



Sun StorageTek™ Storage Archive Manager (SAM) 归档配置和管理指南

Version 4, Update 6

Sun Microsystems, Inc.
www.sun.com

文件号码 820-1728-10
2007 年 5 月, 修订版 A

请将有关本文档的意见和建议提交至: <http://www.sun.com/hwdocs/feedback>

版权所有 2007 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 保留所有权利。

对于本文档中介绍的产品，Sun Microsystems, Inc. 对其所涉及的技术拥有相关的知识产权。需特别指出的是（但不局限于此），这些知识产权可能包含在 <http://www.sun.com/patents> 中列出的一项或多项美国专利，以及在美国和其他国家/地区申请的一项或多项其他专利或待批专利。

本文档及其相关产品的使用、复制、分发和反编译均受许可证限制。未经 Sun 及其许可方（如果有）的事先书面许可，不得以任何形式、任何手段复制本产品或文档的任何部分。

第三方软件，包括字体技术，均已从 Sun 供应商处获得版权和使用许可。

本产品的某些部分可能是从 Berkeley BSD 系统衍生出来的，并获得了加利福尼亚大学的许可。UNIX 是 X/Open Company, Ltd. 在美国和其他国家/地区独家许可的注册商标。

Sun、Sun Microsystems、Sun 徽标、AnswerBook2、docs.sun.com、Solaris、Java 和 Sun StorageTek 是 Sun Microsystems, Inc. 在美国和其他国家/地区的商标或注册商标。

所有 SPARC 商标的使用均已获得许可，它们是 SPARC International, Inc. 在美国和其他国家/地区的商标或注册商标。标有 SPARC 商标的产品均基于由 Sun Microsystems, Inc. 开发的体系结构。

Mozilla 是 Netscape Communications Corporation 在美国和其他国家/地区的商标或注册商标。

OPEN LOOK 和 Sun™ 图形用户界面是 Sun Microsystems, Inc. 为其用户和许可证持有者开发的。Sun 感谢 Xerox 在研究和开发可视或图形用户界面的概念方面为计算机行业所做的开拓性贡献。Sun 已从 Xerox 获得了对 Xerox 图形用户界面的非独占性许可证，该许可证还适用于实现 OPEN LOOK GUI 和在其他方面遵守 Sun 书面许可协议的 Sun 许可证持有者。

美国政府权利 - 商业用途。政府用户应遵循 Sun Microsystems, Inc. 的标准许可协议，以及 FAR（Federal Acquisition Regulations，即“联邦政府采购法规”）的适用条款及其补充条款。

本文档按“原样”提供，对于所有明示或默示的条件、陈述和担保，包括对适销性、适用性或非侵权性的默示保证，均不承担任何责任，除非此免责声明的适用范围在法律上无效。



请回收



Adobe PostScript

目录

前言 xix

1. 概述 1

性能 1

存储设备 3

Sun SAM-Remote 软件 4

使用 Sun Cluster 实现高可用性配置 (HA-SAM) 4

File System Manager 4

▼ 调用 File System Manager 5

设置会话超时 5

创建附加的管理员和用户帐户 6

▼ 创建附加帐户 6

指定权限级别 6

创建多用户帐户 7

通过 File System Manager 管理其他服务器 8

2. 使用自动化库和手动载入的驱动器 9

自动化库操作 10

- ▼ 启动可移除介质的操作 10

- ▼ 停止可移除介质的操作 11

- ▼ 打开自动化库 11

- ▼ 关闭自动化库 12

将卡盒载入自动化库 12

- ▼ 手动载入卡盒 12

从驱动器中卸载卡盒 13

- ▼ 手动卸载卡盒 13

标记卡盒 13

- ▼ 标记或重新标记磁带 14

- ▼ 标记或重新标记光盘 14

- ▼ 审计卷 15

- ▼ 审计直接连接的自动化库 15

使用清洁卡盒 16

- ▼ 使用具有条码的清洁卡盒 16

- ▼ 使用没有条码的清洁卡盒 17

- ▼ 重置清洁循环次数 18

- ▼ 限制清洁循环次数 18

- ▼ 清洁磁带机 19

磁带机自动清洁 19

- ▼ 清除介质错误 20

- ▼ 从驱动器中取出卡住的卡盒 21

目录操作和卡盒的导入与导出	22
跟踪导出的介质 - 历史记录	23
对自动化库执行导入和导出操作	24
▼ 向使用邮箱的库中导入卡盒	24
▼ 从使用邮箱的库中导出卡盒	25
▼ 向不使用邮箱的库中导入卡盒	25
▼ 从不使用邮箱的库中导出卡盒	26
▼ 启用载入通知	26
使用具备加密功能的驱动器	27
手动载入驱动器操作	27
▼ 载入卡盒	27
▼ 卸载卡盒	28
▼ 查看库目录	28
3. 归档	29
归档过程概述	29
归档程序的守护进程	30
归档集	30
归档操作	31
步骤 1: 识别要归档的文件	32
步骤 2: 组合归档请求	34
步骤 3: 预定归档请求	35
步骤 4: 对归档请求中的文件进行归档	37
默认输出样例	38
归档日志文件和事件日志	39
数据验证	41
关于 archiver.cmd 文件	41
▼ 创建或修改 archiver.cmd 文件并传播更改	42
archiver.cmd 文件	43
archiver.cmd 文件示例	44

使用归档程序指令 46

全局归档指令 46

`archivemeta` 指令: 控制是否对元数据进行归档 47

`archmax` 指令: 控制归档文件的大小 47

`bufsize` 指令: 设置归档程序缓冲区大小 48

`drives` 指令: 控制用于归档的驱动器数 49

`examine` 指令: 控制归档扫描 49

`interval` 指令: 指定归档时间间隔 50

`logfile` 指令: 指定归档程序日志文件 51

▼ 备份归档程序日志文件 51

`notify` 指令: 重命名事件通知脚本 51

`ovflmin` 指令: 控制卷溢出 52

`scanlist_squash` 指令: 控制扫描列表合并 53

`setarchdone` 指令: 控制 `archdone` 标志的设置 54

`wait` 指令: 延迟归档程序的启动 54

文件系统指令 54

`fs` 指令: 指定文件系统 55

全局指令和文件系统指令 55

归档集分配指令 55

指定归档集 56

文件时限 *search-criterion*: `-access` 和 `-nftv` 57

文件时限 *search-criterion*: `-after` 57

文件大小 *search-criterion*: `-minsize` 和 `-maxsize` 58

属主和组 *search-criterion*: `-user` 和 `-group` 58

使用模式匹配的文件名 *search-criterion*: `-name regex` 59

释放和登台 *file-attributes*: `-release` 和 `-stage` 61

归档集成员冲突 62

归档副本指令 63

归档之后释放磁盘空间: `-release` 63

延迟释放磁盘空间: `-norelease` 64

同时使用 `-release` 和 `-norelease` 64

设置归档时限 65

自动取消归档 65

为元数据指定多个副本 65

归档集副本参数 66

控制归档文件的大小: `-archmax` 67

设置归档程序缓冲区大小: `-bufsize` 67

指定归档请求的驱动器数: `-drivemax`、`-drivemin` 和 `-drives` 68

最大化卷上的空间: `-fillvsns` 69

指定归档缓冲区锁定: `-lock` 70

创建脱机文件的归档副本: `-offline_copy` 70

指定回收 71

联合归档: `-join path` 71

控制取消归档 72

控制归档文件的写入方式: `-tapenonstop` 73

保留卷: `-reserve` 74

设置归档优先级: `-priority` 77

预定归档: `-startage`、`-startcount` 和 `-startsize` 78

VSN 关联指令 79

VSN 池指令 81

关于磁盘归档 83

配置原则 84

磁盘归档指令 85

▼ 启用磁盘归档 86

磁盘归档示例	88
示例 1	88
示例 2	89
示例 3	90
设计归档操作	91
预备队列	91
归档程序示例	92
示例 1: 无 archiver.cmd 文件	92
示例 2: 数据文件和元数据分开归档	94
示例 3: 将用户文件和数据文件归档至不同的介质	97
示例 4: 将用户文件和数据文件归档至光盘介质	101

4. 释放 105

释放程序进程概述	106
释放程序概念	107
关于部分释放和部分登台	108
系统管理员选项概述	109
用户选项概述	110
关于 releaser.cmd 文件	110
指定与时限和大小相关的释放优先级指令	111
文件时限	111
文件大小	113
fs 指令: 指定用于单个文件系统的指令	113
no_release 和 display_all_candidates 指令: 指定调试	114
min_residence_age 指令: 指定最短驻留时间	114
logfile 指令: 指定日志文件	115
rearch_no_release 指令: 阻止释放重新归档的文件	116
list_size 指令: 调整释放程序备选文件列表的大小	116
指定归档集中所有文件的释放属性	117
规划释放程序的操作	117
手动运行释放程序	119

5. 登台 121

- 关于 `stager.cmd` 文件 121
 - `drives` 指令：指定驱动器数量 122
 - 设置登台缓冲区大小 123
 - 指定日志文件 124
 - 指定登台请求的数量 126
 - `stager.cmd` 文件示例 127
 - 指定归档集中所有文件的登台属性 127
- 排列预备请求的优先顺序 128
 - 全局 VSN 和时限指令 129
 - 全局界限指令或文件系统专用的界限指令 130
- 计算预备请求的总优先级 131
- 设置预备请求的优先级方案 132
 - 示例 1：强制执行登台请求 133
 - 示例 2：强制执行归档请求 133
 - 示例 3：根据介质确定请求的优先级 133
 - 示例 4：确定复杂请求的优先级 134

6. 回收 135

- 回收过程概述 135
- 回收磁带库 137
 - 使用回收指令 137
 - 指定日志文件：`logfile` 指令 138
 - 阻止回收：`no_recycle` 指令 138
 - 指定回收整个自动化库：库指令 139
 - 配置回收程序 140
 - 创建 `recycler.cmd` 文件 140
 - ▼ 创建 `recycler.cmd` 文件 141

- recycler.cmd 文件示例 141
 - ▼ 运行回收程序 143
- 为回收程序创建 crontab 文件 144
 - ▼ 删除 -recycle_ignore 和 ignore 参数 144
- 创建 recycler.sh 文件 145
- 回收磁盘归档卷 145
 - 编辑 archiver.cmd 文件 146
 - ▼ 运行回收程序 147
 - 为回收程序创建 crontab 文件 147
 - ▼ 删除 -recycle_ignore 和 ignore 参数 147
- 回收以实现归档副本保留 148
- 7. 使用 Sun SAM-Remote 软件 149**
 - Sun SAM-Remote 软件概述 149
 - 特性 150
 - 要求 151
 - 限制 151
 - 客户机与服务器之间的交互作用 152
 - Sun SAM-Remote 服务器概述 153
 - Sun SAM-Remote 客户机概述 153
 - Sun SAM-Remote 服务器和 Sun SAM-Remote 客户机之间的交互作用 153
 - 库目录 154
 - 归档 154
 - 配置 Sun SAM-Remote 软件 155
 - 配置示例 155
 - 配置软件 156
 - ▼ 登录至潜在的服务器及客户机主机 156
 - ▼ 验证客户机和服务器配置 157
 - ▼ 编辑 mcf 文件 158

- ▼ 定义 Sun SAM-Remote 客户机 160
- ▼ 在服务器的 mcf 文件中定义 Sun SAM-Remote 服务器 161
- ▼ 创建 Sun SAM-Remote 服务器配置文件 162
- ▼ 启用归档 164
- 使用 Sun SAM-Remote 软件回收 168
 - 在 Sun SAM-Remote 环境中进行回收 - 方法 1 169
 - 服务器 sky 的配置文件 169
 - 客户机 zeke 的配置文件 170
 - ▼ 配置回收过程 - 方法 1 171
 - ▼ 回收 no-data VSN 188
 - ▼ 回收 partially full VSN 190
 - 在 Sun SAM-Remote 环境中进行回收 - 方法 2 192
 - ▼ 配置回收过程 - 方法 2 192
- 8. 高级主题 195**
 - 在 Sun Cluster 环境中配置 SAM-QFS 文件系统 (HA-SAM) 195
 - 开始之前 196
 - 配置说明 197
 - ▼ 验证 SUNwhasam 资源类型注册文件 197
 - ▼ 确认目录和登台程序符号链接 198
 - ▼ 在 Sun Cluster 环境中配置 Sun StorageTek SAM 199
 - 使用 samd hastop 命令 200
 - 将 Sun StorageTek 库用于 HA-SAM 200
 - 使用设备日志 201
 - 何时使用设备日志 202
 - 启用设备日志 202
 - ▼ 使用 samset(1M) 命令启用设备日志 202
 - ▼ 通过编辑 defaults.conf 文件启用设备日志 203

- 使用可移除介质文件 203
 - ▼ 创建可移除介质或卷溢出文件 204
- 使用分段文件 205
 - 归档 206
 - 故障恢复 206
- 使用系统错误工具报告 206
 - ▼ 启用 SEF 报告 207
 - ▼ 生成 SEF 报告输出 208
 - 管理 SEF 日志文件 211
 - SEF sysevent 功能 211
 - ▼ 创建 SEF sysevent 处理器 211

A. 具有供应商特定操作过程的自动化库的基本操作 213

- ADIC/Grau 自动化库 214
 - ▼ 导入卡盒 214
 - ▼ 导出卡盒 215
- Fujitsu LMF 自动化库 215
 - ▼ 导入卡盒 216
 - ▼ 导出卡盒 216
- IBM 3584 UltraScalable 磁带库 217
 - 导入卡盒 217
 - 清洁驱动器 217
 - 分区 217
 - ▼ 取出卡盒 218
- IBM 3494 库 218
 - ▼ 导入卡盒 218
 - ▼ 导出卡盒 219

Sony 8400 PetaSite 直接连接自动化库 219

▼ 导入磁带 219

导出磁带 220

▼ 在邮箱插槽未用作存储插槽时导出磁带 220

▼ 在邮箱插槽用作存储插槽时导出磁带 221

▼ 将卡盒移至另一个插槽 222

Sony 网络连接自动化库 222

▼ 导入卡盒 223

▼ 导出卡盒 223

StorageTek ACSLS 连接自动化库 224

▼ 导入磁带 224

▼ 使用邮箱导出磁带 225

词汇表 227

索引 239

表

表 1-1	自动化库守护进程	3
表 1-2	File System Manager 权限级别	6
表 2-1	samcmd(1M) load 的参数	12
表 2-2	tplabel(1M) 的参数	14
表 2-3	odlabel(1M) 的参数	14
表 2-4	auditslot(1M) 的参数	15
表 2-5	chmed(1M) 的参数	18
表 2-6	chmed(1M) 的参数	20
表 2-7	auditslot(1M) 的参数	20
表 2-8	chmed(1M) 的参数	22
表 2-9	samexport(1M) 的参数	25
表 3-1	归档程序日志文件的字段	40
表 3-2	archiver.cmd 文件指令单位	43
表 3-3	archmax 指令的参数	47
表 3-4	bufsize 指令的参数	48
表 3-5	drives 指令的参数	49
表 3-6	examine 指令的 <i>method</i> 参数的值	50
表 3-7	ovflmin 指令的参数	52
表 3-8	归档集分配指令的参数	56
表 3-9	-access <i>age</i> 的后缀	57

表 3-10	-minsize 和 -maxsize <i>size</i> 的后缀	58
表 3-11	-release 指令的属性	61
表 3-12	-stage 指令的属性	62
表 3-13	归档集副本参数的参数	66
表 3-14	-drivemax、-drivemin 和 -drives 指令的参数	68
表 3-15	归档集分割示例	69
表 3-16	-offline_copy 指令的 <i>method</i> 参数的值	71
表 3-17	归档集格式示例	74
表 3-18	属主集格式示例	75
表 3-19	文件系统格式示例	75
表 3-20	归档优先级	77
表 3-21	-startage、-startcount 和 -startsize 指令的格式	78
表 3-22	VSN 关联指令的参数	79
表 3-23	VSN 池指令的参数	82
表 4-1	影响部分释放的挂载选项	109
表 4-2	用户释放选项	110
表 4-3	用于释放的归档集分配值	117
表 5-1	drives 指令的参数	122
表 5-2	bufsize 指令的参数	123
表 5-3	event 参数的值	124
表 5-4	登台程序日志文件字段	125
表 5-5	archiver.cmd 文件可包含的登台指令	127
表 5-6	界限优先级指令	130
表 5-7	请求优先级计算示例	133
表 6-1	回收方法与介质类型	137
表 6-2	no_recycle 指令的参数	138
表 6-3	库指令中 <i>parameter</i> 的值	139
表 6-4	归档集回收指令	146
表 7-1	samu(1M) R 显示屏幕标志	167
表 8-1	request(1) 命令的参数	204

表 A-1	import(1M) 命令的参数	214
表 A-2	samexport(1M) 命令的参数	215
表 A-3	import(1M) 命令的参数	216
表 A-4	samexport(1M) 命令的参数	216
表 A-5	move(1M) 命令的参数	220
表 A-6	move(1M) 命令的参数	222
表 A-7	import(1M) 命令的参数	223
表 A-8	samexport(1M) 命令的参数	223
表 A-9	import(1M) 命令的参数	224
表 A-10	samexport(1M) 命令的参数	225

前言

本手册（即《Sun StorageTek™ Storage Archive Manager 归档配置和管理指南》）介绍了 Sun StorageTek Storage Archive Manager (SAM) Version 4, Update 6 (4U6) 发行版中所支持的存储和归档管理软件。Sun StorageTek SAM 软件可以自动将文件从联机磁盘复制到归档介质中。归档介质可由联机磁盘或可移动介质卡盒组成。

以下 Sun Solaris™ 操作系统 (Solaris™ Operating System, Solaris OS) 平台支持 Sun StorageTek SAM 4U6 发行版：

- Solaris 9, update 3, 04/03 或更高版本
- Solaris 10

本手册的读者对象是负责配置和维护 Sun StorageTek SAM 软件的系统管理员。并假定系统管理员非常熟悉 Solaris OS 过程，包括创建帐户、执行系统备份和其他基本的 Solaris 系统管理任务。

注 – 您可以购买 Sun StorageTek QFS 和 Sun StorageTek SAM 软件的许可证，以便将 Sun StorageTek QFS 文件系统和 Sun StorageTek SAM 软件中的存储及归档管理器配合运行。这种将 Sun StorageTek QFS 和 Sun StorageTek SAM 软件配合使用的系统称之为 SAM-QFS 系统。

在本手册中，若非必须单独指出，将不会明确列出 SAM-QFS 配置。在本手册中谈到存储及归档管理时，您可以假定对 Sun StorageTek SAM 的参考同样适用于各种 SAM-QFS 配置。同样，当谈到文件系统设计和性能时，您可以假定对 Sun StorageTek QFS 的参考同样适用于各种 SAM-QFS 配置。

本书的结构

本书包括以下章节：

- 第 1 章是概述。
- 第 2 章介绍基本操作。本章所述的信息适用于大多数自动化库和手动载入的设备。
- 第 3 章介绍归档过程。
- 第 4 章介绍释放过程。
- 第 5 章介绍登台过程。
- 第 6 章介绍回收过程。
- 第 7 章介绍如何使用 Sun SAM-Remote 软件。
- 第 8 章介绍有关 Sun StorageTek SAM 操作的高级主题。
- 附录 A 介绍如何依据各类库的专用操作说明来管理相应库中的卡盒。

词汇表定义了本文档及其他 Sun StorageTek QFS 和 Sun StorageTek SAM 文档中使用的术语。

使用 UNIX 命令

本文档不会介绍基本的 UNIX[®] 命令和操作过程，如关闭系统、启动系统和配置设备等。欲获知此类信息，请参阅以下文档：

- 系统附带的软件文档
- Solaris 操作系统的有关文档，其 URL 如下：

<http://docs.sun.com>

Shell 提示符

表 P-1 列出了本手册中使用的 shell 提示符。

表 P-1 Shell 提示符

Shell	提示符
C shell	<i>machine-name%</i>
C shell 超级用户	<i>machine-name#</i>
Bourne shell 和 Korn shell	\$
Bourne shell 和 Korn shell 超级用户	#

印刷约定

表 P-2 列出了本手册采用的印刷约定。

表 P-2 印刷约定

字体	含义	示例
AaBbCc123	命令、文件和目录的名称；计算机屏幕输出	编辑 .login 文件。 使用 <code>ls -a</code> 列出所有文件。 % You have mail.
AaBbCc123	用户键入的内容，与计算机屏幕输出的显示不同	% su Password:
<i>AaBbCc123</i>	保留未译的新词或术语以及要强调的词。要使用实名或值替换的命令行变量。	这些称为 <i>class</i> 选项。 要删除文件，请键入 rm filename 。
新词术语强调	新词或术语以及要强调的词。	您 必须 成为超级用户才能执行此操作。
《书名》	书名	阅读《用户指南》的第 6 章。
[]	在命令语句中，方括号内的参数表示可选参数。	<code>scmadm [-d sec] [-r n[:n][,n]...] [-z]</code>
{ arg arg }	在命令语句中，大括号和竖线表示必须指定其中一个参数。	<code>sndradm -b { phost shost }</code>
\	命令行末尾的反斜杠 (\) 表示此命令未完，从下一行继续。	<code>atm90 /dev/md/rdsk/d5 \ /dev/md/rdsk/d1</code>

相关文档

本手册是介绍 Sun StorageTek QFS 和 Sun StorageTek SAM 软件产品操作的系列文档中的一本。表 P-3 列出了这些产品的完整的 4U6 版文档集。

表 P-3 相关文档

书名	文件号码
《Sun StorageTek QFS 文件系统配置和管理指南》	820-1718-10
《Sun StorageTek QFS 安装和升级指南》	820-1723-10
《Sun StorageTek Storage Archive Manager 文件系统配置和管理指南》	820-1743-10
《Sun StorageTek Storage Archive Manager 安装和升级指南》	820-1733-10
《Sun StorageTek Storage Archive Manager 故障排除指南》	820-1738-10
《Sun StorageTek QFS 和 Sun StorageTek Storage Archive Manager 4.6 发行说明》	820-1748-10
《Sun StorageTek QFS Linux 客户机指南》	820-1753-10

联机访问 Sun 文档

Sun StorageTek QFS 和 Sun StorageTek SAM 软件分发包括一些 PDF 文件，您可以在 Sun 的 Network Storage 文档 Web 站点或 docs.sun.com 查看这些文件。

从 docs.sun.com 访问文档

此 Web 站点包含 Solaris 和其他多个 Sun 软件产品的文档。

1. 转至以下 URL:

<http://docs.sun.com>

屏幕上会出现 docs.sun.com 页面。

2. 通过在搜索框中搜索相应产品来查找相关的产品文档。

从 Sun 的 Network Storage 文档 Web 站点访问文档

此 Web 站点包含 Network Storage 产品的文档。

1. 转至以下 URL:

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

屏幕上会出现 "Storage Software" 页面。

2. 单击 Sun StorageTek SAM 软件的链接。

第三方 Web 站点

Sun 对本文档中提到的第三方 Web 站点的可用性不承担任何责任。对于此类站点或资源中的（或通过它们获得的）任何内容、广告、产品或其他资料，Sun 并不表示认可，也不承担任何责任。对于因使用或依靠此类站点或资源中的（或通过它们获得的）任何内容、产品或服务而造成的或连带产生的实际或名义损坏或损失，Sun 概不负责，也不承担任何责任。

联系 Sun 技术支持

如果您遇到通过本文档无法解决的技术问题，请访问以下网址：

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

使用许可

有关获取 Sun StorageTek SAM 软件许可证的信息，请与 Sun 销售代表或授权的服务供应商 (Authorized Service Provider, ASP) 联系。

安装帮助

要获得安装和配置服务，请拨打 1-800-USA4SUN 联系 Sun 企业服务部门，或联系当地的企业服务销售代表。

Sun 欢迎您提出意见

Sun 致力于提高其文档的质量，并十分乐意收到您的意见和建议。您可以通过以下网址提交您的意见和建议：

<http://docs.sun.com/app/docs/form/comments>

请在您的反馈信息中包含文档的书名和文件号码：《Sun StorageTek SAM 归档配置和管理指南》，文件号码 820-1728-10。

第 1 章

概述

Sun StorageTek SAM 环境提供了具有存储、归档管理以及恢复功能的可配置文件系统。Sun StorageTek SAM 软件通过将文件从联机磁盘高速缓存复制到归档介质对文件进行归档。归档介质可以是另一文件系统中的磁盘分片，也可以是自动或手动载入的存储设备中的可移除磁带或磁光盘卡盒。Sun StorageTek SAM 软件可自动将联机磁盘空间大小维持在站点指定的使用阈值上。它可以释放与已归档的文件数据相关联的磁盘空间，并在需要时将文件恢复到联机磁盘。

本章从技术层面对 Sun StorageTek SAM 组件进行概述。本章包含以下各节：

- [第 1 页的“性能”](#)
- [第 3 页的“存储设备”](#)
- [第 4 页的“Sun SAM-Remote 软件”](#)
- [第 4 页的“File System Manager”](#)

性能

Sun StorageTek SAM 存储和归档管理软件包含一个位于服务器磁盘高速缓存中的高性能 UNIX 文件系统。有关文件系统自身的更多信息，请参见《Sun StorageTek Storage Archive Manager 文件系统配置和管理指南》。

Sun StorageTek SAM 环境中还包含以下其他的组件：

- **归档程序**自动将联机磁盘高速缓存文件复制到归档介质中。归档介质可由联机磁盘文件或可移除介质卡盒组成。默认情况下，归档程序会自动为 Sun StorageTek SAM 文件系统中的所有文件创建一个归档副本，并将此归档副本写入归档介质。您可以配置归档程序，使其在不同归档介质上最多可创建 4 份归档副本。如果文件被分成几段，则每段都会被视为一个文件，且单独进行归档。当基于磁盘的文件符合站点定义的一套选择标准后，归档进程将会启动。

有关归档程序的更多信息，请参见[第 29 页的“归档”](#)。有关分段文件的更多信息，请参见[第 205 页的“使用分段文件”](#)。

- **释放程序**通过释放那些符合条件的已归档文件占用的磁盘块，自动将文件系统的联机磁盘高速缓存大小维持在站点指定的使用百分比阈值上。

释放是指释放已归档文件的数据所占用为主（磁盘）存储空间的过程。它使用两个阈值 - 上限和下限（均用磁盘总空间的百分比表示）来管理联机磁盘高速缓存的可用空间。当联机磁盘占用空间超出上限时，系统会自动开始释放那些符合条件的已归档文件占用的磁盘空间。当联机磁盘占用空间到达下限时，系统会自动停止释放由已归档文件的数据占用的磁盘空间。选择释放文件的依据是文件的大小和文件在联机磁盘中的时限。文件的第一部分可保留在磁盘上，以加快访问速度并屏蔽登台过程的延迟。如果某个文件被分段归档，则可以单独释放该文件的各个部分。有关释放程序的更多信息，请参见第 105 页的“释放”。

- **登台程序**将文件数据恢复到磁盘高速缓存中。当用户或进程请求那些已从磁盘高速缓存中释放的文件数据时，登台程序会自动将文件数据复制回联机磁盘高速缓存中。

如果某个文件的数据块已被释放，则在系统访问该文件时，登台程序会自动将该文件或文件段的数据重新登台到联机磁盘高速缓存中。一旦开始登台操作，读取操作也将随即开始，从而使应用程序可以立即使用文件，而不必等到整个文件完全登台到磁盘高速缓存中。

Sun StorageTek SAM 软件可自动处理登台请求错误。如果返回登台错误，则系统会尝试查找文件的下一个可用归档副本。系统可自动处理的登台错误包括介质错误、介质不可用、自动化库不可用以及其他错误。有关登台的更多信息，请参见第 121 页的“登台”。

- **回收程序**清除包含已过期归档副本的归档卷，以使这些卷可重新使用。

一旦用户修改了文件，归档介质上与这些文件的旧版本相关联的归档副本将被视为过期。这些副本不再有用，因此可以从系统中清除。回收程序可以识别那些其中绝大部分是过期归档副本的归档卷，并将这些卷中的未过期副本移至其他卷中进行保存。

如果可移除介质卷中仅包含过期副本，则可以执行以下某一操作：

- 重新标记该卷，以便可立即重新使用。
- 将该卷的内容导出至离站存储设备中，作为文件变更的历史记录。可使用标准的 UNIX 实用程序从过期的归档副本中恢复文件的以前版本。

回收进程对最终用户是透明的。有关回收的更多信息，请参见第 135 页的“回收”。

存储设备

Sun StorageTek SAM 环境可支持的磁带存储设备和磁光设备种类非常多。Sun StorageTek SAM 所支持的自动化库可根据其与环境连接的方式，分为以下几组：

- **直接连接。**直接连接的库使用小型计算机系统接口 (Small Computer System Interface, SCSI) 直接连接到主机系统。这种连接可以是直接连接，也可以是光纤通道连接。例如，Sun StorageTek 库使用直接连接方式。Sun StorageTek SAM 系统直接使用自动化库的 SCSI 标准控制这些库。
- **网络连接。**Sun StorageTek SAM 软件可配置为库的主机系统的客户机。网络连接的库包括 StorageTek 库、ADIC/Grau 库、IBM 库和 Sony 库等。这些库使用供应商提供的软件包。此时，Sun StorageTek SAM 软件使用自动化库的专用守护进程，与供应商软件进行交互。

表 1-1 列出了不同自动化库专用的守护进程。

表 1-1 自动化库守护进程

守护进程	说明
sam-robotd	监视传输器控制守护进程的执行情况。sam-robotd 守护进程由 sam-amlD 守护进程自动启动。
sam-genericd	控制直接连接的库和介质更换器。还通过 DAS 接口控制 ADIC 库。
sam-stkd	通过 ACSAPI 接口控制 StorageTek 介质更换器。
sam-ibm3494d	通过 lmcPd 接口控制 IBM 3494 磁带库。
sam-sonyd	通过 DZC-8000S 接口控制 Sony 网络连接自动化库。

有关受支持的存储设备列表，请与 Sun Microsystems 销售代表或授权的服务供应商 (Authorized Service Provider, ASP) 联系。

Sun StorageTek SAM 环境中所管理的各设备之间的关系是在主配置文件 `/etc/opt/SUNWsamfs/mcf` 中定义的。mcf 文件指定了 Sun StorageTek SAM 环境中所包括的可移除介质设备、自动化库和文件系统。在 mcf 文件中，每一个设备均指定有唯一的设备标识符。mcf 文件中的条目还可定义手动挂载的归档设备和自动化库目录文件。

系统将尽可能使用标准的 Solaris 磁盘和磁带设备驱动程序。对于 Solaris OS 不可直接支持的设备（例如某些库和光盘设备），Sun StorageTek SAM 软件包中提供了此类设备的专用驱动程序。

Sun SAM-Remote 软件

Sun SAM-Remote 软件是客户机/服务器模型的一种实现，可以使您在 Sun StorageTek SAM 主机系统之间共享库和其他可移除介质设备。您可使用 Sun SAM-Remote 软件配置多个存储客户机，使它们能够从集中式磁带库或磁光盘库中归档和登台文件。例如，如果您的主机系统分布在一个跨越很大地域的网络中，则在一个城市中创建的文件可以归档至离此城市数英里远的库中的卡盒。

有关更多信息，请参见第 149 页的“使用 Sun SAM-Remote 软件”。

使用 Sun Cluster 实现高可用性配置 (HA-SAM)

可以使用 Sun™ Cluster 软件配置 Sun StorageTek SAM 以实现高可用性。Sun Cluster 软件通过启用应用程序故障转移来提供高可用性。系统将定期对主节点进行监视，并且群集软件会自动将 Sun StorageTek SAM 功能从出现故障的主节点重定位到指定的辅助节点。通过在主节点出现故障时允许群集中的其他节点自动承载归档工作负载，Sun Cluster 软件可以显著缩短停机时间，并提高效率。

高可用性 SAM (High-availability SAM, HA-SAM) 依赖于 Sun StorageTek QFS Sun Cluster 代理，因此，必须使用由 Sun StorageTek QFS Sun Cluster 代理挂载并管理的共享 Sun StorageTek QFS 文件系统来安装此配置。

有关更多信息，请参见第 195 页的“在 Sun Cluster 环境中配置 SAM-QFS 文件系统 (HA-SAM)”。

File System Manager

File System Manager 软件是基于浏览器的图形用户界面，您可使用它从一个中心位置配置和控制一个或多个 Sun StorageTek QFS 或 Sun SAM-QFS 服务器。您可以使用所在网络中任何一台主机上的 Web 浏览器访问这个中心位置。

该软件的目标是为执行与 Sun StorageTek QFS 或 Sun SAM-QFS 服务器相关的最常用的任务提供一个易于使用的界面。要对服务器进行进一步的配置或管理，请使用该服务器的命令行界面、脚本、配置文件等。有关更多信息以及安装 File System Manager 的指导，请参见《Sun StorageTek Storage Archive Manager 安装和升级指南》。

▼ 调用 File System Manager

1. 登录到管理站 Web 服务器。
2. 在 Web 浏览器中，访问以下 URL:

```
https://hostname:6789
```

对于 *hostname*，请键入安装 File System Manager 的主机的名称。如果除主机名外还需要指定域名，请按以下格式指定 *hostname: hostname.domainname*。

请注意，此 URL 的开始部分为 `https`，而不是 `http`。

此时将会显示 Sun Java™ Web Console 登录屏幕。

3. 在 "User Name" 提示符下，输入 `root` 或其他有效的用户名。

注 – 如果已从早期版本升级了 File System Manager 软件，则 `samadmin` 用户帐户也可用。您可以在 "User Name" 字段中键入 `samadmin`，然后键入 `samadmin` 密码来获得对 File System Manager 操作的完全访问权限。

4. 在 "Password" 提示符下，输入密码。
5. 单击 "Log In"。
6. 单击 "File System Manager"。

完成以上步骤之后，您便登录到了 File System Manager。

设置会话超时

Java Web Console 框架的默认会话超时时间为 15 分钟。File System Manager 安装程序会将会话超时时间改为 60 分钟。您可以将会话超时值改为不同的值，但是为了保证安全，建议不要将其设为大于 60 分钟的值。

要更改会话超时值，请在管理站上输入以下命令：

```
/opt/SUNWfsmgr/bin/fsmgr session timeout-in-minutes
```

例如，要将超时值改为 45 分钟，请键入：

```
/opt/SUNWfsmgr/bin/fsmgr session 45
```

创建附加的管理员和用户帐户

在完成 File System Manager 的初始配置后，您可以随时创建附加的管理员和 `guest` 帐户。来宾帐户是管理站的本地帐户。

如果删除 File System Manager 软件，删除脚本并不删除您手动创建的任何附加帐户。

使用以下过程创建附加帐户。

▼ 创建附加帐户

1. 在浏览器界面外部，以 `root` 身份登录管理站服务器。
2. 使用 `useradd` 和 `passwd` 命令添加每一个用户。

例如，要添加帐户名为 `bobsmith` 的用户，请键入以下命令：

```
# /usr/sbin/useradd bobsmith
```

```
# /usr/bin/passwd bobsmith
```

以这种方式添加的每个用户帐户具有对 File System Manager 的功能的只读查看权限。要添加其他权限，请参见下节第 6 页的“指定权限级别”。

指定权限级别

您可以为用户指定对 File System Manager 功能的完全或部分访问权限。下表列出了可以为 File System Manager 用户指定的五种权限级别。

表 1-2 File System Manager 权限级别

管理权限级别	说明
<code>com.sun.netstorage.fsmgr.config</code>	用户的访问不受限制。
<code>com.sun.netstorage.fsmgr.operator.media</code>	用户可以添加或删除库、添加或删除独立的驱动器、保留卷序列名 (volume serial name, VSN)、导入 VSN、载入和卸载 VSN、导出 VSN，等等。
<code>com.sun.netstorage.fsmgr.operator.sam.control</code>	用户可以启动、停止或闲置归档操作。
<code>com.sun.netstorage.fsmgr.operator.file</code>	用户可以启动或停止登台，并且可以恢复文件系统。
<code>com.sun.netstorage.fsmgr.operator.filesystem</code>	用户可以挂载或卸载文件系统、编辑挂载选项并执行文件系统检查 (<code>fsck</code>)。

要为用户指定完全或部分配置权限，请在 `/etc/user_attr` 文件中添加以下行：

```
account-name:::auths=privilege-level
```

account-name 是用户帐户的名称，*privilege-level* 是要指定给用户的授权级别。

例如，要为用户帐户 bobsmith 指定完全权限（权限级别 com.sun.netstorage.fsmgr.config），请在 /etc/user_attr 文件中添加以下行：

```
bobsmith::::auths=com.sun.netstorage.fsmgr.config
```

如果只是为了登台和恢复文件系统（权限级别 com.sun.netstorage.fsmgr.operator.file）以及导出、导入和指定 VSN（权限级别 com.sun.netstorage.operator.media）而指定 bobsmith 权限，请在 /etc/user_attr 文件中添加以下行：

```
bobsmith::::auths=com.sun.netstorage.fsmgr.operator.file,  
com.sun.netstorage.fsmgr.operator.media
```

创建多用户帐户

您可以创建一个可由多个用户使用的通用 File System Manager 帐户，然后添加角色，该角色具有一些权限，只有这些用户中的某些用户才可以访问它。

1. 使用 `useradd` 和 `passwd` 命令添加帐户。

例如，要为多用户添加名为 `guest` 的用户帐户，请键入以下命令：

```
# /usr/sbin/useradd guest  
# /usr/bin/passwd guest
```

2. 使用 `roleadd` 和 `passwd` 命令添加角色。

要在 `guest` 帐户下创建具有特殊权限的名为 `admin` 的角色，请键入以下命令：

```
# /usr/sbin/roleadd admin  
# /usr/bin/passwd admin
```

3. 在 /etc/user_attr 文件中指定权限级别。

要指定 `admin` 角色权限来恢复和登台文件系统，请在 /etc/user_attr 文件中添加以下行：

```
admin::::auths=com.sun.netstorage.fsmgr.operator.file  
guest::::type=normal;roles=admin
```

在本示例中，当用户以 `guest` 身份登录时，File System Manager 会提示用户选择 `No Role` 或 `Admin`。如果用户知道 `Admin` 角色的密码，可以选择 `Admin`，输入 `Admin` 密码，从而拥有恢复和登台文件系统的权限。所有其他用户必须选择 `No Role`，并拥有只读权限。

因为具有相同权限级别的多个用户可以同时登录到软件，这样一个用户所做的更改会覆盖另一个用户以前所做的更改，这是很危险的。为了避免发生这种问题，制订了一些有关哪些人可以进行更改以及如何通知其他用户的策略。

通过 File System Manager 管理其他服务器

默认情况下，File System Manager 被设置为管理其所在的服务器。它还可用于管理运行 Sun StorageTek QFS 或 Sun StorageTek SAM 软件的其他服务器，但是必须首先将这些服务器配置为允许 File System Manager 进行访问。配置这些服务器之后，您可以通过浏览器界面管理这些服务器上的文件系统、归档进程和介质。

添加服务器：

1. 在浏览器界面外部，使用 `telnet` 命令连接到所要添加的服务器。以 `root` 身份登录。
2. 使用 `fsmadm(1M) add` 命令将管理站（File System Manager 软件所在的系统）添加到可远程管理此服务器的主机的列表。

所有通过此命令添加到列表的主机都可以远程管理该服务器。

例如：

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

要确保已成功添加管理站，请使用 `fsmadm(1M) list` 命令，并检验命令输出中是否列出了此管理站。

3. 以管理员用户身份登录到 File System Manager 浏览器界面。
4. 在 "Servers" 页面，单击 "Add"。
屏幕上将显示 "Add Server" 窗口。
5. 在 "Server Name" 或 "IP Address" 字段中，分别键入服务器的名称或 IP 地址。
6. 单击 "OK"。

第 2 章

使用自动化库和手动载入的驱动器

自动化库是一种自动控制的设备，它可在无操作人员参与的情况下，自动载入和卸载可移除卡盒。自动化库也可称为介质更换器、自动光盘存储器、机械手、资料库或介质库。

本章从多个方面介绍了在 Sun StorageTek SAM 环境中使用自动化库和手动载入的驱动器时的一些相关事项。此外，本章还介绍在被请求的卷不在库中时，用于提示操作员载入该卷的通知工具。

注 – Sun StorageTek SAM 软件可与许多厂商的自动化库进行交互。有关库型号、固件级别以及其他兼容性方面的信息，请与 Sun 的客户支持部门联系。

某些自动化库的某些具体操作可能与本章所介绍的操作有所不同。要确定对于 Sun StorageTek SAM 环境中所使用的自动化库，其供应商是否提供有额外的操作指导，请参见附录 A，第 213 页的“具有供应商特定操作过程的自动化库的基本操作”。

本章包括以下各节：

- 第 10 页的“自动化库操作”
- 第 27 页的“手动载入驱动器操作”

自动化库操作

有一些基本操作对于所有自动化库来说，基本上完全相同。本节将介绍以下内容：

- 第 10 页的“启动可移除介质的操作”
- 第 11 页的“停止可移除介质的操作”
- 第 11 页的“打开自动化库”
- 第 12 页的“关闭自动化库”
- 第 12 页的“将卡盒载入自动化库”
- 第 13 页的“从驱动器中卸载卡盒”
- 第 13 页的“标记卡盒”
- 第 15 页的“审计卷”
- 第 15 页的“审计直接连接的自动化库”
- 第 16 页的“使用清洁卡盒”
- 第 19 页的“清洁磁带机”
- 第 20 页的“清除介质错误”
- 第 21 页的“从驱动器中取出卡住的卡盒”
- 第 22 页的“目录操作和卡盒的导入与导出”
- 第 26 页的“启用载入通知”
- 第 27 页的“使用具备加密功能的驱动器”

▼ 启动可移除介质的操作

通常，当挂载了 Sun StorageTek SAM 文件系统后，可移除介质的操作便已启动。

- 要手动启动可移除介质操作，而不挂载文件系统，请输入 `samd(1M) start` 命令。

```
# samd start
```

如果在输入上述命令时可移除介质操作已在运行，则会生成以下消息：

```
SAM-FS sam-amld daemon already running
```

有关 `samd(1M)` 命令的更多信息，请参见 `samd(1M)` 手册页。

▼ 停止可移除介质的操作

可以停止可移除介质的操作，并使 Sun StorageTek SAM 系统保持挂载状态。例如，在您希望手动操作库中的卡盒时，可能需要执行此操作。重新启动介质操作之后，未完成的登台操作将被重新执行，而归档操作也将随之恢复。

- 使用 `samcmd(1M) idle` 和 `samd(1M) stop` 命令。

```
samcmd idle eq  
samd stop
```

其中的 `eq`，用于输入被停止的驱动器在 `mcf` 文件中定义的设备序号。要将多个驱动器置于空闲状态，请对 `mcf` 文件中配置的每一个驱动器输入 `samcmd idle eq` 命令。

也可使用 `samu(1M)` 操作员实用程序或使用 File System Manager 将驱动器置于空闲状态。

注 - 在运行 `samd(1M) stop` 命令之前，请将 Sun StorageTek SAM 环境中的驱动器置于空闲状态。此命令的目的是使归档程序、登台程序和其他进程结束当前的任务。如果未能成功运行 `samd(1M) stop` 命令，则在恢复归档、登台或其他活动时可能导致意外的结果。

▼ 打开自动化库

当某个库处于 `on` 状态时，则它已经处于 Sun StorageTek SAM 系统的控制之下，且可以继续执行一般性的操作。打开库时，Sun StorageTek SAM 软件会执行以下操作：

- 根据设备的内部状态来查询设备。它可以发现磁带的位置、是否使用了条码等。
 - 更新目录和其他内部结构。
- 使用 `samcmd(1M) on` 命令。

```
samcmd on eq
```

其中的 `eq`，用于指定自动化库在 `mcf` 文件中定义的设备序号。

另外，还可使用 `samu(1M)` 或 File System Manager 执行此任务。

▼ 关闭自动化库

将自动化库置于 `off` 状态可以停止 I/O 操作，并使该库不再受 Sun StorageTek SAM 控制。此时，卡盒将不能自动移动，自动化库中的驱动器仍处于 `on` 状态。若要执行以下任务，可能需要关闭自动化库：

- 仅停止 Sun StorageTek SAM 自动化库的操作。
- 关闭自动化库的电源。
- 使用 `samcmd(1M) off` 命令。

```
samcmd off eq
```

其中的 `eq`，用于指定所访问的自动化库在 `mcf` 文件中定义的设备序号。

另外，还可使用 `samu(1M)` 或 File System Manager 执行此任务。

将卡盒载入自动化库

所谓的载入是指卡盒从存储插槽移动到驱动器，并为进行数据接收做相应准备的过程。当请求某个卷序列名 (volume serial name, VSN) 以进行归档或登台操作时，系统会自动载入卡盒。

▼ 手动载入卡盒

- 使用 `samcmd(1M) load` 命令。
- 即使驱动器处于 `unavail` 状态，也可使用此命令。此命令具有以下两种格式：

```
samcmd load eq:slot[:partition]  
samcmd load media-type.vsn
```

表 2-1 samcmd(1M) load 的参数

参数	含义
<code>eq</code>	驱动器在 <code>mcf</code> 文件中定义的设备序号。
<code>slot</code>	存储插槽的编号，与库目录中标识的编号相同。
<code>media-type</code>	介质类型。有关有效介质类型的列表，请参见 <code>mcf(4)</code> 手册页。
<code>partition</code>	磁光盘的一面。 <code>partition</code> 必须为 1 或 2。此参数不适用于磁带卡盒。
<code>vsn</code>	指定给卷的卷序列名。

另外，还可使用 `samu(1M)` 或 `File System Manager` 执行此任务。

当手动载入卡盒时，它通常会载入库中的下一个可用驱动器。如果不希望某个驱动器这样载入，请使用 `samu(1M)` 实用程序的 `:unavail` 命令，或使用 `File System Manager` 更改该设备的状态。例如，在故障恢复操作或对磁带进行分析期间，就可能执行此操作。

注 – Sun StorageTek SAM 不支持在直接连接库中混用不同的介质。如果对库进行了分区，则每个分区必须只包含一种介质类型。

从驱动器中卸载卡盒

卸载是指将卡盒从驱动器中取出的过程。当不再需要使用卷时，系统会自动卸载卡盒。也可手动卸载驱动器。

▼ 手动卸载卡盒

- 使用 `samcmd(1M) unload` 命令。

即使驱动器处于 `unavail` 状态，也可以使用此命令。命令如下：

```
samcmd unload eq
```

其中的 `eq`，用于指定驱动器在 `mcf` 文件中定义的设备序号。

另外，还可使用 `samu(1M)` 或 `File System Manager` 执行此任务。

标记卡盒

标记卡盒的过程取决于您要标记的是磁带还是光盘。以下两小节介绍了这些过程。



注意 – 标记和重新标记卡盒可使任何软件都无法再访问卡盒中当前存储的数据。应该仅在确定卡盒中存储的数据不再有用时，才重新标记卡盒。

▼ 标记或重新标记磁带

- 要标记新磁带，请使用 `tplabel(1M)` 命令。

```
tplabel -new -vsn vsn eq:slot
```

- 要重新标记现有磁带，请使用 `tplabel(1M)` 命令。

```
tplabel -old vsn -vsn vsn eq:slot
```

表 2-2 `tplabel(1M)` 的参数

参数	含义
<i>vsn</i>	卷序列名。如果是进行重新标记，新 VSN 可以与旧 VSN 相同。
<i>eq</i>	驱动器在 <code>mcf</code> 文件中定义的设备序号。
<i>slot</i>	磁带的存储插槽编号，与库目录中标识的编号相同。此参数不适用于手动载入的驱动器。

运行上述用以标记或重新标记磁带的命令后，系统会载入磁带并确定其位置，然后写入磁带标签。有关 `tplabel(1M)` 命令的更多信息，请参见 `tplabel(1M)` 手册页。

另外，还可使用 File System Manager 执行此任务。

▼ 标记或重新标记光盘

- 要标记新光盘，请使用 `odlabel(1M)` 命令。

```
odlabel -new -vsn vsn eq:slot:partition
```

- 要重新标记现有光盘，请使用 `odlabel(1M)` 命令。

```
odlabel -old vsn -vsn vsn eq:slot:partition
```

表 2-3 `odlabel(1M)` 的参数

参数	含义
<i>vsn</i>	卷序列名。如果是进行重新标记，新 VSN 可以与旧 VSN 相同。
<i>eq</i>	驱动器在 <code>mcf</code> 文件中定义的设备序号。
<i>slot</i>	磁盘的存储插槽编号，与库目录中标识的编号相同。此参数不适用于手动载入的驱动器。
<i>partition</i>	磁光盘的一面。 <code>partition</code> 值必须为 1 或 2。此参数不适用于磁带卡盒。

运行上述用以标记或重新标记光盘的命令后，系统会载入光盘并确定其位置，然后写入光盘标签。有关 `odlabel(1M)` 命令的更多信息，请参见 `odlabel(1M)` 手册页。

另外，还可使用 `File System Manager` 执行此任务。

▼ 审计卷

有时，需要在库目录中更新所报告的磁带或光盘上的剩余空间。`auditslot(1M)` 命令用于载入包含卷的卡盒、读取标签并更新存储插槽的库目录条目。

- 使用 `auditslot(1M)` 命令审计卷。

```
auditslot [-e] eq:slot[:partition]
```

表 2-4 `auditslot(1M)` 的参数

参数	含义
<code>-e</code>	如果指定了 <code>-e</code> 选项，且介质为磁带，则将会更新剩余的空间。否则，将不会进行更新。
<code>eq</code>	驱动器在 <code>mcf</code> 文件中定义的设备序号。
<code>slot</code>	存储插槽的编号，与库目录中标识的编号相同。此参数不适用于手动载入的驱动器。
<code>partition</code>	磁光盘的一面。 <code>partition</code> 值必须为 1 或 2。此参数不适用于磁带卡盒。

有关 `auditslot(1M)` 命令的更多信息，请参见 `auditslot(1M)` 手册页。

也可以使用 `samu(1M)` 实用程序的 `:audit` 命令或使用 `File System Manager` 执行此任务。

▼ 审计直接连接的自动化库

注 – 此任务不能在通过网络连接的自动化库上执行。

执行全面审计时，会将每一个卡盒载入驱动器、读取标签并更新库目录。在以下情况下应审计库：

- 移动了自动化库中的卡盒，但没有使用 `Sun StorageTek SAM` 命令
- 如果怀疑库目录的状态有问题，且希望对其进行更新（例如在意外断电后）
- 如果在没有配备邮箱的自动化库中添加、取出或移动了卡盒

- 可使用 `samcmd(1M) audit` 命令对直接连接自动化库执行全面审计。

```
samcmd audit eq
```

其中的 `eq`，用于指定自动化库在 `mcf` 文件中定义的设备序号。

也可以使用 `samu(1M)` 实用程序的 `:audit` 命令或使用 File System Manager 执行此任务。

使用清洁卡盒

如果硬件支持清洁磁带的使用，则 Sun StorageTek SAM 环境也支持使用清洁磁带。如果某个磁带机要求清洁，系统会自动载入清洁磁带。

如果您的系统使用编有条码的标签，则在条码标签中，清洁磁带的 VSN 必须为 CLEAN 或以字母 CLN 开头。另外，可以使用 `chmed(1M)` 命令将 VSN 标记为清洁磁带并设置清洁循环计数。系统允许安装多个清洁磁带。

清洁方法因生产商不同而不同。请参见第 213 页的“具有供应商特定操作过程的自动化库的基本操作”以确定是否需要为您的设备执行特殊过程。

▼ 使用具有条码的清洁卡盒

如果清洁卡盒编有条码，则可以使用 `import(1M)` 命令导入清洁卡盒。

1. 确保清洁卡盒的条码为 CLEAN，或以字母 CLN 开头。
2. 可使用 `import(1M)` 命令将清洁卡盒导入自动化库。

```
import eq
```

其中的 `eq`，用于指定自动化库在 `mcf` 文件中定义的设备序号。

例如，以下命令是将一个在 `mcf` 文件中编号为 50 的清洁磁带导入自动化库：

```
# import 50
```

另外，还可使用 `samu(1M)` 或 File System Manager 执行此任务。

Sun StorageTek SAM 系统会将卡盒从邮箱移至存储插槽，并更新库目录。另外，发出此命令后，系统将设置清洁介质标志，并根据介质的类型将访问计数设置为适当的清洁循环次数。每使用卡盒清洁一次驱动器，访问计数便减少一次。

▼ 使用没有条码的清洁卡盒

如果清洁卡盒没有条码，则必须首先将其导入。此时，它不会被自动标记为清洁卡盒。

1. 使用 `import(1M)` 命令将卡盒导入自动化库。

```
import eq
```

其中的 `eq`，用于指定自动化库在 `mcf` 文件中定义的设备序号。

2. 使用 `chmed(1M)` 命令将设备类型更改为清洁卡盒。

```
chmed +C eq:slot
```

其中的 `eq`，用于指定自动化库的设备序号。其中的 `slot`，用于指定清洁卡盒被载入的插槽。

在以下示例中，自动化库的设备序号为 50，且清洁卡盒位于存储插槽 77 中：

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

3. 再次使用 `chmed(1M)` 命令，设置清洁循环次数。

```
chmed -count count-number eq:slot
```

其中的 `eq`，用于指定自动化库的设备序号。其中的 `slot`，用于指定清洁卡盒被载入的插槽。

以下命令示例为上一步骤中的清洁卡盒设置了清洁循环次数：

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

有关 `chmed(1M)` 命令的更多信息，请参见 `chmed(1M)` 手册页。

▼ 重置清洁循环次数

清洁磁带仅在一定的清洁循环次数内有效。可以使用 `samu(1M)` 实用程序的 `:v` 显示屏幕或使用 `File System Manager` 来查看剩余的循环次数。

Sun StorageTek SAM 系统可以跟踪每个清洁磁带的清洁循环次数，并在剩余循环次数等于零时弹出磁带。例如，DLT 清洁磁带的循环次数为 20，而 Exabyte 清洁磁带的循环次数则为 10。每次导入清洁磁带时，系统均会自动将清洁循环次数重置为该类型磁带的最高循环次数。

如果系统支持自动清洁功能，但自动化库中所有清洁磁带的剩余循环次数均为零，则系统会将驱动器设置为关闭状态，并在 Sun StorageTek SAM 日志中记录一条消息。

- 可使用 `chmed(1M)` 命令将清洁磁带的循环次数重置为零。

```
chmed -count count media-type.vsn
```

表 2-5 chmed(1M) 的参数

参数	含义
<i>count</i>	您为清洁磁带重置的清洁循环次数。
<i>media-type</i>	介质类型。有关有效介质类型的列表，请参见 <code>mcf(4)</code> 手册页。
<i>vsn</i>	指定给卷的卷序列名 (volume serial name, VSN)。

▼ 限制清洁循环次数

某些驱动器错误可能会导致重复载入清洁卡盒，直到执行完所有的清洁循环。

- 可以使用 `chmed(1M)` 命令限制清洁卡盒的清洁循环次数。

```
chmed -count count-number eq:slot
```

其中的 *eq*，用于指定自动化库的设备序号。其中的 *slot*，用于指定清洁卡盒被载入的插槽。

例如，以下命令将设备序号为 50 的自动化库的插槽 77 中的清洁卡盒清洁循环次数设置为 20。

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

▼ 清洁磁带机

如果不能进行自动清洁，而且系统使用条码，则可手动执行以下步骤来请求清洁驱动器：

- 使用 `cleandrive(1M)` 命令。

```
cleandrive eq
```

其中的 `eq`，用于指定自动化库在 `mcf` 文件中定义的设备序号。这是要载入清洁卡盒的驱动器。

磁带机自动清洁

从 Sun StorageTek SAM 4U4 开始，通过软件启动的磁带机清洁功能在默认情况下被设置为 `off`，因此用户必须选择是通过硬件启动还是通过软件启动磁带机清洁策略。

通过硬件启动的清洁策略使用了介质更换器内置的自动清洁功能。

- 要使用此功能，可能需要将清洁卡盒插入特殊插槽。
- 有关指导，请参见生产商提供的文档。

通过软件启动的清洁策略使用了 Sun StorageTek SAM 的自动清洁功能。

- 此功能中新增了 `logsense` 选项，用于防止驱动器使用过期的清洁介质。
- 要启用 Sun StorageTek SAM 的自动清洁功能，应当禁用通过硬件启动清洁操作，并在 `defaults.conf` 文件中包含以下行：

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

要调用传统的 Sun StorageTek SAM 自动清洁功能（该功能仅依赖于表示驱动器清洁状态的检测数据），请在 `defaults.conf` 文件中插入以下行：

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

注 – 在具有两个以上驱动器的库上使用自动清洁功能时，建议每个 Sun StorageTek SAM 目录应至少具有两个清洁卡盒。如果多个驱动器需要清洁，但没有足够的清洁卡盒，则系统会将这些驱动器置于 `DOWN` 状态。

▼ 清除介质错误

当卡盒发生硬件或软件错误时，Sun StorageTek SAM 系统会在 VSN 目录中设置 `media error` 标志。对于任何生成 `media error` 信号的给定卡盒，均可使用 `chmed(1M)` 命令清除其错误，然后尝试使用该卡盒。可使用 `samu(1M)` 实用程序的 `v` 显示屏幕和 File System Manager 显示 `media error` 标志。

注 – 删除 `error` 标志可能会引发问题。如果无法确定导致错误的原因以及是否可安全删除此标志，请首先与 Sun 技术支持部门联系。

1. 使用 `chmed(1M)` 命令清除 `media error` 标志。

```
chmed -E media-type.vsn
```

表 2-6 `chmed(1M)` 的参数

参数	含义
<code>media-type</code>	介质类型。有关有效介质类型的列表，请参见 <code>mcf(4)</code> 手册页。
<code>vsn</code>	指定给卷的卷序列名 (volume serial name, VSN)。

2. 运行 `auditslot(1M)` 命令更新剩余空间信息。

```
auditslot -e eq:slot[:partition]
```

表 2-7 `auditslot(1M)` 的参数

参数	含义
<code>-e</code>	如果指定了 <code>-e</code> 选项，且介质为磁带，则将会更新剩余的空间。否则，将不会进行更新。
<code>eq</code>	自动化库或手动载入的驱动器在 <code>mcf</code> 文件中定义的设备序号。
<code>slot</code>	自动化库中存储插槽的编号，与库目录中标识的编号相同。此参数不适用于手动载入的驱动器。
<code>partition</code>	磁光盘的一面。 <code>partition</code> 必须为 1 或 2。此参数不适用于磁带卡盒。

有关 `auditslot(1M)` 命令的更多信息，请参见 `auditslot(1M)` 手册页。

也可使用 `samu(1M)` 实用程序的 `:audit` 命令或 File System Manager 执行此任务。

▼ 从驱动器中取出卡住的卡盒

如果卡盒卡在驱动器中，请执行以下步骤。

1. 使用 `samcmd(1M) off` 命令关闭自动化库中的驱动器。

```
samcmd off eq
```

其中的 `eq`，用于指定驱动器在 `mcf` 文件中定义的设备序号。

另外，还可使用 `samu(1M)` 或 File System Manager 执行此步骤。

2. 使用 `samcmd(1M) off` 命令关闭自动化库。

```
samcmd off eq
```

其中的 `eq`，用于指定库在 `mcf` 文件中定义的设备序号。

另外，还可使用 `samu(1M)` 或 File System Manager 执行此步骤。

3. 从驱动器中物理取出卡盒。

确保不要损坏卡盒或驱动器。

4. 使用 `samcmd(1M) on` 命令打开自动化库和驱动器。

为驱动器和库分别运行一条此命令。

```
samcmd on eq
```

其中的 `eq`，用于指定库或驱动器在 `mcf` 文件中定义的设备序号。

如果自动化库在打开后执行了审计，则本过程结束。

5. 如果自动化库在打开后不执行审计，请执行以下步骤：

a. 将卡盒放回其存储插槽。

b. 使用 `chmed(1M)` 命令调整库目录，以便为损坏的磁带设置占用标志。

此命令的使用格式如下：

```
chmed +o eq:slot
```

表 2-8 chmed(1M) 的参数

参数	含义
<code>eq</code>	自动化库或驱动器在 <code>mcf</code> 文件中定义的设备序号。
<code>slot</code>	库中存储插槽的编号，与库目录中标识的编号相同。此参数不适用于手动载入的驱动器。

有关 `chmed(1M)` 命令的更多信息，请参见 `chmed(1M)` 手册页。

如果您取出卡盒，并希望在以后将其放回，则必须将卡盒导入自动化库。

目录操作和卡盒的导入与导出

在自动化库中物理地添加（导入）和从中取出（导出）卡盒，使您可以执行多项功能，这其中包括：

- 替换卡盒。
- 将卡盒重新定位为离站存储，以备今后进行故障恢复时使用。您可以使用 `chmed(1M)` 命令的 `-I` 选项来指定诸如卡盒存储位置等其他信息。

导入和导出卡盒时，还可更新库目录。

库目录是一个中央仓库，其中包含了 Sun StorageTek SAM 环境在查找自动化库中的卡盒时所需的所有信息。库目录文件是一个二进制 UNIX 文件系统 (UNIX file system, UFS) 驻留文件，其中包括自动化库中每一个插槽的有关信息。该文件中的信息包括：与插槽中的卡盒相关联的一个或多个 VSN；该卡盒的容量和剩余空间；以及指示卡盒只读、写保护、回收和其他状态信息的标志。

Sun StorageTek SAM 环境处理目录的方式不尽相同，这取决于自动化库连接到服务器的方式，具体如下：

- 如果自动化库采用直接连接方式，则库目录的条目与自动化库中的物理插槽之间是一一对应关系。库目录中的第一个条目是自动化库中的第一个插槽。需要某个卡盒时，系统将首先查询库目录，以确定哪一个插槽包含该 VSN，然后发出命令以将该插槽中的卡盒载入驱动器。

注 – Sun StorageTek SAM 不支持在直接连接库中混用不同的介质。如果对库进行了分区，则每个分区必须只包含一种介质类型。

- 如果自动化库采用网络连接方式，则库目录中的条目与自动化库中的插槽不是直接的对应关系。它是自动化库中已知 VSN 的列表。请求某个卡盒时，系统将向供应商的软件发送请求以将该 VSN 载入驱动器。供应商的软件可以确定包含该 VSN 的存储插槽。

每个自动化库处理卡盒导入和导出操作的方式，都因系统特性和供应商提供的软件不同而存在差异。例如，对于 ACL 4/52 库，您需要先运行 `move` 命令将卡盒移至导入/导出设备中，然后才能从自动化库中导出卡盒。

通过网络连接的自动化库将使用它们自己的实用程序来导入和导出卡盒，因此 `import(1M)` 和 `samexport(1M)` 命令只更新 Sun StorageTek SAM 系统使用的库目录条目。如果您使用的是通过网络连接的自动化库，请参见第 213 页的“具有供应商特定操作过程的自动化库的基本操作”了解有关导入和导出卡盒的信息。

跟踪导出的介质 – 历史记录

Sun StorageTek SAM 历史记录可以跟踪记录从自动化库或手动挂载的设备中导出的卡盒。历史记录类似于一个虚拟库，但它没有已定义的硬件设备。与自动化库一样，它也在 `mcf` 文件中进行配置；具有用于记录与其关联的所有卡盒的目录；可以导入和导出卡盒；而且在 File System Manager 中显示为另一自动化库。

可以在 `mcf` 文件中，使用 `hy` 设备类型来配置历史记录。如果没有在 `mcf` 文件中配置历史记录，它将按以下方式创建：

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

在以上条目中，`n+1` 是 `mcf` 文件中的最后一个设备序号加 1。如果希望目录使用不同的设备序号或路径名，则需要对历史记录进行定义。

历史记录第一次启动时，历史记录库目录将被初始化为具有 32 个条目。请确保文件系统中的目录足以容纳整个目录。您可能希望跟踪已从库中导出的现有 Sun StorageTek SAM 卡盒。在这种情况下，您需要按 `build_cat(1M)` 手册页中的说明，根据现有的卡盒建立历史记录目录。

defaults.conf 文件中的以下两个配置指令将影响历史记录的操作：

- 如果使用了 `exported_media = unavailable` 指令，则从自动化库中导出的任何卡盒均会被标记为不能用于历史记录。请求已标记为不可用的卡盒将产生 EIO 错误。
- 如果使用了 `attended = no` 指令，则它将通知历史记录没有操作员处理载入请求。如果已通知历史记录载入卡盒，但实际并未载入，则会产生 EIO 错误。

有关配置的更多信息，请参见 `historian(7)` 和 `defaults.conf(4)` 手册页。

对自动化库执行导入和导出操作

邮箱是自动化库中的一个区域，用于在自动化库中添加和取出卡盒。`import(1M)` 命令用于将卡盒从邮箱移至存储插槽。`samexport(1M)` 命令用于将卡盒从存储插槽移至邮箱。对于大多数库，如果在 Sun StorageTek SAM 软件启动时，邮箱中含有卡盒，则软件会在启动期间自动导入此卡盒。

导入和导出的具体操作过程因生产商而异。请参见第 213 页的“具有供应商特定操作过程的自动化库的基本操作”以确定是否需要为您的设备执行特殊过程。

以下几节介绍如何导入和导出卡盒：

- 第 24 页的“向使用邮箱的库中导入卡盒”
- 第 25 页的“从使用邮箱的库中导出卡盒”
- 第 25 页的“向不使用邮箱的库中导入卡盒”
- 第 26 页的“从不使用邮箱的库中导出卡盒”

▼ 向使用邮箱的库中导入卡盒

1. 使用生产商建议的操作打开邮箱。

邮箱的旁边通常配有一个按钮。有时，邮箱可能只配有一个插槽（在供应商的文档中称为邮槽）。

2. 将卡盒手动放入邮箱。
3. 关闭邮箱。
4. 使用 `import(1M)` 命令导入卡盒。

```
import eq
```

其中的 `eq`，用于指定库在 `mc.f` 文件中定义的设备序号。

系统将卡盒从邮箱移至存储插槽，并更新此卡盒的库目录。

另外，还可使用 `samu(1M)` 或 File System Manager 执行此任务。

▼ 从使用邮箱的库中导出卡盒

1. 使用 **samexport(1M)** 命令将卡盒从存储插槽移至邮箱。

按以下某一种格式使用此命令：

```
samexport eq:slot  
samexport media-type.vsn
```

表 2-9 samexport(1M) 的参数

参数	含义
<i>eq</i>	自动化库在 <i>mcf</i> 文件中定义的设备序号。
<i>slot</i>	自动化库中存储插槽的编号，与库目录中标识的编号相同。
<i>media-type</i>	卡盒的介质类型。有关有效介质类型的列表，请参见 <i>mcf(4)</i> 手册页。
<i>vsn</i>	指定给卷的卷序列名 (volume serial name, VSN)。

另外，还可使用 **samu(1M)** 或 File System Manager 执行此步骤。

2. 使用生产商建议的操作打开邮箱或邮槽。

邮箱的旁边通常配有一个按钮。

▼ 向不使用邮箱的库中导入卡盒

1. 使用 **samcmd(1M) unload** 命令。

```
samcmd unload eq
```

其中的 *eq*，用于指定库在 *mcf* 文件中定义的设备序号。

2. 等待系统完成其当前任务，将自动化库的状态设置为 `off`，然后将当前活动的目录传输至历史记录。
3. 解除自动化库挡门的锁定，打开挡门。
4. 将卡盒载入可用的插槽。
5. 关闭自动化库的挡门，锁定挡门。

自动化库将重新初始化并扫描库中的卡盒。Sun StorageTek SAM 软件将所导入的卡盒的 VSN 添加到库目录中，从而更新该目录。此时，自动化库的状态会设置为 `on`。

▼ 从不使用邮箱的库中导出卡盒

1. 使用 `samcmd(1M) unload` 命令。

```
samcmd unload eq
```

其中的 `eq`，用于指定所访问的库在 `mcf` 文件中定义的设备序号。

2. 等待系统完成其当前任务，将自动化库的状态设置为 `off`，然后将当前活动的目录传输至历史记录。
3. 解除自动化库挡门的锁定，打开挡门。
4. 从各个插槽中取出所需的卡盒。
5. 关闭自动化库的挡门，锁定挡门。

自动化库将重新初始化并扫描自动化库中的卡盒。系统使用库插槽中当前卡盒的 `VSN` 来更新库目录。已取出的卡盒的 `VSN` 会从库目录中删除，并且只保留在历史记录文件中。此时，自动化库的状态会设置为 `on`。

▼ 启用载入通知

Sun StorageTek SAM 软件会定期请求载入卡盒，以满足归档和登台需求。如果被请求的卡盒位于库中，则系统会自动处理该请求。如果被请求的卡盒不在库中，则需要操作员载入卡盒。如果已启用载入通知，则当需要获取不在库中的卡盒时，`load_notify.sh` 脚本会向有关人员发送电子邮件。

1. 成为超级用户。
2. 使用 `cp(1)` 命令，将载入通知脚本从其安装位置复制到可操作的位置。

例如：

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

3. 使用 `more(1)` 或其他命令检查 `defaults.conf` 文件。

确保此文件中包含以下指令：

- `exported_media=available`
- `attended=yes`

默认情况下，系统已设置了这些指令。如果要启用载入通知功能，请确保未更改这些指令。

4. 修改 `load_notify.sh` 脚本以便将通知发送给操作员。

默认情况下，此脚本将向 `root` 用户发送电子邮件。不过，您可以编辑此脚本以将电子邮件发送给其他人员、拨打寻呼机，或提供其他通知方式。

使用具备加密功能的驱动器

如果要将文件归档到具备加密功能的驱动器，请特别注意以下事项：

- 库中不应混用不具备加密功能和具备加密功能的驱动器。
- 一旦驱动器启用了加密，便无法禁用。
- 不能在磁带中混用加密和非加密文件。
- 不能将加密的驱动器附加到包含非加密数据的磁带。
- 但是，启用加密的驱动器可以读取非加密数据。

由于存在上述限制，因此，如果要在归档系统中使用非加密数据和加密数据以及驱动器，必须仔细在 `archiver.cmd` 文件中规划归档集和磁带分配。

手动载入驱动器操作

本节介绍在您配有手动载入的独立驱动器（而不是自动化库）时所执行的操作。每一个手动载入的驱动器都有自己的单插槽库目录。

▼ 载入卡盒

- 要将卡盒载入手动载入设备，请根据生产商的说明将卡盒放入驱动器。

Sun StorageTek SAM 系统会识别已载入的卡盒，读取标签并更新单插槽目录。无需进行其他操作。

▼ 卸载卡盒

- 使用 `samcmd(1M)` `idle` 命令将驱动器置于空闲状态。

```
samcmd idle eq
```

其中的 `eq`，用于指定驱动器在 `mcf` 文件中定义的设备序号。

此命令可以确保无任何归档或登台进程处于活动状态。完成所有 I/O 活动后，驱动器的状态将从 `idle` 切换为 `off`，然后弹出磁带。

如果卡盒是磁带，则先进行倒带，然后才能取出来。如果是光盘，则会自动弹出。有关取出特定卡盒的说明，请参见生产商提供的指导。

另外，还可使用 `samu(1M)` 或 File System Manager 执行此任务。

▼ 查看库目录

- 使用 `samu(1M)` 实用程序的 `:v` 命令。

```
:v eq
```

其中的 `eq`，用于指定库在 `mcf` 文件中定义的设备序号。

第3章

归档

归档是指将文件从 Sun StorageTek SAM 文件系统复制到可移除介质卡盒上的卷，或复制到另一个文件系统的磁盘分区上的卷的过程。使用 Sun StorageTek SAM 归档功能，可以指定哪些文件需要立即归档，哪些文件从不归档，并且还可以执行其他任务。

本章中，术语**归档介质**是指向其中写入归档卷的各种卡盒或磁盘分片。本章介绍归档程序的操作原理、制订站点归档策略的通用原则，以及如何通过创建 `archiver.cmd` 文件来实现这些归档策略。

本章包括以下各节：

- [第 29 页的“归档过程概述”](#)
- [第 41 页的“关于 archiver.cmd 文件”](#)
- [第 46 页的“使用归档程序指令”](#)
- [第 83 页的“关于磁盘归档”](#)
- [第 91 页的“设计归档操作”](#)
- [第 92 页的“归档程序示例”](#)

归档过程概述

归档程序自动将 Sun StorageTek SAM 文件写入归档介质中。对文件进行归档不需要操作人员的参与。文件归档至归档介质上的卷，每个卷由称为卷序列名 (volume serial name, VSN) 的唯一标识符来标识。归档介质中可以包含一个或多个卷。

归档程序在挂载 Sun StorageTek SAM 文件系统时自动启动。通过在以下文件中插入归档指令，您可以为您的站点自定义归档程序的操作：

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

执行归档时，并不一定需要有 `archiver.cmd` 文件。当没有这个文件时，归档程序使用以下默认设置：

- 将所有文件归档至可用卷。
- 所有文件的归档时限为 4 分钟。归档时限是指从最后一次修改文件到开始归档文件所经历的时间。
- 归档时间间隔为 10 分钟。归档时间间隔是指两次完整的归档过程之间相隔的时间。

以下几节介绍归档集的概念，并说明在归档过程中所执行的操作。

归档程序的守护进程

`sam-archiverd` 守护进程用于预定归档活动。`sam-arfind` 进程用于将要归档的文件指定给归档集。`sam-arcopy` 进程用于将需要归档的文件复制到选定的卷。

Sun StorageTek SAM 活动时，`sam-fsd` 将会启动 `sam-archiverd` 守护进程。`sam-archiver` 守护进程将执行 `archiver(1M)` 命令，以读取 `archiver.cmd` 文件并建立用于控制归档操作的表。它将为每一个已挂载的文件系统启动 `sam-arfind` 进程；如果未挂载文件系统，则会停止相关的 `sam-arfind` 进程。然后，`sam-archiverd` 进程将监视 `sam-arfind` 的运行状态，并处理来自操作员或其他进程的信号。

归档集

归档集表示一组要归档的文件。用户可以在任意多个文件系统中定义归档集。同一个归档集中的文件在大小、拥有权、组和目录位置等方面遵守相同的标准。归档集控制归档副本的目的地、归档副本的保留时间以及数据归档之前软件的等待时间。归档集中的所有文件均被复制到与该归档集关联的卷。文件系统中的文件只能属于一个归档集。

文件在创建或修改之后，归档程序会将其复制到归档介质中。归档过程还会复制执行 Sun StorageTek SAM 文件系统操作所必需的数据，包括目录、符号链接、分段文件的索引和归档介质信息。

归档文件与标准 UNIX `tar(1)` 格式兼容。这可确保与 Solaris OS 和其他 UNIX 系统的数据兼容性。由于采用了 `tar(1)` 格式，这样在 Sun StorageTek SAM 环境中的所有数据都丢失时，就可使用标准的 UNIX 工具和命令来恢复文件。

通常，管理员可为归档集取任意名称，但以下情况除外：

- 有两个保留的归档集名称：`no_archive` 和 `allsets`。
`no_archive` 归档集是系统默认定义的归档集。选入此归档集的文件永远不会被归档。临时目录（例如 `/sam1/tmp`）中的文件可能会包括在 `no_archive` 归档集中。
`allsets` 归档集用于定义适用于所有归档集的参数。
- 每个 Sun StorageTek SAM 文件系统的归档集名称都会被保留，以便控制结构信息。Sun StorageTek SAM 文件系统为每个文件系统都提供了一个默认归档集。对于每个文件系统，归档程序不仅为其归档元数据，而且还归档文件数据。文件系统归档集包括目录和链接信息以及不属于其他归档集的任何文件。默认归档集的名称与其关联的文件系统的名称相同，并且不能更改。例如，如果一个文件系统的名称为 `samfs1`，则其归档集的名称应为 `samfs1`。
- 归档集名最长不得超过 29 个字符。所用的字符仅限于 26 个大小写字母、数字 0 至 9 以及下划线字符 (`_`)。

归档操作

默认情况下，归档程序会为每个归档集创建一个副本，但您最多可以要求创建四个副本。归档集和副本份数意味着占用一系列卷。归档副本在其他卷中提供文件的复件。

文件中的数据须经更改后，该文件才被视为归档或重新归档的对象。如果只是访问文件，则不会对其归档。例如，对某个文件运行 `touch(1)` 或 `mv(1)` 命令，并不会导致系统对该文件进行归档或重新归档。

注 – 运行 `mv(1)` 命令只是改变文件的名称，而并没有更改文件数据。在故障恢复时，如果通过 `tar(1)` 文件进行恢复，则运行此命令可能会产生其他后果。有关故障恢复的更多信息，请参见《Sun StorageTek Storage Archive Manager 故障排除指南》。

归档程序依据文件的归档时限来选择要归档的文件，归档时限指的是自上次修改该文件以来所经过的这段时间。用户可以为每一个归档副本定义归档时限。

用户可以使用 `touch(1)` 命令将其文件的默认时间基准，更改成过去较早的时间或将来较远的时间。但是，这可能会导致意外的归档结果。为了避免发生此类问题，归档程序会调节该时间基准，以使它们始终处于文件的创建时间和当前时间之间。

归档优先级是根据文件属性特征以及与归档集关联的文件属性乘数计算出来的。事实上，其计算公式如下所示：

$$\text{archive-priority} = \text{file-property-value} \times \text{property-multiplier}$$

大多数 *file-property-value* 数值为 1（代表 `true`）或 0（代表 `false`）。例如，如果正在创建第 1 个归档副本，则第 1 个属性副本的值为 1，而第 2 个副本、第 3 个副本和第 4 个副本的值为 0。归档时限和文件大小等其他属性的值可以是 0 或 1 之外的其他值。

property-multiplier 的值由归档集的 *-priority* 参数决定。可以为文件的各个方面（例如时限或大小）设定值以确定归档请求的优先级。有关 *-priority* 参数的更多信息，请参见 `archiver.cmd(4)` 手册页。

archive-priority 和 *property-multiplier* 的值是浮点数值。所有属性乘数的默认值均为 0.0。此时，此归档请求被设置为归档请求中的最高文件优先级。

以下几节介绍归档程序执行的步骤，包括从最初的文件扫描进程到文件复制进程。

步骤 1：识别要归档的文件

每一个已挂载的文件系统均有单独的 `sam-arfind` 进程。`sam-arfind` 进程监视每一个文件系统，以确定哪些文件需要归档。当某文件的更改会影响它自身的归档状态时，文件系统都会向其 `sam-arfind` 进程发出通知。这类更改的示例有：对文件进行修改、重新归档、取消归档以及重命名。`sam-arfind` 进程在接到通知后会检查该文件，以确定是否需要对它执行归档操作。

`sam-arfind` 进程根据文件的属性说明，来确定文件所属的归档集。用于确定文件所属归档集的特征如下：

- 文件名中的目录路径部分，以及采用了正则表达式的完整文件名（可选）
- 文件属主的用户名
- 文件属主的组名
- 最小文件大小
- 最大文件大小

如果文件的一个或多个副本到达或超出归档时限，`sam-arfind` 会将该文件添加到归档集的一个或多个归档请求中。归档请求是属于同一个归档集的文件集合。归档请求位于以下目录中：

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

这个目录中的文件是二进制文件，可以使用 `showqueue(1M)` 命令显示。

对于尚未归档和需要重新归档的文件，则采用单独的归档请求。这样可以分别控制这两种文件的预定。

如果文件的一个或多个副本未达到归档时限，该文件所在的目录以及将到达归档时限的时间会被添加到扫描列表中。一旦到达扫描列表中列出的时间，将对目录进行扫描。并将到达归档时限的文件添加到归档请求中。

如果文件处于脱机状态，则 `sam-arfind` 进程会选择将用作归档副本来源的卷。如果文件副本需要重新归档，则 `sam-arfind` 进程将选择包含这个需要重新归档的归档副本的卷。

如果文件已被分成数段，此进程将仅选择已发生更改的文件段进行归档。分段文件的索引不含用户数据，因此这些分段文件将被视为文件系统归档集的成员而单独进行归档。

确定文件是否需要归档的方法有两种：**连续归档**和**扫描**。连续归档即为，归档程序会对文件系统进行处理，以确定哪些文件需要归档。而扫描方法则是，归档程序定期细查文件系统，并选择需要归档的文件。以下几节将介绍这两种方法。

连续归档

连续归档是默认的归档方法（archiver.cmd 文件参数为 `examine=noscan`）。在连续归档方法中，可以通过 `-startage`、`-startcount` 以及 `-startsize` 参数来指定归档集预定的开始条件。这些条件允许您根据已完成的归档工作优化归档时效。例如：

- 如果需要花费一个小时来创建应该同时归档的文件，则可以将 `-startage` 参数设置为一小时（`-startage 1h`），以确保所有文件在预定归档请求前都已创建完毕。
- 可以将 `-startsize` 指定为 150 GB（`-startsize 150g`），以指示归档程序在需要归档的数据达到 150 GB 时，才进行归档。
- 如果知道要生成 3000 个需要归档的文件，则可以指定 `-startcount 3000`，以确保同时对这些文件进行归档。

满足任一项预定开始条件之后，`sam-arfind` 进程会将每个归档请求发送至归档程序守护进程 `sam-archiverd` 进行预定，以将文件复制到归档介质。

有关归档参数的更多信息，请参见第 46 页的“全局归档指令”。

注 - 如果将 `examine` 设置为 `noscan`，将会自动应用以下默认设置：

- `startage`: 10 分钟
 - `startsize`: 10 GB
 - `startcount`: 10,000 个文件
-

扫描归档

如果不使用连续归档方法，则可以在 `archiver.cmd` 文件中指定 `examine=scan` 来指示 `sam-arfind` 进行扫描，以检查需要归档的文件。需要归档的文件会被放入归档请求中。`sam-arfind` 进程将定期扫描每一个文件系统，以确定哪些文件需要归档。第一个扫描过程是目录扫描，在此扫描期间，`sam-arfind` 按从上到下的顺序递归遍历目录树。该进程将检查每一个文件，如果文件不需要进行归档，则将该文件的状态标志设置为 `archdone` 已设置。接下来，`sam-arfind` 对 `.inodes` 文件进行扫描。此时，只检查没有设置 `archdone` 标志的 `inode`。

有关控制 `archdone` 标志的设置的信息，请参见第 54 页的“[setarchdone 指令：控制 archdone 标志的设置](#)”。

完成文件系统扫描之后，`sam-arfind` 进程会将每个归档请求发送至归档程序守护进程 `sam-archiverd` 进行预定，以将文件复制到归档介质。`sam-arfind` 进程随后进入休眠状态，休眠的时间为 `interval=time` 指令所指定的时间。在时间间隔结束时，`sam-arfind` 进程将重新开始扫描。

步骤 2：组合归档请求

`sam-archiverd` 守护进程收到归档请求后，将对其进行组合。本节将介绍此组合过程。

可能无法一次对归档请求中的所有文件进行归档，这取决于归档介质的容量或归档程序命令文件中指定的控制条件。组合是指从归档请求中选择一次要对哪些文件进行归档的过程。完成归档请求的归档复制操作后，如果仍有需要归档的文件，则会重新组合归档请求。

`sam-archiverd` 守护进程会根据某些默认标准和站点特定的标准，来放置归档请求中的文件。默认操作是根据文件系统扫描期间发现文件的顺序，依次将归档请求中的所有文件归档到同一个归档卷中。您可以通过指定站点的特定标准，来控制按什么顺序对文件进行归档，以及如何将它们分布到不同的卷中。这些标准称为归档集参数，其检验顺序依次为：`-reserve`、`-join`、`-sort`、`-rsort`（逆向排序）和 `-drives`。有关这些参数的更多信息，请参见 `archiver.cmd(4)` 手册页。

如果归档请求属于已指定 `-reserve owner` 的归档集，则 `sam-archiverd` 守护进程将依据文件的目录路径、用户名或组名对归档请求中的文件进行排序。它首先选择属于第一个 `owner` 的文件进行归档。剩余文件将在以后进行归档。

如果归档请求属于已指定 `-join method` 的归档集，则 `sam-archiverd` 守护进程将根据指定的组合方法对文件进行分组。如果还指定了 `-sort` 或 `-rsort method`，则 `sam-archiverd` 守护进程将根据指定的排序方法对每个组中的文件进行排序。对于以后的组合和预定进程，每一个组内的组合文件均被视为单个文件。

如果归档请求属于已指定 `-sort` 或 `-rsort method` 的归档集，则 `sam-archiverd` 守护进程将根据指定的排序方法对文件进行排序。`sam-archiverd` 守护进程通常会根据排序方法、归档时限、大小或目录位置将文件组合在一起，具体视排序方法而定。默认情况下，不会对归档请求进行排序，因此归档程序将根据在文件系统扫描期间发现文件的顺序对文件进行归档。

`sam-archiverd` 守护进程可以确定文件是处于联机状态还是脱机状态。如果归档请求中既包含联机文件，也包含脱机文件，则会首先选择联机文件进行归档。

如果未要求归档请求按某种排序方法进行组合或排序，则脱机文件会按归档副本所在的卷来排列。这可确保同一卷中每一个归档集内的所有文件可以按它们在介质上的存储顺序同时登台。在为脱机文件创建多个归档副本期间，脱机文件不会被释放，直到创建完所有要求的副本。与第一个文件处于同一卷中且要登台的所有文件都将被选作首先进行归档的文件。

注 - 如果要归档的文件的顺序与脱机文件所需的卷顺序不符，则在对脱机文件进行归档期间，`-join`、`-sort` 和 `-rsort` 参数可能会对性能造成负面影响。请仅对所要创建的第一个归档副本使用这些参数。开始创建其他副本时，如果归档介质有足够的空间，则其他副本应采用第一个副本的归档顺序进行归档。

在组合之后，归档请求会进入 `sam-archiverd` 守护进程的预定队列，如下节所述。

步骤 3：预定归档请求

`sam-archiverd` 守护进程中的调度程序将在出现以下情况之一时立即执行：

- 归档请求进入预定队列。
- 已完成某个归档请求的归档。
- 从目录服务器收到介质状态发生变化的消息。
- 收到更改归档程序状态的消息。

预定队列中的归档请求按优先级排列。调度程序每次运行时，均会检查所有归档请求，以确定是否可以将其指定给 `sam-arcopy` 进程，从而将它们的文件复制到归档介质中。

要预定归档请求，必须满足以下条件：

- 必须存在用于创建文件副本的驱动器。
- 必须存在可供归档集使用的卷，并且它们有足够的空间来容纳归档请求中的文件。

驱动器

如果归档集指定了 `-drives` 参数，则 `sam-archiverd` 守护进程会将归档请求中选定的文件分配至多个驱动器中。如果此时可用的驱动器数量少于 `-drives` 参数指定的数量，则使用实际可用的数量。

如果归档请求中的文件大小总量小于 `-drivemin` 的值，则只使用一个驱动器。`-drivemin` 值可以是 `-drivemin` 参数指定的值，也可以是 `archmax` 值。`archmax` 值是由 `-archmax` 参数指定的值或为此介质定义的值。有关 `-archmax` 参数和 `archmax=` 指令的更多信息，请参见 `archiver.cmd(4)` 手册页。

如果归档请求中的文件大小总量大于 `-drivemin` 值，则所用驱动器的数量根据文件大小总量除以 `-drivemin` 值来确定。如果所用驱动器的数量少于 `-drives` 参数指定的驱动器数量，则采用实际所用驱动器的数量。

不同的驱动器归档文件所花的时间各不相同。可以使用 `-drivemax` 参数以获得最佳的驱动器利用率。`-drivemax` 参数要求您在重新预定驱动器接受更多数据前，指定将写入该驱动器的最大字节数。

卷

要使归档能够执行，至少必须存在一个卷，该卷具有足够空间，至少可以容纳归档请求中的某些文件。如果归档集最近使用过的卷具有足够空间，则它即为所预定的卷。该卷不应是归档程序正在使用的卷。

如果可用于归档集的卷正在使用中，除非指定了 `-fillvsns` 参数，否则会选择不其他卷。如果指定了该参数，则不能预定归档请求。

如果归档请求太大，无法载入一个卷中，则系统会选择只将这个卷所能容纳的文件归档于其中。如果归档请求包含的文件太大，无法载入一个卷中，并且未为归档请求选择卷溢出功能，则这些文件无法归档。此时，系统会将一则说明此情况的消息发送至日志中。

您可以使用 `-ovflmin` 参数为归档集指定卷溢出功能，或使用 `ovflmin=` 指令为介质指定卷溢出功能。有关 `-ovflmin` 参数和 `ovflmin=` 指令的更多信息，请参见 `archiver.cmd(4)` 手册页。 `ovflmin` 规范决定文件大小阈值，如果文件大小超过该阈值，将指定其他卷或介质进行归档。为归档集指定的 `ovflmin` 值优先于为介质指定的 `ovflmin` 值。

如果文件的大小小于 `ovflmin` 的值，则文件无法归档。此时，系统会将一则说明此情况的消息发送至日志中。如果文件的大小大于 `ovflmin` 的值，则会根据需要指定其他卷。系统将按卷容量的大小顺序选择卷（即先大后小），以最大限度地减少所需卷的数量。如果找不到可用于归档请求的卷，则归档请求会等待，直至具有可用卷为止。

在确定某个特定归档请求的预定优先级时，除归档优先级之外，系统还将参照某些其他属性，例如文件是处于联机状态还是处于脱机状态。有关自定义属性乘数的更多信息，请参见 `archiver.cmd(4)` 手册页中所述的 `-priority` 参数。

对于每一个归档请求，`sam-archiverd` 守护进程均会通过计算归档优先级和各种与系统资源属性相关联的乘数之和，来确定其预定优先级。这些属性与以下各项相关：归档请求排队的时间（秒数）；归档过程中使用的第一个卷是否已载入驱动器；以及其他方面。

`sam-archiverd` 守护进程利用经调整的优先级，指定每一个可进行复制的归档请求，如下节所述。

步骤 4：对归档请求中的文件进行归档

当归档请求已经准备就绪，可以进行归档时，`sam-archiverd` 守护进程将标记归档文件 (`tarball`) 的界限，以使每一个归档文件的大小小于指定的 `-archmax` 值。如果单个文件的大小大于该值，它将成为归档文件中的唯一文件。

对于每一个归档请求和要使用的每一个驱动器，`sam-archiverd` 守护进程均会将归档请求指定给 `sam-arcopy` 进程，以便将文件复制到归档介质。归档信息会输入至 `inode`。

如果已启用归档日志功能，则会创建归档日志条目。

对于每一个已登台的文件，系统会释放其磁盘空间，直到列表中的所有文件归档完毕。

多种错误以及文件状态改变都会导致文件复制失败。这些错误可能包括读取磁盘高速缓存和向卷写入数据时发生的错误。状态的改变可能包括在选择文件后对文件进行了修改，打开文件以写入数据，或文件被删除。

`sam-arcopy` 进程退出后，`sam-archiverd` 守护进程将检查归档请求。如果有文件尚未归档，则会重新组合归档请求。

您还可以在 `archiver.cmd` 文件中指定归档操作超时。指令如下所示：

```
timeout = [operation | media] time
```

`operation` 可以为以下各项之一：

- `read`：从磁盘中读取文件。默认超时时间为 1 分钟。
- `request`：请求归档介质。默认超时时间为 15 分钟。
- `stage`：登台要归档的文件。默认超时时间为 15 分钟。
- `write`：写入归档介质。默认超时时间为 15 分钟。

也可以针对单个介质指定写入操作的超时值。

归档请求队列时间通知

您可以在 `archiver.cmd` 文件中配置 `-queue_time_limit time` 归档集参数，以便在归档请求位于预定队列中的时间超过特定时间时向管理员发出通知。如果归档请求在指定的时间结束时仍位于队列中，则会向系统管理员发送电子邮件。

默认输出样例

代码示例 3-1 显示了 archiver(1M) -l 命令的输出样例。

代码示例 3-1 archiver(1M) -l 命令的输出

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

归档日志文件和事件日志

sam-arfind 和 sam-arcopy 进程使用 syslog 工具和 archiver.sh 在日志文件中记录警告及信息性消息，该日志文件包含每一个已归档文件或自动取消归档的文件的有关信息。日志文件连续地记录归档操作。您可以使用日志文件查找文件的早期副本，这些副本是传统备份的产物。

默认情况下，系统不会生成此文件。可以在 archiver.cmd 文件中使用 logfile= 指令，以指定要创建日志文件，并指定其名称。有关日志文件的更多信息，请参见本章第 46 页的“使用归档程序指令”和 archiver.cmd(4) 手册页。

代码示例 3-2 显示了一些摘自归档程序日志文件的示例行，这些行包括了每个字段的定义。

代码示例 3-2 归档程序日志文件的内容

```
A 2001/03/23 18:42:06 mo 0004A arset0.1 9a089.1329 samfs1 118.51
162514 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
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
```

表 3-1 按从左至右的顺序列出了上面示例中各字段的内容。

表 3-1 归档程序日志文件的字段

字段	示例值	内容
1	A	归档活动，具体如下： <ul style="list-style-type: none"> • A 表示已归档。 • R 表示已重新归档。 • U 表示已取消归档。
2	2001/03/23	归档操作发生的日期，格式为 <i>yyyy/mm/dd</i> 。
3	18:42:06	归档活动发生的时间，格式为 <i>hh:mm:ss</i> 。
4	mo	归档介质类型。有关介质类型的信息，请参见 <i>mcf(4)</i> 手册页。
5	0004A	VSN。对于可移除介质卡盒，这是卷序列名。对于磁盘归档，这是磁盘卷名以及 <i>tar(1)</i> 归档文件的路径。
6	arset0.1	归档集和副本份数。
7	9a089.1329	归档文件（ <i>tar(1)</i> 文件）在介质上的起始物理位置和归档文件中的文件偏移量（采用十六进制格式）。
8	samfs1	文件系统名。
9	118.51	Inode 编号和生成编号。为了保持唯一性，除了 inode 编号之外，还要使用生成编号，因为 inode 编号可以重复使用。
10	162514	在文件仅写入一个卷时，表示文件大小。在文件写入多个卷时，表示文件片段的大小。
11	t0/fdn	相对于文件系统挂载点的文件路径以及名称。
12	f	文件类型，具体如下： <ul style="list-style-type: none"> • d 表示目录。 • f 表示普通文件。 • l 表示符号链接。 • R 表示可移除的介质文件。 • I 表示段索引。 • S 表示数据段。
13	0	溢出文件的一部分或段。如果这个文件是溢出文件，则该值不为零，而对于所有其他文件类型，该值均为零。
14	56	文件归档至的驱动器的设备序号。

数据验证

如果需要，可以为归档副本启用数据验证。此功能将检查复制到第二和/或第三介质的任何数据是否出现数据损坏情况。

数据验证过程将执行写后读 (read-after-write) 验证测试，并在此文件的元数据属性中记录数据有效性的确认信息。

可以使用 `ssum` 选项将文件和目录标记为需要进行验证。子目录可继承其父目录的数据验证属性。可以使用常规的校验和方法来验证写入磁带或磁盘归档文件中的副本。

可以使用 `ssum -e` 命令设置文件或目录的数据验证。这样可强制生成和使用校验和以进行归档和登台操作，并防止在创建完所有归档副本并验证其校验和之前释放文件。只有超级用户才能针对文件或目录设置此属性。

注 - 数据验证功能会给登台程序资源带来额外的负担，因为数据验证请求将与常规登台请求一起放入登台程序队列中。数据验证还会导致挂载更多的磁带，因此可能会影响归档程序和登台程序的性能。由于直至创建了所有归档副本后才能释放文件，因此，使用数据验证还可能需要更多的磁盘高速缓存。

关于 `archiver.cmd` 文件

`archiver.cmd` 文件用于控制归档程序的行为。默认情况下，只要一启动 `sam-fsd` 并挂载 Sun StorageTek SAM 文件系统，归档程序就开始运行。当没有 `archiver.cmd` 文件时，归档程序使用以下默认设置：

- 将所有文件归档至所有可用卷。
- 所有文件的归档时限为 4 分钟。
- 归档时间间隔为 10 分钟。

可以使用归档程序命令文件 (`archiver.cmd`) 中的指令对归档程序的操作进行自定义，以满足站点的归档要求。

▼ 创建或修改 archiver.cmd 文件并传播更改

如果不采用此方法，还可以使用 File System Manager 软件来创建或修改 archiver.cmd 文件。有关更多信息，请参见 File System Manager 联机帮助。

1. (可选) 确定是希望编辑实际的 archiver.cmd 文件，还是要编辑临时的 archiver.cmd 文件。

如果您具有 /etc/opt/SUNWsamfs/archiver.cmd 文件，且系统已在对文件进行归档，请执行此步骤。可考虑将 archiver.cmd 文件复制到一个临时位置，以利于对它进行编辑并在正式使用前进行测试。

2. 使用 vi(1) 或其他编辑器编辑该文件。

为了控制您的站点的归档操作，可根据您的需要添加指令。有关可在这个文件中包括哪些指令的信息，请参见第 46 页的“使用归档程序指令”和第 83 页的“关于磁盘归档”。

3. 保存并关闭该文件。

4. 使用 archiver(1M) -lv 命令验证文件的正确性。

无论您何时更改 archiver.cmd 文件，均应使用 archiver(1M) 命令检查其中是否存在语法错误。使用如下所示的 archiver(1M) 命令，在当前 Sun StorageTek SAM 系统中检验 archiver.cmd 文件：

```
# archiver -lv
```

该命令将生成所有选项的列表，并将 archiver.cmd 文件、卷、文件系统内容和错误的列表写入标准输出文件(stdout)。出现错误时，归档程序将停止运行。

默认情况下，archiver(1M) 命令会检验 /etc/opt/SUNWsamfs/archiver.cmd 文件中的错误。如果正在使用临时的 archiver.cmd 文件，则可在 archiver(1M) 命令中使用 -c 选项并提供这个临时文件的名称。

5. 如果遇到错误，请在文件中更正它们，然后再次运行 archiver(1M) 命令以验证您的更正。

在执行下一步骤前，必须更正所有错误。如果归档程序发现 archiver.cmd 文件中有错误，则它不会对任何文件进行归档。

6. 如果您正在使用临时文件，请将它移动到 /etc/opt/SUNWsamfs/archiver.cmd。

7. 使用 samd(1M) config 命令传播文件更改并重新启动系统。

```
# samd config
```

archiver.cmd 文件

archiver.cmd 文件由以下类型的指令组成:

- 常规指令
- 归档集分配指令
- 归档集指令
- VSN 池指令
- VSN 关联指令

这些指令由从 archiver.cmd 文件读取的文本行组成。每一行指令包含一个或多个字段，它们由空格或制表符隔开。井字符 (#) 后面的任何文本均被视为注释，并且不会受到检查。您可以在第一行的末尾添加一个反斜杠 (\) 将长指令续接至下一行。

archiver.cmd 文件中的某些指令会要求您指定时间单位或字节单位。要指定这样的单位，请使用表 3-2 中的字母之一。

表 3-2 archiver.cmd 文件指令单位

单位后缀	说明
时间后缀:	
s	秒。
m	分钟
h	小时
d	天
w	周
Y	年
大小后缀:	
b	字节
k	千字节 (KB)
M	兆字节 (MB)
G	千兆字节 (GB)
T	兆兆字节 (TB)
P	千兆兆字节 (PB)
E	兆兆兆字节 (EB)

archiver.cmd 文件示例

代码示例 3-3 显示了 archiver.cmd 文件示例。注释指出了各种指令类型。

代码示例 3-3 archiver.cmd 文件示例

```
# Global directives

archivemeta = off
archmax = li 8G
examine = noscan
scanlist_squash = on

# limit the drives

drives = stk50 X

# File selection

fs = samfs1
logfile = /var/adm/samfs1.log
archive .
    1 -norelease 5m
    2 -norelease 5m
fs = samfs2
logfile = /var/adm/samfs2.log
archive .
    1 -norelease 5m
    2 -norelease 5m
fs = samfs3
logfile = /var/adm/samfs3.log
archive .
    1 -norelease 5m
    2 -norelease 5m
fs = samfs4
logfile = /var/adm/samfs4.log
archive .
    1 -norelease 5m
    2 -norelease 5m
```

代码示例 3-3 archiver.cmd 文件示例 (续)

```
fs = samfs5
logfile = /var/adm/samfs5.log
archive .
    1 -norelease 5m
    2 -norelease 5m

# The below information is for tape archiving.
# The recycler is not set up to actually recycle. It is set up for
# data checks and file recovery.

params
allsets -sort path -offline_copy stageahead -reserve set
allsets -recycle_ignore
allsets -recycle_hwm 50 -recycle_mingain 90 -recycle_vsncount 1
allsets.1 -startage 6h -startsize 8G -startcount 90000 -drives X -drivemin XXG
allsets.2 -startage 10h -startsize 12G -startcount 90000 -archmax 12G -drives
X -drivemin XXG
endparams

# Define VSNS for archive sets

vsns
archive.1 li .*
archive.2 li .*
endvsns
```

使用归档程序指令

以下几节对 `archiver.cmd` 指令进行介绍。这些指令是：

- [第 46 页的“全局归档指令”](#)
- [第 54 页的“文件系统指令”](#)
- [第 55 页的“归档集分配指令”](#)
- [第 63 页的“归档副本指令”](#)
- [第 66 页的“归档集副本参数”](#)
- [第 79 页的“VSN 关联指令”](#)
- [第 81 页的“VSN 池指令”](#)

全局归档指令

全局指令用于控制归档程序的整体操作，并可用于优化站点配置的归档程序操作。可以在 `archiver.cmd` 文件中直接添加全局指令，也可以使用 **File System Manager** 软件指定全局指令。有关使用 **File System Manager** 设置全局指令的更多信息，请参见 **File System Manager** 联机帮助。

`archiver.cmd` 文件中的全局指令可以通过第二个字段中的等号 (=) 或不出现其他字段的方式来识别。

在 `archiver.cmd` 文件中，全局指令必须位于所有 `fs=` 指令前面。`fs=` 指令是那些专用于特定文件系统的指令。如果归档程序检测到全局指令位于 `fs=` 指令后面，则会发出一则消息。

archivemeta 指令：控制是否对元数据进行归档

archivemeta 指令控制是否对文件系统元数据进行归档。如果文件系统中的文件经常移动，且目录结构经常发生更改，则最好对文件系统元数据进行归档。相反，如果目录结构非常稳定，则可以禁用元数据归档，从而减少当载入和卸载卡盒时可移除介质驱动器所执行的操作。默认情况下，会对元数据进行归档。

此指令的格式如下：

```
archivemeta=state
```

其中的 *state*，用于指定状态为 on 还是 off。默认设置是 on。

元数据的归档过程取决于您所使用的超级块是第 1 版还是第 2 版，具体如下：

- 如果使用的是第 1 版的文件系统，归档程序会将目录、可移除的介质文件、段索引 inode 以及符号链接归档为元数据。
- 如果使用的是第 2 版的文件系统，归档程序会将目录和段索引 inode 归档为元数据。可移除的介质文件和符号链接会存储在 inode 中，而不是存储在数据块中。归档程序不会对它们进行归档。符号链接则归档为数据。

archmax 指令：控制归档文件的大小

archmax 指令用于指定归档文件的最大大小。将用户文件组合在一起，形成归档文件。达到 *target-size* 值后，将不能再向归档文件中添加用户文件。大型用户文件将写入单个归档文件中。

要更改默认值，请使用以下指令：

```
archmax=media target-size
```

表 3-3 archivemeta 指令的参数

参数	含义
<i>media</i>	介质类型。有关有效介质类型的列表，请参见 mcf(4) 手册页。
<i>target-size</i>	归档文件的最大大小。该值与介质有关。默认情况下，写入光盘的归档文件不得超过 5 MB。对于磁带，归档文件的最大默认大小为 512 MB。

将归档文件的大小设置为较大的值或较小的值都各有优缺点。例如，在使用磁带进行归档时，将 `archmax` 设置成较大的值，可以减少磁带机停止和启动的次数。但是，在写入较大的归档文件时，有可能提前到达磁带末尾，因而浪费了大量磁带空间。一般而言，`archmax` 的设置值不应超过介质容量的百分之五。

此外，您还可为单个归档集设置 `archmax` 指令。

注 – 对于要归档至 Sun StorageTek 5800 介质类型的归档集，`archmax` 指令为无效指令。

bufsize 指令：设置归档程序缓冲区大小

默认情况下，系统使用内存缓冲器将需要归档的文件复制到归档介质。可以使用 `bufsize` 指令来设置非默认的缓冲区大小和锁定缓冲区（可选）。这些操作可以改善系统的性能。您可以尝试不同的 `buffer-size` 值以确定最适合的缓冲区大小。

此指令的格式如下：

```
bufsize=media buffer-size [lock]
```

表 3-4 `bufsize` 指令的参数

参数	含义
<code>media</code>	介质类型。有关有效介质类型的列表，请参见 <code>mcf(4)</code> 手册页。
<code>buffer-size</code>	一个介于 2 到 1024 之间的数字。默认值为 4。这个值乘以该介质类型的 <code>dev_blksize</code> 值，计算结果即为所使用的缓冲区大小。 <code>dev_blksize</code> 的值是在 <code>defaults.conf</code> 文件中指定的。有关此文件的更多信息，请参见 <code>defaults.conf(4)</code> 手册页。
<code>lock</code>	指明归档程序在创建归档副本时是否使用锁定的缓冲区。如果指定 <code>lock</code> ，则归档程序将在 <code>sam-arcopy(1M)</code> 操作期间在内存中的归档缓冲区上设置文件锁定。这可以避免由于为每一个 I/O 请求锁定和取消锁定缓冲区而造成的开销，从而减少占用系统 CPU 的时间。 仅在配有大量内存的大型系统上，才有必要指定 <code>lock</code> 参数。如果内存不足，则可能会导致内存用尽。 只有已为需要归档的文件启用直接 I/O 时， <code>lock</code> 参数才有效。默认情况下，不会指定 <code>lock</code> 参数，文件系统会在所有直接 I/O 缓冲区上设置锁定（也包括用于归档的缓冲区）。有关启用直接 I/O 的更多信息，请参见 <code>setfa(1)</code> 手册页、 <code>sam_setfa(3)</code> 库例程手册页，或 <code>mount_samfs(1M)</code> 手册页中介绍的 <code>-O forcedirectio</code> 选项。

您可以使用 `-bufsize` 和 `-lock` 归档集副本参数，以归档集为单元指定缓冲区的大小和锁定。有关详细信息，请参见第 66 页的“归档集副本参数”。

drives 指令：控制用于归档的驱动器数

默认情况下，归档程序使用自动化库中的所有驱动器进行归档。要限制所用驱动器的数量，请使用 `drives` 指令。

此指令的格式如下：

```
drives=auto-lib count
```

表 3-5 drives 指令的参数

参数	含义
<i>auto-lib</i>	mcf 文件中定义的自动化库的系列集名。
<i>count</i>	用于归档活动的驱动器数量。

另请参见第 68 页的“指定归档请求的驱动器数：-drivemax、-drivemin 和 -drives”，其中介绍了 -drivemax、-drivemin 和 -drives 归档集副本参数。

examine 指令：控制归档扫描

新文件以及已更改的文件都是备选的归档文件。归档程序会通过以下某一方法来查找此类文件：

- 连续归档。当进行连续归档时，归档程序对文件系统进行处理，检测其中刚刚发生更改的文件。
- 基于扫描的归档。当进行基于扫描的归档时，归档程序会定期扫描文件系统，寻找需要归档的文件。

此指令的格式如下：

```
examine=method
```

其中的 *method*，用于指定表 3-6 中所示的关键字之一。

表 3-6 examine 指令的 *method* 参数的值

<i>method</i> 值	含义
<code>noscan</code>	指定连续归档。在执行初次扫描后，仅当目录内容发生更改以及需要进行归档时，才会对其进行扫描。但不对其目录和 <code>inode</code> 信息进行扫描。这种归档方法的性能要好于基于扫描的归档，尤其是当文件系统中文件的数量大于 1,000,000 时。默认设置。
<code>scan</code>	指定基于扫描的归档。第一次的文件系统扫描是执行目录扫描。接下来对 <code>inode</code> 进行扫描。
<code>scandirs</code>	指定仅对目录执行基于扫描的归档。如果归档程序查找到设置了 <code>no_archive</code> 属性的目录，将不对该目录进行扫描。这种目录中含有未更改的文件，跳过这种目录这可以大大缩短归档扫描时间。
<code>scaninodes</code>	指定仅对 <code>inode</code> 执行基于扫描的归档。

interval 指令：指定归档时间间隔

归档程序定期运行以检查所有已挂载的 Sun StorageTek SAM 文件系统的状态。检查时间由归档时间间隔控制。归档时间间隔是指对每一个文件系统执行扫描操作的时间间隔。要更改时间间隔，请使用 `interval` 指令。

仅当未设置连续归档和未指定 `startage`、`startsize` 或 `startcount` 参数时，`interval` 指令才启动完全扫描。如果已设置连续归档 (`examine=noscan`)，则 `interval` 指令将作为默认的 `startage` 值。

此指令的格式如下：

```
interval=time
```

其中的 *time*，用于指定对文件系统执行扫描操作的所希望时间间隔。默认情况下，*time* 以秒计，其值为 600，即 10 分钟。可以指定不同的时间单位，如分钟或小时，如表 3-2 中所述。

如果归档程序接收到 `samu(1M)` 实用程序的 `:arrun` 命令，则将会立即开始扫描所有文件系统。如果 `archiver.cmd` 文件中还指定了 `examine=scan` 指令，则会在运行 `:arrun` 或 `:arscan` 后进行扫描。

如果为文件系统设置了 `hwm_archive` 挂载选项，则可以自动缩短归档时间间隔。此挂载选项指定归档程序在文件系统填满且超过空间占用上限时即开始进行扫描。`high=percent` 挂载选项用于设置文件系统空间占用的上限。

有关指定归档时间间隔的更多信息，请参见 `archiver.cmd(4)` 手册页。有关设置挂载选项的更多信息，请参见 `mount_samfs(1M)` 手册页。

logfile 指令：指定归档程序日志文件

归档程序可以生成一个日志文件，其中包含每一个归档、重新归档或自动取消归档的文件的有关信息。日志文件连续地记录归档操作。要指定日志文件，请使用 `logfile` 指令。

此指令的格式如下：

```
logfile=pathname
```

其中的 *pathname* 用于指定日志文件的绝对路径和名称。默认情况下，系统不会生成此文件。

此外，您还可为单个文件系统设置 `logfile` 指令。

▼ 备份归档程序日志文件

假定您希望通过将前一天的日志文件复制到另一位置，以实现归档程序日志文件的每日备份。确保在归档程序日志文件关闭时，而不是在它打开以进行写入操作时执行复制。

1. 使用 `mv(1)` 命令在 UNIX 文件系统中移动归档程序日志文件。

这可给予 `sam-arfind(1M)` 或 `sam-arcopy(1M)` 一定的操作时间，以完成向归档程序日志文件写入信息。

2. 使用 `mv(1)` 命令将前一天的归档程序日志文件移至 Sun StorageTek SAM 文件系统。

notify 指令：重命名事件通知脚本

`notify` 指令用于设置归档程序的事件通知脚本文件的名称。此指令的格式如下：

```
notify=filename
```

其中的 *filename*，用于指定包含归档程序事件通知脚本的文件名称，或指定该文件的完整路径。

默认文件名如下所示：

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

归档程序执行此脚本，来根据特定站点的要求处理各种事件。可以通过第一个参数的以下关键字之一调用此脚本：`emerg`、`alert`、`crit`、`err`、`warning`、`notice`、`info` 和 `debug`。

其他参数在默认脚本中加以说明。有关更多信息，请参见 `archiver.sh(1M)` 手册页。

ovflmin 指令：控制卷溢出

利用卷溢出功能，可以在多个卷上归档文件。在 `archiver.cmd` 文件中使用 `ovflmin` 指令即可启用卷溢出功能。当文件的大小超过 `ovflmin` 指令的 `minimum-file-size` 参数的值时，归档程序会将此文件的一部分写入至另一个同类型的可用卷中。写入至每一个卷的文件部分称为片段。

注 – 请慎用卷溢出功能，只有在全面检验卷溢出对您的站点所产生的影响后，才可以使用该功能。文件归档在多个卷上会严重地加大故障恢复操作和回收操作的难度。有关更多信息，请参见《Sun StorageTek Storage Archive Manager 故障排除指南》和 `request(1)` 手册页。

归档程序通过 `ovflmin` 指令来控制卷溢出功能。`ovflmin` 指令指定触发溢出过程的文件大小阈值。默认情况下，归档程序会禁用卷溢出功能。

此指令的格式如下：

```
ovflmin = media minimum-file-size
```

表 3-7 `ovflmin` 指令的参数

参数	含义
<code>media</code>	介质类型。有关有效介质类型的列表，请参见 <code>mcf(4)</code> 手册页。
<code>minimum-file-size</code>	触发卷溢出的最小文件大小。

假定许多文件归档在 `mo` 介质卡盒中，并且产生严重的碎片（例如 25%）。这些文件将导致每个卷中存有大量的未用空间。为了更好地利用卷内空间，请将 `mo` 介质的 `ovflmin` 设置成略小于最小文件大小的值。以下指令将其设置为 150 MB：

```
ovflmin=mo 150m
```

请注意，在本示例中启用卷溢出功能还会造成在归档和登台文件时需要载入两个卷，因为每一个文件将溢出到其他卷。

此外，您还可为单个归档集设置 `ovflmin` 指令。

`sls(1)` 命令输出可列出归档副本，并显示各 VSN 上文件的每个片段。[代码示例 3-4](#) 显示了归档程序日志文件，[代码示例 3-5](#) 显示了一个大型文件的 `sls -D` 命令输出，这个大型文件名为 `file50`，归档在多个卷上。

代码示例 3-4 归档程序日志文件示例

```
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
```

[代码示例 3-4](#) 显示了 `file50` 归档在三个卷上，其 VSN 分别为 `DLT000`、`DLT001` 和 `DLT005`。每个片段在卷上的位置和大小显示在第七个和第十个字段中（第一个条目分别是 `7eed4.1` 和 `477609472`），它们与[代码示例 3-5](#) 中显示的 `sls -D` 输出相匹配。有关归档日志条目的完整描述，请参见 `archiver(1M)` 手册页。

[代码示例 3-5](#) 显示了 `sls -D` 命令及其输出。

代码示例 3-5 `sls(1M) -D` 命令及其输出

```
# 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
```

卷溢出文件不能生成校验和。有关使用校验和的更多信息，请参见 `ssum(1)` 手册页。

scanlist_squash 指令：控制扫描列表合并

`scanlist_squash` 参数用于打开或关闭扫描列表合并。默认设置为 `off`。此参数可以是全局性参数，也可以是文件系统专用参数。

当启用此选项时，同一父目录（稍后需要使用 `sam-arfind` 扫描该目录）下的两个或多个子目录中的文件的扫描列表条目会被合并。如果在其中许多子目录都有大量更改的文件系统上执行归档操作，会引发严重的性能降低问题。合并扫描列表时，这些目录向上合并为一个公共的父目录，这将会导致对许多子目录进行深度递归扫描。

setarchdone 指令：控制 archdone 标志的设置

setarchdone 参数是一个全局指令，用于在 sam-arfind 检查文件时，控制 archdone 标志的设置。

此指令的格式如下：

```
setarchdone=on|off
```

创建文件的所有归档副本后，将为该文件设置 archdone 标志，以表示无需进行其他归档操作。在 inode 扫描过程中，归档程序会检测是否已设置 archdone 标志，如果已设置，则归档程序不查找 inode 的路径名。

在目录扫描过程中，归档程序还为从不归档的文件设置 archdone 标志。当扫描大型目录时，这可能是一项费时的操作，并可能影响性能。您可以通过 setarchdone 指令对此活动进行控制。如果将 examine 指令设置为 scandirs 或 noscan，则该指令的默认设置为 off。

该指令只控制从不归档的文件的 archdone 标志的设置。在创建归档副本后，该指令不会影响 archdone 标志的设置。

wait 指令：延迟归档程序的启动

wait 指令会促使归档程序等待来自 samu(1M) 或 File System Manager 的启动信号。默认情况下，归档程序在由 sam-fsd(1M) 启动后即开始归档操作。

此指令的格式如下：

```
wait
```

此外，您还可为单个文件系统设置 wait 指令。

文件系统指令

可在 archiver.cmd 文件的常规指令后面添加专用于特定文件系统的 fs= 指令。遇到 fs= 指令后，归档程序假定后面的所有指令均仅适用于特定的文件系统。

可以按照以下几节所述通过编辑 archiver.cmd 文件，或者通过使用 File System Manager 软件来指定 fs= 指令。有关更多信息，请参见 File System Manager 联机帮助。

fs 指令：指定文件系统

默认情况下，归档控制指令适用于所有文件系统。不过，您可以将某些控制指令的适用范围限定在特定的文件系统上。例如，您可以使用此指令为每一个文件系统指定不同的日志文件。要指定特定的文件系统，可使用 `fs` 指令。

此指令的格式如下：

```
fs=fsname
```

其中的 *fsname*，用于指定在 `mcf` 文件中定义的文件系统名。

这些指令后面出现的常规指令和归档集相关指令仅适用于指定的文件系统，直到出现下一个 `fs=` 指令。

全局指令和文件系统指令

有几个指令既可以指定为适用于所有文件系统的全局指令，也可以指定为专用于某一文件系统的指令。这几个指令如下：

- `interval` 指令。有关此指令的更多信息，请参见第 50 页的“[interval 指令：指定归档时间间隔](#)”。
- `logfile` 指令。有关此指令的更多信息，请参见第 51 页的“[logfile 指令：指定归档程序日志文件](#)”。
- `scanlist_squash` 指令。有关此指令的更多信息，请参见第 53 页的“[scanlist_squash 指令：控制扫描列表合并](#)”。
- `wait` 指令。有关此指令的更多信息，请参见第 54 页的“[wait 指令：延迟归档程序的启动](#)”。

归档集分配指令

默认情况下，文件将作为根据文件系统命名的归档集的一部分进行归档。不过，您可以指定其他归档集以包含具有类似特征的文件。如果某个文件不属于您指定的任何归档集，则它将作为根据文件系统命名的默认归档集的一部分进行归档。

可以按照以下几节所述通过直接编辑 `archiver.cmd` 文件，或者通过使用 `File System Manager` 软件来创建归档集。在 `File System Manager` 中，*archive policy* 用于定义归档集。有关更多信息，请参见 `File System Manager` 联机帮助。

指定归档集

归档集成员指令可以将具有类似特征的文件指定给同一个归档集。这些指令的语法与 `find(1)` 命令类似。每一个归档集分配指令均采用以下格式：

```
archive-set-name path [search-criterion1 search-criterion2 ... ] [file-attribute1 file-attribute2 ... ]
```

表 3-8 归档集分配指令的参数

参数	含义
<code>archive-set-name</code>	您的站点为归档集定义的名称。它必须是归档集分配指令中的第一个字段。归档集名通常暗示属于该归档集的文件特征。指定归档集名时，只能使用 26 个英文字母、数字 (0-9) 和下划线字符 (<code>_</code>)，而不得使用其他特殊字符或空格。归档集名的第一个字符必须是字母。 要阻止对各种文件进行归档，请将 <code>archive-set-name</code> 值指定为 <code>no_archive</code> 。
<code>path</code>	相对于文件系统挂载点的路径。这使得归档集成员指令可适用于多个 Sun StorageTek SAM 文件系统。如果要使该路径包括文件系统的所有文件，请在路径字段中输入句点 (<code>.</code>)。不允许在路径的开头使用斜杠 (<code>/</code>)。 <code>path</code> 所指定目录中的文件及其子目录均视为属于该归档集。
<code>search-criterion1</code> <code>search-criterion2</code>	可以指定零个、一个或多个 <code>search-criterion</code> 参数。指定搜索标准的目的是根据文件大小、文件拥有权以及其他要素限制归档集的范围。有关可用的 <code>search-criterion</code> 参数的信息，请参见以下几节。
<code>file-attribute1</code> <code>file-attribute2</code>	可以指定零个、一个或多个 <code>file-attribute</code> 值。当 <code>sam-arfind</code> 进程在归档期间扫描文件系统时，将为文件设置这些文件属性。

代码示例 3-6 显示了典型的归档集成员指令。

代码示例 3-6 归档集成员指令

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

将文件归入名为 `no_archive` 的归档集中，可以阻止归档程序对这些文件进行归档。代码示例 3-7 显示了阻止对 `tmp` 目录的各级子目录下的文件进行归档的指令行，而且不管 `tmp` 目录在文件系统中的位置如何，都是如此。

代码示例 3-7 阻止归档的归档指令

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

以下几节介绍了可以指定的 `search-criterion` 参数。

文件时限 *search-criterion*: `-access` 和 `-nftv`

可使用 `-access age` 特征来指定用于确定归档集成员的文件时限。如果使用这个特征，访问时间早于 *age* 的文件将被重新归档至其他介质中。其中的 *age*，用于指定一个整数，这个整数后跟表 3-9 中所示的后缀之一。

表 3-9 `-access age` 的后缀

后缀	含义
s	秒
m	分钟
h	小时
d	天
w	周
y	年

例如，可使用这个指令来指定将长时间未访问过的文件重新归档至一个较为便宜的介质中。

在确定时限时，软件将验证文件的访问和修改时间，以确保这些时间大于或等于文件的创建时间，且小于或等于文件的检查时间。对于已“迁移”至某个目录中的文件，此验证可能不会产生预期的行为。在这种情况下，可以使用 `-nftv` (**N**o **F**ile **T**ime **V**alidation, 非文件时间验证) 参数阻止对文件的访问和修改时间进行验证。

文件时限 *search-criterion*: `-after`

可以使用 `-after date-time` 特征将新修改或新创建的文件分组到同一归档集。如果使用这个特征，则只有在指定日期之后创建或修改的文件才可包含在归档集中。

date-time 的格式为 `YYYY-MM-DD[Thh:mm:ss][Z]` (ISO 8601 格式)。如果未指定时间部分，则会视为 `00:00:00`。如果出现 `z`，则表示会将时间解释为国际协调时间 (Coordinated Universal Time, UTC)；否则，会将其解释为本地时间。

文件大小 *search-criterion*: `-minsize` 和 `-maxsize`

可通过 `-minsize size` 和 `-maxsize size` 特征，使用文件的大小来确定归档集成员。其中的 *size*，用于指定一个整数，这个整数的后缀是表 3-10 中所示的某一字母。

表 3-10 `-minsize` 和 `-maxsize size` 的后缀

字母	含义
b	字节
k	千字节 (KB)
M	兆字节 (MB)
G	千兆字节 (GB)
T	兆兆字节 (TB)
P	千兆兆字节 (PB)
E	兆兆兆字节 (EB)

举例来讲，[代码示例 3-8](#) 中的行指定所有介于 500 KB 和 100 MB 之间的文件均属于归档集 `big_files`，超过 100 MB 的文件属于归档集 `huge_files`。

代码示例 3-8 `-minsize` 和 `-maxsize` 指令的使用示例

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

属主和组 *search-criterion*: `-user` 和 `-group`

可通过 `-user name` 和 `-group name` 特征，使用拥有权和组关系来确定归档集成员。在[代码示例 3-9](#) 中，所有属于用户 `sysadmin` 的文件均属于归档集 `adm_set`，所有组名称为 `marketing` 的文件均属于归档集 `mktng_set`。

代码示例 3-9 `-user` 和 `-group` 指令的使用示例

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

使用模式匹配的文件名 *search-criterion*: `-name regex`

可以使用正则表达式指定属于某个归档集的文件名称。作为一项 *search-criterion* 指令, `-name regex` 规范指定任何与正则表达式 *regex* 相匹配的完整路径均将成为归档集的成员。

regex 参数遵守 `regex(5)` 手册页中列出的约定。请注意, 正则表达式不遵守与 UNIX 通配符相同的约定。

所有位于选定目录之下的文件 (及其相对于文件系统挂载点的指定路径) 都将进行模式匹配。这使得您可以在 `-name regex` 字段中创建模式, 以匹配文件名和路径名。

示例

以下指令将归档集 `images` 中的文件限定为那些以 `.gif` 结尾的文件:

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

以下指令选择以字符 `GEO` 开头的文件:

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

您可以将正则表达式与 `no_archive` 归档集结合使用。以下规范可阻止对任何以 `.o` 结尾的文件进行归档:

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

假定您的 `archiver.cmd` 文件中包含[代码示例 3-10](#)中所示的指令行:

代码示例 3-10 正则表达式示例

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

对于此 archiver.cmd 文件，归档程序不会对用户目录或其子目录下的 fred.* 进行归档。如果指定了如代码示例 3-10 中所示的指令，则不对代码示例 3-11 中所示的文件进行归档。

代码示例 3-11 不进行归档的文件（使用代码示例 3-10 中所示的指令）

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

如果指定了如代码示例 3-10 中所示的指令，则将对代码示例 3-12 中所示的文件进行归档。

代码示例 3-12 进行归档的文件（使用代码示例 3-10 中所示的指令）

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

与代码示例 3-10 相对，假定您的 archiver.cmd 文件中包含代码示例 3-13 中所示的行。

代码示例 3-13 archiver.cmd 文件示例

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

代码示例 3-13 中的 archiver.cmd 文件不会对用户根目录下的 fred.* 文件进行归档。但会对用户子目录和 share/marketing 目录下的 fred.* 文件进行归档。在这种情况下，从 share/marketing/ 到下一个斜杠字符 (/) 之间的任何目录均被视为用户的主目录。因此，不会对以下文件进行归档：

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

如果指定了如[代码示例 3-13](#) 中所示的指令，则将对[代码示例 3-14](#) 中所示的文件进行归档。

代码示例 3-14 进行归档的文件（使用[代码示例 3-13](#) 中所示的指令）

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

释放和登台 *file-attributes*: `-release` 和 `-stage`

您可以通过分别使用 `-release` 和 `-stage` 选项，设置与归档集内文件相关联的释放和登台属性。这两种设置都会取代用户以前可能设置的释放或登台属性。

`-release` 选项的格式如下所示：

```
-release attribute
```

`-release` 指令的属性所遵守的约定与 `release(1)` 命令所遵守的约定相同，具体如[表 3-11](#) 所示。

表 3-11 `-release` 指令的属性

属性	含义
a	完成第一个归档副本后，释放文件占用的磁盘空间。
d	重置为默认设置。
n	不释放文件占用的磁盘空间。
p	释放文件占用的部分磁盘空间。

`-stage` 选项的格式如下所示：

```
-stage attribute
```

-stage 指令的属性所遵守的约定与 stage(1) 命令所遵守的约定相同，具体如表 3-12 所示。

表 3-12 -stage 指令的属性

属性	含义
a	联合登台本归档集中的文件。
d	重置为默认设置。
n	不登台本归档集中的文件。

以下示例显示了如何使用文件名规范和文件属性，来部分释放 Macintosh 资源目录：

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

归档集成员冲突

有时，在选择路径和其他文件特征以将文件归入归档集时，可能会造成归档集成员混乱。系统采用以下方式来解决此类情况：

1. 选择归档集中最初的成员定义。
2. 首先选择文件系统本地的成员定义，然后选择全局性的成员定义。
3. 如果某个成员定义与先前的成员定义完全相同，则系统会发出错误通知。

遵照这些规则，用户应在指令文件的前部设置更加严格的成员定义。

如果对特定文件系统的归档操作进行了控制（使用 `fs=fsname` 指令），则归档程序在检验全局指令之前，会先检验文件系统专用指令。这样，文件可以指定给本地归档集（包括 `no_archive` 归档集），而不是指定给全局归档集。在设置 `no_archive` 等全局归档集分配时，就默认采用此规则。

从代码示例 3-15 可以看出，管理员本不希望对以上两个文件系统中的任意 `.o` 文件进行归档。但是，由于对本地归档集分配 `allfiles` 的检验先于全局归档集分配 `no_archive`，因此仍会对 `samfs1` 和 `samfs2` 文件系统中的 `.o` 文件进行归档。

代码示例 3-15 可能存在成员冲突的 archiver.cmd 文件

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

代码示例 3-16 显示了确保能够不对以上两个文件系统中的 .o 文件进行归档的指令。

代码示例 3-16 更正后的 archiver.cmd 文件

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

归档副本指令

如果不指定归档副本，则归档程序将为归档集中的文件写入单独的归档副本。默认情况下，创建文件副本的归档时限为四分钟。如果您需要多个副本，则必须通过归档副本指令指定所有副本（包括第一个副本）。

归档副本指令以 *copy-number* 值 1、2、3 或 4 开头。数字后面是一个或多个用于指定该副本归档特征的参数。

归档副本指令必须紧跟在与它们相关的归档集分配指令的后面。每一个归档副本指令均采用以下格式：

```
copy-number [ -release | -norelease ] [archive-age] [unarchive-age]
```

可以按照这里介绍的方法通过编辑 `archiver.cmd` 文件，或者通过使用 `File System Manager` 软件来指定归档副本指令。有关更多信息，请参见 `File System Manager` 联机帮助。

以下几节介绍这些归档副本指令参数。

归档之后释放磁盘空间：-release

要指定在创建归档副本之后自动释放的文件所占用的磁盘空间，可以在副本份数后面使用 `-release` 指令。此指令的格式如下：

```
-release
```

在**代码示例 3-17** 中，属于组 `images` 的文件在其归档时限达到 10 分钟时进行归档。在创建第一个归档副本后，系统会释放文件占用的磁盘高速缓存空间。

代码示例 3-17 使用了 `-release` 指令的 `archiver.cmd` 文件

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

延迟释放磁盘空间：-norelease

您可能希望在创建多个归档副本之后再释放文件占用的磁盘空间。可以使用 `-norelease` 选项，它将使得系统在所有标记了 `-norelease` 的副本创建完毕之后，才自动释放磁盘高速缓存。

此指令的格式如下：

```
-norelease
```

使用 `-norelease` 指令，可以使归档集在所有副本均已归档后符合被释放的条件，但对于其中的文件，只有在系统调用释放程序并且将它们选择为释放备选文件后才会被释放。

代码示例 3-18 指定了一个名为 `vault_tapes` 的归档集。系统将为其创建两个副本，并且只有在这两个副本创建完毕之后，才释放与该归档集相关的磁盘高速缓存。

代码示例 3-18 使用了 `-norelease` 指令的 `archiver.cmd` 文件

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

对单个副本使用 `-norelease` 指令对自动释放没有效果。

同时使用 -release 和 -norelease

如果要确保归档集在所有副本完成归档之后立即释放其占用的磁盘空间，则您可以同时使用 `-release` 和 `-norelease` 选项。如果同时使用 `-release` 和 `-norelease`，则可使归档程序在创建使用这两个选项组合的所有副本之后立即释放磁盘空间，而不像仅使用 `-norelease` 选项那样，需要等待调用释放程序。

设置归档时限

您可以在归档副本指令中指定归档时限来设置文件的归档时限。归档时限的后缀字符可以指定为 h（代表小时）或 m（代表分钟），如表 3-2 中所示。

在代码示例 3-19 中，目录 data 中的文件将在其归档时限到达 1 个小时后进行归档。

代码示例 3-19 指定了归档时限的 archiver.cmd 文件

```
ex_set data
  1 1h
```

自动取消归档

如果您为文件指定多个归档副本，则可能需要只保留一个归档副本，而自动取消对其他所有副本进行归档。当使用不同归档时限将文件归档至不同介质时，您可能希望执行此操作。

代码示例 3-20 显示了指定取消归档时限的指令。路径 home/users 中文件的第一个副本将在文件修改后六分钟进行归档。当文件的归档时间达到 10 周时，系统会创建第二个和第三个归档副本，并且会取消对第一个副本进行归档。

代码示例 3-20 指定了取消归档时限的 archiver.cmd 文件

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

有关控制取消归档的其他方法，请参见第 72 页的“控制取消归档”。

为元数据指定多个副本

如果需要多个元数据副本，则可以在指令文件中放置副本份数定义，其位置是紧跟在 fs= 指令后。

代码示例 3-21 显示了一个指定了多个元数据副本的 archiver.cmd 文件。

代码示例 3-21 指定了多个元数据副本的 archiver.cmd 文件

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

在本示例中，系统将在 4 小时后创建 `samfs7` 文件系统的元数据的第一个副本，然后在 12 小时后创建第二个副本。

文件系统元数据包括文件系统中的路径名。因此，如果您经常更改目录，那么新路径名将导致创建新归档副本，这会造成系统频繁地载入您为元数据指定的卷。

归档集副本参数

`archiver.cmd` 文件中，归档集参数部分以 `params` 指令开头，以 `endparams` 指令结尾。代码示例 3-22 给出了归档集指令的格式。

代码示例 3-22 归档集副本参数的格式

```
params
archive-set-name .copy-number[R] [ -param1 -param2 ...]
.
.
.
endparams
```

表 3-13 归档集副本参数的参数

参数	含义
<i>archive-set-name</i>	您的站点为归档集定义的名称。通常可暗示属于该归档集的文件特征。可以是 <code>allsets</code> 。指定归档集名时，只能使用 26 个英文字母、数字 (0-9) 和下划线字符 (<code>_</code>)，而不得使用其他特殊字符或空格。归档集名的第一个字符必须是字母。
.	句点 (<code>.</code>) 字符。用于分隔 <i>archive-set-name</i> 和 <i>copy-number</i> 。
<i>copy-number</i>	用于定义归档副本份数的整数。可以是 1、2、3 或 4。
R	指定所定义的参数是用于定义此归档集要重新归档的副本数。例如，可以在 <i>-param1</i> 参数中，通过使用 R 和指定 VSN 将重新归档的副本定向到特定的卷。
<i>-param1</i>	一个或多个参数。以下几小节介绍 <code>params</code> 和 <code>endparams</code> 指令之间可以指定的参数。
<i>-param2</i>	

可以按照这里介绍的方法通过编辑 `archiver.cmd` 文件，或者通过使用 File System Manager 软件来指定归档集副本参数。有关更多信息，请参见 File System Manager 联机帮助。

伪归档集 `allsets` 提供了一种为所有归档集设置默认归档集指令的方法。所有 `allsets` 指令必须位于实际归档集副本指令的前面，其原因是为单个归档集副本设置的参数可以取代由 `allsets` 指令设置的参数。有关 `allsets` 归档集的更多信息，请参见 `archiver.cmd(4)` 手册页。

以下各小节介绍除磁盘归档参数之外的所有其他归档集处理参数。有关磁盘归档的信息，请参见第 83 页的“关于磁盘归档”。

控制归档文件的大小：-archmax

`-archmax` 指令用于设置归档集文件的最大大小。此指令的格式如下：

```
-archmax target-size
```

这条指令与 `archmax` 全局指令非常相似。有关该指令以及 `target-size` 的输入值的信息，请参见第 47 页的“`archmax` 指令：控制归档文件的大小”。

设置归档程序缓冲区大小：-bufsize

默认情况下，需要归档的文件先存储在缓冲区的内存中，然后再写入归档介质。可以使用 `-bufsize` 指令指定非默认的缓冲区大小。这些操作可以改善系统性能。您可以尝试不同的 `buffer-size` 值以确定最适合的缓冲区大小。

此参数的格式如下：

```
-bufsize=buffer-size
```

其中的 `buffer-size`，用于指定一个 2 至 32 的数字。默认值为 4。这个值乘以该介质类型的 `dev_blksize` 值，计算结果即为所使用的缓冲区大小。`dev_blksize` 的值是在 `defaults.conf` 文件中指定的。有关此文件的更多信息，请参见 `defaults.conf(4)` 手册页。

例如，可以按以下方式在 `archiver.cmd` 文件的命令行中指定该参数：

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

与该指令等效的全局指令是 `bufsize=media buffer-size`。有关此指令的更多信息，请参见第 48 页的“`bufsize` 指令：设置归档程序缓冲区大小”。

指定归档请求的驱动器数：

`-drivemax`、`-drivemin` 和 `-drives`

默认情况下，归档程序只使用一个介质驱动器对一个归档集中的文件进行归档。如果归档集中文件众多或有比较大的文件，则使用多个驱动器可以帮助您更有效地归档。另外，如果自动化库中驱动器的运行速度不同，使用多个驱动器可以平衡这些差异，进而提高归档效率。

驱动器指令的格式如下：

```
-drivemax max-size
-drivemin min-size
-drives number
```

表 3-14 `-drivemax`、`-drivemin` 和 `-drives` 指令的参数

参数	含义
<i>max-size</i>	一个驱动器中可归档的最大数据量。
<i>min-size</i>	一个驱动器中可归档的最小数据量。其默认值是 <code>-archmax target-size</code> 的值（如果已指定）或是该介质类型的默认值。 如果指定了 <code>-drivemin min-size</code> 指令，则 Sun StorageTek SAM 软件只在数据量足够大时才使用多个驱动器。作为一个指导原则，可以将 <i>min-size</i> 设置得足够大，以使传输时间远远大于卡盒更改时间（载入、定位、卸载）。
<i>number</i>	用于对此归档集进行归档的驱动器数量。默认设置为 1。

系统会根据所指定的参数来检验归档请求，具体如下：

- 如果归档请求小于 *min-size* 的值，则只使用一个驱动器来写入归档请求。
- 如果归档请求大于 *min-size* 的值，则根据 *min-size* 检验归档请求，并预定适当数量的驱动器，但最多不超过指定的最大驱动器数。
- 如果 *min-size* 的值为 0，则尝试将归档请求分开归档在所指定的所有驱动器上。

当使用 `-drives` 参数时，仅在归档的数据量大于 *min-size* 的值时，才使用多个驱动器。并行使用的驱动器数量是以下两个值中的较小者：

- 归档请求的大小除以 *min-size* 的值
- 由 `-drives` 参数指定的驱动器数量

如果您希望将一个归档请求分割至各个驱动器，但是又不希望将较小的归档请求也分割在全部驱动器上，则可以使用 `-drivemin` 和 `-drives` 参数。这适用于大型文件的操作。

要设置这些参数，您需要考虑文件的创建速度、驱动器数目、载入和卸载驱动器所需的时间以及驱动器的传输速率。

例如，假设您将一个名为 `bigfiles` 的归档集分割至五个驱动器上。根据归档集的大小，表 3-15 中显示了可能的分割方法。

表 3-15 归档集分割示例

归档集大小	驱动器数
< 20 GB	1
> 20 GB 至 < 30 GB	2
> 30 GB 至 < 40 GB	3
> 40 GB 至 < 50 GB	4
> 50 GB	5

代码示例 3-23 显示了 `archiver.cmd` 文件中用于将归档请求分割至多个驱动器的指令行。

代码示例 3-23 用于将归档请求分割至多个驱动器的指令

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

此外，您可以在 `archiver.cmd` 文件中指定以下行：

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

当归档集 `huge_files.2` 中的文件总大小等于或大于介质的 `drivemin` 的两倍时，使用两个驱动器来归档文件。

最大化卷上的空间：-fillvsns

默认情况下，归档程序在写入归档副本时，将从指定给归档集的所有卷中进行选择，进而使用一个空间足以容纳所有文件的卷。这将导致卷不会被完全占用。如果指定了 `-fillvsns`，归档程序则将归档请求拆分为一个较小的组。

指定归档缓冲区锁定：-lock

默认情况下，需要归档的文件先存储在缓冲区的内存中，然后再写入归档介质。如果已启用直接 I/O，则可以使用 `-lock` 参数锁定此缓冲区。此操作可以改善系统性能。

此参数的格式如下：

```
-lock
```

`-lock` 参数指明归档程序在创建归档副本时应当使用锁定的缓冲区。如果指定 `-lock`，归档程序将在 `sam-arcopy(1M)` 操作期间，在内存中的归档缓冲区上设置文件锁定。这样可避免缓冲区进行分页，进而改善系统的性能。

仅在配有大量内存的大型系统上，才有必要指定 `-lock` 参数。如果内存不足，则可能会导致内存用尽。

只有已为需要归档的文件启用直接 I/O 时，`-lock` 参数才有效。默认情况下，不会指定 `-lock` 参数，并且文件系统会在所有直接 I/O 缓冲区（包括用于归档的缓冲区）上设置锁定。有关启用直接 I/O 的更多信息，请参见 `setfa(1)` 手册页、`sam_setfa(3)` 库例程手册页，或 `mount_samfs(1M)` 手册页中介绍的 `-O forcedirectio` 选项。

例如，可以按以下方式在 `archiver.cmd` 文件的命令行中指定该参数：

```
yourset.3 -lock
```

另外，还可通过指定 `bufsize=media buffer-size [lock]` 指令的 `lock` 参数，来指定全局性的、与该参数等效的参数。有关此主题的更多信息，请参见第 48 页的“[bufsize 指令：设置归档程序缓冲区大小](#)”。

创建脱机文件的归档副本：-offline_copy

为文件创建了一个归档副本后，此文件即成为可释放的备选文件。如果在创建所有归档副本前，释放文件并使其处于脱机状态，则归档程序会使用此参数来确定创建其他归档副本时所应使用的方法。在选择所要使用的方法时，要考虑 Sun StorageTek SAM 系统的可用驱动器数以及可用的磁盘高速缓存空间。

此参数的格式如下：

```
-offline_copy method
```

其中的 *method*，用于指定表 3-16 中所示的关键字之一。

表 3-16 `-offline_copy` 指令的 *method* 参数的值

<i>method</i> 值	含义
<code>none</code>	在将文件复制到归档卷中之前，根据需要登台每个文件。默认设置。
<code>direct</code>	不使用高速缓存，而是将文件从脱机卷直接复制到归档卷。此方法假定来源卷和目标卷是不同的卷，而且有两个可用驱动器。如果指定了此方法，则需将 <code>stage_n_window</code> 挂载选项的值增大，使其大于 256 KB 的默认值。有关挂载选项的更多信息，请参见 <code>mount_samfs(1M)</code> 手册页。
<code>stageahead</code>	在对一个文件进行归档的同时，登台另一个文件。系统在将某文件写入目标位置的同时，登台下下一个归档文件。
<code>stageall</code>	在归档前，将所有文件登台到磁盘高速缓存。这种方法仅使用一个驱动器，并假定磁盘高速缓存的空间足以容纳所有文件。

指定回收

回收过程可用于收回由过期归档映像所占用的归档卷中的空间。默认情况下，不进行回收。

要进行回收，可在 `archiver.cmd` 文件和 `recycler.cmd` 文件中指定指令。有关 `archiver.cmd` 文件中支持的回收指令的更多信息，请参见第 135 页的“回收”。

联合归档: `-join path`

如果指定了 `-join path` 参数，归档程序将使用联合归档。如果您希望将整个目录归档至一个卷，并且您知道归档文件实际只占用一个卷，则使用联合归档可以满足您的要求。否则，如果希望将目录保存在一起，请使用 `-sort path` 或 `-rsort path` 参数将文件连续地保存在一起。`-rsort` 参数指定逆向排序。

当归档程序将归档文件写入卷时，它会使用用户文件有效地压缩卷。随后，当从同一目录中访问文件时，会出现一定的延迟，因为登台进程需要在卷中移动以读取下一个文件。为了缩短延迟，可以使用 `-join path` 参数将来自同一目录路径的文件连续地归档至一个归档集副本中。联合归档过程将取代空间效率运算法则，从而将来自同一目录的文件归档在一起。

在文件内容不会发生变化，但您始终需要同时访问一组文件的情况下，联合归档非常有用。例如，在医院里，您可以使用联合归档方法来访问与某位患者有关的所有医疗图片。例如：

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

注 --join path 参数可将来自相同目录的数据文件写入至同一个归档文件。如果目录很多，但其中的文件较小且数量不多，归档程序会创建很多较小的归档文件。这些较小的分散归档文件都有其自己的 tar(1) 头文件，这样会降低系统的写入性能。

此外，因为 -join path 参数指定将所有来自同一个目录的文件归档在一个卷上，所以某一组文件有可能找不到合适的可用卷。在这种情况下，如果不为归档集指定更多的卷，这些文件将无法归档。另一种可能是一组要归档的文件太大，以至于永远不能归档在单个卷上。在这种情况下，这些文件永远无法归档。

对大多数应用场合而言，如果不需要对 -join path 操作进行更为严格的限制，则使用 -sort path 或 -join path 参数是最佳选择。

另外，也可以按文件归档时限、大小或路径对归档集副本中的文件进行排序。age 和 size 参数互相排斥，不可同时使用。[代码示例 3-24](#) 显示了如何使用 -sort 参数以及 age 或 size 参数，来对归档集进行排序。

代码示例 3-24 对归档集进行排序的指令

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

第一行强制归档程序按路径名对归档请求进行排序。示例中的第二行强制归档程序根据文件的归档时限，按从旧到新的顺序，对 cardiac.2 归档集副本进行排序。第三行强制归档程序根据文件的大小，按照从小到大的顺序，对 catscans 归档集副本进行排序。如果希望逆向排序，可以用 -rsort 替代 -sort。

控制取消归档

取消归档是指删除文件或目录的归档条目的过程。归档程序根据上一次访问文件的时间来确定是否取消归档。所有经常访问的数据都可以存储在磁盘等快速介质中，而其他所有不经常访问的旧数据可以存储在磁带中。默认情况下，文件永远不会被取消归档。

例如，假设[代码示例 3-25](#)中所示的 archiver.cmd 文件控制一个经常被访问的文件。此文件始终保留在磁盘上，即使保留时间超过 60 天也是如此。只有在该文件未被访问的时间超过 60 天时才会删除副本 1 信息。

如果归档程序因文件未被访问的时间超过 60 天而删除其副本 1 信息，则当从副本 2 登台文件时，需要从磁带中读取文件。该文件恢复联机后，归档程序会在磁盘上为其创建新的副本 1，并且由此时开始计算为期 60 天的访问周期。如果该文件再次被访问，则 Sun StorageTek SAM 归档程序会为其重新生成新的副本 1。

代码示例 3-25 控制取消归档的指令

```
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]
```

假定某位患者在医院里进行为期四周的治疗。在此期间，该患者的所有文件位于快速介质中（副本 1 = mo）。四周后，该患者出院。在该患者出院后的 60 天内，如果该患者的数据从未被访问过，则会取消归档 inode 中的副本 1 条目，而只保留副本 2 和副本 3 条目。此时，用户可以回收卷以便为新患者腾出空间，从而避免增加磁盘库。如果该患者六个月后到医院做后续护理，则首先从磁带（副本 2）访问数据。现在，归档程序会在磁盘中自动创建一个新的副本 1，以确保在后续护理期间（可能持续数天或数周）可以从快速介质中访问数据。

控制归档文件的写入方式：-tapenonstop

默认情况下，归档程序会在归档文件之间写入一个磁带标记（文件结束 (end of file, EOF) 标签）和另外两个磁带标记。当启动下一个归档文件时，驱动程序会返回到第一个磁带标记后面的位置，因而会造成性能降低。-tapenonstop 参数可以指示归档程序只写入初始的磁带标记。这样，驱动程序只需返回到上一个磁带标记（而不是第一个磁带标记）后面的位置，因而提高了性能。另外，如果指定 -tapenonstop 参数，归档程序将在复制操作的最后输入归档信息。

有关 -tapenonstop 参数的更多信息，请参见 archiver.cmd(4) 手册页。

保留卷：-reserve

默认情况下，归档程序会将归档集副本写入由 `archiver.cmd` 文件中卷关联部分所述的正则表达式所指定的卷。但是，有时可能需要归档集卷只包含来自同一归档集的文件。您可以通过保留卷来满足这一数据存储要求。

`-reserve` 参数可为归档集保留卷。在设置 `-reserve` 参数，并将一个卷指定给某个归档集副本后，该卷的标识符不会指定给其他任何归档集副本，即使某个正则表达式与之相匹配。

注 - 站点使用保留卷时，可能导致频繁的卡盒载入和卸载操作。

选择了供某个归档集使用的卷后，系统会为该卷指定一个保留名称，此保留名称是联结归档集和卷的唯一标识符。

注 - `-reserve` 参数用于保留专供一个归档集使用的卷。如果目录很多，但其中的文件较小且数量不多，则会造成很多较小的归档文件写入至每一个保留卷。这些较小的分散归档文件都有其自己的 `tar(1)` 头文件，会降低系统的性能。

`-reserve` 参数的格式如下所示：

```
-reserve keyword
```

keyword 的值取决于您使用的格式，具体如下：

- **归档集格式** - 如表 3-17 所示，`set` 关键字用于指定保留名称中的归档集部分。

表 3-17 归档集格式示例

指令和关键字	保留名称示例
<code>-reserve set</code>	<code>users.1//</code>
	<code>Data.1//</code>

例如，[代码示例 3-26](#) 的 `archiver.cmd` 文件片段中显示，以归档集名 `allsets` 开头的行将按归档集为所有归档集保留卷。

代码示例 3-26 按归档集保留卷

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

- **属主格式** - `dir`、`user` 和 `group` 关键字。这几个关键字相互排斥，用于指定保留名称中的属主部分。`dir` 关键字使用紧跟在归档集定义的路径规范后面的目录路径部分。[表 3-18](#) 给出了相关示例。

表 3-18 属主集格式示例

指令和关键字	保留名称示例
<code>-reserve dir</code>	<code>proj.1/p105/</code> <code>proj.1/p104/</code>
<code>-reserve user</code>	<code>users.1/user5/</code> <code>users.1/user4/</code>
<code>-reserve group</code>	<code>data.1/engineering/</code>

- **文件系统格式** - `fs` 关键字用于指定保留名称中的文件系统部分。[表 3-19](#) 给出了相关示例。

表 3-19 文件系统格式示例

指令和关键字	保留名称示例
<code>-reserve fs</code>	<code>proj.1/p103/samfs1</code> <code>proj.1/p104/samfs1</code>

在 `archiver.cmd` 文件中，您可为一种、两种或全部三种格式指定 `-reserve` 参数。在归档集参数定义中，这三种格式可以组合使用。

例如，在[代码示例 3-27](#) 所示的 `archiver.cmd` 文件片段中，以 `arset.1` 开头的行根据归档集、组和文件系统创建了一个保留名称。

代码示例 3-27 指定了保留卷的 `archiver.cmd` 文件

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

保留卷的信息存储在库目录中。库目录中的行列出了介质类型、VSN、保留信息和保留日期及时间。保留信息包括归档集、路径名和文件系统三个部分，它们之间由双斜杠 (/ /) 分隔。

注 – 这些双斜杠并不表示路径名；它们仅仅是用于显示保留名称的三个组成部分的分隔符。

如**代码示例 3-28** 所示，在库目录中，描述保留卷的行以字符 #R 开头。

代码示例 3-28 显示保留卷的库目录

```
6 00071 00071 lt 0xe8fe 12 9971464 1352412 0x6a000000 131072 0x
# -il-o-b----- 00-5-24 13:50:02 69-12-31 18:00:00 01-7-13 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----- 07.05.00 15:30:29 69-12-31 18:00:00 13.04.01 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----- 00-6-6 16:03:05 69-12-31 18:00:00 01-7-12 11:02:05
#R lt SLOT22 arset0.2// 2001/03/02 12:11:25
```

注 – 为符合页宽，**代码示例 3-28** 中的某些行已作了删减。

一个或多个保留信息字段可以保留空白，这视 archiver.cmd 文件中定义的选项而定。所列的日期和时间是指创建保留信息的时间。保留行附加到每一个保留卷中（即在归档期间保留用于某个归档集的卷）的文件。

归档程序将卷保留信息记录在库目录文件中。重新标记某个卷后，由于卷中的归档数据实际上已被清除，因此归档程序会自动取消保留该卷。

此外，您还可以使用 reserve(1M) 和 unreserve(1M) 命令分别保留及取消保留卷。有关这些命令的更多信息，请参见 reserve(1M) 和 unreserve(1M) 手册页。

可以使用 samu(1M) 实用程序的 v 显示屏幕来显示保留信息，也可使用 archiver(1M) 或 dump_cat(1M) 命令以**代码示例 3-29** 中所示的格式之一来进行显示。

代码示例 3-29 用于显示保留信息的命令

```
archiver -lv
dump_cat -v catalog-name
```

第 101 页的“**示例 4：将用户文件和数据文件归档至光盘介质**”显示了使用保留卷的完整归档示例。

设置归档优先级: -priority

Sun StorageTek SAM 文件系统为文件的归档操作提供了一个可配置的优先级系统。每一个文件均指定有优先级。文件的优先级是通过文件的属性以及优先级乘数（可在 `archiver.cmd` 文件中为每一个归档集进行设置）计算出来的。文件属性包括联机/脱机、归档时限、创建副本的数量和大小。

默认情况下，归档程序不会对归档请求中的文件进行排序，并且所有属性乘数均为零。这将使归档程序按先发现先归档的顺序对文件进行归档。可以通过设置优先级和排序方法，来控制文件的归档顺序。以下是设置优先级的示例：

- 选择 `priority` 排序方法，以按优先级的顺序对归档请求中的归档文件进行归档。
- 更改 `archive_loaded` 优先级，以减少介质载入次数。
- 更改 `offline` 优先级，以使联机文件的归档时间早于脱机文件。
- 更改 `copy#` 优先级，以按副本顺序创建归档副本。

表 3-20 中列出了归档优先级。

表 3-20 归档优先级

归档优先级	定义
<code>-priority age value</code>	归档时限属性乘数
<code>-priority archive_immediate value</code>	立即归档属性乘数
<code>-priority archive_overflow value</code>	多个归档卷属性乘数
<code>-priority archive_loaded value</code>	已载入归档卷属性乘数
<code>-priority copies value</code>	创建副本属性乘数
<code>-priority copy1 value</code>	副本 1 属性乘数
<code>-priority copy2 value</code>	副本 2 属性乘数
<code>-priority copy3 value</code>	副本 3 属性乘数
<code>-priority copy4 value</code>	副本 4 属性乘数
<code>-priority offline value</code>	文件脱机属性乘数
<code>-priority queuwait value</code>	队列等待属性乘数
<code>-priority rearchive value</code>	重新归档属性乘数
<code>-priority reqrelease value</code>	请求释放属性乘数
<code>-priority size value</code>	文件大小属性乘数
<code>-priority stage_loaded value</code>	已载入登台卷属性乘数
<code>-priority stage_overflow value</code>	多个登台卷属性乘数

其中的 *value*，用于指定一个以下范围之内的浮点数：

$$-3.400000000E+38 \leq \textit{value} \leq 3.402823466E+38$$

有关优先级的更多信息，请参见 `archiver(1M)` 和 `archiver.cmd(4)` 手册页。

预定归档：-startage、-startcount 和 -startsize

归档程序在扫描文件系统时，将识别要归档的文件。它将那些被识别为归档对象的文件放置在名为一个归档请求的列表中。在文件系统扫描结束后，系统会预定对归档请求中的文件进行归档。-startage、-startcount 和 -startsize 归档集参数控制归档的工作负载，并确保及时地对文件进行归档。表 3-21 给出了这些参数的格式。

表 3-21 -startage、-startcount 和 -startsize 指令的格式

指令	含义
-startage <i>time</i>	自扫描过程中标记第一个文件并将其列入归档请求，至开始归档所经历的时间。其中的 <i>time</i> ，用于按第 65 页的“设置归档时限”中所述的格式指定时间。如果此参数未设置，可使用 <code>interval</code> 指令对其进行设置。
-startcount <i>count</i>	归档请求中可包含的文件数。当归档请求中文件数量达到 <i>count</i> 的值时，开始归档。默认情况下，不设置 <i>count</i> 。
-startsize <i>size</i>	归档请求中要归档的所有文件的最小总大小（以字节为单位）。随着归档量逐渐增加，当文件的总大小达到 <i>size</i> 的值时，即开始归档。默认情况下，不设置 <i>size</i> 。

`examine=method` 指令和 `interval=time` 指令，可与 -startage、-startcount 和 -startsize 指令交互。-startage、-startcount 和 -startsize 指令可对归档时效和已完成的归档工作进行优化均衡。这些值可取代 `examine=method` 规范（如果已指定）。有关 `examine` 指令的更多信息，请参见第 49 页的“`examine` 指令：控制归档扫描”。有关 `interval` 指令的更多信息，请参见第 50 页的“`interval` 指令：指定归档时间间隔”。

可以在 `archiver.cmd` 文件中为每个归档副本，指定 `-startage`、`-startcount` 和 `-startsize` 指令。如果指定了这些指令中的多个指令，则在满足第一个条件时，即开始归档操作。如果没有指定这些指令中的任何指令，则根据 `examine=method` 指令预定归档请求，具体如下：

- 如果 `examine=noscan`，则默认值如下：`startage` 10 分钟、`startcount` 10,000 以及 `startsize` 10 GB。当第一个文件进入归档请求后，归档程序将根据 `interval=time` 指令的规范来预定归档请求。这属于连续归档法。默认情况下，`examine=noscan`。
- 如果 `examine=scan|scaninodes|scandirs`，则归档程序在文件系统扫描完成之后，开始预定归档请求进行归档。

`archiver.cmd(4)` 手册页中给出了如何使用这些指令的示例。

VSN 关联指令

`archiver.cmd` 文件的 VSN 关联部分用于为归档集指定卷。此部分以 `vsns` 指令开始，以 `endvsns` 指令结束。

还可以使用 File System Manager 软件配置 VSN 关联。有关更多信息，请参见 File System Manager 联机帮助。

可以通过以下格式的指令将一组卷指定给归档集：

```
archive-set-name .copy-num media-type vsn-expr ... [ -pool vsn-pool-name ... ]
```

表 3-22 VSN 关联指令的参数

参数	含义
<code>archive-set-name</code>	您的站点为归档集定义的名称。它必须是归档集分配指令中的第一个字段。归档集名通常暗示属于该归档集的文件特征。指定归档集名时，只能使用 26 个英文字母、数字 (0-9) 和下划线字符 (_)，而不得使用其他特殊字符或空格。归档集名的第一个字符必须是字母。
<code>copy-num</code>	一个数字，后跟一个或多个用于指定该副本归档特征的参数。归档副本指令以数字开头。该数字 (1、2、3 或 4) 是副本份数。
<code>media-type</code>	介质类型。有关有效介质类型的列表，请参见 <code>mcf(4)</code> 手册页。
<code>vsn-expr</code>	正则表达式。请参见 <code>regex(5)</code> 手册页。
<code>-pool vsn-pool-name</code>	已命名的 VSN 组。

一个关联至少需要三个字段：*archive-set-name*、*copy-num* 以及 *media-type*，并且至少还需要一个卷。*archive-set-name* 和 *copy-num* 的值通过一个句点 (.) 相连。

注 – 如果将 Sun StorageTek SAM 环境配置为按归档集进行回收，则不要将 VSN 指定给多个归档集。

代码示例 3-30 显示了两行 VSN 规范。

代码示例 3-30 VSN 规范（多行）

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

代码示例 3-31 显示了一个 VSN 规范，它使用反斜杠字符 (\) 将上一行续接至下一行。

代码示例 3-31 带有后续行的 VSN 规范

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

以下几个示例使用正则表达式以不同的方法指定了相同的 VSN。

代码示例 3-32 使用以简化符号表示的正则表达式指定 VSN。

代码示例 3-32 使用以简化符号表示的 VSN 规范

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

当归档程序需要选择卷供归档集使用时，它会检查所有自动化库和手动挂载的驱动器中属于选定介质类型的每一个卷，以确定该卷是否符合任何 VSN 表达式的要求。归档程序将选择第一个符合表达式要求且包含足够空间可供归档复制操作使用的卷。例如：

- 以下指令指定，使用名称为 `optic20` 至 `optic39` 的 20 个卷中的任意卷，将属于归档集 `ex_set` 的第 1 个副本中的文件复制到介质类型 `mo`：

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

- 以下指令指定，使用以 `TAPE` 开头的任意卷，将属于归档集 `ex_set` 的第 2 个副本的文件复制到介质类型 `lt`：

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

注 — 在设置 `archiver.cmd` 文件时，请确保将卷指定给用于归档元数据的归档集。每一个文件系统均有一个与其自身名称相同的归档集。有关保存元数据的更多信息，请参见 `samfsdump(1M)` 手册页或《Sun StorageTek Storage Archive Manager 故障排除指南》。

VSN 池指令

`archiver.cmd` 文件中，VSN 池部分以 `vsnpools` 指令开始，以 `endvsnpools` 指令结束或至 `archiver.cmd` 文件的结尾处结束。该部分命名了一组卷。

还可使用 File System Manager 软件配置 VSN 池。有关更多信息，请参见 File System Manager 联机帮助。

VSN 池是一个已命名的卷组。VSN 池非常适用于为某个归档集定义可用的卷，因为它可以提供一个有益于为归档集指定和保留卷的缓冲区。可以使用 VSN 池，按照组织内的各部门、组中的用户、数据类型，以及根据其他方便的分组方法定义多个单独的卷组。

如果某个卷被保留用于归档集，则该卷将不能再供它最初所属的 VSN 池使用。因此，已命名的 VSN 池中的卷数量随卷的使用情况而变化。可以按以下格式输入 `archiver(1M)` 命令来查看 VSN 池：

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

定义 VSN 池的语法如下：

```
vsnpool-name media-type vsn-expr
```

表 3-23 VSN 池指令的参数

参数	含义
<i>vsnpool-name</i>	VSN 池。
<i>media-type</i>	由两个字符表示的介质类型。有关有效介质类型的列表，请参见 <i>mcf(4)</i> 手册页。
<i>vsn-expr</i>	正则表达式。可以指定一个或多个 <i>vsn-expr</i> 参数。请参见 <i>regcmp(3G)</i> 手册页。

以下示例使用了四个 VSN 池：*users_pool*、*data_pool*、*proj_pool* 和 *scratch_pool*。暂用池是指当 VSN 关联中的特定卷或另一个 VSN 池消耗殆尽时，系统临时使用的一组卷。如果三个池中某一个池的卷消耗殆尽，则归档程序选择暂用池 VSN。代码示例 3-33 显示了一个使用四个 VSN 池的 *archiver.cmd* 文件。

代码示例 3-33 VSN 池

```
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
```

有关 VSN 关联的更多信息，请参见第 79 页的“VSN 关联指令”。

关于磁盘归档

归档是指将文件从联机磁盘复制到归档介质的过程。对于磁盘归档，归档介质是文件系统中的联机磁盘。

应用磁盘归档功能，不仅可以将一个 Sun StorageTek SAM 文件系统中的文件归档至同一主机中的另一个文件系统，而且还可以将其归档至其他 Sun Solaris 主机中的另一文件系统。如果在两个主机系统上应用磁盘归档功能，则其中一个系统充当客户机，另一个系统充当服务器。其中客户机系统承载源文件，服务器系统作为承载归档副本的目标系统。

归档文件写入至的文件系统可以是任何一种 UNIX 文件系统，但是，如果磁盘归档副本写入至另一台主机，则该主机上至少必须安装了一个与 Sun StorageTek SAM 软件兼容的文件系统。

归档程序对待归档至磁盘卷的文件的方式与对待归档至库中卷的文件相同。仍然可以创建一个、两个、三个或四个归档副本。如果创建多个归档副本，则可以将其中一个归档副本写入至磁盘卷，而将其他归档副本写入至可移除介质卷。另外，如果您通常将文件归档至 Sun StorageTek SAM 文件系统中的磁盘卷，则归档程序将根据该文件系统的 archiver.cmd 文件中的规则对归档文件副本自身进行归档。

以下概述了归档至联机磁盘与归档至可移除介质的相似点和不同点：

- 与写入至磁光盘或磁带的归档副本不同，写入至磁盘的归档副本不会记录在目录中。另外，磁盘卷中的归档文件不会出现在历史记录中。
- 如果您要将文件归档至可移除介质卷，则在挂载文件系统后，无需更改 archiver.cmd 文件中的任何默认值便可开始归档。相反，如果您要将文件归档至磁盘卷，则在挂载文件系统之前必须编辑 archiver.cmd 文件并定义磁盘归档集。
- 磁盘归档不依赖于 mcf 文件中的条目。您要在 archiver.cmd 文件中指定磁盘归档集，并在 /etc/opt/SUNWsamfs/diskvols.conf 中定义磁盘卷。

注 - 如果只将文件归档至可移除介质卷，则不需要 diskvols.conf 配置文件。

diskvols.conf 文件必须在源文件所在的系统中进行创建。根据归档副本写入位置的不同，此文件还可能包含以下信息：

- 如果归档副本写入至同一主机系统中的文件系统，则 diskvols.conf 文件将定义 VSN 及其路径。
- 如果归档副本写入至另一个 Sun Solaris 主机系统，则 diskvols.conf 文件将包含该服务器系统的主机名。在此情况下，该服务器系统中也必须有一个 diskvols.conf 文件，以定义那些有权向该服务器系统写入数据的客户机。如果希望创建这种客户机/服务器关系，在启动第 86 页的“启用磁盘归档”中所述的过程之前，应确保充当服务器的主机上至少安装了一个 Sun StorageTek SAM 文件系统。

注意 – 如果要在具有多个 SAM-QFS 服务器的环境中使用磁盘归档，则在配置回收程序时必须格外小心。每个 SAM-QFS 服务器的 diskvols.conf 文件必须指向一组唯一的磁盘卷资源规范（磁盘归档目标目录）。如果不同的 SAM-QFS 服务器共享其中一个规范，则从一个 SAM-QFS 服务器中运行回收程序时，将会销毁由其他 SAM-QFS 服务器所管理的磁盘归档数据。

配置原则

虽然没有对磁盘归档卷的驻留位置进行任何限制，但是，建议您不要将该磁盘卷放置在原始文件所在的磁盘上。我们还建议您创建多个归档副本，并将它们写入至不同类型的归档介质。例如，您可以将第 1 个副本归档至磁盘卷，将第 2 个副本归档至磁带，而将第 3 个副本归档至磁光盘。

如果要将文件归档至服务器系统上的文件系统，则归档文件自身还会被归档至与服务器连接的库中的可移除介质卡盒。

磁盘归档指令

当归档至联机磁盘时，归档程序可识别用于定义归档集和配置回收过程的 `archiver.cmd` 指令。它将忽略可移除介质卡盒专用的指令。系统将着重识别以下用于磁盘归档集的指令：

- 第 66 页的“归档集副本参数”中，除以下指令之外的所有回收指令：
 - `-fillvsns`
 - `-ovflmin min-size`
 - `-reserve method`
 - `-tapenonstop`
- 第 146 页的“编辑 `archiver.cmd` 文件”中，除以下指令之外的所有指令：
 - `-recycle_dataquantity size`
 - `-recycle_vsncount count`
- `vsns` 和 `endvsns` 指令，以及 `vsnpools` 和 `endvsnpools` 指令。VSN 关联部分支持磁盘卷，并可使用 `dk` 介质类型定义磁盘卷。卷由一个或多个 VSN 表达式关键字表示。还可以指定从哪些 VSN 池中选择磁盘卷，如代码示例 3-34 所示。

代码示例 3-34 `vsns` 和 `vsnpools` 指令示例

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

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

从 4U6 版软件开始，也可以在 Sun StorageTek 5800 系统上执行磁盘归档。Sun StorageTek 5800 是一种具有集成硬件和软件体系结构的联机存储设备，在这种体系结构中，基于磁盘的存储节点排列在对称群集中。在 `vsns` 指令中，将 Sun StorageTek 5800 磁盘归档介质缩写为 `cb`。

- `clients` 和 `endclients` 指令。如果将文件从客户机主机归档至服务器主机，则服务器系统必须具有一个包含客户机系统名称的 `diskvols.conf` 文件。这些指令的格式如代码示例 3-35 中所示。

代码示例 3-35 `clients` 和 `endclients` 指令的格式

```
clients
client-system1
client-system2
...
endclients
```

其中的 *client-system*，用于指定包含源文件的客户机系统的主机名。

- `-recycle_minobs percent` 回收程序指令。此选项用于设置回收程序对磁盘归档进行重新归档过程中的阈值（默认情况下为 50%）。当磁盘上一个已归档的 `tar` 文件中的无效文件所占百分比达到此阈值时，回收程序会将该归档文件中的所有有效文件移到新的 `tar` 文件中。所有有效文件的移动操作完成之后，回收程序会对原始 `tar` 文件进行标记，以表明它将被从磁盘归档中删除。回收可移动介质时，可忽略这个选项。
- 对于磁盘归档回收，将忽略以下回收指令：
 - `recycle_hwm`
 - `recycle_vsncount`

有关磁盘归档指令的更多信息，请参见 `archiver.cmd(4)` 手册页。

注 – 如果在 Sun StorageTek 5800 上使用磁盘卷进行归档，请注意 Sun StorageTek 5800 并不是传统的文件系统，其安全注意事项不同于其他类型的磁盘存储。有关更多信息，请阅读 Sun StorageTek 5800 文档。

▼ 启用磁盘归档

磁盘归档功能可随时启用。本节介绍的过程假定归档的前期准备工作已经就绪，现在只需在环境中添加磁盘归档功能。如果您是在初始安装过程中启用磁盘归档，请参见《Sun StorageTek Storage Archive Manager 安装和升级指南》以了解相关信息。

注 – 在 4U4 之前的软件版本中，要在 `archiver.cmd` 文件中启用磁盘归档功能，必须通过 `params` 部分的 `-disk_archive` 参数。4U4 软件版本已不再使用此参数，因此，为了使归档功能在 4U4 版本及更高版本中正常工作，必须对使用早期软件版本创建的 `archiver.cmd` 文件进行编辑。有关详细信息，请参见 `archiver.cmd(4)` 手册页。

1. 请确保在要写入磁盘归档副本的主机上，至少安装了一个 **Product Name** 或 **Sun StorageTek SAM** 文件系统。
2. 以超级用户的身份登录到要归档的文件所在的主机系统。
3. 按照《Sun StorageTek Storage Archive Manager 安装和升级指南》中的过程，在包含要归档的文件的主机和将要写入归档副本的主机上启用磁盘归档功能。

4. 在包含要归档的文件的宿主上，使用 `samd(1M) config` 命令传播配置文件的更改，然后重新启动系统。

```
# samd config
```

5. 如果要归档至另一台主机的磁盘，请执行以下步骤：
 - a. 以超级用户的身份登录到将要写入归档副本的宿主系统。
 - b. 使用 `samd(1M) config` 命令传播配置文件更改并重新启动目标系统。

```
# samd config
```

6. 如果要归档至 Sun StorageTek 5800 系统，则必须升级 Sun StorageTek 5800 元数据模式配置。请按照《Sun StorageTek 5800 System Administration Guide》中介绍的过程进行操作。

在代码示例 3-36 中，使用 XML 覆盖来定义 Sun StorageTek SAM 使用的元数据。

代码示例 3-36 STK 5800 上的 Sun StorageTek SAM 的元数据模式

```
<?xml version="1.0" encoding="UTF-8"?>

<metadataConfig>
  <schema>
    <namespace name="com">
      <namespace name="sun">
        <namespace name="samfs">
          <field name="archiveId" type="string" indexable="true"/>
        </namespace>
      </namespace>
    </namespace>
  </schema>

  <fsViews>
  </fsViews>

</metadataConfig>
```

磁盘归档示例

以下是一些磁盘归档配置的示例。

示例 1

在本示例中，名为 disk01、disk02 和 disk04 的 VSN 将写入至 pluto，它是初始源文件所在的主机系统。VSN disk03 将写入至服务器系统 mars 中的 VSN。

代码示例 3-37 显示了客户机系统 pluto 上的 diskvols.conf 文件。

代码示例 3-37 pluto 上的 diskvols.conf 文件

```
# 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
```

代码示例 3-38 显示了服务器系统 mars 上的 diskvols.conf 文件。

代码示例 3-38 mars 上的 diskvols.conf 文件

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

代码示例 3-39 显示了 pluto 上的 archiver.cmd 文件片段。

代码示例 3-39 pluto 上的 archiver.cmd 文件

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

示例 2

在本示例中，文件 `/sam1/testdir0/filea` 位于 `arset0.1` 归档集中，归档程序将该文件的内容复制到目标路径 `/sam_arch1` 中。[代码示例 3-40](#) 显示了 `diskvols.conf` 文件。

代码示例 3-40 `diskvols.conf` 文件

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

[代码示例 3-41](#) 显示了 `archiver.cmd` 文件中与磁盘归档有关的指令行：

代码示例 3-41 `archiver.cmd` 文件中与磁盘归档有关的指令

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

[代码示例 3-42](#) 显示了对已归档至磁盘的 `filea` 文件运行 `sls(1)` 命令时的输出。请注意 `copy 1` 行中的以下内容：

- `dk` 是磁盘归档介质的介质类型
- `disk01` 是 VSN
- `f192` 是磁盘归档 `tar(1)` 文件的路径

代码示例 3-42 `sls(1M)` 的输出

```
# 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
```

示例 3

在本示例中，文件 `/sam2/my_proj/fileb` 位于客户机主机 `snickers` 的归档集 `arset0.1` 中，归档程序将该文件的内容复制到服务器主机 `mars` 的目标路径 `/sam_arch1` 中。

代码示例 3-43 显示了 `snickers` 上的 `diskvols.conf` 文件。

代码示例 3-43 `snickers` 上的 `diskvols.conf` 文件

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

代码示例 3-44 显示了 `mars` 上的 `diskvols.conf` 文件。

代码示例 3-44 `mars` 上的 `diskvols.conf` 文件

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

代码示例 3-45 显示了 `archiver.cmd` 文件中与本示例有关的指令。

代码示例 3-45 `archiver.cmd` 文件中与磁盘归档有关的指令

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

设计归档操作

归档程序使用 `archiver.cmd` 文件进行自动存储管理操作。在写入此文件之前，先回顾一些可以改善 Sun StorageTek SAM 文件系统和归档程序性能以及可帮助确保您以最安全的方式存储数据的通用原则，这是很有益的。

- 保存归档日志。归档日志可为数据恢复提供非常重要的信息，即使在 Sun StorageTek SAM 软件无法使用时也是如此。请将这些日志保存在安全的地方，以防因发生灾难性故障而造成 Sun StorageTek SAM 软件无法使用。
- 使用正则表达式指定卷。允许系统将文件放置在许多不同的卷上。通过正则表达式指定的卷范围可以使系统连续不断地运行。如果您为归档集副本指定特定的卷名，则会造成数据很快充满卷。因此，您不得不频繁地更换介质，从而导致不应有的工作流程问题出现。
- 根据文件的创建和修改频率以及是否需要保存所有修改副本，来设置您的归档时间间隔。请注意，归档时间间隔是指对文件系统执行扫描操作的时间间隔。如果将归档时间间隔设置得太短，则会使归档程序几乎不间断地执行扫描。
- 考虑您要使用的文件系统数量。与单个 Sun StorageTek SAM 文件系统相比，多个 Sun StorageTek SAM 文件系统通常可以提高归档程序的性能。多个文件系统的扫描时间要比单个文件系统少得多。
- 使用目录结构组织 Sun StorageTek SAM 文件系统文件。为了使性能最优，放入一个目录的文件不要多于 10,000 个。
- 请始终制作两个文件副本，并将它们存储至不同的卷上。如果将数据存储至同一种介质类型，则在介质出现物理问题时，您会面临数据丢失的风险。
- 确保定期使用 `samfsdump(1M)` 备份您的元数据（目录结构、文件名，等等）。元数据存储在与文件系统同名的归档集中。在出现故障时，您可以使用此类信息来恢复文件系统。如果您不想备份元数据，可以通过将此归档集指定给不存在的 VSN 来防止归档元数据。有关保存元数据的更多信息，请参见《Sun StorageTek Storage Archive Manager 故障排除指南》或《Sun StorageTek Storage Archive Manager 安装和升级指南》。

预备队列

归档程序和登台程序进程均可请求载入及卸载介质。如果请求的数量超过可用于介质载入的驱动器数量，则多余的请求会发送至预备队列。

预备队列中的归档和登台请求是指那些无法立即满足的请求。默认情况下，系统按先进先出 (First In First Out, FIFO) 的顺序执行预备请求。

您可以为各个预备请求指定不同的优先级。可以通过在预备命令文件中输入指令来覆盖 FIFO 默认值，此预备命令文件的位置为 `/etc/opt/SUNWsamfs/preview.cmd`。有关此文件以及设置归档和登台操作优先级的更多信息，请参见第 128 页的“排列预备请求的优先顺序”。

归档程序示例

本节提供了一些在实际环境中使用的归档过程的示例。

示例 1: 无 archiver.cmd 文件

本示例介绍了在 Sun StorageTek SAM 环境中未使用 archiver.cmd 文件时, 归档程序的操作。该环境包括一个文件系统、一个配有两个驱动器的光盘自动化库和六个卡盒。

[代码示例 3-46](#) 显示了 archiver(1M) -lv 命令的输出。它表明归档程序选择的默认介质类型为 mo。只有 mo 介质可用。

代码示例 3-46 archiver(1M) -lv 输出: 显示归档介质

```
# 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
```

[代码示例 3-47](#) 的输出表明归档程序使用了两个驱动器。它列出了 12 个卷及其存储容量和可用空间。

注 - archiver(1M) -lv 命令只能显示具有可用空间的 VSN。

代码示例 3-47 archiver(1M) -lv 输出: 显示可用的 VSN

```
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-----
```

代码示例 3-48 显示了归档集 `samfs` 同时包含元数据和数据文件。当文件的归档时间达到默认的四分钟（240 秒）时，归档程序将开始创建文件的副本。

代码示例 3-48 `archiver(1M) -lv` 输出：显示归档文件选择

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

代码示例 3-49 的输出表明归档集中的文件按指定的顺序归档至卷中。

代码示例 3-49 `archiver(1M) -lv` 输出：显示归档集和卷

```
Archive sets:
allsets
samfs.1
  media: mo (by default)
  Volumes:
    optic00
    optic01
    optic02
    optic03
    optic10
    optic12
    optic13
    optic20
    optic21
    optic22
    optic23
Total space available: 8.1G
```

示例 2：数据文件和元数据分开归档

本示例说明如何将数据文件和元数据划分至两个不同的归档集。除了光盘自动化库之外，环境中还包括一个手动挂载的 DLT 磁带机。较大的文件归档至磁带，而较小的文件归档至光盘卡盒。

代码示例 3-50 显示了 archiver.cmd 文件的内容。

代码示例 3-50 archiver(1M) -lv 输出：显示 archiver.cmd 文件

```
# archiver 例 v -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[0-2]          # Metadata to optic00 - optic02
14: all.1 mo .*0[3-9] .*[1-2][0-9] # All others for files
15: big.1 lt .*
16: endvsns
```

代码示例 3-51 显示了要使用的介质和驱动器。

代码示例 3-51 archiver(1M) -lv 输出：显示介质和驱动器

```
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-----
```

注 - archiver(1M) -lv 命令只能显示具有可用空间的 VSN。

代码示例 3-52 显示了文件系统的组织结构。大于 512000 字节 (500 KB) 的文件在四分钟后归档；其他所有文件在 30 秒后归档。

代码示例 3-52 archiver(1M) -lv 输出：显示文件系统组织结构

```
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
```

代码示例 3-53 显示了归档集在可移除介质中的分割情况。

代码示例 3-53 archiver(1M) -lv 输出：显示归档集和可移除介质

```
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
```

示例 3：将用户文件和数据文件归档至不同的介质

在本示例中，用户文件和项目数据文件归档至不同的介质。data 目录中的文件按大小分别存储至光盘介质和磁带介质。指定给组 ID pict 的文件将被指定给另一个卷组。归档程序不对目录 tmp 和 users/bob 中的文件进行归档。归档程序每隔 15 分钟进行一次归档，并且保存归档记录。

在本示例中，[代码示例 3-54](#) 显示了 archiver(1M) -lv -c 命令的输出。

代码示例 3-54 archiver(1M) -lv -c 命令的输出

```
# 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]$
```

代码示例 3-54 archiver(1M) -lv -c 命令的输出 (续)

```

33: joe.1 mo optic01$
34: pict.1 mo optic02$
35: pict.2 mo optic03$
36: proj_1.1 mo optic1[0-1]$
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

```

代码示例 3-54 archiver(1M) -lv -c 命令的输出 (续)

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

代码示例 3-54 archiver(1M) -lv -c 命令的输出 (续)

```
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
```

示例 4：将用户文件和数据文件归档至光盘介质

在本示例中，用户文件和项目数据文件归档至光盘介质。

本示例定义了四个 VSN 池；其中三个池分别用于用户、数据和项目，另一个是暂用池。当 `proj_pool` 中的介质用尽后，将使用 `scratch_pool` 来保留卷。本示例介绍如何依据归档集、属主和文件系统等各个组成部分，来为每一个归档集保留卷。归档程序每隔十分钟进行一次归档，并且保存归档日志。

代码示例 3-55 显示了 `archiver.cmd` 文件及归档程序的输出。

代码示例 3-55 `archiver.cmd` 文件及归档程序的输出

```
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
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
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
users.1
  reserve:/user/
  media: mo
  Volumes:
    optic01
    optic02
    optic03
    optic20
    optic21
    optic22
    optic23
  Total space available: 6.0G
```


释放

释放是指释放程序通过识别已归档的文件并释放这些文件在磁盘高速缓存中的副本，从而使磁盘高速缓存空间可再利用的过程。这可以为其他从归档介质中创建或登台的文件腾出空间。释放程序只释放已归档的文件。磁盘高速缓存中不保留已释放文件的数据。

当高速缓存的占用量达到站点指定的磁盘阈值时，Sun StorageTek SAM 软件会自动调用释放程序进程。或者，可以使用 `release(1)` 命令立即释放文件占用的磁盘空间或为文件设置释放参数。

您还可以指定哪些文件在归档之后立即释放、哪些文件从不释放或哪些文件可以部分释放。

本章介绍释放过程和释放程序的操作。本章包含以下各节：

- 第 106 页的“释放程序进程概述”
- 第 107 页的“释放程序概念”
- 第 108 页的“关于部分释放和部分登台”
- 第 110 页的“关于 `releaser.cmd` 文件”
- 第 117 页的“规划释放程序的操作”
- 第 119 页的“手动运行释放程序”

释放程序进程概述

当文件系统的利用率超过所配置的上限时，文件系统管理软件将调用释放程序，该程序将执行以下操作：

1. 读取 `releaser.cmd` 文件并收集用于控制释放进程的指令。
2. 扫描文件系统并收集每个文件的相关信息。
3. 开始按优先级顺序释放文件。

文件系统可能包含成千上万个文件。由于只需释放几个大文件便有可能使文件系统的利用率降至下限，因此跟踪记录每个文件的释放优先级可能很不经济。但是，释放程序又必须检查每一个文件的优先级，否则就不能释放最恰当的备选文件。释放程序通过确定前 10,000 个备选文件来解决这一问题。如果随后的备选文件的优先级不高于前 10,000 个备选文件的最低优先级，则释放程序会忽略随后的备选文件。

确定前 10,000 个备选文件的优先级之后，释放程序会选择释放具有最高优先级的文件。每释放一个文件后，释放程序就会进行一次检查，确定文件系统的高速缓存利用率是否低于下限。如果是，则释放程序将停止释放文件。如果不是，则释放程序将继续按优先级的顺序释放文件。

如果在释放全部 10,000 个备选文件之后，文件系统的利用率仍高于下限，则释放程序将重新确定 10,000 个新备选文件。

如果找不到任何合适的备选文件，则释放程序会退出。例如，在文件没有归档副本时，就会出现这种情况。如果出现这种情况，则 Sun StorageTek SAM 软件将在退出释放程序的一分钟后再次启动释放程序。

可以使用 `high=percent` 和 `low=percent` 文件系统挂载选项来设置上、下限。有关挂载选项的更多信息，请参见 `mount_samfs(1M)` 手册页。

释放程序概念

本节介绍释放程序进程的基本概念：

- 时限 - 是指从发生指定事件开始到现在所经历的时间。文件的 `inode` 节点可以跟踪记录释放程序所使用的以下时间：

- 驻留更改时间
- 数据修改时间
- 数据访问时间

您可以使用带有 `-D` 选项的 `s1s(1)` 命令来查看这些时间。每一种时间都有对应的时限。例如，如果当前时间是上午 10 点 15 分，则在上午 10 点 10 分所修改的文件的数据修改时限为 5 分钟。有关 `s1s(1)` 命令的更多信息，请参见 `s1s(1)` 手册页。

- 备选文件 - 即符合释放条件的文件。在以下情况下，文件不能成为备选文件：
 - 该文件已脱机。
 - 该文件尚未归档。
 - `archiver.cmd` 命令文件为该文件指定了 `-norelease` 属性，并且尚未为该文件创建完所需的副本份数。
 - 该文件标记为“已损坏”。
 - 该文件不是普通文件，而是目录文件、块文件、特殊字符文件或管道文件。
 - 归档程序正在登台该文件以创建另一副本。登台之后，该文件便成为适合释放的文件。
 - 该文件的时限为负数。这种情况可能发生在未正确设置时钟的网络文件系统 (`network file system, NFS`) 客户机上。
 - 文件被标记为不释放。这可使用 `release(1) -n` 命令来指定。
 - 该文件在过去登台的时间小于最短驻留时间设置。有关详细信息，请参见第 114 页的“[min_residence_age 指令：指定最短驻留时间](#)”。
 - 已通过 `release(1)` 命令的 `-p` 选项将该文件标记为部分释放，并且已部分释放该文件。
 - 该文件太小。释放该文件将不会获得太多的空间。
- 优先级 - 是一个表示备选文件级别的数值，该数值取决于用户提供的应用于该备选文件数值属性的权重。总优先级是以下两类优先级之和：时限优先级和大小优先级。释放程序首先释放具有较大优先级数值的备选文件，然后释放具有较小优先级数值的备选文件。
- 权重 - 是一个数值，用于使优先级的计算倾向于包括您感兴趣的文件属性，并排除不感兴趣的文件属性。例如，如果将文件的大小权重设置为 0，则在计算优先级时，不会考虑文件的大小属性。权重是介于 0.0 和 1.0 之间的浮点值。
- 部分释放 - 通过部分释放，文件的起始部分会保留在磁盘高速缓存中，而文件的其余部分会被释放。部分释放功能对于诸如 `filemgr(1)` 等读取文件的起始部分的实用程序非常有用。

关于部分释放和部分登台

释放和登台是两个互为补充的进程。文件在归档后，就可以从联机磁盘高速缓存中完全释放，站点也可以指定只在磁盘高速缓存中保留文件的起始部分（即存根），而释放文件的其余部分。部分释放文件的功能可以使系统在不首先登台文件的情况下，立即访问文件存根中的数据。

作为系统管理员，您可以在挂载文件系统时，指定保持联机的存根的部分释放的默认大小和最大大小。您可以通过 `mount(1M)` 命令或 `File System Manager` 软件设置这些值。有关更多信息，请参见 `File System Manager` 联机帮助。

`mount(1M)` 命令具有以下选项：

- `-o partial=n` 选项 - 设置保持联机的文件存根的默认大小 (n)。 `-o partial=n` 的设置值必须小于或等于 `-o maxpartial=n` 的设置值。
- `-o maxpartial=n` 选项 - 设置保持联机的文件存根的最大大小 (n)。

您可以通过在 `release(1)` 命令中指定 `-p` 选项或在 `sam_release(3)` 库例程中指定 `p` 选项，来指定文件的默认存根大小。要为不同类型的文件或不同的应用程序指定不同大小的文件存根，用户可以在 `release(1)` 命令中指定 `-s` 选项，或在 `sam_release(3)` 库例程中指定 `s` 选项。`-s` 和 `s` 的值必须小于挂载文件系统时与 `mount(1M)` 命令一起使用的 `-o maxpartial` 值。

注 - 即使部分释放了文件，该文件仍会占用大小等于一个 DAU 的磁盘空间。例如，如果将部分释放文件存根设置为 16K，而 DAU 大小为 256K，则即使部分释放了文件，该文件实际占用的磁盘空间仍为 256K。

系统管理员可以使用另一个挂载选项，即 `-o partial_stage=n`，来确定在登台文件的剩余部分之前，应读取多少部分释放存根。当读取的存根量大于 `-o partial_stage=n` 大小规范后，即开始登台文件。

默认情况下，系统将 `-o partial_stage=n` 选项设置为等于部分释放存根的大小。更改该值对文件登台的影响如下：

- 如果将 `-o partial_stage=n` 选项设置为等于部分释放存根的大小，则系统的默认操作是直到应用程序到达部分释放存根的末尾时才允许登台文件。等待到达存根的末尾会推迟应用程序对文件剩余部分的访问。
- 如果将 `-o partial_stage=n` 选项设置为小于部分释放存根的值，则当应用程序超过 `-o partial_stage=n` 选项设置的阈值后，系统将登台文件。这可以加快应用程序对文件数据剩余部分的访问。

例如，假定您设置了以下选项：

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

filemgr(1) 程序读取文件的前 8 KB。此时，系统不会登台文件。有一个视频点播程序读取同一个文件，并且当它读完文件的前 16 KB 时，系统开始登台文件。在挂载并定位归档磁带后，该应用程序会继续读取 2 GB 的磁盘数据。当视频点播程序读完 2 GB 的文件数据后，它会在登台活动之后立即进行读取。由于在该应用程序读取部分文件数据时已经挂载并定位了磁带，因此它不必等待。

有多个命令行选项可影响文件是否可以标记为部分释放。某些选项可由系统管理员启用，而另一些选项可由个别用户启用。以下几节介绍了不同类型的用户可以设置的释放特征。

系统管理员选项概述

作为系统管理员，您可以在挂载文件系统时，更改部分释放的最大值和默认值。表 4-1 列出了可影响部分释放的 mount(1M) 选项。有关 mount(1) 命令的更多信息，请参见 mount_samfs(1M) 手册页。

表 4-1 影响部分释放的挂载选项

mount(1M) 选项	作用
-o maxpartial= <i>n</i>	<p>指定在文件标记为部分释放时，可以在联机磁盘高速缓存中保留的最大空间 (KB)。最大值为 2,097,152 KB，即 2 GB。最小值为 0，即不允许部分释放任何文件。</p> <p>如果指定 -o maxpartial=0，则会禁用部分释放功能。已释放的文件会被完全释放，并且磁盘高速缓存中不会保留文件的任何部分。一旦挂载文件系统，用户便不能再覆盖此选项指定的值。</p> <p>默认情况下，<i>n</i> 参数设置为 16。此设置使得用户可以将文件标记为部分释放，且这个文件最多可在磁盘上保留 16 KB 的数据。</p>
-o partial= <i>n</i>	<p>当用户使用 release(1) 命令的 -p 选项将文件标记为部分释放时，该选项用于设置要在磁盘高速缓存中保留的默认空间 (KB)。<i>n</i> 参数的值必须至少为 8，但可以等于为 -o maxpartial=<i>n</i> 选项指定的值。</p> <p>由于某些应用程序不必访问整个文件便可完成其工作，因此该选项可用于确保应用程序能够从文件的起始部分获得所需的信息。同时，使用此选项还可阻止系统登台不必要的文件。</p> <p>默认情况下，<i>n</i> 为 16。</p> <p>注意：即使文件从磁盘中进行了部分释放，该文件仍会占用大小等于一个 DAU 的磁盘空间。</p>
-o partial_stage= <i>n</i>	<p>指定在访问部分释放的文件时，从归档介质登台整个文件之前应读取 <i>n</i> 字节的文件数据。此选项的设置值通常小于 -o partial 的设置值。其中的 <i>n</i>，用于指定一个介于 0 到 -o maxpartial 规范值之间的整数。默认情况下，设置为 16 或是为 -o partial 选项指定的任何值。</p>
-o stage_n_window= <i>n</i>	<p>将一次可以登台的数据量指定为 <i>n</i>。其中的 <i>n</i>，用于指定一个介于 64 到 2,048,000 之间的整数。默认值为 256 KB。此选项仅适用于已设置 stage -n 属性的文件。</p>

用户选项概述

作为系统管理员，您可以设置文件在释放后，可保留在磁盘高速缓存中的文件存根大小的最大值和默认值。您还可以确定是否为特定的文件系统启用部分释放功能。

不过，通过使用 `release(1)` 命令和 `sam_release(3)` 库例程，用户可以设置其他释放属性以及指定要标记为部分释放的文件。表 4-2 中列出了可以指定部分释放属性的命令和库选项。有关 `release(1)` 命令的更多信息，请参见 `release(1)` 手册页。有关 `sam_release(3)` 库例程的更多信息，请参见 `sam_release(3)` 手册页。

表 4-2 用户释放选项

选项	作用
<code>release(1)</code> 命令和 <code>-p</code> 选项 或 <code>sam_release(3)</code> 库例程和 <code>p</code> 选项	<code>-p</code> 和 <code>p</code> 选项用于将指定的文件标记为部分释放。如果使用这些选项，则文件在释放后可以保留在联机磁盘高速缓存中的数据量，取决于在挂载该文件所在的文件系统时为 <code>-o partial=n</code> 选项设置的值。这些选项不能用于指定保持联机的字节数。
<code>release(1)</code> 命令和 <code>-s partial_size</code> 选项 或 <code>sam_release(3)</code> 库例程和 <code>s</code> 选项	<code>-s</code> 和 <code>s</code> 选项用于将指定的文件标记为部分释放，并指定要在联机磁盘高速缓存中保留的文件数据量。 <code>-s</code> 或 <code>s</code> 选项的参数用于指定要保持联机的数据量 (KB)。用户指定的保持联机的文件数据量不能大于挂载文件系统时指定的 <code>-o maxpartial=n</code> 值。如果用户指定的值大于文件系统的值，则系统会使用文件系统的值，而忽略用户指定的规范。

关于 `releaser.cmd` 文件

`/etc/opt/SUNWsamfs/releaser.cmd` 文件由指定站点特定释放操作的指令行组成。`releaser.cmd` 文件可以包含用于设置释放优先级的指令、用于指定日志文件的指令以及用于执行其他操作的指令。

以下几个小节介绍了 `releaser.cmd` 指令：

- 第 111 页的“指定与时限和大小相关的释放优先级指令”
- 第 113 页的“`fs` 指令：指定用于单个文件系统的指令”
- 第 114 页的“`no_release` 和 `display_all_candidates` 指令：指定调试”
- 第 114 页的“`min_residence_age` 指令：指定最短驻留时间”
- 第 115 页的“`logfile` 指令：指定日志文件”
- 第 116 页的“`rearch_no_release` 指令：阻止释放重新归档的文件”
- 第 116 页的“`list_size` 指令：调整释放程序备选文件列表的大小”
- 第 117 页的“指定归档集中所有文件的释放属性”

有关这些指令的更多信息，请参见 `releaser.cmd(4)` 手册页。

可使用 **File System Manager** 软件配置某些全局释放指令。有关更多信息，请参见 **File System Manager** 联机帮助。

指定与时限和大小相关的释放优先级指令

释放程序根据 `releaser.cmd` 文件中定义的指令所确定的优先级顺序来释放文件系统 中的文件。在释放文件时，它既考虑文件的时限，又考虑文件的大小。默认情况下，站 点首先释放最大且最旧的文件，而将最小且最新的文件保留在磁盘中。以下部分介绍了 释放程序在确定文件系统中文件的释放优先级时如何考虑文件的时限和大小。

有关这些释放指令的更多信息，请参见 `releaser.cmd(4)` 手册页。

文件时限

释放程序在确定与时限相关的文件释放优先级要素时，将考虑以下可能的时限：

- 从最后一次访问文件到现在的时限
- 从最后一次修改文件到现在的时限
- 从文件在磁盘高速缓存中的驻留状态发生更改到现在的时限

默认情况下，文件时限是指以下三个文件时限中最小的一个。

在某些情况下，可能会优先考虑由最近访问时间、修改时间和驻留状态更改时间得出的 简单时限。在另外一些情况下，您可能希望文件的访问时限优先于修改时限。您可以使 用指令指定在计算文件的释放优先级时要使用的加权时限优先级。

[代码示例 4-1](#) 显示了时限优先级指令的格式。

代码示例 4-1 时限优先级指令的格式

```
weight_age = float
weight_age_access = float
weight_age_modification = float
weight_age_residence = float
```

- `weight_age` 指令指定针对文件的默认时限（文件访问时限、修改时限和驻留时限三者中的较小者）给定加权因子。其中的 *float*，用于指定一个介于 0.0 到 1.0 之间的浮点数。默认情况下，*float* = 1.0。

该指令不能与 `weight_age_access`、`weight_age_modification` 或 `weight_age_residence` 指令结合使用。

- `weight_age_access`、`weight_age_modification` 和 `weight_age_residence` 指令指定由这些可能时限中的一个、二个或三个组合起来而确定的文件时限。其中的 *float*，用于指定一个介于 0.0 到 1.0 之间的浮点数。默认情况下，*float* = 1.0。

这些指令不能与 `weight_age` 指令结合使用。

如果同时使用了 `weight_age_access`、`weight_age_modification` 和 `weight_age_residence` 指令，则文件的与时限相关的优先级的计算过程如下：

1. 收集每一个文件的可能时限数据。
2. 将文件时限数据与 `releaser.cmd` 文件中指定的加权因子相乘。
3. 将每个乘积加在一起，如以下等式所示。

$$\begin{aligned} & \text{file access age} * \text{weight_age_access} \\ & + \text{file modification age} * \text{weight_age_modification} \\ & + \text{file residency age} * \text{weight_age_residence} \\ \hline & = \text{age_related_priority} \end{aligned}$$

代码示例 4-2 显示了 `releaser.cmd` 文件中的指令行，它们指定在计算文件的释放优先级时，只考虑文件的驻留时限，而忽略修改时限和访问时限。

代码示例 4-2 `releaser.cmd` 文件片段

```
weight_age_residence = 1.0
weight_age_modify = 0.0
weight_age_access = 0.0
```

计算文件的与时限相关的优先级之后，将其与文件的与大小相关的优先级相乘。下一节介绍了如何计算与文件大小相关的优先级。

文件大小

释放程序在确定与大小相关的文件释放优先级要素时将考虑文件的大小。文件的大小（以 4 KB 块为单位）乘以 `weight_size` 指令指定的权重，即可得出与大小相关的文件释放优先级要素。

`weight_size` 指令的格式如下所示：

```
weight_size = float
```

其中的 *float*，用于指定一个介于 0.0 到 1.0 之间的浮点数。默认情况下，*float* = 1.0。

代码示例 4-3 显示了 `releaser.cmd` 文件，该文件指定在计算文件的释放优先级时，忽略 `samfs1` 和 `samfs2` 文件系统中所有文件的大小。

代码示例 4-3 `releaser.cmd` 文件

```
# 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/samfs1.releaser.log
```

`fs` 指令：指定用于单个文件系统的指令

您可以在 `releaser.cmd` 文件中使用 `fs = family-set-name` 指令指定 `fs =` 指令之后的指令仅适用于指定的文件系统。

此指令的格式如下：

```
fs = family-set-name
```

其中的 *family-set-name*，用于指定 `mcf` 文件中的系列集名称。

位于第一个 `fs =` 指令前面的指令是应用于所有文件的全局指令。`fs =` 指令后面的指令可以覆盖全局指令。本章所述的指令既可用于全局指令，也可用作专用于一个文件系统的指令。

`releaser.cmd(4)` 手册页中提供了 `fs =` 指令的示例。

no_release 和 display_all_candidates 指令：指定调试

在调节或调试释放程序时，no_release 和 display_all_candidates 指令可能非常有用。这几个指令如下：

- no_release 指令可以防止文件从联机磁盘高速缓存中被删除。可以使用此指令来检查 releaser.cmd 文件中的指令，这不会真正地释放文件。

此指令的格式如下：

```
no_release
```

- display_all_candidates 指令可以将所有释放备选文件的名称写入至日志文件。

此指令的格式如下：

```
display_all_candidates
```

由于释放程序可将释放备选文件的名称写入至日志文件，而不是真正从文件系统中释放这些文件，因此在调试过程中，这些指令很有帮助。

min_residence_age 指令：指定最短驻留时间

min_residence_age 指令可用于指定文件在成为释放备选文件之前，必须在文件系统中驻留的最短时间。

此指令的格式如下：

```
min_residence_age = time
```

其中的 *time*，用于指定驻留时间，以秒为单位。默认时间为 600 秒，即 10 分钟。*time* 设置的最大或最小值并没有具体规定。

logfile 指令：指定日志文件

如果在 `releaser.cmd` 文件中指定了 `logfile` 指令，则释放程序会将其活动日志附加到指定的文件，或创建该文件（如果指定的文件不存在）。

此指令的格式如下：

```
logfile = filename
```

其中的 *filename*，用于指定日志文件的名称。

[代码示例 4-4](#) 显示了日志文件样例。

代码示例 4-4 释放程序日志文件示例

```
Releaser begins at Wed Apr 28 17:29:06 2006
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 2006) 10001 min, 500 blks /sam1/testdir0/filevp
10500 (R: Wed Apr 21 18:48:10 CDT 2006) 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
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 2006
```

`releaser(1M)` 手册页介绍了日志文件中包含的信息。随着释放程序的每一次运行，日志的大小会不断增加，因此，您应该进行规划以轮转日志文件，或省略 `logfile` 关键字。

`rearch_no_release` 指令：阻止释放重新归档的文件

默认情况下，系统会释放标记为重新归档的文件。如果在 `releaser.cmd` 文件中指定 `rearch_no_release` 指令，则释放程序不会释放标记为重新归档的文件。此指令的格式如下：

```
rearch_no_release
```

`list_size` 指令：调整释放程序备选文件列表的大小

可以使用 `list_size` 指令来指定释放程序备选文件的数量。如果您注意到释放程序为了达到下限而释放所需数量的文件之前，在对多个文件系统执行扫描，那么可能需要考虑增大这个值以使之大于默认值 10,000。在包含许多小文件的文件系统中就可能存在这样的情况。您可以从释放程序日志文件中，获取有关释放程序的活动信息。

此指令的格式如下：

```
list_size = number
```

其中的 *number*，用于指定一个介于 10 到 2,147,483,648 之间的整数。

指定归档集中所有文件的释放属性

archiver.cmd 文件中的大多数指令会影响归档过程，但归档集分配指令允许您指定应用于归档集中所有文件的释放属性。

表 4-3 显示了与释放相关的归档集分配值。

表 4-3 用于释放的归档集分配值

指令	作用
-release a	指定释放程序应在创建第一个归档副本后，释放归档组中的文件。如果您需要为每一个文件创建多份归档副本，请勿使用此选项。在此情况下，系统会登台第一份副本以创建第二份副本。
-release d	重置为默认设置。
-release n	指定不释放归档组中的文件。
-release p	指定释放程序应在归档文件后，部分释放归档组中的文件。

有关这些及其他 archiver.cmd 指令的更多信息，请参见第 29 页的“归档”。

规划释放程序的操作

您有必要为您的站点确定高速缓存中文件的特征。如果只需登台几千字节的小文件，载入磁带是不经济的，因此您可能希望配置系统以将小文件保留在高速缓存中。代码示例 4-5 显示了 releaser.cmd 文件中用于优先释放最大文件的指令。

代码示例 4-5 优先释放最大文件的指令

```
weight_size = 1.0
weight_age = 0.0
```

此外，您可能希望将最近修改过的文件保留在高速缓存中，因为您不久可能会重新修改它们。这可以避免因登台文件以进行修改而产生的开销。在这种情况下，可使用第二组时限权重（自最后一次修改文件以后的时限）。[代码示例 4-6](#) 显示了 `releaser.cmd` 文件中按从最早修改的文件到最新修改的文件的顺序对文件进行加权的指令。

代码示例 4-6 优先释放最早修改的文件的指令

```
weight_size = 0.0
weight_age_access = 0.0
weight_age_modify = 1.0
weight_age_residence = 0.0
```

不过，大多数情况并不采用这种简单直接的方式。

假定您希望首先释放最大的文件。文件系统中包括成百上千个大小相同的小文件和数个大文件。最终，释放程序将释放所有大文件。如果指定 `weight_age = 0.0`，则释放程序基本上按任意顺序释放小文件，因为它们的大小都相同，并且具有相同的释放优先级。

这种情况下，您可以设置 `weight_age = 0.01`，以首先释放大小相等的小文件中最旧的那一个。

此外，您还可以设置 `weight_size = 1.0` 和 `weight_age = 0.01`。

这两个指令通过优先选择较小的、不经常访问的文件（而不是较大的、经常访问的文件）作为释放备选文件，因而它们违背了首先释放最大文件的原则。不过，可以通过将 `weight_age` 的值设置为小于 `weight_size` 的值来降低这一影响。

例如，按照上面的设置，已登台 100 分钟的 4 KB 文件与刚刚登台的 8 KB 文件具有相同的释放优先级。但是，如果设置 `weight_age = 0.001`，则 4 KB 文件必须已登台 1,000 分钟才能与刚刚登台 8 KB 文件具有相同的优先级。

可以使用 `no_release` 和 `display_all_candidates` 指令以及手动运行释放程序以获得按优先级顺序排列的备选文件列表，这将有助于调整优先级权重。

手动运行释放程序

有时，您可能需要手动运行释放程序。要进行此项操作，您需要知道文件系统的挂载点以及您希望释放程序达到的下限。

例如，要释放 `/sam1` 文件系统中的文件，直至其占用率达到 47%，可以 `root` 身份登录，并键入以下命令：

```
# /opt/SUNWsamfs/sbin/sam-releaser /sam1 47 1.0
```

命令行选项将覆盖在 `releaser.cmd` 文件中指定的选项。当释放程序运行时，它将在屏幕上显示有关信息并将信息写入至释放程序日志文件（如果已在 `releaser.cmd` 文件中指定了一个日志文件）。有关更多信息，请参见 `sam-releaser(1M)` 手册页。

登台

登台是将文件数据从近线或脱机存储设备复制回联机存储设备的过程。

本章介绍 Sun StorageTek SAM 文件登台功能。本章包含以下各节：

- 第 121 页的“关于 `stager.cmd` 文件”
- 第 128 页的“排列预备请求的优先顺序”
- 第 131 页的“计算预备请求的总优先级”
- 第 132 页的“设置预备请求的优先级方案”

关于 `stager.cmd` 文件

可以使用 `stager.cmd` 文件指定登台程序的行为。该文件的完整路径名为 `/etc/opt/SUNWsamfs/stager.cmd`。登台程序的默认行为如下：

- 登台程序尝试使用库中的所有驱动器来登台文件。
- 登台缓冲区的大小由介质类型决定，并且不锁定登台缓冲区。
- 不写日志文件。
- 一次最多可以激活 1000 个登台请求。

可以使用 `stager.cmd` 文件指定有关指令来覆盖这些默认行为。您可以配置登台程序，以便立即登台文件、从不登台文件、指定部分登台以及指定其他登台操作。例如，不登台功能可供随机从大文件访问一小段记录的应用程序使用；启用该功能时，系统不需将文件登台为联机文件便可直接从归档介质中访问数据。

本节的后半部分将介绍登台程序的指令。有关登台程序指令的其他信息，请参见 `stager.cmd(4)` 手册页。

第 127 页的“[stager.cmd 文件示例](#)”显示了编写完毕的已设置所有可能指令的 `stager.cmd` 文件。

注 – 如果使用的是 File System Manager 软件，则可以通过“File System Summary”页或“File System Details”页来控制登台。您可以浏览文件系统并查看各个文件的状态，可以使用过滤器查看某些文件并选择要登台的特定文件。可以选择从哪个副本登台，也可以让系统选择该副本。

要设置登台程序指令，可以使用 `vi (1)` 或其他编辑器来编辑 `/etc/opt/SUNWsamfs/stager.cmd` 文件。然后使用带 `config` 选项的 `samd(1M)` 命令传播文件更改并重新启动系统：

```
# samd config
```

有关可在这个文件中添加哪些指令的信息，请参见以下几节：

- [第 122 页的“drives 指令：指定驱动器数量”](#)
- [第 123 页的“设置登台缓冲区大小”](#)
- [第 124 页的“指定日志文件”](#)
- [第 126 页的“指定登台请求的数量”](#)

drives 指令：指定驱动器数量

默认情况下，登台程序在登台文件时使用所有可用的驱动器。如果登台程序使所有驱动器处于繁忙状态，则会影响归档程序的活动。`drives` 指令用于指定登台程序可用的驱动器数量。此指令的格式如下：

```
drives = library count
```

表 5-1 `drives` 指令的参数

参数	含义
<i>library</i>	Sun StorageTek SAM <code>mcf</code> 文件中出现的库的系列集名。
<i>count</i>	所要使用的驱动器的最大数量。默认情况下，此数量与在 <code>mcf</code> 文件中为该库配置的驱动器数量相同。

例如，以下指令行指定只使用 dog 系列集库中的一个驱动器来登台文件：

```
drives = dog 1
```

有关 mcf 文件的更多信息，请参见 mcf(4) 手册页。

还可以使用 File System Manager 软件来指定此指令。有关更多信息，请参见 File System Manager 联机帮助。

设置登台缓冲区大小

默认情况下，登台程序在将要登台的文件从归档介质恢复回至联机磁盘高速缓存之前，首先将此文件读入缓冲区中的内存。可以使用 `bufsize` 指令来设置非默认的缓冲区大小和锁定缓冲区（可选）。这些操作可以改善系统性能。您可以尝试不同的 `buffer-size` 值以确定最适合的缓冲区大小。此指令的格式如下：

```
bufsize = media buffer-size [lock]
```

表 5-2 `bufsize` 指令的参数

参数	含义
<code>media</code>	指定 mcf(4) 手册页中列出的某归档介质类型。
<code>buffer-size</code>	一个介于 2 到 1024 之间的数字。默认值为 4。这个值乘以该介质类型的 <code>dev_blksize</code> 值，计算结果即为所使用的缓冲区大小。 <code>dev_blksize</code> 的值是在 <code>defaults.conf</code> 文件中指定的。为 <code>buffer-size</code> 指定的数值越大，所使用的内存就越多。有关此文件的更多信息，请参见 <code>defaults.conf(4)</code> 手册页。
<code>lock</code>	<p><code>lock</code> 参数指明登台程序在登台归档副本时应当使用锁定的缓冲区。如果指定 <code>lock</code>，归档程序将在复制操作期间在内存中的归档缓冲区上设置文件锁定。这可以避免由于为每一个 I/O 请求锁定和取消锁定缓冲区而造成的开销，从而减少占用系统 CPU 的时间。</p> <p>仅在配有大量内存的大型系统上，才有必要指定 <code>lock</code> 参数。如果内存不足，则可能会导致内存用尽。</p> <p>只有已为需要登台的文件启用直接 I/O 时，<code>lock</code> 参数才有效。默认情况下，不会指定 <code>lock</code> 参数，并且文件系统会在所有直接 I/O 缓冲区上设置锁定（包括用于登台的缓冲区）。有关启用直接 I/O 的更多信息，请参见 <code>setfa(1)</code> 手册页、<code>sam_setfa(3)</code> 库例程手册页，或 <code>mount_samfs(1M)</code> 手册页中介绍的 <code>-O forcedirectio</code> 选项。</p>

还可以使用 File System Manager 软件来指定此指令。有关更多信息，请参见 File System Manager 联机帮助。

指定日志文件

可以要求 Sun StorageTek SAM 软件收集文件登台事件的有关信息，并将这些信息写入日志文件。logfile 指令用于指定登台程序可在其中写入记录信息的日志文件。此指令的格式如下：

```
logfile=filename [ event ]
```

其中的 *filename* 用于指定完整的路径名，

event 用于指定一个或多个登台事件。如果要指定多个事件，请使用空格将各个事件分隔开。表 5-3 中列出了一些可能的事件规范。

表 5-3 *event* 参数的值

值	操作
all	记录所有登台事件。
start	记录文件开始登台的时间。
finish	记录文件结束登台的时间。默认启用。
cancel	记录操作员取消登台操作的时间。默认启用。
error	记录登台错误。默认启用。

指定了日志文件后，登台程序会将每一个登台文件的有关信息写入至日志文件中的一行或多行。该行中包括文件名、登台日期和时间以及卷序列号 (volume serial number, VSN) 等信息。

以下指令行指定了文件 /var/adm/stage.log:

```
logfile=/var/adm/stage.log
```

代码示例 5-1 显示了登台程序日志文件的示例。

代码示例 5-1 登台程序日志文件示例

```
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 -
S 2003/12/16 14:07:31 dk disk01 23.201d 3005.1381 556807 /sam1/testdir0/fileeq
1 root other root 0 -

S 2003/12/16 14:07:47 dk disk01 26.c4d 2831.1113 1428718 /sam1/testdir0/fileez
1 root other root 0 -
S 2003/12/16 14:07:47 dk disk02 1b.835 3736.59 1787855 /sam1/testdir1/filedp 1
root other root 0 -
```

如代码示例 5-1 所示，登台程序日志文件包含的信息行可拆分为九个字段。表 5-4 对登台程序日志文件字段的内容进行了介绍。

表 5-4 登台程序日志文件字段

字段	示例值	内容描述
1	S	登台活动 - s 表示启动，c 表示已取消，E 表示错误，F 表示已完成。
2	2003/12/16	登台操作发生的日期，格式为 <i>yyyy/mm/dd</i> 。
3	14:06:27	登台操作发生的时间，格式为 <i>hh:mm:ss</i> 。
4	dk	归档介质类型。有关介质类型的信息，请参见 <i>mcf(4)</i> 手册页。
5	disk01	VSN。

表 5-4 登台程序日志文件字段（续）

字段	示例值	内容描述
6	e.76d	归档文件（tar(1) 文件）在介质上的起始物理位置和归档文件中的文件偏移量（采用十六进制格式）。
7	2557.1759	Inode 编号和生成编号。为了保持唯一性，除了 inode 编号之外，还要使用生成编号，因为 inode 编号可以重复使用。
8	1743132	文件的大小。
9	/sam1/testdir0/filebu	文件的名称。
10	1	归档副本数。
11	root	文件的用户 ID。
12	other	文件的组 ID。
13	root	请求者的组 ID。
14	0	待登台文件所在驱动器的设备序号。
15	-	此字段中的 v 表示针对文件使用数据验证。

还可以使用 File System Manager 软件来指定此指令。有关更多信息，请参见 File System Manager 联机帮助。

指定登台请求的数量

可以使用 `maxactive` 指令指定一次可激活的登台请求的数量。

此指令的格式如下：

```
maxactive=number
```

默认情况下，*number* 的值为 4000。所允许的最小值为 1。所允许的最大值为 500,000。

例如，以下指令行指定队列中最多可以同时存在 500 个登台请求。

```
maxactive=500
```

stager.cmd 文件示例

代码示例 5-2 显示了 stager.cmd 文件的一个示例。

代码示例 5-2 stager.cmd 文件示例

```
# This is stager.cmd file /etc/opt/SUNWsamfs/stager.cmd
drives=dog 1
bufsize=od 8 lock
logfile=/var/adm/stage.log
maxactive=500
```

指定归档集中所有文件的登台属性

archiver.cmd 文件中的大多数指令会影响归档过程，但归档集分配指令允许您指定应用于归档集中所有文件的登台属性。

第 3 章中详细地介绍了归档集分配指令及其参数。表 5-5 显示了归档集分配指令可包含的登台指令。

表 5-5 archiver.cmd 文件可包含的登台指令

指令	作用
-stage a	指定归档集中的文件应联合登台。
-stage d	重置为默认设置。
-stage n	指定不登台归档集中的文件。

有关这些指令及其他 archiver.cmd 指令的更多信息，请参见第 29 页的“归档”。

排列预备请求的优先顺序

归档程序和登台程序进程均可请求载入及卸载介质。如果请求的数量超过可用于介质载入的驱动器数量，则多余的请求会发送至预备队列。

预备队列中的归档和登台请求是指那些无法立即满足的请求。默认情况下，系统按先进先出 (First In First Out, FIFO) 的顺序执行预备请求。

预备队列中的条目数量取决于 `defaults.conf` 文件中的 `previews=` 指令。有关更改该指令值的信息，请参见 `defaults.conf(4)` 手册页。

您可以为各个预备请求指定不同的优先级。可以通过在预备命令文件中输入指令来覆盖 FIFO 默认值，此预备命令文件的位置为 `/etc/opt/SUNWsamfs/preview.cmd`。

此文件根据请求是用于文件登台还是归档来安排预备请求。可以提高特定 VSN 的优先级。`preview.cmd` 文件中的设置也可根据上限 (high-water mark, HWM) 或下限 (low-water mark, LWM) 设置来重新排定所有文件或特定文件系统预备请求的优先级。

`sam-amld` 守护进程在启动时读取预备请求指令。这些指令必须是每一条独占一行。如果在 `sam-amld` 守护进程正在运行时更改此文件，那么必须重新启动 `sam-amld` 守护进程，以使更改生效。注释行以井字符 (#) 开头，并且延伸至行的末尾。有关此文件的更多信息，请参见 `preview.cmd(4)` 手册页。

`preview.cmd` 文件可以包含以下指令类型：

- 全局指令，应用于所有文件系统。这些指令必须出现在第一行 `fs =` 前。
- 专用于单个文件系统的指令。在文件中，这些指令必须出现在所有全局指令之后。

文件系统指令以 `fs = file-system-name` 开头。该指令用于指定文件系统，其后的所有指令均应用于该文件系统。文件中可以包含多个文件指令块。文件系统指令的应用范围到出现下一行 `fs =` 或到达文件末尾为止。

注 — 当多条指令影响一个文件系统时，专用于该文件系统的指令将覆盖全局指令。

全局 VSN 和时限指令

VSN 指令和时限优先级指令是全局指令，因此它们在 `preview.cmd` 文件中位于文件系统专用指令之前。

VSN 优先级指令的格式如下：

```
vsn_priority = value
```

该指令是一个静态优先级因子，表示标有高优先级的 VSN 的总优先级将要增加的值。`vsn_priority` 的默认值为 1000.0。当 VSN 被安排为预备请求时，必须已设置了其优先级标志，才能获得此增加值。可以使用带 `p` 选项的 `chmed(1M)` 命令，来设置优先级标志（例如 `chmed +p lt.AAA123`）。只有提交的所有 VSN 请求尚不是预备请求时，此标志才有效。

时限优先级指令的格式如下：

```
age_priority = factor
```

该指令也是一个静态优先级因子，尽管它的整体影响是动态的。`age_priority` 因子与请求成为预备请求的秒数（即请求等待的时间）相乘，其结果将与请求的总优先级相加。请求的等待时间越长，时限因子就越大。设置该因子有助于确保较旧的请求，不会被具有其他较高优先级因子的较新请求无限期地取代。

如果该因子大于 1.0，则它可以增加时间因子在计算总优先级中的重要性。如果小于 1.0，则会降低时间因子的重要性。如果将该因子设置为 0.0，则在计算总优先级时不考虑时间因子。

对于没有设置优先级标志的 VSN，系统将根据它在队列中等待的时间提高其优先级。这样，该 VSN 的优先级可能会高于以后进入队列且已设置优先级标志的 VSN。

全局界限指令或文件系统专用的界限指令

预备请求的界限指令既可以用作全局指令，也可用作文件系统专用指令。界限优先级指令用于确定预备请求的界限优先级，如下等式所示。

$$\begin{aligned} & \text{lwm_priority} + \\ & \text{lhwm_priority} + \\ & \text{hlwm_priority} + \\ & \text{hwm_priority} \\ & \hline & = \text{water mark priority} \end{aligned}$$

当界限优先级因子是正数时，计算出的总优先级所产生的影响是：提高归档请求的优先级而降低登台请求的优先级。反之，当界限优先级因子是负数时，归档请求的总优先级会降低，这会导致系统优先处理登台请求，然后再处理归档请求。界限优先级因子为 0.0（或根本不指定命令）表示不采取任何特殊的操作。有关更多信息，请参见第 133 页的“[示例 1：强制执行登台请求](#)”中的示例。

表 5-6 显示了四个界限优先级指令及其参数。

表 5-6 界限优先级指令

优先级指令	参数
<code>lwm_priority = value</code>	其中的 <i>value</i> ，用于指定当文件系统低于下限时，希望归档请求的界限优先级因子更改的数量。默认设置为 0.0。
<code>lhwm_priority = value</code>	其中的 <i>value</i> ，用于指定当文件系统从低于下限上升至下限以上，但仍然低于上限时，希望归档请求的界限优先级因子更改的数量。这通常表示文件系统中的文件正在增加。默认设置为 0.0。
<code>hlwm_priority = value</code>	其中的 <i>value</i> ，用于指定当文件系统从超过上限降至上限以下，但仍然高于下限时，希望归档请求的界限优先级因子更改的数量。这通常表示释放程序不能释放足够的磁盘空间，以使文件系统低于下限。默认设置为 0.0。
<code>hwm_priority = value</code>	其中的 <i>value</i> ，用于指定当文件系统超过上限时，希望归档请求的界限优先级因子更改的数量。默认设置为 0.0。

总之，这四个界限设置用于创建包括百分比值（表示文件系统的占用率）以及上限和下限设置级别在内的动态优先级因子。指定给预备请求的值取决于优先级因子是全局性的、文件系统专用的、还是未设置。

当文件系统的情况发生变化时，系统将根据以下条件重新计算与该文件系统关联的每一个 VSN 的优先级：相应的界限优先级设置；是否使用 `chmed(1M)` 命令的 `p` 选项设置了优先级标志。

界限优先级仅用于计算与归档有关的介质请求，而不能用于计算与登台有关的介质请求。

[代码示例 5-3](#) 显示了一些设置，这些设置可用于启用释放程序以释放足够的磁盘空间，从而使文件系统低于下限。

代码示例 5-3 使文件系统低于下限的设置

```
lhwm_priority = -200.0
hlwm_priority = 100.0
```

计算预备请求的总优先级

预备请求的优先级数值由静态和动态因子共同决定。数值越大，优先级就越高。静态优先级因子在生成请求时进行设置。一旦请求生成并进入等待执行状态，静态优先级因子对总优先级的影响将不会发生变化。在请求等待执行期间，动态优先级因子可以提高或降低请求的总优先级。

预备请求的总优先级是所有优先级因子的总和，其计算方式如下：

```
total_priority = vsn_priority + wm_priority + (age_priority *
time_in_sec_as_preview_request)
```

设置预备请求的优先级方案

除非具有如下的强制性理由，否则请勿更改默认的预备请求 FIFO 方案：

- 确保首先处理登台请求，然后处理归档请求。
- 确保归档请求在文件系统将要充满时获得最高优先级。
- 将使用特定介质组的请求排在预备请求列表的顶部。

代码示例 5-4 显示了可以满足这三个条件的 `preview.cmd` 文件示例。

代码示例 5-4 `preview.cmd` 文件示例

```
# 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
```

对于用户对数据访问要求极高、VSN 驱动器数受限制或将文件归档作为后台功能的环境，可以使用 `preview.cmd` 文件来改变存储系统资源执行登台请求的方式。可以对 `preview.cmd` 文件的设置进行自定义，以支持上述所有方案，并影响已配置的 Sun StorageTek SAM 环境。

由于该文件中的设置对数据并无影响，因此建议尝试并调整不同的指令设置，以便在权衡每一个预备请求的优先级时，能够在归档请求和登台请求之间找到适当的平衡点。

示例 1：强制执行登台请求

以下计算示例显示如何使用一个负 `wm_priority` 值来确保登台请求优先于归档请求。本示例假定：

- 队列中已经有多个请求的等待时间达到了 100 秒。
- `vsn_priority` 的默认值为 1000。

表 5-7 显示了如何计算请求的总优先级。

表 5-7 请求优先级计算示例

优先级	计算
具有优先级的归档 VSN，下限：	$1000 + (-200) + (1 \times 100) = 900$
具有优先级的登台 VSN，下限：	$1000 + 0 + (1 \times 100) = 1100$
无优先级的登台 VSN，下限：	$0 + 0 + (1 \times 100) = 100$

示例 2：强制执行归档请求

当在环境中权衡将文件登台回用户与将新文件归档至介质这二者的重要性时，最为关心的是超过上限的情形。在此情况下，如果没有足够的满足归档要求的文件来降低充满状态文件系统的占用率，则满足待定的归档请求是防止文件系统充满的最佳方法。

在此情况下，只需在 `preview.cmd` 文件中进行以下设置：

```
hwm_priority = 500.0
```

示例 3：根据介质确定请求的优先级

假定有一个面向项目的环境，在此环境下，特定用户正在处理使用特定 VSN 的文件组，并且与其他用户相互隔离。在这种环境中，某些项目有时可能具有较高的优先级；因此，它们需要具有优先使用可用系统存储资源的权利。这时可以使用以下指令配置 `preview.cmd` 文件，以适当给予用户及其介质优先使用介质驱动器的权利：

```
hwm_priority = 5000.0
```

然后，对于每一个属于优先级用户群组的 VSN，输入以下信息：

```
# chmed +p lt. VSN
```

以后，每一个要求访问特定 VSN 的请求均优先于预备队列中的其他待挂载请求。

将来，要降低用户介质的优先级，请为每一个 VSN 输入相反的命令：

```
# chmed -p lt.media-type
```

注 - 如果已设置 chmed(1M) 命令的 p 标志，则要求访问选定 VSN 组的请求将始终优先于预备请求队列中的其他请求。

示例 4：确定复杂请求的优先级

假设存在两个具有以下要求的 Sun StorageTek SAM 文件系统：

- 请求在队列中的等待时间不能太长 (age_priority)。
- 当其中的一个文件系统低于下限时，应优先处理登台请求。
- 当其中的一个文件系统高于下限且低于上限时，无需区分归档请求或登台请求的优先级。

代码示例 5-5 显示了受影响的指令。

代码示例 5-5 指令

```
lwm_priority = -200.0
lhwm_priority = 0.0
hlwm_priority = 0.0
```

其他指令保持不变。

当其中的一个文件系统超过上限时，优先处理归档请求。

假定两个文件系统同时超过了上限，则应首先防止第二个文件系统（例如 samfs2）充满。例如，当 samfs1 是一个用户工作文件系统，而 samfs2 是一个关键文件系统时，便会出现这种情况。

代码示例 5-6 显示了可根据前面列出的要求确定请求优先级的 preview.cmd 文件。

代码示例 5-6 显示如何确定复杂请求优先级的 preview.cmd 文件

```
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

回收是指从归档卷中收回空间的过程。回收程序与归档程序配合工作，以收回由无用的归档副本占用的空间。当用户修改某个文件时，即可从系统中清除与该文件的旧版本相关联的归档副本。回收程序可以识别那些其中绝大部分是过期归档副本的归档卷，并将这些卷中的非过期副本移动到其它卷中。当某个给定的卷中只包含过期副本时，即可执行站点定义的操作。例如，可以重新标记此类卷以便立即重新使用此类卷，或将其中的数据导出至离站存储设备，从而单独保存文件更改的历史记录。由于回收过程只与用户的数据文件有关，因此用户不会觉察到回收过程。

本章介绍回收过程和回收指令。其中包括以下主题：

- [第 135 页](#)的“回收过程概述”
- [第 137 页](#)的“回收磁带库”
- [第 145 页](#)的“回收磁盘归档卷”
- [第 148 页](#)的“回收以实现归档副本保留”

回收过程概述

回收程序负责将过期归档副本占用的空间，保持在由站点指定参数定义的最低水平。在任何时候，给定归档卷的空间均由以下各项组成：

- 当前数据，由当前有效的归档映像组成
- 过期数据，由当前不再有效的归档映像组成
- 可用空间，由未被当前有效或过期的归档映像占用的空间组成

卷的容量是指卷中可用于存储数据的总空间量。例如，对于一个已写入 3 GB 数据的 10 GB 磁带卷来说，它的容量为 10 GB，可用空间为 7 GB。

对于全新的归档介质或新标记的归档介质，其容量等于可用空间。当将数据归档至该介质时，可用空间会减少，而当前数据空间会增加。

当更改或删除文件系统中已归档的文件时，这些文件的归档映像会过期，并且其类别由当前数据类别变为过期数据类别。这些映像占用的物理空间并没有发生变化；只是文件系统中已没有指向该空间的文件。只有回收空间，才能删除这些映像并使它们占用的空间变为可用空间。回收程序的目标是将过期数据占用的空间转变为可用空间，而丝毫不损坏任何当前数据。

回收程序和归档程序协同工作，具体如下：

1. 回收程序使用 `rearchive` 属性来标记卷中所有当前（有效）的归档映像。
2. 如果是将文件归档至可移除介质，则回收程序使用 `recycle` 属性来标记所选择的归档卷。这可阻止归档程序再向此卷中写入归档映像。
3. 归档程序将所有已做标记的映像移至另一个卷。此操作过程称为**重新归档**。当归档程序将当前的归档映像从旧卷移至新卷后，旧卷中只包含可用空间和过期数据空间。如果是归档至可移除介质卡盒，则可以重新标记并重新使用该卡盒。如果是归档至磁盘，则回收程序将删除包含已过期归档映像的文件。

回收程序可以定期运行。每次启动该程序后，它会尽力完成所有工作。在归档程序可以重新归档文件之前，回收程序必须完成要重新归档的副本的标记工作。

有时，设置了 `rearchive` 属性的已过期的归档映像仍会保留在介质中。这可能会发生在下列情况下：

- 在回收程序标记已过期的归档映像后，归档程序没有运行。
- 在移动未过期的归档映像时，归档程序没有可使用的介质。
- 归档程序出现了其他各种异常。

在两次运行期间，回收程序将状态信息保留在库目录和索引节点中。在回收过程中，可以使用 `s1s(1)` 命令及其 `-D` 选项来显示文件的相关信息。`s1s(1)` 命令的输出可以显示，是否已经预定对某个文件进行重新归档。

配置回收程序之前，请注意以下事项：

- `archiver.cmd` 文件中的指令按归档集控制回收过程。`recycler.cmd` 文件中的指令按库控制回收过程。此外，`recycler.cmd` 文件还控制着回收程序的一般操作。有关回收程序指令的信息，请参见第 137 页的“使用回收指令”。
- 请勿回收包含可移除介质文件的卷。可移除介质文件可以使用 `request(1)` 命令来创建。包含可移除介质文件的卷不能清除干净。
- 当 Sun StorageTek SAM 文件系统正在执行维护操作时，请勿运行回收程序。回收程序使用 `.inodes` 文件和 `mcf` 文件，来识别当前文件或过期文件以及与文件系统关联的设备。如果缺少 `.inodes` 和 `mcf` 文件中的正确信息，回收程序可能会将当前归档数据视为过期数据而对其执行回收操作。
- 所有 Sun StorageTek SAM 文件系统必须都已挂载，方可运行回收程序。如果要对在线磁盘执行回收操作，则必须挂载包含此磁盘卷的文件系统，并且主机系统必须可访问。

注意 – 如果要在具有多个 SAM-QFS 服务器的环境中使用磁盘归档，则在配置回收程序时必须格外小心。每个 SAM-QFS 服务器的 `diskvols.conf` 文件必须指向一组唯一的磁盘卷资源规范（磁盘归档目标目录）。如果不同的 SAM-QFS 服务器共享其中一个规范，则从一个 SAM-QFS 服务器中运行回收程序时，将会销毁由其他 SAM-QFS 服务器所管理的磁盘归档数据。

默认情况下，系统不会启用回收程序。必须通过输入 `sam-recycler(1M)` 命令来启动回收过程。

回收磁带库

您可以手动或者通过 `cron(1)` 作业输入 `sam-recycler(1M)` 命令来启动磁带库的回收过程。表 6-1 中显示了回收方法。

表 6-1 回收方法与介质类型

回收方法	介质和说明
按自动化库	可移除介质卡盒。 当按库进行归档时，可以在 <code>recycler.cmd</code> 文件中指定回收指令。
按归档集	可移除介质卡盒和磁盘。 当按归档集进行归档时，您会将所有回收指令放置在 <code>archiver.cmd</code> 文件中。

如表 6-1 所示，既可以按库进行回收，也可以按归档集进行回收。如果归档至磁盘，则只能按归档集进行回收。

使用回收指令

`recycler.cmd` 文件可以接受的指令如以下几节所述：

- 第 138 页的“指定日志文件：`logfile` 指令”
- 第 138 页的“阻止回收：`no_recycle` 指令”
- 第 139 页的“指定回收整个自动化库：库指令”
- 第 148 页的“回收以实现归档副本保留”

指定日志文件：logfile 指令

logfile 指令用于指定回收程序日志文件。此指令的格式如下：

```
logfile = filename
```

其中的 *filename*，用于指定日志文件的路径。

以下是 logfile= 指令行的一个示例：

```
logfile = /var/adm/recycler.log
```

阻止回收：no_recycle 指令

no_recycle 指令可以阻止卷的回收。要指定 VSN，可以使用正则表达式以及一个或多个特定介质类型。

此指令的格式如下：

```
no_recycle media-type VSN-regexP [ VSN-regexP ... ]
```

表 6-2 no_recycle 指令的参数

参数	含义
<i>media-type</i>	mcf(4) 手册页中列出的某一介质类型。
<i>VSN-regexP</i>	用于描述卷的一个或多个以空格分隔的正则表达式。有关信息，请参见 regexP(5) 手册页，或参见第 59 页的“使用模式匹配的文件名 search-criterion: -name regex ”。

通过指定介质类型，可以阻止回收存储在特定介质类型上的卷。可以通过指定一个或多个 *VSN-regexP* 规范，来标识那些免于回收的特定卡盒。

例如，以下指令行可阻止回收程序回收那些 VSN 标识符以 DLT 开头的磁带卷：

```
no_recycle lt DLT.*
```

指定回收整个自动化库：库指令

库指令可用于为那些与特定库关联的 VSN 指定各种不同的回收参数。

此指令的格式如下：

```
library parameter [ parameter ... ]
```

其中的 *library*，用于指定库名称，该名称与在 *mcF* 文件的系列集字段中指定的名称相同。

而其中的 *parameter*，用于指定一个或多个以空格分隔的 *parameter* 关键字（见表 6-3）。

表 6-3 库指令中 *parameter* 的值

<i>parameter</i> 的值	含义
-dataquantity size	回收程序清除含有有用数据的卷时可重新归档调度的最大数据量。默认值为 1 GB。
-hwm percent	库上限。默认值为 95。
-ignore	阻止回收此库中的卷的指令。在测试 <i>recycler.cmd</i> 文件时，此指令十分有用。
-mail email-address	回收电子邮件发送至的电子邮件地址。默认情况下，系统不发送电子邮件。
-mingain value	最小 VSN 增益百分比。默认值为 50。
-vsncount count	可回收卷的最大数量。默认值为 1。

以以下的指令行为例：

```
gr47 -hwm 85 -ignore -mail root -mingain 40
```

它为库 *gr47* 指定了以下各项：

- 当库中卷的占用率达到 85% 时，应考虑对库执行回收操作。
- 最小增益百分比为 40%。
- 只能回收一个卷。这也是一个默认设置。
- 将回收消息发送至 *root* 用户。

此外，重新归档的数据量最多为 1 GB。这是一个默认值，因此未在 *recycler.cmd* 文件中指定。

配置回收程序

启动回收程序后，第 139 页的“指定回收整个自动化库：库指令”中指定的默认回收程序设置将会生效。有关回收程序的更多信息，请参见 `sam-recycler(1M)` 手册页。

配置磁带库的回收程序涉及以下各节所述的任务，顺序如下：

1. 第 140 页的“创建 `recycler.cmd` 文件”
2. 第 146 页的“编辑 `archiver.cmd` 文件”（此步骤对于回收磁带库为可选项）
3. 第 143 页的“运行回收程序”
4. 第 144 页的“为回收程序创建 `crontab` 文件”
5. 第 144 页的“删除 `-recycle_ignore` 和 `ignore` 参数”
6. 第 145 页的“创建 `recycler.sh` 文件”

创建 `recycler.cmd` 文件

如果要回收库中卡盒上的归档副本，请创建 `recycler.cmd` 文件。

注 – 即使按归档集执行回收操作，也应在 `recycler.cmd` 文件中配置每一个库。这可以确保回收程序能够回收不属于归档集的 `VSN`（如有必要）。

`recycler.cmd` 文件中包含通用的回收指令，也可包含针对 Sun StorageTek SAM 环境中每一个库的指令。典型的 `recycler.cmd` 文件中包含以下指令行：

- `logfile=` 指令行，用于指定回收程序日志文件。系统将回收消息和回收报告写入至该文件。
- 针对每一个含有待回收卷的库的一行或多行指令。该指令行必须包含要回收的库的系列集名（来自 `mcf` 文件）。系列集名使回收程序可以识别库。有关回收指令的信息，请参见第 137 页的“使用回收指令”。

由于正在创建 `recycler.cmd` 中的行，并且尚未经过测试，因此应使用 `ignore` 关键字。本过程后面的步骤中会指导您删除 `ignore` 关键字。

▼ 创建 `recycler.cmd` 文件

1. 成为超级用户。
2. 使用 `vi(1)` 或其他编辑器打开文件 `/etc/opt/SUNWsamfs/recycler.cmd`。
3. 添加本章所述的一行或多行指令来控制回收程序的活动。
4. 保存并关闭该文件。

或者，您可以使用 File System Manager 软件来创建 `recycler.cmd` 文件。有关更多信息，请参见 File System Manager 联机帮助。

`recycler.cmd` 文件示例

代码示例 6-1 显示了 `recycler.cmd` 文件的一个示例。

代码示例 6-1 `recycler.cmd` 文件示例

```
logfile = /usr/tmp/recycler.log
stk30 -hwm 51 -mingain 60 -ignore -mail root
```

以下部分介绍代码示例 6-1 中指定的参数。

`-hwm 51` 参数

通过指定上限，可设置介质使用率百分比，低于此百分比时便无法进行回收。此百分比是指库中已用空间与总容量的比率。例如，某个库含有 10 个 20 GB 的磁带，其中三个磁带的利用率为 100%，另外七个磁带的利用率均为 30%，则介质利用率为：

$$\frac{(3 * 1.00 + 7 * 0.30) * 20G}{(10 * 20G)} * 100\% = 51\%$$

请注意，该计算方法并不区分当前数据和过期数据，它只考虑介质的使用量。

在本示例中，如果空间利用率为 51% 或更小，回收程序不会自动选择自动化库的任何 VSN 进行回收。

注 – 可以使用以下命令设置回收标志，从而强制回收某 VSN：

```
# chmed +c lt.VSN
```

设置 `+c` 标志后，归档程序不会再向此卷中写入任何归档映像。可以通过 `samu(1M)` 实用程序来查看 `+c` 标志。有关更多信息，请参见 `chmed(1M)` 和 `samu(1M)` 手册页。有关如何使用 `samu(1M)` 操作员实用程序的信息，请参见《Sun StorageTek QFS 文件系统配置和管理指南》。

-mingain 60 参数

minimum VSN gain percentage (最小 VSN 增益百分比) 用于设置通过回收卡盒所获得的空间增加量的下限。例如, 在自动化库的某个卡盒中, 95% 为当前数据空间, 另外 5% 是过期数据空间, 因此通过回收该卡盒所获得的增益百分比仅为 5%。为获得此空间 (即 5%) 而移动另外 95% 的空间, 这可能是不值得的。将最小增益百分比设置为 6% 或更大值, 可以防止回收程序自动选择此 VSN。

-ignore 参数

-ignore 参数用于防止回收程序回收某个特定的库。在配置回收程序时, 请使用此参数。

-mail 参数

-mail 参数用于指定回收程序在回收指定的库后发送电子邮件。电子邮件的主题行如下所示:

```
Robot robot-name recycle
```

表 6-2 显示了消息正文样例。

代码示例 6-2 回收消息样例

```
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.
```

▼ 运行回收程序

请按照以下说明手动运行回收程序。

1. 运行 sam-recycler(1M) 命令。

回收程序将读取 recycler.cmd 文件。

2. 检查标准输出日志、Sun StorageTek SAM 日志和 /var/adm/messages，查看是否存在来自回收程序的错误消息。

如果有错误，请更正文件。

代码示例 6-3 显示了回收可移除介质卡盒时的回收程序日志文件样例。

代码示例 6-3 可移除介质卡盒的回收程序日志文件示例

```
===== 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.
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 =====

```

为回收程序创建 crontab 文件

如果系统运行正常，则可以为超级用户创建 crontab 条目，以定期运行回收程序。所选的运行频率视您的站点条件而定。

有关创建 crontab 条目的说明，请参见 cron(1M) 手册页。

以下是超级用户的 crontab 文件中的条目示例，它指定 cron 守护进程在每个奇数小时内每隔五分钟运行一次回收程序：

```

5 1,3,5,7,9,11,13,15,17,19,21,23 * * * /opt/SUNWsamfs/sbin/sam-recycler

```

▼ 删除 -recycle_ignore 和 ignore 参数

如果在配置过程中使用了 -recycle_ignore 或 ignore 参数禁用回收，请使用以下过程删除这些参数。如果不使用以下过程，可以选择使用 File System Manager。有关更多信息，请参见 File System Manager 联机帮助。

1. 使用 vi(1) 或其他编辑器删除 archiver.cmd 文件中的 -recycle_ignore 参数。
2. 使用 vi(1) 或其他编辑器删除 recycler.cmd 文件中的 ignore 参数。

现在，回收程序便真正开始执行回收过程。

创建 `recycler.sh` 文件

如果您要回收可移除介质卡盒上的归档副本，请创建 `recycler.sh` 文件。如果只将文件归档至磁盘，则请勿执行本步骤。

在归档程序将 VSN 中的所有当前映像重新归档至另一个 VSN 后，回收程序将执行 `recycler.sh` 脚本。您应确定您的站点对清除已回收卡盒的要求。某些站点选择重新标记或重新使用卡盒；而其他站点选择从自动化库中取出卡盒以便将来用于访问历史文件。

回收程序使用以下参数调用 `/opt/SUNWsamfs/scripts/recycler.sh` 脚本：

```
Media type: $1  VSN: $2  Slot: $3  Eq: $4
```

有关该脚本的示例，请参见 `recycler.sh(1M)` 手册页或 `/opt/SUNWsamfs/examples/recycler.sh`。后者显示了如何重新标记已回收的 VSN 并向超级用户发送邮件。

有关更多信息，请参见 `recycler(1M)` 和 `recycler.sh(1M)` 手册页。

回收磁盘归档卷

配置用于回收磁盘归档卷的回收程序涉及以下各节所述的任务，顺序如下：

1. [第 146 页的“编辑 `archiver.cmd` 文件”](#)
2. [第 147 页的“运行回收程序”](#)
3. [第 147 页的“为回收程序创建 `crontab` 文件”](#)
4. [第 147 页的“删除 `-recycle_ignore` 和 `ignore` 参数”](#)

编辑 archiver.cmd 文件

如果要归档至磁盘，必须编辑 archiver.cmd 文件才能进行回收。

如果要按库进行回收，此步骤为可选项。

要编辑 archiver.cmd 文件，请执行第 42 页的“创建或修改 archiver.cmd 文件并传播更改”中所述的步骤。

或者，可以使用 File System Manager 编辑 archiver.cmd 文件。有关更多信息，请参见 File System Manager 联机帮助。

要按照归档集启用回收，必须在 params 指令和 endparams 指令之间添加归档集回收指令。表 6-4 显示了可以使用的归档集回收指令。

表 6-4 归档集回收指令

指令	功能
<code>-recycle_dataquantity size</code>	限制回收程序可重新调度归档（以清除含有有用数据的磁盘卷）的数据量。默认情况下，将忽略磁盘归档回收的限制。
<code>-recycle_ignore</code>	阻止回收归档集。
<code>-recycle_mailaddr mail-address</code>	将回收程序消息发送至指定的电子邮件地址。
<code>-recycle_mingain percent</code>	通过为磁盘卷设置 mingain 标记，限制对归档集中卷的回收。mingain 以与卷关联的过期数据的百分比形式表示。当卷的过期数据超过 mingain 百分比时，回收程序便开始回收卷。默认设置为 50%。
<code>-recycle_minobs percent</code>	通过为回收程序对磁盘归档卷进行的重新归档过程设置阈值，限制回收程序选择卷中的 tar 文件。当磁盘上一个已归档的 tar 文件中的过期文件所占百分比达到此阈值时，回收程序会开始将该归档文件中的当前文件移到新的 tar 文件中。所有当前文件的移动操作完成之后，原始 tar 文件将被标记为要从磁盘归档中删除的备选文件。默认设置为 50%。
<code>-rearch_stage_copy copy-number</code>	允许从选择的（更快的）副本登台以重新归档。

有关归档程序指令的更多信息，请参见第 3 章，或参见 archiver.cmd(4) 手册页。

▼ 运行回收程序

请按照以下说明手动运行回收程序。

1. 运行 `sam-recycler(1M)` 命令。

回收程序将读取 `recycler.cmd` 文件。

2. 检查标准输出日志、Sun StorageTek SAM 日志和 `/var/adm/messages`，查看是否存在来自回收程序的错误消息。

如果有错误，请更正文件。

代码示例 6-4 显示了回收磁盘归档文件时的回收程序日志文件样例。

代码示例 6-4 磁盘归档文件的回收程序日志文件示例

```
---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)
```

为回收程序创建 crontab 文件

如果系统运行正常，则可以为超级用户创建 `crontab` 条目，以定期运行回收程序。运行频率视您的站点条件而定。

有关创建 `crontab` 条目的说明，请参见 `cron(1M)` 手册页。

以下是超级用户的 `crontab` 文件中的条目示例，它指定 `cron` 守护进程在每个奇数小时内每隔五分钟运行一次回收程序：

```
5 1,3,5,7,9,11,13,15,17,19,21,23 * * * /opt/SUNWsamfs/sbin/sam-recycler
```

▼ 删除 `-recycle_ignore` 和 `ignore` 参数

如果在配置过程中使用了 `-recycle_ignore` 或 `ignore` 参数禁用回收，请使用以下过程删除这些参数。如果不使用以下过程，可以选择使用 `File System Manager`。有关更多信息，请参见 `File System Manager` 联机帮助。

1. 使用 `vi(1)` 或其他编辑器删除 `archiver.cmd` 文件中的 `-recycle_ignore` 参数。

2. 使用 `vi(1)` 或其他编辑器删除 `recycler.cmd` 文件中的 `ignore` 参数。

现在，回收程序便真正开始执行回收过程。

回收以实现归档副本保留

作为常规回收过程的替代方法，可以将 `sam-nrecycler(1M)` 工具与 `File System Manager` 的备份和恢复点功能结合使用。此工具可删除过期归档副本并释放归档卷，从而有助于使用 `Sun StorageTek SAM` 转储文件来实现归档保留的功能。如果要利用此功能，必须使用此回收程序替代现有的 `sam-recycler` 命令。

`sam-nrecycler(1M)` 工具可以扫描文件系统元数据和 `Sun StorageTek SAM` 转储文件，以确定哪些可移除介质和磁盘归档卷包含归档映像。在非高峰时间可以通过 `crontab(1)` 文件调用此工具，也可以使用 `sam-nrecycler` 命令随时调用此工具。`nrecycler` 通过扫描所有文件系统 `.inodes` 文件和指定的 `Sun StorageTek SAM` 转储文件，来识别可移除介质卷或磁盘归档 `tar` 文件中存在的所有归档映像。通过扫描文件系统和 `Sun StorageTek SAM` 转储文件，`nrecycler` 可以确定是否存在不包含任何归档映像的卷以及是否可回收这些卷中的空间。如果某个可移除介质卷不包含任何归档映像，则可以安全地重新标记卡盒。如果某个磁盘归档 `tar` 文件不包含任何归档映像，则可以安全地从磁盘归档目录中删除此 `tar` 文件。

必须使用 `/etc/opt/SUNWsamfs/nrecycler.cmd` 文件提供 `sam-nrecycler(1M)` 的指令。您必须指定一个包含待搜索的 `Sun StorageTek SAM` 转储文件的目录路径。如果未在命令文件中指定目录，则不会进行回收。目录列表必须完整，且必须包含所有的 `Sun StorageTek SAM` 转储文件。

您也可以在 `nrecycler.cmd` 文件中包含 `logfile=` 指令行，以指定 `nrecycler` 日志文件。系统将回收消息和回收报告写入至该文件。

当 `sam-nrecycler(1M)` 检测到某个可移除介质卷仅包含可用空间或过期数据空间并可安全地进行重新标记时，它将调用 `sam-nrecycler.sh` 脚本。此脚本可以使用原始 `VSN` 或新 `VSN` 来重新标记卡盒。然后，它可以从库中导出卡盒，或者执行其他的用户定义操作。

当 `sam-nrecycler` 检测到某个磁盘归档卷仅包含可用空间或过期数据空间时，它将解除未使用磁盘归档 `tar` 文件的链接。

有关 `sam-nrecycler(1M)` 的更多信息，请参见 `sam-nrecycler(1M)` 手册页。

第7章

使用 Sun SAM-Remote 软件

借助 Sun SAM-Remote 客户机和 Sun SAM-Remote 服务器构成的客户机/服务器实施体系，您可以在 Sun StorageTek SAM 主机系统之间共享库和其他可移除介质设备。您可使用 Sun SAM-Remote 配置多个存储客户机，使它们能够从集中式磁带库或磁光盘库中归档和登台文件。例如，如果您的主机系统分布在一个跨越很大地域的网络中，则在一个城市中创建的文件可以归档至离此城市数英里远的库中的卡盒。

本章包括以下各节：

- 第 149 页的 “Sun SAM-Remote 软件概述”
- 第 155 页的 “配置 Sun SAM-Remote 软件”
- 第 168 页的 “使用 Sun SAM-Remote 软件回收”

Sun SAM-Remote 软件概述

本概述包含下列主题：

- 第 150 页的 “特性”
- 第 151 页的 “要求”
- 第 151 页的 “限制”
- 第 152 页的 “客户机与服务器之间的交互作用”

特性

图 7-1 显示了一个配有两台 Sun SAM-Remote 主机系统服务器的环境。每台服务器均具有两台客户机。

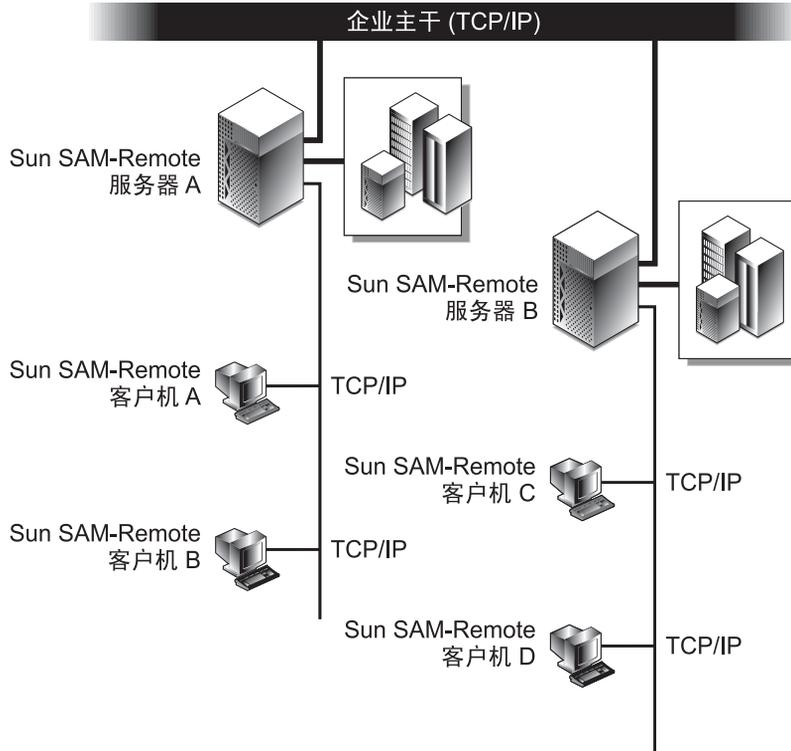


图 7-1 Sun SAM-Remote 服务器和客户机

Sun SAM-Remote 软件具有以下优点:

- 允许在两台或多台 Sun SAM-Remote 客户机之间远程共享价值昂贵的可移除介质资源（例如库）。
- 允许客户机向服务器转移数据。
- 允许多个 Sun StorageTek SAM 服务器彼此互为主机。在 Sun SAM-Remote 环境中，服务器是在 mcf 文件中被配置成 ss 设备类型的主机系统。

您可以配置 Sun SAM-Remote 服务器和客户机，以便在两台或多台 Sun Solaris 主机系统之间提供多份归档副本。例如，您可以对两台运行 Sun StorageTek SAM 软件的 Solaris 系统进行配置，使它们相互成为对方的 Sun SAM-Remote 服务器和 Sun SAM-Remote 客户机。这样配置的好处在于，您不仅可以为每一台服务器创建本地副本，而且还可以在另一台服务器上创建额外的数据归档副本。文件系统可在使用标准 NFS 的服务器之间共享。当无法访问本地库时，Sun SAM-Remote 软件会从归档副本中自动恢复文件数据。因此，使用这两台服务器的用户可以连续地访问各自的数据。即使在他们的主存储库不可用时，也是如此。

要求

尝试配置 Sun SAM-Remote 环境之前，请确保您的环境安装了以下软件和硬件：

- SPARC® 或 x64 系统应安装已获得使用许可的、可操作的 Sun StorageTek SAM 4U0 版或更高版本的存储和归档管理软件包。
- 主机系统应安装相同修订版本的 Sun StorageTek SAM 软件和相同的修补程序集合。如果某些主机系统需要升级，请参见《Sun StorageTek Storage Archive Manager 安装和升级指南》。
- 一个或多个用作 Sun SAM-Remote 服务器的主机系统，其中至少应安装一个 SAM-QFS 文件系统。
- 客户机和安装 Sun StorageTek SAM 软件的服务器之间应通过 TCP/IP 网络进行网络连接。

限制

Storage Archive Manager 对待远程库中卡盒的方式与对待本地库中卡盒的方式完全一样。不过，Sun SAM-Remote 软件存在以下限制：

- 您可以尝试使用 Sun SAM-Remote 回收介质，但前提是您已彻底测试您的环境。有关详细信息，请参见第 168 页的“使用 Sun SAM-Remote 软件回收”。
- Sun SAM-Remote 客户机上只有一个守护进程可以与 Sun SAM-Remote 服务器通信。
- Sun StorageTek SAM 软件（进而是 SAM-Remote）不能在共享 Sun StorageTek QFS 文件系统上的 Sun StorageTek QFS 客户机上运行。当运行 Sun StorageTek SAM 软件和 SAM-Remote 的服务器既是某些 Sun StorageTek QFS 文件系统的元数据服务器，又是其他 Sun StorageTek QFS 文件的客户机时，这些软件只能在那些将该服务器作为元数据服务器的文件系统上运行。

客户机与服务端之间的交互作用

Sun SAM-Remote 客户机和 Sun SAM-Remote 服务器通过 TCP/IP 连接进行交互。各个 Sun SAM-Remote 客户机之间的网络可以是 Solaris OS 支持的任何网络类型，如以太网、快速以太网或光纤通道等。

图 7-2 显示了 Sun SAM-Remote 客户机和 Sun SAM-Remote 服务器之间的交互作用。

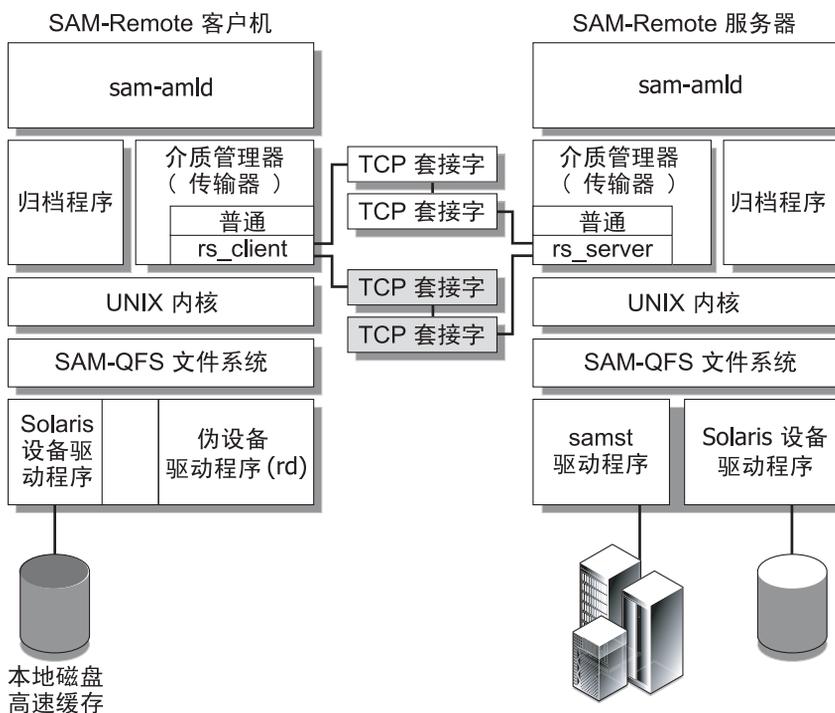


图 7-2 Sun SAM-Remote 服务器和客户机之间的交互作用

Sun SAM-Remote 服务器概述

Sun SAM-Remote 服务器不仅包括功能完备的 Sun StorageTek SAM 存储管理主机，而且还包括用于定义各个客户机要共享的库的 Sun SAM-Remote 服务器守护进程。Sun SAM-Remote 服务器上必须至少配置一个 SAM-QFS 文件系统。

通过在服务器系统的 `/etc/opt/SUNWsamfs/mcf` 文件中添加 `ss` 设备类型行，您可将主机系统定义成为 Sun SAM-Remote 服务器。您必须为每台服务器提供唯一的系列集名称。每个守护进程最多可以配置十台客户机。要配置十台以上的客户机，请在 `mcf` 文件中为要配置的每十台客户机添加一条额外的远程服务器条目。有关服务器守护进程的更多信息，请参见 `sam-remote(7)` 手册页。

Sun SAM-Remote 客户机概述

Sun SAM-Remote 客户机是一个 Sun StorageTek SAM 主机系统，该主机系统可用于建立包含多个伪设备的 Sun SAM-Remote 客户机守护进程。

通过在客户机系统的 `/etc/opt/SUNWsamfs/mcf` 文件中添加 `sc` 设备类型行，您可将主机系统定义成为 Sun SAM-Remote 客户机。有关客户机守护进程的更多信息，请参见 `sam-remote(7)` 手册页。

伪设备定义与 Sun SAM-Remote 服务器上的实际可移除介质设备相关的网络连接路径。伪设备的设备类型为 `rd`，即 **remote device**（远程设备）的缩写。您可以在 Sun SAM-Remote 客户机的 `/etc/opt/SUNWsamfs/mcf` 文件中定义伪设备。Sun SAM-Remote 守护进程和伪设备均与一台特定的服务器相关联。

Sun SAM-Remote 守护进程支持客户机的伪设备，并且不限制每台客户机的伪设备数量。客户机可以使用的实际伪设备数是可配置的。在确定每台客户机可用的伪设备数目时，您可以将客户机和服务器之间同时发生的数据传输流的个数作为伪设备的数目。定义多个伪设备时，将会增加网络的总通信量。身为系统管理员，您可以根据需要确定系统实际所需的伪设备数。

Sun SAM-Remote 服务器和 Sun SAM-Remote 客户机之间的交互作用

Sun SAM-Remote 服务器守护进程 `sam-serverd` 在端口 1000 上监听客户机。您可以在 Sun Solaris `/etc/services` 目录中使用服务名 `rmtsam` 配置一个不同的端口。当 Sun SAM-Remote 客户机连接至 Sun SAM-Remote 服务器时，`sam-serverd` 守护进程将在另一端口上建立连接，然后使用已定义的端口，将此端口的编号传送给该客户机。套接字的大小将传递给客户机。套接字大小是可配置的，有关更多的详细信息，请参见第 155 页的“配置 Sun SAM-Remote 软件”。

库目录

Sun SAM-Remote 库目录是 Sun SAM-Remote 服务器上的目录子集。客户机目录将实时进行更新。分配给 Sun SAM-Remote 客户机目录的插槽仅受 Sun SAM-Remote 服务器的控制。

初始化期间，系统会生成一个客户机目录，并根据 Sun SAM-Remote 服务器目录文件中的信息，将其传递给 Sun SAM-Remote 客户机。主机和客户机之间建立连接后，可供客户机使用的介质将被标为“可用”。如果客户机和服务器之间的连接中断，则客户机上的介质会被标为“不可用”。您可以通过 `samu(1M) v` 显示屏幕查看介质是否可用。客户机上 `samu(1M) v` 显示屏幕所显示的信息是服务器上 `v` 显示屏幕所显示信息的一部分。因此，最好通过 Sun SAM-Remote 服务器上的 `samu(1M) v` 显示屏幕访问介质目录。有关 Sun SAM-Remote 服务器客户机文件的更多信息，请参见第 155 页的“配置 Sun SAM-Remote 软件”。有关如何使用 `samu(1M)` 操作员实用程序的信息，请参见《Sun StorageTek QFS 文件系统配置和管理指南》。

对目录所做的更改将根据需要在主机之间传递。例如，当服务器目录的更改内容涉及与某个客户机相关的介质类型时，则这些更改内容将会传送给该客户机，同时会更新该客户机目录。

归档

Sun SAM-Remote 归档处理过程与 Sun StorageTek SAM 归档处理过程相同。Sun SAM-Remote 客户机发出的挂载请求将添加到服务器的挂载请求表中。然后，客户机等待服务器发出一则表示介质已挂载的消息。介质可用时，即会开始归档。

配置 Sun SAM-Remote 软件

本节介绍如何对 Sun SAM-Remote 服务器和客户机软件进行初始配置。包括以下几节内容：

- 第 155 页的“配置示例”
- 第 156 页的“配置软件”

配置示例

图 7-3 显示了本章所述过程使用的配置样例。本章中的示例说明了如何配置一个名为 `chicago` 的 Sun SAM-Remote 服务器。

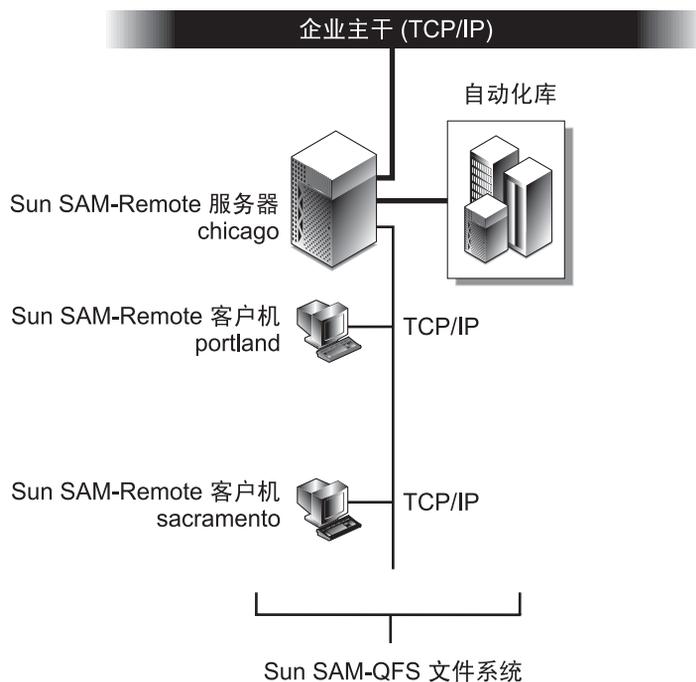


图 7-3 Sun SAM-Remote 配置示例

`portland` 和 `sacramento` 上的 Sun StorageTek SAM 文件系统将 `chicago` 用作它们的 Sun SAM-Remote 服务器。

配置软件

以下过程介绍了如何在 Sun SAM-Remote 服务器以及一台或多台 Sun SAM-Remote 客户机上配置 Sun SAM-Remote 软件。必须按以下各节所述的顺序执行这些过程：

1. 第 156 页的 “登录至潜在的服务器及客户机主机”
2. 第 157 页的 “验证客户机和服务器配置”
3. 第 158 页的 “编辑 mcf 文件”
4. 第 160 页的 “定义 Sun SAM-Remote 客户机”
5. 第 161 页的 “在服务器的 mcf 文件中定义 Sun SAM-Remote 服务器”
6. 第 162 页的 “创建 Sun SAM-Remote 服务器配置文件”
7. 第 164 页的 “启用归档”

▼ 登录至潜在的服务器及客户机主机

对潜在的服务器和所有潜在的客户机主机执行以下步骤。

1. 以超级用户身份登录至 **Sun SAM-Remote 服务器**。

对于您要在其中安装 Sun SAM-Remote 软件的服务器系统，您必须具有超级用户访问权限。

2. 以超级用户身份登录至 **Sun SAM-Remote 客户机**。

对于您要在其中安装 Sun SAM-Remote 软件的所有客户机系统，您必须具有超级用户访问权限。

▼ 验证客户机和服务器配置

为了确保您已安装了所需的软件版本，请在要配置为 Sun SAM-Remote 环境一部分的每个系统上执行以下步骤。

1. 运行带有 `-l` 选项的 `pkginfo(1M)` 命令，并检查其输出。

代码示例 7-1 显示了 `pkginfo(1M)` 命令的输出样例。

代码示例 7-1 使用 `pkginfo(1)`

```
chicago# 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)

chicago#
```

必须在要配置为 Sun SAM-Remote 环境一部分的所有客户机和服务器主机上，安装发行版本和修订版本相同的 Sun StorageTek SAM 软件。

根据代码示例 7-1 中的输出示例，您会发现服务器 (chicago) 运行的软件版本是 4U0.5，因此该服务器所在环境中的其他任何系统也必须运行 4U0.5 版本的软件。

2. 运行带有 `-p` 选项的 `showrev(1M)` 命令，并检查其输出。

代码示例 7-2 显示了 `showrev(1M)` 命令的输出样例。

代码示例 7-2 使用 `showrev(1M)`

```
chicago# showrev -p | grep SUNWsamfs
Patch: 113546-07 Obsoletes: Requires: Incompatibles: Packages:
SUNWsamfs
chicago#
```

必须在要配置为 Sun SAM-Remote 环境一部分的所有客户机和服务器主机上，安装相同的修补程序集。

根据代码示例 7-2 中的输出示例，您会发现服务器运行的修补程序版本是 113546-07，因此该服务器所在环境中的其他任何系统也必须运行 113546-07 版本的修补程序。

如果某些系统运行的是以前版本的软件或修补程序，则必须将所有系统升级到最新的软件版本。以代码示例 7-1 为例，如果您在系统中运行的 Sun StorageTek SAM 版本早于版本 4U0.5，则您必须将该系统至少升级至版本 4U0.5。

有关升级软件的信息，请参见《Sun StorageTek Storage Archive Manager 安装和升级指南》。

▼ 编辑 mcf 文件

1. 在 Sun SAM-Remote 服务器上，停止 Sun StorageTek SAM 功能。

- a. 运行带有 `idle eq` 选项的 `samcmd(1M)` 命令，将受 Sun StorageTek SAM 软件控制的每个可移除介质驱动器置于空闲状态。

```
# samcmd idle eq
```

其中的 `eq`，用于指定所访问的可移除介质驱动器在 `mcf` 文件中定义的设备序号。

有关 `samcmd(1M)` 命令的更多信息，请参见 `samcmd(1M)` 手册页。

也可以使用 `samu(1M)` 操作员公用程序将驱动器置于空闲状态。有关如何使用 `samu(1M)` 操作员实用程序的信息，请参见《Sun StorageTek QFS 文件系统配置和管理指南》。

注 – 在运行 `samd stop` 命令之前，Sun StorageTek SAM 环境中的驱动器应该处于空闲状态。此命令的目的是使归档程序、登台程序和其他进程结束当前的任务。此外，您还可以卸载卡盒，并将它们放入各自的存储插槽内。

- b. 运行带有 `stop` 选项的 `samd(1M)` 命令，停止 `sam-amld` 守护进程及其子进程。

```
# samd stop
```

`samd(1M)` 命令安装在 `/opt/SUNWsamfs/sbin` 中。

2. 在每台客户机上，使用 vi(1) 或其他编辑器编辑现有的 Sun StorageTek SAM /etc/opt/SUNWsamfs/mcf 文件。

此步骤的目的是将该主机定义为 Sun SAM-Remote 客户机。代码示例 7-3 显示了客户机 portland 上已编辑完成的 mcf 文件。此 mcf 文件定义了文件系统，并显示 Sun SAM-Remote 客户机 portland 已定义成为 Sun SAM-Remote 服务器 chicago 的客户机。

在本章的示例中，您必须为客户机 sacramento 完成相同的配置过程。对于此系统，请编辑 mcf 文件，并将 portland 的 mcf 文件中的最后一组条目行复制到 sacramento 的 mcf 文件。这几行用于将 chicago 的主机定义为 Sun SAM-Remote 客户机。

代码示例 7-3 portland 上的 mcf 文件

```
# mcf file on portland
#
# Sun StorageTek 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
```

客户机上的 mcf 文件条目包括 Sun SAM-Remote 客户机的单行条目，以及您要配置的设备每个设备的伪设备条目。这些条目遵循 mcf(4) 手册页中定义的语法。

第一组条目定义 Sun StorageTek QFS 文件系统。

第二组条目将 Sun SAM-Remote 客户机 portland 定义成为 Sun SAM-Remote 服务器 chicago 的客户机。第一行定义 Sun SAM-Remote 服务器自身，如下所述。

- 设备标识符，由客户机配置文件的名称组成，稍后将在第 160 页的“定义 Sun SAM-Remote 客户机”中创建。
- 设备序号，由 1 到 65535 之间的唯一数字组成。
- 设备类型，由两个字母组成的助记符，表示 Sun SAM-Remote 客户机。

- 系列集，由要在该特定服务器上使用的守护进程的系列集名称组成。对于每台客户机，Sun SAM-Remote 服务器均可拥有一个服务器守护进程。
- 设备状态，处于打开或关闭状态。
- 其他参数，这里指由目录文件的路径组成的可选字段。

此 mcf 文件中的最后两个条目定义了 Sun SAM-Remote 伪设备，如 rd 设备类型所示。伪设备定义 Sun SAM-Remote 服务器上实际设备的网络连接。伪设备条目在系统重新引导时创建。您可以定义不限数量的伪设备。

▼ 定义 Sun SAM-Remote 客户机

Sun SAM-Remote 客户机的配置文件中包含一个单行条目，此单行条目由 Sun SAM-Remote 服务器的名称组成。如第 158 页的“编辑 mcf 文件”的步骤 2 中所示，此客户机配置文件的完整路径名在客户机的 mcf 文件中加以指定。

1. 在每台客户机上，使用 vi(1) 或其他编辑器打开 Sun SAM-Remote 客户机配置文件。

例如：

```
portland# vi /etc/opt/SUNWsamfs/rmt200
```

2. 在此文件中添加 Sun SAM-Remote 服务器的名称。

代码示例 7-4 显示了已为客户机 portland 编辑的配置文件。它指向名为 chicago 的 Sun SAM-Remote 服务器。

代码示例 7-4 客户机配置文件

```
portland# cat /etc/opt/SUNWsamfs/rmt200
chicago
```

▼ 在服务器的 mcf 文件中定义 Sun SAM-Remote 服务器

注 – 您必须在该 Sun SAM-Remote 服务器的 mcf 文件中至少配置一个 Sun StorageTek SAM 文件系统。

- 在 Sun SAM-Remote 服务器上，使用 vi(1) 或其他编辑器编辑现有的 Sun StorageTek SAM /etc/opt/SUNWsamfs/mcf 文件。

假定编辑服务器 chicago 上的 mcf 文件。编辑后的文件中定义了一个 Sun StorageTek QFS 文件系统，并将 chicago 定义为 Sun SAM-Remote 服务器。

代码示例 7-5 显示了 chicago 上的 mcf 文件。

代码示例 7-5 chicago 上的 mcf 文件

```
# 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
```

有关条目语法的介绍，请参见第 158 页的“编辑 mcf 文件”。

▼ 创建 Sun SAM-Remote 服务器配置文件

Sun SAM-Remote 服务器配置文件用于定义每一个客户机将要使用的磁盘缓冲区特性和介质。每个服务器守护进程可以配置十台客户机。如果您想支持更多客户机，则必须配置其他 Sun SAM-Remote 服务器守护进程，如前面第 158 页的“编辑 mcf 文件”（步骤 2）中及第 160 页的“定义 Sun SAM-Remote 客户机”中所述。

代码示例 7-6 显示了服务器配置文件示例，`/etc/opt/SUNWsamfs/rmt200`。此文件位于 Sun SAM-Remote 服务器 `chicago` 上，其中定义了客户机 `portland` 和 `sacramento`。

1. 在服务器上，使用 `vi(1)` 或其他编辑器打开 Sun SAM-Remote 服务器配置文件。
2. 编写服务器配置文件。

如代码示例 7-6 所示，服务器配置文件中包含每个客户机的多行条目。井符号 (#) 表示注释行。注释行右侧的所有内容均被忽略。

代码示例 7-6 服务器配置文件 `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
```

代码示例 7-7 显示了 Sun SAM-Remote 服务器配置文件的格式。

代码示例 7-7 服务器配置文件格式

```
client-name
    [ parameter1 ]
media
    eq media-type regex
    [ eq media-type regex ]
    [ . . . ]
endmedia
```

a. 编写 *client-name* 字段。

client-name 定义调用 Sun SAM-Remote 守护进程所服务的每个客户机的网络名称。*client-name* 中的第一个字符必须是该行中的首字符。*client-name* 可以被指定为网络名称、IP 地址或全限定域名。

client-name 之后的 *parameter* (如果指定) 和介质规范 (直至下一个客户机定义之前), 均专用于该客户机。*parameter* 和 *media* 定义必须缩进一个空格或制表符。

b. (可选) 编写 *parameter* 字段。

参数行采用 *keyword = value* 赋值对表示。您可以使用 *parameter* 字段指定网络块大小。*net_block_size* 参数指定此客户机的套接字所用的网络块大小 (以 KB 为单位)。此参数的格式如下:

```
net_blk_size=size
```

其中的 *size*, 用于指定 $4 \leq size \leq 64$ 范围内的一个整数。默认值为 4, 即 4096 字节。*parameter* 行必须缩进一个空格或制表符。

c. 编写 *media* 和 *endmedia* 关键字字段。

media 和 *endmedia* 关键字是服务器配置文件中必需的关键字。它们定义客户机可以使用的介质归档卷。这些介质关联按以下方式指定:

代码示例 7-8 服务器配置文件中的介质规范

```
media
    eq media-type (regex)
    [ eq media-type (regex) ]
    [ . . . ]
endmedia
```

media 和 *endmedia* 关键字界定了 Sun SAM-Remote 服务器配置文件的介质定义区域。*eq media-type regex* 行是介质定义行。*media* 定义必须缩进一个空格或制表符。*regex* 数据必须置于圆括号内。

介质类型规范包括以下几个要素：

参数	定义
<i>eq</i>	库的设备序号。 具有不同介质的网络连接库可以拥有多个 <i>eq media-type regex</i> 行，以便为每一种介质类型指定不同的 <i>eq media-type regex</i> 行。
<i>media-type</i>	由两个字符表示的特定介质类型，例如 <i>lt</i> 。一般介质类型规范在 <i>mcf</i> 文件中有效，但不适用于 <i>media-type</i> 规范。有关有效介质类型的信息，请参见 <i>mcf(4)</i> 手册页。 如果您的网络连接库拥有多个介质类型，请指定多个介质定义行。 例如，以下是有效的介质类型定义： <pre>media 100 lt (VSN1) 100 lt (VSN2) endmedia</pre>
<i>regex</i>	用于归档文件的卡盒的卷序列名 (volume serial name, VSN)。每一个指定的 VSN 必须表示为扩展的正则表达式，并且 VSN 必须置于圆括号内。有关扩展正则表达式的信息，请参见 <i>egrep(1)</i> 手册页。 有关正则表达式的信息，请参见 <i>regcomp(3C)</i> 手册页。

注 – 请勿让多个客户机使用同一个物理介质卡盒。此外，如果 Sun SAM-Remote 服务器在 Sun SAM-Remote 环境之外拥有自身的文件系统，则不建议让客户机和服务器共用介质卡盒。

▼ 启用归档

以下步骤用于启用归档功能并完成配置过程。

1. 验证客户机上的 *archiver.cmd* 文件。

您可能需要执行以下任务，具体取决于您的配置：

- 确保服务器配置文件中定义的 VSN 已指定给 *archiver.cmd* 文件中的正确归档集。
- 如果以下指令所应用的归档集将归档至 Sun SAM-Remote 服务器连接的库，则应从 Sun SAM-Remote 客户机上的 *archiver.cmd* 文件中删除这些指令：
 - `-tapenonstop`
 - `-offline_copy direct`

2. 运行带有 start 选项的 `samd(1M)` 命令，启动服务器和客户机上的 Sun StorageTek SAM 进程。

为确保系统读取服务器和客户机上的新配置文件，您必须启动或重新启动 Sun StorageTek SAM 软件。

在客户机和服务器上输入以下命令：

```
server# samd start
```

有关启动及重新启动 Sun StorageTek SAM 更多的完整指导，请参见《Sun StorageTek Storage Archive Manager 安装和升级指南》。

3. 在服务器和客户机上调用 `samu(1M)`。

此步骤的目的是验证主机之间的连接情况。使用 `samu(1M)` 实用程序的 `s` 和 `R` 显示屏幕显示 Sun SAM-Remote 的连接状态。有关 `samu(1M)` 的更多信息，请参见 `samu(1M)` 手册页或《Sun StorageTek QFS 文件系统配置和管理指南》。

代码示例 7-9 显示了 Sun SAM-Remote 客户机 portland 上的 `samu(1M)` 状态 `s` 显示屏幕。请注意，设备类型 `sc` 表示 Sun SAM-Remote 客户机。该行以下的消息表示已建立与服务器 `chicago` 的连接。

代码示例 7-9 客户机 `samu(1M)` `s` 显示屏幕

```
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
      server chicago connected
rd    201 on     /dev/samrd/rd0       200  -----r
rd    202 on     /dev/samrd/rd1       200  -----r
hy    203 on     historian              203  -----
```

代码示例 7-10 显示了 Sun SAM-Remote 服务器 chicago 上的 samu(1M) 状态 s 显示屏幕。请注意，设备类型 ss 表示 Sun SAM-Remote 服务器。此显示屏幕上的信息表明该系统是一个 Sun SAM-Remote 服务器。

代码示例 7-10 chicago 上的服务器 samu(1M) s 显示屏幕

```
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 -----
```

代码示例 7-11 显示了 Sun SAM-Remote 服务器 chicago 上的 samu(1M) Sun SAM-Remote R 显示屏幕。

代码示例 7-11 chicago 上的服务器 samu(1M) R 显示屏幕

```
Remote server eq: 50          addr: 00001ca0 4.0.5 Wed May 02
14:55:37
message:

Client: portland
client index - 0
network block size - 4096
max file size - 0           flags - c0000000
min file size - 8
```

如果您具有多个 Sun SAM-Remote 客户机，则可以按 CONTROL-f 组合键来滚动查看客户机。

在代码示例 7-11 中，连接的客户机名为 portland。“client index” 字段表示此客户机的索引号为 0（为此服务器守护进程定义的客户机索引号可以是 0 到 9 之间的数字）。最大文件大小、最小文件大小及网络块大小的单位均为“字节”。“标志”用于表示连接的状态，如表 7-1 中所示：

表 7-1 samu(1M) R 显示屏幕标志

标志	含义
0x00000000	没有连接。
0xc0000000	已建立连接。

4. 从服务器中使用 samu(1M) 实用程序的 `v` 显示屏幕，以确保可以在客户机上使用目录。

您应该可以查看每台客户机可用的 Sun SAM-Remote 目录。在 samu(1M) 中，输入以下命令：

```
:v eq
```

其中的 `eq`，用于指定 Sun SAM-Remote 客户机守护进程的设备序号（与在 `mcf` 文件中定义的设备序号相同）。

代码示例 7-12 显示了 chicago 上的 samu(1M) 显示屏幕，表示 portland 可访问的卷。

代码示例 7-12 从 chicago 上查看时可用的卷

```
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. 在每台客户机上，运行带有 `-A` 选项的 archiver(1M) 命令。

本步骤用于验证是否可将文件从客户机归档到服务器。此命令允许从归档程序写入列表，包括服务器上的 VSN。有关此命令的信息，请参见 archiver(1M) 手册页。

如果文件未归档，请参见《Sun StorageTek Storage Archive Manager 故障排除指南》。

使用 Sun SAM-Remote 软件回收

本节介绍有关在 Sun SAM-Remote 环境中执行回收过程的信息。Sun Microsystems 建议您仅在出现这里所述的特殊情况时才有必要在 Sun SAM-Remote 环境中执行回收过程。

由于回收过程包括释放卡盒上的空间以便储存更多的数据，因此如果回收过程没有正确配置，则回收程序可能会破坏归档卡盒上的重要数据。

注 – Sun StorageTek SAM 软件不强制执行这些限制。

为了避免丢失数据，遵循以下限制非常重要：

- 在 Sun SAM-Remote 环境中使用回收程序之前，必须完全理解回收程序的每一个步骤。如果执行命令的顺序不正确，或在错误的系统上执行了回收命令，则可能导致无法挽回的数据丢失。确保在执行任何命令之前，已对该命令的操作进行了分析，如 `tplabel(1M)` 可能会删除 Sun SAM-Remote 客户机或 Sun SAM-Remote 服务器上的数据。
- Sun SAM-Remote 服务器和 Sun SAM-Remote 客户机上的回收活动一定不能重叠。这样的操作可能会导致意外的重新标记卡盒以及无法挽回的数据丢失。
- 不要回收包含可移除介质文件的卡盒。
- 在 Sun SAM-Remote 客户机和服务器环境中，客户机和服务器并不清楚对方的文件系统、数据文件以及 `inode` 文件。因此，服务器和客户机各自必须专用一组特定的卡盒，而不能相互使用对方的卡盒。

您可以通过在 Sun SAM-Remote 服务器的 `/etc/opt/SUNWsamfs/recycler.cmd` 文件中创建 `no_recycle` 列表，以防止意外回收 Sun SAM-Remote 客户机所用的 VSN。不过，对 `no_recycle` 列表中的卷使用 `chmed(1M)` 命令的 `+c` 选项时，应多加小心。使用此命令为某个卷设置回收标志 (`+c`) 时，此操作将覆盖 `/etc/opt/SUNWsamfs/recycler.cmd` 文件中的 `no_recycle` 列表。

- 请勿尝试在同一天既回收 Sun SAM-Remote 服务器上的卷，又回收 Sun SAM-Remote 客户机上的卷。

只有出现以下情况时，才可以在 Sun SAM-Remote 环境中执行回收过程：

- 系统中的每一个 VSN 由不同的客户机系统使用，或由服务器使用。任何 VSN 中的文件只能来自一个系统。
- Sun SAM-Remote 客户机只具有那些包含该客户机归档映像的 VSN 的目录条目。除这些 VSN 的目录条目之外，不具有其他任何 VSN 的目录条目。服务器配置文件中的介质定义行（即 `eq media-type regex` 行）中的 `regex` 值必须与客户机目录中指定的卷保持一致。此外，客户机目录中的 `regex` 规范不能指定相同的卷。
- 系统按归档集执行归档过程。使用 Sun SAM-Remote 时，必须按归档集（而不是库）执行回收过程。

以下几小节介绍了两种通过 Sun SAM-Remote 客户机和服务器启用回收过程的方法。方法如下：

- 第 169 页的“在 Sun SAM-Remote 环境中进行回收 - 方法 1”
- 第 192 页的“在 Sun SAM-Remote 环境中进行回收 - 方法 2”

在 Sun SAM-Remote 环境中进行回收 - 方法 1

本节中的过程介绍了一种在 Sun SAM-Remote 环境中启用回收过程的方法。在此环境中，服务器命名为 sky，客户机命名为 zeke。



注意 - 只有在严格执行本过程中的步骤，且您的配置经测试可以正确执行回收过程之后，才可以在 Sun SAM-Remote 环境中使用回收程序。

服务器 sky 的配置文件

服务器的 mcf 文件和服务器配置文件中必须具有 Sun SAM-Remote 的配置信息。以下代码示例显示了这些文件。

代码示例 7-13 显示了服务器 sky 上的 mcf 文件。

代码示例 7-13 服务器 sky 上的 mcf 文件

```
# 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
/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
```

代码示例 7-14 显示了服务器 sky 上的服务器配置文件。

代码示例 7-14 服务器 sky 上的服务器配置文件

```
# 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
  media
  70 at 00002[0-9]
  endmedia
```

客户机 zeke 的配置文件

客户机的 mcf 文件和客户机配置文件中必须具有 Sun SAM-Remote 的配置信息。以下代码示例显示了这些文件。

代码示例 7-15 显示了客户机 zeke 上的 mcf 文件。

代码示例 7-15 客户机 zeke 上的 mcf 文件

```
# 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
```

代码示例 7-16 显示了客户机 zeke 上的客户机配置文件。

代码示例 7-16 客户机 zeke 上的客户机配置文件

```
# cat /etc/opt/SUNWsamfs/sky
# File /etc/opt/SUNWsamfs/sky on Sun SAM-Remote client zeke:
sky
```

▼ 配置回收过程 - 方法 1

配置回收过程包括一个归档和回收的测试。由于测试周期各不相同，本过程可能要一天或两天才能完成，具体取决于文件归档和回收的频率。

在开始以下过程之前，请阅读第 6 章中有关回收程序的信息。在 Sun SAM-Remote 环境中运行回收程序时，您必须完全理解回收过程的每一步骤。如果您不熟悉回收过程，请抽出一段时间进行学习。

注 - 请勿在服务器上使用 `chmed(1M)` 命令为客户机的 VSN 设置回收标志 (+c)。该操作会改写服务器上 `/etc/opt/SUNWsamfs/recycler.cmd` 文件中的 `no_recycle` 列表。

1. 确保 Sun SAM-Remote 客户机和服务器配置正确，且可以进行归档。

有关配置和验证 Sun SAM-Remote 环境的更多信息，请参见第 155 页的“配置 Sun SAM-Remote 软件”，该节提供了有关配置 Sun SAM-Remote 客户机和服务器的详细信息。本过程包括用于确保归档的步骤。

2. 打开客户机系统上的 `archiver.cmd` 文件，添加回收指令。

本示例中，将按归档集（而不是库）来执行回收过程。因此，指定按归档集进行回收的指令必须出现在 `archiver.cmd` 文件中。

代码示例 7-17 显示了客户机 zeke 上的 archiver.cmd 文件。为了与回收程序进行通信，此文件已经过编辑。

代码示例 7-17 客户机 zeke 上的 archiver.cmd 文件

```
# 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 = 1t 2G
archmax = at 5G
drives = skyr5 4 # use up to four drives for remote archiving.
fs = samfs1
    1 4h
archiveset testdir0
    1 1m
    2 1m
defaultset .
    1 1m
    2 1m

params
# 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
```

代码示例 7-17 客户机 zeke 上的 archiver.cmd 文件 (续)

```
# 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
```

代码示例 7-17 中显示的指令用于执行以下操作：

- `-recycle_hwm` 指令用于设置归档集在库中的上限。当 VSN 的利用率超出此指令设置的百分比时，系统会开始回收归档集。
- `-recycle_ignore` 指令只是临时插入。此指令可在您配置并检测环境之前，防止进行回收过程。您可在以后的步骤中删除此指令。
- 将 `-recycle_mingain` 指令设置为高，以限制需要重新获取空间的工作的数量，从而保证工作效率。
- `-recycle_vsncount 1` 指令指定回收程序一次清空一个 VSN。清空第一个 VSN 之后，随后开始清空第二个。在任何指定时刻，队列中都有一个 VSN 被重新标记，一个 VSN 被清空。因此，回收过程不会造成系统崩溃。

3. 打开客户机上 `recycler.cmd` 文件，指定用于接收回收日志输出的日志文件。

代码示例 7-18 显示了客户机 zeke 上经过编辑的 `recycler.cmd` 文件，其中指定了回收程序日志文件。

代码示例 7-18 客户机 zeke 上的 `recycler.cmd` 文件

```
#
# This is the /etc/opt/SUNWsamfs/recycler.cmd file
# on client zeke.
#
logfile = /var/opt/SUNWsamfs/log/recycler
```

4. 验证服务器上的 archiver.cmd 文件是否指定按归档集执行回收过程。

使用 Sun SAM-Remote 时，必须指定按归档集（而不是库）执行回收过程。因此，指定按归档集进行回收的指令必须出现在 archiver.cmd 文件中。

代码示例 7-19 显示了服务器 sky 上的 archiver.cmd 文件。此文件指定按归档集执行归档过程。

代码示例 7-19 服务器 sky 上的 archiver.cmd 文件

```
# 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
    2 1m
params
```

代码示例 7-19 服务器 sky 上的 archiver.cmd 文件 (续)

```
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
```

5. 编辑服务器上的 recycler.cmd 文件。

修改此文件以指定以下各项：

- 用于接收回收程序输出的回收程序日志文件。
- 用于保护 Sun SAM-Remote 客户机所用 VSN 的 no_recycle 指令。根据配置要求，Sun SAM-Remote 客户机将其归档副本 2 写入至 Sun SAM-Remote 服务器库中的卡盒。您需要输入 no_recycle 指令，以防止 Sun SAM-Remote 服务器回收 Sun SAM-Remote 客户机用以归档的 VSN。

代码示例 7-20 显示了服务器 sky 上经过编辑的 recycler.cmd 文件，其中指定了回收程序日志文件。

代码示例 7-20 服务器 sky 上的 recycler.cmd 文件

```
#
# This is the /etc/opt/SUNWsamfs/recycler.cmd file
# on Sun SAM-Remote server sky.
#
logfile = /var/opt/SUNWsamfs/recycler/recycler.log
adic1 -ignore
no_recycle at 00002[0-9] # Prevents VSNs assigned to zeke from
                        # being recycled.
```

6. 在 Sun SAM-Remote 客户机上，使用 `sam-recycler(1M)` 命令测试回收程序。

在 Sun SAM-Remote 客户机系统上运行回收程序。此测试用于查看回收程序是否可以正确识别配置文件中指定的设备和 VSN。

例如，您可以使用以下命令初次测试回收程序的操作：

```
zeke# sam-recycler -dvx
```

此项测试非常重要，这是因为：如果回收程序检测到它所运行于的系统在其目录（包括历史记录目录）中列出的特定 VSN 上没有归档映像，`recycler.sh` 脚本可以调用要标记的卡盒。标记卡盒将会破坏卡盒上的所有数据。Sun SAM-Remote 客户机和 Sun StorageTek SAM 服务器之间不存在通信，无法告知对方是否具有归档副本。从本地的 Sun StorageTek SAM 文件系统在本本地提供所有的此类信息。

回收程序将会运行并将其活动记录至回收程序日志文件中。回收程序日志文件在 `recycler.cmd` 文件中定义。有关 `sam-recycler(1M)` 命令的更多信息，请参见 `sam-recycler(1M)` 手册页。

7. 检查回收程序日志文件。

您要查找的是以下消息：

```
Recycling is ignored on this archive set.
```

代码示例 7-21 显示了日志文件样例。

代码示例 7-21 客户机 zeke 上的回收程序日志文件

```
# 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
     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
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
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---      -----Percent-----      defaultset.1
-----Status-----      Count      Bytes      Use Obsolete Free      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---      -----Percent-----      defaultset.2
-----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
    
```

```

          ---Archives---      -----Percent-----      archiveset.1
-----Status-----      Count      Bytes      Use Obsolete Free      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---      -----Percent-----      archiveset.2
-----Status-----      Count      Bytes      Use Obsolete Free      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
    
```

代码示例 7-21 客户机 zeke 上的回收程序日志文件（续）

```
---Archives---      -----Percent-----      stk_l20
-----Status-----      Count      Bytes      Use Obsolete Free      Library:Type:VSN
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 =====
```

8. 在 Sun SAM-Remote 服务器上运行 sam-recycler(1M) 命令，验证回收程序没有回收为 Sun SAM-Remote 客户机保留的任何 VSN。

例如：

```
sky# sam-recycler -dvx
```

上述命令将会运行回收程序，并将其活动写入至回收程序日志文件中。有关 sam-recycler(1M) 命令的更多信息，请参见 sam-recycler(1M) 手册页。

代码示例 7-22 显示了回收程序日志文件样例。

代码示例 7-22 回收程序日志文件

```
# recycler.log file from server sky.
===== Recycler begins at Mon Jun  4 09:50:44 2001 =====
Initial 6 catalogs:
0 Family: adic1          Path: /var/opt/SUNWsamfs/catalog/adic1
  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
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.
  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.
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.
  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.
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:

```

-----Status-----	Count	Bytes	Use	Obsolete	Free	testset.1 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
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 =====

```

9. 分析服务器和客户机的 `recycler.log` 文件，以选择适于回收的 VSN。

`recycler.log` 文件的末尾有一个 Status 栏。客户机日志文件中具有以下状态类型条目的 VSN 是适于回收的 VSN：

- `no-data` VSN。要回收 `no-data` VSN，请参见第 188 页的“回收 `no-data` VSN”。
- `partially full`。要回收 `partially full` VSN，请参见第 190 页的“回收 `partially full` VSN”。

在服务器日志文件中，最适于回收的 VSN 是那些在 Count、Bytes 和 Use 栏中具有 0 值的 VSN。

▼ 回收 `no-data` VSN

`no-data` VSN 是最容易回收的 VSN。对于这些 VSN，其 Count、Bytes 和 Use 字段值均为 0（零）。

1. 检查客户机的 `recycler.log` 文件，确定是否有 `no-data` VSN。

可以考虑回收客户机 `zeke` 的 VSN 000029 和 000026，因为它们是 `no-data` VSN，如代码示例 7-23 中所示。

代码示例 7-23 客户机 `zeke` 上的 `recycler.log` 文件

```
# From the client zeke recycler.log file:
-----Status-----      ---Archives---      -----Percent-----      defaultset.2
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. 检查服务器的 `recycler.log` 文件，确定您在上一步骤中选择的 VSN 是否同样出现在服务器的回收程序日志文件中。

您希望验证来自服务器的有效数据是否没有归档在这些 VSN 上。

代码示例 7-24 显示了服务器 `recycler.log` 文件中 `no_recycle` VSN 的数据。对于 VSN 000029 和 000026，服务器的 `recycler.log` 文件中的数据与客户机的 `recycler.log` 文件中的数据是相同的。

代码示例 7-24 服务器 sky 上的 `recycler.log` 文件

```
# From the Server log file:
      ---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
.
.
.
```

3. 如果来自服务器的有效数据没有归档在选定的 VSN 上，请使用 `tplabel(1M)` 或 `odlabel(1M)` 命令重新标记该 VSN。

注 – 此操作将会破坏 VSN 上的所有数据并收回 VSN 的空间。

例如，对于磁带 VSN 000029，可以输入以下命令：

```
sky# tplabel -vsn 000029 -old 000029 at.000029
```

重新标记 VSN 000029 后，您将重新得到该 VSN 上 100% 的空间。

如果介质是磁光盘，则应使用 `odlabel(1M)` 命令。有关 `odlabel(1M)` 命令的更多信息，请参见 `odlabel(1M)` 手册页。

▼ 回收 partially full VSN

您也可回收那些具有 partially full 状态的 VSN。

1. 检查客户机的 recycler.log 文件，确定是否有 partially full VSN。

您可以考虑回收客户机 zeke 的 VSN 000025，因为其状态为 partially full，如代码示例 7-25 中所示。

代码示例 7-25 客户机 zeke 上的 recycler.log 文件

```
# From the client zeke recycler.log file:
-----Status-----      ---Archives---      -----Percent-----      defaultset.2
                          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
```

表中显示，VSN 000025 的 6% 空间在使用中。这些是回收此 VSN 之前必须重新归档的有效归档映像。本过程中的以下步骤介绍如何确保这些有效归档映像重新归档至另一个 VSN。

2. 检查服务器的 recycler.log 文件，确保来自服务器的有效数据没有归档在该 VSN 上。

服务器的 recycler.log 文件指出 VSN 000025 有 6% 的可用空间，这与客户机的 recycler.log 文件中报告的百分比相同。服务器并不清楚客户机的归档映像，因此服务器将报告所有剩余的 94% 空间由过时的归档映像占用。

代码示例 7-26 服务器 sky 上的 recycler.log 文件

```
# From the Server log file:
-----Status-----      ---Archives---      -----Percent-----      adic1
                          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
.
.
.
```

3. 在服务器上，使用带有 **+c** 选项的 **chmed(1M)** 命令重新归档 VSN 上的有效文件。

```
sky# chmed +c at.000025
```

有关 **chmed(1M)** 命令的更多信息，请参见 **chmed(1M)** 手册页。

4. 在客户机上，使用 **sam-recycler(1M)** 命令再次运行回收程序。

```
zeke# sam-recycler -dvx
```

此命令用于标记要重新归档至另一个 VSN 的每一个有效文件。

5. 启动归档程序。

您可以通过让归档程序正常运行，也可以通过从客户机上的 **samu(1M)** 实用程序中键入 **:arrun** 来启动归档程序。有关 **:arrun** 命令的更多信息，请参见 **samu(1M)** 手册页。

6. 归档完成后，运行 **sam-recycler(1M)** 命令，在客户机上重新运行回收程序。

```
zeke# sam-recycler -dvx
```

这可确保所有有效文件均已重新归档。

7. 如果 **Count**、**Bytes** 和 **Use** 字段值均为 0（零），请使用 **tplabel(1M)** 或 **odlabel(1M)** 命令从服务器重新标记该 VSN。

例如，对于磁带 VSN 000025，可以使用以下命令：

```
sky# tplabel -vsn 000025 -old 000025 at.000025
```

此命令重新标记 VSN 并破坏该 VSN 上的所有数据。该 VSN 被重新标记后，您可以得到该 VSN 的 88% 空间。

如果介质是磁光盘，则应使用 **odlabel(1M)** 命令。有关 **odlabel(1M)** 命令的更多信息，请参见 **odlabel(1M)** 手册页。

在 Sun SAM-Remote 环境中进行回收 - 方法 2

本节介绍使用 Sun SAM-remote 软件回收卷的另一种方法。



注意 - 只有在严格执行本过程中的步骤，且您的配置经测试可以正确执行回收过程之后，才可以在 Sun SAM-Remote 环境中使用回收程序。

▼ 配置回收过程 - 方法 2

1. 在 **Sun SAM-Remote** 客户机上运行 `sam-recycler(1M)` 命令，确定最适于回收的卷。
例如：

```
client# sam-recycler -dvx
```

2. 分析回收程序日志文件以回收备选文件。

`recycler.log` 文件的末尾有一个 `Status` 栏。客户机日志文件中具有以下状态类型条目的 `VSN` 是适于回收的 `VSN`：

- `no-data VSN`。要回收 `no-data VSN`，请参见第 188 页的“回收 `no-data VSN`”。
- `partially full`。要回收 `partially full VSN`，请参见第 190 页的“回收 `partially full VSN`”。

在服务器日志文件中，最适于回收的 `VSN` 是那些在 `Count`、`Bytes` 和 `Use` 栏中具有 0 值的 `VSN`。

3. 在 **Sun SAM-Remote** 服务器上，运行 `chmed(1M)` 命令，在选定的 `VSN` 上设置回收标志。

例如：

```
server# chmed +c at.00025
```

4. 在 Sun SAM-Remote 客户机上，运行 `sam-recycler(1M)` 命令，回收在 Sun SAM-Remote 客户机上选定的 VSN。

例如：

```
client# sam-recycler -dvx
```

5. 等到所回收的 VSN 完全清空了归档映像。
客户机上的归档程序执行此项操作。
6. 在 Sun SAM-Remote 服务器上，运行 `tplabel(1M)` 或 `odlabel(1M)` 命令，根据归档介质来重新标记卷。
7. 在 Sun SAM-Remote 服务器上，清除所有标志（例如 R 或 C），以免将卷用于在 Sun SAM-Remote 客户机上执行归档。

高级主题

本章介绍超出基本系统管理和使用范围的高级主题。

本章包括以下各节。

- 第 195 页的“在 Sun Cluster 环境中配置 SAM-QFS 文件系统 (HA-SAM)”
- 第 201 页的“使用设备日志”
- 第 203 页的“使用可移除介质文件”
- 第 205 页的“使用分段文件”
- 第 206 页的“使用系统错误工具报告”

在 Sun Cluster 环境中配置 SAM-QFS 文件系统 (HA-SAM)

高可用性 Sun StorageTek SAM (HA-SAM) 是在基于 SPARC® 和 x64 硬件的 Solaris™ (SPARC 支持 Solaris 9 和 10, x64 仅支持 Solaris 10) 上运行的 Sun Cluster 软件和 SAM-QFS 文件系统之间的接口。HA-SAM Sun Cluster 代理会定期监视主节点上 Sun StorageTek SAM 操作的运行状况。如果出现不可修复的问题, 它会将 Sun StorageTek SAM 归档和登台操作切换到运行状况良好的节点。主动/被动配置既支持自动故障转移, 也支持强制性故障转移。仅支持双节点主动/被动配置。

为使磁带归档和登台在故障转移之后继续进行, 磁带机必须对正在运行 HA-SAM 的群集中的所有节点上都可见, 但不应将它们配置为 Sun StorageTek SAM 共享驱动器。HA-SAM 还支持磁盘归档, 并且磁盘归档文件对群集中的所有节点都可见。

HA-SAM 依赖于 Sun StorageTek QFS Sun Cluster 代理, 并假定 Sun StorageTek QFS 文件系统由 Sun StorageTek QFS 代理进行挂载并管理。HA-SAM 要求将 Sun StorageTek SAM 目录和登台程序目录从标准位置链接到 HAStoragePlus 文件系统中的一个目录。HA-SAM 应该是包含 Sun StorageTek QFS 和目录资源的资源组中的资源。

要配置 HA-SAM Sun Cluster 代理，必须已经配置 Sun StorageTek QFS Sun Cluster 代理。以下说明假定您已按照《Sun StorageTek QFS 安装和升级指南》和《Sun StorageTek QFS 文件系统配置和管理指南》中所述，在使用共享 Sun StorageTek QFS 文件系统的 Sun Cluster 环境中配置了 Sun StorageTek QFS。

注 – HA-SAM 仅可用于 Sun StorageTek SAM 和 Sun StorageTek QFS 软件 4U6 版本及更高版本。

开始之前

配置此功能之前应注意以下要求和限制。

- 要配置 HA-SAM 的各节点应运行相同版本的 Solaris（9 或 10）和 Sun Cluster（3.1u4、3.2 或更新版本）。不支持混用不同版本的 Solaris 或 Sun Cluster 软件。仅支持 Solaris 操作系统。如果要将 3.1u4 版的 Sun Cluster 软件与 HA-SAM 配合使用，请务必应用 Sun Cluster 软件的最新修补程序。
- 运行 HA-SAM 的群集中的节点必须具有相同类型的体系结构：SPARC 或 x64。不支持混用不同的体系结构。
- 最多只能配置两个节点。
- 仅支持主动/被动配置，不支持主动/主动配置。
- 仅在 HA-SAM 文件系统的主动节点上支持对 HA-SAM 文件系统的主动 I/O。
- 仅支持共享的 Sun StorageTek QFS 文件系统。同时支持 ma 和 ms 类型的文件系统。不支持独立的 Sun StorageTek QFS 环境。
- 此配置不支持任何软件卷管理器。
- 在 HA-SAM 环境中，必须为任何非 HA-SAM Sun StorageTek QFS 文件系统指定 nosam 挂载选项。
- 必须在同一资源组中配置 HA-SAM 资源、QFS 文件系统和 HAStoragePlus 文件系统。必须为非 HA-SAM 文件系统另外创建 Sun Cluster 资源组。
- 使用 SUNW.hasam 资源类型时，不能在 /etc/vfstab 文件中指定 bg 挂载选项。
- 需要光纤磁带机。磁带机必须通过光纤网络对所有系统可见，但不应配置为 SAM-QFS 共享驱动器。
- 用于磁盘归档的磁盘卷必须对所有节点都可见。
- 不能将活动的元数据服务器和潜在的元数据服务器配置为 SAM-Remote 客户机或服务
- Sun StorageTek SAM 目录和登台程序目录必须位于默认位置：
/var/opt/SUNWsamfs/。如果位于其他任何位置，则群集节点将无法找到它。
- 在配置 HA-SAM 之前，请检验群集中所需节点上的所有 Sun StorageTek SAM 操作均正常工作。
- 仅支持高可用性 (highly available, HA) 代理，不支持可伸缩代理。

- 此配置不支持 Oracle 软件，但支持将 HA-SAM 代理作为 Oracle 备份的 RMAN 目标。
- 自动故障转移会在五分钟之后强制执行故障转移，以避免其他潜在的 HA 代理出现问题。
- 在自动故障转移过程中，活动的归档和登台操作将在 5 分钟之内超时并终止。
- 在故障转移过程中，为了避免回收程序可能造成数据丢失的复杂情况，必须对所有 SAM-QFS 文件系统进行故障转移。
- 支持 Sun StorageTek 网络连接磁带库，但不支持 ADIC 或 Sony 8400 PetaSite 系列自动化磁带库。
- 不能通过 File System Manager 浏览器界面管理 HA-SAM 环境。
- 在强制性故障转移过程中，可能会在 `/var/adm/messages` 日志文件中显示有关空闲磁带机的错误消息。可以放心地忽略这些消息。它们仅表明在故障转移之前，HA-SAM 没有时间将磁带机置于空闲状态。
- 故障转移之后，目录中的卷可能会标有 E 标志。当软件无法在磁带的结尾处写入正确标签时，可能会出现这种情况。要清除此错误，请参见第 20 页的“清除介质错误”。

配置说明

在配置 HA-SAM 之前，必须使用 Sun Cluster `scrgadm` 命令配置以下扩展属性：

- `QFSName` – 提供以逗号分隔的共享 QFS 系列集名称列表
- `CatalogFileSystem` – 指定在 `/etc/vfstab` 中指定的群集 HAStoragePlus 文件系统

注 – 有关使用 `scrgadm` 命令的更多详细信息，请参见 Sun Cluster 管理指南。

▼ 验证 SUNwhasam 资源类型注册文件

- 使用以下命令验证 SUNwhasam 资源类型注册 (Resource Type Registration, RTR) 文件的链接是否正确：

```
# ls -l /usr/cluster/lib/rgm/rtreg/SUNW.hasam
lrwxrwxrwx 1 root  other  32 Feb 26 10:59
/usr/cluster/lib/rgm/rtreg/SUNW.hasam ->
/opt/SUNWsamfs/sc/etc/SUNW.hasam
```

▼ 确认目录和登台程序符号链接

为了实现高可用性配置，SAM-QFS 目录和登台程序队列必须位于 HAStoragePlus 文件系统中。在软件安装过程中，应已在 `/var/opt/SUNWsamfs` 中创建指向共享 HAStoragePlus 文件系统的符号链接。必须先确认这一点，然后再继续执行配置过程。

- 验证 `/var/opt/SUNWsamfs/catalog` 和 `/var/opt/SUNWsamfs/stager` 中是否包含指向 HAStoragePlus 挂载点的符号链接。

- a. 查看目录和登台程序文件。

```
# ls -l /var/opt/SUNWsamfs/catalog /var/opt/SUNWsamfs/stager
```

- b. 验证输出中显示的符号链接是否类似于以下示例。

```
lrwxrwxrwx 1 root  other    19 Sep 30 11:05
/var/opt/SUNWsamfs/catalog -> /sam_shared/catalog
lrwxrwxrwx 1 root  other    18 Sep 30 11:05
/var/opt/SUNWsamfs/stager -> /sam_shared/stager
```

如果符号链接存在，请继续执行第 199 页的“在 Sun Cluster 环境中配置 Sun StorageTek SAM”。如果链接不存在，请继续执行步骤 c。

- c. 将所有的现有 Sun StorageTek SAM 目录信息复制到一个临时位置。
- d. 从其当前位置删除现有的 `catalog` 和 `stager` 文件。

```
# rm -rf /var/opt/SUNWsamfs/catalog /var/opt/SUNWsamfs/stager
```

- e. 创建指向 HAStoragePlus 挂载点的 `/var/opt/SUNWsamfs/catalog` 和 `/var/opt/SUNWsamfs/stager` 符号链接。

以下是在 `/sam_shared` 中挂载的 HAStoragePlus 文件系统示例。

```
# ln -s /sam_shared/catalog /var/opt/SUNWsamfs/catalog
# ln -s /sam_shared/stager /var/opt/SUNWsamfs/stager
```

- f. 将在步骤 c 中保存的目录信息复制到新的共享目录。

```
# cp -rp /var/tmp/catalog/* /var/opt/SUNWsamfs/catalog
```

▼ 在 Sun Cluster 环境中配置 Sun StorageTek SAM

在本过程示例中，配置了两个文件系统：qfs1 和 qfs2。

1. 创建 HA-SAM 资源组。

```
# scrgadm -a -g SAM-HA -h scnode-A,scnode-B
```

2. 创建并配置 SUNW.qfs 资源。

```
# scrgadm -a -g SAM-HA -t SUNW.qfs -j qfs-res \  
-x QFSFileSystem=/global/qfs1,/global/qfs2
```

3. 创建并配置 Sun StorageTek SAM 共享资源。

```
# scrgadm -a -j sam-hastp -g SAM-HA -t SUNW.HAStoragePlus \  
-x FilesystemMountPoints=/sam_shared -x AffinityOn=TRUE
```

4. 创建 HA-SAM 资源类型。

```
# scrgadm -a -t SUNW.hasam -f /opt/SUNWsamfs/sc/etc/SUNW.hasam
```

5. 创建并配置 SUNW.hasam 资源。

```
# scrgadm -a -g SAM-HA -t SUNW.hasam -j sam-ha \  
-x QFSName=qfs1,qfs2 -x CatalogFileSystem=/sam_shared
```

6. 创建资源组中各资源之间的相关性。

```
# scrgadm -c -j qfs-res -y Resource_dependencies=sam-hastp  
# scrgadm -c -j sam-ha -y Resource_dependencies=qfs-res
```

7. 使用 `scswitch(1M) -Z -g` 命令将资源组置于联机状态。

```
# scswitch -Z -g SAM-HA
```

注 - 必须先挂载 Sun StorageTek QFS 文件系统，然后再将资源组置于联机状态。

8. 确保此资源组在所有已配置节点上均可正常工作。

```
# scswitch -z -g SAM-HA -h scnode-B  
# scswitch -z -g SAM-HA -h scnode-A
```

使用 `samd hastop` 命令

`samd` 命令具有一个专用于 HA-SAM 的选项。`samd hastop` 命令用于在停止 `sam-amld` 及其子项之前，停止归档程序和登台程序守护进程。

在独立或共享的 Sun StorageTek SAM 环境中，管理员不能使用 `samd hastop` 命令。此外，不能对受 HA-SAM 控制的节点使用常规 `samd stop` 命令。

将 Sun StorageTek 库用于 HA-SAM

如果在 HA-SAM 环境中使用 Sun StorageTek 库，则需要通知 STK 守护进程在进行强制性故障转移时使用强制卸载而不是普通卸载。要实现此目的，请在将 HA-SAM 资源置于联机状态时，在活动的 Sun Cluster 节点上创建 `/var/run/hasam_running` 文件。这是一个具有 `root` 权限的零字节文件。如果在节点上检测到此文件，则 `sam-stkd` 和 `sam-stk_helper` 将使用 `force_media` 卸载磁带机上的磁带。这样会在强制性故障转移之后，成功地对库和磁带机进行初始化。在节点上将 SAM 资源置于脱机状态时，会删除 `hasam_running` 文件。

为了保证 Sun StorageTek 库的故障转移行为是正确的，请确保在发现 Sun Cluster 节点上具有处于联机状态的 HA-SAM 资源时未删除 `/var/run/hasam_running`。

使用设备日志

设备日志工具可以提供设备专用的错误信息，您可以使用这些信息来分析某些类型的设备问题。它可帮助您确定自动化库、磁带机或光盘驱动器发生故障事件的顺序。设备日志工具并不能收集软介质错误（如可恢复的读取错误）。

系统将设备日志消息分别写入至各个不同的日志文件。每一个自动化库、磁带和光盘驱动器设备均有一个日志文件，并且历史记录也有一个日志文件。日志文件位于 `/var/opt/SUNWsamfs/devlog` 目录下。每个日志文件的名称与设备序号的名称相对应。

例如，假设您的环境中具有 Sun StorageTek SAM 文件系统和一个配有两个光盘驱动器的 Hewlett Packard 光盘库。

代码示例 8-1 显示了 mcf 文件。

代码示例 8-1 mcf 文件示例

```
/dev/samst/c1t5u0 40 hp hp40 - etc/opt/SUNWsamfs/hp40_cat
/dev/samst/c1t4u0 41 mo hp40 -
/dev/samst/c1t6u0 42 mo hp40 -
```

代码示例 8-2 显示了 `/var/opt/SUNWsamfs/devlog` 文件。

代码示例 8-2 devlog 文件

```
# pwd
/var/opt/SUNWsamfs/devlog
# ls
40      41      42      43
#
```

设备 43 是历史记录。

何时使用设备日志

设备日志可以方便地生成许多日志消息，特别适用于已打开所有设备的全部日志选项且具有大量设备活动的场合。设备日志设置最初设置为下面的默认值：

```
err retry syserr date
```

如果您怀疑 Sun StorageTek SAM 环境中配置的某个设备出现问题，则应为该设备启用额外的日志事件。另外，如果您的服务供应商建议您这样做，则也应启用设备日志。在这些情况下，请将事件设置为 `detail`。在极为特殊的情况下，您的服务供应商可能建议您将某个设备的事件设置为 `all`。这可以添加额外的日志信息。不过，一般而言，在系统运行时设置过多的日志并无益处，甚至是不可行的。

当运行 `samexplorer(1M)` 命令时，系统会自动收集设备日志信息。这样，作为问题分析活动的一部分，文件系统服务人员可以复查任何可能的设备错误信息。

启用设备日志

可按以下两种方法之一启用设备日志，具体说明请参见以下两个小节：

- [第 202 页的“使用 `samset\(1M\)` 命令启用设备日志”](#)
- [第 203 页的“通过编辑 `defaults.conf` 文件启用设备日志”](#)

▼ 使用 `samset(1M)` 命令启用设备日志

- 使用 `samset(1M)` 命令。

例如：

```
# samset devlog eq event
```

其中的 `eq`，用于指定要为其记录消息的设备的设备序号。

其中的 `event`，用于指定 `samset(1M)` 手册页中列出的一个或多个事件。如果要指定多个事件，请用空格进行分隔。

▼ 通过编辑 `defaults.conf` 文件启用设备日志

1. 成为超级用户。
2. 使用 `vi(1)` 或其他编辑器打开文件 `/etc/opt/SUNWsamfs/defaults.conf`。
3. 在 `defaults.conf` 文件中添加 `devlog` 指令。

```
devlog eq event
```

其中的 `eq`，用于指定要为其记录消息的设备的设备序号。

其中的 `event`，用于指定 `samset(1M)` 手册页中列出的一个或多个事件。如果要指定多个事件，请用空格进行分隔。

When a Sun StorageTek SAM 文件系统会在启动时，自动将每个可用设备的事件类型设置为 `default`。另外，还可以使用 `samset(1M)` 命令来确定每一个设备日志的当前设置。

4. 保存并关闭 `defaults.conf` 文件。
5. 使用 `samd(1M) config` 命令应用 `defaults.conf` 文件中的更改。

```
# samd config
```

使用可移除介质文件

您可以使用 `request(1)` 命令手动创建、写入和读取那些不使用磁盘高速缓存来缓冲数据的文件。采用这种方式创建的文件称为可移除介质文件。

注 – 使用 `request(1)` 命令将会忽略归档程序的典型功能。

与典型的 Sun StorageTek SAM 文件类似，可移除介质文件也具有权限、用户名、群组名和大小属性。但是，可移除介质文件的数据并不会驻留在磁盘高速缓存中。因此，您可以创建文件大小大于磁盘高速缓存空间的可移除介质文件，并将它们写入可移除介质卡盒。

对于您在 `request(1)` 命令中指定的文件，系统会为它在 `.inodes` 文件中创建索引节点条目。Sun StorageTek SAM 文件系统会从索引节点条目中读取信息。多个可移除介质文件可以驻留在同一个卷中。

如果可移除介质文件存储在多个卷中，则该文件称为**卷溢出文件**。卷溢出功能允许在多个卡盒的多个卷中存储单个大型文件。如果您有非常大的文件，文件大小超出了所选介质的容量，则卷溢出功能将非常有用。

您必须依次读取和写入可移除介质文件。如果请求的卷驻留在 `mcf` 文件所定义的自动化库中，则 **Sun StorageTek SAM** 文件系统会自动挂载该卷。

如果卷中存在可移除介质文件，则回收程序不能对该卷进行回收。回收程序希望只有已归档的文件驻留在指定用于归档的特定卷中。此外，可移除介质文件也永远不会被归档。NFS 不支持可移除介质文件。

▼ 创建可移除介质或卷溢出文件

1. 使用 `tplabel(1M)` 或 `odlabel(1M)` 命令分别标记磁带或磁光盘卡盒。

有关这些命令的信息，请参见各自的手册页。

2. 使用 `request(1M)` 命令。

而且该命令至少应该使用以下选项：

```
request -m media-type -v vsn [vsn/vsn ...] [-l vsn-file] input-file
```

表 8-1 `request(1)` 命令的参数

参数	含义
<i>media-type</i>	可移除介质卡盒的介质类型。有关有效 <i>media-type</i> 规范的信息，请参见 <code>mcf(4)</code> 手册页。
<i>vsn</i>	可移除介质卡盒的卷序列名 (volume serial name, VSN)。如果指定了多个 VSN，则将创建卷溢出文件。您最多可以为卷溢出文件指定 256 个 VSN。应该使用正斜杠字符 (/) 来分隔 <i>vsn</i> 参数。所指定的 VSN 不应该是 Sun StorageTek SAM 环境中用于自动归档的卷。归档时，系统均会将下一个要归档的文件添加至当前数据的末尾，并将 EOF 标签移至数据的后面。
<i>vsn-file</i>	一个包含 VSN 列表的输入文件。如果您拥有许多 VSN，则在输入文件中指定 VSN 列表要比在命令行上指定更加便捷。
<i>input-file</i>	要写入至可移除介质卡盒的文件。该文件必须驻留在 Sun StorageTek SAM 文件系统中。

以下命令用于创建可移除介质文件：

```
# request -m lt -v aaa rem1
```

以下命令用于在三个卷中创建卷溢出文件：

```
# request -m lt -v TAPE01/TAPE02/TAPE03 large.file
```

有关如何创建可移除介质文件的详细示例，请参见 `request(1)` 手册页。

使用分段文件

Sun StorageTek SAM 环境支持分段文件。将文件分成多个段可以提高磁带存储检索速度、改善存取性能以及增强大文件的可管理性。分段文件的大小可以大于物理磁盘高速缓存。无论何时，分段文件都可能只有一部分驻留在磁盘高速缓存中。

可以使用 `segment(1)` 命令来指定分段大小。您所设置的分段大小不能小于当前文件的大小。

分段文件支持磁带分散读写功能。将文件分段后，可将文件同时分散读写至多个磁带设备中，这样大大缩短了存储各个文件段的时间。由于用户只需恢复所需的文件段（而不是整个文件），因此提高了数据访问速度。

由于只有发生更改的文件部分才需要重新归档，因此分段还可以提高归档效率。文件的各个分段可以并行归档，并且分段文件可以并行登台。这提高了系统的归档和恢复性能。

用户可以为文件、目录或整个文件系统启用分段功能。分段文件支持其他所有 Sun StorageTek SAM 功能。

注 – 无法对分段文件实现 `mmap` 功能。因此，分段文件不能是二进制可执行文件。

以下几节说明了分段文件与非分段文件之间的差异。有关分段文件的更多信息，请参见 `segment(1)` 或 `sam_segment(3)` 手册页。

归档

对于分段文件，归档单位是文件段自身，而不是整个文件。所有归档属性和优先级均应用于单个文件段，而不应用于整个文件。

您可以通过在 `archiver.cmd` 文件中为归档集指定 `-drives` 和 `-drivemin` 参数来分散读写某个段。例如，假定文件系统中有一个大小为 100 MB 的分段文件，其段大小为 10 MB。如果 `archiver.cmd` 文件中使用 `-drives 2` 指令定义归档集，则该文件将并行归档至两个驱动器。段 1、3、5、7 和 9 归档在第一个驱动器中，而段 2、4、6、8 和 10 归档在第二个驱动器中。

只归档修改过的分段。最多可以为每一个文件段创建四个归档副本。Sun StorageTek SAM 也支持对文件段使用卷溢出功能。

注 – 分段文件的索引不含用户数据。它被视为元数据并指定给文件系统归档集。

故障恢复

有关在出现故障时恢复分段文件的信息，请参见《Sun StorageTek Storage Archive Manager 故障排除指南》。

使用系统错误工具报告

系统错误工具 (system error facility, SEF) 报告系统用于捕获自动化库中的磁带设备生成的日志检测数据，然后将这些数据写入日志文件并转换成可读的格式。它包括以下项目：

- 日志文件，包含从磁带设备日志检测页收集的数据。
- `sefreport(1M)` 命令，以可读的格式将日志文件写入至 `stdout`。该日志文件可以作为用户分析脚本的输入项。

日志检测页因供应商而异。有关参数代码、控制位和参数值的含义，请参见每一个特定设备的供应商文档。

独立磁带机不支持 SEF。对于那些不支持 `tapealert(1M)` 功能的旧 SCSI-2 设备，SEF 报告功能非常有用。有关更多信息，请参见 `tapealert(1M)` 手册页。

▼ 启用 SEF 报告

1. 成为超级用户。
2. 使用 `mkdir(1)` 命令创建 SEF 目录。

例如：

```
# mkdir /var/opt/SUNWsamfs/sef
```

3. 使用 `touch(1)` 命令创建日志文件。

在安装报告系统后，您随时可以通过创建 `sefdata` 日志文件来启用 SEF 报告功能。起初，SEF 日志文件必须为空。

以下命令显示要在默认位置创建的 SEF 日志文件。

```
# touch /var/opt/SUNWsamfs/sef/sefdata
```

4. 使用 `samd(1M) stop` 和 `samd(1M) start` 初始化 SEF 报告功能。

```
# samd stop  
# samd start
```

生成的 SEF 数据将添加至日志文件的末尾。

注 – 只要 `sefdata` 日志文件存在，就会启用 SEF 报告功能。要禁用 SEF 报告功能，必须重命名或删除该文件。

您可以对 SEF 报告功能进行配置，以便在其他位置记录和读取日志检测数据。有关从其他位置读取日志检测文件的更多信息，请参见 `sefreport(1M)` 手册页。

▼ 生成 SEF 报告输出

使用 `sefreport(1M)` 命令之前，请确保 `/opt/SUNWsamfs/sbin` 位于您的命令路径中。SEF 报告输出的内容由标题行和日志检测数据组成。

● 使用 `sefreport(1M)` 命令生成 SEF 输出。

以下是 `sefreport(1M)` 命令的最常用选项：

- `-d` 选项。 `-d` 选项用于生成附加的设备信息。它将为每一条记录写入附加的包含设备序号和路径名的标题行。这便于用户搜索和查找与特定设备相关的 SEF 记录。
- `-v` 或 `-t` 选项。
 - `-v` 选项用于生成详细的信息。它将与设备序号、页码和 VSN 相关的信息添加至记录的每一行。这使用户可以只选择与特定设备或特定卷相关的行。
 - `-t` 用于生成包含文字说明的日志检测输出。对于日志检测数据输出的每一行，报告中都包含了一个额外的字符串，字符串的内容包括：设备序号、页码、VSN 和参数编码说明。

不要在同一命令行中指定 `-t` 和 `-v` 选项。它们是互相排斥的。

例如，下面的 SEF 命令从默认位置读取 SEF 日志文件，写入每一个设备的设备编号和路径名，并生成输出：

```
# sefreport -d /var/opt/SUNWsamfs/sef/sefdata > sef.output
```

代码示例 8-3 显示了 `sef.output` 文件的内容。

代码示例 8-3 `sef.output` 的内容

```
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
```

代码示例 8-3 sef.output 的内容 (续)

```

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

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

```

代码示例 8-3 sef.output 的内容 (续)

```
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
  param code  control  param value
    00h       74h      0x0
.
.
.
```

有关 SEF 日志文件的更多信息 (包括内容和格式), 请参见 `sefdata(4)` 手册页。有关可选 SEF 报告格式的更多信息, 请参见 `sefreport(1M)` 手册页。

管理 SEF 日志文件

SEF 日志文件的管理方式，与任何其他 Sun StorageTek SAM 日志文件相同。您可以定期运行 `cron(1)` 作业，以将当前日志文件保存至另一位置、删除旧 SEF 文件、创建新（空）的 SEF 文件或执行其他文件管理任务。

另外，用户还可使用 `log_rotate.sh(1M)` 实用程序来循环更新该日志文件。

有关用于管理 SEF 日志文件的工具的更多信息，请参见 `cron(1)` 或 `log_rotate.sh(1M)` 手册页。

SEF sysevent 功能

除了使用 SEF 日志文件之外，您还可以使用 Solaris sysevent 功能来获取磁带机 SCSI 日志检测错误计数器的第 2 页和第 3 页，以便了解介质分析信息。默认情况下，SEF sysevent 功能处于启用状态，并设置为在卸载之前轮询一次。SEF sysevent 的行为由 `defaults.conf` 和 `samset` 控制。

在 `defaults.conf` 文件中，`sef` 参数可用于按设备序号启用 SEF sysevent 功能，或者用于指定日志检测轮询频率。有关更多信息，请参见 `defaults.conf(4)` 手册页。

▼ 创建 SEF sysevent 处理器

1. 使用类似于以下的命令创建 `/var/tmp/xx` 文件：

```
#!/bin/ksh
echo "$@" >> /var/tmp/xx.dat
exit 0
```

2. 使 `/var/tmp/xx` 文件成为可执行文件：

```
# chmod a+rx /var/tmp/xx
```

3. 键入以下命令，将 SEF sysevent 处理器添加到 syseventd 文件：

```
# syseventadm add -vSUNW -pSUNWsamfs -cDevice -sSEF
/var/tmp/xx \"\$VENDOR\" \"\$PRODUCT\" \"\$USN\" \"\$REV\" \
$TOD \"$EQ_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
```

此命令将创建包含 SEF sysevent 处理器 /var/tmp/xx 的 /etc/sysevent/config/SUNW,SUNWsamfs,Device,sysevent.conf 文件，并将该事件处理器载入 syseventd 守护进程。

注 – Solaris 8 操作系统不支持 syseventadm(1M) 命令。您必须手动创建 /etc/sysevent/config/SUNW,SUNWsamfs,Device,sysevent.conf 文件及其内容，然后键入 `kill -HUP syseventd` 重新启动 syseventd 守护进程。

4. 要载入 SEF sysevent 处理器，请使用 `kill -HUP syseventd` 命令激活 /var/tmp/xx SEF sysevent 处理器。

有关 SEF sysevent 使用方法的更多信息，请参见 sefsysevent(4) 手册页。

具有供应商特定操作过程的自动化库的基本操作

Sun StorageTek SAM 环境中可以包含来自众多不同生产商的库。对于其中大多数库，您应该使用第 9 页的“使用自动化库和手动载入的驱动器”中所述的操作过程进行操作。但是，某些库需要执行供应商提供的特定操作过程，本章即介绍这些过程。

注 – Sun StorageTek SAM 软件可与许多生产商的自动化库兼容。有关自动化库的型号、固件级别以及其他兼容性方面的信息，请与 Sun 的销售代表或授权服务提供商联系。

本章将介绍以下自动化库：

- 第 214 页的“ADIC/Grau 自动化库”
- 第 215 页的“Fujitsu LMF 自动化库”
- 第 217 页的“IBM 3584 UltraScalable 磁带库”
- 第 218 页的“IBM 3494 库”
- 第 219 页的“Sony 8400 PetaSite 直接连接自动化库”
- 第 222 页的“Sony 网络连接自动化库”
- 第 224 页的“StorageTek ACSLS 连接自动化库”

ADIC/Grau 自动化库

如果您使用的是 ADIC/Grau 自动化库，请使用本节所述的过程导入和导出卡盒。这些过程不同于第 9 页的“使用自动化库和手动载入的驱动器”中所述的过程。

由于从物理上向 ADIC/Grau 自动化库中添加和从中取出卡盒时，使用的是供应商提供的实用程序，因此 Sun StorageTek SAM 接口（import(1M)、samexport(1M) 和 File System Manager）只能影响库目录。

注 – x64 硬件平台上的 Sun StorageTek SAM 软件不支持 ADIC/Grau 网络连接库。

▼ 导入卡盒

要导入卡盒，请执行以下步骤。

1. 使用 ADIC/Grau 命令将卡盒从物理上移入自动化库。
2. 使用 Sun StorageTek SAM import(1M) 命令更新库目录。

此命令的使用格式如下：

```
import -v volser eq
```

表 A-1 import(1M) 命令的参数

参数	含义
<i>volser</i>	要添加的 <i>volser</i> 。grauaci 接口在使用新条目更新库目录之前，将检验 ADIC/Grau 自动化库是否具有 <i>volser</i> 信息。
<i>eq</i>	所访问的设备在 <i>mcf</i> 文件中定义的设备序号。

▼ 导出卡盒

要导出卡盒，请执行以下步骤。

1. 使用 **Sun StorageTek SAM** `samexport(1M)` 命令从库目录中删除条目。
按以下某一种格式使用此命令：

```
samexport eq:slot  
samexport media-type.vsn
```

表 A-2 samexport(1M) 命令的参数

参数	含义
<i>eq</i>	所访问的设备在 <code>mcf</code> 文件中定义的设备序号。
<i>slot</i>	自动化库中存储插槽的编号，与库目录中标识的编号相同。
<i>media-type</i>	介质类型。有关有效介质类型的列表，请参见 <code>mcf(4)</code> 手册页。
<i>vsn</i>	指定给卷的卷序列名 (volume serial name, VSN)。

导出每一个 VSN 后，`samexport(1M)` 命令将更新库目录，并将每一个 VSN 的库目录条目从库目录移至历史记录。

2. 使用 **ADIC/Grau** 命令将卡盒从物理上移出自动化库。

Fujitsu LMF 自动化库

如果您使用的是 Fujitsu LMF 自动化库，请使用本节所述的过程导入和导出卡盒。这些过程不同于第 9 页的“使用自动化库和手动载入的驱动器”中所述的过程。

由于从物理上向 Fujitsu LMF 自动化库中添加或从中取出卡盒时，使用的是供应商提供的实用程序，因此 Sun StorageTek SAM 接口 (`import(1M)`、`samexport(1M)` 和 File System Manager) 只能影响库目录。

注 – x64 硬件平台上的 Sun StorageTek SAM 软件不支持 Fujitsu LMF 网络连接库。

▼ 导入卡盒

要导入卡盒，请执行以下步骤。

1. 使用 **Fujitsu** 命令将卡盒从物理上移入自动化库。
2. 使用 **Sun StorageTek SAM import(1M)** 命令更新库目录。

此命令的使用格式如下：

```
import -v volser eq
```

表 A-3 import(1M) 命令的参数

参数	含义
<i>volser</i>	要添加的 <i>volser</i> 。fujitsulmf 接口在使用新条目更新库目录之前，将检验 LMF 自动化库是否具有 <i>volser</i> 信息。
<i>eq</i>	所访问的设备在 <i>mcf</i> 文件中定义的设备序号。

▼ 导出卡盒

要导出卡盒，请执行以下步骤。

1. 使用 **Sun StorageTek SAM samexport(1M)** 命令从库目录中删除条目。

按以下某一种格式使用此命令：

```
samexport eq:slot  
samexport media-type.vsn
```

表 A-4 samexport(1M) 命令的参数

参数	含义
<i>eq</i>	所访问的设备在 <i>mcf</i> 文件中定义的设备序号。
<i>slot</i>	自动化库中存储插槽的编号，与库目录中标识的编号相同。
<i>media-type</i>	介质类型。有关有效介质类型的列表，请参见 <i>mcf(4)</i> 手册页。
<i>vsn</i>	指定给卷的卷序列名 (volume serial name, VSN)。

导出每一个 VSN 后，**samexport(1M)** 命令将更新库目录，并将每一个 VSN 的库目录条目从 Sun StorageTek SAM 库目录移至 Sun StorageTek SAM 历史记录。

2. 使用 **Fujitsu** 命令将卡盒从物理上移出自动化库。

IBM 3584 UltraScalable 磁带库

Sun StorageTek SAM 环境可支持 IBM 3584 UltraScalable 磁带库。以下几节从多方面介绍了此库的操作方法，这些操作不同于第 9 页的“使用自动化库和手动载入的驱动器”中所述的过程。

注 – x64 硬件平台上的 Sun StorageTek SAM 软件不支持 IBM 3584 UltraScalable 库。

导入卡盒

启动 Sun StorageTek SAM 软件时，邮箱中的卡盒不会自动导入。

清洁驱动器

要在 Sun StorageTek SAM 环境中使用此库，应该禁用自动清洁，而启用手动清洁。此过程在《IBM 3584 UltraScalable Tape Library Planning and Operator Guide》(IBM 的出版物 GA32-0408-01) 中进行了介绍。ibm3584(7) 手册页中也对此进行了介绍。

分区

此库可以包含多个磁带驱动器。如果您使用了多个驱动器，则可能需要将这—个物理库划分成两个、三个或四个逻辑库。如果您已将此库划分成两个或更多的逻辑库，则在将 IBM 3584 库添加到 Sun StorageTek SAM 环境中之前，应确保这些逻辑库可以正常工作。

从已分区的库中导出卡盒时，只有从中导出卡盒的逻辑库才可以访问该抽屉插槽。如果您手动取出卡盒并将其重新插入，则任何逻辑分区均可访问该卡盒。

▼ 取出卡盒

要从已分区的库中取出卡盒，请执行以下步骤。

1. 打开挡门。
2. 取出卡盒。
3. 关闭挡门。
4. 等待挡门锁定，然后取消锁定。
5. 打开挡门。
6. 放回卡盒。
7. 关闭挡门。

有关在 Sun StorageTek SAM 环境中将此库用作逻辑分区库的更多信息，请参见 IBM 的相关文档或 [ibm3584\(7\)](#) 手册页。

IBM 3494 库

Sun StorageTek SAM 环境可支持 IBM 3494 库。以下几节从多方面介绍了此库的操作方法，这些操作不同于第 9 页的“使用自动化库和手动载入的驱动器”中所述的过程。

注 - x64 硬件平台上的 Sun StorageTek SAM 软件不支持 IBM 3494 网络连接库。

▼ 导入卡盒

要导入卡盒，请执行以下步骤。

1. 将新的介质放入 I/O 插槽中。
2. 关闭挡门。

自动化库将锁定挡门，并介质移至存储区域。一次只能导入 100 个卷。

如果已将库配置为 `access=private`，则这是您需要执行的最后一个步骤。移动介质后这个库会通知守护进程，且介质会被添加到目录中。
3. 如果已将库配置为 `access=shared`，请运行 `import(1M)` 命令将介质添加到目录中。

▼ 导出卡盒

1. 使用 `export(1M)` 命令导出卡盒。
此命令会将介质移至 I/O 区域，并打开操作员面板上的输出模式指示灯。
2. 从物理上将介质从 I/O 区域中取出。

Sony 8400 PetaSite 直接连接自动化库

Sony 8400 PetaSite 系列自动化库不同于其他型号的 Sony 产品，因为它配有八插槽导入和导出邮箱（插槽编号为 400 至 407）。因此，该系统可以方便快捷地执行导入和导出操作。此自动化库使用条码读取器。

由于邮箱插槽可用作存储插槽，因此 Sun StorageTek SAM 库目录可以跟踪记录邮箱插槽。

注 – x64 硬件平台上的 Sun StorageTek SAM 软件不支持 Sony 8400 PetaSite 库。

注 – 本节中的信息仅适用于 Sony 8400 PetaSite 直接连接自动化库。这些信息既不适用于 Sony B9 和 B35 直接连接的自动化库，也不适用于第 222 页的“[Sony 网络连接自动化库](#)”。

▼ 导入磁带

要导入磁带，请执行以下步骤。

1. 按下自动化库前面板上的打开/关闭按钮，打开自动化库的挡门。
2. 将卡盒载入邮箱插槽。
3. 按下自动化库前面板上的打开/关闭按钮，手动关闭邮箱的挡门。
自动化库将在关闭挡门后检查邮箱插槽，以获取卡盒条码。如果条码有问题，该插槽的 in（进）和 out（出）指示灯均会闪烁。

4. 运行 `import(1M)` 命令，以使 Sun StorageTek SAM 系统识别导入的卡盒。

此命令的使用格式如下：

```
import eq
```

其中的 `eq`，用于指定所访问的设备在 `mcf` 文件中定义的设备序号。

另外，还可使用 File System Manager 来执行此步骤。有关更多信息，请参见 File System Manager 联机帮助。

导出磁带

导出磁带卡盒的步骤取决于您是否将邮箱插槽用作存储插槽。

▼ 在邮箱插槽未用作存储插槽时导出磁带

在邮箱插槽未用作存储插槽时，请使用以下步骤导出卡盒。

1. 运行 `move(1M)` 命令将卡盒移至邮箱插槽（插槽编号为 400 至 407）。

此命令的使用格式如下：

```
move source-slot destination-slot eq
```

表 A-5 `move(1M)` 命令的参数

参数	含义
<code>source-slot</code>	卡盒当前所在插槽的插槽编号。
<code>destination-slot</code>	卡盒将要移至的插槽编号。
<code>eq</code>	所访问的设备在 <code>mcf</code> 文件中定义的设备序号。

2. 按下自动化库前面板上的打开/关闭按钮。
此时，挡门会打开。
3. 从邮箱插槽中取出卡盒。
4. 按下自动化库前面板上的打开/关闭按钮，手动关闭邮箱的挡门。

5. 运行 `samexport(1M)` 命令，以使 Sun StorageTek SAM 系统识别导出的卡盒。
此命令的使用格式如下：

```
samexport eq
```

其中的 `eq`，用于指定所访问的设备在 `mcf` 文件中定义的设备序号。

另外，还可使用 File System Manager 来执行此步骤。有关更多信息，请参见 File System Manager 联机帮助。

▼ 在邮箱插槽用作存储插槽时导出磁带

如果将邮箱插槽用作存储插槽，并且要导出的卡盒位于其中一个邮箱插槽，则请使用以下步骤导出卡盒。

1. 按下自动化库前面板上的打开/关闭按钮。
此时，挡门会打开。
2. 从邮箱插槽中取出卡盒。
3. 按下自动化库前面板上的打开/关闭按钮，手动关闭邮箱的挡门。
4. 运行 `samexport(1M)` 命令，以使 Sun StorageTek SAM 系统识别导出的卡盒。
此命令的使用格式如下：

```
samexport eq
```

其中的 `eq`，用于指定所访问的设备在 `mcf` 文件中定义的设备序号。

另外，还可使用 File System Manager 来执行此步骤。有关更多信息，请参见 File System Manager 联机帮助。

▼ 将卡盒移至另一个插槽

要将卡盒移至另一个插槽，请执行以下步骤。

1. 确保来源插槽中具有卡盒，而目标插槽中没有卡盒。
2. 运行 `move(1M)` 命令。

此命令的使用格式如下：

```
move eq:source-slot destination-slot
```

表 A-6 `move(1M)` 命令的参数

参数	含义
<code>eq</code>	所访问的设备在 <code>mcf</code> 文件中定义的设备序号。
<code>source-slot</code>	卡盒当前所在插槽的插槽编号。
<code>destination-slot</code>	卡盒将要移至的插槽编号。

另外，还可使用 File System Manager 来执行此步骤。有关更多信息，请参见 File System Manager 联机帮助。

Sony 网络连接自动化库

如果您使用的是 Sony 网络连接自动化库，请使用本节所述的过程导入和导出卡盒。这些过程不同于第 9 页的“使用自动化库和手动载入的驱动器”中所述的过程。

由于从物理上向 Sony 自动化库中添加和从中取出卡盒时，所使用的是供应商提供的实用程序，因此 Sun StorageTek SAM 接口（`import(1M)`、`samexport(1M)` 和 File System Manager）只能影响库目录。

注 – x64 硬件平台上的 Sun StorageTek SAM 软件不支持 Sony 网络连接库。

▼ 导入卡盒

要导入卡盒，请执行以下步骤。

1. 使用 **Sony** 命令将卡盒从物理上移入自动化库。
2. 使用 **import(1M)** 命令更新库目录。

此命令的使用格式如下：

```
import -v [ " ] volser [ " ] eq
```

表 A-7 import(1M) 命令的参数

参数	含义
" "	引号。如果 <i>volser</i> 值包含空格，则必须包含在引号内。
<i>volser</i>	要添加的 <i>volser</i> 。PSC API 接口在使用新条目更新库目录之前，将检验 Sony 自动化库是否具有 <i>volser</i> 信息。如果卡盒从物理上并没有位于库中，则会在历史记录目录中添入一个条目。
<i>eq</i>	所访问的库在 <i>mcf</i> 文件中定义的设备序号。

▼ 导出卡盒

要导出卡盒，请执行以下步骤。

1. 使用 **samexport(1M)** 命令从库目录中删除条目。

按以下某一种格式使用此命令：

```
samexport eq:slot  
samexport media-type.vsn
```

表 A-8 samexport(1M) 命令的参数

参数	含义
<i>eq</i>	所访问的设备在 <i>mcf</i> 文件中定义的设备序号。
<i>slot</i>	自动化库中存储插槽的编号，与库目录中标识的编号相同。
<i>media-type</i>	介质类型。有关有效介质类型的列表，请参见 <i>mcf(4)</i> 手册页。
<i>vsn</i>	指定给卷的卷序列名 (volume serial name, VSN)。

导出每一个 VSN 后，**samexport(1M)** 命令将更新库目录，并将每一个 VSN 的库目录条目从库目录移至历史记录。

2. 使用 **Sony** 命令将卡盒从物理上移出自动化库。

StorageTek ACSLS 连接自动化库

如果您使用的是 StorageTek ACSLS 连接的自动化库，请使用本节所述的过程导入和导出卡盒。这些过程不同于第 9 页的“使用自动化库和手动载入的驱动器”中所述的过程。

邮箱是自动化库中的一个区域，它用于向自动化库中添加和从中取出卡盒。某些 StorageTek 自动化库一次只能导入和导出一个卡盒。StorageTek 9714 和 StorageTek 9710 就属于受 Sun StorageTek SAM 环境支持的带邮箱的 StorageTek 自动化库。StorageTek 9730 使用邮件插槽。在 StorageTek 文档中，通常将邮箱和邮箱插槽称为卡盒访问端口 (cartridge access port, CAP)。

在向 ACSLS 连接的自动化库中导入和从中导出卡盒时，请记住以下几点：

- 导入卡盒时，Sun StorageTek SAM 命令仅影响库目录。import(1M) 命令并不会从物理上将卡盒插入自动化库中。您必须使用 ACSLS 命令才能从物理上导入卡盒。
- 导出卡盒时，除非在 samexport(1M) 命令中使用 -f 选项，否则 Sun StorageTek SAM 命令仅影响库目录。使用 -f 选项后，Sun StorageTek SAM 系统会将卷放入 CAP 中，并相应地更新目录。如果不使用 -f 选项，则仅更新目录而不会将卷放入 CAP 中，因此您仍必须使用 ACSLS 命令从物理上导出卡盒。

按照协议，应该由您来负责维护 ACSLS 清单和 Sun StorageTek SAM 目录。

另外，还可使用 samu(1M) 或 File System Manager 来执行导入和导出过程。有关更多信息，请参见 File System Manager 联机帮助。

▼ 导入磁带

- 按以下格式使用 import(1M) 命令：

```
import -v vsn eq
```

表 A-9 import(1M) 命令的参数

参数	含义
<i>vsn</i>	指定给卷的卷序列名 (volume serial name, VSN)。
<i>eq</i>	所访问的设备在 mcf 文件中定义的设备序号。

import(1M) 命令将在库目录中添加一个新的 VSN。如果历史记录中包含此 VSN，则 Sun StorageTek SAM 软件会将 VSN 信息从历史记录移至库目录。

▼ 使用邮箱导出磁带

您既可以按插槽导出磁带卡盒，也可以按 VSN 导出磁带卡盒。

- 按以下格式之一使用 `samexport(1M)` 命令：

```
samexport [-f] eq:slot
samexport [-f] media-type.vsn
```

表 A-10 `samexport(1M)` 命令的参数

参数	含义
<code>-f</code>	Sun StorageTek SAM 系统用于将卷放入卡盒访问端口 (cartridge access port, CAP) 中，并相应地更新目录的规范。
<code>eq</code>	所访问的设备在 <code>mcf</code> 文件中定义的设备序号。
<code>slot</code>	自动化库中存储插槽的编号，与库目录中标识的编号相同。
<code>media-type</code>	介质类型。有关有效介质类型的列表，请参见 <code>mcf(4)</code> 手册页。
<code>vsn</code>	指定给卷的卷序列名 (volume serial name, VSN)。

导出每一个 VSN 后，`samexport(1M)` 命令将更新库目录，并将每一个 VSN 的库目录条目从库目录移至历史记录。

词汇表

B

**备份存储
(backup storage)**

一组文件的快照，旨在防止意外丢失数据。备份不仅包括文件的属性，而且还包括关联的数据。

**本地文件系统
(local file system)**

安装在 Sun Cluster 系统的某一个节点上的文件系统。它对于其他节点来说，可用性不高。此外，本地文件系统也指安装在服务器上的文件系统。

C

超级块 (superblock)

文件系统的一种数据结构，用于定义文件系统的基本参数。超级块将被写入存储系列集中的所有分区，并标识该系列集各个分区的成员。

**磁盘分配单元
(disk allocation unit)**

请参见 DAU。

**磁盘分散读写
(disk striping)**

跨多个磁盘记录同一文件的过程。该方法可提高存取性能，进而增加整体存储能力。另请参见分散读写 (striping)。

**磁盘高速缓存
(disk cache)**

文件系统软件的磁盘驻留部分，用于在联机磁盘高速缓存与归档介质之间创建并管理数据文件。单个磁盘分区或整个磁盘均可用作磁盘高速缓存。

**磁盘缓冲区
(disk buffer)**

在 Sun SAM-Remote 配置中，磁盘缓冲区是指服务器系统上用于将数据从客户机归档至服务器的缓冲区。

**磁盘空间阈值
(disk space threshold)**

由管理员定义的磁盘高速缓存的最大利用率或最小利用率。释放程序 (releaser) 可根据这些预定义的磁盘空间阈值来控制磁盘高速缓存的使用情况。

存储插槽 (storage slot)

自动化库中的位置，其中存储了卡盒（如果卡盒并未在驱动器中使用）。

**存储系列集
(storage family set)**

由一组磁盘组成，并由一个磁盘系列设备表示。

D

DAU

磁盘分配单元 (Disk allocation unit, DAU)。联机存储的基本单位。也称作块大小。

登台 (staging)

将近线文件或脱机文件从归档存储中恢复至联机存储的过程。

**多读取器文件系统
(Multireader file
system)**

一种具备单写入、多读取特点的文件系统，允许您指定挂载到多个主机上的文件系统。多个主机可读取该文件系统，但只有一个主机可向该文件系统写入数据。多个读取主机可通过 mount(1M) 命令的 -o reader 选项指定。一个写入主机可通过 mount(1M) 命令的 -o writer 选项指定。有关 mount(1M) 命令的更多信息，请参见 mount_samfs(1M) 手册页。

F

FDDI

光纤分布式数据接口 (Fiber-distributed data interface, FDDI) 是一种局域网数据传输标准，最大传输距离在 200 km (124 英里) 以内。FDDI 协议基于令牌环协议。

FTP

文件传输协议 (File transfer protocol)。一种用于通过 TCP/IP 网络在两个主机之间传输文件的 Internet 协议。

**范围阵列
(extent array)**

位于文件索引节点 (inode) 内的阵列，用于定义指定给该文件的每个数据块的磁盘位置。

分区 (partition)

设备的一部分或磁光盘卡盒的一面。

分散读写 (striping) 一种以交错方式将所有文件同时写入若干逻辑磁盘的数据存储方法。SAM-QFS 文件系统提供两种类型的分散读写。即“硬分散读写 (hard striping)” (使用分散读写组) 和“软分散读写 (soft striping)” (使用 `stripe=x` 挂载参数)。硬分散读写在设置文件系统时启用, 您需要在 `mcf` 文件中定义分散读写组。软分散读写则通过 `stripe=x` 挂载参数启用, 您可针对各个文件系统或单个文件更改它。通过设置 `stripe=0` 可禁用它。如果文件系统由多个具相同数量元素的分散读写组组成, 则可同时使用硬分散读写和软分散读写。另请参见**循环 (round-robin)**。

分散读写大小 (stripe size) 向分散读写的下一个设备写入数据前要分配的磁盘分配单元 (DAU) 数。如果使用 `stripe=0` 挂载选项, 文件系统将采用循环存取方式, 而不是分散读写存取方式。

分散读写组 (striped group) 文件系统中的一组设备, 在 `mcf` 文件中被定义为一个或多个 `gXXX` 设备。系统将分散读写组视作一个逻辑设备, 并始终按照磁盘分配单元 (DAU) 的大小进行分散读写。

G

共享主机文件 (shared hosts file) 创建共享文件系统时, 系统将主机文件中的信息复制到元数据服务器上的共享主机文件中。可通过运行 `samsharefs(1M) -u` 命令更新此信息。

挂载点 (mount point) 文件系统所挂载到的目录。

光纤通道 (Fibre Channel) ANSI 制定的标准, 用于指定设备之间的高速串行通信。光纤通道常被用作 SCSI-3 中的一种总线结构。

归档程序 (archiver) 一种可自动控制文件到可移除卡盒的复制操作的软件程序。

归档存储 (archive storage) 归档介质上已创建的文件数据的副本。

归档介质 (archive media) 归档文件写入到的介质。归档介质可以是库中的可移除磁带或磁光盘 (magneto-optical) 卡盒。此外, 归档介质也可以是另一系统上的某一挂载点。

H

回收程序 (recycler) 一种 Sun StorageTek SAM 实用程序，用于收回由过期归档副本占用的卡盒空间。

J

机械手 (robot) 自动化库的一部分，用于在存储插槽和驱动器之间移动卡盒。也称作**传输设备 (transport)**。

计时器 (timer) 一种配额软件，用于跟踪从用户达到软限制开始直到对该用户施加硬限制为止的时间段。

间接块 (indirect block) 包含一系列存储块的磁盘块。文件系统最多可有三级间接块。第一级间接块包含一系列用于数据存储的块。第二级间接块包含一系列第一级间接块。第三级间接块包含一系列第二级间接块。

介质 (media) 磁带或光盘卡盒。

介质回收 (media recycling) 对仅有很少归档文件的归档介质进行回收和再利用的过程。

近线存储 (nearline storage) 一种可移除介质存储。近线存储在访问之前需要自动挂载。近线存储通常比联机存储便宜，但所需的访问时间相对长一些。

镜像写入 (mirror writing) 在互不相连的磁盘组上维护文件的两份副本的过程，可防止单个磁盘损坏所导致的数据丢失。

卷 (volume) 卡盒上用于共享数据的命名区域。一个卡盒可以有一个或多个卷。双面卡盒有两个卷，每一面为一个卷。

卷溢出 (volume overflow) 一种允许跨多个卷保存单个文件的功能。对于使用超大型文件（超过了每个卡盒的容量）的站点，卷溢出功能非常有用。

K

- 卡盒 (cartridge)** 一种包含了记录数据的介质的物理实体，如磁带或光盘。有时称作**介质**或**卷**。
- 可寻址存储 (addressable storage)** 包括联机存储 (online)、近线存储 (nearline)、离站存储 (offsite) 和脱机存储 (offline) 等存储空间，用户可通过 Sun StorageTek QFS 或 Sun StorageTek SAM 文件系统访问这些空间。
- 可移除介质文件 (removable media file)** 一种特殊类型的用户文件，可直接从它所驻留的可移除介质卡盒（如磁带或光盘卡盒）中访问。此外，该文件也用于写入归档和登台 (stage) 文件数据。
- 客户机-服务器 (client-server)** 分布式系统中的交互模型。在该模型中，一个站点中的程序可向另一个站点上的程序发送请求并等待回应。发送请求的程序称作“**客户机 (client)**”。提供响应的程序称作“**服务器 (server)**”。
- 库 (library)** 请参见**自动化库 (automated library)**。
- 库目录 (library catalog)** 请参见**目录 (catalog)**。
- 块大小 (block size)** 请参见**DAU**。
- 块分配图 (block allocation map)** 一种显示磁盘上每个可用存储块的位图。该位图可指出每个块的状态：是在使用中还是空闲。
- 宽限期 (grace period)** 对于磁盘配额而言，宽限期是指达到软限制之后，系统允许用户继续创建文件并分配存储空间的时间。

L

- LAN** 局域网 (Local area network)。
- LUN** 逻辑单元号 (Logical unit number)。
- 离站存储 (offsite storage)** 远离服务器的存储，用于灾难恢复。
- 联机存储 (online storage)** 可即时访问的存储，如磁盘高速缓存。
- 连接 (connection)** 建立在两个协议模块之间的通道，可提供稳定可靠的数据流传输服务。TCP 连接就是一台计算机上的 TCP 模块到另一台计算机上的 TCP 模块的连接。

M

mcf 主配置文件 (master configuration file)。初始化期间读取的文件，用于定义文件系统环境中各个设备（拓扑结构）之间的关系。

名称空间 (name space) 一组文件的元数据部分，用于标识文件、文件属性和存储位置。

目录 (catalog) 自动化库中的 VSN 记录。每个自动化库都有一个目录，而且一个站点有一个记录所有自动化库的历史记录。另请参见 VSN。

目录 (directory) 一种指向文件系统中其他文件和目录的文件数据结构。

N

NFS 网络文件系统 (Network file system)。一种由 Sun 发布的文件系统，可对异构网络上的远程文件系统进行透明访问。

NIS Sun OS 4.0（最低）网络信息服务 (Network Information Service)。一种分布式网络数据库，包含网络上系统和用户的相关重要信息。NIS 数据库存储在主服务器和所有从属服务器上。

内核 (kernel) 用于提供基本系统功能的中央控制程序。UNIX 内核可创建并管理各个进程，并提供不同功能以访问文件系统。此外，UNIX 内核还可提供常规安全性以及通信功能。

P

配额 (quota) 允许用户使用的系统资源量。

Q

驱动器 (drive) 一种向可移除介质卷传入数据或从中传出数据的机械装置。

全局指令 (global directive) 应用于所有文件系统的归档程序指令和释放程序指令。第一个 fs = 行之前显示的都是全局指令。

R

- RAID** 独立磁盘冗余阵列 (Redundant array of independent disks)。一种使用若干独立磁盘来可靠地存储文件的磁盘技术。该技术可在单个磁盘出现故障时防止数据丢失，并可提供容错磁盘环境以及比单个磁盘更高的吞吐量。
- RPC** 远程过程调用 (Remote procedure call)。NFS 用于实现自定义网络数据服务器的底层数据交换机制。
- 软限制 (soft limit)** 对于磁盘配额而言，软限制是指用户可以暂时超过的文件系统资源（块或索引节点）阈值限制。如果超过软限制，系统将启动一个计时器。当超过软限制达到一定时间，系统将无法再分配更多的系统资源，除非您将文件系统的使用率降至软限制水平以下。

S

- SAM-QFS** 一种组合了 Sun StorageTek SAM 软件和 Sun StorageTek QFS 文件系统的配置。SAM-QFS 不仅为用户和管理员提供了高速的标准 UNIX 文件系统接口，而且还提供了若干存储及归档管理实用程序。SAM-QFS 既可以使用 Sun StorageTek SAM 命令集中的许多命令，也可以使用标准 UNIX 文件系统命令。
- samfsdump** 一个程序，为给定文件组创建控制结构转储文件并复制所有控制结构信息。该程序与 UNIX tar(1) 实用程序类似，但通常不复制文件数据。另请参见 **samfsrestore**。
- samfsrestore** 一个程序，用于从控制结构转储文件中恢复索引节点和目录信息。另请参见 **samfsdump**。
- SCSI** 小型计算机系统接口 (Small Computer System Interface)。一种电子通信技术规范，常用于像磁盘、磁带机和自动化库这样的外围设备。
- Sun SAM-Remote 服务器 (Sun SAM-Remote server)** 既是一台功能完备的 Sun StorageTek SAM 存储管理服务器，又是一个 Sun SAM-Remote 服务器守护进程。它可定义 Sun SAM-Remote 客户机之间共享的库。
- Sun SAM-Remote 客户机 (Sun SAM-Remote client)** 一种具有客户机守护进程的 Sun StorageTek SAM 系统。它包含许多伪设备，也有自己的库设备。客户机用以存储一个或多个归档副本的归档介质是由 Sun SAM-Remote 服务器决定的。
- 设备日志记录 (device logging)** 一项可配置功能，用于提供设备特定的错误信息，以供分析设备问题。

设备扫描程序 (device scanner)	该软件用于定期监视所有手动挂载的可移除设备，并检测是否存在可供用户或其他进程请求的已安装卡盒。
设备系列集 (family device set)	请参见系列集 (family set) 。
审计 (全面) (audit (full))	载入卡盒以验证其 VSN 的过程。对于磁光盘卡盒，其容量和空间信息将在确定后被输入到自动生成的库目录中。另请参见 VSN 。
释放程序 (releaser)	一种 Sun StorageTek SAM 组件，用于标识归档文件并释放其磁盘高速缓存副本，从而增加磁盘高速缓存的可用空间。释放程序可根据阈值的上下限来自动调整联机磁盘存储量。
释放优先级 (release priority)	用于确定文件系统中的文件在归档后的释放先后顺序。释放优先级的计算方法是：将文件的各个属性值与该属性对应的权数相乘，然后将所有相乘结果取和。
数据设备 (data device)	在文件系统中，数据设备指存储文件数据的一个或一组设备。
索引节点 (inode)	Index node （索引节点）的缩写。是文件系统用于描述文件的一种数据结构。一个索引节点可描述与文件相关联的所有属性（除了名称）。这些属性包括：所有权、访问、权限、大小和磁盘系统上的文件位置。
索引节点文件 (inode file)	文件系统上的一种特殊文件 (.inodes)，包含了驻留在文件系统上的所有文件的索引节点结构。索引节点的大小是 512 字节；索引节点文件属于元数据文件。在文件系统中，元数据文件与文件数据分开存储。

T

tar	磁带归档 (Tape archive)。一种用于归档映像的标准文件和数据记录格式。
TCP/IP	传输控制协议/Internet 协议 (Transmission Control Protocol/Internet Protocol)。Internet 协议 (IP) 负责主机到主机的寻址、路由和数据包传递；传输控制协议 (TCP) 负责在各个应用点之间可靠地传递数据。
脱机存储 (offline storage)	使用前需要操作员先将其载入。

V

vfstab 文件 (**vfstab file**)

vfstab 文件包含文件系统的挂载选项。在命令行中指定的挂载选项会覆盖在 /etc/vfstab 文件中指定的挂载选项，但在 /etc/vfstab 文件中指定的挂载选项会覆盖在 samfs.cmd 文件中指定的挂载选项。

VSN 卷序列名 (Volume serial name)。如果是将数据归档至可移除介质卡盒，VSN 是写入卷标中的磁带和光盘的逻辑标识符。如果是将数据归档至磁盘高速缓存，VSN 是磁盘归档集的唯一名称。

W

WORM 单次写入多次读取 (write once read many)。介质的一种存储分类，即只能写入一次，但可多次读取。

网络连接自动化库 (**network attached automated library**)

由不同供应商（如 StorageTek、ADIC/Grau、IBM 或 Sony 等）生产的库，由供应商提供的软件包控制。Sun StorageTek SAM 文件系统通过使用自动化库的专用 Sun StorageTek SAM 介质更换器守护进程，从而实现与供应商软件的连接。

伪设备 (**pseudo device**)

未关联任何硬件的软件子系统或驱动程序。

文件系统 (file system)

一种由文件和目录组成的多层结构集合。

文件系统专用指令 (**file-system-specific directives**)

位于 archiver.cmd 文件中的全局指令后的归档程序指令和释放程序指令。不同文件系统有不同的文件系统专用指令，但都以 fs = 开头。文件系统专用指令的作用域一直到下一条 fs = 指令行或文件结束标记。如果有多条指令作用于一个文件系统，则文件系统专用指令优先于全局指令。

X

系列集 (family set) 由一组独立物理设备（如某个自动化库中的磁盘组或驱动器组）所代表的存储设备。另请参见**存储系列集 (storage family set)**。

小型计算机系统接口 (Small Computer System Interface) 请参见 SCSI。

循环 (round-robin) 一种按顺序将全部文件写入若干逻辑磁盘的数据存取方法。当将单个文件写入磁盘时，文件将整个写入第一个逻辑磁盘。然后，第二个文件将写入下一个逻辑磁盘，依此类推。每个文件的大小决定了 I/O 的大小。

另请参见**磁盘分散读写 (disk striping)** 和**分散读写 (striping)**。

Y

以太网 (Ethernet) 一种局域分组交换网络技术。以太网最初是针对同轴电缆设计的。但现在，它同样适用于屏蔽双绞线电缆。以太网是一种 10 MB/s 或 100 MB/s 的局域网。

硬限制 (hard limit) 对于磁盘配额而言，硬限制是文件系统资源、数据块或索引节点 (inode) 的最大限制，用户不能超过该限制。

预分配 (preallocation) 在磁盘高速缓存中预先保留一定数量的连续空间以备写入文件的过程。只能对大小为零的文件指定预分配。有关更多信息，请参见 `setfa(1)` 手册页。

元数据 (metadata) 与数据有关的数据。元数据是用于在磁盘上定位文件的确切数据位置的索引信息。元数据由以下各项的有关信息组成：文件、目录、访问控制列表、符号链接、可移除介质、分段文件和分段文件索引。

元数据设备 (metadata device) 用于存储文件系统元数据的设备，如固态硬盘或镜像设备等。在单独的设备上保存文件数据和元数据可以提高性能。在 `mcf` 文件中，元数据设备被声明为 `ma` 文件系统中的 `mm` 设备。

远程过程调用 (remote procedure call) 请参见 **RPC**。

Z

- 直接 I/O (direct I/O)** 一种针对大数据块对齐连续 I/O 的属性。`setfa(1)` 命令的 `-D` 选项即为直接 I/O 选项。该选项可为文件或目录设置直接 I/O 属性。如果应用于目录，直接 I/O 属性是可以继承的。
- 直接访问 (direct access)** 一种文件属性（永不登台），可指定近线 (**nearline**) 文件直接从归档介质上访问，而无需在磁盘高速缓存中接收。
- 直接连接库 (direct attached library)** 使用 SCSI 接口直接连接到服务器上的自动化库。通过 SCSI 连接的库直接由 Sun StorageTek SAM 软件控制。
- 主机文件 (hosts file)** 主机文件包含共享文件系统中所有主机的列表。如果要将文件系统初始化为 Sun StorageTek QFS 共享文件系统，则必须先创建主机文件 `/etc/opt/SUNWsamfs/hosts.fs-name`，然后再创建文件系统。`sammkfs(1M)` 命令在创建文件系统时将使用此主机文件。您可以在以后使用 `samsharefs(1M)` 命令来替换或更新此主机文件的内容。
- 自动化库 (automated library)** 一种自动控制设备，可在没有操作人员参与的情况下自动载入或卸载可移除介质卡盒。自动化库包含一个或多个驱动器，以及一种用于将卡盒移入或移出存储插槽和驱动器的传输机制。
- 租借 (lease)** 授予客户机主机在指定时间段内对文件进行操作的权限。元数据服务器负责向每一台客户机主机发放租借。根据具体情况，可对租借进行续借以允许客户机主机继续操作文件。

索引

A

-access 归档程序指令, 57

ACSAPI 接口, 3

ACSLs 连接库, 224

ADIC/Grau 自动化库
操作, 214

age_priority preview.cmd 指令, 129

allsets 归档集, 31, 67

archivemeta 归档程序指令, 47

archiver(1M) 命令, 42, 167
示例, 92
输出样例, 38

archiver.cmd 文件, 29, 30, 91

-access 和 -nftv 指令, 57

archivemeta 指令, 47

archmax 指令, 47, 67

bufsize 指令, 48, 67

编辑 SAM-Remote, 164

创建, 42

drivemax、drivemin 和 drives 归档请求指令, 68

drives 指令, 49

多个元数据副本, 65

endparams 指令, 66

examine 指令, 49

fillvsns 归档请求参数, 69

fs 指令, 55

分散读写分段文件, 206

副本份数指令, 63

概述, 41

归档集分配, 55

归档时限, 65

回收指令, 71, 136

ignore 指令, 144, 147

interval 指令, 50

join path 参数, 71

lock 归档请求参数, 70

logfile 指令, 51

minsize 和 maxsize 指令, 58

name 指令, 59

norelease 指令, 64

notify 指令, 51

offline_copy 参数, 70

ovflmin 指令, 52

params 指令, 66

配置 SAM-Remote 环境中的回收过程, 172

配置回收程序以进行磁盘归档, 146

全局指令, 46

-recycle_dataquantity 指令, 146

-recycle_ignore 指令, 144, 146, 147

-recycle_mailaddr 指令, 146

-recycle_mingain 指令, 146

release 指令, 63

reserve 参数, 74

SAM-Remote 示例, 174

startage、startcount 和 startsize 参数, 78

释放过程中的角色, 117

释放和登台指令, 61

示例, 44

tapenonstop 参数, 73

- wait 指令, 54
- user 和 group 指令, 58
- vsn 关联指令, 79
- vsnpools 参数, 81
- vsns 和 endvsns 参数, 80
- 文件系统指令, 54
- 优先级参数, 77
- 在登台过程中的角色, 127
- 指令, 43, 46
 - 自动取消归档, 65
- archiver.sh 脚本, 51
- archmax 归档程序指令, 47, 67
- auditslot(1M) 命令, 15, 20

B

- bufsize 登台程序指令, 123
- bufsize 归档程序指令, 48, 67
- 标记卡盒, 13
- 部分释放, 105
 - 概述, 108
 - 用户选项, 110

C

- chmed(1M) 命令, 18, 20, 141, 191, 192
- cleandrive(1M) 命令, 19
- crontab 条目, 回收程序, 144, 147
- 重新归档, 已定义, 136
- 磁带存储设备, 请参见自动化库
- 磁带清洁设置, 19
- 磁光设备, 请参见自动化库
- 磁盘高速缓存
 - 释放优先级, 2
- 磁盘归档, 83
 - 配置, 84
 - 配置回收过程, 84, 137, 145
 - 启用, 86
 - 示例, 88
 - 在 Sun StorageTek 5800 上, 85
 - 指令, 85
- 错误消息, 用于回收程序, 143, 147

D

- defaults.conf 文件
 - exported_media 指令, 24, 27
 - 看管指令, 24, 27
 - 启用设备日志, 203
- devlog 文件, 201
- devlog 指令, 203
- diskvols.conf 文件, 83
- display_all_candidates 释放程序指令, 114
- drivemax 归档程序指令, 68
- drivemin 归档程序指令, 68
- drives 登台程序指令, 122
- drives 归档程序指令, 49
- drives 归档集参数指令, 68
- DZC-8000S 接口, 3
- 当前数据空间, 已定义, 135
- 导出介质, 概述, 22
- 导入介质, 概述, 22
- 登台程序
 - 部分登台, 108
 - 错误处理, 2
 - 概述, 2, 121
 - 记录活动, 124
 - 强制执行登台程序请求, 133
 - 日志文件字段, 125
 - 设置登台缓冲区大小, 123
 - 设置登台请求的数量, 126
 - 已定义, 121
 - 预备队列, 128
 - 在登台过程中的归档角色, 127
 - 指定驱动器数量, 122
 - 指令, 121
- 登台请求的错误处理, 2
- 独立驱动器
 - 载入介质, 27

E

- endparams 归档程序指令, 66
- endvsnpools 归档程序指令, 81
- examine 归档程序指令, 49
- export(1M) 命令, 219
- exported_media 指令, 24

F

File System Manager

创建帐户, 6

概述, 4

管理远程服务器, 8

使用 `sam-nrecycler`, 148

`-fillvsns` 归档请求参数, 69

`fs` 归档程序指令, 55

`fs` 释放程序指令, 113

Fujitsu LMF 自动化库操作, 215

分段文件, 205

归档, 206

G

`-group` 归档程序指令, 58

高可用性 SAM, 4, 195

归档操作超时, 37

归档程序

`allsets` 归档集, 67

`-archmax` 参数, 35

保留的 VSN, 76

保留卷, 74

操作概述, 31

操作原理, 29

超时, 37

磁盘归档, 83

磁盘归档配置, 84

磁盘归档指令, 85

`-drivemin` 参数, 35

`-drives` 参数, 35

队列时间通知, 37

`-fillvsns` 参数, 36

分段文件, 206

副本定义指令, 63

概述, 1

归档归档请求, 37

归档集, 30

归档集参数, 34

归档集成员, 55

归档集成员冲突, 62

归档集处理指令, 66

归档介质的定义, 1, 29

归档请求, 34

归档时间间隔的定义, 30

归档时限的定义, 30

归档优先级, 31

`-join` 参数, 34

控制归档扫描, 49

控制归档文件的大小, 67

控制卷溢出, 52

控制取消归档, 72

控制使用的驱动器数量, 49

控制文件大小, 47

联合归档, 71

连续归档, 33, 49

默认值, 30

`-ovflmin` 参数, 36

启用磁盘归档, 86

强制执行归档请求, 133

确定归档时限, 72

`-reserve` 参数, 34

日志文件, 39

日志文件示例, 39

`-sort` 和 `-rsort` 参数, 34

扫描归档, 34

设置归档程序缓冲区大小, 48, 67

设置归档时限, 65

设置优先级, 77

设置自动取消归档, 65

示例, 92

使用正则表达式, 59

守护进程, 30

数据验证, 41

VSN 池, 81

VSN 关联指令, 79

延迟归档程序的启动, 54

已定义, 29

预备队列, 91, 128

预定归档, 78

预定归档请求, 35

原则, 91

在 `archiver.cmd` 中指定文件系统, 55

在登台过程中的角色, 127

识别要归档的文件, 32

指定归档缓冲区锁定, 70

- 指定归档请求的驱动器数, 68
- 指定归档日志文件, 51
- 指定归档时间间隔, 50
- 指定文件系统数据的副本份数, 65
- 指令, 43, 46
- 重命名事件通知脚本, 51
- 阻止归档, 56
- 归档的队列时间通知, 37
- 归档副本保留, sam-nrecycler, 148
- 归档集
 - 副本份数, 63
 - 路径, 56
 - 名称, 56
 - 搜索标准, 56
 - 文件属性, 56
- 归档请求, 34
 - 预定, 35
- 归档请求文件, 32
- 归档日志
 - 备份, 91
- 过期数据空间, 已定义, 135

H

HA-SAM

- 简介, 4
- 配置, 195

HASStoragePlus 文件系统, 196

hlwm_priority preview.cmd 文件指令, 130

-hwm 回收程序指令, 141

hwm_archive 挂载选项, 50

hwm_priority preview.cmd 文件指令, 130

回收程序

- 编辑 archiver.cmd 文件以进行磁盘归档, 146
- crontab 条目, 144, 147
- 操作原理, 136
- 磁盘归档副本, 136
- 磁盘归档配置, 145
- 对磁盘归档进行重新归档, 86
- 方法, 137
- 概述, 2, 135
- 忽略库, 142

- 配置, 136
- recycler.cmd 文件示例, 141
- recycler.sh 脚本, 145
- 上限指令, 141
- 使用 chmed(1M) 命令强制执行, 141
- VSN 最小增益指令, 142
- 用于归档副本保留的 nrecycler, 148
- 邮件通知选项, 142
- 指定日志文件, 138
- 指定自动化库的回收过程, 139
- 指令, 137
- 阻止回收, 138
- 回收程序日志文件, 176, 180, 188, 192
 - no-data VSN, 188
 - partially full VSN, 190
- 回收指令, 71

I

IBM 3494 自动化库

- 操作, 218

IBM 3584 自动化库

- 操作, 217
- 分区, 217
- 清洁, 217

idle 命令, 11

ignore 回收程序指令, 142, 144, 147

import(1M) 命令, 16, 22, 24, 214, 216, 218, 220, 223, 224

interval 归档程序指令, 50

J

-join path 归档程序指令, 71

机械手, 请参见自动化库

技术支持, xxiii

加密, 使用磁带机, 27

介质

- 错误, 20
- 库, 请参见自动化库
- 卸载, 13
- 移动, 22
- 载入, 12

卷溢出

ovflmin 归档程序指令, 52

示例, 52

文件, 204

K

卡盒

标记, 13

导出, 25, 26

导入, 24, 25

导入和导出, 22

清洁, 16

删除, 21

卸载, 13, 28

载入, 12, 27

看管指令, 24

可移除介质

启动, 10

停止, 11

文件, 203

可用空间, 已定义, 135

库回收程序指令, 139

库历史记录, 23

库目录

查看, 28

概述, 22

L

lhwm_priority preview.cmd 文件指令, 130

list_size 释放程序指令, 116

lmcpd 接口, 3

load_notify.sh 脚本, 26

-lock 归档程序指令, 70

log_rotate.sh 脚本, 211

logfile 登台程序指令, 124

logfile 归档程序指令, 51

logfile 回收程序指令, 138

logfile 释放程序指令, 115

lwm_priority preview.cmd 文件指令, 130

历史记录, 23

联合归档, 71

连续归档, 49

M

-mail 回收程序指令, 142

maxactive 登台程序指令, 126

-maxsize 归档程序指令, 58

mcf 文件, 3

库历史记录, 23

SAM-Remote 配置, 153

min_residence_age 释放程序指令, 114

-mingain 回收程序指令, 142

-minsize 归档程序指令, 58

move(1M) 命令, 220, 222

mount(1M) 命令

部分释放和登台选项, 108

默认值

归档程序, 30

目录, 概述, 22

N

-name 归档程序指令, 59

NFS 文件共享, 151

-nftv 归档程序指令, 57

no_archive 归档集, 31, 56

no_recycle 回收程序指令, 138, 175

no_release 释放程序指令, 114

no-data VSN, 188

-norelease 归档程序指令, 64

notify 归档程序指令, 51

nrecycler.cmd 文件, 148

O

-o maxpartial 挂载选项, 108

-o partial 挂载选项, 108

-o partial_stage 挂载选项, 108

odlabel(1M) 命令, 14, 189, 191, 193, 204
-offline_copy 归档程序指令, 70
ovflmin 归档程序指令, 52

P

params 归档程序指令, 66
partially full VSN, 190
pkginfo(1M) 命令, 157
-pool 归档程序指令, 79
preview.cmd 文件, 128
 age_priority 指令, 129
 hlwm_priority 指令, 130
 hwm_priority 指令, 130
 lhwm_priority 指令, 130
 lwm_priority 指令, 130
 设置优先级, 132
 vsn_priority 指令, 129
 指令, 128
-priority 归档程序指令, 77

Q

-queue_time_limit 参数, 37
清洁磁带机, 16
清洁卡盒, 16
清洁循环, 重置, 18
请求文件, 参见可移除介质文件
驱动器, 清洁, 16
驱动器, 使用加密, 27
取消归档, 65, 72

R

rearch_no_release 释放程序指令, 116
-recycle_dataquantity 归档程序指令, 146
-recycle_hwm 归档程序指令, 173
-recycle_ignore 归档程序指令, 144, 146, 147, 173
-recycle_mailaddr 归档程序指令, 146
-recycle_mingain 归档程序指令, 146, 173

recycle_minopbs percent 回收程序指令, 86
-recycle_vsncount 归档程序指令, 173
recycler.cmd 文件
 创建, 140
 -hwm 指令, 141
 ignore 指令, 142, 144, 147
 库指令, 139
 logfile 指令, 138
 -mail 指令, 142
 -mingain 指令, 142
 no_recycle 指令, 138
 SAM-Remote 配置, 173, 175
 示例, 141
 在 SAM-Remote 环境中, 168
recycler.sh 脚本, 145, 176
-release 归档程序指令, 61, 63
release 和 norelease 指令, 同时使用, 64
release(1) 命令, 105
 部分释放, 110
releaser.cmd 文件, 110, 117
 display_all_candidates 指令, 114
 fs 指令, 113
 list_size 指令, 116
 logfile 指令, 115
 min_residence_age 指令, 114
 no_release 指令, 114
 rearch_no_release 指令, 116
 weight_age 指令, 112
 weight_age_access 指令, 112
 weight_age_modify 指令, 112
 weight_age_residence 指令, 112
 weight_size 指令, 113
 文件时限指令, 111
request(1) 命令, 136, 203
 参数, 204
-reserve 归档程序指令, 74
日志检测页, 206
日志文件
 备份, 51, 91
 登台程序, 124
 管理 SEF 日志文件, 211
 归档程序, 39, 51
 回收程序, 176, 180, 188, 192

- 回收程序错误消息, 143, 147
- 启用设备日志, 202
- SEF 日志文件, 206
- 设备日志记录, 201
- 释放程序, 115

容量, 已定义, 135

软件

- 文档, xxii

S

sam_release(3) 库例程, 110

sam-amld 守护进程, 128

sam-archiverd 守护进程, 30, 34 - 37

sam-arcopy 进程, 30

- 日志文件, 39

sam-arfind 进程, 30, 32

- 日志文件, 39

samcmd(1M) 命令, 11

- audit 选项, 16
- idle 选项, 28, 158
- load 选项, 12
- off 选项, 12
- on 选项, 11
- unload 选项, 13, 25 - 26

samd hastop 命令, 使用 HA-SAM, 200

samd(1M) 命令

- start 选项, 10, 165
- stop 选项, 158, 200

samexport(1M) 命令, 22, 25, 215, 216, 221, 223, 225

samfsdump(1M) 命令, 91

sam-genericd 守护进程, 3

sam-ibm3494d 守护进程, 3

sam-nrecycler(1M) 命令, 148

sam-recycler(1M) 命令, 137, 143, 147, 176, 180, 191, 192, 193

SAM-Remote

- 安装, 155
- 编辑 archiver.cmd 文件以配置回收过程, 172
- 编辑 mcf 文件, 158
- 编辑 recycler.cmd 文件, 173
- 服务器配置概述, 153

- 服务器配置文件, 162
- 概述, 150
- 归档, 154
- 回收, 168
- 回收程序的服务器配置, 169
- 回收程序的客户机配置, 170
- 回收指令, 173
- 技术概述, 152
- 客户机配置概述, 153
- 客户机配置文件, 160
- 客户机与服务器之间的交互作用, 153
- 库目录, 154
- 目录, 167
- 配置, 155
- 配置回收过程, 171
- 配置回收过程 (方法 2) , 192
- 配置示例, 155
- 启用归档, 164

samu(1M) R 显示屏幕, 166

- 伪设备, 153
- 限制, 151
- 要求, 151
- 预定回收程序, 190

sam-robotsd 守护进程, 3

sam-serverd 守护进程, 153

samset(1M) 命令

- 启用设备日志, 202

sam-sonyd 守护进程, 3

sam-stkd 守护进程, 3

samu(1M)

- arrun 命令, 191
- R 显示屏幕, 166
- s 显示屏幕, 165
- v 显示屏幕, 167

scrgadm 命令, 197

SEF, 206

- 报告输出, 208
- 日志文件, 211
- sysevent 处理器, 211

sefdata 文件, 207

sefreport(1M) 命令, 206

- 选项, 208

segment(1) 命令, 205

- showqueue(1M) 命令, 32
- showrev(1M) 命令, 158
- Sony 8400 PetaSite 自动化库
 - 操作, 219
- Sony 网络连接自动化库
 - 操作, 222
- ssum -e, 数据验证, 41
- stage 归档程序指令, 61
- stager.cmd 文件, 121
 - bufsize 指令, 123
 - 创建, 122
 - drives 指令, 122
 - logfile 指令, 124
 - maxactive 指令, 126
 - 示例, 127
- startage 归档程序指令, 78
- startcount 归档程序指令, 78
- startsize 归档程序指令, 78
- Sun Cluster
 - 高可用性 SAM, 195
 - SAM 配置, 4
- Sun StorageTek 5800, 磁盘归档, 85, 87
- Sun StorageTek ACSLS 连接自动化库
 - 操作, 224
- Sun StorageTek 库, 使用 HA-SAM, 200
- SUNWhasam RTR 文件, 197
- sysevent 功能, 211
- syseventadm(1M) 命令, 212
- syseventd 文件, 212
- 上限, 106
 - preview.cmd 文件指令, 130
 - 使用回收程序, 141
- 设备日志记录, 201
 - 何时使用, 202
 - 启用, 202
 - 事件, 202
- 审计
 - 卷, 15
 - 自动化库, 15
- 释放程序
 - archiver.cmd 文件的角色, 117
 - 备选文件定义, 107
 - 部分释放, 105, 107
 - 部分释放, 用户选项, 110
 - 部分释放选项, 109
 - 操作原理, 106
- fs 指令, 113
- 概述, 2, 105, 106
- 命令文件, 110
- 配置, 117
- 权重, 107
- 日志文件, 115
- 释放优先级指令, 111
- 手动操作, 119
- 文件时限, 107
- 优先级, 107
- 指令, 110

使用 SAM-Remote 归档, 154, 164

守护进程

- sam-amld, 128
- sam-archiverd, 34
- samarchiverd, 30
- sam-genericd, 3
- sam-ibm3494d, 3
- sam-robotd, 3
- sam-serverd, 153
- sam-sonyd, 3
- sam-stkd, 3
- 自动化库守护进程, 3

数据验证, 41

数据验证的校验和, 41

T

- tapealert(1M)
 - 使用 SEF, 206
- tapenonstop 归档程序指令, 73
- tplabel(1M) 命令, 14, 189, 191, 193, 204
- 条码
 - 用于清洁卡盒, 16

U

- user 归档程序指令, 58

V

VSN

- 池, 示例, 101
 - 池指令, 81
 - 关联指令, 79
 - 使用正则表达式, 81
 - 最小回收增益, 142
- vsn_priority_preview.cmd 指令, 129
- vsnpools 归档程序指令, 81

W

- wait 归档程序指令, 54
- weight_age 释放程序指令, 112
- weight_age_access 释放程序指令, 112
- weight_age_modify 释放程序指令, 112
- weight_age_residence 释放程序指令, 112
- weight_size 释放程序指令, 113
- wm_priority_preview.cmd 文件指令, 130
- 文档, xxii
- 文件, 设置属性, 61
- 文件系统
- 概述, 1

X

- 系统错误工具, 参见 SEF
- 下限
- preview.cmd 文件指令, 130
- 消息文件, 143, 147
- 卸载介质, 13
- 许可
- 常规信息, xxiii

Y

- 邮箱, 24
- 预备队列, 91
- 预备请求
- 按照 VSN 确定优先级, 129
 - 按照上限和下限确定优先级, 130
 - 按照时限确定优先级, 129

- 规划, 132
 - 计算优先级, 131
 - 界限指令, 130
 - 配置示例, 133
 - 确定优先级, 128
- 预备请求的 VSN 优先级, 129
- 预备请求的界限指令, 130
- 元数据副本, 65

Z

- 在 Sun SAM-Remote 环境中执行回收过程, 168
- 载入介质, 12
- 手动载入的驱动器, 27
- 载入通知, 启用, 26
- 正则表达式, 59
- 指令
- 归档, 46
- 转储文件, 使用 sam-nrecycler, 148
- 自动化库
- ADIC/Grau, 214
 - 操作, 22
 - 打开, 11
 - 导入和导出, 24
 - Fujitsu LMF, 215
 - 供应商专用过程, 213
 - 关闭, 12
 - IBM 3494, 218
 - IBM 3584, 217
 - 历史记录, 23
 - 目录, 22
 - 清洁, 16
 - SCSI 连接, 参见自动化库, 直接连接
 - Sony 8400 PetaSite, 219
 - Sony 网络连接, 222
 - Sun StorageTek ACSLS 连接, 224
 - 审计, 15
 - 守护进程, 3
 - 网络连接, 3
 - 已定义, 9
 - 指定回收参数, 139
 - 直接连接, 3
- 自动清洁, 19

