conf 檔案，伺服器系統可定義被給予寫入此系統權限的用戶端。如果您要建立此用戶端/伺服器關係，在開始進行第 81 頁「啓動磁碟歸檔」中的程序之前，請先確定做為伺服器的主機上至少有安裝一個 Sun StorEdge SAM-FS 檔案系統。

配置設定指引

雖然對磁碟歸檔磁碟區所在的位置沒有限制，但建議磁碟區位於與原始檔案所在的磁碟不同的磁碟上。用戶端系統中的歸檔備份將優先寫入伺服器系統上的磁碟區。建議您製作多個歸檔備份，並寫入多個類型的歸檔媒體。例如，備份 1、備份 2 與備份 3 可分別歸檔至磁碟區、磁帶及磁光碟。

如果您在將檔案歸檔至伺服器系統上的檔案系統，歸檔檔案可自行歸檔至程式庫中可移除的媒體匣，該程式庫與目標伺服器相連。

磁碟歸檔指令

在歸檔至線上磁碟時，歸檔器可辨識大多的 archiver.cmd 指令。其所辨識的指令可定義歸檔組並配置回收。由於某些指令是專門針對用於可移除的媒體卡匣，所以則會忽略這些在磁碟歸檔環境中不具意義的指令。具體而言，系統會辨識用於磁碟歸檔組的以下指令：

- 第 61 頁「歸檔組備份參數」中的所有回收指令，除了以下指令之外：
 - -fillvsns
 - -ovflmin 最小大小
 - -reserve 方法
 - -tapenonstop
- 第 136 頁「步驟 2：編輯 archiver.cmd 檔案」中的所有指令，除了以下指令之外：
 - -recycle_dataquantity 大小
 - -recycle_vsncount 計數

- `vsns` 和 `endvsns` 指令與 `vsnpools` 和 `endvsnpools` 指令。VSN 關聯章節中支援磁碟區且與 `dk` 媒體檔案定義。磁碟區是由一個或多個 VSN 表示式關鍵字 (一般表示式) 通知。除 VSN 之外，您亦可指定從其中選擇磁碟區的 VSN 池。VSN 池是已命名的磁碟區集合。例如：

程式碼範例 3-36 `vsns` 和 `vsnpools` 指令之範例

```
vsnpools
data_pool dk disk0[0-5]
endvsnpools

vsns
arset0.1 dk disk10 disk1[2-5]
arset1.1 dk -pool data_pool
endvsns
```

- `clients` 和 `endclients` 指令。如果您以將用戶端主機來源歸檔到伺服器主機的方式實行磁碟歸檔，您必須在伺服器主機上配置 `diskvols.conf` 檔案。伺服器系統上的 `diskvols.conf` 檔案必須包含用戶端系統的名稱。這些指令的格式如下所示：

程式碼範例 3-37 `clients` 和 `endclients` 指令的格式

```
clients
用戶端系統 1
用戶端系統 2
...
endclients
```

關於用戶端系統，請指定含有來源檔的用戶端系統的主機名稱。

- `-recycle_minobs percent` 回收器指令。此選項是用來設定磁碟歸檔之回收器重新歸檔程序的執行緒。當磁碟上已歸檔之 `tar` 檔案中過時的檔案的百分比已達此執行緒，回收器會開始從歸檔將有效檔案移動至新的 `tar` 檔案。所有有效檔案都已移動後，原始的 `tar` 檔案會標記為要從磁碟歸檔移除的候選者。可移除的媒體回收會忽略此選項。預設為 50%。

要取得更多有關磁碟歸檔指令的資訊，請參閱 `archiver.cmd(4)` 線上手冊。

▼ 啓動磁碟歸檔

您可以隨時啓動磁碟歸檔。本節中的程序假設您已經在進行歸檔，而且您正將磁碟歸檔新增到您的環境中。如果您將磁碟歸檔啓動做為初始安裝的一部分，相關資訊則請參閱「Sun StorEdge SAM-FS 安裝與升級指南」。如果您在安裝時新增磁碟歸檔，則請勿使用本程序 (因為本程序包含不需要的步驟)。

備註 – 在軟體先前至 4U4 版本中，磁碟歸檔是使用 `params` 節中的 `-disk_archive` 參數啓用 `archiver.cmd` 檔案。不再使用此參數，而且必須編輯以先前軟體版本建立的 `archiver.cmd` 檔案，以於 4U4 軟體中正確運作歸檔。請參閱 `archiver.cmd(4)` 線上手冊以取得詳細資訊。

1. 確定您要寫入磁碟歸檔備份的主機上至少有安裝一個 **Sun StorEdge QFS** 檔案系統。
2. 在含有要歸檔的檔案之主機系統上成為超級使用者。
3. 若要啟動磁碟歸檔，請按照「Sun StorEdge SAM-FS 安裝與升級指南」中的程序進行。Sun StorEdge SAM-FS 初始安裝程序含有一個名為**啟動磁碟歸檔**的步驟。該步驟分為以下兩個程序。
4. 在含有要歸檔的檔案之主機上，使用 `samd(1M) config` 指令傳遞配置檔案變更，然後重新啟動系統。

例如：

```
# samd config
```

5. 在寫入歸檔備份的主機系統上成為超級使用者 (選用)。
只有在要歸檔到其他主機上的磁碟時才執行此步驟。
6. 在要寫入歸檔備份的主機上，使用 `samd(1M) config` 指令傳遞配置檔案變更，然後重新啟動目標系統 (選用)。
只有在要歸檔到其他主機上的磁碟時才執行此步驟。

例如：

```
# samd config
```

磁碟歸檔範例

範例 1

程式碼範例 3-38 顯示常駐於用戶端系統 pluto 上的 diskvols.conf 檔案。

程式碼範例 3-38 pluto 上的 diskvols.conf 檔案

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on pluto
# VSN Name      [Host Name:]Path
#
disk01          /sam_arch1
disk02          /sam_arch2/proj_1
disk03          mars:/sam_arch3/proj_3
disk04          /sam_arch4/proj_4
```

在之前的 diskvols.conf 檔案中，標識為 disk01 與 disk02 和 disk04 的 VNS 將被寫入原始來源檔案所在的主機系統。VSN disk03 將被寫入伺服器系統 mars 上的 VSN。

程式碼範例 3-39 顯示伺服器系統 mars 上的 diskvols.conf 檔案。

程式碼範例 3-39 mars 上的 diskvols.conf 檔案

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on mars
#
clients
pluto
endclients
```

程式碼範例 3-40 顯示 pluto 上的 archiver.cmd 檔案的一個片段。

程式碼範例 3-40 pluto 上的 archiver.cmd 檔案

```
vsns
arset1.2 dk disk01
arset2.2 dk disk02 disk04
arset3.2 dk disk03
endvsns
```

範例 2

在此範例中，檔案 `/sam1/testdir0/filea` 位於 `arset0.1` 的歸檔組中，而歸檔器則會將 `/sam1/testdir0/filea` 的內容備份到名為 `/sam_arch1` 的目標路徑。程式碼範例 3-41 顯示 `diskvols.conf` 檔案。

程式碼範例 3-41 `diskvols.conf` 檔案

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf
#
# VSN Name      [Host Name:]Path
#
disk01          /sam_arch1
disk02          /sam_arch12/proj_1
```

程式碼範例 3-42 顯示針對磁碟歸檔的 `archiver.cmd` 檔案文字行：

程式碼範例 3-42 `archiver.cmd` 檔案中針對磁碟歸檔的指令

```
.
vsns
arset0.1 dk disk01
endvsns
.
```

以下顯示了為歸檔至磁碟的檔案 `filea` 執行 `sls(1)` 指令後的輸出。在程式碼範例 3-43 中，請注意以下內容：

- `dk` 為媒體類型或磁碟歸檔媒體
- `disk01` 為 VSN
- `f192` 為磁碟歸檔 `tar(1)` 檔案的路徑

程式碼範例 3-43 `sls(1M)` 的輸出

```
# sls -D /sam1/testdir0/filea
/sam1/testdir0/filea:
mode: -rw-r----- links: 1 owner: root group: other
length: 797904 admin id: 0 inode: 3134.49
archdone;
copy 1: ---- Dec 16 14:03 c0.1354 dk disk01 f192
access: Dec 19 10:29 modification: Dec 16 13:56
changed: Dec 16 13:56 attributes: Dec 19 10:29
creation: Dec 16 13:56 residence: Dec 19 10:32
```

範例 3

在此範例中，檔案 `/sam2/my_proj/fileb` 在歸檔組 `arset0.1` 中的用戶端主機 `snickers` 上，歸檔器會將此檔案內容複製到伺服器主機 `mars` 上的目標路徑 `/sam_arch1`。

程式碼範例 3-44 顯示 `snickers` 上的 `diskvols.conf` 檔案。

程式碼範例 3-44 `snickers` 上的 `diskvols.conf` 檔案

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on snickers
#
# VSN Name      [Host Name:]Path
#
disk01          mars:/sam_arch1
```

程式碼範例 3-45 顯示 `mars` 上的 `diskvols.conf` 檔案。

程式碼範例 3-45 `mars` 上的 `diskvols.conf` 檔案

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on mars
#
clients
snickers
endclients
```

程式碼範例 3-46 顯示 `archiver.cmd` 檔案中與此範例相關的指令。

程式碼範例 3-46 `archiver.cmd` 檔案中針對磁碟歸檔的指令

```
.
vsns
arset0.1 dk disk01
endvsns
.
```

規劃歸檔作業

歸檔器將使用 `archiver.cmd` 檔案自動執行儲存管理作業。在寫入此檔案時，檢閱某些可提高 Sun StorEdge SAM-FS 檔案系統與歸檔器的一般指引很有用。這樣可確定以最可能安全的方式儲存資料。

每個站點在其計算的應用程式、資料儲存硬體及軟體中都是唯一的。下列建議都是 Sun Microsystems 經驗的結晶。在為站點寫入 `archiver.cmd` 檔案時，請確定已考量以下方面以反映站點的資料儲存請求。

- 儲存歸檔記錄。歸檔記錄檔可提供復原資料的必要資訊，甚至在 Sun StorEdge SAM-FS 軟體無法使用時也可提供。建議將這些記錄保留在安全的地方，以備在 Sun StorEdge SAM-FS 軟體不可用而發生災難性損毀時使用。
- 為磁碟區使用一般表示式。可允許系統將檔案儲存在許多不同的磁碟區，以讓其為您工作。磁碟區範圍 (使用一般表示式指定的) 允許系統連續執行。為歸檔組備份使用指定名稱可快速填充磁碟區，從而在移除一張媒體或使用其他媒體更換它時會導致不適當的工作流程。
- 您可根據建立與修改檔案的頻率、及是否要儲存所有的修改備份確定歸檔間隔。請記住，歸檔間隔是兩次檔案系統掃描之間的時間。非常短的歸檔間隔可讓歸檔器幾乎保持連續掃描。
- 考量使用的檔案系統數目。一般而言，與單一 Sun StorEdge SAM-FS 檔案系統相較之下，多個 Sun StorEdge SAM-FS 檔案系統可以提升歸檔器的效能。歸檔器對每個檔案系統使用單獨的程序。掃描多個檔案系統比掃描單一檔案系統需要的時間相對較少。
- 使用目錄捷徑組織 Sun StorEdge SAM-FS 檔案系統中的檔案。基於效能考量，Sun Microsystems 建議您在一個目錄中儲存的檔案不要超過 10,000 個。
- 務必最少在兩個單獨的磁碟區中製作兩個檔案備份。如果媒體發生實體問題，將資料儲存在單一媒體類型上將很危險。如果可能的話，請不要依賴單一歸檔備份。
- 確定定期使用 `samfsdump(1M)` 傾印中介資料。中介資料 (目錄結構、檔案名稱等等) 將被儲存在與檔案系統具有相同名稱的歸檔組中。您可在發生災難時使用此資訊以復原檔案系統。如果不想這樣做，您可將此歸檔組指派給不存在的 VSN，以阻止此資料歸檔。要取得更多有關預留中介資料的資訊，請參閱「Sun StorEdge SAM-FS 疑難排解指南」或「Sun StorEdge SAM-FS 安裝與升級指南」。

預覽佇列

歸檔器與分段備份程式程序均可請求載入或卸載媒體。如果請求數目超過可用於媒體載入的磁碟機數目，則超過該數目的請求將被傳送至預覽佇列。

預覽佇列中的歸檔與分段備份請求，均為無法立即滿足的請求。根據預設值，系統將按先進先出 (FIFO) 順序來滿足預覽請求。

您可以將不同的優先權指派給預覽請求，亦可在預覽程序檔中輸入指令以覆寫 FIFO 預設值。所輸入的指令將寫入 `/etc/opt/SUNwsamfs/preview.cmd`。若需更多關於此檔案和歸檔與分段備份設定優先權的資訊，請參閱第 122 頁「設定預覽請求的優先權」。

歸檔器範例

表 3-24 顯示本節中使用的所有範例之目錄結構。

表 3-24 目錄結構範例

最上層目錄	第一層子目錄	第二層子目錄	第三層子目錄
/sam	/projs	/proj_1	/katie
/sam	/projs	/proj_1	/sara
/sam	/projs	/proj_1	/wendy
/sam	/projs	/proj_2	/joe
/sam	/projs	/proj_2	/katie
/sam	/users	/bob	
/sam	/users	/joe	
/sam	/users	/katie	
/sam	/users	/sara	
/sam	/users	/wendy	
/sam	/data		
/sam	/tmp		

範例 1

此範例將展現未使用 `archiver.cmd` 檔案時的歸檔器動作。在此範例中，Sun StorEdge SAM-FS 環境包括一個檔案系統、一個有兩個磁碟機的光學自動程式庫及六個卡匣。

程式碼範例 3-47 顯示由 `archiver(1M) -lv` 指令產生的輸出。它顯示歸檔器選定的預設媒體是 `mo` 類型。只有 `mo` 媒體可用。

程式碼範例 3-47 `archiver(1M) -lv` 輸出範例 (第一部分)

```
# archiver -lv
Notify file: /etc/opt/SUNWsamfs/scripts/archiver.sh

Archive media:
media:lt archmax: 512.0M Volume overflow not selected
media:mo archmax: 4.8M Volume overflow not selected
```

程式碼範例 3-48 顯示的輸出表示歸檔器使用兩個磁碟機。它會列出 12 個磁碟區、儲存容量及可用空間。

備註 – archiver(1M) -lv 指令僅於有空間之 VSN 顯示。

程式碼範例 3-48 archiver(1M) -lv 輸出範例 (第二部分)

```
Archive libraries:
Device:hp30 drives_available:2 archive_drives:2
Catalog:
mo.optic00          capacity: 1.2G space: 939.7M -il-o-----
mo.optic01          capacity: 1.2G space: 934.2M -il-o-----
mo.optic02          capacity: 1.2G space: 781.7M -il-o-----
mo.optic03          capacity: 1.2G space: 1.1G   -il-o-----
mo.optic10          capacity: 1.2G space: 85.5M  -il-o-----
mo.optic11          capacity: 1.2G space: 0      -il-o-----
mo.optic12          capacity: 1.2G space: 618.9k -il-o-----
mo.optic13          capacity: 1.2G space: 981.3M -il-o-----
mo.optic20          capacity: 1.2G space: 1.1G   -il-o-----
mo.optic21          capacity: 1.2G space: 1.1G   -il-o-----
mo.optic22          capacity: 1.2G space: 244.9k -il-o-----
mo.optic23          capacity: 1.2G space: 1.1G   -il-o-----
```

程式碼範例 3-49 顯示包括在歸檔組 samfs 中的中介資料與資料檔。在檔案的歸檔時間達到預設的四分鐘 (240 秒) 時，歸檔器將為這些檔案製作一個備份。

程式碼範例 3-49 archiver(1M) -lv 輸出範例 (第三部分)

```
Archive file selections:
Filesystem samfs  Logfile:
samfs Metadata
  copy:1 arch_age:240
samfs1 path:..
  copy:1 arch_age:240
```

程式碼範例 3-50 顯示歸檔組中以指定順序歸檔至磁碟區的檔案。

程式碼範例 3-50 archiver(1M) -lv 輸出範例 (第四部分)

```
Archive sets:
allsets
samfs.1
  media: mo (by default)
Volumes:
  optic00
  optic01
  optic02
```

程式碼範例 3-50 archiver(1M) -lv 輸出範例 (第四部分) (續)

```
    optic03
    optic10
    optic12
    optic13
    optic20
    optic21
    optic22
    optic23
Total space available:    8.1G
```

範例 2

此範例顯示如何將資料檔分成兩個從中介資料中獨立的歸檔組。除了中的光學自動程式庫之外，還有一個手動掛載的 DLT 磁帶機第 83 頁「範例 2」。大檔案歸檔至磁帶，而小檔案將歸檔至光碟匣。

程式碼範例 3-51 顯示 archiver.cmd 檔案的內容。

程式碼範例 3-51 顯示 archiver.cmd 檔案的 archiver(1M) -lv 輸出 (第一部分)

```
# archiver -lv -c example2.cmd
Reading archiver command file "example2.cmd"
1: # Example 2 archiver command file
2: # Simple selections based on size
3:
4: logfile = /var/opt/SUNWsamfs/archiver/log
5: interval = 5m
6:
7: # File selections.
8: big . -minsize 500k
9: all .
10:    1 30s
11:
12: vsns
13: samfs.1 mo .*[0-2]          # Metadata to optic00 - optic02
14: all.1 mo .*[3-9] .*[1-2][0-9] # All others for files
15: big.1 lt .*
16: endvsns
```

程式碼範例 3-52 顯示要使用的媒體和磁碟機，而非 DLT 增加部分及其預設值。

程式碼範例 3-52 顯示媒體和磁碟機的 archiver(1M) -lv 輸出 (第二部分)

```
Notify file: /etc/opt/SUNWsamfs/scripts/archiver.sh
Archive media:
media:lt archmax: 512.0M Volume overflow not selected
```

程式碼範例 3-52 顯示媒體和磁碟機的 archiver(1M) -lv 輸出 (第二部分) (續)

```
media:mo archmax: 4.8M Volume overflow not selected
Archive libraries:
Device:hp30 drives_available:0 archive_drives:0
Catalog:
mo.optic00 capacity: 1.2G space: 939.7M -il-o-----
mo.optic01 capacity: 1.2G space: 934.2M -il-o-----
mo.optic02 capacity: 1.2G space: 781.7M -il-o-----
mo.optic03 capacity: 1.2G space: 1.1G -il-o-----
mo.optic04 capacity: 1.2G space: 983.2M -il-o-----
mo.optic10 capacity: 1.2G space: 85.5M -il-o-----
mo.optic11 capacity: 1.2G space: 0 -il-o-----
mo.optic12 capacity: 1.2G space: 618.9k -il-o-----
mo.optic13 capacity: 1.2G space: 981.3M -il-o-----
mo.optic20 capacity: 1.2G space: 1.1G -il-o-----
mo.optic21 capacity: 1.2G space: 1.1G -il-o-----
mo.optic22 capacity: 1.2G space: 244.9k -il-o-----
mo.optic23 capacity: 1.2G space: 1.1G -il-o-----
Device:lt40 drives_available:0 archive_drives:0
Catalog:
lt.TAPE01 capacity: 9.5G space: 8.5G -il-o-----
lt.TAPE02 capacity: 9.5G space: 6.2G -il-o-----
lt.TAPE03 capacity: 9.5G space: 3.6G -il-o-----
lt.TAPE04 capacity: 9.5G space: 8.5G -il-o-----
lt.TAPE05 capacity: 9.5G space: 8.5G -il-o-----
lt.TAPE06 capacity: 9.5G space: 7.4G -il-o-----
```

備註 — archiver(1M) -lv 指令僅於有空間之 VSN 顯示。

程式碼範例 3-53 顯示檔案系統的組織。大於 512000 位元組 (500 KB) 的檔案在四分分鐘後歸檔；所有其他檔案在 30 秒後歸檔。

程式碼範例 3-53 顯示檔案系統組織的 archiver(1M) -lv 輸出 (第三部分)

```
Archive file selections:
Filesystem samfs Logfile: /var/opt/SUNWsamfs/archiver/log
samfs Metadata
copy:1 arch_age:240
big path:. minsize:502.0k
copy:1 arch_age:240
all path:.
copy:1 arch_age:30
```

程式碼範例 3-54 顯示以下輸出中的可移除的媒體之歸檔組部分。

程式碼範例 3-54 顯示歸檔組和可移除的媒體的 archiver(1M) -lv 輸出 (第四部分)

```
Archive sets:
allsets
all.1
  media: mo
Volumes:
  optic03
  optic04
  optic10
  optic12
  optic13
  optic20
  optic21
  optic22
  optic23
Total space available: 6.3G
big.1
  media: lt
Volumes:
  TAPE01
  TAPE02
  TAPE03
  TAPE04
  TAPE05
  TAPE06
Total space available: 42.8G
samfs.1
  media: mo
Volumes:
  optic00
  optic01
  optic02
Total space available: 2.6G
```

範例 3

在此範例中，使用者檔案與專案資料檔將歸檔至各種媒體。data 目錄中的檔案將被按大小與光學及磁帶媒體隔離。指定給群組 ID pict 的檔案將被指定給其他組的磁碟區。tmp 與 users/bob 目錄中的檔案將不歸檔。歸檔將以 15 分鐘的間隔執行，並保留歸檔記錄。

程式碼範例 3-55 顯示此範例。

程式碼範例 3-55 archiver(1M) -lv -c 指令輸出

```
# archiver -lv -c example3.cmd
Reading archiver command file "example3.cmd"
1: # Example 3 archiver command file
2: # Segregation of users and data
3:
4: interval = 30s
5: logfile = /var/opt/SUNWsamfs/archiver/log
6:
7: no_archive tmp
8:
9: fs = samfs
10: no_archive users/bob
11: prod_big data -minsize 50k
12:   1 1m 30d
13:   2 3m
14: prod data
15:   1 1m
16: proj_1 projs/proj_1
17:   1 1m
18:   2 1m
19: joe . -user joe
20:   1 1m
21:   2 1m
22: pict . -group pict
23:   1 1m
24:   2 1m
25:
26: params
27: prod_big.1 -drives 2
28: prod_big.2 -drives 2
29: endparams
30:
31: vsns
32: samfs.1 mo optic0[0-1]$
33: joe.1 mo optic01$
34: pict.1 mo optic02$
35: pict.2 mo optic03$
36: proj_1.1 mo optic1[0-1]$
```

程式碼範例 3-55 archiver(1M) -lv -c 指令輸出 (續)

```
37: proj_1.2 mo optic1[2-3]$
38: prod.1 mo optic2.$
39: joe.2 lt 0[1-2]$
40: prod_big.1 lt 0[3-4]$
41: prod_big.2 lt 0[5-6]$
42: endvsns

Notify file: /etc/opt/SUNWsamfs/scripts/archiver.sh

Archive media:
media:lt archmax: 512.0M Volume overflow not selected
media:mo archmax: 4.8M Volume overflow not selected

Archive libraries:
Device:hp30 drives_available:0 archive_drives:0
Catalog:
mo.optic00      capacity: 1.2G space: 939.7M -il-o-----
mo.optic01      capacity: 1.2G space: 934.2M -il-o-----
mo.optic02      capacity: 1.2G space: 781.7M -il-o-----
mo.optic03      capacity: 1.2G space: 1.1G -il-o-----
mo.optic04      capacity: 1.2G space: 983.2M -il-o-----
mo.optic10      capacity: 1.2G space: 85.5M -il-o-----
mo.optic11      capacity: 1.2G space: 0 -il-o-----
mo.optic12      capacity: 1.2G space: 618.9k -il-o-----
mo.optic13      capacity: 1.2G space: 981.3M -il-o-----
mo.optic20      capacity: 1.2G space: 1.1G -il-o-----
mo.optic21      capacity: 1.2G space: 1.1G -il-o-----
mo.optic22      capacity: 1.2G space: 244.9k -il-o-----
mo.optic23      capacity: 1.2G space: 1.1G -il-o-----

Device:lt40 drives_available:0 archive_drives:0
Catalog:
lt.TAPE01      capacity: 9.5G space: 8.5G -il-o-----
lt.TAPE02      capacity: 9.5G space: 6.2G -il-o-----
lt.TAPE03      capacity: 9.5G space: 3.6G -il-o-----
lt.TAPE04      capacity: 9.5G space: 8.5G -il-o-----
lt.TAPE05      capacity: 9.5G space: 8.5G -il-o-----
lt.TAPE06      capacity: 9.5G space: 7.4G -il-o-----

Archive file selections:
Filesystem samfs Logfile: /var/opt/SUNWsamfs/archiver/log
samfs Metadata
copy:1 arch_age:240
no_archive Noarchive path:users/bob
prod_big path:data minsize:50.2k
copy:1 arch_age:60 unarch_age:2592000
copy:2 arch_age:180
```

程式碼範例 3-55 archiver(1M) -lv -c 指令輸出 (續)

```
prod path:data
  copy:1 arch_age:60
proj_1 path:projs/proj_1
  copy:1 arch_age:60
  copy:2 arch_age:60
joe path:. uid:10006
  copy:1 arch_age:60
  copy:2 arch_age:60
pict path:. gid:8005
  copy:1 arch_age:60
  copy:2 arch_age:60
no_archive Noarchive path:tmp
samfs path:.
  copy:1 arch_age:240

Archive sets:
allsets

joe.1
media: mo
Volumes:
  optic01
Total space available: 934.2M

joe.2
media: lt
Volumes:
  TAPE01
  TAPE02
Total space available: 14.7G

pict.1
media: mo
Volumes:
  optic02
Total space available: 781.7M

pict.2
media: mo
Volumes:
  optic03
Total space available: 1.1G

prod.1
media: mo
Volumes:
  optic20
```

程式碼範例 3-55 archiver(1M) -lv -c 指令輸出 (續)

```
    optic21
    optic22
    optic23
Total space available:   3.3G

prod_big.1
media: lt drives:2
Volumes:
    TAPE03
    TAPE04
Total space available:  12.1G

prod_big.2
media: lt drives:2
Volumes:
    TAPE05
    TAPE06
Total space available:  16.0G

proj_1.1
media: mo
Volumes:
    optic10
Total space available:  85.5M

proj_1.2
media: mo
Volumes:
    optic12
    optic13
Total space available: 981.9M

samfs.1
media: mo
Volumes:
    optic00
    optic01
Total space available:   1.8G
```

範例 4

在此範例中，使用者檔案與專案資料檔將歸檔至光學媒體。請注意，程式碼範例 3-56 並不使用在表 3-24 中分段備份的目錄結構。

將定義四個 VSN 池；三個池用於使用者、資料及專案，一個是 `scratch pool` (臨時池)。`proj_pool` 在超出媒體時，它將依賴 `scratch_pool` 預留磁碟區。此範例顯示如何基於歸檔組元件、所有者元件及檔案系統元件為每個歸檔組預留磁碟區。歸檔將以 10 分鐘的間隔執行，並保留歸檔記錄。

程式碼範例 3-56 顯示 `archiver.cmd` 檔案和歸檔器輸出。

程式碼範例 3-56 `archiver.cmd` 檔案和歸檔器輸出

```
Reading archiver command file "example4.cmd"
1: # Example 4 archiver command file
2: # Using 4 VSN pools
3:
4: interval = 30s
5: logfile = /var/opt/SUNWsamfs/archiver/log
6:
7: fs = samfs
8: users users
9:     1 10m
10:
11: data data
12:     1 10m
13:
14: proj projects
15:     1 10m
16:
17: params
18: users.1 -reserve user
19: data.1 -reserve group
20: proj.1 -reserve dir -reserve fs
21: endparams
22:
23: vsnpools
24: users_pool mo optic0[1-3]$
25: data_pool mo optic1[0-1]$
26: proj_pool mo optic1[2-3]$
27: scratch_pool mo optic2.$
28: endvsnpools
29:
30: vsn
31: samfs.1 mo optic00
32: users.1 mo -pool users_pool -pool scratch_pool
33: data.1 mo -pool data_pool -pool scratch_pool
34: proj.1 mo -pool proj_pool -pool scratch_pool
```

程式碼範例 3-56 archiver.cmd 檔案和歸檔器輸出 (續)

```
Reading archiver command file "example4.cmd"
35: endvsns

Notify file: /etc/opt/SUNWsamfs/scripts/archiver.sh

Archive media:
media:mo archmax: 4.8M Volume overflow not selected

Archive libraries:
Device:hp30 drives_available:0 archive_drives:0
Catalog:
mo.optic00 capacity: 1.2G space: 939.7M -il-o-----
mo.optic01 capacity: 1.2G space: 934.2M -il-o-----
mo.optic02 capacity: 1.2G space: 781.7M -il-o-----
mo.optic03 capacity: 1.2G space: 1.1G -il-o-----
mo.optic04 capacity: 1.2G space: 983.2M -il-o-----
mo.optic10 capacity: 1.2G space: 85.5M -il-o-----
mo.optic11 capacity: 1.2G space: 0 -il-o-----
mo.optic12 capacity: 1.2G space: 618.9k -il-o-----
mo.optic13 capacity: 1.2G space: 981.3M -il-o-----
mo.optic20 capacity: 1.2G space: 1.1G -il-o-----
mo.optic21 capacity: 1.2G space: 1.1G -il-o-----
mo.optic22 capacity: 1.2G space: 244.9k -il-o-----
mo.optic23 capacity: 1.2G space: 1.1G -il-o-----

Archive file selections:
Filesystem samfs Logfile: /var/opt/SUNWsamfs/archiver/log
samfs Metadata
copy:1 arch_age:240
users path:users
copy:1 arch_age:600
data path:data
copy:1 arch_age:600
proj path:projects
copy:1 arch_age:600
samfs path:.
copy:1 arch_age:240

VSN pools:
data_pool media: mo Volumes:
optic10
Total space available: 85.5M

proj_pool media: mo Volumes:
optic12
optic13
```

程式碼範例 3-56 archiver.cmd 檔案和歸檔器輸出 (續)

```
Reading archiver command file "example4.cmd"
  Total space available: 981.9M

scratch_pool media: mo Volumes:
  optic20
  optic21
  optic22
  optic23
  Total space available:   3.3G

users_pool media: mo Volumes:
  optic01
  optic02
  optic03
  Total space available:   2.7G

Archive sets:
allsets

data.1
  reserve:/group/
  media: mo
  Volumes:
    optic10
    optic20
    optic21
    optic22
    optic23
  Total space available:   3.4G

proj.1
  reserve:/dir/fs
  media: mo
  Volumes:
    optic12
    optic13
    optic20
    optic21
    optic22
    optic23
  Total space available:   4.2G

samfs.1
  media: mo
  Volumes:
    optic00
  Total space available: 939.7M
```

程式碼範例 3-56 archiver.cmd 檔案和歸檔器輸出 (續)

```
Reading archiver command file "example4.cmd"
```

```
users.1
```

```
  reserve:/user/
```

```
media: mo
```

```
Volumes:
```

```
  optic01
```

```
  optic02
```

```
  optic03
```

```
  optic20
```

```
  optic21
```

```
  optic22
```

```
  optic23
```

```
Total space available: 6.0G
```

第4章

釋放

釋放是指釋放器透過識別歸檔檔案並釋放其磁碟快取記憶體記憶體備份，來取得可用磁碟快取記憶體空間的程序。這樣即可騰出空間，用於從歸檔媒體建立或分段備份其他檔案。釋放器僅可以釋放歸檔檔案。釋放檔案將會使檔案在磁碟快取記憶體中不具有任何資料。

當到達特定站點磁碟臨界值時，Sun StorEdge SAM-FS 軟體會自動呼叫釋放器程序。或者，您可以使用 `release(1)` 指令立即釋放檔案的磁碟空間，或設定檔案的釋放參數。若需更多有關釋放器程序的資訊，請參閱 `sam-releaser(1M)` 線上手冊。

釋放器所包含的功能，可讓您指定在歸檔後立即釋放檔案、永不釋放檔案或部份釋放檔案。部分釋放功能非常有用，因為某些應用程式 (例如：`filemgr(1)`) 僅讀取檔案的開頭。透過使用部分釋放功能，檔案的一部分將留在磁碟快取記憶體中，而其餘部分則會被釋放。在讀取仍留在磁碟快取記憶體中的第一部分檔案時，無需觸發將檔案其餘部分從歸檔媒體分段備份回磁碟快取記憶體的程序。本章將說明上述功能及其他功能。

本章包含下列主題：

- 第 99 頁 「釋放程序簡介」
- 第 102 頁 「關於部分釋放和部分分段備份」
- 第 105 頁 「關於 `releaser.cmd` 檔案」
- 第 112 頁 「規畫釋放器作業」
- 第 114 頁 「手動執行釋放器」

釋放程序簡介

在檔案系統使用率超過其配置的高值參數時，File System Manager 軟體將啟動釋放器。首先，釋放器會讀取 `releaser.cmd` 檔案並收集控制釋放程序的指令。然後，它會掃描檔案系統並收集有關每個檔案的資訊。最後，釋放器會在掃描整個檔案系統之後，按優先權順序開始釋放檔案。

只要檔案系統仍高於配置的低值參數，釋放器就會繼續釋放檔案。釋放器通常會釋放足夠的空間，以讓檔案系統降至低值參數以下。如果釋放器找不到要釋放的任何檔案，則會結束。如果以後有更多檔案可以釋放，釋放器就會執行。在高於高值參數時，檔案系統會每分鐘啟動釋放器一次。

高值和低值參數是以 `high=percent` 和 `low=percent` 檔案系統掛載選項所設定的。若需更多有關掛載選項的資訊，請參閱 `mount_samfs(1M)` 線上手冊。

作業理論

一個檔案系統可以包含數千個檔案。由於僅釋放幾個大型檔案即可使檔案系統恢復至其低值參數，因此追蹤所有檔案的釋放優先權會成一種浪費。但是，釋放器必須檢查每個檔案的優先權，否則可能會遺漏最佳釋放候選者。釋放器透過僅識別前 10,000 個候選者來處理此情況。

識別前 10,000 個候選者後，如果後面所有候選者的優先權均低於前 10,000 個候選者中的最低優先權，釋放器將放棄後面的所有候選者。

釋放器在確定前 10,000 候選者的優先權後，會選擇優先權最高的檔案進行釋放。釋放每個檔案之後，釋放器會檢查檔案系統快取記憶體使用率是否低於低值參數。如果低於低值參數，釋放器會停止釋放檔案。如果高於低值參數，釋放器會按優先權順序繼續釋放檔案。

如果在釋放器釋放所有 10,000 個候選者後，檔案系統仍高於低值參數，釋放器會重新開始識別 10,000 個新的候選者。

如果找不到任何可用的候選者，釋放器就會結束。這種情況有可能會發生，例如在檔案沒有歸檔備份時。若是發生，Sun StorEdge SAM-FS 軟體會在一分鐘過後開始再次釋放。

定義

本節說明本章中使用的專有名詞。

時間

時間概念指的是從發生特定事件到現在所經歷的時間。檔案的 `inode` 會追蹤釋放器所用的以下各種時間：

- 常駐變更時間
- 資料修改時間
- 資料存取時間

您可以使用 `sls(1)` 指令的 `-D` 選項來檢視這些時間。每個時間均有相應的經過時間。例如：如果現在是 10:15 a.m.，檔案的修改時間是 10:10 a.m.，則資料修改時間是 5 分鐘。若需得更多有關 `sls(1)` 指令的資訊，請參閱 `sls(1)` 線上手冊。

候選者

候選者指的是符合釋放條件的檔案。檔案不能成為候選者的原因如下：

- 檔案已離線。
- 檔案尚未歸檔。
- `archiver.cmd` 程序檔案為該檔案指定了 `-norelease` 屬性，且尚未製作要求的備份。
- 檔案已標記為「損壞」。
- 檔案不是一般檔案，而是目錄、區塊、特殊字元檔案或管線。
- 歸檔器正在分段備份檔案以製作額外的備份。在歸檔器分段備份檔案之後，該檔案即可釋放。
- 檔案時間為負數。時鐘設定不正確的 NFS 用戶端通常會發生此情況。
- 檔案被標記為永不釋放。您可以使用 `release(1) -n` 指令來進行指定。
- 檔案在過去的某個時間進行分段備份，且該時間短於最短常駐時間設定。如需更多資訊，請參閱第 108 頁「指定最短常駐時間：`min_residence_age`」。
- 已使用 `release(1)` 指令的 `-p` 選項，將檔案標記為部分釋放，並且已經部分釋放該檔案。
- 檔案太小。

優先權

優先權是一個數值，它表示以使用者提供並套用於該候選者數字屬性的權數為基礎的候選檔案等級。總優先權為以下兩種優先權的總合：時間優先權和大小優先權。

系統將先釋放優先權數值較大的候選檔案，然後再釋放優先權數值較小的候選檔案。

權數

權數為偏壓優先權計算以包括感興趣的檔案屬性，並排除不感興趣的檔案屬性之數值。例如：如果大小權數設定為零，檔案的大小屬性將被排除在優先權計算之外。權線是指 0.0 至 1.0 的浮點值。

部分釋放

透過指定將檔案的開頭部分留在磁碟快取記憶體中，而釋放其餘部分，即可部分釋放該檔案。在使用只讀取檔案開頭的公用程式 (`filemgr(1)`) 時，部分釋放功能非常有用。

關於部分釋放和部分分段備份

釋放與分段備份均為補充程序。檔案在歸檔後可以從線上磁碟快取記憶體中完全釋放；站點亦可指定將檔案的開頭部分 (*stub*) 留在磁碟快取記憶體中，而釋放其餘部分。部分釋放檔案的功能，可讓您在沒有分段備份檔案的情況下立即存取檔案 *stub* 中的資料。

在掛載檔案系統時，系統管理員可以指定保持線上狀態的 *stub* 之預設部分釋放大小與最大值。系統管理員可於 `mount(1M)` 指令或在 File System Manager 軟體上設定。請參閱 File System Manager 線上手冊以取得更多資訊。`mount(1M)` 指令選項如下：

- 指定 `-o partial=n` 選項，以設定要維持線上的檔案 *stub* 之預設大小 (*n*)。
`-o partial=n` 設定必須小於或等於 `-o maxpartial=n` 設定。可能的最小設定是 `-o partial=8 KB`。預設值是 `-o partial=16 KB`。
- 指定 `-o maxpartial=n` 選項，以設定要維持線上的檔案 *stub* 之最大大小 (*n*)。要限制可保持線上狀態的檔案 *stub* 之大小，請使用 `-o maxpartial=n` 選項，並指定大小等於可保持線上狀態的最大 *stub*。要停用部分釋放功能，請指定 `-o maxpartial=0`。

使用者可以在 `release(1)` 指令中指定 `-p` 選項或在 `sam_release(3)` 程式庫常式中指定 `p` 選項，以指定檔案的預設 *stub* 大小。要為不同類型的檔案或不同應用程式指定不同大小的檔案 *stub*，使用者可以在 `release(1)` 指令中指定 `-s` 選項或在 `sam_release(3)` 程式庫常式中指定 `s` 選項。`-s` 與 `s` 值必須小於掛載檔案系統時在 `mount(1M)` 指令中使用的 `-o maxpartial` 值。

另一個掛載選項 `-o partial_stage=n` 可讓系統管理員建立在分段備份檔案其餘部分之前，必須讀取的部分釋放 *stub* 數目。換言之，超過 `-o partial_stage=n` 大小指定的讀取將會初始化檔案分段備份。

依據預設，`-o partial_stage=n` 選項將設定為部分釋放 *stub* 的大小。您可以配置此值，但它會給檔案分段備份帶來下列影響：

- 如果 `-o partial_stage=n` 選項設定為部分釋放 *stub* 的大小，在應用程式達到部分釋放 *stub* 的結尾之前，預設運作方式會防止分段備份檔案。在達到 *stub* 結尾之前持續等待，將會導致在存取檔案其餘部分時出現延遲。
- 如果 `-o partial_stage=n` 選項設定為小於部分釋放 *stub* 的值，則會發生下列情況。在應用程式交錯 `-o partial_stage=n` 選項設定的臨界值後，檔案的其餘部分就會被分段備份。這樣就降低了在存取檔案資料其餘部分時發生延遲的可能性。

範例。假設下列選項生效：

- -o partial_stage=16 (即 16 KB)
- -o partial=2097152 (即 2 GB)
- -o maxpartial=2097152 (即 2 GB)

目前使用的是 filemgr(1) 程式，它讀取檔案的前 8 KB。該檔案沒有分段備份。即時影像播放程式讀取相同的檔案，在讀取檔案的前 16 KB 之後，該檔案被分段備份。在歸檔磁帶掛載並定位時，該應用程式將繼續讀取 2 GB 的磁碟資料。在即時影像播放程式讀取 2 GB 的檔案資料之後，它會在完成分段備份作業後立即進行讀取。應用程式無需等待，因為磁帶掛載與定位工作在應用程式讀取部分檔案資料時執行。

有多個指令行選項會影響是否可將檔案標記為部分釋放。某些選項由系統管理員啟動，其他選項則可由個別使用者啟動。下列章節說明可由各種使用者設定的釋放特性。

系統管理員選項摘要

系統管理員可在掛載檔案系統時變更部份釋放的最高值與預設值。表 4-1 中的 mount(1M) 選項會影響部分釋放。若需更多有關 mount(1) 指令的資訊，請參閱 mount_samfs(1M) 線上手冊。

表 4-1 部分釋放的掛載選項

mount(1M) 選項	作用
-o maxpartial= <i>n</i>	<p>在檔案標記為部分釋放時，決定在線上磁碟快取記憶體中可保留的最大空間容量 (KB)。最大值是 2,097,152 KB，即 2 GB。最小值是 0，它會防止部分釋放任何檔案。</p> <p>如果指定 -o maxpartial=0，部分釋放功能將停用，被釋放的檔案將會完全釋放，該檔案的任何部分均不會保留在磁碟快取記憶體中。在檔案系統掛載後，使用者將無法覆寫在此選項中指定的值。</p> <p>依據預設，<i>n</i> 引數會設為 16。此設定可讓使用者以磁碟上保留的最大數量 (16 KB) 來標記要部分釋放的檔案。</p>

表 4-1 部分釋放的掛載選項 (續)

<code>-o partial=<i>n</i></code>	<p>設定在使用者使用 <code>release(1)</code> 指令的 <code>-p</code> 選項將檔案標記為部分釋放時，在磁碟快取記憶體中保留的預設空間容量 (KB)。 <i>n</i> 引數必須至少為 8，最大可等於為 <code>-o maxpartial=<i>n</i></code> 選項指定的值。</p> <p>由於某些應用程式無需存取整個檔案即可完成工作，因此該選項可用於確保應用程式具有必需的檔案開頭部分。此外，使用此選項亦可防止對檔案進行不必要的分段備份。</p> <p>預設值是 <code>-o partial=16</code>。</p>
<code>-o partial_stage=<i>n</i></code>	<p>指定在存取部分釋放的檔案時，必須先讀取 <i>n</i> 位元組的檔案，然後再從歸檔媒體中分段備份整個檔案。此值通常設定為低於 <code>-o partial</code> 設定的容量。關於 <i>n</i>，請指定從 0 至 <code>-o maxpartial</code> 規定的整數值。依據預設，它將設定為 16 或為 <code>-o partial</code> 選項指定的任何值。</p>
<code>-o stage_n_window=<i>n</i></code>	<p>將在某段任意時間內要分段備份的資料容量指定為 <i>n</i>。關於 <i>n</i>，請指定從 64 至 2,048,000 的整數值。預設值為 256 KB。此選項僅適用於已設定 <code>stage -n</code> 屬性的檔案。</p>

使用者選項摘要

系統管理員可以設定在釋放檔案後，可保留在磁碟快取記憶體中的檔案 `stub` 大小之最大值與預設值。系統管理員亦可決定是否為特定檔案系統啟動部分釋放功能。

但是，使用者可以透過使用 `release(1)` 指令與 `sam_release(3)` 程式庫常式，來設定其他釋放屬性，並可指定要標記為部分釋放的檔案。決定部份釋放屬性的指令與程式庫選項，如表 4-2 中所示。若需更多有關 `release(1)` 指令的資訊，請參閱 `release(1)` 線上手冊。若需更多有關 `sam_release(3)` 程式庫常式的資訊，請參閱 `sam_release(3)` 線上手冊。

表 4-2 使用者釋放選項

選項	作用
<code>release(1)</code> 指令和 <code>-p</code> 選項 或 <code>sam_release(3)</code> 程式庫常式和 <code>p</code> 選項	<p><code>-p</code> 與 <code>p</code> 選項可將指定的檔案標記為部分釋放。如果使用了這兩個選項，檔案被釋放後保留在線上磁碟快取記憶體中的檔案容量，將取決於在掛載該檔案所在的檔案系統時設定的 <code>-o partial=<i>n</i></code> 選項值。這兩個選項不能用於指定保持線上狀態的位元組數目。</p>
<code>release(1)</code> 指令和 <code>-s partial_size</code> 選項 或 <code>sam_release(3)</code> 程式庫常式和 <code>s</code> 選項	<p><code>-s</code> 與 <code>s</code> 選項可將指定的檔案標記為部分釋放，並可指定保留在線上磁碟快取記憶體中的檔案容量。<code>-s</code> 或 <code>s</code> 選項的引數將指定保持線上狀態的容量 (KB)。使用者為保持線上狀態的檔案指定的容量，不能大於在掛載檔案系統時為 <code>-o maxpartial=<i>n</i></code> 值指定的容量。如果使用者指定的值大於檔案系統的值，系統將使用檔案系統的值，而忽略使用者指定的值。</p>

關於 `releaser.cmd` 檔案

`/etc/opt/SUNWsamfs/releaser.cmd` 檔案由指定站點特定釋放動作的指令行組成。`releaser.cmd` 檔案可以包含設定釋放優先權、指定記錄檔及其他動作的指令。

下列章節說明 `releaser.cmd` 指令：

- 第 105 頁 「指定與時間和大小相關的釋放優先權指令」
- 第 107 頁 「為個別檔案系統指定指令：`fs`」
- 第 108 頁 「指定除錯指令：`no_release` 和 `display_all_candidates`」
- 第 108 頁 「指定最短常駐時間：`min_residence_age`」
- 第 109 頁 「指定記錄檔：`logfile`」
- 第 110 頁 「限制釋放重新歸檔的檔案：`rearch_no_release`」
- 第 111 頁 「調整釋放器候選者清單的大小：`list_size`」

若需更多有關這些指令的資訊，請參閱 `releaser.cmd(4)` 線上手冊。某些全域釋放指令可使用 File System Manager 軟體配置。請參閱 File System Manager 線上手冊以取得更多資訊。

指定與時間和大小相關的釋放優先權指令

系統將使用由 `releaser.cmd` 檔案中定義的指令決定的優先權順序，從檔案系統中釋放檔案。在此過程中，系統將同時考量檔案時間與大小。依據預設，站點將先釋放最大且最早的檔案，而將最小且最新的檔案留在磁碟上。下列章節說明釋放器在決定檔案系統中的檔案釋放優先權時，如何考量檔案的時間與大小。

若需更多有關釋放器指令的資訊，請參閱 `releaser.cmd(4)` 線上手冊。

檔案時間

在決定檔案釋放優先權中與時間相關的元件時，釋放器將考量下列可能的時間：

- 自上次存取之後的時間
- 自上次修改之後的時間
- 在磁碟快取記憶體中變更常駐之後的時間

在某些情況下，您可能要優先於檔案的修改時間來考量檔案的存取時間。在其他情況下，您可能只想考量從最晚存取時間、修改時間及常駐變更時間中衍生的簡單時間。

依據預設，檔案時間是指檔案下列三個時間中的較晚時間：

- 檔案存取時間

- 檔案修改時間
- 檔案常駐時間

您可以使用指令來指定在計算檔案的釋放優先權時，使用加權後的時間優先權。

程式碼範例 4-1 顯示時間優先權指令的格式。

程式碼範例 4-1 時間優先權指令格式

```
weight_age = 浮點
weight_age_access = 浮點
weight_age_modification = 浮點
weight_age_residence = 浮點
```

- `weight_age` 指令可指定為檔案的預設時間 (檔案存取時間、修改時間或常駐時間中的較短時間) 提供一個加權係數。關於浮點，請在下列範圍內指定浮點數字：
 $0.0 \leq \text{浮點} \leq 1.0$ 。依據預設，`浮點 = 1.0`。

此指令不能與 `weight_age_residence`、`weight_age_modify` 或 `weight_age_access` 指令一起指定。

- `weight_age_residence`、`weight_age_modify` 及 `weight_age_access` 指令可指定檔案的時間決定於這三個可能時間的其中一個、兩個或三個之組合。關於 `float`，請在下列範圍內指定浮點數字： $0.0 \leq \text{float} \leq 1.0$ 。依據預設，`float = 1.0`。

這些指令不能與 `weight_age` 指令一起指定。

如果使用了 `weight_age_residence`、`weight_age_modify` 及 `weight_age_access` 指令，該檔案與時間相關的優先權將根據所有這三個時間的組合進行計算。首先，系統將為每個檔案的可能時間收集檔案時間資料，然後將檔案時間資料乘以在 `releaser.cmd` 檔案中指定的加權係數，最後將時間資料乘以每個加權係數所得的乘積相加，以計算檔案與時間相關的優先權 (例如：程式碼範例 4-2 中的方程式所示)：

程式碼範例 4-2 優先權計算

```
file access age * weight_age_access
+ file modification age * weight_age_modification
+ file residency age * weight_age_residence
-----
= age_related_priority
```

範例。程式碼範例 4-3 顯示 `releaser.cmd` 檔案中的文字行指定在計算檔案的釋放優先權時，僅考量檔案的常駐時間 (修改時間與存取時間均將被忽略)：

程式碼範例 4-3 `releaser.cmd` 檔案片段

```
weight_age_residence = 1.0
weight_age_modify = 0.0
weight_age_access = 0.0
```

計算出檔案與時間相關的優先權後，再將其乘以檔案與大小相關的優先權。與大小相關的優先權之計算方法如下列章節所述。

檔案大小

在決定檔案釋放優先權中與大小相關的元件時，釋放器將考量檔案的大小。將檔案大小(以 4 KB 區塊為單位) 乘以為 `weight_size` 指令指定的權數，即可獲得檔案釋放優先權中與大小相關的元件。

`weight_size` 指令的格式如下所示：

```
weight_size = 浮點
```

關於浮點，請在下列範圍內指定浮點數字： $0.0 \leq \text{浮點} \leq 1.0$ 。依據預設，`浮點 = 1.0`。

範例。程式碼範例 4-4 顯示一個 `releaser.cmd` 檔案，其指定在計算檔案的釋放優先權時，忽略 `samfs1` 和 `samfs2` 檔案系統中所有檔案的檔案大小。

程式碼範例 4-4 `releaser.cmd` 檔案

```
# releaser.cmd file
logfile = /var/adm/default.releaser.log
weight_size = 0.0
#
fs = samfs1
weight_age = 1.0
logfile = /var/adm/samfs1.releaser.log
#
fs = samfs2
weight_age_modify = 0.3
weight_age_access = 0.03
weight_age_residence = 1.0
logfile = /var/adm/samfs2.releaser.log
```

為個別檔案系統指定指令：`fs`

您可以在 `releaser.cmd` 檔案中使用 `fs = 系列組名稱` 指令，以表示 `fs =` 指令隨後的指令僅適用於該名稱的檔案系統。此指令具有以下格式：

```
fs = 系列組名稱
```

關於系列組名稱，請在 `mcf` 檔案中指定「系列組」名稱。

第一個 `fs` = 指令之前的所有指令均為全域指令，適用於所有檔案系統。`fs` = 指令之後的所有指令將覆寫全域指令。本章所述的指令可用作全域指令或某個檔案系統特定的指令。

`releaser.cmd(4)` 線上手冊包含 `fs` = 指令的範例。

指定除錯指令：`no_release` 和 `display_all_candidates`

在對釋放器進行微調或除錯時，`no_release` 與 `display_all_candidates` 指令非常有用。這些指令如下所示：

- `no_release` 指令可防止從線上磁碟快取記憶體中移除檔案。您可以在不釋放檔案的情況下，使用此指令來檢查 `releaser.cmd` 中的指令。此指令具有以下格式：

```
no_release
```

- `display_all_candidates` 指令可將所有釋放候選者的名稱寫入記錄檔。此指令具有以下格式：

```
display_all_candidates
```

此指令在除錯時非常有用，因為釋放器僅會將釋放候選者的名稱寫入記錄檔，而實際上並不會將其從檔案系統中釋放。

指定最短常駐時間：`min_residence_age`

`min_residence_age` 指令可讓您指定檔案在成為釋放候選者之前，必須在檔案系統中常駐的最短時間。此指令具有以下格式：

```
min_residence_age = 時間
```

關於時間，請指定以秒為單位的時間。預設時間是 600，即 10 分鐘。沒有實用的最短或最長時間設定。

指定記錄檔：logfile

如果在 `releaser.cmd` 檔案中指定了 `logfile` 指令，釋放器會將其作業記錄附加至指定的檔案名稱；如果不存在指定的檔案名稱，釋放器會建立該檔案名稱。此指令具有以下格式：

```
logfile = 檔案名稱
```

關於檔案名稱，請指定記錄檔的名稱。

程式碼範例 4-5 顯示一個記錄檔範例 (請注意，為了符合頁面大小，某些文字行已自動換行)。

程式碼範例 4-5 釋放器記錄檔範例

```
Releaser begins at Wed Apr 28 17:29:06 1999
inode pathname      /sam1/.inodes
low-water mark      24%
weight_size         1
weight_age          1
fs equipment ordinal 1
family-set name     samfs1
started by sam-amld? yes
release files?      yes
display_all_candidates? no
---before scan---
blocks_now_free:    3481504
lwm_blocks:         3729362
---scanning---
10501 (R:Wed Apr 21 18:47:50 CDT 1999) 10001 min, 500 blks /sam1/testdir0/filevp
10500 (R:Wed Apr 21 18:48:10 CDT 1999) 10000 min, 500 blks /sam1/testdir0/filewq
...
---after scan---
blocks_now_free:    3730736
lwm_blocks:         3729362
archnodrop: 0
already_offline: 0
bad_inode_number: 0
damaged: 0
extension_inode: 0
negative_age: 0
nodrop: 1
not_regular: 9
number_in_list: 675
released_files: 202
too_new_residence_time: 0
too_small: 2
```

程式碼範例 4-5 釋放器記錄檔範例 (續)

```
total_candidates: 675
total_inodes: 1376
wrong_inode_number: 0
zero_arch_status: 689
zero_inode_number: 0
zero_mode: 0
CPU time: 2 seconds.
Elapsed time: 10 seconds.
Releaser ends at Wed Apr 28 17:29:16 1999
```

releaser(1M) 線上手冊說明了記錄檔中包含的資訊。由於每次執行釋放器後記錄的大小就會增加，因此請務必考量減少記錄的大小或省略 logfile 關鍵字。

程式碼範例 4-6 顯示 `---after scan---` 此行下的統計之間存在的數學關係：

程式碼範例 4-6 程式碼範例 4-5 的 `---after scan---` 此行之後的數學關係

```
total_inodes = wrong_inode_number +
zero_inode_number +
zero_mode +
not_regular +
extension_inode +
zero_arch_status +
already_offline +
damaged +
nodrop +
archnodrop +
too_new_residence_time +
too_small +
negative_age +
total_candidates
released_files = total_candidates
```

限制釋放重新歸檔的檔案：rearch_no_release

依據預設，系統將釋放標記為重新歸檔的檔案。如果在 `releaser.cmd(4)` 檔案中指令了 `rearch_no_release` 指令，釋放器則不會釋放標記為要重新歸檔的檔案。此指令具有以下格式：

```
rearch_no_release
```

調整釋放器候選者清單的大小：list_size

您可以使用 `list_size` 指令來指定釋放器候選者的數量。如果您有注意到，釋放器在其釋放必要的檔案數量以達到低值參數之前進行多個檔案系統掃描，您可能想要考慮將此值提高到大於預設值 10,000 的標準。在含有許多小檔案的檔案系統中可能有這種現象。您可以從釋放器記錄檔取得有關釋放器作業的資訊。此指令具有以下格式：

```
list_size = 數目
```

關於數目，請指定一個整數，如 $10 \leq \text{數目} \leq 2,147,483,648$ 。

archiver.cmd 檔案在釋放中的角色

雖然 `archiver.cmd` 檔案中的大多數指令會影響歸檔，但歸檔組指定指令可讓您指定套用於歸檔組中所有檔案的釋放屬性。

歸檔組指定指令具有以下格式：

```
歸檔組名稱 路徑 [搜尋條件 ...] [檔案屬性]
```

表 4-3 顯示了與釋放有關的檔案屬性。

表 4-3 歸檔設定引數檔案屬性

指令	作用
<code>-release a</code>	指定在製作第一個歸檔備份後，必須釋放歸檔組中的檔案。若要為每個檔案製作多個歸檔備份，請勿使用此選項。在此情況下，系統將分段備份備份 1 以便製作備份 2。
<code>-release d</code>	重設回預設值。
<code>-release n</code>	指定永不釋放歸檔組中的檔案。
<code>-release p</code>	指定在歸檔後必須部分釋放歸檔組中的檔案。

若需更多有關上述及其他 `archiver.cmd` 指令的資訊，請參閱第 29 頁「歸檔」。

規畫釋放器作業

為站點確定快取記憶體中的檔案特性很有必要。如果您只要分段備份少許 KB，載入磁帶則過於繁複。因此您可能會想要偏壓您的系統使其在快取記憶體中保留小檔案。程式碼範例 4-7 顯示在 `releaser.cmd` 檔案中用來先釋放最大的檔案的指令。

程式碼範例 4-7 先釋放最大檔案的指令

```
weight_size = 1.0
weight_age = 0.0
```

您亦可能要在快取記憶體中保留最近修改的檔案，因為最近修改的檔案在不久之後可能會再次被修改。這樣，無需分段備份檔案即可進行修改。在這種情況下，請使用第二組時間權數。程式碼範例 4-8 顯示在 `releaser.cmd` 檔案中以嚴格的順序 (從最早修改到最近修改的檔案) 將檔案加權所使用的指令。

程式碼範例 4-8 先釋放最早修改的檔案的指令

```
weight_size = 0.0
weight_age_access = 0.0
weight_age_modify = 1.0
weight_age_residence = 0.0
```

但是，正如下列範例所展示，大多數情況並非如此簡單。

範例 1。 假設您要先釋放最大的檔案。目前有幾百個大小相同的小檔案，以及幾個大檔案。雖然這些小檔案的累計大小可能會超過一個最大檔案的大小，但釋放器最終仍會釋放所有大檔案。如果指定了 `weight_age = 0.0`，釋放器會按隨機順序釋放這些小檔案，因為這些小檔案的大小均相同，因此它們具有相同的釋放優先權。

在此情況下，您可以設定 `weight_age = 0.01` 以做為附加條件。釋放器將會先釋放大小相同的兩個檔案中時間較早的檔案。

範例 2。 本範例介紹了一種更佳方法，用於指定如何先釋放最大檔案。

設定 `weight_size = 1.0` 和 `weight_age = 0.01`。

這些指令將違反最大檔案優先的規則，它會將較小且存取時間較早 (而非較大且存取時間較晚) 的檔案視為較佳候選者。但您可以將 `weight_age` 設定為小於 `weight_size`，以便儘可能減少此影響。例如，在上述設定下，100 分鐘之前分段備份的 4 KB 檔案，與剛剛分段備份的 8 KB 檔案具有相同的釋放優先權。

釋放器會隨機選擇要釋放的檔案。如果它選擇 4 KB 的檔案，則會違反最大檔案優先規則。將 `weight_age` 設定為較小值 (如設定為 0.001) 可減少此影響。如果 4 KB 的檔案在 1,000 分鐘之前分段備份，它將與剛剛分段備份的 8 KB 檔案具有相同的優先權。

您可以使用 `no_release` 與 `display_all_candidates` 指令並手動執行釋放器，以取得按調整優先權權數時使用的優先權順序排列的候選者清單。

手動執行釋放器

有時，您可能要手動執行釋放器。要執行此作業，您必須知道檔案系統的掛載點以及釋放器應該嘗試達到的低值參數。

例如，若要在 /sam1 檔案系統達到 47 % 時釋放其中的檔案，請以 root 登入並鍵入以下指令：

```
# /opt/SUNWsamfs/sbin/sam-releaser /sam1 47 1.0
```

所有指令行選項會覆寫任何好 releaser.cmd 檔案中指定的選項。在釋放器執行時，其會將資訊寫入螢幕與釋放器記錄檔 (如果已在 releaser.cmd 檔案中指定)。若需更多相關資訊，請參閱 sam-releaser(1M) 線上手冊。

第5章

分段備份

分段備份是指將檔案資料從近線或離線儲存裝置複製回線上儲存裝置的程序。分段備份功能可讓您立即分段備份檔案、永不分段備份檔案、指定部分分段備份及其他分段備份動作。永不分段備份功能用途廣泛，例如從大檔案中隨機存取小記錄的應用程式即可使用此功能；在此功能啟動時，系統將直接從歸檔媒體存取資料，而不先在線上分段備份檔案。

本章描述 Sun StorEdge SAM-FS 檔案分段備份功能。包含有以下主題：

- 第 115 頁 「關於 stager.cmd 檔案」
- 第 122 頁 「設定預覽請求的優先權」
- 第 125 頁 「計算預覽請求的總優先權」
- 第 125 頁 「設定預覽請求優先機制」

關於 stager.cmd 檔案

您可以使用 `stager.cmd` 檔案來指定分段備份程式的運作方式。此檔案的完整路徑名稱是 `/etc/opt/SUNWsamfs/stager.cmd`。依據預設，分段備份程式將執行以下動作：

- 分段備份程式嘗試使用程式庫中的所有磁碟機分段備份檔案。
- 分段備份緩衝區大小由媒體類型決定，分段備份緩衝區未鎖定。
- 未寫入任何記錄檔。
- 最多可同時啟動 1000 個分段備份請求。

`stager.cmd` 檔案可讓您指定覆寫預設運作方式的指令。本章節的其餘部分將說明分段備份程式指令。若需關於分段備份程式指令的額外資訊，請參閱 `stager.cmd(4)` 線上手冊。

第 121 頁 「`stager.cmd` 檔案範例」顯示所有可能的指令均已設定後的已完成 `stager.cmd` 檔案。

程式碼範例 5-1 顯示本章範例所用的 mcf 檔案範例。

程式碼範例 5-1 本章範例所用的 mcf 檔案

```
#
# Sun StorEdge SAM-FS file system configuration example
#
# Equipment      Eq Eq Family Dev Additional
# Identifier     Or Tp Set   St Parameters
# -----
samfs1          60 ms samfs1
/dev/dsk/c1t1d0s6 61 md samfs1 on
/dev/dsk/c2t1d0s6 62 md samfs1 on
/dev/dsk/c3t1d0s6 63 md samfs1 on
/dev/dsk/c4t1d0s6 64 md samfs1 on
/dev/dsk/c5t1d0s6 65 md samfs1 on
#
samfs2          2 ms samfs2
/dev/dsk/c1t1d0s0 15 md samfs2 on
/dev/dsk/c1t0d0s1 16 md samfs2 on
#
/dev/samst/c0t2d0 20 od -      on
/dev/samst/c1t2u0 30 rb dog  on /var/opt/SUNWsamfs/catalog/dogcat
/dev/samst/c1t5u0 31 od dog  on
/dev/samst/c1t6u0 32 od dog  on
/dev/rmt/0cbtn   40 od -      on
/dev/samst/c1t3u1 50 rb bird on /var/opt/SUNWsamfs/catalog/birdcat
/dev/rmt/2cbtn   51 tp bird  on
```

▼ 建立或修改 stager.cmd 檔案及傳遞您的變更

1. 使用 vi(1) 或其他編輯器來編輯 stager.cmd 檔案。

此檔案的完整路徑如下所示：

```
/etc/opt/SUNWsamfs/stager.cmd
```

要取得有關您可以在此檔案中包括的指令之資訊，請參閱以下小節。

- 第 117 頁 「指定磁碟機數目」
- 第 117 頁 「設定分段備份緩衝區大小」
- 第 118 頁 「指定記錄檔」
- 第 121 頁 「指定分段備份請求數目」

2. 儲存並關閉 `stager.cmd` 檔案。
3. 使用 `samd(1M)` 指令及其 `config` 選項來傳遞檔案變更，然後重新啟動系統。

```
# samd config
```

指定磁碟機數目

依據預設，分段備份程式在分段備份檔案時會使用所有可用磁碟機。如果分段備份程式使所有磁碟機都處於忙碌中，則會干擾歸檔器的作業。`drives` 指令可以指定可供分段備份程式使用的磁碟機數目。此指令具有以下格式：

```
drives = 程式庫 計數
```

表 5-1 `drives` 指令的引數

引數	意義
程式庫	程式庫的「系列組」名稱，如 Sun StorEdge SAM-FS mcf 檔案中所示。
計數	表示要使用的最大磁碟機數目。依據預設，此值等於在 mcf 檔案中為此程式庫配置的磁碟機數目。

例如：以下指令行指定只有 dog 系列組程式庫中的一台磁碟機用於分段備份檔案：

```
drives = dog 1
```

若需更多有關 mcf 檔案的資訊，請參閱 `mcf(4)` 線上手冊。

此指令也可使用 File System Manager 軟體指定。請參閱 File System Manager 線上手冊以取得更多資訊。

設定分段備份緩衝區大小

依據預設，要分段備份的檔案將先讀取至記憶體中的緩衝區，然後再從歸檔媒體回復至線上磁碟快取記憶體。您可以使用 `bufsize` 指令指定非預設的緩衝區大小 (選用)，也可以鎖定緩衝區。這些動作可以提高效能，並且您可嘗試各種緩衝區大小值。此指令具有以下格式：

```
bufsize = 媒體 緩衝區大小 [ lock ]
```

表 5-2 *bufsize* 指令的引數

引數	意義
媒體	指定 <i>mcf(4)</i> 線上手冊上的清單中列出的歸檔媒體類型。
緩衝區大小	指定 2 至 32 的數字。預設值為 4。此值會與該媒體類型的 <i>dev_blksize</i> 值相乘，而使用產生的緩衝區大小。 <i>dev_blksize</i> 可在 <i>defaults.conf</i> 檔案中指定。緩衝區大小所指定的數字越大，使用的記憶體就越多。若需更多有關此檔案的資訊，請參閱 <i>defaults.conf(4)</i> 線上手冊。
lock	<p>lock 引數表示在分段備份歸檔備份時，分段備份程式是否應該使用鎖定的緩衝區。如果指定了 lock，分段備份程式在備份作業過程中會將檔案鎖定在記憶體的分段備份緩衝區內。這樣即可免除根據每個 I/O 請求來鎖定與解除鎖定緩衝區的作業，從而減少了系統 CPU 時間。</p> <p>lock 引數應該僅在具有大容量記憶體的大系統中指定。記憶體不足會導致出現記憶體不夠用的情況。</p> <p>僅在為要分段備份的檔案啟動了直接 I/O 時，lock 引數才會生效。依據預設，lock 參數並未指定，且檔案系統會鎖定所有直接 I/O 緩衝區，包括要分段備份的緩衝區。若需更多有關啟動直接 I/O 的資訊，請參閱 <i>setfa(1)</i> 線上手冊、<i>sam_setfa(3)</i> 程式庫常式線上手冊，或 <i>mount_samfs(1M)</i> 線上手冊上的 <i>-O forcedirectio</i> 選項。</p>

例如，此指令可在 *stager.cmd* 檔案的以下此行中指定：

```
bufsize=od 8 lock
```

此指令也可使用 File System Manager 軟體指定。請參閱 File System Manager 線上手冊以取得更多資訊。

指定記錄檔

您可以請求 Sun StorEdge SAM-FS 軟體收集檔案分段備份事件資訊，並將其寫入記錄檔中。*logfile* 指令可以指定分段備份程式可將記錄資訊寫入至其中的記錄檔。此指令具有以下格式：

```
logfile=檔案名稱 [ 事件 ]
```

關於檔案名稱，請指定完整的路徑名稱。

關於事件，請指定一或多個分段備份事件。如果您指令了一個以上的事件，請使用空格將每個事件隔開。依據預設，以下為啟動的事件：finish cancel error。以下為可能的事件規定：

表 5-3 事件引數的關鍵字

事件	動作
all	記錄所有分段備份事件。
start	檔案分段備份開始的時候做記錄。
finish	檔案分段備份結束的時候做記錄。依據預設啟動。
cancel	操作者取消分段備份的時候錄。依據預設啟動。
error	記錄分段備份錯誤。依據預設啟動。

指定記錄檔後，分段備份程式會在記錄檔中為每個分段備份過的檔案寫入一或多行資訊。此行包含諸如檔案名稱、分段備份的日期與時間以及 VSN 之類的資訊。

以下指令行將指定檔案 /var/adm/stage.log：

```
logfile=/var/adm/stage.log
```

程式碼範例 5-2 顯示了分段備份程式記錄檔範例。

程式碼範例 5-2 分段備份程式記錄檔範例

```
S 2003/12/16 14:06:27 dk disk01 e.76d 2557.1759 1743132 /sam1/testdir0/filebu 1
root other root 0
F 2003/12/16 14:06:27 dk disk01 e.76d 2557.1759 1743132 /sam1/testdir0/filebu 1
root other root 0
S 2003/12/16 14:06:27 dk disk02 4.a68 1218.1387 519464 /sam1/testdir1/fileaq 1
root other root 0
S 2003/12/16 14:06:43 dk disk01 13.ba5 3179.41 750880 /sam1/testdir0/filecl 1
root other root 0
F 2003/12/16 14:06:43 dk disk01 13.ba5 3179.41 750880 /sam1/testdir0/filecl 1
root other root 0
S 2003/12/16 14:06:59 dk disk01 17.167b 1155.1677 1354160 /sam1/testdir0/filedb
1 root other root 0
F 2003/12/16 14:06:59 dk disk01 17.167b 1155.1677 1354160 /sam1/testdir0/filedb
1 root other root 0
S 2003/12/16 14:06:59 dk disk02 f.f82 3501.115 1458848 /sam1/testdir1/filecb 1
root other root 0
S 2003/12/16 14:07:15 dk disk01 1f.473 1368.1419 636473 /sam1/testdir0/fileed 1
root other root 0
S 2003/12/16 14:07:15 dk disk02 16.f15 3362.45 1065457 /sam1/testdir1/filecz 1
root other root 0
```

程式碼範例 5-2 分段備份程式記錄檔範例 (續)

```
S 2003/12/16 14:07:31 dk disk01 23.201d 3005.1381 556807 /saml/testdir0/fileeq
1 root other root 0
S 2003/12/16 14:07:47 dk disk01 26.c4d 2831.1113 1428718 /saml/testdir0/fileez
1 root other root 0
S 2003/12/16 14:07:47 dk disk02 1b.835 3736.59 1787855 /saml/testdir1/filedp 1
root other root 0
```

如程式碼範例 5-2 所示，分段備份程式記錄檔含有分成九個欄位的多行資訊。表 5-4 描述分段備份程式記錄檔欄位的內容。

表 5-4 分段備份程式記錄檔欄位

欄位	內容說明
1	分段備份作業。S 代表啓動。C 代表已取消。E 代表發生錯誤。F 代表完成。
2	以 <i>yyyy/mm/dd</i> 格式表示的分段備份動作日期。
3	以 <i>hh:mm:ss</i> 格式表示的分段備份動作時間。
4	歸檔媒體類型。若需有關媒體類型的資訊，請參閱 <i>mcf(4)</i> 線上手冊。
5	VSN。
6	歸檔檔案在媒體 (<i>tar(1)</i> 檔案) 中的實體開始位置，以及檔案偏移在十六進位歸檔檔案上的實體開始位置。
7	Inode 號碼與產生號碼。重新使用 inode 號碼之後，在確定唯一性時，除了使用 inode 號碼之外，還要使用產生號碼。
8	檔案長度。
9	檔案名稱。
10	歸檔備份編號。
11	檔案的使用者 ID。
12	檔案的群組 ID。
13	請求者的群組 ID。
14	檔案分段備份的磁碟機之「設備序號」。

此指令也可使用 File System Manager 軟體指定。請參閱 File System Manger 線上手冊以取得更多資訊。

指定分段備份請求數目

您可以使用 `maxactive` 指令，指定可同時啟動的分段備份請求數目。此指令具有以下格式：

```
maxactive=數目
```

依據預設，數目為 4000。可允許的最小數目為 1。

例如，以下指令行指定佇列中同時存在的分段備份請求不能超過 500 個：

```
maxactive=500
```

stager.cmd 檔案範例

程式碼範例 5-3 顯示 `stager.cmd` 檔案範例。

程式碼範例 5-3 `stager.cmd` 檔案範例

```
# This is stager.cmd file /etc/opt/SUNWsamfs/stager.cmd
drives=dog 1
bufsize=od 8 lock
logfile=/var/adm/stage.log
maxactive=500
```

archiver.cmd 檔案在分段備份中的角色

雖然 `archiver.cmd` 檔案中的大多數指令會影響歸檔，但歸檔組指定指令可讓您指定套用於歸檔組中所有檔案的分段備份屬性。歸檔組指定指令具有以下格式：

```
歸檔組名稱 路徑 [搜尋條件 ...] [檔案屬性]
```

第 29 頁「歸檔」一章完整描述歸檔組指定指令及其引數。表 5-5 顯示可在歸檔組指定指令中出現為檔案屬性的分段備份指令。

表 5-5 可出現在 archiver.cmd 檔案中的分段備份 archiver.cmd

指令	作用
-stage a	指定應該聯合分段備份歸檔組中的檔案。
-stage d	重設回預設值。
-stage n	指定永不分段備份歸檔組中的檔案。

若需更多有關上述及其他 archiver.cmd 指令的資訊，請參閱第 29 頁「歸檔」。

設定預覽請求的優先權

歸檔器與分段備份程式均可請求載入或卸載媒體。如果請求數目超過可用於媒體載入的磁碟機數目，則超過該數目的請求將被傳送至預覽佇列。

優先權預覽佇列中的歸檔與分段備份請求，均為無法立即滿足的請求。依據預設，系統將按先進先出 (FIFO) 順序來滿足預覽請求。

預覽佇列中可以存在的項目數量，決定於 defaults.conf 檔案中的 previews= 指令。若需有關變更此指令值的資訊，請參閱 defaults.conf(4) 線上手冊。

您可以將不同的優先權指派給預覽請求，亦可在預覽程序檔中輸入指令以覆寫 FIFO 預設值。所輸入的指令將寫入以下位置：

```
/etc/opt/SUNWsamfs/preview.cmd
```

此檔案將根據請求對檔案進行分段備份還是歸檔來排定預覽請求。您亦可增加指定 VSN 的優先權。此外，preview.cmd 檔案中的設定亦可根據高值參數 (HWM) 或低值參數 (LWM) 設定，為所有或指定檔案系統重新排定預覽請求的優先權。

sam-amld 常駐程式會在啟動時讀取預覽指令。您必須每行指定一個指令。如果您在 sam-amld 常駐程式執行時變更此檔案，您則必須重新啟動 sam-amld 常駐程式以使其生效。註解行以井字號 (#) 開頭，延伸至該行的結尾。若需更多有關此檔案的資訊，請參閱 preview.cmd(4) 線上手冊。

以下兩種指令可能會出現在 preview.cmd 檔案中：

- 全域指令：適用於所有檔案系統。這些指令必須位於第一個 fs = 行的前面。

- 某個檔案系統特定的指令：位於全域指令的後面。與 `archiver.cmd` 檔案相似，`preview.cmd` 檔案包含個別檔案系統特定的指令。個別檔案系統特定的指令在檔案中必須位於所有全域指令的後面。

檔案系統指令必須以 `fs = 檔案系統名稱` 指令開頭。此指令將命名所有後續指令均適用的檔案系統。一個檔案中可能會出現多個檔案指令區塊。在下一個 `fs =` 行或檔案結尾之前，檔案系統指令均適用。

備註 – 在多個指令影響一個檔案系統時，特定檔案系統特定的指令將覆寫全域指令。

全域 VSN 與時間指令

VSN 與時間優先權指令均為全域指令。如果這兩種指令存在於 `preview.cmd` 檔案中，它們必須位於所有檔案系統特定指令的前面。換言之，它們必須位於所有 `fs =` 指令的前面。VSN 優先權指令具有以下格式：

```
vsn_priority = 值
```

此指令是一個靜態優先權係數。其表示標記為高優先權 VSN 的 VSN 之總優先權每次增加的值。`vsn_priority` 的預設值是 1000.0。在將 VSN 排定為預覽請求以擷取此值時，必須設定其優先權旗標。請使用 `chmed(1M)` 指令的 `p` 選項來設定優先權旗標 (如 `chmed +p lt.AAA123`)。設定此旗標將對目前還不是預覽請求的所有 VSN 已提交請求均有效。時間優先權指令具有下列格式：

```
age_priority = 係數
```

此指令是一個靜態優先權係數。其總體作用是動態的。系統會將 `age_priority` 係數乘以某個請求成為預覽請求的秒數。所得的結果將新增至該請求的總優先權。等待滿足請求的時間越長，時間係數就越大。設定此係數有助於確保較舊的請求，不會無限制地被具有其他高優先權係數的較新請求取代。

如果此係數大於 1.0，在計算總優先權時會增加時間係數的優先權。如果此係數小於 1.0，則會減少時間係數的優先權。如果將係數設定為 0.0，則在計算總優先權時會排除時間係數。

未設定優先權旗標的 VSN 之優先權，將隨著它在佇列中的停留時間而增加。其優先權可能會高於以後進入佇列且已設定優先權旗標的 VSN。

全域或檔案系統特定參數指令

參數預覽請求指令可以用作全域或檔案系統用指令。參數優先權指令會決定預覽請求的參數優先權 (`wm_priority`)。程式碼範例 5-4 顯示 `wm_priority` 係數為數個設定的總合。

程式碼範例 5-4 `wm_priority` 計算

```
lwm_priority +
lhwm_priority +
hlwm_priority +
hwm_priority
-----
= wm_priority
```

在 `wm_priority` 係數為正數時，計算出的總優先權結果將會增加，從而使歸檔請求的優先權高於分段備份請求的優先權。但是，`wm_priority` 係數亦可能為負數。在此情況下，歸檔請求的總優先權將會降低，從而可能導致分段備份請求的優先權高於歸檔請求的優先權。如果設定為 0.0 (或不指定任何指令)，則表示在檔案系統處於此情況時，不對歸檔請求採取任何特殊動作。若需更多相關資訊，請參閱第 126 頁「範例 1：強制分段備份請求」中的範例。

表 5-6 顯示了四種參數優先權指令及其引數

表 5-6 參數優先權指令

優先權指令	引數
<code>lwm_priority = 值</code>	關於值，請指定在檔案系統低於 LWM 等級時，歸檔請求的 <code>wm_priority</code> 係數之變更數量。預設值是 0.0。
<code>lhwm_priority = 值</code>	關於值，請指定在檔案系統從低於 LWM 等級上升至高於 LWM 等級，但仍低於 HWM 等級時，歸檔請求的 <code>wm_priority</code> 係數之變更數量。這通常表示檔案系統即將填滿。預設值是 0.0。
<code>hlwm_priority = 值</code>	關於值，請指定在檔案系統從高於 HWM 等級下降至低於 HWM 等級，但仍高於 LWM 等級時，歸檔請求的 <code>wm_priority</code> 係數之變更數量。這通常表示釋放器無法釋放足夠的磁碟空間，以讓檔案系統低於 LWM 等級。預設值是 0.0。
<code>hwm_priority = 值</code>	關於值，請指定在檔案系統高於 HWM 等級時，歸檔請求的 <code>wm_priority</code> 係數之變更數量。預設值是 0.0。

這四個參數設定共同建立了動態優先權係數，此係數包含表示檔案系統裝滿程度的百分比，以及所設定的 HWM 與 LWM 等級。指定給預覽請求的值決定於係數是全域係數、檔案系統特定係數，還是未設定。

在檔案系統從一種狀況變更為另一種狀況時，系統將根據適當的參數優先權設定，使用 (或不使用) `chmed(1M)` 指令的 `p` 選項，重新計算與該檔案系統相關的每個 VSN 之優先權。

參數優先權僅用於計算歸檔媒體請求，而不用於計算分段備份媒體請求。

以下指令範例顯示如何在檔案系統處於 HLWM 時稍微增加歸檔請求的優先權。程式碼範例 5-5 顯示用來啟動釋放器以釋放磁碟空間，讓檔案系統足以降到低於 LWM 的設定。

程式碼範例 5-5 降到低於 LWM 的設定

```
lhwm_priority = -200.0  
hlwm_priority = 100.0
```

計算預覽請求的總優先權

預覽請求的優先權數值由多個靜態與動態係數共同決定。數字越大，優先權就越高。靜態優先權係數在產生請求時設定。在產生並等待滿足請求後，其結果不會變更總優先權。動態優先權係數在等待滿足請求時可以增加或減少請求的總優先權。

預覽請求的總優先權為所有優先權係數之和。其計算方法如下所示：

```
total_priority = vsn_priority + wm_priority + (age_priority *  
time_in_sec_as_preview_request)
```

設定預覽請求優先機制

僅在系統迫切需要時，才有必要變更預設預覽請求的 FIFO 配置。如果出現以下情況，可能需要變更預設預覽請求的 FIFO 配置：

- 情況 1：確保先處理分段備份請求，然後再處理歸檔請求。
- 情況 2：確保歸檔請求在檔案系統即將填滿時取得最高優先權。
- 情況 3：將使用一組指定媒體的請求移至預覽請求清單的頂部。

對於使用者存取資料非常重要、VSN 磁帶機有限或檔案歸檔做為背景功能執行的環境，您可以使用 `preview.cmd` 檔案來改變儲存設備系統資源為分段備份請求服務的方式。您可以自訂 `preview.cmd` 檔案中的設定，使其支援任何上述的分析藍本及影響已配置的 Sun StorEdge SAM-FS 環境。

由於資料不受本檔案設定的影響，因此在衡量每個預覽請求的優先權時，您可以多方嘗試調整指令設定，以平衡歸檔請求與分段備份請求。

程式碼範例 5-6 顯示有提到前述三種情況的 `preview.cmd` 檔案範例。

程式碼範例 5-6 `preview.cmd` 檔案範例

```
# condition 1
lwm_priority = -200.0
lhwm_priority = -200.0
hlwm_priority = -200.0
# condition 2
hwm_priority = 500.0
# condition 3
age_priority = 1.0
```

範例 1：強制分段備份請求

下列設定範例展示了確保分段備份請求優先權高於歸檔請求優先權的其中一種方法。此範例假設：

- 有多個請求在佇列中等待了 100 秒鐘。
- `vsn_priority` 的預設值是 1000。

表 5-7 顯示了如何計算請求的總優先權。

表 5-7 請求優先權範例

優先	計算
歸檔含優先權的 VSN，LWM：	$1000 + (-200) + (1 \times 100) = 900$
分段備份含優先權的 VSN，LWM：	$1000 + 0 + (1 \times 1100) = 100$
分段備份不含優先權的 VSN， LWM：	$0 + 0 + (1 \times 100) = 100$

此範例顯示：在其他係數相等時，如果 `wm_priority` 為負值，則可能會導致分段備份請求的優先權高於歸檔請求的優先權。

範例 2：強制歸檔請求

在將檔案分段備份回使用者與將新檔案歸檔至媒體具有相同優先權的環境中，最值得考量的是超過 HWM。在此情況下，如果沒有足夠符合其歸檔請求的檔案來降低檔案系統填充百分比，完成待處理歸檔請求防止檔案系統填滿的另一個最佳方法。

在此情況下，`preview.cmd` 檔案可以非常簡單，如下所示：

```
hwm_priority = 500.0
```

範例 3：依媒體排定請求的優先權

在專案導向的環境中，指定使用者可能會處理使用指定 VSN 且與其他使用者隔離的檔案群組。在此環境中，某些專案在某些時候可能會具有較高的優先權，因此可能需要從可用系統儲存設備資源取得較高的優先權。您可以使用以下指令來配置 `preview.cmd` 檔案，以讓使用者及其媒體對媒體磁碟機具有適當的優先權：

```
hwm_priority = 5000.0
```

然後，對於優先權使用者群組中的每個 VSN，請輸入以下資訊：

```
# chmed +p lt.AAA123 ## 或使用的任何 VSN
```

此後，請求 VSN AAA123 (或在使用任何 VSN 時) 的每個請求，在預覽佇列中均將位於其他待處理掛載請求之上。

要在以後取消排定使用者媒體的優先權，請為每個 VSN 輸入以下相反的指令：

```
# chmed -p lt.AAA123 ## 或使用的任何媒體類型
```

範例 4：複合排定優先權

假設有兩個 Sun StorEdge SAM-FS 檔案系統，它們具有以下請求：

- 所有請求均不應在佇列中等待太久 (`age_priority`)。
- 在檔案系統低於 LWM 時，分段備份請求應該優先。
- 當檔案系統在 LWM 以上但在 HWM 以下時，則不需對歸檔或分段備份請求一對一排定優先權。程式碼範例 5-7 顯示受到影響的指令。

程式碼範例 5-7 指令

```
lwm_priority = -200.0  
lhwm_priority = 0.0  
hlwm_priority = 0.0
```

在此情況下，其他指令保持不變。

如果其中一個檔案系統超過 HWM，歸檔請求應該優先。

如果兩個檔案系統都超過 HWM，尤其要注意防止第二個檔案系統 (如 `samfs2`) 填滿。如果 `samfs1` 是使用者工作的檔案系統，而 `samfs2` 是關鍵系統的檔案系統，則更應如此。

在所有情況下，如果設定了 `chmed(1M)` 指令的 `p` 旗標，則選定 VSN 群組的請求在預覽請求佇列中優先。

程式碼範例 5-8 顯示根據上述清單請求排定請求優先權的 `preview.cmd` 檔案。

程式碼範例 5-8 `preview.cmd` 檔案

```
age_priority = 100.0
vsn_priority = 20000.0
lhwm_priority = -200.0
hlwm_priority = -200.0
fs = samfs1
hwm_priority = 1000.0
fs = samfs2
hwm_priority = 5000.0
```

回收

回收是重新恢復歸檔磁碟區空間的程序。回收器使用歸檔器以重新恢復未使用的歸檔備份所佔用的空間。在使用者修改檔案時，與舊版本相關的歸檔備份可從系統中清除。回收器會識別過期的歸檔備份所佔比例最大的磁碟區，並會指示將未過期的備份移至不同的磁碟區。如果指定磁碟區僅存在過期備份，則會執行站點定義的動作。例如，磁碟區可以重新製作標籤以立即重新使用，或匯出至遠端儲存裝置，從而保留檔案變更的單獨歷史記錄。使用者不瞭解回收程序，因為它與使用者的資料檔案相關。

本章包含以下主題：

- 第 129 頁 「回收程序簡介」
- 第 131 頁 「使用回收指令」
- 第 133 頁 「規劃回收作業」

回收程序簡介

回收器會將過期的歸檔備份所使用的空間容量，保持為站點指定的參數所定義的最小值。在任何時候，指定歸檔磁碟區的空間由以下部分組成：

- 目前資料指的是用於目前作用中歸檔影像的空間。
- 過期資料指的是目前過期的歸檔影像所使用的空間。
- 可用空間指並非目前作用中或過期歸檔影像所使用的空間。

磁碟區的容量指磁碟區中資料空間的總容量。例如，10 GB 的磁帶磁碟區中寫入 3 GB 的資料，則其容量是 10 GB，可用空間是 7 GB。

新的或重新製作標籤的歸檔媒體一開始的所有容量均為可用空間。隨著資料歸檔至媒體，可用空間的容量將會減少，而目前資料的容量將會增加。

隨著檔案系統中歸檔檔案的變更或移除，其歸檔影像將會過期並從目前的資料分類移至過期的資料分類。這些影像所使用的實體空間保持不變；僅在檔案系統中不再有檔案指向此空間。

這些過期的影像 (即過期的資料) 最終將佔用所有的可用空間。僅在空間可回收時，這些影像才能移除，其佔用的空間才能釋放。回收器的目的即是將過期資料所使用的空間轉換成可用空間，而不會遺失任何目前的資料。

例如：可移除的媒體卡匣 (例如磁帶) 就只能附加，而無法重新寫入原位置。重新使用卡匣的唯一方法是將目前所有的資料移出卡匣，並為卡匣重新製作標籤，然後再次從頭開始使用卡匣。

您需要輸入 `sam-recycler(1M)` 指令才可初始化回收。可以透過手動或透過 `cron(1)` 工作來執行。表 6-1 顯示回收方法。

表 6-1 回收方法和媒體類型

回收方法	媒體和注意事項
使用自動程式庫	可移除的媒體卡匣。 當您使用程式庫進行歸檔時，您是將回收指令輸入到 <code>recycler.cmd</code> 檔案中。
使用歸檔組	可移除的媒體卡匣和磁碟。 當您使用歸檔組進行歸檔時，您不是使用 <code>recycler.cmd</code> 檔案，而是將所有回收指令輸入到 <code>archiver.cmd</code> 檔案中。

如表 6-1 顯示，您可以選擇使用程式庫或使用歸檔組進行回收。如果您要歸檔到磁碟，您就只能使用歸檔組進行回收。

回收器和歸檔器會一起運作，詳情如下：

1. 回收器會將磁碟區上存有的所有目前 (有效) 的歸檔影像標示為 `rearchive` 屬性。
2. 如果您要歸檔到可移除的媒體，回收器則會將選定的歸檔磁碟區標示為 `recycle` 屬性。這樣即可防止歸檔器繼續將任何其他歸檔影像寫入磁碟區。
3. 歸檔器會將所有標示的影像移動到另一個磁碟區。此作業就叫作重新歸檔。在歸檔器將目前的歸檔影像從舊磁碟區移動到新磁碟區之後，舊磁碟區則只含有可用空間和過期空間。如果您要歸檔到可移除的媒體卡匣，您可以重新製作卡匣標籤及重新使用卡匣。如果您要歸檔到磁碟，回收器則會移除含有過期歸檔影像的檔案。

回收器設計為定期執行。回收器在每次啓動時會執行儘可能多的工作。回收器必須先完成標示重新歸檔的備份，歸檔器才能進行檔案的重新歸檔。

有時設為 `rearchive` 屬性的過期歸檔影像仍會存在於媒體上。在以下情況下可能會發生此情況：

- 歸檔器未在回收器標示過期的歸檔影像之後執行。
- 在移動未過期的歸檔影像時，歸檔器沒有可以使用的媒體。
- 其他歸檔器異常現象。

在兩次執行之間，回收器會將狀態資訊保留在程式庫目錄檔與 `inodes` 中。在回收程序期間，您可以使用 `sls(1)` 指令及其 `-D` 選項來顯示檔案的相關資訊。`sls(1)` 指令的輸出會顯示檔案是否已排定要進行重新歸檔。

使用回收指令

`recycler.cmd` 檔案接受以下章節中說明的指令：

- 第 131 頁「指定記錄檔：`logfile` 指令」
- 第 131 頁「防止回收：`no_recycle` 指令」
- 第 132 頁「指定回收整個自動程式庫：程式庫指令」

指定記錄檔：`logfile` 指令

`logfile` 指令會指定回收器記錄檔。此指令具有以下格式：

```
logfile = 檔案名稱
```

關於**檔案名稱**，請指定記錄檔的路徑。

以下是 `logfile=` 指令行範例：

```
logfile=/var/adm/recycler.log
```

防止回收：`no_recycle` 指令

`no_recycle` 指令可讓您防止回收磁碟區。要指定 `VSN`，可使用正規表示式與一個或多個指定的媒體類型。此指令具有以下格式：

```
no_recycle 媒體類型 VSN_regex [ VSN_regex ... ]
```

表 6-2 no_recycle 指令的引數

引數	意義
媒體類型	在 mcf(4) 線上手冊中指定媒體類型。
VSN_regex	指定一個或多個用空格隔開的正規表示式以說明磁碟區。若需關於 regex 格式的資訊，請參閱 regexp(5) 線上手冊，或參閱第 54 頁「使用模式比對的檔案名稱搜尋條件：-name regex」。

指定**媒體類型**即可防止回收儲存在特定類型的媒體中的磁碟區。一個或多個 VSN_regex 指定可讓您使用正規表示式識別要排除在回收之外的指定卡匣。

例如，以下指令行排除在回收任何磁帶磁碟區之外，其 VSN 識別標記開頭為 DLT：

```
no_recycle lt DLT.*
```

指定回收整個自動程式庫：程式庫指令

程式庫指令可讓您為指定程式庫相關的 VSN 指定不同的回收參數。此指令具有以下格式：

```
程式庫 參數 [ 參數 ... ]
```

關於**程式庫**，請指定如 mcf(4) 檔案「系列組」欄位中所指定的程式庫名稱。

關於**參數**，請指定表 6-3 中一或多個以空格隔開的參數關鍵字。

表 6-3 程式庫指令的參數值

參數	動作
-dataquantity 大小	透過清除磁碟區的有用資料，限制回收器可排定用於重新歸檔的資料數量。預設值是 1 GB。
-hwm 百分比	表示程式庫高值參數。預設值是 95。
-ignore	防止回收此程式庫中的磁碟區。此指令在測試 recycler.cmd 檔案時有用。
-mail 電子郵件位址	將回收電子郵件訊息傳送至指定的電子郵件位址。依據預設，不會傳送電子郵件。
-mingain 值	表示最低 VSN 擷取。預設值是 50。
-vsncount 計數	限制回收至計數的磁碟區數目。預設值是 1。

例如以下指令行：

```
gr47 -hwm 85 -ignore -mail root -mingain 40
```

該指令行指定了程式庫 `gr47` 的以下內容：

- 在程式庫中的磁碟區耗盡 85% 時，應考慮回收程式庫。
- 最小擷取百分比為 40%。
- 重新歸檔的資料不能大於 1 GB。此為預設值，因此不會在 `recycler.cmd` 檔案中指定。
- 僅回收一個磁碟區。這也是預設設定。
- 回收訊息將透過電子郵件傳送至 `root`。

規劃回收作業

設定回收器之前，請注意下列內容：

- `archiver.cmd` 檔案中的指令按歸檔組控制回收。`recycler.cmd` 檔案中的指令按程式庫控制回收。此外，`recycler.cmd` 檔案控制一般的回收器運作方式。若需有關回收器指令的資訊，請參閱第 131 頁「使用回收指令」。
- 請勿回收含有可移除的媒體檔案的磁碟區。您是使用 `request(1)` 指令來建立可移除的媒體檔案。回收器不會保留使用 `request(1)` 指令建立的可移除的媒體檔案。具有可移除的媒體檔案的磁碟區將不會耗盡。
- 在 Sun StorEdge SAM-FS 檔案系統上執行維護時，請勿執行回收器。回收器使用 `.inodes` 檔案與 `mcf` 檔案以協助識別目前或過期的檔案，以及與檔案系統相關的裝置。這些檔案中缺少適當的資訊可導致目前歸檔的資料成為過期資料並被回收。
- 在回收器執行時，所有 Sun StorEdge SAM-FS 檔案系統都必須掛載。如果從線上磁碟回收，包含磁碟區的檔案系統必須已掛載，並且主機系統必須可存取。

回收器依預設未啓用。您必須輸入 `sam-recycler(1M)` 指令才可初始化回收。在初始化回收器時，第 132 頁「指定回收整個自動程式庫：程式庫指令」中指定的預設回收器設定會生效。若需更多有關釋放器的資訊，請參閱 `sam-recycler(1M)` 線上手冊。

以下章節說明配置回收器的程序。本程序包含以下步驟：

- 第 134 頁「步驟 1：建立 `recycler.cmd` 檔案」
- 第 136 頁「步驟 2：編輯 `archiver.cmd` 檔案」
- 第 137 頁「步驟 3：執行回收器」
- 第 139 頁「步驟 4：建立回收器的 `crontab` 檔案」
- 第 139 頁「步驟 5：移除 `-recycle_ignore` 和 `ignore` 參數」

- 第 139 頁「步驟 6：建立 `recycler.sh` 檔案」

如果您要歸檔到程式庫中的卡匣，此程序包括建立 `recycler.cmd` 檔案與編輯 `archiver.cmd` 檔案 (選用)。如果您要歸檔到磁碟，您僅可按歸檔組歸檔才能對這些磁碟區啟動回收，並且編輯 `archiver.cmd` 檔案。下列程序說明使用 `recycler.cmd` 和 `archiver.cmd` 檔案配置任何歸檔媒體的回收器。或者，回收可使用 `File System Manager` 軟體配置。若需更多資訊，請參閱 `File System Manager` 線上手冊。如果您透過 `File System Manager` 配置回收，您仍需要完成下列的步驟 3、4 和 6。

▼ 步驟 1：建立 `recycler.cmd` 檔案

如果要回收程式庫的卡匣中的歸檔備份，請執行此步驟。

如果您要回收磁碟區中的歸檔備份，您將無法完成此步驟，因為回收是由 `archiver.cmd` 檔案中的指令所控制。若需有關在 `archiver.cmd` 檔案中配置回收的資訊，請參閱第 136 頁「步驟 2：編輯 `archiver.cmd` 檔案」。

`recycler.cmd` 檔案包括一般回收指令，也可以包含 `Sun StorEdge SAM-FS` 環境中每個程式庫所用的指令。若需有關回收指令的資訊，請參閱第 131 頁「使用回收指令」。

即使要按歸檔組回收，仍然應該在 `recycler.cmd` 檔案中配置每個程式庫。這可確保沒有成爲歸檔組的 `VSN` 在需要時可以回收。

一般的 `recycler.cmd` 檔案包含以下指令行：

- 一個指定回收器記錄檔的 `logfile=` 指令行。系統會將回收訊息與回收回報寫入此檔案。
- 一個或多個指令行，用於每個包含要回收磁碟區的程式庫。此行必須包含要回收程式庫的「系列組」名稱 (在 `mcf` 檔案中)。這可讓回收器識別程式庫。

由於仍在建立的 `recycler.cmd` 文字行尚未經過測試，因此請使用 `ignore` 關鍵字。您將在此程序的稍後步驟中移除 `ignore` 關鍵字。

若要建立 `recycler.cmd` 檔案，請執行以下步驟：

1. 成爲超級使用者。
2. 使用 `vi(1)` 或其他編輯器來開啟 `/etc/opt/SUNWsamfs/recycler.cmd` 檔案。
3. 新增一或多個本章所述的指令以控制回收器作業。
4. 儲存並關閉檔案。

recycler.cmd 檔案範例

程式碼範例 6-1 顯示 recycler.cmd 檔案範例。

程式碼範例 6-1 recycler.cmd 檔案範例

```
logfile = /usr/tmp/recycler.log  
stk30 -hwm 51 -mingain 60 -ignore -mail root
```

下列章節說明程式碼範例 6-1 中指定的參數。

-hwm 51 參數

指定高值參數即可設定媒體使用的百分比，低於此百分比回收將無法發生。此百分比是指程式庫中使用的空間佔其總容量的比率。例如：一個容納 10 個 20 GB 磁帶的程式庫，其中有三個使用了 100% 的空間，剩餘七個每個使用了 30%，媒體使用率的百分比如下所示：

$$\left((3 * 1.00 + 7 * 0.30) * 20G \right) / \left(10 * 20G \right) * 100\% = 51\%$$

請注意，此計算不區別目前資料與過期資料。僅表示使用的媒體數目。

在此範例中，如果公用程式百分比等於或小於 51%，回收器不會自動選擇回收任何自動程式庫的 VSN。

備註 – 您可以藉由使用以下指令設定回收旗標以強制回收 VSN：

```
# chmed +c lt.AAA123
```

當設定了 +c 旗標時，歸檔器就不會繼續將任何其他歸檔影像寫入磁碟區中。可透過 samu(1M) 公用程式檢視 +c 旗標。若需更多相關資訊，請參閱 chmed(1M) 和 samu(1M) 線上手冊。若需關於使用 samu(1M) 公用程式的資訊，請參閱「Sun StorEdge QFS 配置與管理指南」。

-mingain 60 參數

最低 VSN 擷取百分比設定回收卡匣所擷取空間容量的較低限制。例如：如果自動程式庫的卡匣中有 95% 的目前資料與 5% 的過期資料，則透過回收卡匣僅擷取 5% 的資料。因此不太值得移動其他 95% 的資料以擷取此空間。將最低擷取設定為等於或大於 6% 即可抑制回收器自動選擇此範例 VSN。

另一個範例是卡匣中的過期資料為 90%、目前資料為 5%、可用空間為 5%。這將在回收時擷取 90% 的資料。

-ignore 參數

-ignore 參數可防止回收器回收特定的程式庫，應在配置回收器時使用。

-mail 參數

-mail 參數指定回收器在回收發生在指定的程式庫時傳送郵件。郵件訊息具有以下主旨行：

```
Robot 自動裝置名稱 recycle
```

表 6-2 顯示訊息內容範例。

程式碼範例 6-2 回收訊息範例

```
I will recycle VSN vsn.
Cannot find any candidate VSN in this media changer.
Previously selected VSN vsn is not yet finished recycling.
Previously selected VSN vsn is now finished recycling. It will now
be post-recycled.
```

▼ 步驟 2：編輯 archiver.cmd 檔案

如果要按歸檔組回收，請執行此步驟。如果您要歸檔到磁碟，按歸檔組回收是可能執行回收的唯一方法。因此如果您要歸檔到磁碟，您就必須完成此步驟以進行回收。

如果要按程式庫回收，您可繼續下一步驟。

- 若要編輯 archiver.cmd 檔案，請執行第 40 頁「建立或修改 archiver.cmd 檔案及傳遞您的變更」程序。

您爲了按歸檔組啟動回收而新增到 archiver.cmd 檔案的指令，必須出現於 params 和 endparams 指令之間。表 6-4 顯示您可以使用的歸檔組回收指令。

表 6-4 歸檔組回收指令

指令	功能
-recycle_dataquantity 大小	透過清除磁碟區的有用資料，限制回收器可排定用於重新歸檔的資料數量。
-recycle_hwm 百分比	設定高值參數百分比。
-recycle_ignore	防止回收歸檔組。

表 6-4 歸檔組回收指令 (續)

指令	功能
-recycle_mailaddr 郵件位址	將回收器訊息傳送到郵件位址。
-recycle_mingain 百分比	限制回收可能透過百分比或其他來增加其可用空間的 VSN。
-recycle_vsncount 計數	限制重新歸檔至計數的磁碟區數目。

若需更多有關上述指令的資訊，請參閱第 29 頁「歸檔」，或參閱 archiver.cmd(4) 線上手冊。

▼ 步驟 3：執行回收器

1. 核發 `sam-recycler(1M)` 指令。
回收器將讀取 `recycler.cmd` 檔案。
2. 檢查回收器的輸出記錄、Sun StorEdge SAM-FS 記錄及 `/var/adm/messages` 中有無任何錯誤訊息。
如有錯誤，請更正檔案。

程式碼範例 6-3 顯示回收可移除的媒體卡匣的回收器記錄記錄檔範例。

程式碼範例 6-3 可移除的媒體卡匣的回收器記錄檔範例

```

===== Recycler begins at Wed Dec 12 14:05:21 2001 =====
Initial 2 catalogs:

0  Family: m160                Path: /var/opt/SUNWsamfs/catalog/m160
   Vendor: ADIC                Product: Scalar 100
   SLOT          ty    capacity      space vsn
     0           at    25.0G          25.0G CLN005
     1           at    48.5G           6.1G 000003
     2           at    48.5G          32.1G 000004
     3           at    48.5G          35.1G 000005
     4           at    48.5G          44.6G 000044
     5           at    48.5G          45.1G 000002
     6           at    48.5G          45.9G 000033
     7           at    48.5G          48.5G 000001

Total Capacity: 364.8G bytes, Total Space Available: 282.3G bytes
Volume utilization 22%, high 95% VSN_min 50%
Recycling is ignored on this robot.

```

程式碼範例 6-3 可移除的媒體卡匣的回收器記錄檔範例 (續)

```

1 Family: hy Path: /var/opt/SUNWsamfs/catalog/historian
Vendor: Sun SAM-FS Product: Historian
SLOT ty capacity space vsn
(no VSNs in this media changer)
Total Capacity: 0 bytes, Total Space Available: 0 bytes
Volume utilization 0%, high 95% VSN_min 50%
Recycling is ignored on this robot.

8 VSNs:

---Archives--- -----Percent----- m160
----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 87 13 m160:at:000003
no-data VSN 0 0 0 33 67 m160:at:000004
no-data VSN 0 0 0 27 73 m160:at:000005
no-data VSN 0 0 0 8 92 m160:at:000044
no-data VSN 0 0 0 7 93 m160:at:000002
no-data VSN 0 0 0 5 95 m160:at:000033
empty VSN 0 0 0 0 100 m160:at:CLN005
empty VSN 0 0 0 0 100 m160:at:000001

Recycler finished.

===== Recycler ends at Wed Dec 12 14:05:32 2001 =====

```

程式碼範例 6-4 顯示了回收磁碟歸檔檔案的回收器記錄檔範例。

程式碼範例 6-4 磁碟歸檔檔案的回收器記錄檔範例

```

---Archives--- -----Percent-----
----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
new candidate 0 0 0 41 59 <none>:dk:disk01

677 files recycled from VSN disk01 (mars:/sam4/copy1)
0 directories recycled from VSN disk01 (mars:/sam4/copy1)

```

▼ 步驟 4：建立回收器的 crontab 檔案

如果系統按預期執行，您即可為超級使用者建立一個 crontab 項目以定期執行回收器。視站點的情況而定，您可能希望每兩個小時執行回收器的次數不要多於一次。

- 建立 crontab 項目。

若需相關資訊，請參閱 cron(1M) 線上手冊。

root 的 crontab 檔案中的以下範例項目確保 cron 常駐程式在每個奇數小時之後，每五分鐘執行一次回收器：

```
5 1,3,5,7,9,11,13,15,17,19,21,23 * * * /opt/SUNWsamfs/sbin/sam-recycler
```

▼ 步驟 5：移除 -recycle_ignore 和 ignore 參數

1. 使用 vi(1) 或其他編輯器來從 archiver.cmd 檔案移除 -recycle_ignore 參數。

2. 使用 vi(1) 或其他編輯器來從 recycler.cmd 檔案移除 ignore 參數。

現在即開始回收。

▼ 步驟 6：建立 recycler.sh 檔案

如果要回收可移除的媒體卡匣中的歸檔備份，請執行此步驟。如果僅歸檔至磁碟，請不要執行此步驟。

在 VSN 中所有目前影像已重新歸檔至其他 VSN 時，回收器將執行 recycler.sh 程序檔。若需程序檔的範例，請參閱 recycler.sh(1M) 線上手冊。

/opt/SUNWsamfs/examples/recycler.sh 中提供了另一個範例，顯示如何重新製作已回收 VSN 的標籤，以及傳送郵件給超級使用者。

回收器使用以下引數呼叫 /opt/SUNWsamfs/sbin/recycler.sh 程序檔：

```
Media type: $1 VSN: $2 Slot: $3 Eq: $4
```

在回收器決定 VSN 已耗盡所有已知作用中的歸檔備份時，將會呼叫 /opt/SUNWsamfs/sbin/recycler.sh 程序檔。您應決定無需已回收卡匣的站點請求。某些站點選擇為卡匣製作標籤並重新使用卡匣；其他則選擇移除自動程式庫中的卡匣，以稍後用於存取歷史檔案。若需更多相關資訊，請參閱 recycler(1M) 和 recycler.sh(1M) 線上手冊。

第7章

使用 Sun SAM-Remote 軟體

Sun SAM-Remote 用戶端與 Sun SAM-Remote 伺服器構成了用戶端/伺服器機制，可讓您在 Sun StorEdge SAM-FS 主機系統之間共用程式庫及其他可移除的媒體裝置。Sun SAM-Remote 可讓您設定歸檔與分段備份中央程式庫或磁光碟庫中檔案的多個儲存用戶端。例如：如果您的主機系統所在網路跨越廣闊的地理區域，您可以將在某個城市建立的檔案歸檔至遠在異地的程式庫中的卡匣。

本章包含下列主題：

- 第 141 頁 「Sun SAM-Remote 軟體簡介」
- 第 146 頁 「設定 Sun SAM-Remote 軟體」
- 第 159 頁 「使用 Sun SAM-Remote 軟體進行回收」

Sun SAM-Remote 軟體簡介

本簡介中涵蓋下列主題：

- 第 141 頁 「功能」
- 第 143 頁 「需求」
- 第 143 頁 「限制」
- 第 143 頁 「技術簡介」

功能

Sun SAM-Remote 軟體具有下列優點：

- 可讓您設定遠端共用，以便在一個或多個 Sun SAM-Remote 用戶端共用寶貴的可移除的媒體資源，例如：程式庫。
- 可讓用戶端將資料移至伺服器。

- 啓用多重 Sun StorEdge SAM-FS 伺服器使其成爲彼此的主機。在 Sun SAM-Remote 環境中，伺服器是指在 mcf 檔案中設備類型設定爲 ss 的主機系統。

您可以設定 Sun SAM-Remote 伺服器與用戶端，以在兩個或多個 Sun Solaris 主機系統之間提供多個歸檔備份。例如：您可將執行 Sun StorEdge SAM-FS 軟體的兩個 Solaris 系統，設定爲彼此的 Sun SAM-Remote 伺服器與 Sun SAM-Remote 用戶端。此配置的優點之一，就是可以爲在其他伺服器上具有資料額外歸檔備份的每台伺服器建立本機備份。您可以使用標準 NFS 在伺服器之間共用檔案系統。在對本機程式庫的存取中斷時，Sun SAM-Remote 軟體會自動從歸檔備份擷取檔案資料。因此，即使主要儲存程式庫無法使用，伺服器使用者仍可繼續存取其資料。

圖 7-1 顯示了設定有兩台 Sun SAM-Remote 主機系統伺服器的環境。每台伺服器具有兩個用戶端。

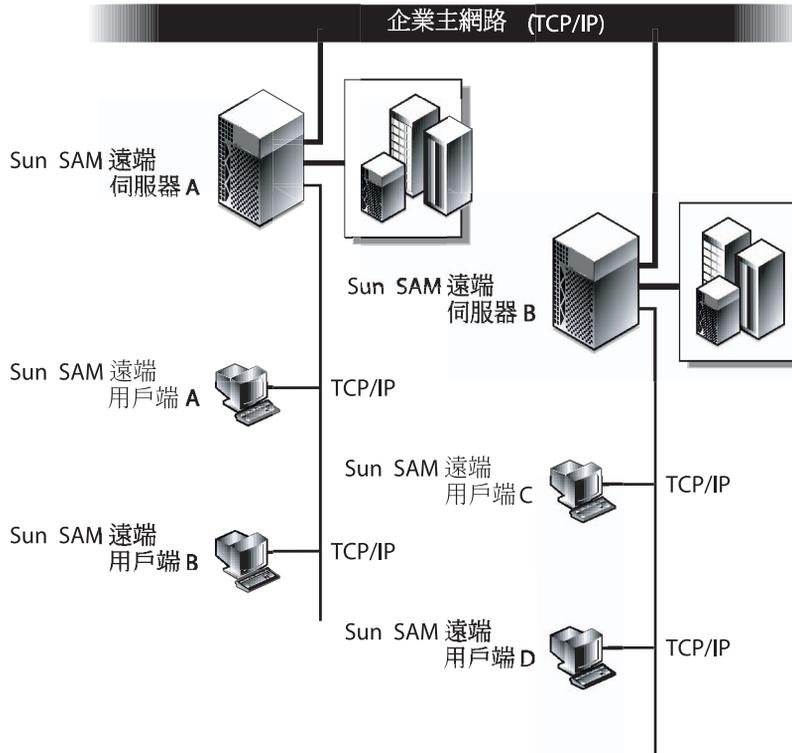


圖 7-1 Sun SAM-Remote 伺服器與用戶端

需求

嘗試設定 Sun SAM-Remote 環境之前，請確定您的環境包含下列軟體與硬體：

- 具有已授權、已安裝和可作業的 Sun StorEdge SAM-FS 4U0 或較新的儲存與歸檔管理套裝軟體的 SPARC 或 x64 系統。
- 具有相同的 Sun StorEdge SAM-FS 軟體版本等級和相同已安裝之修補程式集的主機系統。如果某些主機系統必須升級，您可以在 Sun StorEdge SAM-FS Installation and Upgrade Guide 中找到有關該主題的資訊。
- 作為 Sun SAM-Remote 伺服器，且至少安裝了一個 Sun SAM-QFS 檔案系統的一個主機系統。
- 網路連線在用戶端和安裝 Sun StorEdge SAM-FS 軟體的伺服器間執行 TCP/IP 連線。

限制

儲存與歸檔管理員會以相同方式處理遠端程式庫與本機程式庫中的卡匣。但是，下列資訊指明了 Sun SAM-Remote 軟體的限制：

- 您可以使用 Sun SAM-Remote 回收媒體，但只能在徹底測試您的環境後，才嘗試執行此操作。如需更多資訊，請參閱第 159 頁「使用 Sun SAM-Remote 軟體進行回收」。
- Sun SAM-Remote 用戶端上只有一個常駐程式可以與 Sun SAM-Remote 伺服器通訊。
- Sun StorEdge SAM-FS 軟體和 SAM-Remote 無法在共用的 Sun StorEdge QFS 檔案系統之上 Sun StorEdge QFS 用戶端作業。當在某些 Sun StorEdge QFS 檔案系統之中介資料伺服器和 Sun StorEdge QFS 之用戶端的伺服器上執行時，Sun StorEdge SAM-FS 和 SAM-Remote 僅會在伺服器為中介資料伺服器之檔案系統上作業。

技術簡介

Sun SAM-Remote 用戶端使用 TCP/IP 連線與 Sun SAM-Remote 伺服器互動。Sun SAM-Remote 用戶端之間的網路可以是 Sun Solaris 作業環境支援的任何網路類型，例如：乙太網路、快速乙太網路或光纖通道。

圖 7-2 顯示了 Sun SAM-Remote 用戶端與 Sun SAM-Remote 伺服器之間的互動。

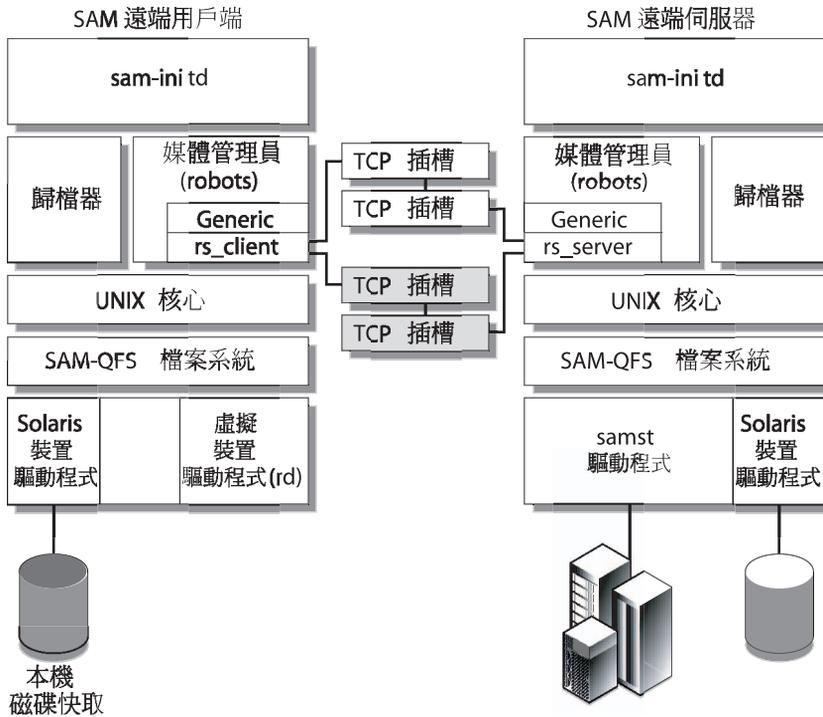


圖 7-2 Sun SAM-Remote 伺服器與用戶端之間的互動

Sun SAM-Remote 伺服器簡介

Sun SAM-Remote 伺服器組成包括全容量 Sun StorEdge SAM-FS 儲存管理主機，以及定義用戶端間共用的程式庫之 Sun SAM-Remote 伺服器常駐程式。Sun SAM-Remote 伺服器上須至少安裝一個 SAM-QFS 檔案系統。

在伺服器系統的 `/etc/opt/SUNWsamfs/mcf` 檔案中新增一行，並將設備類型設定為 `ss`，即可將主機系統定義為 Sun SAM-Remote 伺服器。您必須為每台伺服器提供一個唯一的系列組名稱。您最多可以為每個常駐程式設定 10 個用戶端。要配置多於 10 個用戶端，請位您要配置的每個 10 個用戶端在 `mcf` 檔案增加額外的遠端伺服器。若需更多有關伺服器常駐程式的資訊，請參閱 `sam-remote(7)` 線上手冊。

Sun SAM-Remote 用戶端簡介

Sun SAM-Remote 用戶端為 Sun StorEdge SAM-FS 主機系統，其建立包含虛擬裝置數目的 Sun SAM-Remote 用戶端常駐程式。

在用戶端系統的 `/etc/opt/SUNWsamfs/mcf` 檔案中新增一行，並將設備類型設定為 `sc`，即可將主機系統定義為 Sun SAM-Remote 用戶端。若需更多有關用戶端常駐程式的資訊，請參閱 `sam-remote(7)` 線上手冊。

虛擬裝置可定義 Sun SAM-Remote 伺服器上實際可移除的媒體裝置的網路連線。虛擬裝置的設備類型為 `rd`，亦即 *remote device* (遠端裝置) 的快捷鍵。您可以在 Sun SAM-Remote 用戶端的 `/etc/opt/SUNWsamfs/mcf` 檔案中定義虛擬裝置。Sun SAM-Remote 常駐程式與虛擬裝置均與一台特定伺服器相關聯。

Sun SAM-Remote 常駐程式支援每個用戶端虛擬裝置數目並無限制。用戶端使用的實際虛擬裝置數目是可以設定的。在確定應該為每個用戶端設定多少個虛擬裝置時，請將其設定為用戶端與伺服器之間的同步資料傳輸數。因為定義的虛擬裝置越多，網路流量總負載可能會越大。作為系統管理員，您可以自行決定系統所需的實際虛擬裝置數目。

Sun SAM-Remote 伺服器與 Sun SAM-Remote 用戶端之間的互動

Sun SAM-Remote 伺服器常駐程式，`sam-serverd` 偵聽連接埠 1000 上的用戶端。您可以在配有 `rmtsam` 服務名稱的 Sun Solaris `/etc/services` 目錄上配置不同的連接埠。在 Sun SAM-Remote 用戶端連線到 Sun SAM-Remote 伺服器時，`sam-serverd` 常駐程式會在另一個連接埠上建立連線，並使用所定義的連接埠將此連接埠號碼傳送到該用戶端。通訊端大小也會傳送到該用戶端。通訊端大小是可以設定的，在第 146 頁「設定 Sun SAM-Remote 軟體」中將詳細說明。

程式庫目錄

Sun SAM-Remote 程式庫目錄是 Sun SAM-Remote 伺服器上的目錄子集。用戶端目錄檔會即時更新。只有 Sun SAM-Remote 伺服器才能控制分配給 Sun SAM-Remote 用戶端目錄檔的插槽。

初始化之後，系統會根據 Sun SAM-Remote 伺服器目錄檔中的資訊建立用戶端目錄檔，並將其傳送到 Sun SAM-Remote 用戶端。在主機與用戶端之間建立連線後，會將用戶端可以使用的媒體標記為可用。如果用戶端與伺服器之間的連線中斷，則會將用戶端上的媒體標記為無法使用。您可以透過 `samu(1M) v` 顯示來檢視媒體可用性。用戶端上的 `samu(1M) v` 顯示中出現的資訊，是伺服器上 `v` 顯示中出現的資訊子集。通常，您應該透過 Sun SAM-Remote 伺服器上的 `samu(1M) v` 顯示來存取媒體目錄檔。若需更多有關 Sun SAM-Remote 伺服器用戶端檔案的資訊，請參閱第 146 頁「設定 Sun SAM-Remote 軟體」。若需關於使用 `samu(1M)` 公用程式的資訊，請參閱「Sun StorEdge QFS 配置和管理指南」。

如有必要，系統將在主機之間傳送對目錄檔所作的變更。任何伺服器目錄檔中涉及用戶端相關媒體類型的變更，均會傳送到用戶端，並在隨後更新用戶端目錄檔。

歸檔

Sun SAM-Remote 歸檔程序與 Sun StorEdge SAM-FS 的歸檔程序相同。Sun SAM-Remote 用戶端會使掛載請求新增到伺服器的掛載請求表。然後，用戶端會等待伺服器傳回指明媒體是否已掛載的回應訊息。只要媒體可以使用，便會開始歸檔程序。

設定 Sun SAM-Remote 軟體

本節說明如何執行 Sun SAM-Remote 伺服器與用戶端軟體的初始配置。其包含下列章節：

- 第 146 頁 「配置範例」
- 第 147 頁 「配置軟體」

配置範例

圖 7-3 說明了本章的程序中所使用的配置範例。本章中的範例將說明如何設定稱為 chicago 的 Sun SAM-Remote 伺服器。

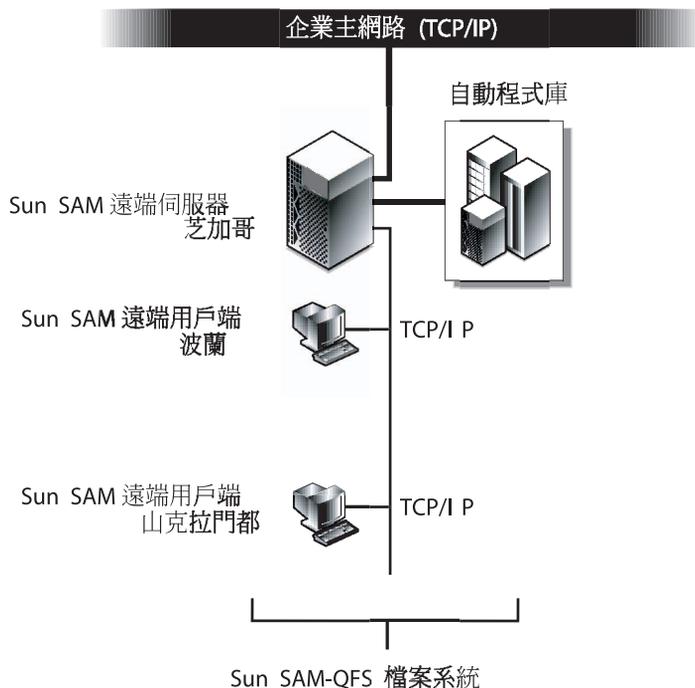


圖 7-3 Sun SAM-Remote 配置範例

在 portland 和 sacramento 上的 Sun StorEdge SAM-FS 檔案系統使用 chicago 做為 Sun SAM-Remote 伺服器。

本章範例中，Sun StorEdge SAM-FS 檔案系統會將其部分歸檔備份寫入受 chicago 控制的卡匣。

配置軟體

下列程序將說明如何在 Sun SAM-Remote 伺服器，以及一個或多個 Sun SAM-Remote 用戶端上設定 Sun SAM-Remote 軟體。這些程序必須依照下列所示的順序執行。

1. 第 147 頁 「登入潛在的伺服器與用戶端主機」
2. 第 147 頁 「確認用戶端與伺服器配置」
3. 第 149 頁 「編輯 mcf 檔案」
4. 第 151 頁 「定義 Sun SAM-Remote 用戶端」
5. 第 152 頁 「在伺服器的 mcf 檔案中定義 Sun SAM-Remote 伺服器」
6. 第 153 頁 「建立 Sun SAM-Remote 伺服器配置檔案」
7. 第 155 頁 「啓用歸檔」

在下列步驟中，您將登入主機系統，確認現有軟體版本，並在必要時升級軟體。

▼ 登入潛在的伺服器與用戶端主機

您必須以超級使用者的身份登入所有潛在的伺服器與用戶端主機。

1. 以超級使用者的身份登入 Sun SAM-Remote 伺服器。

您必須對要安裝 Sun SAM-Remote 軟體的伺服器系統具有超級使用者權限。

2. 以超級使用者的身份登入 Sun SAM-Remote 用戶端。

您必須對要安裝 Sun SAM-Remote 軟體的用戶端系統具有超級使用者權限。

▼ 確認用戶端與伺服器配置

下列步驟可確保您在要設定為 Sun SAM-Remote 環境組成部分的系統上安裝所需的軟體版本。

1. 在要設定為 Sun SAM-Remote 用戶端或伺服器的所有主機上發出含 -l 選項的 pkginfo(1M) 指令。

您必須在要設定為 Sun SAM-Remote 環境組成部分的所有用戶端與伺服器主機上，均安裝相同的 Sun StorEdge SAM-FS 軟體版本和修改等級。例如：

程式碼範例 7-1 使用 pkginfo(1)

```
portland# pkginfo -l SUNWsamfs
  PKGINST:  SUNWsamfs
    NAME:    Sun SAM-FS and Sun SAM-QFS software Solaris 2.8
CATEGORY:  system
  ARCH:    sparc
VERSION:   4.0.5,REV=5.8.2003.01.12
  VENDOR:   Sun Microsystems, Inc.
  PSTAMP:   boomerang-20020712183351
INSTDATE:  Jan 20 2003 07:30
HOTLINE:   Please contact your local service provider
STATUS:    completely installed
  FILES:    489 installed pathnames
            12 shared pathnames
            1 linked files
            51 directories
            179 executables
            35813 blocks used (approx)

portland#
```

2. 檢查 pkginfo(1) 指令的執行結果。

透過程式碼範例 7-1 中所示的執行結果範例，您可以看到該伺服器執行的軟體版本為 4U0.5，因此與此伺服器同屬於某個環境的所有系統也都必須執行 4U0.5 版本。

這是假設 Sun StorEdge SAM-FS 環境已正確設定並可正常運作。

3. 在要設定為 Sun SAM-Remote 用戶端或伺服器的所有主機上發出含 -p 選項的 showrev(1M) 指令。

您必須在要設定為 Sun SAM-Remote 環境組成部分的所有用戶端與伺服器主機上，均安裝相同的修補程式集。例如：

程式碼範例 7-2 使用 showrev(1M)

```
portland# showrev -p | grep SUNWsamfs
Patch: 113546-07 Obsoletes: Requires: Incompatibles: Packages:
SUNWsamfs
portland#
```

4. 檢查 `showrev(1M)` 指令的執行結果。

透過程式碼範例 7-2 中所示的執行結果範例，修補程式 113546-07，因此與此伺服器同屬於某個環境的所有系統也都必須執行修補程式 113546-07。

5. 為該環境中要設定的每個系統重複執行步驟 1、步驟 2、步驟 3 及步驟 4。

6. (選用) 根據需要升級軟體。

如果 `pkginfo(1)` 指令的輸出資訊顯示要納入 Sun SAM-Remote 環境的所有系統，均執行相同的軟體版本與修補程式版本，則可略過此步驟。

如果要設定為 Sun SAM-Remote 環境組成部分的某些系統，執行較舊版本的軟體或修補程式，請將所有系統均升級為最新軟體版本。使用程式碼範例 7-1 做為範例，如果您正在任何系統上執行早於版本 4U0.5 的 Sun StorEdge SAM-FS 版本，您至少必須升級至 4U0.5。

若需有關執行軟體升級的資訊，請參閱「Sun StorEdge SAM-FS Installation and Upgrade Guide」。

▼ 編輯 `mcf` 檔案

1. 從 Sun SAM-Remote 伺服器停止 Sun StorEdge SAM-FS 功能。

a. 執行 `samcmd(1M)` 指令和其 `idle eq` 選項在 Sun StorEdge SAM-FS 軟體的控制下使可移除的多媒體磁碟機閒置。

例如：

```
# samcmd idle eq
```

引數	定義
<code>eq</code>	要定址的可移除的媒體磁碟機之設備序號，如 <code>mcf</code> 檔案中所定義。

為該環境中的每個可移除的媒體磁碟機執行 `samcmd(1M)` 指令。如需更多有關 `samcmd(1M)` 指令的資訊，請參閱 `samcmd(1M)` 線上手冊。

或者，您也可以使用 `samu(1M)` 運算子公用程式來閒置磁碟機。若需關於使用 `samu(1M)` 公用程式的資訊，請參閱「Sun StorEdge QFS 配置和管理指南」。

備註 – 在您執行 `samd stop` 指令之前，應該先閒置 Sun StorEdge SAM-FS 環境中的磁碟機。這將允許歸檔器、分段備份程式及其他程序完成目前工作，同時也允許系統卸載卡匣，並將卡匣插入其儲存插槽。

b. 執行含 `stop` 選項的 `samd(1M)` 指令，以停止執行 `sam-initd` 常駐程式及其子程序。

```
# samd stop
```

`samd(1M)` 指令安裝在 `/opt/SUNWsamfs/sbin` 中。

2. 在用戶端上，使用 `vi(1)` 或其他編輯器編輯現有的 **Sun StorEdge SAM-FS** `/etc/opt/SUNWsamfs/mcf` 檔。

此步驟的目標是要將主機定義為 Sun SAM-Remote 用戶端。程式碼範例 7-3 表示用戶端 `portland` 上已編輯的 `mcf` 檔案。`mcf` 檔案將定義檔案系統，並顯示 Sun SAM-Remote 用戶端 `portland` 正被定義為 Sun SAM-Remote 伺服器 `chicago`。

程式碼範例 7-3 `portland` 上的 `mcf` 檔案

```
# mcf file on portland
#
# Sun StorEdge QFS file system
#
# Equipment      Eq   Eq   Family   Dev  Additional
# Identifier      Ord  Ty   Set      St   Parameters
# =====
samfs1           1   ms   samfs1   on
/dev/dsk/c1t1d0s0  10  md   samfs1   on   /dev/rdisk/c1t1d0s0
/dev/dsk/c1t2d0s0  12  md   samfs1   on   /dev/rdisk/c1t2d0s0
#
# Define Sun SAM-Remote Client portland to Sun SAM-Remote server chicago
#
/etc/opt/SUNWsamfs/rmt200 200  sc   chicagoss on /var/opt/SUNWsamfs/catalog/tcat
/dev/samrd/rd0          201  rd   chicagoss on
/dev/samrd/rd1          202  rd   chicagoss on
```

用戶端上的 `mcf` 項目組成包括 Sun SAM-Remote 用戶端的一行項目，以及要設定的每個裝置之虛擬裝置項目。這些項目遵循 `mcf(4)` 線上手冊中定義的語法。

第一組項目定義 Sun StorEdge QFS 檔案系統。

第二組項目將 Sun SAM-Remote 用戶端 (`portland`) 定義為 Sun SAM-Remote 伺服器 (`chicago`)。第一行定義 Sun SAM-Remote 伺服器本身。這些欄位如下所示：

- [Equipment Identifier] (設備識別碼) 欄位為用戶端配置檔案的路徑名稱，該名稱稍後將在第 151 頁「定義 Sun SAM-Remote 用戶端」中建立。在本範例中，配置檔案命名為 `/etc/opt/SUNWsamfs/rmt200`。
- [Equipment Ordinal] (設備序號) 欄位包含唯一的數目，例如：
 $1 < equipment_ordinal < 65535$ 。此設備序號為 200。
- [Equipment Type] (設備類型) 欄位包含兩個字母的快捷字元 `sc`，用於識別 Sun SAM-Remote 用戶端。

- [Family Set] (系列組) 欄位 (chicagoss) 與伺服器的系列組名稱相同。該名稱是在此特定伺服器上使用的常駐程式之系列組名稱。Sun SAM-Remote 伺服器的每個用戶端可以具有一個伺服器常駐程式。
- [Device State] (裝置狀態) 欄位指定為 on (開啓)。
- [Additional Parameters] (其他參數) 欄位為選用欄位。如上所示，目錄檔路徑可在此處指定。

本 mcf 檔案的最後兩個項目定義 Sun SAM-Remote 虛擬裝置。虛擬裝置可定義 Sun SAM-Remote 伺服器上實際裝置的網路連線。這些項目如下所示：

- [Equipment Identifier] (設備識別碼) 欄位為虛擬裝置所用的 /dev/samrd/rd* 項目之路徑名稱。這些項目在系統重新啓動時建立。您可以定義虛擬裝置的無限制數目。
 - [Equipment Type] (設備類型) 欄位為表示虛擬裝置的兩個字母的快捷字元 rd。
 - [Family Set] (系列組) 欄位 (chicagoss) 與用戶端項目的系列組名稱相同。
3. (選擇性) 在額外的用戶端上，使用 vi(1) 或其他編輯器編輯現有的 Sun StorEdge SAM-FS /etc/opt/SUNWsamfs/mcf 檔。

如果您具有其他用戶端，則必須為額外的每個 Sun SAM-Remote 用戶端完成此步驟。請按照步驟 2 中所列的相同程序進行操作。

在本章的範例中，您必須為用戶端 sacramento 完成相同的配置程序。對於此系統，編輯 mcf 檔案並從 portland 的 mcf 檔案複製行的最後一組至 sacramento 的 mcf 檔案。這些行可將 chicago 主機定義為 Sun SAM-Remote 用戶端。

▼ 定義 Sun SAM-Remote 用戶端

Sun SAM-Remote 用戶端的配置檔案包含單行項目：Sun SAM-Remote 伺服器的名稱。如步驟 2 的第 149 頁「編輯 mcf 檔案」所示，此用戶端配置檔案的完整路徑名稱在用戶端的 mcf 檔案中指定。

1. 在用戶端上，使用 vi(1) 或其他編輯器開啟稱為 Sun SAM-Remote 用戶端配置檔案的檔案。

例如：

```
portland# vi /etc/opt/SUNWsamfs/rmt200
```

2. 編輯該檔案並只包含 Sun SAM-Remote 伺服器名稱。

此步驟產生一個只包含一行的檔案。

程式碼範例 7-4 顯示於 portland 編輯後的用戶端配置檔案。其指向稱為 chicago 的 Sun SAM-Remote 伺服器。

程式碼範例 7-4 用戶端配置檔案

```
portland# cat /etc/opt/SUNWsamfs/rmt200
chicago
```

3. 為每個 Sun SAM-Remote 用戶端重複執行步驟 1 與步驟 2。

如果您具有多個用戶端，請在每個用戶端上建立用戶端檔案。

▼ 在伺服器的 mcf 檔案中定義 Sun SAM-Remote 伺服器

本步驟將在伺服器的 mcf 檔案中定義 Sun SAM-Remote 伺服器。

- 在 Sun SAM-Remote 伺服器上，使用 vi(1) 或其他編輯器編輯現有的 Sun StorEdge SAM-FS /etc/opt/SUNWsamfs/mcf 檔案，將此系統定義為 Sun SAM-Remote 伺服器。

在本步驟的範例中，將編輯伺服器 chicago 上的 mcf 檔案。產生的 mcf 檔案定義 Sun StorEdge QFS 檔案系統並且定義 chicago 做為 Sun SAM-Remote 伺服器。

程式碼範例 7-5 顯示在 chicago 上的 mcf 檔案。

程式碼範例 7-5 chicago 上的 mcf 檔案

```
# mcf file on Sun SAM-Remote server chicago:
# Eq Identifier Eq Ord  Eq Typ Fam Set Dev St  Addl Params
#
samfs1          1   ms   samfs1 on
/dev/dsk/c2t6d0s0 11  md   samfs1 on /dev/rdisk/c2t6d0s0
/dev/dsk/c2t6d0s1 12  md   samfs1 on /dev/rdisk/c2t6d0s1
#
# define a tape library that client portland can use:
/dev/samst/c0t3u0 100 rb   rb100 on /var/opt/SUNWsamfs/catalog/rb100.cat
/dev/rmt/0cbn    101 tp   rb100 on
/dev/rmt/1cbn    102 tp   rb100 on

# Define Sun SAM-Remote server chicago
#
/etc/opt/SUNWsamfs/rmt200 50 ss      chicagoss on
```

這些項目遵循 mcf(4) 中定義的語法，在此範例檔案中，如下所示：

- [Equipment Identifier] (設備識別碼) 欄位為伺服器配置檔案的路徑名稱，該名稱在下列程序中設定。在本範例中，該檔案命名為 /etc/opt/SUNWsamfs/rmt200。
- [Equipment Ordinal] (設備序號) 欄位包含唯一的數目，例如：
 $1 \leq \text{equipment_ordinal} \leq 65535$ 。在此範例中，設備序號為 50。

- [Equipment Type] (設備類型) 欄位包含兩個字母的快捷字元 `ss`，可識別 Sun SAM-Remote 伺服器。
- [Family Set] (系列組) 欄位 (`chicagoss`) 與用戶端上 `mcf` 檔案所用的系列組名稱一致。請注意，您可以為 Sun SAM-Remote 伺服器定義多個伺服器常駐程式。
- [Device State] (裝置狀態) 欄位為選用欄位，在本範例中指定為 `on` (開啓)。
- [Additional Parameters] (其他參數) 欄位為選用欄位。

備註 – 您必須至少有一個於 Sun SAM-Remote 伺服器的 `mcf` 檔案中已配置之 Sun StorEdge SAM-FS 檔案系統。

▼ 建立 Sun SAM-Remote 伺服器配置檔案

Sun SAM-Remote 伺服器配置檔案可定義磁碟緩衝區特性，以及用於每個用戶端的媒體。您可以為每個伺服器常駐程式設定十個用戶端。如果要支援更多的用戶端，則必須如之前在第 149 頁「編輯 `mcf` 檔案」(步驟 2) 以及第 151 頁「定義 Sun SAM-Remote 用戶端」中所述，設定另一個 Sun SAM-Remote 伺服器常駐程式。

1. 在伺服器上，使用 `vi(1)` 或其他編輯器開啟稱為 **Sun SAM-Remote 伺服器配置檔案** 的檔案。
2. 編寫伺服器配置檔案。

程式碼範例 7-6 顯示伺服器配置範例檔案 `/etc/opt/SUNWsamfs/rmt200`，該檔案位於 Sun SAM-Remote 伺服器 `chicago` 上。此檔案將定義用戶端 `portland` 與 `sacramento`。

程式碼範例 7-6 伺服器配置檔案 `rmt200`

```
#
# Sun SAM-Remote server config file /etc/opt/SUNWsamfs/rmt200
#
portland
    media
    100 at (000031|000032|000034|000035|000037|000038)
    endmedia
#
sacramento
    media
    100 at (000131|000132|000134|000135|000137|000138)
    endmedia
```

如程式碼範例 7-6 所示，伺服器配置檔案組成包括每個用戶端的多行項目。井字號 (#) 表示註解行。註解行右側的內容均可忽略。

程式碼範例 7-7 顯示 Sun SAM-Remote 伺服器配置檔案格式。

程式碼範例 7-7 伺服器配置檔案格式

```
用戶端名稱
  [ 參數 1 ]
  媒體
    eq 媒體類型 regex
    [ eq 媒體類型 regex ]
    [ . . . ]
  endmedia
```

下列步驟將說明如何編寫伺服器配置檔案。

a. 編寫用戶端名稱欄位。

用戶端名稱將定義叫用此 Sun SAM-Remote 常駐程式所適用的每個用戶端網路名稱。用戶端名稱的第一個字元必須是該行的第一個字元。用戶端名稱可以指定為網路名稱、IP 位址或完整有效的網域名稱。

自用戶端名稱之後到下一個用戶端定義之前的參數 (如果已指定的話) 與媒體規格，均專屬於此用戶端。參數與媒體定義必須縮排幾個空白或 tab 字元。

b. (選用) 編寫參數欄位。

參數行以關鍵字 = 值對表示。您可以使用參數欄位指定網路區塊大小。

net_block_size (網路區塊大小) 參數將指定此用戶端通訊端所用的網路區塊大小 (以 KB 為單位)。此參數的格式如下所示：

```
net_blk_size= 大小
```

對於大小，指定從 $4 \leq \text{大小} \leq 64$ 的整數。預設為 4，指定 4096 位元組。

參數行必須定義縮排幾個空白或 tab 字元。

c. 編寫 media 與 endmedia 的關鍵字欄位。

伺服器配置檔案中需要 media 與 endmedia 關鍵字。這些關鍵字可定義用戶端使用的媒體歸檔磁碟區。這些媒體關聯指定如下：

程式碼範例 7-8 伺服器配置檔案中的媒體規格

```
媒體
  eq 媒體類型 (regex)
  [ eq 媒體類型 (regex) ]
  [ . . . ]
endmedia
```

`media` 與 `endmedia` 關鍵字將限定 Sun SAM-Remote 伺服器配置檔案的媒體定義區域。`eq` 媒體類型 `regex` 行為媒體定義行。媒體必須定義縮排幾個空白或 `tab` 字元。`regex` 資料必須加上括號。

媒體類型規格的要素如下所示：

引數	定義
<code>eq</code>	程式庫的設備序號。 含混合媒體的網路連接程式庫，可以具有多個 <code>eq</code> 媒體類型 <code>regex</code> 行。因此，請為每個媒體類型指定不同的 <code>eq</code> 媒體類型 <code>regex</code> 行。
媒體類型	兩個字元的特定媒體類型。請注意， <code>mcf</code> 檔案中有效的一般媒體類型規格不適用於媒體類型規格。此規格必須用於特定媒體類型 (例如： <code>1t</code>)。若需有關有效媒體類型的資訊，請參閱 <code>mcf(4)</code> 線上手冊。 如果您具有含多種媒體類型的網路連接程式庫，請指定多個媒體定義行。
<code>regex</code>	檔案所要歸檔的卡匣磁碟區序列名稱 (VSN)。每個 VSN 指定必須表示為延伸式常規表示式，且 VSN 必須加上括號。若需有關延伸正規式的資訊，請參閱 <code>egrep(1)</code> 線上手冊。 您可以為每個媒體類型指定多個媒體定義行，以便定義媒體時比較靈活。例如：下列定義就是一個有效的媒體類型定義： <pre>media 100 1t (VSN1) 100 1t (VSN2) endmedia</pre> 若需有關正規式的資訊，請參閱 <code>regcomp(3C)</code> 線上手冊。

備註 – 請禁止多個用戶端使用相同的實體媒體卡匣。此外，如果 Sun SAM-Remote 伺服器在 Sun SAM-Remote 環境以外具有自己的檔案系統，建議此用戶端與伺服器不要使用同一個卡匣。

▼ 啟用歸檔

下列步驟將啟用歸檔並完成配置程序。

1. 確認用戶端上的 `archiver.cmd` 檔案。

依您的配置而定，您可能需要執行下列工作：

- 確定伺服器配置檔案中定義的 VSN 已指派給 `archiver.cmd` 檔案中的正確歸檔組。
- 如果下列指令適用於要歸檔至與 Sun SAM-Remote 伺服器連接的程式庫之歸檔組，請從 Sun SAM-Remote 用戶端上的 `archiver.cmd` 檔案中移除這些指令：
 - `-tapenonstop`
 - `-offline_copy direct`

2. 執行 **samd(1M)** 指令與其 **start** 選項以啟動伺服器與用戶端上的 **Sun StorEdge SAM-FS** 程序。

為確保系統讀取伺服器與用戶端上的新配置檔案，您必須啟動或重新啟動 Sun StorEdge SAM-FS 軟體。

在用戶端與伺服器上輸入下列指令：

```
server# samd start
```

若需關於啟動和重新啟動 Sun StorEdge SAM-FS 的完整指示，請參閱「Sun StorEdge SAM-FS Installation and Upgrade Guide」。

3. 在伺服器與用戶端上呼叫 **samu(1M)**。

此步驟的目的是確認主機之間的連線。使用 **samu(1M)** 公用程式的 **s** 與 **R** 畫面，顯示 Sun SAM-Remote 連線狀態。若需更多有關 **samu(1M)** 的資訊，請參閱 **samu(1M)** 線上手冊或「Sun StorEdge QFS 配置和管理指南」。

程式碼範例 7-9 顯示 Sun SAM-Remote 用戶端 **portland** 上的 **samu(1M)** 狀態 **s** 畫面。請注意裝置類型 **sc**，其表示 Sun SAM-Remote 用戶端。該行下面的訊息表示已建立伺服器 **chicago** 之間的連線。

程式碼範例 7-9 用戶端 **samu(1M)** **s** 畫面

```
Device status          samu    4.0.5 Wed May 02 14:44:44
License: License never expires.

ty   eq state   device_name          fs status   pos
ms   1 on      samfs1               1 m-----

md   10 on     /dev/dsk/c1t1d0s0    1  -----

md   12 on     /dev/dsk/c1t2d0s0    1  -----

s9   35 on     /dev/samst/c0t5u0    35 m-----r
     move complete

lt   36 on     /dev/rmt/0cbn        35 -----p
     empty

lt   37 on     /dev/rmt/1cbn        35 -----p
     empty

lt   38 on     /dev/rmt/2cbn        35 --l-----r
     idle

lt   39 on     /dev/rmt/3cbn        35 --l-----r
     idle

sc   200 on    /etc/opt/SUNWsamfs/rmt200    200 -----r
     server chicago connected

rd   201 on    /dev/samrd/rd0       200 -----r
```

程式碼範例 7-9 用戶端 samu(1M) s 畫面 (續)

```
rd    202 on    /dev/samrd/rd1          200 -----r
hy    203 on    historian                203 -----
```

程式碼範例 7-10 顯示 Sun SAM-Remote 伺服器 chicago 上的 samu(1M) 狀態 s 畫面。請注意裝置類型 ss，其表示 Sun SAM-Remote 伺服器。此畫面指明此系統為 Sun SAM-Remote 伺服器。

程式碼範例 7-10 伺服器 samu(1M) chicago 上的 s 畫面

```
Device status          samu    4.0.5 Tue Apr 24 14:49:43
License: License never expires.

ty    eq state  device_name          fs status  pos
ms    1 on     samfs1              1 m-----

md    11 on    /dev/dsk/c2t6d0s0    1 -----

md    12 on    /dev/dsk/c2t6d0s1    1 -----

ss    50 on    /etc/opt/SUNWsamfs/rmt200  50 -----r

sl    100 on   /dev/samst/c0t3u0    100 m-----r

at    101 on   /dev/rmt/0cbn        100 -----p
      initializing

at    102 on   /dev/rmt/1cbn        100 -----p
      initializing

hy    103 on   historian              103 -----
```

程式碼範例 7-11 顯示 Sun SAM-Remote 伺服器 chicago 上的 samu(1M) Sun SAM-Remote R 畫面。

程式碼範例 7-11 伺服器 samu(1M) chicago 上的 R 畫面

```
Remote server eq: 50          addr: 00001ca0 4.0.5 Wed May 02
14:55:37

message:

Client: portland
client index - 0
```

程式碼範例 7-11 伺服器 samu(1M) chicago 上的 R 畫面 (續)

```
network block size - 4096
max file size - 0                      flags - c0000000
min file size - 8
```

如果您具有多個 Sun SAM-Remote 用戶端，則可按下 CONTROL-f 按鍵組合以捲動用戶端。

在程式碼範例 7-11 中，連接用戶端的命名為 portland。client index (用戶端索引) 欄位表示此用戶端是此伺服器常駐程式可能定義的 0 至 9 個用戶端中的第 0 個用戶端。所列的最大檔案大小、最小檔案大小及網路區塊大小均以位元組為單位。旗標表示連線狀態，如下所示：

表 7-1 samu(1M) R 畫面旗標

旗標	意義
0x00000000	無連線。
0xc0000000	已建立連線。

4. 在伺服器上，使用 samu(1M) 公用程式以確定目錄檔在用戶端上可以使用。

對於每個用戶端，使用 samu(1M) 公用程式的 v 畫面顯示 VSN，您應該能夠檢視適用於該用戶端的 Sun SAM-Remote 目錄檔。透過 samu(1M) 輸入下列指令：

```
:v eq
```

eq 必須是 Sun SAM-Remote 用戶端常駐程式的設備序號，如 mcf 檔案中所定義。

程式碼範例 7-12 顯示 chicago 上的 samu(1M) 畫面。在 chicago 上指定 :v 200 即可顯示此畫面。此畫面顯示了 portland 可從 chicago 存取的磁碟區。

程式碼範例 7-12 自 chicago 檢視的可用磁碟區

```
Robot VSN catalog by slot : eq 200 samu 4.0.5 Wed May 02 15:24:13
count 32
slot      access      time  count use flags      ty vsn
-----
1         2003/01/02 10:40    0  0% -il-o-b-R-U-  at 000032
2         2003/01/02 11:41    0  0% -il-o-b-R---  at 000034
3         2003/01/02 12:42  170  91% -il-o-b-----  at 000035
4         2003/01/02 13:43   20   7% -il-o-b-----  at 000037
5         2003/01/02 14:44    0  0% -il-o-b-----  at 000038
6         2003/01/02 13:41    0  0% -il-o-b-----  at 000031
```

5. 在用戶端上，發出含 `-A` 選項的 `archiver(1M)` 指令。

在此步驟中，您將確認檔案是否從用戶端歸檔至伺服器。使用 `archiver(1M)` 指令及其 `-A` 選項即可執行此操作。此選項允許從歸檔器寫入清單，此清單包含伺服器上的 VSN。有關此指令的資訊，請參閱 `archiver(1M)` 線上手冊。

如果檔案未歸檔，請參閱「Sun StorEdge SAM-FS Troubleshooting Guide」以取得有關如何對歸檔器進行疑難排解的資訊。

使用 Sun SAM-Remote 軟體進行回收

本節包含有關使用 Sun SAM-Remote 進行回收的資訊。Sun Microsystems 建議您最好只在本章所述非常特定的情況下，在 Sun SAM-Remote 環境中進行回收。請務必遵循本章中所述的回收限制，否則會導致資料遺失。Sun StorEdge SAM-FS 軟體中沒有這些限制的強制。

由於回收程序涉及釋放卡匣上的空間以取得更多資料，因此，如果回收程序設定不正確，回收器可能會損毀歸檔匣上的重要資料。



注意 – 在 Sun SAM-Remote 環境下使用回收器必須完全瞭解回收器的每個步驟。以錯誤的順序或在錯誤的系統上執行指令，可能會導致資料遺失無法復原。在執行任何指令之前，請確定已分析指令的動作，例如：`tplabel(1M)`，這會刪除 Sun SAM-Remote 用戶端或 Sun SAM-Remote 伺服器上的資料。

切勿在 Sun SAM-Remote 伺服器與 Sun SAM-Remote 用戶端上同時進行回收活動。否則，可能會導致意外重新標記卡匣與資料遺失無法復原。

您無法回收包含可移除的媒體檔案的卡匣。

在 Sun SAM-Remote 用戶端與伺服器環境中，用戶端與伺服器無法辨識彼此的檔案系統、資料檔案及 `inode` 檔案。伺服器與用戶端各自都必須具有特定卡匣組的專屬使用權，彼此不會使用對方的卡匣。您可以在 Sun SAM-Remote 伺服器的 `/etc/opt/SUNWsamfs/recycler.cmd` 檔案中建立 `no_recycle` 清單，來防止意外回收 Sun SAM-Remote 用戶端使用的 VSN。但是，請小心使用 `no_recycle` 清單中磁碟區上 `chmed(1M)` 指令的 `+c` 選項。使用此指令設定磁碟區上的回收旗標 (`+c`) 時，該動作會覆寫 `/etc/opt/SUNWsamfs/recycler.cmd` 檔案中的 `no_recycle` 清單。

請勿嘗試在同一天回收 Sun SAM-Remote 伺服器與 Sun SAM-Remote 用戶端上的磁碟區。

唯有符合下列條件時，才能在 Sun SAM-Remote 環境下進行回收：

- 系統中的每個 VSN 只能由一個用戶端系統或伺服器使用。任何 VSN 上的檔案不能由多個系統使用。
- 除了包含該用戶端歸檔影像的 VSN 之外，Sun SAM-Remote 用戶端沒有任何 VSN 的目錄項目。伺服器配置檔案的媒體定義行中的 *regex* (*eq media_type regex* 行) 必須同意用戶端目錄中指定的磁碟區。此外，用戶端目錄中的 *regex* 規格無法指定相同的磁碟區。
- 根據歸檔組進行歸檔。使用 Sun SAM-Remote 時，必須根據歸檔組 (而不是程式庫) 進行回收。

本章說明使用 Sun SAM-Remote 用戶端與伺服器啓用回收的兩種方法。方法如下所示：

- 第 160 頁 「於 Sun SAM-Remote 環境中進行回收 — 方法 1」
- 第 183 頁 「於 Sun SAM-Remote 環境進行回收 — 方法 2」

於 Sun SAM-Remote 環境中進行回收 — 方法 1

本節程序說明在 Sun SAM-Remote 環境中啓用回收的一種方法。本章節的範例環境中，伺服器名稱爲 *sky*，用戶端名稱爲 *zeke*。本程序顯示了如何設定 Sun SAM-Remote 在兩個不同程式庫的卡匣上建立檔案歸檔備份。使用屬於 *zeke* 的 StorageTek 程式庫將可寫入歸檔備份 1。使用連接至 *sky* 的 ADIC 程式庫將可遠端寫入歸檔備份 2。下列章節顯示了以上兩個系統的相關檔案。



注意 – 請在完成本程序全部步驟並在測試配置以查看回收進行是否正確後，才在 Sun SAM-Remote 環境下使用回收器。

伺服器 *sky* 的配置檔案

伺服器在 *mcf* 檔案及其伺服器配置檔案中必須具有 Sun SAM-Remote 配置資訊。下列代碼範例顯示了這些檔案。

程式碼範例 7-13 顯示伺服器 *sky* 上的 *mcf* 檔案。

程式碼範例 7-13 伺服器 *sky* 上的 *mcf* 檔案

```
# This is the mcf file for the server (sky).
# The server parameters file (rmt1000) points
#   back to the correct automated library's equipment number
#   (70) for the ADIC Scalar 1000.
#
samfs1          100   ma   samfs1   on
/dev/dsk/c0t0d0s5 110   mm   samfs1   on   /dev/rdisk/c0t0d0s5
/dev/dsk/c3t2d0s3 120   mr   samfs1   on   /dev/rdisk/c3t2d0s3
```

程式碼範例 7-13 伺服器 sky 上的 mcf 檔案 (續)

```
/dev/dsk/c3t2d0s4    121    mr    samfs1    on    /dev/rdisk/c3t2d0s4

samfs2                139    ma    samfs2    on
/dev/dsk/c3t4d0s3    140    mm    samfs2    on    /dev/rdisk/c3t4d0s3
/dev/dsk/c3t4d0s4    141    mr    samfs2    on    /dev/rdisk/c3t4d0s4

# ADIC Scalar 1000
/dev/samst/c0t0u0 70 rb adic1 - /var/opt/SUNWsamfs/catalog/adic1
/dev/rmt/0bn      71    at    adic1    on
/dev/rmt/1bn      72    at    adic1    on
/dev/rmt/2bn      73    at    adic1    on
/dev/rmt/3bn      74    at    adic1    on
/dev/rmt/4bn      75    at    adic1    on
/dev/rmt/5bn      76    at    adic1    on
/dev/rmt/11bn     77    at    adic1    on
/dev/rmt/10bn     78    at    adic1    on
/dev/rmt/9bn      79    at    adic1    on
/dev/rmt/8bn      80    at    adic1    on
/dev/rmt/7bn      81    at    adic1    on
/dev/rmt/6bn      82    at    adic1    on

# Define Sun SAM-Remote server skyrs
/etc/opt/SUNWsamfs/rmt1000 1000 ss skyrs on
```

程式碼範例 7-14 顯示伺服器 sky 上的伺服器配置檔案。

程式碼範例 7-14 伺服器 sky 上的伺服器配置檔案

```
# Server configuration file /etc/opt/SUNWsamfs/rmt1000 on sky.
# The eq of the automated library MUST match the eq of the
# automated library that you want to use in the mcf file.

zeke
  媒體
  70 at 00002[0-9]
endmedia
```

用戶端 zeke 的配置檔案

用戶端在 mcf 檔案及其用戶端配置檔案中必須具有 Sun SAM-Remote 配置資訊。下列代碼範例顯示了這些檔案。

程式碼範例 7-15 顯示用戶端 zeke 上的 mcf 檔案。

程式碼範例 7-15 用戶端 zeke 上的 mcf 檔案

```
# mcf file for client (zeke)
#
samfs1          10  ms  samfs1  on
/dev/dsk/c1t3d0s0  11  md  samfs1  on  /dev/rdisk/c1t3d0s0
/dev/dsk/c1t3d0s1  12  md  samfs1  on  /dev/rdisk/c1t3d0s1
/dev/dsk/c1t3d0s3  13  md  samfs1  on  /dev/rdisk/c1t3d0s3

# Define a StorageTek L20 with 1 drive and 20 slots (including cap)
/dev/samst/c0t2u0  50  rb  stk_l20  on  /var/opt/SUNWsamfs/catalog/L20_cat
/dev/rmt/0hbn     51  lt  stk_l20  on

# Define zeke as a Sun SAM-Remote client using sky as the server
/etc/opt/SUNWsamfs/sky 200  sc  skyrs    on  /var/opt/SUNWsamfs/catalog/sky_cat
/dev/samrd/rd0      201  rd  skyrs    on
/dev/samrd/rd1      202  rd  skyrs    on
/dev/samrd/rd2      203  rd  skyrs    on
/dev/samrd/rd3      204  rd  skyrs    on
```

程式碼範例 7-16 顯示用戶端 zeke 上的用戶端配置檔案。

程式碼範例 7-16 用戶端 zeke 上的用戶端配置檔案

```
# cat /etc/opt/SUNWsamfs/sky
# File /etc/opt/SUNWsamfs/sky on Sun SAM-Remote client zeke:
sky
```

▼ 配置回收 — 方法 1

下列程序顯示了如何設定回收程序。本程序包含歸檔與回收的測試。由於需要測試，因此本程序可能需要一兩天才能完成，視歸檔與回收檔案有多頻繁而定。

備註 – 請勿在伺服器上使用 `chmed(1M)` 指令來為用戶端 VSN 設定回收旗標 (+c)。該動作會覆寫伺服器 `/etc/opt/SUNWsamfs/recycler.cmd` 檔案中的 `no_recycle` 清單。

1. 閱讀第 129 頁「回收」中有關回收器的資訊。

在 Sun SAM-Remote 環境下使用回收器必須完全瞭解回收程序中的步驟。如果您不熟悉回收程序，請立即花些時間加以瞭解。

2. 確認 Sun SAM-Remote 用戶端與伺服器已正確配置且會進行歸檔。

如需更多有關配置與驗證 Sun SAM-Remote 環境的資訊，請參閱第 146 頁「設定 Sun SAM-Remote 軟體」，其中包含有關設定 Sun SAM-Remote 用戶端與伺服器的詳細資訊。該程序包含確定進行歸檔的步驟。

3. 編輯用戶端系統上的 archiver.cmd 檔案，然後新增回收指令。

在本範例中，根據歸檔組（而不是程式庫）進行回收。指定由歸檔組進行回收的指令必須出現在 archiver.cmd 檔案中。

程式碼範例 7-17 顯示用戶端 zeke 上的 archiver.cmd 檔案。此檔案已編輯以與回收器進行通訊。

程式碼範例 7-17 用戶端 zeke 上的 archiver.cmd 檔案

```
# This is file /etc/opt/SUNWsamfs/archiver.cmd
#   on Sun SAM-Remote client zeke.
#
# wait

logfile = /var/opt/SUNWsamfs/archiver/archiver.log
trace = /var/opt/SUNWsamfs/trace/archiver all

interval = 1m

no_archive tmp
no_archive .

archmax = lt 2G
archmax = at 5G

drives = skyr4 # use up to four drives for remote archiving.

fs = samfs1
    1 4h
archiveset testdir0
    1 1m
    2 1m
defaultset .
    1 1m
    2 1m

params
```

程式碼範例 7-17 用戶端 zeke 上的 archiver.cmd 檔案 (續)

```
# Start with mingain high to reduce workload.
# If you need more recycling, reduce mingain.
# If too much recycling, increase High Water Mark.
archiveset.1 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
archiveset.1 -recycle_ignore
defaultset.1 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
defaultset.1 -recycle_ignore

# Remote directives.
# Use up to three drives per archive set.
# Load will split to two drives at 100m, to three drives at 150m.
archiveset.2 -drives 3 -drivemin 50m
defaultset.2 -drives 3 -drivemin 50m

# Remote directives.
# Start with mingain high to reduce workload.
# If you need more recycling, reduce mingain.
# If too much recycling, increase High Water Mark.
archiveset.2 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
archiveset.2 -recycle_ignore
defaultset.2 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
defaultset.2 -recycle_ignore
endparams

vsns
samfs1.1      lt 000173      # local copy.
archiveset.1 lt ^CEL        # local copy.
archiveset.2 at 00002[0-4]  # remote copy, sky ait-2
                                     # tapes 20 through 24.
defaultset.1 lt ^CSM        # local copy.
defaultset.2 at 00002[5-9]  # remote copy, sky ait-2
                                     # tapes 25 through 29.
endvsns
```

程式碼範例 7-17 中所示的指令執行如下：

- `-recycle_hwm` 指令為歸檔組設定了程式庫的高值參數。VSN 的使用率超過此百分比時，會開始回收歸檔組。
- `-recycle_ignore` 指令只是暫時插入。此指令可防止在設定與測試環境之前進行回收。您可以在稍後的步驟中移除此指令。
- 將 `-recycle_mingain` 指令設定為高，以限制重新取得空間所需的工作量。也就是說，此指令設定為高可確保效率。
- `-recycle_vsncount 1` 指令可防止回收超過系統負荷。此指令指定回收器一次清空一個 VSN。清空第一個 VSN 時，會選擇另一個要開始清空的 VSN。因此，此時佇列中有一個要重新標記的 VSN 及一個要清空的 VSN。

4. 編輯用戶端上的 `recycler.cmd` 檔案，並指定記錄檔以接收回收記錄輸出。

已編輯下列用戶端 `zeke` 上的 `recycler.cmd` 檔案，以指定回收器記錄檔：

程式碼範例 7-18 用戶端 `zeke` 上的 `recycler.cmd` 檔案

```
#
# This is the /etc/opt/SUNWsamfs/recycler.cmd file
# on client zeke.
#
logfile = /var/opt/SUNWsamfs/log/recycler
```

5. 確認已寫入伺服器上的 `archiver.cmd` 檔案，以按歸檔組指定回收。

使用 Sun SAM-Remote 時，您必須指定根據歸檔組 (而不是程式庫) 進行回收。指定由歸檔組進行回收的指令必須出現在 `archiver.cmd` 檔案中。

程式碼範例 7-19 顯示 `sky` 伺服器上的 `archiver.cmd` 檔案。此檔案指定按歸檔組進行歸檔。

程式碼範例 7-19 伺服器 `sky` 上的 `archiver.cmd` 檔案

```
# This is the archiver.cmd for the server (sky).
#
# Number of drives: 10
# Number of Mounted Filesystems: 1
# Number of Tests per Filesystem: 1
# Number of Archive Copies per Test: 2

#wait
#trace = /var/opt/SUNWsamfs/trace/archiver all

logfile = /var/opt/SUNWsamfs/log/archiver
interval = 1m
no_archive .
archmax = at 5G
drives = adic1 6

fs = samfs1
    1 4h
testset testdir0
    1 1m
    2 1m
allsam1 .
    1 1m
```

程式碼範例 7-19 伺服器 sky 上的 archiver.cmd 檔案 (續)

```
2 1m

params
allsam1.1 -drives 4 -drivemin 50m
allsam1.1 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
allsam1.1 -recycle_ignore
allsam1.2 -drives 4 -drivemin 50m
allsam1.2 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
allsam1.2 -recycle_ignore
testset.1 -drives 4 -drivemin 50m
testset.1 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
testset.1 -recycle_ignore
testset.2 -drives 4 -drivemin 50m
testset.2 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
testset.2 -recycle_ignore
endparams

vsns
samfs1.1 at 000000
allsam1.1 at 00000[1-5] # vsns 1 through 5.
allsam1.2 at 00000[6-9] # vsns 6 through 9.
testset.1 at 00001[0,4] # vsns 10 and 14.
testset.2 at 00001[5,9] # vsns 15 and 19.
endvsns
```

6. 編輯伺服器上的 recycler.cmd 檔案。

請使用編輯器修改檔案以指定下列項目：

- 從回收器接收執行結果的回收器記錄檔。
- Sun SAM-Remote 用戶端 VSN 的 no_recycle 指令。Sun SAM-Remote 用戶端設定為將其備份 2 歸檔備份寫入 Sun SAM-Remote 伺服器程式庫中的卡匣。要防止用於歸檔的 Sun SAM-Remote 用戶端使用的 VSN 被 Sun SAM-Remote 伺服器回收，必須要有 no_recycle 指令。

已編輯下列伺服器 sky 上的 recycler.cmd 檔案，以指定回收器記錄檔：

程式碼範例 7-20 伺服器 sky 上的 recycler.cmd 檔案

```
#
# This is the /etc/opt/SUNWsamfs/recycler.cmd file
# on Sun SAM-Remote server sky.
#
logfile = /var/opt/SUNWsamfs/recycler/recycler.log
```

程式碼範例 7-20 伺服器 sky 上的 recycler.cmd 檔案 (續)

```
adicl -ignore
no_recycle at 00002[0-9] # Prevents VSNs assigned to zeke from
                          # being recycled.
```

7. 使用 sam-recycler(1M) 指令在 Sun SAM-Remote 用戶端上測試回收器。

在 Sun SAM-Remote 用戶端系統上執行回收器。這項測試可查看回收器是否已正確地確認配置檔案中指定的裝置與 VSN。此測試是重要的，因為如果回收器偵測到它正在執行的系統在任何該系統的目錄 (包括記錄器目錄) 中列出的特定 VSN 上沒有歸檔影像，recycler.sh 指令碼可能會需求標記卡匣。標記卡匣會損毀卡匣上的所有資料。Sun SAM-Remote 用戶端與 Sun StorEdge SAM-FS 伺服器之間沒有通訊方式可通知對方有關歸檔副本的存在。所有此類資訊都是由本機 Sun StorEdge SAM-FS 檔案系統提供。

例如，您可以使用下列指令執行回收器的初始測試：

```
zeke# sam-recycler -dvx
```

回收器將執行並將其活動記錄至回收器記錄檔。回收器記錄檔將在 recycler.cmd 檔案中定義。若需更多有關 sam-recycler(1M) 指令的資訊，請參閱 sam-recycler(1M) 線上手冊。

8. 檢查回收器記錄檔。

您將尋找下列訊息：

```
Recycling is ignored on this archive set.
```

程式碼範例 7-21 顯示範例記錄檔。

程式碼範例 7-21 用戶端 zeke 上的回收器記錄檔

```
# recycler.log from client zeke.

===== Recycler begins at Mon Jun  4 09:49:41 2001 =====
Initial 7 catalogs:

0  Family: stk_l20                Path: /var/opt/SUNWsamfs/catalog/L20_cat
   Vendor: STK                    Product: L20
   SLOT                           ty    capacity      space vsn
   0                               lt    33.0G         33.0G 000173
   1                               lt    32.8G         44.1M CEL170
   2                               lt    33.0G         33.0G CEL139
   4                               lt    32.8G         16.8G CFC504
```

程式碼範例 7-21 用戶端 zeke 上的回收器記錄檔 (續)

```

5          lt          33.0G          33.0G CFC503
6          lt          32.9G          0      CSM689
7          lt          32.9G          19.6G CSM690
8          lt          33.0G          33.0G CSM691
9          lt          33.0G          33.0G CSM692
10         lt          10.0G          10.0G CLN018
11         lt          33.0G          33.0G 000766
Total Capacity: 339.2G bytes, Total Space Available: 244.3G bytes
Volume utilization 27%, high 95% VSN_min 50%
Recycling is ignored on this robot.

1 Family: skyrs          Path: /var/opt/SUNWsamfs/catalog/sky_cat
Vendor: (NULL)          Product: (NULL)
SLOT          ty          capacity          space vsn
0             at          48.5G          23.3G 000020
1             at          23.8G          23.8G 000021
2             at          48.5G          48.5G 000022
3             at          48.5G          48.5G 000023
4             at          48.5G          48.5G 000024
5             at          48.5G          2.6G 000025
6             at          48.5G          361.4k 000026
7             at          48.5G          48.5G 000027
8             at          48.5G          48.5G 000028
9             at          48.5G          0      000029
Total Capacity: 460.8G bytes, Total Space Available: 292.5G bytes
Volume utilization 36%, high 95% VSN_min 50%
Recycling is ignored on this robot.

2 Family: hy          Path: /var/opt/SUNWsamfs/catalog/historian
Vendor: Sun SAM-FS    Product: Historian
SLOT          ty          capacity          space vsn
(no VSNs in this media changer)
Total Capacity: 0 bytes, Total Space Available: 0 bytes
Volume utilization 0%, high 95% VSN_min 50%
Recycling is ignored on this robot.

3 Family: defaultset.1 Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS    Product: Archive set

```

程式碼範例 7-21 用戶端 zeke 上的回收器記錄檔 (續)

```

SLOT          ty      capacity      space vsn
  0           lt       33.0G        33.0G 000766
  1           lt       33.0G        33.0G 000173
  2           lt       32.9G         0   CSM689
  3           lt       32.9G       19.6G CSM690
  4           lt       33.0G        33.0G CSM691
  5           lt       33.0G        33.0G CSM692
Total Capacity: 197.6G bytes, Total Space Available: 151.5G bytes
Volume utilization 23%, high 60% VSN_min 90%
Recycling is ignored on this archive set.

4 Family: defaultset.2          Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS              Product: Archive set
SLOT          ty      capacity      space vsn
  0           lt       32.9G         0   CSM689
  1           at       48.5G       23.3G 000020
  2           at       23.8G       23.8G 000021
  3           at       48.5G         2.6G 000025
  4           at       48.5G      361.4k 000026
  5           at       48.5G       48.5G 000027
  6           at       48.5G       48.5G 000028
  7           at       48.5G         0   000029
Total Capacity: 348.0G bytes, Total Space Available: 146.8G bytes
Volume utilization 57%, high 60% VSN_min 90%
Recycling is ignored on this archive set.

5 Family: archiveset.1          Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS              Product: Archive set
SLOT          ty      capacity      space vsn
  0           lt       32.8G       44.1M CEL170
  1           lt       32.8G       16.8G CFC504
  2           lt       33.0G       33.0G CFC503
Total Capacity: 98.6G bytes, Total Space Available: 49.8G bytes
Volume utilization 49%, high 60% VSN_min 90%
Recycling is ignored on this archive set.

6 Family: archiveset.2          Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS              Product: Archive set

```

程式碼範例 7-21 用戶端 zeke 上的回收器記錄檔 (續)

SLOT	ty	capacity	space	vsn
0	at	48.5G	23.3G	000020
1	at	23.8G	23.8G	000021
2	at	48.5G	48.5G	000022
3	at	48.5G	48.5G	000023
4	at	48.5G	48.5G	000024
Total Capacity: 218.0G bytes, Total Space Available: 192.8G bytes				
Volume utilization 11%, high 60% VSN_min 90%				
Recycling is ignored on this archive set.				
21 VSNs:				
---Archives---				
-----Status-----	Count	Bytes	Use Obsolete Free	defaultset.1 Library:Type:VSN
in multiple sets	0	0	0 100 0	stk_l20:lt:CSM689
partially full	111	2.8G	8 31 61	stk_l20:lt:CSM690
empty VSN	0	0	0 0 100	stk_l20:lt:000173
empty VSN	0	0	0 0 100	stk_l20:lt:CSM691
empty VSN	0	0	0 0 100	stk_l20:lt:CSM692
empty VSN	0	0	0 0 100	stk_l20:lt:000766
---Archives---				
-----Status-----	Count	Bytes	Use Obsolete Free	defaultset.2 Library:Type:VSN
no-data VSN	0	0	0 100 0	skyrs:at:000029
no-data VSN	0	0	0 99 1	skyrs:at:000026
partially full	111	2.8G	6 88 6	skyrs:at:000025
empty VSN	0	0	0 0 100	skyrs:at:000028
empty VSN	0	0	0 0 100	skyrs:at:000027
---Archives---				
-----Status-----	Count	Bytes	Use Obsolete Free	archiveset.1 Library:Type:VSN
no-data VSN	0	0	0 99 1	stk_l20:lt:CEL170
partially full	677	2.3G	8 40 52	stk_l20:lt:CFC504
empty VSN	0	0	0 0 100	stk_l20:lt:CFC503
---Archives---				
-----Status-----	Count	Bytes	Use Obsolete Free	archiveset.2 Library:Type:VSN
in multiple sets	0	0	0 51 49	skyrs:at:000020
empty VSN	0	0	0 0 100	skyrs:at:000022
empty VSN	0	0	0 0 100	skyrs:at:000023
empty VSN	0	0	0 0 100	skyrs:at:000024
in multiple sets	0	0	0 0 100	skyrs:at:000021
---Archives---				
-----Status-----	Count	Bytes	Use Obsolete Free	stk_l20 Library:Type:VSN

程式碼範例 7-21 用戶端 zeke 上的回收器記錄檔 (續)

```
empty VSN          0          0          0          0          100 stk_l20:lt:CLN018
partially full    13        80.3k          0          0          100 stk_l20:lt:CEL139

Recycler finished.

===== Recycler ends at Mon Jun  4 09:49:53 2001 =====
```

9. 從 Sun SAM-Remote 伺服器中發出 `sam-recycler(1M)` 指令以測試回收器。

請確定回收器沒有回收為 Sun SAM-Remote 用戶端保留的任何 VSN。

例如：

```
zeke# sam-recycler -dvx
```

上述指令將執行回收器並將其活動寫入回收器記錄檔。若需更多有關 `sam-recycler(1M)` 指令的資訊，請參閱 `sam-recycler(1M)` 線上手冊。

程式碼範例 7-22 顯示範例回收器記錄檔。

程式碼範例 7-22 回收器記錄檔

```
# recycler.log file from server sky.

===== Recycler begins at Mon Jun  4 09:50:44 2001 =====
Initial 6 catalogs:

0 Family: adicl          Path: /var/opt/SUNWsamfs/catalog/adicl
  Vendor: ADIC          Product: Scalar 1000
  SLOT                 ty      capacity      space vsn
    0                   at      1.3G          1.2G 000001
    1                   at      1.3G          1.3G 000002
    2                   at      1.3G          1.3G 000004
    3                   at      48.5G         0     000010
    4                   at      48.5G         0     000011
    5                   at      48.5G        43.5G 000018
    6                   at      48.5G         0     000019
    7                   at      48.5G        23.3G 000020
    8                   at      23.8G        23.8G 000021
    9                   at      48.5G        48.5G 000022
   10                  at      48.5G        48.5G 000023
   11                  at      48.5G        48.5G 000024
   12                  at      48.5G         2.6G 000025
   13                  at      48.5G        361.4k 000026
```

程式碼範例 7-22 回收器記錄檔 (續)

```

14          at          48.5G          48.5G 000027
15          at          48.5G          48.5G 000028
16          at          48.5G           0 000029
17          at          1.3G           1.3G 000005
18          at          48.5G          48.5G 000016
19          at          23.8G          23.8G CLN001
20          at          23.8G          23.8G CLN002
21          at          23.8G          23.8G CLN004
22          at          23.8G          23.8G CLN003
23          at          48.5G          421.6M 000015
24          at          1.3G           1.3G 000000
25          at          48.5G           0 000013
26          at          1.3G           1.3G 000003
27          at          48.5G          43.6G 000007
28          at          48.5G          41.8G 000008
29          at          48.5G          46.9G 000006
30          at          48.5G          48.3G 000009
31          at          48.5G           0 000014
32          at          48.5G           0 000012
33          at          48.5G          40.1G 000017
Total Capacity: 1.2T bytes, Total Space Available: 708.7G bytes
Volume utilization 43%, high 95% VSN_min 50%
Recycling is ignored on this robot.

```

```

1 Family: hy                      Path: /var/opt/SUNWsamfs/catalog/historian
Vendor: Sun SAM-FS                Product: Historian
SLOT          ty  capacity          space vsn
(no VSNs in this media changer)
Total Capacity: 0 bytes, Total Space Available: 0 bytes
Volume utilization 0%, high 95% VSN_min 50%
Recycling is ignored on this robot.

```

```

2 Family: testset.1              Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS                Product: Archive set
SLOT          ty  capacity          space vsn
0             at  48.5G           0 000010
1             at  48.5G           0 000014
Total Capacity: 97.1G bytes, Total Space Available: 0 bytes
Volume utilization 100%, high 60% VSN_min 90%: *** Needs recycling ***
Recycling is ignored on this archive set.

```

```
3 Family: testset.2          Path: /etc/opt/SUNWsamfs/archiver.cmd
  Vendor: Sun SAM-FS        Product: Archive set
SLOT      ty      capacity      space vsn
   0             at        48.5G         0   000019
   1             at        48.5G        421.6M 000015
Total Capacity:  97.1G bytes, Total Space Available: 421.6M bytes
Volume utilization 99%, high 60% VSN_min 90%: *** Needs recycling ***
Recycling is ignored on this archive set.
```

```
4 Family: allsam1.1        Path: /etc/opt/SUNWsamfs/archiver.cmd
  Vendor: Sun SAM-FS        Product: Archive set
SLOT      ty      capacity      space vsn
   0             at         1.3G         1.2G 000001
   1             at         1.3G         1.3G 000002
   2             at         1.3G         1.3G 000004
   3             at         1.3G         1.3G 000005
   4             at         1.3G         1.3G 000003
Total Capacity:  6.5G bytes, Total Space Available: 6.3G bytes
Volume utilization 3%, high 60% VSN_min 90%
Recycling is ignored on this archive set.
```

```
5 Family: allsam1.2        Path: /etc/opt/SUNWsamfs/archiver.cmd
  Vendor: Sun SAM-FS        Product: Archive set
SLOT      ty      capacity      space vsn
   0             at        48.5G        43.6G 000007
   1             at        48.5G        41.8G 000008
   2             at        48.5G        46.9G 000006
   3             at        48.5G        48.3G 000009
Total Capacity: 194.2G bytes, Total Space Available: 180.6G bytes
Volume utilization 6%, high 60% VSN_min 90%
Recycling is ignored on this archive set.
```

Need to select candidate for media changer testset.1 to free up 39.8G bytes.
Quantity of data to move limited to (no limit) bytes and 1 VSNs.
Checking 000010. Need to free 39.8G, quantity limit: (no limit), VSN count: 1.

程式碼範例 7-22 回收器記錄檔 (續)

```
VSN is in correct media changer... good.
VSN is not already recycling... good.
VSN has no request files... good.
VSN has no 'archive -n' files...good.
VSN was not specified as "no_recycle" in recycler.cmd file... good.
VSN does not exceed VSN count limit... good.
VSN does not exceed data quantity limit... good.
VSN meets minimum gain requirement.
Recycling is ignored on this media changer - VSN not marked for recycling.
Checking 000014. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN is in correct media changer... good.
VSN is not already recycling... good.
VSN has no request files... good.
VSN has no 'archive -n' files...good.
VSN was not specified as "no_recycle" in recycler.cmd file... good.
VSN exceeds VSN count limit - skipped.
Checking 000019. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000015. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000001. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000003. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000004. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000005. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000002. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000008. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000007. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000006. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000009. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000011. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000029. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000013. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000012. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000026. Need to free 0E, quantity limit: (no limit), VSN count: 0.
```

程式碼範例 7-22 回收器記錄檔 (續)

```
VSN not in correct media changer.
Checking 000025. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000020. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000017. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000018. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN003. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000021. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000022. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000027. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000028. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000023. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000024. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000016. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN001. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN002. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN004. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000000. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
No candidate was found in this media changer.

Need to select candidate for media changer testset.2 to free up 38.8G bytes.
Quantity of data to move limited to (no limit) bytes and 1 VSNs.
Checking 000010. Need to free 38.8G, quantity limit: (no limit), VSN count: 1.
VSN not in correct media changer.
Checking 000014. Need to free 38.8G, quantity limit: (no limit), VSN count: 1.
VSN not in correct media changer.
Checking 000019. Need to free 38.8G, quantity limit: (no limit), VSN count: 1.
VSN is in correct media changer... good.
VSN is not already recycling... good.
VSN has no request files... good.
VSN has no 'archive -n' files...good.
VSN was not specified as "no_recycle" in recycler.cmd file... good.
```

程式碼範例 7-22 回收器記錄檔 (續)

```
VSN does not exceed VSN count limit... good.
VSN does not exceed data quantity limit... good.
VSN meets minimum gain requirement.
Recycling is ignored on this media changer - VSN not marked for recycling.
Checking 000015. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN is in correct media changer... good.
VSN is not already recycling... good.
VSN has no request files... good.
VSN has no 'archive -n' files...good.
VSN was not specified as "no_recycle" in recycler.cmd file... good.
VSN exceeds VSN count limit - skipped.
Checking 000001. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000003. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000004. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000005. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000002. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000008. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000007. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000006. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000009. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000011. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000029. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000013. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000012. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000026. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000025. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000020. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000017. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000018. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
```

程式碼範例 7-22 回收器記錄檔 (續)

```

Checking CLN003. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000021. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000022. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000027. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000028. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000023. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000024. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000016. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking CLN001. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking CLN002. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking CLN004. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000000. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
No candidate was found in this media changer.
34 VSNs:

          ---Archives---      -----Percent-----      testset.1
-----Status-----      Count      Bytes      Use Obsolete Free      Library:Type:VSN
no-data VSN                0           0           0   100       0      adic1:at:000010
no-data VSN                0           0           0   100       0      adic1:at:000014

          ---Archives---      -----Percent-----      testset.2
-----Status-----      Count      Bytes      Use Obsolete Free      Library:Type:VSN
no-data VSN                0           0           0   100       0      adic1:at:000019
partially full            677         2.3G       5    93        2      adic1:at:000015

          ---Archives---      -----Percent-----      allsam1.1
-----Status-----      Count      Bytes      Use Obsolete Free      Library:Type:VSN
partially full            97          173.8M     1     9        90     adic1:at:000001
no-data VSN                0           0           0     2        98     adic1:at:000003
no-data VSN                0           0           0     2        98     adic1:at:000004
empty VSN                  0           0           0     0       100     adic1:at:000005
empty VSN                  0           0           0     0       100     adic1:at:000002

          ---Archives---      -----Percent-----      allsam1.2
-----Status-----      Count      Bytes      Use Obsolete Free      Library:Type:VSN

```

程式碼範例 7-22 回收器記錄檔 (續)

```

no-data VSN          0      0      0    13    87  adic1:at:000008
partially full      98    1.6G    3     7    90  adic1:at:000007
no-data VSN          0      0      0     3    97  adic1:at:000006
empty VSN            0      0      0     0   100  adic1:at:000009

      ---Archives---      -----Percent-----      adic1
-----Status-----      Count      Bytes      Use Obsolete Free      Library:Type:VSN
no-data VSN          0      0      0    100     0  adic1:at:000011
no_recycle VSN       0      0      0    100     0  adic1:at:000029
no-data VSN          0      0      0    100     0  adic1:at:000013
no-data VSN          0      0      0    100     0  adic1:at:000012
no_recycle VSN       0      0      0     99     1  adic1:at:000026
no_recycle VSN       0      0      0     94     6  adic1:at:000025
no_recycle VSN       0      0      0     51    49  adic1:at:000020
no-data VSN          0      0      0     17    83  adic1:at:000017
no-data VSN          0      0      0     10    90  adic1:at:000018
empty VSN            0      0      0     0   100  adic1:at:CLN003
no_recycle VSN       0      0      0     0   100  adic1:at:000021
no_recycle VSN       0      0      0     0   100  adic1:at:000022
no_recycle VSN       0      0      0     0   100  adic1:at:000027
no_recycle VSN       0      0      0     0   100  adic1:at:000028
no_recycle VSN       0      0      0     0   100  adic1:at:000023
no_recycle VSN       0      0      0     0   100  adic1:at:000024
empty VSN            0      0      0     0   100  adic1:at:000016
empty VSN            0      0      0     0   100  adic1:at:CLN001
empty VSN            0      0      0     0   100  adic1:at:CLN002
empty VSN            0      0      0     0   100  adic1:at:CLN004
partially full      12    88.3k    0     0   100  adic1:at:000000

Recycler finished.

===== Recycler ends at Mon Jun  4 09:51:05 2001 =====

```

選擇 VSN 進行回收時，檢查顯示欄資料的回收器記錄檔的最後一部分。最左欄標題為 Status (狀態)。在上述回收器記錄檔中，Status 欄顯示了數個 no_recycle (無回收) 狀態的 VSN。這些是用戶端使用的 VSN。

最佳回收候選者是在 Count (計數)、Bytes (位元組) 及 Use (使用) 欄中數值為 0 的候選者。清單中的最後一個 VSN 顯示其狀態為 partially full (部分全滿)。Count、Bytes 及 Use 統計分別為 12、88.3k 及 0 的 VSN 不是良好的回收候選者。

10. 分析用戶端與伺服器 recycler.log 檔案。

此步驟說明如何選擇屬於回收候選者的 VSN。

從用戶端檢查 `recycler.log` 檔案。Status 欄靠近檔案的結尾。狀態項目為下列類型的 VSN 是回收候選者：

- `no-data VSN` (無資料 VSN)。要回收 `no-data VSN`，請參閱第 179 頁「回收 `no-data VSN`」。
- `partially full`。要回收 `partially full VSN`，請參閱第 181 頁「回收 `partially full VSN`」。

▼ 回收 `no-data VSN`

`no-data VSN` 是最益於回收的 VSN。這些 VSN 的 `Count`、`Bytes` 及 `Use` 欄位均為 0 (零)。

1. 檢查用戶端中的 `recycler.log` 檔案，查看是否有任何 `no-data VSN`。

您可以考慮使用本章範例的用戶端 `zeke` 的 VSN 000029 與 000026 來進行回收，因為它們是 `no-data VSN`。這可以從程式碼範例 7-23 判斷，該範例顯示了 `zeke` 上的用戶端 `recycler.log` 檔案。

程式碼範例 7-23 用戶端 `zeke` 上的 `recycler.log` 檔案

```
# From the client zeke recycler.log file:
```

-----Status-----	Count	Bytes	Use	Obsolete	Free	Library:Type:VSN
no-data VSN	0	0	0	100	0	skyrs:at:000029
no-data VSN	0	0	0	99	1	skyrs:at:000026
partially full	111	2.8G	6	88	6	skyrs:at:000025
empty VSN	0	0	0	0	100	skyrs:at:000028
empty VSN	0	0	0	0	100	skyrs:at:000027

2. 檢查伺服器中的 `recycler.log` 檔案，並判斷伺服器的回收器記錄檔案中是否同樣顯示了您從上一步中選擇的 VSN。

請嘗試確認伺服器中使用中的資料未歸檔在這些 VSN 上。

程式碼範例 7-24 顯示伺服器的 `recycler.log` 檔案中用於 `no_recycle VSN` 的資料。由於在上一步中已選擇 VSN 000029 與 000026 來進行回收，因此伺服器 `recycler.log` 檔案中的資料與用戶端 `recycler.log` 檔案中的資料相同。

程式碼範例 7-24 伺服器 `sky` 上的 `recycler.log` 檔案

```
# From the Server log file:
```

-----Status-----	Count	Bytes	Use	Obsolete	Free	Library:Type:VSN
no-data VSN	0	0	0	100	0	adic1:at:000011
no_recycle VSN	0	0	0	100	0	adic1:at:000029zeke
no-data VSN	0	0	0	100	0	adic1:at:000013

程式碼範例 7-24 伺服器 sky 上的 recycler.log 檔案 (續)

no-data VSN	0	0	0	100	0	adic1:at:000012
no_recycle VSN	0	0	0	99	1	adic1:at:000026
no_recycle VSN	0	0	0	94	6	adic1:at:000025
no_recycle VSN	0	0	0	51	49	adic1:at:000020
no-data VSN	0	0	0	17	83	adic1:at:000017
no-data VSN	0	0	0	10	90	adic1:at:000018
empty VSN	0	0	0	0	100	adic1:at:CLN003
no_recycle VSN	0	0	0	0	100	adic1:at:000021
no_recycle VSN	0	0	0	0	100	adic1:at:000022
no_recycle VSN	0	0	0	0	100	adic1:at:000027
no_recycle VSN	0	0	0	0	100	adic1:at:000028
no_recycle VSN	0	0	0	0	100	adic1:at:000023
no_recycle VSN	0	0	0	0	100	adic1:at:000024
empty VSN	0	0	0	0	100	adic1:at:000016
empty VSN	0	0	0	0	100	adic1:at:CLN001
empty VSN	0	0	0	0	100	adic1:at:CLN002
empty VSN	0	0	0	0	100	adic1:at:CLN004
partially full	12	88.3k	0	0	100	adic1:at:000000

3. (選用) 使用 `tplabel(1M)` 或 `odlabel(1M)` 指令重新標記 VSN。

如果伺服器使用中的資料未歸檔在該 VSN 上，您可以重新標記 VSN。

備註 – 這將損毀 VSN 上的所有資料，並重新恢復空間。

例如：對於磁帶 VSN 000029，請使用下列指令：

```
server# tplabel -vsn 000029 -old 000029 at.000029
```

重新標記此 VSN 000029 後，您將重新取得該 VSN 上的所有空間。

如果媒體是磁光碟，請使用 `odlabel(1M)` 指令。如需有關 `odlabel(1M)` 指令的詳細資訊，請參閱 `odlabel(1M)` 線上手冊。

4. 設計回收排程。

在 Sun SAM-Remote 軟體還未啓用的 Sun StorEdge SAM-FS 環境中，您可以建立 `cron(1)` 工作使回收自動執行。但是，如果已啓用 Sun SAM-Remote 軟體，則無法自動執行回收器。



注意 – 在 Sun SAM-Remote 伺服器上進行回收時，切勿在 Sun SAM-Remote 用戶端上進行回收活動。您應定期手動回收以滿足站點的需要。以此方式進行回收需要執行更多操作。但是，這是唯一可確定資料免於重新標記卡匣錯誤的方法。

▼ 回收 partially full VSN

您也可以回收報告 partially full 狀態的 VSN。

1. 檢查用戶端中的 recycler.log 檔案，查看是否有任何 partially full VSN。

您可以考慮使用本章範例用戶端 zeke 中的 VSN 000025 來進行回收，因為其狀態為 partially full。您可以從程式碼範例 7-25 判斷，該範例顯示了 zeke 上的用戶端 recycler.log 檔案。

程式碼範例 7-25 用戶端 zeke 上的 recycler.log 檔案

```
# From the client zeke recycler.log file:
```

-----Status-----	Count	Bytes	Use	Obsolete	Free	defaultset.2 Library:Type:VSN
no-data VSN	0	0	0	100	0	skyrs:at:000029
no-data VSN	0	0	0	99	1	skyrs:at:000026
partially full	111	2.8G	6	88	6	skyrs:at:000025
empty VSN	0	0	0	0	100	skyrs:at:000028
empty VSN	0	0	0	0	100	skyrs:at:000027

VSN 000025 顯示了其 6% 的空間在使用中。必須重新歸檔這些使用中的歸檔影像，才能重新回收此 VSN。本程序中的下列步驟顯示了如何確定這些使用中的歸檔影像重新歸檔到其他 VSN。

2. 檢查伺服器端中的 recycler.log 檔案，以確定伺服器使用中的資料未歸檔在該 VSN 上。

例如：查看上一步中選擇用於回收的程式碼範例 7-26 中的 VSN 000025 資料。伺服器的 recycler.log 檔案顯示了 VSN 000025 有 6% 的可用空間，這與用戶端的 recycler.log 檔案中報告的百分比相同。由於伺服器無法辨識用戶端的歸檔影像，因此伺服器無法報告佔用百分比為 6% 的使用中歸檔影像與 88% 的過時影像。伺服器報告剩餘的 94% 都被過時歸檔影像佔用。

程式碼範例 7-26 伺服器 sky 上的 recycler.log 檔案

```
# From the Server log file:
```

-----Status-----	Count	Bytes	Use	Obsolete	Free	adic1 Library:Type:VSN
no-data VSN	0	0	0	100	0	adic1:at:000011
no_recycle VSN	0	0	0	100	0	adic1:at:000029
no-data VSN	0	0	0	100	0	adic1:at:000013
no-data VSN	0	0	0	100	0	adic1:at:000012
no_recycle VSN	0	0	0	99	1	adic1:at:000026
no_recycle VSN	0	0	0	94	6	adic1:at:000025
no_recycle VSN	0	0	0	51	49	adic1:at:000020
no-data VSN	0	0	0	17	83	adic1:at:000017

程式碼範例 7-26 伺服器 sky 上的 recycler.log 檔案 (續)

no-data VSN	0	0	0	10	90	adic1:at:000018
empty VSN	0	0	0	0	100	adic1:at:CLN003
no_recycle VSN	0	0	0	0	100	adic1:at:000021
no_recycle VSN	0	0	0	0	100	adic1:at:000022
no_recycle VSN	0	0	0	0	100	adic1:at:000027
no_recycle VSN	0	0	0	0	100	adic1:at:000028
no_recycle VSN	0	0	0	0	100	adic1:at:000023
no_recycle VSN	0	0	0	0	100	adic1:at:000024
empty VSN	0	0	0	0	100	adic1:at:000016
empty VSN	0	0	0	0	100	adic1:at:CLN001
empty VSN	0	0	0	0	100	adic1:at:CLN002
empty VSN	0	0	0	0	100	adic1:at:CLN004
partially full	12	88.3k	0	0	100	adic1:at:000000

3. 使用 VSN 上的 `chmed(1M)` 指令與 `+c` 選項。

例如：在本程序中，指令如下所示：

```
server# chmed +c at.000025
```

此指令可顯示您要重新歸檔此 VSN 上使用中檔案的回收器。要重新歸檔的檔案包含 Use 欄中用戶端 `recycler.log` 檔案報告的 6%。如需有關 `chmed(1M)` 指令的詳細資訊，請參閱 `chmed(1M)` 線上手冊。

4. 使用 `sam-recycler(1M)` 指令以再次執行回收器。

例如：在本程序中，指令如下所示：

```
client# sam-recycler -dvx
```

這將標記要重新歸檔的每個使用中檔案，並顯示每個使用中檔案應重新歸檔至其他 VSN 的歸檔器。

5. 啟動歸檔器。

要執行此操作，可讓歸檔器正常執行，或透過用戶端上的 `samu(1M)` 公用程式鍵入 `:arrun` 以啟動歸檔器。如需更多有關 `:arrun(1M)` 指令的資訊，請參閱 `samu(1M)` 線上手冊。

6. 完成歸檔後，發出 `sam-recycler(1M)` 指令以在用戶端上重新執行回收器。

這可確保所有使用中檔案已重新歸檔。

例如：在本程序中，指令如下所示：

```
client# sam-recycler -dvx
```

7. (選用) 使用 `tplabel(1M)` 或 `odlabel(1M)` 指令從伺服器重新標記 VSN。

如果 Count、Bytes 及 Use 欄位均為 0 (零)，您可以重新標記伺服器中的 VSN。

例如：在本程序中，您可以使用下列指令重新標記磁帶 VSN：

```
server# tplabel -vsn 000025 -old 000025 at.000025
```

上述指令會重新標記 VSN 並損毀 VSN 上的所有資料。重新標記此 VSN 後，您將重新取得該 VSN 上 88% 的空間。

如果媒體是磁光碟，請使用 `odlabel(1M)` 指令。如需有關 `odlabel(1M)` 指令的詳細資訊，請參閱 `odlabel(1M)` 線上手冊。

8. 設計回收排程。

在 Sun SAM-Remote 軟體還未啓用的 Sun StorEdge SAM-FS 環境中，您可以建立 `cron(1)` 工作使回收自動執行。但是，如果已啓用 Sun SAM-Remote 軟體，則無法自動執行回收器。



注意 – 在 Sun SAM-Remote 伺服器上進行回收時，切勿在 Sun SAM-Remote 用戶端上進行回收活動。您應定期手動回收以滿足站點的需要。以此方式進行回收需要執行更多操作。但是，這是唯一可確定資料免於重新標記卡匣錯誤的方法。

於 Sun SAM-Remote 環境進行回收 — 方法 2

本章節說明了您可以使用 Sun SAM-remote 軟體回收磁碟區的另一種方法。



注意 – 請在完成本程序全部步驟並在測試配置以查看回收進行是否正確後，才在 Sun SAM-Remote 環境下使用回收器。

▼ 設定回收 — 方法 2

1. 在 Sun SAM-Remote 用戶端上，發出 `sam-recycler(1M)` 指令以判斷最佳回收候選者的磁碟區。

例如：

```
client# sam-recycler -dvx
```

您可以透過分析回收器記錄檔案加以判斷。

2. 在 Sun SAM-Remote 伺服器上，發出 `chmed(1M)` 指令以設定選定 VSN 上的回收旗標。

例如：

```
server# chmed +c at.00025
```

3. 在 Sun SAM-Remote 用戶端上，發出 `sam-recycler(1M)` 指令以回收 Sun SAM-Remote 用戶端上選定的 VSN。

例如：

```
client# sam-recycler -dvx
```

4. 請稍候，直到要回收的 VSN 完全清空歸檔影像。
用戶端上的歸檔器將執行此操作。
5. 在 Sun SAM-Remote 伺服器上，發出 `tplabel(1M)` 或 `odlabel(1M)` 指令，以在磁碟區完全清空歸檔影像後重新標記磁碟區。
6. 在 Sun SAM-Remote 伺服器上，清除任何防止磁碟區在 Sun SAM-Remote 用戶端上進行歸檔（例如：R 或 c）的旗標。

此外，在 Sun SAM-Remote 伺服器上回收磁碟區時，切勿在 Sun SAM-Remote 用戶端上進行這些回收活動。

第8章

進階主題

本章說明基本系統管理與用法範圍之外的進階主題。

將介紹以下主題。

- 第 185 頁 「使用裝置記錄」
 - 第 188 頁 「使用可移除式媒體檔案」
 - 第 190 頁 「使用區段檔案」
 - 第 191 頁 「使用系統錯誤功能報告」
-

使用裝置記錄

裝置記錄功能提供可用於分析某些類型裝置問題的指定裝置之錯誤資訊。它可協助決定自動程式庫、磁帶機或光碟機的事件失敗順序。請注意，裝置記錄工具不會收集軟體媒體錯誤 (例如：可復原的讀取錯誤)。

裝置記錄訊息將寫入個別記錄檔。每個自動程式庫、每個磁帶與光碟機裝置及記錄器 (historian) 均有一個記錄檔。記錄檔位於 `/var/opt/SUNWsamfs/devlog`。每個記錄檔的名稱與設備序號的名稱相同。

範例：假設一個 Sun StorEdge SAM-FS 檔案系統與一個有兩個光學磁帶機的 Hewlett Packard 光學程式庫。

程式碼範例 8-1 顯示 mcf 檔案。

程式碼範例 8-1 mcf 檔案範例

```
/dev/samst/c1t5u0 40 hp hp40 - etc/opt/SUNWsamfs/hp40_cat
/dev/samst/c1t4u0 41 mo hp40 -
/dev/samst/c1t6u0 42 mo hp40 -
```

程式碼範例 8-2 顯示 /var/opt/SUNWsamfs/devlog 檔案。

程式碼範例 8-2 devlog 檔案

```
# pwd
/var/opt/SUNWsamfs/devlog
# ls
40      41      42      43
#
```

裝置 43 是記錄器。

何時使用裝置記錄檔

裝置記錄檔可輕易產生許多記錄檔訊息，特別是在所有裝置的全部記錄選項已開啓並且裝置活動頻繁時。最初的裝置記錄設定已設爲以下預設值：

```
err retry syserr date
```

如果懷疑 Sun StorEdge SAM-FS 環境中配置的某個裝置有問題，啓動該裝置的額外記錄檔事件不失爲適當的方法。此外，如果服務供應商建議您啓用裝置記錄檔，您也可以這麼做。在這類情況下，將事件設爲 `detail`。在極端情況下，服務供應商可能會建議您將裝置的事件設爲 `all`。這將新增額外的記錄檔資訊。但一般來說，使用過多的記錄檔執行系統可能不太有用或切實可行。

裝置記錄檔資訊會在 `samexplorer(1M)` 指令發出時自動收集。這可讓檔案系統服務將任何可能的裝置錯誤資訊作爲問題分析活動的一部份來檢閱。

啓動裝置記錄檔

有兩種方法您可用來啓用裝置記錄檔。

同時對於方法 1 與方法 2：

- `eq` 是 `mcf` 檔案中裝置的「設備序號」或所有設備的關鍵字 `all`。
- 裝置記錄檔事件在 `samset(1M)` 線上手冊中列出。裝置記錄檔事件也列在下方。請注意，裝置記錄檔訊息僅使用英文。`event` 是以下清單中的一個或多個類型：
 - `all`
 - `date`
 - `default`
 - `detail`

- err
- event
- label
- mig
- module
- msg
- none
- retry
- stage
- stage_ck
- syserr
- tapealert
- time

您可以用以下兩種方式之一來啓動裝置記錄檔。這些程序如下所示：

- 第 187 頁「使用 `samset(1M)` 指令以啓動裝置記錄檔」
- 第 187 頁「編輯 `defaults.conf` 檔案以啓動裝置記錄檔」

▼ 使用 `samset(1M)` 指令以啓動裝置記錄檔

- 使用 `samset(1M)` 指令

例如：

```
# samset devlog eq 事件
```

關於 `eq`，請指定要記錄訊息的裝置之「設備序號」。

關於 `event`，請指定一或多個第 186 頁「啓動裝置記錄檔」中所述的事件。如果您指定了一個以上的事件，請以空格字元區隔事件。

要取得更多有關 `samset(1M)` 指令的資訊，請參閱 `samset(1M)` 線上手冊。

▼ 編輯 `defaults.conf` 檔案以啓動裝置記錄檔

1. 成為超級使用者。
2. 使用 `vi(1)` 或其他編輯器來開啟 `/etc/opt/SUNWsamfs/defaults.conf` 檔案。

3. 將 `devlog` 指令新增到 `defaults.conf` 檔案中。

新增以下指令：

```
devlog eq 事件
```

關於 `eq`，請指定要記錄訊息的裝置之「設備序號」。

關於 `event`，請指定一或多個第 186 頁「啟動裝置記錄檔」中所述的事件。如果您指定了一個以上的事件，請以空格字元區隔事件。

當 Sun StorEdge SAM-FS 檔案系統啟動時，它會自動將每個可用裝置的事件類型設為 `default`。您也可以使用 `samset(1M)` 指令來決定每個裝置記錄檔的目前設定。

4. 儲存並關閉 `defaults.conf` 檔案。
5. 使用 `samd(1M) config` 指令來傳遞 `defaults.conf` 檔案變更。

```
# samd config
```

使用可移除式媒體檔案

您可使用 `request(1)` 指令手動建立、寫入及讀取不使用磁碟快取記憶體來緩衝資料的檔案。以此方式建立的檔案稱為可移除式媒體檔案。

可移除式媒體檔案因為具有權限、使用者名稱、群組名稱及大小屬性而顯得與一般的 Sun StorEdge SAM-FS 檔案相似。然而，資料並不位於磁碟快取中。因此，您可以建立大於磁碟快取記憶體的檔案，並將它寫入可移除式媒體卡匣。系統會針對您在 `request(1)` 指令上指定的檔案，在 `.inodes` 檔案中建立 `inode` 項目。使用者無需瞭解檔案從可移除式媒體的何處開始。(對於在磁碟快取記憶體中含有資料的檔案也是如此。) Sun StorEdge SAM-FS 檔案系統會從 `inode` 項目讀取資訊。多個可移除式媒體檔案可位於相同磁碟區中。

如果可移除式媒體檔案橫跨多個磁碟區，此則稱為磁碟區溢位檔案。磁碟區溢位功能可讓系統將單一檔案橫跨於多個卡匣上的多個磁碟區。磁碟區溢位檔案是一種可移除式媒體檔案。如果您有超出選定媒體容量之極大檔案，磁碟區溢位功能則會很有用。

▼ 建立可移除式媒體或磁碟區溢位檔案

1. 使用 `tplabel(1M)` 或 `odlabel(1M)` 指令標示磁帶或磁光碟匣。
要取得有關這些指令的資訊，請參閱其個別線上手冊。

2. 使用 request(1) 指令

至少，使用以下選項：

```
request -m 媒體類型 -v vsn [vsn/vsn ...] [-l vsn 檔案] 輸入檔案
```

表 8-1 request(1) 指令的引數

引數	意義
媒體類型	可移除式媒體卡匣的媒體類型。要取得有關有效媒體類型規定的資訊，請參閱 mcf(4) 線上手冊。
vsn	可移除式媒體卡匣的「磁碟區序號」。 如果您指定一個以上的 vsn，則表示您正在建立磁碟區溢位檔案。您最多可以對磁碟區溢位檔案指定 256 個 vsn。使用斜線字元 (/) 來區隔 vsn 引數。 指定的 vsn 不應在用於自動歸檔的 Sun StorEdge SAM-FS 環境中的磁碟區之間。歸檔會將下一個要歸檔的檔案附加至目前資料的末端，並在每次將 EOF 標籤移出資料。
vsn 檔案	一個含有 vsn 清單的輸入檔案。如果您有許多 vsn，在輸入檔案中指定 vsn 清單會比在指令行上簡單。
輸入檔案	要寫入可移除式媒體卡匣的檔案。此檔案必須位於 Sun StorEdge SAM-FS 檔案系統中。

範例 1。以下指令會建立可移除式媒體檔案：

```
# request -m lt -v aaa rem1
```

範例 2。以下指令會在三個磁碟區上建立磁碟區溢位檔案：

```
# request -m lt -v TAPE01/TAPE02/TAPE03 large.file
```

您必須依序讀取和寫入可移除式媒體檔案。如果磁碟區位於一個 mcf 檔案中所定義的自動程式庫，Sun StorEdge SAM-FS 檔案系統會自動掛載請求的磁碟區。

磁碟區如果存在可移除式媒體檔案，則可防止回收該磁碟區。回收器僅預期歸檔檔案位於指定用於歸檔的特定磁碟區。此外，可移除式媒體檔案將永不歸檔。

NFS 不支援可移除式媒體檔案。

使用 request(1) 指令會略過歸檔器的一般功能。

有關說明如何建立可移除式媒體檔案的範例，請參閱 request(1) 線上手冊。

使用區段檔案

Sun StorEdge SAM-FS 環境可支援區段檔案。區段檔案可提高超大容量檔案的磁帶儲存裝置之取得速度、存取及易管理性。區段檔案的容量可超過實體磁碟快取。使用區段檔案，則可能在任一時刻僅讓部分檔案位於磁碟快取中。

`segment(1)` 指令允許您指定區段的大小。無法將區段大小設為小於目前檔案的大小。

區段檔案支援磁帶資料平行儲存區。在檔案分段之後，可將它同時串列於多個磁帶裝置，這樣可大量減少儲存檔案區段所需的時間。允許使用者僅取得所需的檔案區段而非整個檔案，可加速資料的存取。

由於僅歸檔檔案中變更的部分，因此分段可提高歸檔效率。檔案中的區段可同時歸檔，區段檔案可同時分段備份。這樣會在歸檔與取得時增加效能。

分段可在檔案、目錄或整個檔案系統中啟動。區段檔案可支援所有其他的 Sun StorEdge SAM-FS 功能。

以下章節說明區段檔案與非區段檔案之間的區別。要取得更多有關區段檔案的資訊，請參閱 `segment(1)` 或 `sam_segment(3)` 線上手冊。

歸檔

對於區段檔案，可歸檔單位指的是區段本身而並非檔案。所有歸檔屬性與優先權將套用於個別區段而並非檔案。

歸檔的單位是區段。您可以藉由指定 `archiver.cmd` 檔案中的歸檔組之 `-drives` 和 `-drivemin` 參數來資料平行儲存區段。

例如，假設檔案系統中有一個 100 MB 的區段檔案，其區段大小是 10 MB。如果 `archiver.cmd` 檔案使用 `-drives 2` 指令定義一個歸檔組，此檔案將同時歸檔至 2 台磁碟機。區段 1、3、5、7 及 9 使用第一台磁碟機歸檔，區段 2、4、6、8 及 10 則使用第二台磁碟機歸檔。

僅已修改的區段將會歸檔（並非整個檔案）。每個區段最多可建立四個歸檔備份。Sun StorEdge SAM-FS 也可支援區段的磁碟區溢位。

備註 – 區段檔案的索引不包含使用者資料。將視為中介資料，指定給檔案系統歸檔組。

災難復原

若需關於發生災難時復原區段檔案的資訊，請參閱「Sun StorEdge SAM-FS Troubleshooting Guide」。

使用系統錯誤功能報告

系統錯誤工具 (SEF) 報告系統可擷取自動程式庫中磁帶裝置的記錄檔感應資料，將其寫入記錄檔，然後將資料轉換成人類可讀取的形式。其由下列內容組成：

- 一個包含磁帶裝置記錄檔感應頁資料的記錄檔。
- 一個 `sefreport(1M)` 指令，用於將記錄檔以人類可讀取的格式寫入 `stdout`。此記錄檔可用作輸入使用者提供的分析指令碼。

各廠商提供的記錄檔感應頁會不同。有關參數代碼、控制位元及參數值的含義，請參閱每個指定裝置的供應商文件資料。

獨立磁帶機不支援 SEF。對於不支援 `tapealert(1M)` 功能的舊型 SCSI-2 裝置，SEF 報告是最有用的。要取得更多相關資訊，請參閱 `tapealert(1M)` 線上手冊。

▼ 啓動 SEF 報告

1. 成為超級使用者。
2. 使用 `mkdir(1)` 指令建立 SEF 目錄。

例如：

```
# mkdir /var/opt/SUNWsamfs/sef
```

3. 使用 `touch(1)` 指令建立記錄檔。

透過建立 `sefdata` 記錄檔，可在安裝之後隨時啓動 SEF 報告。最初，SEF 記錄檔必須是空的。

```
# touch /var/opt/SUNWsamfs/sef/sefdata
```

之前的範例指令顯示了 `/var/opt/SUNWsamfs/sef/sefdata` 中建立的 SEF 記錄檔。這是預設位置。

4. 使用 `samd(1M) stop` 和 `samd(1M) start` 來初始化 SEF 報告。

例如：

```
# samd stop
# samd start
```

SEF 資料將在產生之後附加至記錄檔。

您可將 SEF 報告設定為從替代位置記錄與讀取記錄檔感應資料。要取得更多有關從替代位置讀取記錄檔感應資料的資訊，請參閱 `sefreport(1M)` 線上手冊。

SEF 報告輸出

在使用 `sefreport(1M)` 指令之前，確定 `/opt/SUNWsamfs/sbin` 位於指令路徑中。SEF 報告輸出由標題行與記錄檔感應資料組成。

在標題行的後面，將會列印記錄中每個頁面的記錄檔感應資料。對於每個記錄檔感應頁，將會列印識別頁面代碼的行，其後面是一行欄標題。接著會列印資料（每行三欄），其具有以下標題：`param code`、`control` 及 `param value`。所有資料以十六進位符號產生。

▼ 產生 SEF 輸出

● 使用 `sefreport(1M)` 指令來產生 SEF 輸出。

以下選項是 `sefreport(1M)` 指令中最常用的選項：

- `-d` 選項：`-d` 選項產生額外的裝置資訊。此選項可將每個記錄中包含「設備序號」與路徑名稱的額外標題行寫入裝置。這樣可輕易搜尋並找出有關指定裝置的 SEF 記錄。
- `-v` 選項或 `-t` 選項：
 - `-v` 選項以詳細資訊模式產生資訊。此選項可將有關「設備序號」、頁面代碼及 VSN 的資訊附加至記錄的每行。這樣則可能僅選擇有關指定裝置或指定欄的文字行。
 - `-t` 選項會產生含有文字說明的記錄檔感應輸出。對於記錄檔感應資料輸出的每一行，報告都會包括一個含有「設備序號」、頁碼、VSN 及參數碼說明的額外資料平行儲存區。

請勿在相同的指令行上指定 `-t` 和 `-v` 選項。它們會彼此排除。

例如，下列 SEF 指令可從預設位置讀取 SEF 記錄檔，寫入每個裝置的裝置號碼與路徑名稱，然後產生執行結果：

```
# sefreport -d /var/opt/SUNWsamfs/sef/sefdata > sef.output
```

程式碼範例 8-3 顯示了 sef.output 檔案的內容。

程式碼範例 8-3 sef.output 內容

```
Record no. 1
Mon Mar 26 11:17:48 2001 STK      9840          1.25 VSN 002981
  Eq no. 32   Dev name /dev/rmt/1cbn

  PAGE CODE 2
  param code  control  param value
    00h       74h     0x0
    01h       74h     0x0
    02h       74h     0x0
    03h       74h     0x0
    04h       74h     0x0
    05h       74h     0x40050
    06h       74h     0x0

  PAGE CODE 3
  param code  control  param value
    00h       74h     0x0
    01h       74h     0x0
    02h       74h     0x0
    03h       74h     0x0
    04h       74h     0x0
    05h       74h     0x140
    06h       74h     0x0

  PAGE CODE 6
  param code  control  param value
    00h       74h     0x0

Record no. 2
Mon Mar 26 11:30:06 2001 STK      9840          1.25 VSN 002999
  Eq no. 31   Dev name /dev/rmt/0cbn

  PAGE CODE 2
  param code  control  param value
    00h       74h     0x0
    01h       74h     0x0
    02h       74h     0x0
    03h       74h     0x0
    04h       74h     0x0
    05h       74h     0x1400a0
    06h       74h     0x0
```

程式碼範例 8-3 sef.output 內容 (續)

```
PAGE CODE 3
param code control param value
    00h      74h      0x0
    01h      74h      0x0
    02h      74h      0x0

    03h      74h      0x0
    04h      74h      0x0
    05h      74h      0x190
    06h      74h      0x0
```

```
PAGE CODE 6
param code control param value
    00h      74h      0x0
```

```
Record no. 3
Mon Mar 26 11:30:23 2001 STK          9840          1.25 VSN 002981
Eq no. 32   Dev name /dev/rmt/1cbn
```

```
PAGE CODE 2
param code control param value
    00h      74h      0x0
    01h      74h      0x0
    02h      74h      0x0
    03h      74h      0x0
    04h      74h      0x0
    05h      74h      0x18400f0
    06h      74h      0x0
```

```
PAGE CODE 3
param code control param value
    00h      74h      0x0
    01h      74h      0x0
    02h      74h      0x0
    03h      74h      0x0
    04h      74h      0x0
    05h      74h      0x1e0
    06h      74h      0x0
```

```
PAGE CODE 6
```

程式碼範例 8-3 sef.output 內容 (續)

param code	control	param value
00h	74h	0x0
.		

備註 – 上述執行結果經過刪略以用於此手冊。

若需更多有關 SEF 記錄檔的資訊 (包含其內容與格式)，請參閱 `sefdata(4)` 線上手冊。
若需更多有關選擇使用的 SEF 報告格式之資訊，請參閱 `sefreport(1M)` 線上手冊。

管理 SEF 記錄檔

管理 SEF 記錄檔的方式和管理任何其他 Sun StorEdge SAM-FS 記錄檔一樣。您可以定期執行 `cron(1)` 工作以將目前的記錄檔儲存至其他位置、刪除舊的 SEF 檔案、建立新的 (空的) SEF 檔案或執行其他工作。

您也可以使用 `log_rotate.sh(1M)` 公用程式以循環此記錄檔。

若需更多有關管理 SEF 記錄檔的工具之資訊，請參閱 `cron(1)` 或 `log_rotate.sh(1M)` 線上手冊。

SEF sysevent 功能

除了 SEF 記錄檔外，使用 Solaris sysevent 即可使用媒體分析的磁帶機 SCSI 記錄感應錯誤計數器第 2 和第 3 頁。SEF sysevent 功能是依照預設在卸載前以一次輪詢間隔啟用。SEF sysevent 運作方式是由 `defaults.conf` 和 `samset` 控制。

▼ 建立 SEF sysevent 處理程式

1. 建立 `/var/tmp/xx` 檔案與下列相似：

```
#!/bin/ksh
echo "$@" >> /var/tmp/xx.dat
exit 0
```

2. 使 `/var/tmp/xx` 檔案可執行：

```
# chmod a+rx /var/tmp/xx
```

3. 輸入下列指令，增加 SEF sysevent 處理程式到 `syseventd(1M)` 檔案：

```
# syseventadm add -vSUNW -pSUNWsamfs -cDevice -sSEF
/var/tmp/xx \"\$VENDOR\" \"\$PRODUCT\" \"\$USN\" \"\$REV\"
\$TOD \$SEQ_ORD \"\$NAME\" \$INQ_TYPE \"\$MEDIA_TYPE\"
```

```
\ "$VSN\" \ $LABEL_TIME \ $LP2_PC0 \ $LP2_PC1 \ $LP2_PC2 \ $LP2_PC3  
\ $LP2_PC4 \ $LP2_PC5 \ $LP2_PC6 \ $LP3_PC0 \ $LP3_PC1 \ $LP3_PC2  
\ $LP3_PC3 \ $LP3_PC4 \ $LP3_PC5 \ $LP3_PC6 \ $WHERE \ $sequence  
  
# syseventadm restart
```

此指令建立包含 SEF sysevent handler /var/tmp/xx 的 /etc/sysevent/config/SUNW,SUNWsamfs,Device,sysevent.conf 檔案以及將事件處理程式載入 syseventd 常駐程式。使用 syseventadm(1M) 指令時需要雙引號，因為資料平行儲存區可能為空值且資料為定位。

備註 – syseventadm(1M) 指令不存在於 Solaris 8。您必須手動建立 /etc/sysevent/config/SUNW,SUNWsamfs,Device,sysevent.conf 檔案和內容，然後輸入 `pkill -HUP syseventd` 以重新啟動 syseventd 常駐程式。

4. 要載入 SEF sysevent 處理程式，請執行 `pkill -HUP syseventd` 以啟動 /var/tmp/xx SEF sysevent 處理程式。

若需得更多關於 SEF sysevent 用法的資訊，請參閱 `sefsysevent(4)` 線上手冊。

使用供應商指定作業程序的程式庫基本作業

您可以將許多不同製造商的程式庫包含在 Sun StorEdge SAM-FS 環境中。針對大部分的程式庫，您應該使用第 7 頁「使用自動程式庫和手動載入磁碟機」中說明的作業程序。但某些程式庫具有廠商指定的作業程序，這些作業程序將在本章中說明。

備註 – Sun StorEdge SAM-FS 軟體可以與許多製造商的自動程式庫相容。請諮詢 Sun 銷售代表或是授權的服務供應商，以獲得有關程式庫型號、韌體等級，以及其他相容性資訊。

本章將說明以下自動程式庫：

- 第 197 頁「ADIC/Grau 自動程式庫」
- 第 199 頁「Fujitsu LMF 自動程式庫」
- 第 201 頁「IBM 3584 UltraScalable 程式庫」
- 第 202 頁「IBM 3494 程式庫」
- 第 203 頁「Sony 直接連結式 8400 PetaSite 自動程式庫」
- 第 206 頁「Sony 網路連結式自動程式庫」
- 第 207 頁「StorageTek ACSLS 連結式自動程式庫」

ADIC/Grau 自動程式庫

如果擁有 ADIC/Grau 自動程式庫，請使用本節的程序以匯入與匯出卡匣。這些程序與第 7 頁「使用自動程式庫和手動載入磁碟機」中說明的不同。

因為您實際上使用供應商提供的公用程式，來新增和移除 ADIC/Grau 自動程式庫中的卡匣，Sun StorEdge SAM-FS 介面 (`import(1M)`、`samexport(1M)` 和 File System Manager) 只會影響程式庫目錄檔。

備註 – 在 x64 硬體平台上的 Sun StorEdge SAM-FS 軟體不支援 ADIC/Grau 網路連結式程式庫。

▼ 匯入卡匣

要匯入卡匣，請執行以下步驟：

1. 使用 **ADIC/Grau** 指令以將卡匣實際移至程式庫。
2. 使用 **Sun StorEdge SAM-FS** `import(1M)` 指令更新程式庫目錄檔。

請用以下格式使用此指令：

```
import -v volser eq
```

表 8-2 `import(1M)` 指令的引數

引數	意義
<i>volser</i>	欲新增的 <i>volser</i> 。在使用新項目更新程式庫目錄檔之前， <code>grauaci</code> 介面會確認 ADIC/Grau 自動程式庫具有 <i>volser</i> 資訊。
<i>eq</i>	要被定址的裝置之「設備序號」，就如 <code>mcf</code> 檔案所定義。

▼ 匯出卡匣

要匯出卡匣，請執行以下步驟：

1. 使用 **Sun StorEdge SAM-FS** `samexport(1M)` 指令從程式庫目錄檔移除項目。

請用下列其中一種格式使用此指令：

```
samexport eq:插槽  
samexport 媒體類型.vsn
```

表 8-3 samexport(1M) 指令的引數

引數	意義
<i>eq</i>	要被定址的裝置之「設備序號」，就如 <i>mcf</i> 檔案所定義。
插槽	表示在程式庫目錄檔中識別的自動程式庫中儲存插槽的號碼。
媒體類型	表示媒體類型。如需有效媒體類型清單，請參閱 <i>mcf(4)</i> 線上手冊。
<i>vsn</i>	表示指定給磁碟區的磁碟區序列名稱。

samexport(1M) 指令會在匯入每個 VSN 時更新程式庫目錄檔，並將每個 VSN 的程式庫目錄檔項目從程式庫目錄檔移至記錄器。

2. 使用 **ADIC/Grau** 指令以將卡匣實際移出程式庫。

Fujitsu LMF 自動程式庫

如果擁有 Fujitsu LMF 自動程式庫，請使用本節的程序以匯入與匯出卡匣。這些程序與第 7 頁「使用自動程式庫和手動載入磁碟機」中說明的不同。

因為您實際上使用供應商提供的公用程式，來新增或移除 Fujitsu LMF 自動程式庫中的卡匣，Sun StorEdge SAM-FS 介面 (*import(1M)*、*samexport(1M)* 和 File System Manager) 只會影響程式庫目錄檔。

備註 – 在 x64 硬體平台上的 Sun StorEdge SAM-FS 軟體不支援 Fujitsu LMF 網路連結式程式庫。

▼ 匯入卡匣

要匯入卡匣，請執行以下步驟：

1. 使用 **Fujitsu** 指令以將卡匣實際移至程式庫。
2. 使用 **Sun StorEdge SAM-FS** *import(1M)* 指令更新程式庫目錄檔。

請用以下格式使用此指令：

```
import -v volser eq
```

表 8-4 import(1M) 指令的引數

引數	意義
<i>volser</i>	欲新增的 <i>volser</i> 。在使用新項目更新程式庫目錄檔之前， <i>fujitsulmf</i> 介面會確認 LMF 自動程式庫具有 <i>volser</i> 資訊。
<i>eq</i>	要被定址的裝置之「設備序號」，就如 <i>mcf</i> 檔案所定義。

▼ 匯出卡匣

要匯出卡匣，請執行以下步驟：

1. 使用 **Sun StorEdge SAM-FS** `samexport(1M)` 指令從程式庫目錄檔移除項目。
以下列格式之一使用此指令：

```
samexport eq:插槽  
samexport 媒體類型.vsn
```

表 8-5 samexport(1M) 指令的引數

引數	意義
<i>eq</i>	要被定址的裝置之「設備序號」，就如 <i>mcf</i> 檔案所定義。
插槽	表示在程式庫目錄檔中識別的自動程式庫中儲存插槽的號碼。
媒體類型	表示媒體類型。如需有效媒體類型清單，請參閱 <i>mcf(4)</i> 線上手冊。
<i>vsn</i>	表示指定給磁碟區的磁碟區序列名稱。

`samexport(1M)` 指令會在匯入每個 *VSN* 時更新程式庫目錄檔，並將每個 *VSN* 的程式庫目錄檔項目從 **Sun StorEdge SAM-FS** 程式庫目錄檔移至 **Sun StorEdge SAM-FS** 記錄器。

2. 使用 **Fujitsu** 指令以將卡匣實際移出程式庫。

IBM 3584 UltraScalable 程式庫

Sun StorEdge SAM-FS 環境中支援 IBM 3584 UltraScalable 程式庫。以下章節說明此程式庫作業與第 7 頁「使用自動程式庫和手動載入磁碟機」說明程序不同的地方。

匯入卡匣

當啟動 Sun StorEdge SAM-FS 軟體時，並不會自動匯入信箱中的卡匣。

清潔磁碟機

若要在 Sun StorEdge SAM-FS 環境中使用此程式庫，請停用自動清潔然後啟動主控清潔。IBM 刊物 GA32-0408-01「IBM 3584 UltraScalable 程式庫規劃與操作指南」中有說明本程序。ibm3584(7) 線上手冊中也有說明。

分割

此程式庫可容納幾個磁碟機。如果使用多個磁碟機，則可能會將此一個實體程式庫切分成兩個、三個或四個邏輯程式庫。如果已將程式庫切分成兩個或更多的邏輯程式庫，在將 IBM 3584 程式庫新增到 Sun StorEdge SAM-FS 環境之前，請確定這些邏輯程式庫操作正常。

當卡匣從已分割的程式庫中匯出時，僅有匯出該卡匣的邏輯程式庫可存取拔取器插槽。如果移除並手動重新插入卡匣，則任何邏輯分割區均可存取卡匣。

▼ 移除卡匣

以下步驟說明此情況中所使用的移除動作：

1. 開啟門蓋。
2. 移除卡匣。
3. 關閉門蓋。
4. 等待門蓋鎖定，然後解除鎖定。
5. 開啟門蓋。
6. 更換卡匣。

7. 關閉門蓋。

如需使用本程式庫當做 Sun StorEdge SAM-FS 環境中之邏輯分割程式庫的詳細資訊，請參閱 IBM 文件資料或 [ibm3584\(7\)](#) 線上手冊。

IBM 3494 程式庫

Sun StorEdge SAM-FS 環境中支援 IBM 3494 程式庫。以下章節說明此程式庫作業與第 7 頁「使用自動程式庫和手動載入磁碟機」說明程序不同的地方。

備註 – 在 x64 硬體平台上的 Sun StorEdge SAM-FS 軟體不支援 IBM 3494 網路連結式程式庫。

▼ 匯入卡匣

要匯入卡匣，請執行以下步驟：

1. 將新的媒體置入 I/O 插槽。
2. 關閉門蓋。

程式庫會鎖定門蓋然後將媒體移至儲存區內。您一次只能匯入 100 個磁碟區。

如果程式庫配置為 `access=private`，則這是您需要執行的最後一步。程式庫會通知常駐程式媒體已經移動，並且新增至目錄檔中。

3. 使用 `import(1M)` 指令將媒體新增至目錄 (選用)

只有在程式庫配置為 `access=shared` 才需要執行本步驟。

如果程式庫配置為 `access=shared`，請執行 `import(1M)` 指令將媒體新增至目錄。

▼ 匯出卡匣

1. 使用 `export(1M)` 指令匯出卡匣。

本指令將媒體移至 I/O 區域，並開啓作業面板上的輸出模式燈號。

2. 實際將媒體從 I/O 區域移除。

Sony 直接連結式 8400 PetaSite 自動程式庫

Sony 8400 PetaSite 系列自動程式庫與其他 Sony 型號不同，因為它有一個八個插槽的匯入與匯出信箱 (插槽 400–407)。由於此原因，此系統中的匯入與匯出作業更為簡單直接。此自動程式庫使用條碼讀取器。

因為信箱插槽可以用來當做儲存插槽，Sun StorEdge SAM-FS 程式庫目錄檔會持續追蹤信箱插槽。

備註 – 本章節中的資訊僅適用於 Sony 直接連結式 8400 PetaSite 自動程式庫。本資訊與 Sony 直接連結式 B9 和 B35 自動程式庫無關，並且也跟第 206 頁「Sony 網路連結式自動程式庫」無關。

▼ 匯入磁帶

要匯入磁帶，請按照以下步驟執行。

1. 按下自動程式庫前面板上的開啟/關閉按鈕，以開啟自動程式庫的門蓋。
2. 將卡匣載入信箱插槽。
3. 按下自動程式庫前面板上的開啟/關閉按鈕，然後手動關閉信箱的門蓋。

在關閉門蓋之後，自動程式庫會檢查信箱插槽的卡匣條碼。如果條碼有問題，該插槽的 in 和 out 指示燈都將會閃爍。

4. 使用 `import(1M)` 指令使 Sun StorEdge SAM-FS 系統能夠識別匯入的卡匣。

請用以下格式使用此指令：

```
import eq
```

關於 `eq` 值，請指定要被定址的裝置之「設備序號」，如 `mcf` 檔案所定義。您也可以使用 File System Manager 執行此步驟。

匯出磁帶

匯出磁帶卡匣的程序將視是否將信箱插槽用作儲存插槽而定。

▼ 在不將信箱插槽當做儲存插槽的情況下匯出磁帶

當沒有將信箱插槽用作儲存插槽時，使用以下程序匯出卡匣。

1. 執行 `move(1M)` 指令以將卡匣移至信箱插槽 (插槽 400-407)。

請用以下格式使用此指令：

```
move 來源插槽 目標插槽 eq
```

表 8-6 `move(1M)` 指令的引數

引數	意義
來源插槽	表示卡匣目前所連接插槽的插槽號碼。
目標插槽	表示不應將卡匣移至插槽的插槽號碼。
<code>eq</code>	要被定址的裝置之「設備序號」，就如 <code>mc.f</code> 檔案所定義。

2. 按下自動程式庫前面板上的開啟/關閉按鈕。
門蓋將會開啓。
3. 移除信箱插槽中的卡匣。
4. 按下自動程式庫前面板上的開啟/關閉按鈕，然後手動關閉信箱的門蓋。
5. 執行 `samexport(1M)` 指令使 **Sun StorEdge SAM-FS** 系統能夠識別匯出的卡匣。
請用以下格式使用此指令：

```
samexport eq
```

關於 `eq` 值，請指定要被定址的裝置之「設備序號」，如 `mc.f` 檔案所定義。
您也可以使用 **File System Manager** 執行此步驟。

▼ 將信箱插槽用作儲存插槽以匯出磁帶

在將信箱插槽用作儲存插槽，並且要匯出的卡匣在其中一個信箱插槽中時，使用以下程序匯出卡匣。

1. 按下自動程式庫前面板上的開啟/關閉按鈕。
門蓋將會開啓。

2. 移除信箱插槽中的卡匣。
3. 按下自動程式庫前面板上的開啟/關閉按鈕，然後手動關閉信箱的門蓋。
4. 執行 `samexport(1M)` 指令使 Sun StorEdge SAM-FS 系統能夠識別匯出的卡匣。
請用以下格式使用此指令：

```
samexport eq
```

關於 `eq` 值，請指定要被定址的裝置之「設備序號」，如 `mcf` 檔案所定義。
您也可以使用 File System Manager 執行此步驟。

▼ 如何將卡匣移至不同插槽

要將卡匣移至不同插槽，請按照以下步驟執行：

1. 確定來源插槽已被佔用，並且目標插槽是空的。
2. 執行 `move(1M)` 指令。
請用以下格式使用此指令：

```
move eq:來源插槽 目標插槽
```

表 8-7 `move(1M)` 指令的引數

引數	意義
<code>eq</code>	要被定址的裝置之「設備序號」，就如 <code>mcf</code> 檔案所定義。
來源插槽	表示卡匣目前所連接插槽的插槽號碼。
目標插槽	表示不應將卡匣移至插槽的插槽號碼。

您也可以使用 File System Manager 執行此步驟。

Sony 網路連結式自動程式庫

如果擁有 Sony 網路連結式自動程式庫，請使用本節的程序以匯入與匯出卡匣。這些程序與第 7 頁「使用自動程式庫和手動載入磁碟機」中說明的不同。

因為您實際上使用供應商提供的公用程式，來新增和移除 Sony 自動程式庫中的卡匣，Sun StorEdge SAM-FS 介面 (`import(1M)`、`samexport(1M)` 和 `File System Manager`) 只會影響程式庫目錄。

備註 – 在 x64 硬體平台上的 Sun StorEdge SAM-FS 軟體不支援 Sony 網路連結式程式庫。

▼ 匯入卡匣

要匯入卡匣，請執行以下步驟：

1. 使用 `Sony` 指令以將卡匣實際移至程式庫。
2. 使用 `import(1M)` 指令更新程式庫目錄。

請用以下格式使用此指令：

```
import -v [ " ] volser [ " ] eq
```

表 8-8 `import(1M)` 指令的引數

引數	意義
" "	表示引號。如果 <code>volser</code> 包含空格，則必須加上引號。
<code>volser</code>	欲新增的 <code>volser</code> 。在使用新項目更新程式庫目錄之前，PSC API 介面會驗證 Sony 自動程式庫具有 <code>volser</code> 資訊。如果卡匣並非在程式庫中實際存在，項目將儲存在記錄器目錄中。
<code>eq</code>	要被定址的程式庫之「設備序號」，如 <code>mcf</code> 檔案所定義。

▼ 匯出卡匣

要匯出卡匣，請執行以下步驟：

1. 使用 `samexport(1M)` 指令從程式庫目錄移除項目。
請用下列其中一種格式使用此指令：

```
samexport eq:插槽  
samexport 媒體類型.vsn
```

表 8-9 samexport(1M) 指令的引數

引數	意義
<code>eq</code>	要被定址的裝置之「設備序號」，就如 <code>mcf</code> 檔案所定義。
插槽	表示在程式庫目錄檔中識別的自動程式庫中儲存插槽的號碼。
媒體類型	表示媒體類型。如需有效媒體類型清單，請參閱 <code>mcf(4)</code> 線上手冊。
<code>vsn</code>	表示指定給磁碟區的磁碟區序列名稱。

`samexport(1M)` 指令會在匯入每個 VSN 時更新程式庫目錄，並將每個 VSN 的程式庫目錄項目從程式庫目錄移至記錄器。

2. 使用 `Sony` 指令以將卡匣實際移出程式庫。

StorageTek ACSLS 連結式自動程式庫

如果擁有 StorageTek ACSLS 網路連結式自動程式庫，請使用本節的程序以匯入與匯出卡匣。這些程序與第 7 頁「使用自動程式庫和手動載入磁碟機」中說明的不同。

信箱是用於放入卡匣與將卡匣移出自動程式庫的區域。某些 StorageTek 自動程式庫一次只能匯入與匯出一個卡匣。Sun StorEdge SAM-FS 環境支援之使用信箱的 StorageTek 自動程式庫範例包含 StorageTek 9714 和 StorageTek 9710。StorageTek 9730 使用郵件插槽。在 StorageTek 文件資料中，信箱與信箱插槽經常稱為 `CAP`。

當從 ACSLS 連結式自動程式庫匯入和匯出卡匣時，請注意以下事項：

- 當匯入卡匣時，Sun StorEdge SAM-FS 指令只會影響程式庫目錄。`import(1M)` 指令不會將卡匣實際插入自動程式庫。必須使用 ACSLS 指令才能實際匯入卡匣。

- 當匯出卡匣時，Sun StorEdge SAM-FS 指令只會影響程式庫目錄，除非您同時使用 `samexport(1M)` 指令的 `-f` 選項。使用 `-f` 選項會指示 Sun StorEdge SAM-FS 系統將磁碟區置於 Cartridge Access Port (CAP) 並且更新對應的目錄。如果不指定 `-f` 選項，目錄會更新，但是因為磁碟區並未置於 CAP 中，所以仍然必須使用 ACSL S 指令實際匯出卡匣。

您的責任是持續將 ACSL S 庫存與 Sun StorEdge SAM-FS 目錄保持一致。

您也可以使用 `samu(1M)` 或 File System Manager 執行匯入與匯出程序。

▼ 匯入磁帶

- 若要匯入磁帶卡匣，請使用 `import(1M)` 指令。

請用以下格式使用此指令：

```
import -v vsn eq
```

表 8-10 `import(1M)` 指令的引數

引數	意義
<code>vsn</code>	表示指定給磁碟區的磁碟區序列名稱。
<code>eq</code>	要被定址的裝置之「設備序號」，就如 <code>mcf</code> 檔案所定義。

`import(1M)` 指令會導致新的 VSN 出現在程式庫目錄中。如果記錄器中已存在 VSN，Sun StorEdge SAM-FS 軟體會將 VSN 資訊從記錄器移至程式庫目錄。

▼ 使用信箱匯出磁帶

可使用插槽或 VSN 匯出磁帶卡匣。

- 若要匯出磁帶卡匣，請使用 `samexport(1M)` 指令。

請用下列其中一種格式使用此指令：

```
samexport [-f] eq:插槽  
samexport [-f] 媒體類型.vsn
```

表 8-11 samexport(1M) 指令的引數

引數	意義
-f	指示 Sun StorEdge SAM-FS 系統將磁碟區置於 Cartridge Access Port (CAP) 並且更新對應的目錄。
eq	要被定址的裝置之「設備序號」，就如 mcf 檔案所定義。
插槽	表示在程式庫目錄檔中識別的自動程式庫中儲存插槽的號碼。
媒體類型	表示媒體類型。如需有效媒體類型清單，請參閱 mcf(4) 線上手冊。
vsn	表示指定給磁碟區的磁碟區序列名稱。

samexport(1M) 指令會在匯入每個 VSN 時更新程式庫目錄，並將每個 VSN 的程式庫目錄項目從程式庫目錄移至記錄器。

字彙表

D

DAU 磁碟分配單元線上儲存裝置的基本單元。也稱為區塊大小。

F

FDDI 光纖分散式資料介面是區域網路中的資料傳輸標準，其範圍最多可延伸至 200 公里 (124 英哩里)。FDDI 通訊協定以記號環網路通訊協定為基礎。

FTP 檔案傳輸協定。在兩個主機之間透過 TCP/IP 網路傳送檔案的網際網路通訊協定。

I

inode 索引節點。檔案系統用來描述檔案的資料結構。inode 描述與檔案關聯的所有屬性，不包含名稱。屬性包含所有權、存取、權限、大小及檔案在磁碟系統上的位置。

inode 檔案 一個在檔案系統上的特殊檔案 (.inodes)，包含所有位在檔案系統中的檔案之 inode 結構。Inode 的長度是 512 位元組。Inode 檔案是中介資料檔案，與檔案系統中的檔案資料分隔開。

L

- LAN** 區域網路。
- LUN** 邏輯單元號碼。

M

- mcf** 主要配置檔案。這是在初始化時間讀取的檔案，此檔案將定義一個檔案系統環境中的裝置之間的關係 (拓撲)。

N

- NFS** 網路檔案系統。由 Sun 發行的一種檔案系統，提供連接到異質網路上的遠端檔案系統之透明存取。
- NIS** Sun OS 4.0 (至少) Network Information Service (網路資訊服務)。包含有關網路上的系統和使用者的主要資訊的一種分散式網路資料庫。NIS 資料儲存在主要伺服器 and 所有從屬伺服器上。

R

- RAID** 備援磁碟陣列。磁碟技術可使用數個獨立磁碟以穩定地儲存檔案。其可保護資料免於單一磁碟故障所造成的資料遺失，它可以提供容錯磁碟環境，而且可以提供比個別磁碟更高的流量。
- RPC** 遠端程序呼叫。NFS 所使用的基本資料交換機制，以實行自訂的網路資料伺服器。

S

SAM-QFS 將 Sun StorEdge SAM-FS 軟體與 Sun StorEdge QFS 檔案系統結合的配置。SAM-QFS 為使用者與管理員提供高速、標準的 UNIX 檔案系統介面，以搭配儲存與歸檔管理公用程式使用。其使用 Sun StorEdge SAM-FS 指令集中的許多指令，以及標準 UNIX 檔案系統指令。

samfsdump 可建立控制結構傾印的程式，並為特定檔案群組複製所有的控制結構資訊。其與 UNIX tar(1) 公用程式類似，但是通常不會複製檔案資料。請同時參閱 *samfsrestore*。

samfsrestore 可從控制結構傾印復原 inode 與目錄資訊的程式。請同時參閱 *samfsdump*。

SCSI 小型電腦系統介面。電子通訊規格一般用於週邊裝置，例如：磁碟機、磁帶機及自動程式庫。

Sun SAM-Remote

伺服器 既是全容量 Sun StorEdge SAM-FS 儲存管理伺服器，也是 Sun SAM-Remote 伺服器常駐程式，將定義在 Sun SAM-Remote 用戶端之間共用的程式庫。

Sun SAM-Remote

用戶端 具有用戶端常駐程式的 Sun StorEdge SAM-FS 系統，包含一些虛擬裝置，也可以擁有本身的程式庫裝置。用戶端需要 Sun SAM-Remote 伺服器才能為一個或多個歸檔備份進行媒體歸檔。

T

tar 磁帶歸檔。歸檔影像使用的標準檔案和資料記錄格式。

TCP/IP 傳輸控制協定/網際網路協定。網際網路協定負責主機對主機的定址與路由、封包傳送 (IP) 及在應用程式點 (TCP) 之間穩定傳送資料。

V

VSN 磁碟區序列名稱。若要歸檔到可移除式媒體卡匣，VSN 是指寫入到磁碟區標籤的磁帶與光碟之間的邏輯識別碼。在歸檔到磁碟快取的環境中，其為磁碟歸檔組的唯一名稱。

W

WORM 寫入一次，多次讀取。媒體儲存裝置歸類是指只能寫入一次，但可以讀取多次的媒體。

一畫

乙太網路 一種本機區域、封包交換的網路技術。原先的設計使用同軸電纜，現在都使用遮蔽式雙絞線。乙太網路是每秒 10 或 100 MB 的 LAN。

三畫

小型電腦系統介面 請參閱 SCSI。

四畫

中介資料 有關資料的資料。中介資料是用來尋找磁碟上的檔案之正確資料位置的索引資訊。它包含有關檔案、目錄、存取控制清單、符號連結、可移除式媒體、區段檔案及區段檔案索引等資訊。

中介資料裝置 一個儲存檔案系統中介資料的裝置 (例如：實體狀態磁碟或鏡像裝置)。將檔案資料和中介資料儲存在不同的裝置上可以增進效能。在 **mcf(4)** 檔案中，中介資料裝置被宣告為 **ma** 檔案系統中的 **mm** 裝置。

分割區 裝置的一部分或磁光碟匣的一面。

卡匣 包含用於錄製資料的媒體之實際實體，例如磁帶或光碟。有時亦稱為媒體、磁碟區或媒介。

五畫

可定址儲存裝置	包含使用者透過 Sun StorEdge QFS 或 Sun StorEdge SAM-FS 檔案系統來參照的線上、近線、遠端及離線儲存裝置的儲存空間。
可移除式媒體檔案	特殊使用者檔案類型可以直接在其常駐的可移除式媒體卡匣上存取，例如：磁帶或光碟匣。也可用來寫入歸檔與分段備份檔案資料。
本機檔案系統	安裝在 Sun Cluster 系統的某個節點上的檔案系統，而且不對其他節點提供高度可用性。也是一個安裝在獨立式伺服器上的檔案系統。
用戶端伺服器	在一個分散式系統互動模式下，一方的程式傳送請求給另一方的程式，然後等待回應。發出請求的程式稱為用戶端。滿足回應的程式稱為伺服器。
目錄	指向檔案系統內其他檔案與目錄的檔案資料結構。
目錄檔	自動程式庫內的 VSN 記錄。每個自動程式庫都有一個目錄，而在一個站點中，所有自動程式庫會有一個記錄器。

六畫

光纖通道	ANSI 標準可指定裝置之間的高速序列通訊。光纖通道是 SCSI-3 內的一個匯流排架構。
全域指令	套用於所有檔案系統，且出現在第一個 <code>fs =</code> 行之前的歸檔器與釋放器指令。
名稱空間	一組檔案的中介資料部分，可以識別出檔案、檔案屬性及其儲存位置。
回收器	Sun StorEdge SAM-FS 公用程式可重新取得卡匣上由過期的歸檔備份所佔用的空間。
多重讀取器檔案系統	這是一種單一寫入器、多重讀取器的功能，可讓您指定一個可掛載在多個主機上的檔案系統。多部主機可以讀取該檔案系統，但是只有一部主機可以寫入檔案系統。要指定多重讀取器，請使用 <code>mount(1M)</code> 指令及其 <code>-o reader</code> 選項。要指定單一寫入器主機，則使用 <code>mount(1M)</code> 指令及其 <code>-o writer</code> 選項。如需有關 <code>mount(1M)</code> 指令的更多資訊，請參閱 <code>mount_samfs(1M)</code> 線上手冊。
自動程式庫	一種自動裝置，可自動載入與卸載可移除式媒體卡匣，不需操作者介入。一個自動程式庫包含一個或多個磁帶機，而且含有傳輸機制，可以在儲存插槽與磁帶機之間來回移動卡匣。

七畫

- 分段備份** 從歸檔儲存裝置將近線或離線檔案複製回線上儲存裝置的程序。
- 系列組** 一個儲存裝置代表一個獨立實體裝置群組，例如：一組磁碟或自動程式庫內的磁帶機。另請參閱**儲存裝置系列組**。
- 系列裝置組** 請參閱**系列組**。

八畫

- 延伸陣列** 在一個檔案的 `inode` 之內的陣列為定義每個指定給檔案的資料區塊之磁碟位置。
- 直接 I/O** 用於大型區塊對齊循序式 I/O 的屬性。`setfa(1)` 指令的 `-D` 選項為直接 I/O 選項。它會將檔案或目錄設成直接 I/O 屬性。如果套用於某個目錄，直接 I/O 屬性將會被繼承。
- 直接存取** 檔案屬性 (永不分段備份)，表示近線檔案可以直接從歸檔媒體中存取，不需要擷取到磁碟快取記憶體。
- 直接連結式程式庫** 使用 SCSI 介面直接與伺服器連接的自動程式庫。SCSI 連結式程式庫直接由 Sun StorEdge SAM-FS 軟體控制。
- 近線儲存裝置** 必須先自動掛載才可以存取的可移除式媒體儲存裝置。近線儲存裝置通常比線上儲存裝置的價格便宜，不過它將耗用更多的存取時間。

九畫

- 軟式限制** 在磁碟配額上，使用者可以暫時超過的檔案系統資源 (區塊與 `inodes`) 的臨界值限制。超過軟式限制的話會啟動計時器。當您超過軟式限制並到達所指定的時間，除非您減少檔案系統的使用率到軟式限制以下，否則將無法分配更多的系統資源。
- 計時器** 配額軟體會記錄使用者達到軟式限制的開始時間，以及對使用者實行強制限制的結束時間。

十畫

- 核心** 提供基本系統功能的中央控制程式。**UNIX** 核心會建立與管理程序，提供存取檔案系統的功能、提供一般性安全性，以及供應通訊功能。
- 租約** 一個讓用戶端主機取得在特定的時間內對某個檔案執行作業的權限的功能。程序檔伺服器會發出租約給每個用戶端主機。若要繼續進行檔案作業，您必須更新租約。
- 配額** 使用者獲准使用的系統資源量。

十一畫

- 區塊大小** 請參閱 *DAU*。
- 區塊分配對映** 一個點陣圖，分段備份磁碟上的每個可用儲存區塊，並指出區塊已被使用或尚未被佔用。
- 強制限制** 對於磁碟配額，此為使用者不能超過的檔案系統資源、區塊和 **inode** 的最大限制。
- 掛載點** 掛載檔案系統的目錄。
- 連線** 兩個協定模組之間的路徑，提供穩定的串流傳送服務。**TCP** 連線會從一部機器的 **TCP** 模組延伸到另一部的 **TCP** 模組。

十二畫

- 備份儲存** 一組檔案的快照是要避免資料突然遺失。備份包含檔案的屬性與相關資料。
- 媒體** 磁碟或光碟卡匣。
- 媒體回收** 對具有很少使用中檔案的歸檔媒體進行回收或再使用的程序。
- 循環式** 資料存取方式會將整個檔案以循序的方式寫入邏輯磁碟上。當單一檔案寫入磁碟時，整個檔案會寫入第一個邏輯磁碟。第二個檔案會寫入下一個邏輯磁碟，依此類推。每個檔案的大小決定 **I/O** 的大小。
- 請同時參閱**磁碟資料平行儲存**以及**資料平行儲存**。
- 程式庫** 請參閱**自動程式庫**。
- 程式庫目錄** 請參閱**目錄**。

虛擬裝置	沒有相關硬體的軟體子系統或驅動程式。
超級區塊	檔案系統中定義檔案系統基本參數的資料結構。超級區塊將寫入到儲存裝置系列組的所有分割區，並識別該組中的分割區成員關係。
間接區塊	包含一系列儲存區塊的磁碟區塊。檔案系統擁有最多三個層級的間接區塊。第一層間接區塊包含一系列供資料儲存使用的區塊。第二層間接區塊包含一系列第一層間接區塊。第三層間接區塊包含一系列第二層間接區塊。

十三畫

裝置記錄	一種可配置的功能，提供裝置特定的錯誤資訊，以便分析裝置問題。
裝置掃描器	會定期監視所有手動掛載的可移除式裝置是否存在，並可由使用者或其他程序要求以偵測已掛載卡匣是否存在的軟體。
資料平行儲存	資料存取方式會以交錯的方式，同步將檔案寫入邏輯磁碟。SAM-QFS 檔案系統提供兩種資料平行儲存方式：「強制資料平行儲存」，使用資料平行儲存群組，以及「軟式資料平行儲存」，使用 <code>stripe=x</code> 掛載參數。設定檔案系統並且要求在 <code>mcf(4)</code> 檔案內的資料平行儲存群組定義時，即啟用強制資料平行儲存。軟式資料平行儲存則是透過 <code>stripe=x</code> 掛載參數啟用，而且可以就檔案系統或個別檔案做變更。設定 <code>stripe=0</code> 即可停用。若是檔案系統由多個具有相同元素數量的資料平行儲存群組組成，則強制與軟式資料平行儲存皆可使用。請同時參閱循環式。
資料平行儲存區大小	在寫入到下一個資料平行儲存裝置以前，所要分配的磁碟分配單元 (DAU) 數量。若是使用 <code>stripe=0</code> 掛載選項，檔案系統使用循環存取，而非資料平行儲存存取。
資料平行儲存群組	檔案系統內的一群裝置，在 <code>mcf(4)</code> 檔案中被定義為一個或多個 <code>gXXX</code> 裝置。資料平行儲存群組可視為一個邏輯裝置，通常會將資料平行儲存成與磁碟分配單元 (DAU) 的大小一樣。
資料裝置	在檔案系統中，儲存檔案資料的一個裝置或一組裝置。
預先分配	在磁碟快取上預留連續空間的程序，供寫入檔案使用。僅有其大小為零的檔案可以指定預先分配。如需更多資訊，請參閱 <code>setfa(1)</code> 線上手冊。

十四畫

磁碟快取記憶體	檔案系統軟體中的磁碟常駐部份，用來建立和管理線上磁碟快取與歸檔媒體之間的資料檔案。個別的磁碟分割區或一整個磁碟都可以用來做為磁碟快取記憶體。
磁碟空間臨界值	磁碟快取使用率的最大或最小層級是由管理者定義。釋放器會依據這些預先定義的磁碟空間臨界值，以控制磁碟快取使用率。

磁碟分配單元	請參閱 <i>DAU</i> 。
磁碟區	卡匣上供共用資料使用的命名區域。一個卡匣有一個或多個磁碟區。雙面卡匣有兩個磁碟區，一面一個。
磁碟區溢位	讓系統將單一檔案跨越到多個磁碟區的功能。在站點使用超過其個別卡匣容量的大型檔案時，磁碟區溢位是很有幫助的。
磁碟資料平行儲存	將檔案記錄到數個磁碟的程序，因此增進存取效能並提高整體的儲存容量。另請參閱資料平行儲存。
磁碟緩衝區	在 SUN SAM-Remote 配置中，這是在伺服器系統上用來從用戶端歸檔至伺服器時使用的緩衝區。
磁碟機	一個可以在可移除式媒體磁碟區中來回傳送資料的機制。
網路連結式自動程式庫	使用供應商所提供的套裝軟體加以控制的程式庫 (例如：來自 StorageTek、ADIC/Grau、IBM 或 Sony 之程式庫)。Sun StorEdge SAM-FS 檔案系統可以使用特別為自動程式庫設計的 Sun StorEdge SAM-FS 媒體抽換裝置常駐程式做為與供應商軟體之間的介面。
遠端程序呼叫	請參閱 <i>RPC</i> 。
遠端儲存裝置	距離伺服器較遙遠的儲存裝置，且用於災難復原。
自動裝置	在儲存插槽與磁帶機之間移動卡匣的自動程式庫組成部分。又稱為「 <i>transport</i> 」(傳輸器)。
寬限期	若是磁碟配額，這是允許使用者在到達軟式限制以後建立檔案和分配儲存空間的時間量。

十五畫

稽核 (完整)	載入卡匣以驗證其 VSN 的程序。對磁光碟匣而言，會判斷其容量與空間資訊，並放進自動程式庫的目錄中。
線上儲存裝置	立即可使用的儲存裝置，例如：磁碟快取儲存裝置。

十七畫

儲存插槽	磁帶機卡匣未在磁碟機中使用時在自動程式庫內的位置。如果程式庫是直接連結式，儲存插槽的內容就會被保留在自動程式庫的目錄中。
儲存裝置系列組	共同以單一磁碟系列裝置的形態分段備份的一組磁碟。

檔案系統 檔案與目錄的階層式集合。

檔案系統特定指令 跟隨 archiver.cmd 檔案中的全域指令之歸檔器和釋放器指令，為特定檔案系統的特定指令，且以 fs = 為開頭。在遇到下一個 fs = 指令行、或檔案結束之前，檔案系統特定指令均有效。如果有數個影響檔案系統的指令，則檔案系統特定指令會覆寫全域指令。

十八畫

歸檔媒體 歸檔檔案所寫入的媒體。歸檔媒體可以是可移除式磁帶或程式庫中的磁光碟匣。此外，歸檔媒體可以是另外一個系統上的掛載點。

歸檔器 自動控制將檔案複製到可移除式卡匣的歸檔程式。

歸檔儲存 建立在歸檔媒體上的檔案資料副本。

離線儲存裝置 需要操作者介入才能載入的儲存裝置。

十九畫

鏡像寫入 在無關的磁碟組上為檔案保存兩份副本的程序，以防止因其中一個磁碟發生故障而導致資料遺失。

二十畫

釋放器 Sun StorEdge SAM-FS 元件可識別已歸檔檔案並釋放其磁碟快取備份，因此得到更多可用的磁碟快取空間。釋放器會根據臨界值的高低自動對線上磁碟儲存裝置總量進行管制。

釋放優先權 在檔案系統中已歸檔的檔案被釋放的優先權。釋放優先權是根據檔案特性的不同權數之相乘，然後將結果加總來計算。

索引

英文字母

- access 歸檔器指令, 53
- ACSAPI 介面, 3
- ACSLs 連接的程式庫, 207
- ADIC/Grau 自動程式庫
作業, 197
- age_priority preview.cmd 指令, 123
- allsets 歸檔組, 30, 62
- archivemeta 歸檔器指令, 43
- archiver(1M) 指令, 40, 159
 - 範例, 86
 - 輸出範例, 36
- archiver.cmd 檔案, 29, 37, 84
 - access 和 -nftv 指令, 53
 - archivemeta 指令, 43
 - archmax 指令, 44, 62
 - bufsize 指令, 44, 62
 - drivemax, drivemin, 和磁碟機歸檔需求指令, 63
 - drives 指令, 45
 - endparams 指令, 61
 - fillvsns 歸檔需求參數, 64
 - fs 指令, 51
 - ignore 指令, 139
 - join 路徑參數, 66
 - logfile 指令, 47
 - minsize 和 maxsize 指令, 53
 - norelease 指令, 59
 - offline_copy 參數, 65
 - ovflmin 指令, 48
 - params 指令, 61
 - recycle_dataquantity 指令, 136
 - recycle_hwm 指令, 136
 - recycle_ignore 指令, 136, 139
 - recycle_mailaddr 指令, 137
 - recycle_mingain 指令, 137
 - recycle_vsncount 指令, 137
 - SAM-Remote 範例, 165
 - startage, startcount, 及 startsize 參數, 74
 - tapenonstop 參數, 68
 - vsn 關聯指令, 75
 - vsnpools 參數, 77
 - vsns 和 endvsns 參數, 75
 - 分段備份中的角色, 121
 - 全域指令, 43
 - 名稱指令, 54
 - 回收指令, 66, 133
 - 多重中介資料副本, 60
 - 自動取消歸檔, 60
 - 使用 SAM-Remote 配置回收, 163
 - 使用者和群組指令, 54
 - 保留參數, 69
 - 建立, 40
 - 指令, 41, 43
 - 配置回收器, 136
 - 通知指令, 48
 - 等候指令, 50
 - 間隔指令, 47
 - 資料平行儲存區段檔案, 190
 - 範例, 42

- 編輯 SAM-Remote, 155
- 複製數目指令, 58
- 優先權參數, 72
- 檔案系統指令, 50
- 檢查指令, 46
- 歸檔時間, 60
- 歸檔組指定, 51
- 簡介, 39
- 鎖定歸檔需求參數, 64
- 釋放和分段備份指令s, 56
- 釋放的角色, 111
- 釋放指令, 59
- archiver.sh(1M) 程序檔, 48
- archmax 歸檔器指令, 44, 62
- auditslot(1M) 指令, 15, 20
- bufsize 分段備份程式指令, 117
- bufsize 歸檔器指令, 44, 62
- chmed(1M) 指令, 17, 19, 20, 135, 182, 184
- cleandrive(1M) 指令, 19
- crontab 項目, 回收器, 139
- defaults.conf 檔案
 - exported_media 指令, 23, 26
 - 有人照料指令, 23, 26
 - 啓用裝置記錄檔, 187
- devlog 指令, 188
- devlog 檔案, 186
- diskvols.conf 檔案, 79
- display_all_candidates 釋放器指令, 108
- drivemax 歸檔器指令, 63
- drivemin 歸檔器指令, 63
- drives 分段備份程式指令, 117
- drives 歸檔組參數指令, 63
- drives 歸檔器指令, 45
- DZC-8000S 介面, 3
- endparams 歸檔器指令, 61
- endvsnpools 歸檔器指令, 77
- examine 歸檔器指令, 46
- export(1M) 指令, 202
- exported_media 指令, 23
- fillvsns 歸檔需求參數, 64
- fs 歸檔器指令, 51
- fs 釋放器指令, 107
- Fujitsu LMF 自動程式庫作業, 199
- group 歸檔器指令, 54
- hlwm_priority preview.cmd 檔案指令, 124
- hwm 回收器指令, 135
- hwm_archive 掛載選項, 47
- hwm_priority preview.cmd 檔案指令, 124
- IBM 3494 自動程式庫
 - 作業, 202
- IBM 3584 自動程式庫
 - 分割, 201
 - 作業, 201
 - 清潔, 201
- idle 指令, 10
- ignore 回收器指令, 136, 139
- import(1M) 指令, 17, 24, 198, 199, 202, 203, 206, 208
- interval 歸檔器指令, 47
- join path 歸檔器指令, 66
- lhwm_priority preview.cmd 檔案指令, 124
- list_size 釋放器指令, 111
- lmcpd 介面, 3
- load_notify.sh(1M) 指令檔, 25
- lock 歸檔器指令, 64
- log_rotate.sh(1M) 指令檔, 195
- Logfile 分段備份程式指令, 118
- Logfile 歸檔器指令, 47
- Logfile 釋放器指令, 109
- lwm_priority preview.cmd 檔案指令, 124
- mail 回收器指令, 136
- maxactive 分段備份程式指令, 121
- maxsize 歸檔器指令, 53
- mcf 檔案, 3
 - SAM-Remote 配置, 144
 - 程式庫記錄器, 23
- min_residence_age 釋放器指令, 108
- mingain 回收器指令, 135
- minsize 歸檔器指令, 53
- mount(1M) 指令
 - 部份釋放和分段備份選項, 102
- move(1M) 指令, 204, 205

- name 歸檔器指令, 54
- NFS 檔案共用, 142
- nftv 歸檔器指令, 53
- no_archive 歸檔組, 30
- no_recycle 回收器指令, 131, 166
- no_release 釋放器指令, 108
- no-data VSN, 179
- norelease 歸檔器指令, 59
- notify 歸檔器指令, 48
- o maxpartial 掛載選項, 102
- o partial_stage 掛載選項, 102
- o 部份掛載選項, 102
- odlabel(1M) 指令, 14, 180, 183, 184, 188
- offline_copy 歸檔器指令, 65
- ovflmin 歸檔器指令, 48
- params 歸檔器指令, 61
- partially full VSN, 181
- pkginfo(1M) 指令, 147
- pool 歸檔器指令, 75
- preview.cmd 檔, 122
 - age_priority 指令, 123
 - hlwm_priority 指令, 124
 - hwm_priority 指令, 124
 - lhwm_priority 指令, 124
 - lwm_priority 指令, 124
 - vsu_priority 指令, 123
 - 指令, 122
 - 設定優先, 125
- priority 歸檔器指令, 72
- research_no_release 釋放器指令, 110
- recycle_dataquantity 歸檔器指令, 136
- recycle_hwm 歸檔器指令, 136, 164
- recycle_ignore 歸檔器指令, 136, 139, 164
- recycle_mailaddr 歸檔器指令, 137
- recycle_mingain 歸檔器指令, 137, 164
- recycle_minopbs 百分比回收器指令, 80
- recycle_vsncount 歸檔器指令, 137, 164
- recycler.cmd 檔案
 - hwm 指令, 135
 - ignore 指令, 136, 139
 - logfile 指令, 131
 - mail 指令, 136
 - mingain 指令, 135
 - no_recycle 指令, 131
 - SAM-Remote, 159
 - 建立, 134
 - 配置 SAM-Remote, 165, 166
 - 程式庫指令, 132
 - 範例, 135
 - recycler.sh 程序檔, 139, 167
 - release 歸檔器指令, 56, 59
 - release(1) 指令, 99
 - 部分釋放, 104
 - releaser.cmd 檔案, 105, 112
 - display_all_candidates 指令, 108
 - fs 指令, 107
 - list_size 指令, 111
 - logfile 指令, 109
 - min_residence_age 指令, 108
 - no_release 指令, 108
 - research_no_release 指令, 110
 - weight_age 指令, 106
 - weight_age_access 指令, 106
 - weight_age_modify 指令, 106
 - weight_age_residence 指令, 106
 - weight_size 指令, 107
 - 檔案時間指令, 105
 - request(1) 指令, 133, 188
 - 引數, 189
 - reserve 歸檔器指令, 69
 - sam_release(3) 程式庫常式, 104
 - sam-amld 常駐程式, 122
 - sam-archiverd 常駐程式, 33 - 36, 37
 - sam-arcopy 程序, 37
 - 記錄檔, 37
 - sam-arfind 程序, 31, 37
 - 記錄檔, 37
 - samcmd(1M) 指令, 10
 - 卸載選項, 13, 24 - 25
 - 開啓選項, 11
 - 閒置選項, 26, 149
 - 載入選項, 12
 - 稽核選項, 16
 - 關閉選項, 12

- samd(1M) 指令, 10
 - 停止選項, 150
 - 啟動選項, 11, 156
- samexport(1M) 指令, 22, 24, 198, 200, 204, 207, 208
- samfsdump(1M) 指令, 85
- sam-genericd 常駐程式, 3
- sam-ibm3494d 常駐程式, 3
- sam-recycler(1M) 指令, 130, 133, 137, 167, 171, 182, 183, 184
- SAM-Remote
 - samu(1M) R 畫面, 157
 - 用戶端和伺服器互動, 145
 - 用戶端配置檔案, 151
 - 用戶端配置簡介, 144
 - 目錄檔, 158
 - 回收, 159
 - 回收指令, 164
 - 回收器用戶端配置, 162
 - 回收器伺服器配置, 160
 - 回收器排程, 180, 183
 - 安裝, 146
 - 伺服器配置檔案, 153
 - 伺服器配置簡介, 144
 - 技術簡介, 143
 - 限制, 143
 - 配置, 146
 - 配置回收, 162
 - 配置回收 (方法 2), 183
 - 配置範例, 146
 - 啟用歸檔, 155
 - 程式庫目錄, 145
 - 虛擬裝置, 145
 - 需求, 143
 - 編輯 archiver.cmd 檔案以配置回收, 163
 - 編輯 mcf 檔案, 149 - 151
 - 編輯 recycler.cmd 檔案, 165
 - 編輯伺服器 mcf 檔案, 152
 - 歸檔, 146
 - 簡介, 141
- sam-robotsd 常駐程式, 3
- samset(1M) 指令
 - 啟用裝置記錄檔, 187
- sam-sonyd 常駐程式, 3
- sam-stkd 常駐程式, 3
- samu(1M)
 - arrun 指令, 182
 - R 畫面, 157
 - s 畫面, 156
 - v 畫面, 158
- SEF, 191
 - sysevent 處理程式, 195
 - 記錄檔, 195
 - 報告輸出, 192
- sefdata 檔案, 191
- sefreport(1M) 指令, 191
 - 選項, 192
- segment(1) 指令, 190
- showqueue(1M) 指令, 32
- showrev(1M) 指令, 148
- Sony 8400 PetaSite 自動程式庫
 - 作業, 203
- Sony 網路連接的自動程式庫
 - 作業, 206
- stage 歸檔器指令, 56
- stager.cmd 檔案, 115
 - bufsize 指令, 117
 - drives 指令, 117
 - logfile 指令, 118
 - maxactive 指令, 121
 - 建立, 116
 - 範例, 121
- startage 歸檔器指令, 74
- startcount 歸檔器指令, 74
- startsize 歸檔器指令, 74
- StorageTek ACSLS 連接的自動程式庫
 - 作業, 207
- sysevent 功能, 195
- syseventadm(1M) 指令, 196
- tapealert(1M)
 - 與 SEF, 191
- tapenonstop 歸檔器指令, 68
- tplabel(1M) 指令, 13, 180, 183, 184, 188
- user 歸檔器指令, 54
- VSN
 - pool 指令, 77

- 池, 範例, 95
- 使用常規表示式, 76
- 最低回收增益, 135
- 關聯指令, 75
- vsn_priority preview.cmd 指令, 123
- vsnpools 歸檔器指令, 77
- wait 歸檔器指令, 50
- weight_age 釋放器指令, 106
- weight_age_access 釋放器指令, 106
- weight_age_modify 釋放器指令, 106
- weight_age_residence 釋放器指令, 106
- weight_size 釋放器指令, 107
- wm_priority preview.cmd 檔案指令, 124

四畫

- 中介資料副本, 60
- 分段備份程式
 - 分段備份中的歸檔角色, 121
 - 分段備份請求的數目設定, 121
 - 定義的, 115
 - 指令, 115
 - 指定磁碟機數目, 117
 - 記錄作業, 118
 - 記錄檔案欄位, 120
 - 強制分段備份程式請求, 126
 - 設定分段備份緩衝區大小, 117
 - 部分分段備份, 102
 - 錯誤程序, 2
 - 簡介, 2, 115
- 文件, xx

五畫

- 卡匣
 - 卸載, 13, 26
 - 清潔, 16
 - 移除, 20
 - 匯入, 23, 24
 - 匯入與匯出, 22
 - 匯出, 24, 25
 - 載入, 12, 26

- 製作標籤, 13
- 可用空間, 已定義, 129
- 可移除的媒體
 - 指令, 8
 - 停止, 10
 - 啟動, 11
 - 檔案, 188
- 正規表示式, 54
- 目前資料, 已定義, 129
- 目錄, 簡介, 22

六畫

- 回收指令, 66
- 回收器
 - crontab 項目, 139
 - recycler.cmd 檔案範例, 135
 - recycler.sh 程序檔, 139
 - 方法, 130
 - 回復歸檔磁碟歸檔, 80
 - 作業理論, 130
 - 防止回收, 131
 - 忽略程式庫, 136
 - 指令, 131
 - 指定自動程式庫的回收, 132
 - 指定記錄檔, 131
 - 配置, 133
 - 高值參數指令, 135
 - 強制使用 chmed(1M) 指令, 135
 - 最低 VSN 增益指令, 135
 - 郵件通知選項, 136
 - 磁碟歸檔備份, 130
 - 編輯 archiver.cmd 檔案, 136
 - 簡介, 2, 129
- 回收器記錄檔, 167, 171, 179
 - no-data VSN, 179
 - partially full VSN, 181
- 有人照料指令, 23
- 自動清潔, 19
- 自動程式庫
 - ADIC/Grau, 197
 - Fujitsu LMF, 199
 - IBM 3494, 202

IBM 3584, 201

SCSI 連結式，請參閱自動程式庫，直接連結式
Sony 8400 PetaSite, 203

Sony 網路連接, 206

目錄檔, 22

作業, 22

供應商指定程序, 197

定義的, 7

直接連結式, 2

指令, 8

指定回收參數, 132

記錄器, 22

專有名詞, 9

常駐程式, 3

清潔, 18

透過 ACSLS 連接的 StorageTek, 207

開啓, 11

匯入與匯出, 23

網路連結式, 2

稽核, 16

關閉, 11

自動裝置，請參閱自動程式庫

七畫

低值參數, 99

preview.cmd 檔案指令, 124

技術支援, xxi

系統錯誤功能, 請參閱 SEF

八畫

使用 SAM-Remote 歸檔, 146, 155

使用 Sun SAM-Remote 進行回收, 159

卸載媒體, 13

取消歸檔, 60, 67

九畫

信箱, 23

指令

歸檔, 43

重新歸檔, 已定義, 130

十畫

容量, 已定義, 129

記錄感應頁, 191

記錄器, 22

記錄檔回收指令, 131

記錄檔案

分段備份程式, 118

回收器, 167, 171, 179

回收器錯誤訊息, 137

啓用裝置記錄, 186

備份, 48, 85

裝置記錄, 185

管理 SEF 記錄檔, 195

請參閱記錄檔, 191

歸檔器, 37, 47

釋放器, 109

訊息檔案, 137

高值參數, 99

preview.cmd 檔案指令, 124

具有回收器, 135

十一畫

區段檔案, 190

歸檔, 190

參數預覽請求指令, 124

常駐程式

sam-amld, 122

sam-archiverd, 33

samarchiverd, 37

sam-genericd, 3

sam-ibm3494d, 3

sam-robotd, 3

sam-sonyd, 3

sam-stkd, 3

自動程式庫常駐程式, 3

授權

一般資訊, xxi

條碼

清潔卡匣, 17

- 清潔卡匣, 16
- 清潔週期, 重設, 16
- 清潔磁帶機, 18
- 處理分段備份請求時發生錯誤, 2
- 軟體
 - 文件, xx
- 連續歸檔, 46
- 部分釋放, 99
 - 使用者選項, 104
 - 簡介, 102

十二畫

- 媒體
 - 卸載, 13
 - 強制優先, 127
 - 移動, 22
 - 程式庫, 請參閱自動程式庫
 - 載入, 12
 - 錯誤, 19
- 程式庫目錄
 - 簡介, 22
- 程式庫目錄檔
 - 檢視, 27
- 程式庫回收指令, 132
- 程式庫記錄器, 22

十三畫

- 匯入媒體, 簡介, 22
- 裝置記錄, 185
 - 何時使用, 186
 - 事件, 186
 - 啓用, 186
- 載入通知, 啓用, 25
- 載入媒體, 12
 - 手動載入磁碟機, 26
- 過期資料, 已定義, 129
- 預設
 - 歸檔器, 30
- 預覽佇列, 85

- 預覽請求
 - VSN 決定優先權, 123
 - 計畫, 125
 - 計算優先權, 125
 - 時間決定優先權, 123
 - 配置範例, 126
 - 高和低值參數決定優先權, 124
 - 參數指令, 124
- 預覽請求的 VSN 優先權, 123

十四畫

- 磁光裝置, 請參閱自動程式庫
- 磁帶清潔設定, 19
- 磁帶儲存裝置, 請參閱自動程式庫
- 磁碟快取記憶體
 - 釋放優先權, 2
- 磁碟區溢位
 - ovflmin 歸檔器指令, 48
 - 範例, 49
 - 檔案, 188
- 磁碟機, 清潔, 16, 18
- 磁碟歸檔, 78
 - 指令, 79
 - 配置, 79
 - 啓用, 81
 - 範例, 82
- 製作卡匣標籤, 13

十五畫

- 稽核
 - 自動程式庫, 16
 - 磁碟區, 15
- 請求檔案, 請參閱可卸除式媒體檔案

十六畫

- 獨立磁碟機
 - 載入媒體, 26
- 錯誤訊息, 針對回收器, 137

十七畫

檔案, 設定屬性, 56

檔案系統

簡介, 1

檔案系統管理員

建立帳號, 5

管理遠端伺服器, 6

簡介, 3

十八畫

歸檔記錄

備份, 85

歸檔組

名稱, 52

搜尋條件, 52

路徑, 52

複製數目, 58

檔案屬性, 52

歸檔需求, 33

排程, 34

歸檔需求檔案, 31

歸檔器

allsets 歸檔組, 62

-archmax 參數, 35

-drivemin 參數, 34

-drives 參數, 34

-fillvsns 參數, 35

-join 參數, 33

-ovflmin 參數, 35

-sort 和 -rsort 參數, 34

VSN pool, 77

VSN 關聯指令, 75

已掃描的歸檔, 33

分段備份中的角色, 121

在 archiver.cmd 中指定檔案系統, 51

作業理論, 29

作業簡介, 31

決定歸檔時間, 67

使用常規表示式, 54

定義的, 29

延遲歸檔器啟動, 50

阻止歸檔, 52

保留參數, 33

指引, 84

指令, 41, 43

指定檔案系統資料的備份, 60

指定歸檔記錄檔案, 47

指定歸檔間隔, 47

指定歸檔需求的磁碟機數目, 63

指定歸檔緩衝鎖定, 64

重新命名事件通知程序檔, 48

記錄檔案, 37

記錄檔範例, 37

區段檔案, 190

常駐程式, 37

強制歸檔請求, 126

控制回復歸檔, 67

控制使用的磁碟機數量, 45

控制磁碟區溢位, 48

控制檔案大小, 44

控制歸檔掃描, 46

控制歸檔檔案的大小, 62

排程歸檔需求, 34

啟用磁碟歸檔, 81

設定自動取消歸檔, 60

設定歸檔時間, 60

設定歸檔器緩衝區大小, 44, 62

設定屬性, 72

連續歸檔, 32, 46

備份定義指令, 58

預留的 VSN, 70

預留磁碟區, 69

預設, 30

預覽佇列, 85

磁碟歸檔, 78

磁碟歸檔指令, 79

磁碟歸檔配置, 79

範例, 86

辨識要歸檔的檔案, 31

歸檔一個歸檔需求, 36

歸檔時間定義, 30

歸檔排程, 74

歸檔組, 30

歸檔組成員, 51

歸檔組成員衝突, 57

歸檔組參數, 33

- 歸檔組處理指令, 61
- 歸檔媒體定義, 1, 29
- 歸檔間隔定義, 30
- 歸檔需求, 33
- 歸檔優先權, 32
- 簡介, 1
- 關聯歸檔, 66

十九畫

- 關聯歸檔, 66

二十畫

- 釋放與回復釋放指令, 同時使用, 60

釋放器

- archiver.cmd 檔案角色, 111
- fs 指令, 107
- 手動作業, 114
- 作業理論, 100
- 指令, 105
- 指令檔案, 105
- 候選者定義, 101
- 記錄檔, 109
- 配置, 112
- 部分釋放, 99, 102
- 部份釋放, 使用者選項, 104
- 部份釋放選項, 103
- 優先權, 101
- 檔案時間, 100
- 簡介, 2, 99
- 釋放優先權指令, 105
- 權數, 101

