

Sun Storage Archive Manager 5.3 配置和管理指南

版权所有 © 2011, 2012, Oracle 和/或其附属公司。保留所有权利。

本软件和相关文档是根据许可证协议提供的，该许可证协议中规定了关于使用和公开本软件和相关文档的各种限制，并受知识产权法的保护。除非在许可证协议中明确许可或适用法律明确授权，否则不得以任何形式、任何方式使用、拷贝、复制、翻译、广播、修改、授权、传播、分发、展示、执行、发布或显示本软件和相关文档的任何部分。除非法律要求实现互操作，否则严禁对本软件进行逆向工程设计、反汇编或反编译。

此文档所含信息可能随时被修改，恕不另行通知，我们不保证该信息没有错误。如果贵方发现任何问题，请书面通知我们。

如果将本软件或相关文档交付给美国政府，或者交付给以美国政府名义获得许可证的任何机构，必须符合以下规定：

U.S. GOVERNMENT END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

本软件或硬件是为了在各种信息管理应用领域内的一般使用而开发的。它不应被应用于任何存在危险或潜在危险的应用领域，也不是为此而开发的，其中包括可能会产生人身伤害的应用领域。如果在危险应用领域内使用本软件或硬件，贵方应负责采取所有适当的防范措施，包括备份、冗余和其它确保安全使用本软件或硬件的措施。对于因在危险应用领域内使用本软件或硬件所造成的一切损失或损害，Oracle Corporation 及其附属公司概不负责。

Oracle 和 Java 是 Oracle 和/或其附属公司的注册商标。其他名称可能是各自所有者的商标。

Intel 和 Intel Xeon 是 Intel Corporation 的商标或注册商标。所有 SPARC 商标均是 SPARC International, Inc 的商标或注册商标，并应按照许可证的规定使用。AMD、Opteron、AMD 徽标以及 AMD Opteron 徽标是 Advanced Micro Devices 的商标或注册商标。UNIX 是 The Open Group 的注册商标。

本软件或硬件以及文档可能提供了访问第三方内容、产品和服务的方式或有关这些内容、产品和服务的信息。对于第三方内容、产品和服务，Oracle Corporation 及其附属公司明确表示不承担任何种类的担保，亦不对其承担任何责任。对于因访问或使用第三方内容、产品或服务所造成的任何损失、成本或损害，Oracle Corporation 及其附属公司概不负责。

目录

前言	13
1 关于 SAM-QFS	15
SAM-QFS 的组成部分	15
归档	15
释放	16
回写	16
回收	16
支持的存储设备	17
SAM-Remote 软件	17
2 配置用于归档的存储设备	19
任务列表：配置用于归档的存储设备	19
添加用于归档的磁带设备	20
包含配置信息的文件	20
▼ 如何添加用于归档的磁带设备（命令行）	21
▼ 如何添加用于归档的磁带机接口目标 ID 和 LUN（命令行）	22
▼ 如何添加用于归档的磁带设备 (SAM-QFS Manager)	23
添加用于归档的库或磁光盘驱动器	24
▼ 如何使用 SAM-QFS Manager 在 SCSI 或 FC 环境中配置设备支持	24
▼ 如何为直接连接库配置设备支持	25
直接连接库初始化失败后的恢复	25
检验并实施配置	26
▼ 如何检验是否已配置所有设备并实施更改	26
处理 st.conf 文件中的错误	27

3	执行其他 SAM-QFS 配置	29
	使用 NFS 客户机系统共享文件系统	29
	▼ 如何使用 NFS 共享文件系统	29
	▼ 如何在客户机中挂载文件系统	30
	编辑 defaults.conf 文件	31
	可以从 defaults.conf 进行控制的特性	31
	▼ 如何定制默认值	33
	配置远程通知功能	33
	▼ 如何启用远程通知功能	33
	▼ 如何禁用远程通知功能	34
	添加管理员组	35
	▼ 如何添加管理员组	35
	▼ 如何启用系统日志记录	35
	配置其他 Sun Storage 产品	36
4	为网络连接自动化库创建参数文件	37
	为网络连接自动化库创建参数文件	37
	▼ 如何为网络连接自动化库创建参数文件	37
	配置 StorageTek ACSLS 连接自动化库参数文件	38
	配置 ADIC/Grau 自动化库参数文件	40
	配置 IBM 3494 自动化库参数文件	41
	配置 Sony 网络连接自动化库参数文件	43
	关于共享驱动器	45
5	检查库中的驱动器顺序	47
	检查库中的驱动器顺序	47
	▼ 如何检查配有前面板的库的驱动器顺序	47
	▼ 如何检查未配有前面板的磁带库的驱动器顺序	48
	▼ 如何检查未配有前面板的磁光盘库的驱动器顺序	49
	▼ 如何检查网络连接库的驱动器顺序	50
6	填充目录	53
	填充目录	53
	▼ 如何填充具有许多卷的自动化库	53

▼ 如何填充具有少量卷的自动化库	54
▼ 如何填充 IBM 3494 自动化库	54
▼ 如何快速填充 StorageTek ACSLS 连接库	55
StorageTek ACSLS 连接自动化库：常见问题和错误消息	55
StorageTek ACSLS 参数文件错误	55
StorageTek ACSLS 库错误	55
VSN 导入错误	56
7 管理自动化库和手动装入的驱动器	57
特定于供应商的库操作过程	57
▼ 如何启动可移除介质的操作	57
▼ 如何停止可移除介质的操作	58
▼ 如何打开自动化库	58
▼ 如何关闭自动化库	59
▼ 如何手动装入卡盒	59
▼ 如何手动卸载卡盒	60
标记卡盒及取消卡盒标记	60
▼ 如何标记或重新标记磁带	60
▼ 如何标记或重新标记光盘	61
▼ 如何审计卷	62
▼ 如何审计直接连接自动化库	62
使用清洁卡盒	63
▼ 如何使用具有条码的清洁卡盒	63
▼ 如何使用没有条码的清洁卡盒	63
▼ 如何重置清洁循环次数	64
▼ 如何限制清洁循环次数	64
▼ 如何自动清洁磁带机	65
▼ 如何手动清洁磁带机	65
▼ 如何清除介质错误	65
▼ 如何从驱动器中取出卡住的卡盒	66
目录操作和卡盒的导入与导出	67
跟踪导出的介质—历史记录	68
关于从自动化库执行导入和导出操作	69
▼ 如何从具有邮箱的库导入卡盒	69
▼ 如何从具有邮箱的库导出卡盒	70

▼ 如何从没有邮箱的库导入卡盒	70
▼ 如何从没有邮箱的库导出卡盒	70
▼ 如何启用装入通知	71
使用具备加密功能的驱动器	71
手动装入驱动器操作	72
▼ 如何将卡盒装入手动装入的设备	72
▼ 如何卸载卡盒	72
▼ 如何查看库目录	72
8 管理特定于供应商的库	73
StorageTek ACSLS 连接自动化库	73
▼ 如何导入磁带	74
▼ 如何使用邮箱导出磁带	74
ADIC/Grau 自动化库	75
▼ 如何导入卡盒	75
▼ 如何导出卡盒	75
IBM 3584 UltraScalable 磁带库	76
导入卡盒	76
清洁驱动器	76
分区	76
▼ 如何从逻辑库中取出卡盒	77
IBM 3494 库	77
▼ 如何导入卡盒	77
▼ 如何导出卡盒	78
Sony 8400 PetaSite 直接连接自动化库	78
▼ 如何导入磁带	78
▼ 如何在邮箱插槽未用作存储插槽时导出磁带	79
▼ 如何在邮箱插槽用作存储插槽时导出磁带	79
▼ 如何将卡盒移至另一个插槽	80
Sony 网络连接自动化库	80
▼ 如何导入卡盒	80
▼ 如何导出卡盒	81
9 关于归档	83
计划归档操作	83

预备队列	84
归档程序守护进程	84
归档操作	85
步骤 1: 确定要归档的文件	86
步骤 2: 组合归档请求	87
步骤 3: 预定归档请求	88
步骤 4: 对归档请求中的文件进行归档	90
archiver -l 命令默认输出样例	91
归档操作的日志文件和事件日志记录	91
数据验证	93
10 配置归档程序	95
关于 archiver.cmd 文件	95
归档指令	96
归档集指令	96
▼ 如何使用命令行创建 archiver.cmd 文件	97
▼ 如何使用 SAM-QFS Manager 创建 archiver.cmd 文件	97
archiver.cmd 文件示例	98
关于磁盘归档	108
关于 diskvols.conf 文件	109
磁盘归档指令	110
▼ 如何在客户机主机上启用磁盘归档	112
▼ 如何在服务器主机上配置磁盘归档	113
▼ 如何启用磁盘归档	114
磁盘归档示例	115
11 归档指令 (archiver.cmd)	119
全局归档指令	119
archivemeta 指令: 控制是否对元数据进行归档	120
archmax 指令: 控制归档文件的大小	120
bufsize 指令: 设置归档程序缓冲区大小	121
drives 指令: 控制用于归档的驱动器数	121
examine 指令: 控制归档扫描	122
interval 指令: 指定归档时间间隔	122
logfile 指令: 指定归档程序日志文件	123

notify 指令：重命名事件通知脚本	123
ovflmin 指令：控制卷溢出	124
scanlist_squash 指令：控制扫描列表合并	125
setarchdone 指令：控制 archdone 标志的设置	125
wait 指令：延迟归档程序启动	126
文件系统指令	126
fs 指令：指定文件系统	126
全局指令作为文件系统指令	127
归档副本指令	127
-release 指令：归档后释放磁盘空间	127
-norelease 指令：延迟磁盘空间释放	128
同时使用 -release 和 -norelease	128
设置归档时限	128
自动取消归档	128
为元数据指定多个副本	129
12 归档集指令 (archiver.cmd)	131
关于归档集指令	131
归档集分配指令	132
文件时限 <i>search-criterion</i> 参数：-access 和 -nftv	133
文件时限 <i>search-criterion</i> 参数：-after	133
文件大小 <i>search-criterion</i> 参数：-minsize 和 -maxsize	134
属主和组 <i>search-criterion</i> 参数：-user 和 -group	134
使用模式匹配的文件名 <i>search-criterion</i> 参数：-name regex	135
释放和回写 <i>file-attributes</i> 参数：-release 和 -stage	136
归档集中的成员冲突	137
归档集副本参数	137
控制归档文件的大小：-archmax 参数	138
设置归档程序缓冲区大小：-bufsize 参数	139
为归档请求指定驱动器数：-drivemax、-drivemin 和 -drives	139
最大化卷上的空间：-fillvsns 参数	141
指定归档缓冲区锁定：-lock 参数	141
创建脱机文件的归档副本：-offline_copy 参数	141
指定回收	142
排序归档文件：-sort 和 -rsort 参数	142

控制取消归档	142
控制归档文件的写入方式: <code>-tapenonstop</code> 参数	143
保留卷: <code>-reserve</code> 参数	143
设置归档优先级: <code>-priority</code> 参数	145
预定归档: <code>-startage</code> 、 <code>-startcount</code> 和 <code>-startsize</code> 参数	147
VSN 关联指令	147
VSN 池指令	149
13 SAM-QFS 中的数据完整性验证	151
关于 DIV	151
在 SAM-QFS 中使用 DIV 的最低硬件和软件要求	152
在 SAM-QFS 中配置 DIV	152
▼ 如何在 SAM-QFS 中配置 DIV	152
DIV 使用示例	153
磁带检验 (<code>tpverify</code>) 命令	154
磁带检验 (<code>tpverify</code>) 命令返回值	154
14 关于释放	157
释放程序进程	157
释放程序概念	158
时限	158
备选文件	158
优先级	159
权重	159
部分释放	159
关于部分释放和部分回写	160
系统管理员选项摘要	161
用户选项摘要	162
15 配置回写程序	163
关于回写	163
关于 <code>stager.cmd</code> 文件	164
▼ 如何创建 <code>stager.cmd</code> 文件	164
<code>drives</code> 指令: 指定进行回写的驱动器数量	165

bufsize 指令：设置回写缓冲区大小	165
logfile 指令：指定日志文件	166
maxactive 指令：指定回写请求的数量	168
归档集分配指令：指定归档集中所有文件的回写属性	168
copysel 指令：指定回写的副本顺序	168
确定预备请求的优先级	169
关于 preview.cmd 文件	169
▼如何设置全局 VSN 和时限优先级指令	170
▼如何设置全局或特定于文件系统的界限指令	171
设置预备请求的优先级方案	172
16 配置回收程序	175
关于回收	175
回收过程	176
规划回收	176
回收方法	177
控制回收	178
启动回收程序	178
在可移除介质卡盒上配置回收	179
创建 recycler.cmd 文件	179
logfile 指令：指定日志文件	179
no_recycle 指令：阻止回收	180
library 指令：指定回收自动化库	181
创建 recycler.sh 文件	183
配置磁盘归档卷的回收	183
编辑 archiver.cmd 文件	183
磁盘归档的回收程序日志记录	184
回收以实现归档副本保留	184
17 SAM-QFS 高级主题	187
使用设备日志记录	187
何时使用设备日志	188
▼如何通过使用 samset 命令启用设备日志	188
▼如何通过编辑 defaults.conf 文件启用设备日志	188
使用可移除介质文件	189

创建可移除介质或卷溢出文件	189
使用分段文件	190
归档分段文件	191
使用系统错误工具报告	191
▼ 如何启用 SEF 报告	192
▼ 如何生成 SEF 报告输出	192
管理 SEF 日志文件	194
SEF sysevent 功能	195
▼ 如何创建 SEF sysevent 处理程序	195
18 使用 Sun SAM-Remote 软件	197
Sun SAM-Remote 软件概述	197
系统要求	198
软件限制	199
客户机与服务器之间的交互作用	199
配置 Sun SAM-Remote 软件	202
▼ 步骤 1：登录至潜在的服务器及客户机主机	204
▼ 步骤 2：检验客户机和服务器软件	204
▼ 步骤 3：在每台客户机上编辑 mcf 文件	205
▼ 步骤 4：创建 Sun SAM-Remote 客户机配置文件	207
▼ 步骤 5：编辑服务器的 mcf 文件	207
▼ 步骤 6：创建服务器的配置文件	208
▼ 步骤 7：启用归档	210
使用 Sun SAM-Remote 软件回收	212
在 Sun SAM-Remote 环境中回收（方法 1）	213
▼ 如何配置回收（方法 1）	215
关于如何配置回收（方法 1）的示例	218
▼ 如何回收 no-data VSN	227
▼ 如何回收 partially full VSN	228
在 Sun SAM-Remote 环境中回收（方法 2）	230
▼ 如何配置回收（方法 2）	230

前言

《Sun Storage Archive Manager 5.3 配置和管理指南》提供了有关管理 Sun Storage Archive Manager 软件的信息。

本书的目标读者

本指南的目标读者是对 Sun Storage Archive Manager 软件管理感兴趣的系统管理员。

阅读本书之前

开始管理 SAM-QFS 软件之前，请先阅读《Sun QFS 和 Sun Storage Archive Manager 5.3 安装指南》。

相关书籍

- 《Sun QFS 和 Sun Storage Archive Manager 5.3 安装指南》
- 《Sun QFS 文件系统 5.3 配置和管理指南》
- 《Sun QFS 和 Sun Storage Archive Manager 5.3 发行说明》
- 《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》

获取 Oracle 支持

Oracle 客户可通过 My Oracle Support 获取电子支持。有关信息，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>，或访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>（如果您听力受损）。

印刷约定

下表介绍了本书中的印刷约定。

表 P-1 印刷约定

字体或符号	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。 machine_name% you have mail.
AaBbCc123	用户键入的内容，与计算机屏幕输出的显示不同	machine_name% su Password:
<i>aabbcc123</i>	要使用实名或值替换的命令行占位符	删除文件的命令为 <code>rm filename</code> 。
<i>AaBbCc123</i>	保留未译的新词或术语以及要强调的词	这些称为 <i>Class</i> 选项。 注意： 有些强调的项目在联机时以粗体显示。
新词术语强调	新词或术语以及要强调的词	高速缓存 是存储在本地的副本。 请勿保存文件。
《书名》	书名	阅读《用户指南》的第 6 章。

命令中的 shell 提示符示例

下表显示了 Oracle Solaris OS 中包含的缺省 UNIX shell 系统提示符和超级用户提示符。请注意，在命令示例中显示的缺省系统提示符可能会有所不同，具体取决于 Oracle Solaris 发行版。

表 P-2 shell 提示符

shell	提示符
Bash shell、Korn shell 和 Bourne shell	\$
Bash shell、Korn shell 和 Bourne shell 超级用户	#
C shell	machine_name%
C shell 超级用户	machine_name#

关于 SAM-QFS

Sun Storage Archive Manager (SAM-QFS) 软件提供了具有存储、归档管理及恢复功能的可配置文件系统。SAM-QFS 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。归档介质可包括以下各项：

- 其他文件系统中的磁盘分片
- 自动或手动装入的存储设备中的可移除磁带或磁光盘卡盒

SAM-QFS 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。

SAM-QFS 的组成部分

SAM-QFS 归档生命周期包含以下各节中介绍的阶段：

- 第 15 页中的“归档”
- 第 16 页中的“释放”
- 第 16 页中的“回写”
- 第 16 页中的“回收”

归档

归档程序自动将联机磁盘高速缓存中的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。归档程序要求您配置 `archiver.cmd` 文件，以定义要归档的内容。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。当文件符合站点定义的一套选择条件时，归档进程将会启动。有关更多信息，请参见 [第 9 章，关于归档](#)。

释放

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。

释放是指释放已归档文件的数据所占用的主（磁盘）存储空间的过程。使用上限和下限（用磁盘总空间的百分比表示）来管理联机磁盘高速缓存的空闲空间。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。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。文件的第一部分可保留在磁盘上，以加快访问速度并屏蔽回写过程的延迟。有关更多信息，请参见第 14 章，[关于释放](#)。

回写

回写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，回写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，回写程序会自动将该文件或文件段的数据重新回写到联机磁盘高速缓存中。一旦开始回写操作，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回写到磁盘高速缓存中。

SAM-QFS 软件可自动处理回写请求错误。如果返回回写错误，则系统会尝试查找文件的下一个可用归档副本。系统可自动处理的回写错误包括介质错误、介质不可用、自动化库不可用以及其他错误。有关更多信息，请参见第 15 章，[配置回写程序](#)。

回收

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。

当用户修改文件时，与这些文件的旧版本相关联的归档副本将被视为过期。由于这些副本不再有用，因此可从系统中清除。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。回收进程对最终用户是透明的。

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异地存储设备中，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。

有关更多信息，请参见第 16 章，[配置回收程序](#)。

支持的存储设备

SAM-QFS 环境可支持的磁带存储设备和磁光设备种类非常多。SAM-QFS 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，Oracle StorageTek 库使用直接连接方式。SAM-QFS 系统使用自动化库的 SCSI 标准直接控制这些库。
- **网络连接**—可将 SAM-QFS 软件配置为库主机系统的客户机。网络连接库包括 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，SAM-QFS 软件使用自动化库专用的守护进程，与供应商软件协作。

SAM-QFS 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 SAM-QFS 环境中包括的可移除介质设备、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。

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

有关支持的存储设备的列表，请与 Oracle 销售代表联系。有关配置存储设备的信息，请参见第 2 章，[配置用于归档的存储设备](#)。

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 SAM-QFS 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回写文件。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 18 章，[使用 Sun SAM-Remote 软件](#)。

配置用于归档的存储设备

仅当您计划将数据归档至磁带或磁光盘介质时，才执行本章中的任务。

任务列表：配置用于归档的存储设备

下表显示了配置用于归档的存储设备的过程。

任务	说明	更多信息
创建设备列表	取得要配置的设备的清单。	请参见表 2-1。
添加磁带设备	对要添加到 SAM-QFS 环境中的每个磁带机执行此任务。	请参见第 20 页中的“添加用于归档的磁带设备”第 20 页中的“添加用于归档的磁带设备”。
添加磁带机接口目标 ID 和 LUN	如果磁带机是通过 SCSI 或 FC 接口连接的，请执行此任务。	请参见第 22 页中的“如何添加用于归档的磁带机接口目标 ID 和 LUN（命令行）”。
添加库或磁光盘驱动器	如果有要在 SAM-QFS 环境中使用的磁光盘驱动器、SCSI 连接自动化库或 FC 连接自动化库，请执行此任务。	请参见第 24 页中的“添加用于归档的库或磁光盘驱动器”。
验证配置后的设备	确认已正确配置所有设备。	请参见第 26 页中的“检验并实施配置”。
启用存储设备配置	重新引导系统以使更改生效。	
创建参数文件	为网络连接自动化库创建参数文件。	请参见第 4 章，为网络连接自动化库创建参数文件。

添加用于归档的磁带设备

第 19 页中的“任务列表：配置用于归档的存储设备”中的过程包含一个示例，该示例基于下表中显示的清单列表。

表 2-1 清单列表—要配置的设备

设备名、生产商和型号	目标 ID	LUN	节点 WWN
SCSI 连接磁带机			
QUANTUM DLT7000	1	0	不适用
QUANTUM DLT7000	2	0	不适用
FC 连接磁带机			
StorageTek 9840	不适用	0	500104f00043abfc
StorageTek 9840	不适用	0	500104f00045eeaf
IBM ULT3580-TD1	不适用	0	500104f000416304
IBM ULT3580-TD1	不适用	0	500104f000416303
SCSI 连接自动化库			
StorageTek 9730	0	0	不适用
FC 连接自动化库			
StorageTek L700	不适用	0	500104f00041182b

注 - 设备名称的显示方式与其在搜索操作输出中的显示方式一样。

包含配置信息的文件

以下文件包含配置信息：

- `/kernel/drv/st.conf`—配置通过 SCSI 或 FC 连接至服务器的磁带机。
- `/kernel/drv/samst.conf`—配置 Sun Storage Archive Manager (SAM) 软件在默认情况下能够识别的以下设备：
 - 直接连接自动化库
 - 通过 SCSI 或 FC 连接至服务器的磁光盘驱动器

SAM 软件包中包含 `/opt/SUNWsamfs/examples/st.conf_changes` 文件，该文件含有在 Oracle Solaris 内核中默认不受支持的磁带机的配置信息。

▼ 如何添加用于归档的磁带设备（命令行）

也可以通过 SAM-QFS Manager 添加磁带设备。有关信息，请参见第 23 页中的“如何添加用于归档的磁带设备 (SAM-QFS Manager)”。

- 1 创建 `/kernel/drv/st.conf` 文件的备份副本。

例如：

```
# cp /kernel/drv/st.conf /kernel/drv/st.conf.orig
```

- 2 在 `/kernel/drv/st.conf` 文件中，删除 `tape-config-list` 条目中的井号 (#)。

- 3 修改 `/opt/SUNWsamfs/examples/st.conf_changes` 文件。

对于清单列表中的每个磁带机，执行以下操作：

- a. 查找设备定义条目。

例如，搜索示例清单中的 Quantum DLT 7000 磁带机可以找到以下条目：

```
"QUANTUM DLT7000", "DLT 7000 tape drive", "dlt7-tape",
```

- b. 将此条目从 `st.conf_changes` 文件复制到 `st.conf` 文件中，放在 `tape-config-list` 行的后面。

以下示例显示了生成的 `st.conf` 文件。

```
...
tape-config-list=
"QUANTUM DLT7000", "DLT 7000 tape drive", "dlt7-tape",
...
```

请注意该条目中最后一个字符串（括在引号中），该字符串是磁带配置值。在本示例中，最后一个字符串为 `"dlt7-tape"`。

- c. 在 `/opt/SUNWsamfs/examples/st.conf_changes` 文件中，找到以此磁带配置值开头的行。

在本示例中，该值为：

```
dlt7-tape = 1,0x36,0,0xd679,4,0x82,0x83,0x84,0x85,3;
```

- d. 将磁带配置值复制到 `st.conf` 文件中，并放在设备定义行的后面。

以下示例显示了现在包含在 `st.conf` 文件中的行。

```
...
tape-config-list=
"QUANTUM DLT7000", "DLT 7000 tape drive", "dlt7-tape";
dlt7-tape = 1,0x36,0,0xd679,4,0x82,0x83,0x84,0x85,3;
...
```

- e. 对添加的每类磁带机重复这些步骤。

注 – 在 `st.conf_changes` 文件中，使用相同磁带配置的每个设备定义都重复使用一个磁带配置值。在 `st.conf` 文件中，只为每个磁带配置值加入一个条目。例如，Sony SDT-5000 和 Sony SDT-5200 磁带机均使用 "DAT" 作为结尾字符串。DAT 磁带配置值使用一个条目即可。

- 4 将最后一个设备定义行末尾的逗号字符 (,) 替换为分号字符 (;)。

以下示例显示了包含 Quantum DLT 7000、StorageTek 9840 和 IBM ULT3580 磁带机定义的 `st.conf` 文件。分号在 "CLASS_3580" 后面。

```
...
tape-config-list=
"QUANTUM DLT7000", "DLT 7000 tape drive", "dlt7-tape",
"STK 9840", "STK 9840 Fast Access", "CLASS_9840",
"IBM ULT3580-TD1", "IBM 3580 Ultrium", "CLASS_3580";
dlt7-tape = 1,0x36,0,0xd679,4,0x82,0x83,0x84,0x85,3;
CLASS_9840 = 1,0x36,0,0x1d679,1,0x00,0;
CLASS_3580 = 1,0x24,0,0x418679,2,0x00,0x01,0;
...
```

- 5 保存您所做的更改，并关闭 `/opt/SUNWsamfs/examples/st.conf_changes` 文件。

由于在第 22 页中的“如何添加用于归档的磁带机接口目标 ID 和 LUN（命令行）”过程中要继续编辑 `st.conf` 文件，因此不要关闭该文件。

▼ 如何添加用于归档的磁带机接口目标 ID 和 LUN（命令行）

此过程说明如何针对硬件清单列表中通过 SCSI 或 FC 接口连接的每个磁带机验证目标 ID 和 LUN 条目，以及在必要时向 `st.conf` 文件中添加这些条目。

注 – 请勿使用此过程为磁光盘驱动器添加接口信息。请参见第 24 页中的“添加用于归档的库或磁光盘驱动器”。

- 1 打开 `/kernel/drv/st.conf` 文件。
- 2 配置 SCSI 内核接口以连接磁带机。
 - a. 搜索采用以下格式的条目，以找到 SCSI 目标 ID 和 LUN 的列表：

```
name="st" class="scsi" target=target lun=lun;
```

target 是所找到的每个 SCSI 驱动器的目标 ID，*lun* 是所找到的每个 SCSI 驱动器的相应 LUN。

- b. 查找与硬件清单列表上的每个 SCSI 目标 ID 和 LUN 相对应的条目。请参见示例 2-1。请注意，一些条目可能占用两行。

- 如果条目前面有井号(#)，请删除该字符。井号(#)表示该行是注释。
 - 如果缺少条目，请为所需的 SCSI 目标 ID 和 LUN 行创建条目。遵循步骤 2a 中显示的格式，并使用硬件清单列表中的信息。
- 3 如果未使用 StorageTek SAN Foundation 软件 I/O 堆栈，请在 SCSI 目标 ID 和 LUN 列表后为每个 FC 连接设备创建一行。请参见示例 2-2。

使用以下格式：

```
name="st" parent="fp" lun=lun fc-port-wwn="world-wide-name"
```

lun 指定驱动器的 LUN。*world-wide-name* 指定驱动器的全球名称 (World Wide Name, WWN)。

- 4 保存您所做的更改，并关闭 `st.conf` 文件。

示例 2-1 对应于磁带机的条目

下面的示例说明，有两个条目分别与连接到 LUN 0、目标 ID 分别为 1 和 2 的两个 Quantum DLT 7000 磁带机对应，如表 2-1 所示。

```
name="st" class="scsi" target=1 lun=0;
name="st" class="scsi" target=2 lun=0;
```

示例 2-2 添加支持磁带机的行

以下示例显示了支持表 2-1 中所包含的 StorageTek 9840 和 IBM ULT3580 磁带机的行。

```
name="st" parent="fp" lun=0 fc-port-wwn="500104f00043abfc"
name="st" parent="fp" lun=0 fc-port-wwn="500104f00045eeaf"
name="st" parent="fp" lun=0 fc-port-wwn="500104f000416304"
name="st" parent="fp" lun=0 fc-port-wwn="500104f000416303"
```

▼ 如何添加用于归档的磁带设备 (SAM-QFS Manager)

- 1 在左侧窗格中，单击 "Archive Media" (归档介质)。
- 2 在 "Tape Library Summary" (磁带库摘要) 窗口中，单击 "Add" (添加)。
- 3 按照向导中的步骤添加设备。

添加用于归档的库或磁光盘驱动器

`inquiry.conf` 文件中列出了所有支持的设备。

`/kernel/drv/samst.conf` 文件包含一系列 SCSI 和 FC 条目，并与 `/opt/SUNWsamfs/examples/inquiry.conf` 文件一起定义可存在于 SAM-QFS 环境中的设备。

下面的过程说明如何验证 `samst.conf` 文件中的条目，以及如何在必要时更新这个文件。

更新 `samst.conf` 文件，如下所述：

- 如果只有网络连接自动化库，则无需检验设备支持或对其进行更新。
- 如果使用 SCSI 或 FC 接口将磁带库连接到服务器，请参见第 24 页中的“如何使用 SAM-QFS Manager 在 SCSI 或 FC 环境中配置设备支持”，以搜索含有最新驱动器信息的磁带库。
- 如果直接连接库目标号大于 6 或 LUN 标识符大于 1，请参见第 25 页中的“如何为直接连接库配置设备支持”。

▼ 如何使用 SAM-QFS Manager 在 SCSI 或 FC 环境中配置设备支持

对环境中的每个设备重复此过程。

- 1 在 SAM-QFS Manager "Managed Hosts" (受管理的主机) 页面上，选择要添加库的服务器的名称。
屏幕上将显示 "File Systems Summary" (文件系统摘要) 页面。
- 2 展开 "Archive Media" (归档介质) 部分，然后选择 "Tape Libraries" (磁带库)。
屏幕上将显示 "Tape Library Summary" (磁带库摘要) 页面。
- 3 单击 "Add" (添加) 启动 "Add a Library" (添加库) 向导。
- 4 按照其中的步骤添加设备。
完成向导步骤后，`samst.conf` 文件会自动更新。

▼ 如何为直接连接库配置设备支持

- 1 创建 `/kernel/drv/samst.conf` 文件的备份副本。

```
# cp /kernel/drv/samst.conf /kernel/drv/samst.conf.orig
```

- 2 打开 `/kernel/drv/samst.conf` 文件。
- 3 添加 SCSI 连接磁光盘驱动器或 SCSI 连接库。

- a. 搜索采用以下格式的条目，以找到 SCSI 目标 ID 和 LUN 的列表：

```
name="samst" class="scsi" target=target lun=lun;
```

`target` 是所找到的每个 SCSI 驱动器的目标 ID，`lun` 是所找到的每个 SCSI 驱动器的相应 LUN。

- b. 查找与清单列表上的每个 SCSI 目标 ID 和 LUN 相对应的条目。

例如，StorageTek 9730 自动化库连接到目标 0 和 LUN 0。与该接口对应的行将如下所示：

```
name="samst" class="scsi" target=0 lun=0;
```

请注意，如果包含回车符，一些条目可能占用两行。

- 如果条目以井号 (#) 开头，请删除该字符。井号 (#) 表示该行是注释。
- 如果缺少条目，请为 SCSI 目标和 LUN 行创建条目。遵循步骤 3a 中显示的格式，并使用硬件清单列表中的信息。

- 4 为清单列表中每个 FC 连接磁光盘驱动器或 FC 连接自动化库创建一行。

使用以下格式：

```
name="samst" parent="fp" lun=lun fc-port-wwn="world-wide-name"
```

`lun` 指定驱动器的 LUN，`world-wide-name` 指定驱动器的 WWN。

以下示例显示了所添加的支持表 2-1 中的 StorageTek L700 磁带机的行。

```
name="samst" parent="fp" lun=0 fc-port-wwn="500104f00041182b"
```

- 5 保存您所做的更改，并退出 `samst.conf` 文件。

直接连接库初始化失败后的恢复

在尝试打开 `samst` 设备时出现 `ENOENT` 错误之后，直接连接库可能无法初始化。下面的示例说明如何排除 STK SL500 库，以便 `samstsamst` 设备在挂载时根据需要绑定。在 `/kernel/drv/scsi_vhci.conf` 文件中添加类似于下面的行，从而将 STK SL500 替换为与所使用的库类型对应的字符串：

```
#
device-type-scsi-options-list =
"STK      SL500", "disable-option";

disable-option = 0x7000000;
```

有关更多信息，请参见 `scsi_vhci(7D)` 手册页。

检验并实施配置

检验是否已配置所有设备并实施已修改或新设备条目。

▼ 如何检验是否已配置所有设备并实施更改

- 1 使用 `cfgadm` 命令列出包含在 SAM-QFS 环境中的设备。

例如：

```
# cfgadm -al
```

Ap_Id	Type	Receptacle	Occupant	Condition
c0	scsi-bus	connected	configured	unknown
c0::dsk/c0t6d0	CD-ROM	connected	configured	unknown
c1	fc-private	connected	configured	unknown
c1::500000e0103c3a91	disk	connected	configured	unknown
c2	scsi-bus	connected	unconfigured	unknown
c3	scsi-bus	connected	unconfigured	unknown
c4	scsi-bus	connected	configured	unknown
c4::dsk/c4t1d0	disk	connected	configured	unknown
c4::dsk/c4t2d0	disk	connected	configured	unknown
c5	fc-fabric	connected	configured	unknown
c5::100000e00222ba0b	disk	connected	unconfigured	unknown
c5::210000e08b0462e6	unknown	connected	unconfigured	unknown
c5::210100e08b2466e6	unknown	connected	unconfigured	unknown
c5::210100e08b27234f	unknown	connected	unconfigured	unknown
c5::500104f00043abfc	tape	connected	configured	unknown
c5::500104f00043bc94	tape	connected	configured	unknown
c5::500104f00045eeaf	tape	connected	configured	unknown
c5::500104f000466943	tape	connected	configured	unknown
c5::500104f00046b3d4	tape	connected	configured	unknown
c5::500104f0004738eb	tape	connected	configured	unknown
c6	fc	connected	unconfigured	unknown
c7	scsi-bus	connected	unconfigured	unknown
c8	scsi-bus	connected	unconfigured	unknown
usb0/1	usb-kbd	connected	configured	ok
usb0/2	usb-mouse	connected	configured	ok
usb0/3	unknown	empty	unconfigured	ok
usb0/4	unknown	empty	unconfigured	ok

- 2 检查输出，以确保其中显示了您希望在 SAM-QFS 环境中配置的所有设备。
如果未配置设备，请使用 `cfgadm` 命令进行配置。请参见 `cfgadm(1M)` 手册页。

您可能会收到一条类似以下内容的错误：

```
# cfmadm -c configure -o force_update c4::500104f000489fe3
cfmadm: Library error: failed to create device node: 500104f00043abfc: Device busy
```

尽管存在此错误，但 cfmadm 命令仍会处理请求。

3 重新引导系统，使对 st.conf 和 samst.conf 文件所做的更改生效。

```
# reboot -- -r
```

处理 st.conf 文件中的错误

如果在安装 SAM-QFS 软件期间未能正确地配置 st.conf 文件，则可能会出错。

sam-log 文件中的以下消息表明未对 /kernel/drv/st.conf 进行适当的更改。

```
May 18 12:38:18 baggins genu-30[374]: Tape device 31 is default
type. Update '/kernel/drv/st.conf'.
```

下面是与 sam-log 消息对应的设备日志消息：

```
1999/05/18 12:34:27*0000 Initialized. tp
1999/05/18 12:34:28*1002 Device is QUANTUM , DLT7000
1999/05/18 12:34:28*1003 Serial CX901S4929, rev 2150
1999/05/18 12:34:28*1005 Known as Linear Tape(lt)
1999/05/18 12:34:32 0000 Attached to process 374
1999/05/18 12:38:18 1006 Slot 1
1999/05/18 12:38:18 3117 Error: Device is type default. Update /kernel/drv/st.conf
```


执行其他 SAM-QFS 配置

本章概述了完成 Sun Storage Archive Manager (SAM-QFS) 或 Sun QFS 环境的配置可能需要执行的其他任务。某些任务是可选的，这取决于特定的环境。

注 - 开始之前，请按照《Sun QFS 和 Sun Storage Archive Manager 5.3 安装指南》中所述安装 Sun QFS 或 SAM-QFS 软件。

使用 NFS 客户机系统共享文件系统

本节介绍了如何与网络文件系统 (network file system, NFS) 客户机共享文件系统。

某些 NFS 挂载参数可能会影响已挂载 NFS 的 Sun Storage Archive Manager 文件系统的性能。您可以在 `/etc/vfstab` 文件中设置这些参数，如下所述：

- `timeo = n`。此值用于将 NFS 超时设置为十分之 n 秒。默认值为一秒的十分之一。出于性能考虑，请使用默认值。您可以根据自己系统的需要相应地增大或减小此值。
- `rsiz = n`。此值用于将读缓冲区大小设置为 n 字节。在 NFS 2 中，将默认值 (8192) 更改为 32768。在 NFS 3 中，保留默认值 32768 不变。
- `wsiz = n`。此值用于将写缓冲区大小设置为 n 字节。在 NFS 2 中，将默认值 (8192) 更改为 32768。在 NFS 3 中，保留默认值 32768 不变。有关这些参数的更多信息，请参见 `mount_nfs(1M)` 手册页。

▼ 如何使用 NFS 共享文件系统

此过程使用 Oracle Solaris `share` 命令实现从远程系统挂载文件系统。`share` 命令通常放置在 `/etc/dfs/dfstab` 文件中，当您进入 `init` 状态 3 时，Oracle Solaris OS 将自动执行此命令。

- 1 将 `share` 命令添加到 `/etc/dfs/dfstab` 文件。

例如：

```
# share -F nfs -o rw=client1:client2 -d "SAM-FS" /samfs1
```

- 2 使用 `ps` 命令确定 `nfs.server` 是否正在运行。

例如：

```
# ps -ef | grep nfsd
root      694      1  0   Apr 29 ?          0:36 /usr/lib/nfs/nfsd -a 16
en17     29996 29940  0 08:27:09 pts/5    0:00 grep nfsd

# ps -ef | grep mountd
root      406      1  0   Apr 29 ?          95:48 /usr/lib/autofs/automountd
root      691      1  0   Apr 29 ?          2:00 /usr/lib/nfs/mountd
en17     29998 29940  0 08:27:28 pts/5    0:00 grep mountd
```

在本样例输出中，包含 `/usr/lib/nfs` 的行表明已挂载 NFS 服务器。

- 3 如果 `nfs.server` 未运行，请将其启动：

```
# svcadm enable nfs/server
```

- 4 （可选）。如果想要立即对文件系统实现 NFS 共享，请在 `root shell` 提示符下键入 `share` 命令。

如果在 Oracle Solaris OS 引导时，NFS 共享文件系统不存在，则不会启动 NFS 服务器。

以下示例显示了用于启用 NFS 共享的命令。在该文件中添加第一个共享条目之后，您必须切换到运行级别 3。

```
# init 3
# who -r
.          run-level 3  Dec 12 14:39    3    2    2
# share
-          /samfs1 -  "SAM-FS"
```

▼ 如何在客户机中挂载文件系统

在客户机系统中，将服务器的文件系统挂载到近便的挂载点。

注 - 如果所请求的文件位于必须装入到 DLT 磁带机的卡盒上，如果所有的磁带机已满或者磁带机响应缓慢，则文件系统对 NFS 客户机请求的响应可能会有显著的延迟。因此，系统可能会生成错误而不重试该操作。为避免这种情况，建议您在挂载文件系统时，启用 `hard` 选项，或启用 `soft`、`retrans` 和 `timeo` 选项。如果使用 `soft` 选项，还需指定 `retrans=120`（或更大值）以及 `timeo=3000`。

- 1 在 NFS 客户机系统上，向 `/etc/vfstab` 文件添加一行以便将服务器的文件系统挂载到近便的挂载点。

以下示例将在 `/samfs1` 挂载点处挂载 `server:/samfs1`：

```
server:/samfs1 - /samfs1 nfs - yes hard,intr,timeo=60
```

2 保存并关闭 `/etc/vfstab` 文件。

3 运行 `mount` 命令。

例如，可以使用以下 `mount` 命令来挂载 `samfs1` 文件系统：

```
# mount /samfs1
```

如果需要，也可以让自动挂载程序执行此步骤。根据您的站点过程，将 `server:/samfs1` 添加到自动挂载程序图表中。有关更多信息，请参见 `automountd(1M)` 手册页。

编辑 defaults.conf 文件

`/opt/SUNWsamfs/examples/defaults.conf` 文件包含了可用于控制 Sun Storage Archive Manager 环境中的自动化库操作的指令。完成初始安装后，可随时更改这些设置，例如，根据站点库信息的变更而进行更改。如果您在系统运行后更改了 `defaults.conf` 文件中的信息，则必须运行命令将 `defaults.conf` 文件的更改传播到文件系统。

以下示例显示了示例 `defaults.conf` 文件中的行。此文件显示了多个可以影响自动化库配置参数。

示例 3-1 defaults.conf 文件示例

```
exported_media = unavailable
attended = yes
tape = lt
log = LOG_LOCAL7
timeout = 300
# trace
# all on
# endtrace
labels = barcodes_low
lt_delay = 10
lt_unload = 7
lt_blksize = 256
```

`/opt/SUNWsamfs/examples/defaults.conf` 文件中包含另一个样例文件。

有关文件内容的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`defaults.conf(4)`”手册页。

可以从 defaults.conf 进行控制的特性

本节介绍了两个可以从 `defaults.conf` 文件进行控制的常见特性。有关更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`defaults.conf(4)`”。

条码

如果您有一个使用条码读取器的磁带库，则可以配置系统将磁带标签设置为与条码标签的前几个或后几个字符相同。可在 `defaults.conf` 文件中设置 `labels` 指令，如下表所示。

表 3-1 defaults.conf 文件中的 labels 指令

指令	操作
<code>labels = barcodes</code>	默认设置。将条码的前六个字符作为标签。此设置可以使归档程序在已选定磁带的情况下自动标记空白介质上的新介质。
<code>labels = barcodes_low</code>	将条码的后六个字符作为标签。
<code>labels = read</code>	读取磁带的标签。此设置可以防止归档程序自动标记新介质。

如果 `labels = barcodes` 或 `labels = barcodes_low` 生效，则对于所挂载的用于执行写操作的任何磁带，如果该磁带可写、未标记且具有可读条码，Sun SAM 系统将在开始写入操作之前为该磁带写入标签。

驱动器计时值

可以使用 `dev_unload` 和 `dev_delay` 指令分别设置设备卸载以及卸载等待时间。

`dev_unload` 参数的格式如下所示：

```
dev_unload = seconds
```

对于 `dev`，按照 `mcf(4)` 手册页中所述指定设备类型。

其中的 `seconds` 用于指定运行 `unload` 命令之后希望系统等待的秒数。此处指定的时间供自动化库在卸载卡盒之前弹出卡盒、打开挡门和执行其他操作。默认值为 0。

`dev_delay` 指令的格式如下所示：

```
dev_delay = seconds
```

其中的 `dev`，用于指定 `mcf(4)` 手册页中所定义的设备类型。

其中的 `seconds`，用于指定卡盒装入和同一个卡盒可以被卸载之间的最小间隔秒数。默认值为 30。

例如：

```
# hp_delay = 10
# lt_unload = 7
```

▼ 如何定制默认值

- 1 使用 `cp` 命令将 `/opt/SUNWsamfs/examples/defaults.conf` 复制到其运行位置。

例如：

```
# cp /opt/SUNWsamfs/examples/defaults.conf /etc/opt/SUNWsamfs/defaults.conf
```

- 2 编辑文件。

编辑要更改的控制系统的行为方式的行。删除所更改行的第 1 列中的井号 (#)。

- 3 使用 `pkill` 命令将 `SIGHUP` 信号发送至 `sam-fsd` 守护进程。

例如：

```
# pkill -HUP sam-fsd
```

此命令将重新启动 `sam-fsd` 守护进程，使之识别 `defaults.conf` 文件中的更改。

配置远程通知功能

您可以对软件进行配置，以便在环境中出现了潜在问题时通知您。系统可以将消息发送至您所选择的管理站。该软件中的简单网络管理协议 (SNMP) 软件可以管理网络设备（如服务器、自动化库和驱动器）之间的信息交换过程。

Sun SAM 管理信息库 (Management Information Base, MIB) 用于定义 Sun SAM 软件能够检测的问题类型或事件。该软件可以检测配置错误、`tapealert` 事件以及其他异常的系统活动。有关更多信息，请参见 `/var/snmp/mib/SUN-SAM-MIB.mib` 文件。

以下过程介绍如何启用和禁用远程通知功能。

▼ 如何启用远程通知功能

开始之前 确保已经对管理站进行配置，并保证它可以正常运行。

- 1 检查 `/etc/hosts` 文件内容，确保已定义用于接收通知的管理站。

以下样例文件中定义了一个主机名为 `mgmtconsole` 的管理站。

```
999.9.9.9      localhost
999.999.9.999  loggerhost    loghost
999.999.9.998  mgmtconsole
999.999.9.9    samserver
```

- 如果已定义管理站，请关闭文件。
- 如果未定义管理站，请添加定义，然后保存并关闭文件。

- 2 打开 `/etc/opt/SUNWsamfs/scripts/sendtrap` 文件，然后检验 `TRAP_DESTINATION='hostname'` 指令。

该行指定将远程通知消息发送到安装有 Sun Storage Archive Manager 软件的服务器的端口 161。

- 如果要更改主机名或端口，请将 `TRAP_DESTINATION` 指令行替换为 `TRAP_DESTINATION="mgmt-console-name:port"`。请注意，在新指令中，应使用引号 ("")，而非撇号 (')。
- 如果希望将远程通知消息发送到多台主机，请按以下格式指定指令：

```
TRAP_DESTINATION="mgmt-console-name:port [mgmt-console-name:port]"
```

例如：

```
TRAP_DESTINATION="localhost:161 doodle:163 mgmt_station:1162"
```

- 3 检验 `COMMUNITY="public"` 指令。此行相当于密码，阻止未经授权查看或使用 SNMP 陷阱消息。

根据管理站的团体字符串值执行以下操作之一：

- 如果管理站的团体字符串也被设置为 `public`，则不必编辑该值。
- 如果管理站的团体字符串设置为 `public` 以外的值，请编辑此指令，用管理站中使用的值替换 `public`。

- 4 保存对 `/etc/opt/SUNWsamfs/scripts/sendtrap` 所做的更改，并退出该文件。

▼ 如何禁用远程通知功能

默认情况下，系统启用远程通知功能。如果您希望禁用远程通知功能，请执行本过程。

- 1 如果 `/etc/opt/SUNWsamfs/defaults.conf` 文件不存在，请使用 `cp` 命令将 `/opt/SUNWsamfs/examples/defaults.conf` 复制到 `/etc/opt/SUNWsamfs/defaults.conf`。

- 2 在 `/etc/opt/SUNWsamfs/defaults.conf` 文件中，找到指定 SNMP 警报的行。

```
# alerts=on
```

- 3 删除符号 `#`，并将 `on` 更改为 `off`。

编辑后，该行如下所示：

```
# alerts=off
```

- 4 保存您所做的更改，并退出该文件。

- 5 使用 `kill` 命令将 `SIGHUP` 信号发送至 `sam-fsd` 守护进程。

```
# kill -HUP sam-fsd
```

此命令将重新启动 `sam-fsd` 守护进程，使之识别 `defaults.conf` 文件中的更改。

添加管理员组

默认情况下，只有超级用户可以执行管理员命令。不过，您可以在安装期间输入管理员组名。安装期间，`pkgadd` 进程会提示您输入此组名。

管理员组成员可以执行除以下命令之外的所有管理员命令：`star`、`samfsck`、`samgrowfs`、`sammkfs` 和 `samd`。这些管理员命令位于 `/opt/SUNWsamfs/sbin` 文件中。

安装软件包后，可以使用 `set_admin` 命令添加或删除管理员组。此操作与您在安装软件包期间选择管理员组具有相同的功能。您必须以超级用户的身份登录，才能使用 `set_admin` 命令。此外，您还可以取消此选择功能，以便只允许超级用户执行 `/opt/SUNWsamfs/sbin` 中的程序。有关更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`set_admin(1M)`”。

▼ 如何添加管理员组

- 1 选择某个组名，或从环境中选择现有的组。

- 2 使用 `groupadd` 命令，或编辑 `/etc/group` 文件。

以下示例是组文件中的一个条目，它用于指定软件的管理员组。在此示例中，`samadm` 组由 `adm` 和 `operator` 用户组成。

```
samadm::1999:adm,operator
```

▼ 如何启用系统日志记录

软件使用标准 Oracle Solaris `syslog` 接口记录错误、注意事项、警告和其他消息。默认情况下，Sun SAM 工具为 `local7`。

- 1 打开 `/etc/syslog.conf` 文件。
- 2 在 `/opt/SUNWsamfs/examples/syslog.conf_changes` 文件中，找到类似于以下示例的日志记录行：

```
local7.debug /var/adm/sam-log
```

注 - 上述条目只有一行，且字段之间用制表符（而不是空格）隔开。

local7 是默认设置。如果在 `/etc/syslog.conf` 文件中指定了其他日志设置（而不是 local7），则需要编辑 `defaults.conf` 文件，重新对它进行设置。有关更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`defaults.conf(4)`”。

- 3 将 `/opt/SUNWsamfs/examples/syslog.conf_changes` 中的日志记录行添加到 `/etc/syslog.conf` 文件。

例如：

```
# cp /etc/syslog.conf /etc/syslog.conf.orig
# cat /opt/SUNWsamfs/examples/syslog.conf_changes >> /etc/syslog.conf
```

- 4 创建一个空日志文件，并向 `syslogd` 发送 HUP 信号。

例如，要在 `/var/adm/sam-log` 中创建一个日志文件，并将 HUP 发送至 `syslogd` 守护进程，请键入以下命令：

```
# touch /var/adm/sam-log
# pkill -HUP syslogd
```

有关更多信息，请参见 `syslog.conf(4)` 和 `syslogd(1M)` 手册页。

- 5 （可选）使用 `log_rotate.sh` 命令启用日志文件轮转功能。

日志文件可能会变得很大，可以使用 `log_rotate.sh` 命令帮助您管理日志文件。有关更多信息，请参见 `log_rotate.sh(1M)` 手册页。

配置其他 Sun Storage 产品

Sun SAM-QFS 的安装和配置过程到此已经完成。此时可配置相关存储产品。例如，要配置 Sun SAM-Remote 软件，请参见第 18 章，使用 [Sun SAM-Remote](#) 软件。

为网络连接自动化库创建参数文件

如果您的 Sun Storage Archive Manager (SAM-QFS) 环境中存在网络连接自动化库，则每个库必须有一个参数文件。必须先按照第 2 章，配置用于归档的存储设备中所述配置您的存储设备。

将自动化库连接到 SAM-QFS 环境的方式有两种：将它们直接连接到服务器上，或是将它们连接到此环境的网络上。通过 SCSI 或光纤通道 (Fibre Channel, FC) 连接方式连接的库称为直接连接库。通过网络连接方式连接的库称为网络连接库。本章介绍如何为要在环境中使用的每个网络连接库创建参数文件。

注 - 以下几节中的示例和讨论既提及了网络连接自动化库的参数文件，也提及了 mcf 文件。mcf 文件是 SAM-QFS 软件的主配置文件。有关创建 mcf 文件的更多信息，请参见《Sun QFS 和 Sun Storage Archive Manager 5.3 安装指南》中的第 7 章“配置文件系统环境”。本节中既提及了参数文件，也提及了 mcf 文件，因为这两个文件会相互引用。

为网络连接自动化库创建参数文件

您必须为要在环境中使用的每个网络连接库创建参数文件。

▼ 如何为网络连接自动化库创建参数文件

1 转到 `/etc/opt/SUNWsamfs` 目录。

尽管可将参数文件写入任何目录，但最常用的位置是 `/etc/opt/SUNWsamfs`。

注 - 在创建 mcf 文件时，已在参数文件中加入了全路径名。确保 mcf 文件指向您创建的参数文件的正确位置。

- 2 创建一个新文件，其名称对应于您要配置的库。
例如，对于 Sony 网络连接库，可将文件命名为 `sonyparams`。
- 3 在文件中键入一系列参数条目。
有关针对特定库类型应包含的参数条目的详细信息，请参见本章中的相关各节。

配置 StorageTek ACSLS 连接自动化库参数文件

StorageTek ACSLS 连接自动化库交互操作的方式，在很多方面都非常类似于 SAM-QFS 系统与直接连接自动化库交互操作的方式。不过，安装和配置 StorageTek ACSLS 连接自动化库的过程还需要其他的步骤。

StorageTek ACSLS 软件包用于控制自动化库。守护进程软件通过 ACSAPI 接口控制 StorageTek 自动化库。

注 – SAM-QFS Manager 支持自动搜索和配置 ACSLS 网络连接库。在 SAM-QFS Manager 中配置库之前，无需配置参数文件。有关更多信息，请参见 SAM-QFS Manager 联机帮助。

开始之前，确保符合以下情况，而且已准备好将 StorageTek ACSLS 连接自动化库添加到 SAM-QFS 环境中：

- StorageTek ACSLS 自动化库可以正常运行。
- 已安装 StorageTek ACSLS 软件包，且该软件正在运行。

有关介绍如何创建空参数文件的说明，请参见第 37 页中的“[如何为网络连接自动化库创建参数文件](#)”。

在 StorageTek 参数文件中键入一系列参数条目。

下表显示了要使用的关键字。

参数	说明
<code>access = user-id</code>	(可选) 指定 StorageTek 软件用于访问控制的用户标识值。如果未指定此参数，则访问控制字符串为空字符串，表示不存在 <code>user-id</code> 。
<code>hostname = hostname</code>	指定运行 StorageTek ACSLS 接口的服务器的主机名。
<code>portnum = portnum</code>	指定在 ACSLS 和 SAM-QFS 软件之间进行通信所使用的端口号。有关 <code>portnum</code> 参数的信息，请参见《 Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual 》中的“ <code>stk(7)</code> ”。

参数	说明
<code>ssihost = hostname</code>	指定局域网中连接到 ACSLS 主机的 SAM-QFS 服务器的名称。只有当环境中包含多宿主 SAM-QFS 服务器时，才需指定该参数。其默认值为本地主机的名称。
<code>ssi_inet_port = ssi_inet-port</code>	指定用于传入响应的固定端口号，并指定在防火墙环境中 SSI 用于传入 ACSLS 响应的端口。指定 0 或介于 1024 到 65535 之间的值。非零值强制 SSI 将此端口用于传入 ACSLS 响应。
<code>csi_hostport = csi-port</code>	指定 ACSLS 服务器上用于接收 StorageTek SSI 守护进程发送的 ACSLS 请求的端口。指定 0 或介于 1024 到 65535 之间（包含 1024 和 65535）的值。如果设置为 0，或保留未设置，系统会查询 ACSLS 服务器上的端口映射器。
<code>capid = (acs = acsnum, lsm = lsmnum, cap = capnum)</code>	就 StorageTek 库而言，指定要在指定了 <code>export -f</code> 命令时使用的卡盒访问点 (cartridge access point, CAP)。 <code>capid</code> 说明以左括号开始，随后是三个名称值对，最后是右括号。使用逗号、冒号或空格来分隔名称值对。其中的 <code>acsnum</code> ，用于根据 StorageTek 库中的配置，指定此 CAP 的异步通信服务器 (asynchronous communications server, ACS) 编号。其中的 <code>lsmnum</code> ，用于根据 StorageTek 库中的配置，指定此 CAP 的长度子网掩码 (length subnet mask, LSM) 编号。其中的 <code>capnum</code> ，用于根据 StorageTek 库中的配置，指定此 CAP 的 CAP 编号。
<code>capacity = (index = value, [index = value]...)</code>	指定所支持的卡盒的容量。使用逗号分隔名称值对，并用括号括起字符串。 其中的 <code>index</code> 用于指定所提供的 <code>media_type</code> 文件的索引，该文件位于以下 ACSLS 目录中： <code>/export/home/ACSSS/data/internal/mixed_media/media_types.dat</code> 其中的 <code>value</code> ，用于指定此类型卡盒的容量（以 1024 字节为单位）。通常，只有对于新卡盒类型的索引或者要覆盖所支持的容量时，才需要提供容量条目。
<code>device-path-name = (acs = value, lsm = value, panel = value, drive = value) [shared]</code>	指定客户机上设备的路径。为每个连接到此客户机的驱动器指定一个 <code>device-path-name =</code> 条目。此参数用于描述 StorageTek 自动化库中的驱动器。该说明以左括号开始，随后是四个 <code>keyword = value</code> 赋值对，最后是右括号。使用逗号、冒号或空格来分割 <code>keyword = value</code> 对。 <code>shared</code> 关键字是可选的，它指定可以在两台或更多台主机的两个或更多个 SAM 进程之间共享驱动器。 有关实现共享驱动器的更多信息，请参见第 45 页中的“关于共享驱动器”和 <code>stk(7)</code> 手册页。 对于 <code>value</code> 的指定，使用 ACSLS 查询驱动器命令提供的以下信息： <ul style="list-style-type: none"> ■ <code>acs</code>—在 StorageTek 库中为驱动器配置的 ACS 编号 ■ <code>lsm</code>—在 StorageTek 库中为驱动器配置的 LSM 编号 ■ <code>panel</code>—在 StorageTek 库中为驱动器配置的 PANEL 编号 ■ <code>drive</code>—在 StorageTek 库中为驱动器配置的 DRIVE 编号

以下示例显示了 StorageTek ACSLS 连接自动化库的参数文件：

```
#
# This is file: /etc/opt/SUNWsamfs/stk50
#
hostname = baggins
portnum = 50014
access = some_user # No white space allowed in user_id
ssi_inet_port = 0
csi_hostport = 0
capid = (acs=0, lsm=1, cap=0)
/dev/rmt/0cbn = (acs=0, lsm=1, panel=0, drive=1) shared
/dev/rmt/1cbn = (acs=0, lsm=1, panel=0, drive=2)
```

注 – 《Sun QFS 文件系统 5.3 配置和管理指南》中的“Oracle Solaris OS 平台上共享文件系统的配置示例”显示了一个 StorageTek ACSLS 连接自动化库参数文件示例。示例文件 `mcf` 指向 `/etc/opt/SUNWsamfs` 目录中的文件 `stk50`。

配置 ADIC/Grau 自动化库参数文件

在 SAM-QFS 环境中，ADIC/Grau 自动化库通过 `grauaci` 接口运行。该接口使用由 ADIC/Grau 提供的 DAS/ACI 3.12 接口。有关 DAS/ACI 接口的更多信息，请参见 ADIC/Grau 文档。

注 – x64 硬件平台上的 SAM-QFS 软件不支持 ADIC/Grau 网络连接库。

开始之前，确保符合以下情况，而且已准备好将 ADIC/Grau 自动化库添加到 SAM-QFS 环境中：

- ADIC/Grau 自动化库可正常运行。
- ADIC/Grau 库已经在分布式 AML 服务器 (Distributed AML Server, DAS) 上运行。
- 在此客户机的 DAS 配置文件中，`avc`（用于防止卷争用）和 `dismount` 参数已设置为 `true`。

有关介绍如何创建空参数文件的说明，请参见第 37 页中的“如何为网络连接自动化库创建参数文件”。

在 ADIC/Grau 参数文件中键入一系列参数条目。

以名称值对的形式指定这些参数，如 `keyword=value`。不同的参数用于标识各种 ADIC/Grau 自动化库、与库关联的驱动器以及服务器名。所有参数条目都区分大小写，因此所输入的值必须与在 DAS 配置文件和 `mcf` 文件中指定的值完全相同。

下表列出了 ADIC/Grau 参数文件中必须出现的参数。

参数	说明
<code>client = client-id</code>	DAS 配置文件中定义的客户机的名称。此参数是必需的。
<code>server = server-id</code>	运行 DAS 服务器代码的服务器的主机名。此参数是必需的。
<code>acidrive drive-id = path</code>	在 DAS 配置文件中配置的驱动器的名称。 <i>path</i> 指定在 mcf 文件 Equipment Identifier 字段中输入的驱动器路径。指定给客户机的每个驱动器都有一行 <code>acidrive</code> 参数行。

注释可以出现在任何行的任何位置，但它们必须以井号 (#) 开头。系统会忽略井号右侧的字符。

如果 ADIC/Grau 库中包含多种介质类型，则每一种介质类型均有介质转换器。每一个介质转换器在 DAS 配置中均应有唯一的客户机名、唯一的库目录和唯一的参数文件。

以下样例 ADIC/Grau 参数文件定义一个支持 DLT 磁带的 ADIC/Grau 自动化库，和一个支持 Hewlett-Packard 光驱的 ADIC/Grau 自动化库。

```
# This is file: /etc/opt/SUNWsamfs/grau50
#
client = DASclient
server = DAS-server
#
# the name "drive1" is from the DAS configuration file
#
acidrive drive1 = /dev/rmt/0cbn
#
# the name "drive2" is from the DAS configuration file
#
acidrive drive2 = /dev/rmt/1cbn
```

注 - 《Sun QFS 文件系统 5.3 配置和管理指南》中的“Oracle Solaris OS 平台上共享文件系统的配置示例”显示了一个 ADIC/Grau 网络连接自动化库参数文件示例。示例 mcf 文件指向 /etc/opt/SUNWsamfs 目录中的 grau50 文件。

/var/opt/SUNWsamfs/.grau 目录中包含了诊断信息，这些信息在故障排除时非常有用。

系统在此目录中创建的文件将命名为 `grau-log-eq`，其中 *eq* 是在 mcf 文件中定义的设备序号。有关更多信息，请参见《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“grauaci(7)”和《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“mcf(4)”。

配置 IBM 3494 自动化库参数文件

在 SAM-QFS 环境中，需要安装 IBM `lmcpd` 守护进程软件包，IBM 3494 自动化磁带库才可运行。您可以从 IBM 获取 IBM `lmcpd` 守护进程软件包。

注 - x64 硬件平台上的 SAM-QFS 软件不支持 IBM 3494 网络连接库。

开始之前，确保符合以下情况，而且已准备好将 IBM 3494 自动化库添加到 SAM-QFS 环境中：

- IBM 3494 自动化库可正常运行。
- 已安装 IBM lmcpsd 守护进程软件包，且该程序正在运行。
- 已经配置 `/etc/ibmatl.conf` 文件，且该文件有效。
- IBM 3494 自动化库既可作为单独的物理库，也可用作多个逻辑库。如果您将该库拆分为多个逻辑库，请为每个逻辑库创建一个参数文件。

有关介绍如何创建空参数文件的说明，请参见第 37 页中的“[如何为网络连接自动化库创建参数文件](#)”。

在 IBM 3494 参数文件中键入一系列参数条目（`keyword = value` 和 `pathname = value`）。

所有参数都区分大小写。下表说明了如何指定参数。

参数	说明
<code>name = name</code>	您（作为系统管理员）在 <code>/etc/ibmatl.conf</code> 文件中指定的名称。该名称也是该库的符号名称。必须提供此参数。无默认值。
<code>category = hexnumber</code>	<code>0x0001</code> 和 <code>0xfeff</code> 之间一个十六进制的数字。默认情况下，SAM-QFS 软件为它控制下的介质将此值设置为 4。如果您已经将物理库拆分为多个逻辑库，请确保每个逻辑库的 <code>hexnumber</code> 值互不相同。该参数指定将哪些磁带分配给哪个库。在您将介质导入库中时，它们将被添加到目录中，而且它们的 <code>category = value</code> 将更改为此 <code>category = hexnumber</code> 参数所指定的值。
<code>access = permission</code>	有效的值为 <code>shared</code> 和 <code>private</code> 。 <ul style="list-style-type: none"> ■ 如果您将库用作一个物理库，请将该参数值指定为 <code>private</code>。这是默认值。 ■ 如果您将库拆分为多个逻辑库，请将该参数值指定为 <code>shared</code>。
<code>device-pathname = device-number</code>	其中的 <code>device-pathname</code> ，用于指定驱动器的路径。与此计算机连接的库中的每个驱动器都必须有一个 <code>device-pathname</code> 条目。每个 <code>device-pathname</code> 条目必须与 <code>mcf</code> 文件中对应条目的 <code>Equipment Identifier</code> 值相匹配。 <code>device-number</code> 为 IBM 文档中所述的设备编号。您可以通过运行 IBM <code>mtlib</code> 实用程序来获取此编号。

注释可以出现在任何行的任何位置，但它们必须以井号 (#) 开头。系统会忽略井号右侧的字符。

以下样例 `/etc/ibmatl.conf` 文件使用从 IBM 提供的 `mtlib` 实用程序获取的信息。

```
#
# This is file: /etc/ibmatl.conf
# Set this file up according to the documentation supplied by IBM.
3493a 198.174.196.50 test1
```

lmcpd 守护进程运行后，使用 IBM mtlb 实用程序获取设备编号。以下示例显示了 mtlb 的输出。

```
# mtlb -l 3493a -D
0, 00145340 003590B1A00
1, 00145350 003590B1A01
```

以下示例显示了一个 IBM 3494 库的参数文件样例。

```
#
# This is file: /etc/opt/SUNWsamfs/ibm50
#
name = 3493a # From /etc/ibmatl.conf
/dev/rmt/1bn = 00145340 # From mtlb output
/dev/rmt/2bn = 00145350 # From mtlb output
access=private
category = 5
```

注 - 《Sun QFS 文件系统 5.3 配置和管理指南》中的“Oracle Solaris OS 平台上共享文件系统的配置示例”显示了一个 IBM 3494 网络连接自动化库参数文件示例。示例文件 mcf 指向 /etc/opt/SUNWsamfs 目录中的文件 ibm50。

配置 Sony 网络连接自动化库参数文件

在 SAM-QFS 环境中，Sony 网络连接自动化库通过 DZC-8000S 应用程序接口库软件包运行。该软件包用于向 PetaSite 控制器 (PetaSite Controller, PSC) 提供应用编程接口 (application programming interface, API)。有关 DZC-8000S 接口的更多信息，请参见 Sony 提供的《Sony PetaSite Application Interface Library DZC-8000S》。

注 - x64 硬件平台上的 SAM-QFS 软件不支持 Sony 网络连接库。

开始之前，确保符合以下情况，而且已准备好将 Sony 网络连接自动化库添加到 SAM-QFS 环境中：

- Sony 网络连接自动化库可以正常运行。
- 已安装 Sony PSC 配置文件，且该文件有效。

在 Sony 参数文件中键入一系列参数条目。

有关介绍如何创建空参数文件的说明，请参见第 37 页中的“如何为网络连接自动化库创建参数文件”。

不同的参数值用于标识各种 Sony 自动化库、与库关联的驱动器以及主机名。所有参数和值条目都区分大小写，因此所键入的值必须与在配置文件和 mcf 文件中指定的值完全相同。

注 - 本节所述的内容仅适用于通过 Sony DZC-8000S 接口进行网络连接的 Sony 自动化库，Sony B9 和 B35 直接连接自动化库或 Sony 8400 PetaSite 直接连接自动化库不需要参数文件。

下表列出了 Sony 参数文件中必须出现的参数。所有参数都是必需参数。

参数	说明
<code>userid = user-id</code>	介于 0 到 65535 之间（包含 0 和 65535）的数字。如果指定 0 之外的编号，则必须是 PSC ID。user-id 参数用于在初始化 PetaSite 自动化库函数期间标识用户。
<code>server = server-id</code>	运行 PSC 服务器的服务器的主机名。
<code>sonydrive drive-id = path</code>	对于 <i>drive-id</i> ，指定 PSC 配置文件中配置的驱动器盒编号。在 mcf 文件中定义的每个驱动器都有一行 sonydrive 参数行。其中的 <i>path</i> 用于指定在 mcf 文件中的 Equipment Identifier 字段中输入的驱动器路径。

注释可以出现在任何行的任何位置，但它们必须以井号 (#) 开头。系统会忽略井号右侧的字符。

以下示例显示了 Sony 网络连接自动化库的参数文件：

```
#
# This is file: /etc/opt/SUNWsamfs/sonyfile
#
# The userid identifies the user during initialization of
# the PetaSite library functions
#
userid = 65533
#
# europa is the hostname for the server running
# the DZC-8000S server code.
#
server = europa
#
# The bin numbers 1001 and 1002 are from the PSC
# configuration file.
#
sonydrive 1001 = /dev/rmt/lc1bn
sonydrive 1002 = /dev/rmt/2c1bn
```

注 - 《Sun QFS 文件系统 5.3 配置和管理指南》中的“Oracle Solaris OS 平台上共享文件系统的配置示例”显示了一个 Sony 网络连接自动化库参数文件示例。示例文件 `mcf` 指向 `/etc/opt/SUNWsamfs` 目录中的文件 `sonyfile`。

关于共享驱动器

通常，SAM-QFS 进程独占控制库的驱动器，如主机系统的 `mcf` 文件中所声明的那样。但是，在许多情况下，驱动器都是在各个 `mcf` 文件中定义的，这些文件分别由相互独立的 SAM-QFS 进程副本所使用。在进程不使用驱动器时，驱动器将保持空闲状态。

共享驱动器功能允许使用两个或更多个 `mcf` 文件来定义同一个驱动器，这使得此驱动器可用于多个 SAM-QFS 进程。但是，多个进程无法共享介质。每个 SAM-QFS 进程必须仍维护自己的 VSN 集。

共享驱动器功能有时非常有用。例如，在某个库连接到 SAM-QFS 环境中的多个主机系统时。SAM-QFS 进程对驱动器的使用状态进行协调，使库中的驱动器保持繁忙状态。

您可以配置某些网络连接库，让它们在多个主机系统上的多个 SAM-QFS 进程之间共享一个或所有介质驱动器。所有 StorageTek ACSLS 连接库都支持 SAM-QFS 环境中的共享驱动器。

要设置一个或多个共享驱动器，请在参数文件中为每个要共享的驱动器指定 `shared` 关键字。`shared` 关键字的设置因每个生产商的库而定，因此，请参见特定于供应商的部分以获取更多信息。

注 - 默认情况下，共享驱动器中的卡盒可以在卸载之前空闲 60 秒。要更改此计时，请将 `defaults.conf` 文件中的 `shared_unload` 值更改为新值（以秒为单位）。

检查库中的驱动器顺序

如果自动化库包含多个驱动器，则这些驱动器在 `mcf` 文件中列出的顺序必须与自动化库的控制器显示它们的顺序相同。库控制器识别为第一个驱动器的驱动器必须是 `mcf` 文件中该库的第一个驱动器条目，依此类推。此顺序可以不同于 `/var/adm/messages` 文件中报告的设备顺序。

本章提供了检查驱动器顺序的过程。检查过程取决于三个因素，一是您的自动化库是否具有前面板，二是它安装的是磁带机还是磁光盘驱动器，三是它是直接连接还是通过网络连接。每个过程都将库驱动器映射至 SCSI 目标 ID，再将 SCSI 目标 ID 映射至远程磁带设备。

检查库中的驱动器顺序

▼ 如何检查配有前面板的库的驱动器顺序

有些库具有用于显示驱动器信息的面板。

以下过程只是一个总体规划。实际步骤取决于您的具体库产品，因此，有关驱动器识别和目标识别的信息，请参阅供应商文档。

- 1 根据供应商文档检验驱动器顺序。
- 2 在前面板中，检查每个驱动器的 SCSI 目标 ID 或全球名称 (World Wide Name, WWN)。
- 3 记录每个驱动器和驱动器目标的报告顺序。
- 4 在 `mcf` 文件中，确保驱动器目标的顺序与自动化库控制器显示驱动器的顺序相同。

- 5 如果进行了任何更改，请检验 `mcf` 文件并测试驱动器。然后，将更改传播到系统的其余部分。

将卡盒装入驱动器之后，您可以通过用眼睛观察驱动器，也可以使用 `samu` 实用程序的 `r` 显示屏幕，来确定驱动器是否已被激活。有关更多信息，请参见《[Sun QFS 文件系统 5.3 配置和管理指南](#)》。

▼ 如何检查未配有前面板的磁带库的驱动器顺序

- 1 停止 SAM-QFS 软件，确保在本过程中未使用任何驱动器。
- 2 获取 `/dev/rmt/` 中设备的列表。

```
# ls -l /dev/rmt/?
lrwxrwxrwx 1 root root 42 Jan 10 2000 /dev/rmt/0 ->
../../../../devices/pci@1f,4000/scsi@2,1/st@2,0:
lrwxrwxrwx 1 root root 42 Jan 10 2000 /dev/rmt/1 ->
../../../../devices/pci@1f,4000/scsi@4,1/st@5,0:
lrwxrwxrwx 1 root root 42 Jan 10 2000 /dev/rmt/2 ->
../../../../devices/pci@1f,4000/scsi@4,1/st@6,0:
lrwxrwxrwx 1 root other 40 Dec 13 2000/dev/rmt/3 ->
../../../../devices/pci@1f,4000/scsi@4/st@1,0:
lrwxrwxrwx 1 root root 40 Jun 20 2001 /dev/rmt/4 ->
../../../../devices/pci@1f,4000/scsi@4/st@2,0:
lrwxrwxrwx 1 root root 40 Jun 20 2001 /dev/rmt/5 ->
../../../../devices/pci@1f,4000/scsi@4/st@3,0:
lrwxrwxrwx 1 root root 40 Jun 20 2001 /dev/rmt/6 ->
../../../../devices/pci@1f,4000/scsi@4/st@4,0:
lrwxrwxrwx 1 root root 40 Sep 14 2001 /dev/rmt/7 ->
../../../../devices/pci@1f,4000/scsi@2/st@2,0:
lrwxrwxrwx 1 root root 40 Sep 14 2001 /dev/rmt/8 ->
../../../../devices/pci@1f,4000/scsi@2/st@3,0:
lrwxrwxrwx 1 root root 40 Sep 14 2001 /dev/rmt/9 ->
../../../../devices/pci@1f,4000/scsi@2/st@4,0:
```

- 3 将磁带装入库的驱动器 1 中。
运行 `samload` 命令，确保其他驱动器未装入磁带。
- 4 通过对每个 `/dev/rmt/` 条目运行以下命令，获取驱动器和磁带位置的相关信息：

```
# mt -f /dev/rmt/x status
```

`/dev/rmt/x` 条目将返回与库的驱动器 1 相对应的信息。以下示例显示了 `mt` 命令输出，指示驱动器中有磁带。

```
# mt -f /dev/rmt/0 status
DLT 7000 tape drive tape drive:
  sense key(0x2)= Not Ready   residual= 0   retries= 0
  file no= 0   block no= 0
```

- 5 对每个库驱动器重复上述步骤。
创建一个表，用于显示与每个 `/dev/rmt/` 条目对应的库驱动器。

例如：

```
drive 1 = /dev/rmt/4 -> ../../devices/pci@1f,4000/scsi@4/st@2,0:
drive 2 = /dev/rmt/7 -> ../../devices/pci@1f,4000/scsi@2/st@2,0:
...
```

6 更新 mcf 文件，按照自动化库的控制器显示驱动器的顺序来列出驱动器。

在本例中，mcf 文件以下列项开头：

```
# Equipment      Eq  Eq    Family  Device  Additional
# Identifier     Ord Type  Set     State   Parameters
#-----
/dev/rmt/4       31  li    ibm3580 on
/dev/rmt/7       32  li    ibm3580 on
...
```

7 检验 mcf 文件并测试驱动器。

8 将更改传播到系统的其余部分。

有关更多信息，请参见《Sun QFS 文件系统 5.3 配置和管理指南》。

▼ 如何检查未配有前面板的磁光盘库的驱动器顺序

1 停止 SAM-QFS 软件，确保在本过程中未使用任何驱动器。

2 获取 /dev/samst/ 中设备的列表。

```
# ls -l /dev/samst/?
```

3 通过库前面板手动将磁光盘卡盒装入库的驱动器 1。

确保其他驱动器为空。

4 通过对每个 /dev/samst/ 条目运行以下命令，获取驱动器和磁带位置的相关信息：

```
# dd if=/dev/samst/x bs=2k isseek=3374 of=/tmp/foo count=10
```

/dev/samst/ 条目将返回与库的驱动器 1 相对应的信息。以下示例显示的状态消息表明，光盘卡盒已装入选定的设备中。

```
# dd if=/dev/samst/c0t3u0 bs=2k isseek=3374 of=/tmp/junk count=10
10+0 records in
10+0 records out
```

5 对每个库驱动器重复上述步骤。

创建一个表，用于显示与每个 /dev/samst/ 条目对应的库驱动器：

```
drive 1 = /dev/samst/4 -> ../../devices/pci@1f,4000/scsi@4/st@2,0:
drive 2 = /dev/samst/7 -> ../../devices/pci@1f,4000/scsi@2/st@2,0:
...
```

6 更新 mcf 文件，按照自动化库的控制器显示驱动器的顺序来列出驱动器。

在本例中，mcf 文件以下列项开头：

```
# Equipment      Eq  Eq      Family  Device  Additional
# Identifier      Ord Type   Set     State   Parameters
#-----
/dev/samst/4      31  li     ibm3580  on
/dev/samst/7      32  li     ibm3580  on
...
```

7 检验 mcf 文件并测试驱动器

。

8 将更改传播到系统的其余部分。

有关更多信息，请参见《Sun QFS 文件系统 5.3 配置和管理指南》。

▼ 如何检查网络连接库的驱动器顺序

1 停止 SAM-QFS 软件，确保在本过程中未使用任何驱动器。

2 获取 /dev/rmt/ 中设备的列表。

```
# ls -l /dev/rmt/*[0-9] | awk '{print $9, $10, $11}'
/dev/rmt/0 -> /devices/pci@8,700000/SUNW,qlc@4,1/fp@0,0/st@w500104f0006041f0,0:
/dev/rmt/1 -> /devices/pci@8,700000/SUNW,qlc@4,1/fp@0,0/st@w500104f0006041f3,0:
/dev/rmt/2 -> /devices/pci@8,700000/SUNW,qlc@4,1/fp@0,0/st@w500104f00043cbb8,0:
/dev/rmt/3 -> /devices/pci@8,700000/SUNW,qlc@5,1/fp@0,0/st@w500104f0006041ea,0:
/dev/rmt/4 -> /devices/pci@8,700000/SUNW,qlc@5,1/fp@0,0/st@w500104f0006041ed,0:
/dev/rmt/5 -> /devices/pci@8,700000/SUNW,qlc@4/fp@0,0/st@w500104f00060420e,0:
/dev/rmt/6 -> /devices/pci@8,700000/SUNW,qlc@4/fp@0,0/st@w500104f000604211,0:
/dev/rmt/7 -> /devices/pci@8,700000/SUNW,qlc@4/fp@0,0/st@w500104f000604214,0:
/dev/rmt/8 -> /devices/pci@8,700000/SUNW,qlc@5/fp@0,0/st@w500104f000604208,0:
/dev/rmt/9 -> /devices/pci@8,700000/SUNW,qlc@5/fp@0,0/st@w500104f00060420b,0:
```

接下来，使用 `luxadm` 输出和 `ACSL display` 命令输出将每个驱动器的序列号与库中的物理位置相关联。

3 显示每个设备的序列号。

```
# luxadm display /dev/rmt/x
```

4 使用 `ACSL display` 显示每个序列号的驱动器标识符。

```
ACSSA> display drive * -f serial_num
2007-10-11 10:49:12      Display Drive
Acs  Lsm  Panel  Drive  Serial_num
0    2    10    12    331000049255
0    2    10    13    331002044567
0    2    10    14    331002057108
0    2    10    15    331002042417
0    2    10    16    331002031352
0    2    10    17    HU92K00200
```

```
0 2 10 18 HU92K00208
0 3 10 10 1200019405
0 3 10 11 1200019442
0 3 10 12 1110150718
```

- 5 创建一个表，显示标识符之间的关系。

```
Device          SSN          Drive Identifier
/dev/rmt/0 -> 331000049255 -> (acs=0, lsm=2, panel=10, drive=12)
/dev/rmt/1 -> 331002044567 -> (acs=0, lsm=2, panel=10, drive=13)
/dev/rmt/2 -> 331002057108 -> (acs=0, lsm=2, panel=10, drive=14)
```

- 6 在 `mcf` 文件中，检验驱动器的顺序是否与表一致。

- 7 检验 `mcf` 文件并测试驱动器。

- 8 将更改传播到系统的其余部分。

有关更多信息，请参见《[Sun QFS 文件系统 5.3 配置和管理指南](#)》。

填充目录

挂载文件系统后，SAM-QFS 软件会为 mcf 文件中配置的每个自动化库创建目录。不过，如果您具有网络连接自动化库，则必须填充库目录。所采用的方法取决于您在目录中添加的卷数。

填充目录

用于填充库目录的方法取决于您在目录中添加的卷数。

▼ 如何填充具有许多卷的自动化库

可对 StorageTek ACSLS 连接自动化库、ADIC/Grau 自动化库、Sony 网络连接自动化库和 IBM 3494 自动化库执行此过程。

在创建输入文件时，请注意以下事项：

- 这个文件中的每一行都包括了四个字段。每一行用于标识一个卷。为每个卷指定插槽号、VSN、条码和介质类型。

注 - 磁带在网络连接自动化库中的插槽位置与卷在 Sun Storage Archive Manager (SAM-QFS) 库目录中的插槽编号之间没有任何关系。

- 使用空格或制表符分隔文件中的各个字段。
- 如果 VSN 中包含一个或多个空格，则 VSN 名称必须括在引号 (") 内。

1 创建一个输入文件，其中包含插槽号、卷的 VSN、条码号和介质类型。

以下示例显示了 input_vsns 样例文件。

```
0 TAPE01 "TAPE 01" lt
1 TAPE02 TAPE02 lt
2 TAPE03 TAPE03 lt
```

2 使用 `build_cat` 命令创建目录。

```
# build_cat input-file catalog-file
```

参数	说明
<code>input-file</code>	指定输入文件的名称，通常是包含 VSN 列表的文件。
<code>catalog-file</code>	指定库目录的完整路径。默认情况下，SAM-QFS 软件创建一个目录，并将其写入 <code>/var/opt/SUNWsamfs/catalog/family-set-name</code> 。 <code>family-set-name</code> 派生自此自动化库的 mcf 文件条目。另外，如果您在 mcf 文件的 Additional Parameters 字段指定了目录名，还可以将该目录文件名指定为 <code>catalog-file</code> 。

例如，您可以指定以下 `build_cat` 命令：

```
# build_cat input_vsns /var/opt/SUNWsamfs/catalog/grau50
```

▼ 如何填充具有少量卷的自动化库

可对 ADIC/Grau 自动化库、Sony 网络连接自动化库、StorageTek ACSLS 连接自动化库和 IBM 3494 自动化库执行此过程。

对每个您希望包括在此目录中的卡盒执行此过程。卡盒必须实际存在于自动化库中。如果没有卡盒，条目会记录在历史记录中。历史记录跟踪从自动化库或手动挂载的设备中导出的卡盒。有关历史记录的更多信息，请参见第 68 页中的“跟踪导出的介质—历史记录”。

● 使用 `samimport` 命令将目录条目导入默认目录。

```
# samimport -v VSN eq
```

参数	内容
VSN	指定卷的 VSN 标识符。如果 VSN 名称中包含一个或多个空格，则 VSN 名称必须括在双引号 (" ") 内。
<code>eq</code>	指定在 mcf 文件中为设备指定的设备序号。

例如：

```
# samimport -v TAPE01 50
```

▼ 如何填充 IBM 3494 自动化库

仅当您 IBM 3494 库用作物理库（在 mcf 文件中指定了 `access=private`）时，才可以使用此过程。如果您已将此库拆分为多个逻辑库，请勿使用此过程。

注 – 如果您具有一个 IBM 3494 库，该库被拆分为多个逻辑库（在 IBM 3494 参数文件中指定了 `access=shared`），请使用以上方法之一来填充目录：第 53 页中的“如何填充具有许多卷的自动化库”或第 54 页中的“如何填充具有少量卷的自动化库”。

- 将介质卡盒插入邮槽。
库将自动创建一个目录，其中包含该介质卡盒。

▼ 如何快速填充 StorageTek ACSLS 连接库

与第 53 页中的“如何填充具有许多卷的自动化库”或第 54 页中的“如何填充具有少量卷的自动化库”中所述的方法相比，此填充库目录的方法要更快一些。

- 使用 `samimport` 命令及 `-c` 和 `-s` 选项，从 VSN 池导入。
有关更多信息，请参见《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“`samimport(1M)`”。

StorageTek ACSLS 连接自动化库：常见问题和错误消息

如果 StorageTek ACSLS 连接 StorageTek 自动化库的配置文件中存在错误，则系统将会生成多条错误消息。这些示例显示了系统遇到的常见问题以及在遇到问题时生成的消息。

StorageTek ACSLS 参数文件错误

```
May 23 09:26:13 baggins stk-50[3854]: initialize:
Syntax error in stk configuration file line 4.
May 23 09:26:13 baggins stk-50[3854]: initialize:
Syntax error in stk configuration file line 5.
```

检查 StorageTek 参数文件是否有语法错误。确保每一行均以关键字或注释开头。有关 StorageTek 参数文件的更多信息，请参见《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“`stk(7)`”。

StorageTek ACSLS 库错误

以下示例显示了冻结在初始化状态的驱动器。

```
May 23 09:29:48 baggins stk-50[3854]: main: Waiting for 2 drive(s) to initialize
May 23 09:29:59 baggins stk-50[3854]: main: Waiting for 2 drive(s) to initialize
May 23 09:30:39 baggins stk-50[3854]: main: Waiting for 2 drive(s) to initialize
```

```
.
.
.
May 23 09:31:19 baggins stk-50[3854]: main: 2 drive(s) did not initialize.
```

以下示例显示了 `samu` 实用程序的 `:r` 输出。

```
ty eq status act use state vsn
sg 51 -----p 0 0% off drive set off due to ACS reported state
sg 52 -----p 0 0% off drive set off due to ACS reported state
lt 61 -----p 0 0% off drive set off due to ACS reported state
tp 62 ----- 0 0% off empty
```

冻结在初始化状态或未初始化的驱动器会指示配置错误。

- 检验 ACSLS 软件是否正在运行。
- 检验主机名。
- 使用 `ping` 命令确定是否可访问主机。
- 检查 StorageTek 参数文件中指定的 `portnum`。例如，在 ACSLS 5.3 中，用于其他应用程序的默认端口号为 50004。尝试一个更高的端口号，例如 50014。

VSN 导入错误

此示例显示了使用 `samimport` 命令将某个 VSN 导入至库目录，但 StorageTek 自动化库中没有该 VSN 后所生成的消息。只有在 ACSLS 管理的自动化库中存在该卡盒时，才能成功运行 `samimport` 命令。

```
May 20 15:09:33 baggins stk-50[6117]: view_media
returned:STATUS_VOLUME_NOT_IN_LIBRARY
May 20 15:09:33 baggins stk-50[6117]: add_to_cat_req: view_media:
failed:STATUS_VOLUME_NOT_IN_LIBRARY. A
```

`sam-stkd` 守护进程使用 `ssi.sh` 脚本来确保 SSI 守护进程副本 `ssi_so` 正在运行。如果 `ssi_so` 退出，该守护进程会启动另一个副本。如果您的站点有自己的 `ssi.sh` 版本，请修改此脚本，使其在收到 `SIGTERM` 信号后退出。守护进程发送 `SIGTERM` 信号来停止进程。`/opt/SUNWsamfs/examples/ssi.sh` 文件中包含了一个 `ssi.sh` 脚本示例。如果 `/etc/opt/SUNWsamfs/scripts/ssi.sh` 中没有包含 `ssi.sh` 脚本，则系统会在安装期间将该脚本复制到其中。

管理自动化库和手动装入的驱动器

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

本部分从多个方面介绍了在 SAM-QFS 环境中使用自动化库和手动装入的驱动器时的一些相关事项。此外，本部分还介绍了当所请求的卷不在库中时，用于提示操作员装入该卷的通知工具。

注 - SAM-QFS 软件可与许多生产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 Oracle 支持部门联系。

特定于供应商的库操作过程

某些自动化库的某些具体操作可能与本节所介绍的操作有所不同。要确定自动化库在 SAM-QFS 环境中是否具有特定于供应商的其他操作说明，请参见第 8 章，[管理特定于供应商的库](#)。

▼ 如何启动可移除介质的操作

通常，当挂载了文件系统后，可移除介质的操作便已启动。

- 手动启动可移除介质的操作，而不挂载任何文件系统。

```
# samd start
```

如果在运行 `samd start` 命令时可移除介质操作已经在运行，您会看到以下消息：

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

▼ 如何停止可移除介质的操作

可以停止可移除介质的操作，并使文件系统保持挂载状态。例如，如果要手动操作卡盒，可能需要执行此操作。

1 运行 `idle` 命令，使归档程序、回写程序和其他进程完成当前任务。

也可使用 `samu` 操作员实用程序或使用 SAM-QFS Manager 将驱动器置于空闲状态

```
# samcmd aridle
# samcmd stidle
```

注 - 如果未按照这些过程操作，可能导致磁带介质出现问题。

对每个空闲磁带机运行 `samd unload` 命令。如果尝试在不卸载空闲驱动器的情况下重新启动该驱动器，则在恢复归档、回写和其他活动时，可能发生不可预测的事件。

2 通过 `samcmd r` 命令监视磁带机活动。

a. 等待至所有磁带机均已停止。

b. 卸载磁带机。

```
# samcmd unload eq
```

其中的 `eq` 用于指定在 `mcf` 文件中为要卸载的驱动器定义的设备编号。

要卸载多个驱动器，请对每个驱动器运行 `samcmd unload`。

c. 如果驱动器为空，请通过运行 `samd stop` 命令停止可移除介质的操作。

重新启动介质操作之后，未完成的回写操作将被重新执行，而归档操作也将随之恢复。

▼ 如何打开自动化库

处于 `on` 状态的库受 SAM-QFS 系统控制，可以继续执行常规操作。打开库时，SAM-QFS 软件会执行以下操作：

- 查询设备的状态，例如磁带的位置、是否使用条码等等。
- 更新目录和其他内部结构。

● 启动一个自动化库。

```
# samcmd on eq
```

其中的 `eq` 用于指定在 `mcf` 文件中为自动化库定义的设备编号。

另外，还可使用 `samu` 或 SAM-QFS Manager 执行此任务。

▼ 如何关闭自动化库

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 SAM-QFS 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。关闭自动化库以执行以下任务：

- 仅停止该自动化库的 SAM-QFS 操作。
 - 关闭自动化库的电源。
- 关闭自动化库。

```
# samcmd off eq
```

其中的 `eq` 用于指定在 `mcf` 文件中为所访问自动化库定义的设备编号。

另外，还可使用 `samu` 或 SAM-QFS Manager 执行此任务。

▼ 如何手动装入卡盒

装入卡盒后，卡盒从存储插槽移动到驱动器，并准备好接收数据。请求卷序列名 (volume serial name, VSN) 以进行归档或回写时，将自动装入卡盒。可以通过运行以下命令之一随时装入卡盒。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。

```
1 # samcmd load eq: slot[:partition]
```

```
2 # samcmd load media-type. vsn
```

参数	定义
<code>eq</code>	指定在 <code>mcf</code> 文件中定义的驱动器设备编号。
<code>slot</code>	指定存储插槽的编号，与在库目录中识别的编号相同。
<code>media-type</code>	指定介质类型。有关有效介质类型的列表，请参见《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“ <code>mcf(4)</code> ”。
<code>partition</code>	指定磁光盘的一个面。 <code>partition</code> 值必须为 1 或 2。此参数不适用于磁带卡盒。
<code>vsn</code>	指定分配给卷的卷序列名。

另外，您还可使用 `samu` 或 SAM-QFS Manager 执行此任务。

当手动装入卡盒时，它将被装入到库中的下一个可用驱动器中。如果要使驱动器不可用于此用途，可以使用 `samu` 实用程序的 `:unavail` 命令，或者使用 SAM-QFS Manager 更改设备的状态。

注 - SAM-QFS 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。

▼ 如何手动卸载卡盒

卡盒卸载后，即从驱动器中移除。不再需要某个卷时，会自动进行卸载。即使驱动器处于 `unavail` 状态，也可以随时卸载驱动器。

- **# samcmd unload eq**

其中的 `eq` 用于指定在 `mcf` 文件中为驱动器定义的设备编号。

另外，还可使用 `samu` 或 SAM-QFS Manager 执行此任务。

标记卡盒及取消卡盒标记

如果您配有独立的磁带或光盘设备，或您的自动化库无条码读取器，则必须按照本节所述来标记卡盒。要标记卡盒，对磁带使用 `tplabel` 命令，对光盘则使用 `odlabel` 命令。这些命令将创建 SAM-QFS 软件可以读取的卡盒标签。

如果库使用条码，默认情况下将设置 `labels = barcodes`。结果是使用前六个字符作为卷序列号 (volume serial number, VSN)。

如果您的库使用条码，并且您希望让后六个字符成为卡盒的 VSN 标签，请编辑 `/etc/opt/SUNWsamfs/defaults.conf` 文件，并添加以下行：

```
labels = barcodes_low
```

当软件装入具有条码的卡盒以用于执行写操作时，它将在开始执行写操作前，在卡盒上写入标签。卡盒必须可写、未标记且具有可读条码。

▼ 如何标记或重新标记磁带



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。

- 标记磁带。

- 要标记新磁带，请使用以下 `tplabel` 命令。

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

- 要重新标记现有磁带，请使用以下 `tplabel` 命令。

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

参数	定义
<code>vsn</code>	指定卷序列名。如果是进行重新标记，新 VSN 可以与旧 VSN 相同。
<code>eq</code>	指定在 <code>mcf</code> 文件中定义的驱动器设备编号。
<code>slot</code>	指定磁带存储插槽的编号，与在库目录中识别的编号相同。此参数不适用于手动装入的驱动器。

运行上述用以标记或重新标记磁带的命令后，系统会装入磁带并确定其位置，然后写入磁带标签。

另外，您还可使用 SAM-QFS Manager 执行此任务。

示例 7-1 标记磁带

```
# tplabel -vsn TAPE01 -new 50:0
```

▼ 如何标记或重新标记光盘

● 标记光盘。

- 要标记新光盘，请使用以下 `odlabel` 命令。

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

- 要重新标记现有光盘，请使用以下 `odlabel` 命令。

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

参数	定义
<code>vsn</code>	指定卷序列名。如果是进行重新标记，新 VSN 可以与旧 VSN 相同。
<code>eq</code>	指定在 <code>mcf</code> 文件中定义的驱动器设备编号。
<code>slot</code>	指定磁盘存储插槽的编号，与在库目录中识别的编号相同。此参数不适用于手动装入的驱动器。
<code>partition</code>	指定磁光盘的一个面。partition 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，系统会装入光盘并确定其位置，然后写入光盘标签。

另外，您还可使用 SAM-QFS Manager 执行此任务。

示例 7-2 标记光盘

```
# odlabel -vsn OPTIC01 -new 30:1:1
```

▼ 如何审计卷

有时，库目录需要更新所报告的磁带或光盘上的剩余空间。`auditslot` 命令用于装入包含卷的卡盒、读取标签并更新存储插槽的库目录条目。

- 使用以下命令更新剩余空间量：

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

参数	定义
<code>-e</code>	如果介质为磁带，更新剩余空间。否则，将不会进行更新。
<code>eq</code>	指定在 <code>mcf</code> 文件中定义的驱动器设备编号。
<code>slot</code>	指定存储插槽的编号，与在库目录中识别的编号相同。此参数不适用于手动装入的驱动器。
<code>partition</code>	指定磁光盘的一个面。 <code>partition</code> 值必须为 1 或 2。此参数不适用于磁带卡盒。

有关更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`auditslot(1M)`”。

也可以使用 `samu` 实用程序的 `:audit` 命令或使用 SAM-QFS Manager 执行此任务。

▼ 如何审计直接连接自动化库

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

- 移动了自动化库中的卡盒，但没有使用 SAM-QFS 命令
- 不确定库目录的状态（例如，在断电之后）
- 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

- 对直接连接自动化库执行完全审计。

```
# samcmd audit eq
```

其中的 `eq` 用于指定在 `mcf` 文件中为自动化库定义的设备编号。

也可以使用 `samu` 实用程序的 `:audit` 命令或使用 SAM-QFS Manager 执行此任务。

使用清洁卡盒

如果硬件支持使用清洁磁带功能，则 SAM-QFS 环境也支持使用清洁磁带功能。如果某个磁带机要求清洁，系统会自动装入清洁磁带。

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` 命令将 VSN 标记为清洁磁带并设置计数。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 8 章，[管理特定于供应商的库](#)，确定是否为您的设备推荐了专门过程。

▼ 如何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 `samimport` 命令将其导入。此命令会将卡盒从邮箱移至存储插槽，并更新库目录。此外，系统将设置清洁介质标志，并根据介质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。

- 1 确保清洁卡盒的条码为 CLEAN，或者以字母 CLN 开头。
- 2 将清洁卡盒导入自动化库中。

```
# samimport eq
```

其中的 `eq` 用于指定在 `mcf` 文件中为自动化库定义的设备编号。

另外，还可使用 `samu` 或 SAM-QFS Manager 执行此任务。

▼ 如何使用没有条码的清洁卡盒

如果卡盒没有条码，则不会将其识别为清洁卡盒。导入卡盒后，必须对其进行标识。

- 1 将卡盒导入自动化库中。

```
# samimport eq
```

其中的 `eq` 用于指定在 `mcf` 文件中为自动化库定义的设备编号。

- 2 将类型更改为清洁卡盒。

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

`eq` 指定自动化库的设备编号，`slot` 指定装入清洁卡盒的插槽。

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

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

3 设置清洁循环计数。

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

eq 指定自动化库的设备编号，*slot* 指定装入清洁卡盒的插槽。

以下命令示例将卡盒的清洁计数设置为 20。

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

▼ 如何重置清洁循环次数

清洁卡盒对于次数有限的清洁循环很有帮助。SAM-QFS 系统在剩余的循环次数等于零时弹出卡盒。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。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为 10。可以使用 *samu* 实用程序的 *:v* 显示屏幕或使用 SAM-QFS Manager 来查看剩余的循环次数。

如果支持自动清洁功能，但自动化库中的所有清洁卡盒的清洁循环计数都为零，则驱动器状态设置为 *off*，并在 SAM-QFS 日志中记录一条消息。

● 重置清洁磁带的循环次数。

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

参数	定义
<i>count</i>	指定您要为清洁磁带重置的清洁循环次数。
<i>media-type</i>	指定介质类型。有关有效介质类型的列表，请参见《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“mcf(4)”。
<i>vsn</i>	指定分配给卷的卷序列名 (volume serial name, VSN)。

▼ 如何限制清洁循环次数

某些驱动器错误可能会导致重复装入清洁卡盒，直到执行完所有的清洁循环。

● 限制清洁卡盒的清洁循环次数。

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

eq 指定自动化库的设备编号，*slot* 指定载装入清洁卡盒的插槽。

示例 7-3 示例

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

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

▼ 如何自动清洁磁带机

从 Sun Storage Archive Manager 4.4 发行版开始，软件启动的磁带机清洁的默认设置为 off。可以采用下列方式之一启用自动清洁功能：

- 使用介质转换器的自动清洁功能，该功能可能需要对清洁卡盒进行特殊放置。有关指导，请参见生产商提供的文档。
- 启用 SAM-QFS 自动清洁功能：

1 根据生产商提供的文档禁用介质转换器的清洁功能。

2 编辑 `defaults.conf` 文件以添加以下行：

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

`logsense` 选项可以防止驱动器使用过期的清洁介质。如果要仅使用检测数据确定清洁介质的状态，请在 `defaults.conf` 文件中添加以下行：

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

注 - 如果使用的自动清洁功能具有一个包含两个以上驱动器的库，则每个目录至少要使用两个清洁卡盒。如果没有足够的清洁卡盒可供使用，则任何需要清洁的驱动器都将置于 DOWN 状态。

▼ 如何手动清洁磁带机

如果不能进行自动清洁，而且系统使用条码，则可以随时请求清洁驱动器。

- 清洁磁带机。

```
# cleandrive eq
```

其中的 `eq` 用于指定在 `mcf` 文件中为自动化库定义的设备编号。

清洁卡盒将装入到此驱动器中。

▼ 如何清除介质错误



注意 - 删除 `error` 标志可能会引发问题。如果不确定出错的原因以及是否可以安全删除该标志，则不要使用此过程。请与 Oracle 技术支持部门联系。

如果卡盒遇到硬件或软件错误，SAM-QFS 系统将在 VSN 目录中设置 `media error` 标志。`media error` 标志显示在 `samu` 实用程序的 `v` 显示屏幕和 SAM-QFS Manager 中。

可以清除该错误以重置标志，然后可以尝试使用卡盒。

1 清除卡盒的 `media error` 标志。

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

参数	定义
<code>media-type</code>	指定介质类型。有关有效介质类型的列表，请参见《 Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual 》中的“ <code>mcf(4)</code> ”。
<code>vsname</code>	指定分配给卷的卷序列名 (volume serial name, VSN)。

2 更新库目录的剩余空间信息。

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

参数	定义
<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>	在 <code>mcf</code> 文件中为自动化库或手动装入的驱动器定义的设备编号。
<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装入的驱动器。
<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。

有关更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`auditslot(1M)`”。

也可以使用 `samu` 实用程序的 `:audit` 命令或使用 SAM-QFS Manager 执行此任务。

▼ 如何从驱动器中取出卡住的卡盒

1 关闭自动化库中的驱动器。

```
# samcmd off eq
```

其中的 `eq` 用于指定在 `mcf` 文件中为驱动器定义的设备编号。

另外，还可使用 `samu` 或 SAM-QFS Manager 执行此步骤。

2 关闭自动化库。

```
# samcmd off eq
```

其中的 `eq` 用于指定在 `mcf` 文件中为库定义的设备编号。

另外，还可使用 `samu` 或 `SAM-QFS Manager` 执行此步骤。

3 从驱动器中物理取出卡盒。

注意不要损坏卡盒或驱动器。

4 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` 用于指定在 `mcf` 文件中为库以及驱动器定义的设备编号。

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` 命令调整库目录，以便为损坏的磁带设置占用标志。

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

参数	定义
<code>eq</code>	指定在 <code>mcf</code> 文件中为自动化库或驱动器定义的设备编号。
<code>slot</code>	指定库中存储插槽的编号，与在库目录中识别的编号相同。此参数不适用于手动装入的驱动器。

如果您从插槽中取出卡盒，并希望在以后将其放回，则必须将卡盒导入到自动化库中。

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：

- 更换卡盒。
- 将卡盒重新定位为异地存储，以备今后进行故障恢复时使用。
使用 `chmed -I` 命令指定更多信息，如卡盒的存储位置。

导入和导出卡盒时，还可更新库目录。

库目录是一个中央系统信息库，其中包含了 `SAM-QFS` 环境在查找自动化库中的卡盒时所需的所有信息。库目录文件是二进制 `UNIX` 文件系统 (`UNIX file system, UFS`) 驻留文件。该文件中的信息包括以下各项：

- 与插槽中存储的卡盒关联的一个或多个 `VSN`

- 卡盒中剩余的容量和空间
- 指示卡盒的只读、写保护、回收和其他状态信息的标志

SAM-QFS 环境处理目录的方式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。需要卡盒时，系统将查询库目录，以确定哪个插槽包含 VSN。然后系统会运行一条命令，以将卡盒从该插槽装入驱动器中。

注-SAM-QFS 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对应关系。它是已知存在于自动化库中的 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 装入驱动器。供应商的软件将定位包含该 VSN 的存储插槽。

每个自动化库都根据系统特性和供应商提供的软件来处理卡盒导入和导出。例如，对于 ACL 4/52 库，您必须先运行 `move` 命令将卡盒移至导入或导出设备中，然后才能从自动化库中导出卡盒。

网络连接自动化库使用自己的实用程序导入及导出卡盒。因此，`samimport` 和 `samexport` 命令只更新 SAM-QFS 系统所使用的库目录条目。有关导入和导出网络连接库中卡盒的信息，请参见第 8 章，[管理特定于供应商的库](#)。

跟踪导出的介质—历史记录

SAM-QFS 历史记录跟踪从自动化库或手动挂载的设备导出的卡盒。历史记录类似于一个虚拟库，但它没有已定义的硬件设备。其在以下方面与自动化库类似：

- 在 `mcf` 文件中配置

使用 `hy` 类型的设备。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
```

在上述条目中，`n+1` 为 `mcf` 文件中的最后一个设备编号加 1。要为目录使用不同的设备编号或路径名称，应在 `mcf` 文件中定义历史记录。

- 具有一个目录，该目录记录与之关联的所有卡盒的条目

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。您的站点可能希望跟踪已从库中导出的现有 SAM-QFS 卡盒。在这种情况下，必须如 [《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》](#) 中的“`build_cat(1M)`”所述从现有卡盒构建历史记录目录。

`defaults.conf` 文件中的以下配置指令将影响历史记录的操作：

- `exported_media = unavailable` 将从自动化库导出的任何卡盒都标记为对历史记录不可用。请求这些卡盒时将产生 EIO 错误。
- `attended = no` 向历史记录声明，没有可处理装入请求的操作员。如果请求装入尚未装入的卡盒，将产生 EIO 错误。
有关更多配置信息，请参见《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“`historian(7)`”和《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“`defaults.conf(4)`”。
- 可以导入和导出卡盒
导入和导出的具体操作过程因生产商而异。要确定是否为您的设备推荐了专门过程，请参见第 8 章，[管理特定于供应商的库](#)。
- 在 SAM-QFS Manager 中显示为另一个自动化库

关于从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移除卡盒。`samimport` 命令用于将卡盒从邮箱移至存储插槽。`samexport` 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 SAM-QFS 软件启动时，邮箱中含有卡盒，则软件将自动导入卡盒。

▼ 如何从具有邮箱的库导入卡盒

- 1 打开邮箱。
使用生产商建议的操作（通常是使用邮箱旁的按钮）。有时，邮箱可能只配有一个插槽（在供应商的文档中称为**邮箱**）。
- 2 将卡盒手动放入邮箱。
- 3 关闭邮箱。
- 4 导入卡盒。

```
# samimport eq
```

其中的 `eq` 用于指定在 `mcf` 文件中为库定义的设备编号。

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` 或 SAM-QFS Manager 执行此任务。

▼ 如何从具有邮箱的库导出卡盒

- 将卡盒从存储插槽移至邮箱。

使用以下格式之一：

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

参数	定义
<code>eq</code>	指定在 <code>mcf</code> 文件中为自动化库定义的设备编号。
<code>slot</code>	指定自动化库中存储插槽的编号，与在库目录中识别的编号相同。
<code>media-type</code>	指定卡盒的介质类型。有关有效介质类型的列表，请参见《 Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual 》中的“ <code>mcf(4)</code> ”。
<code>vsn</code>	指定分配给卷的卷序列名 (volume serial name, VSN)。

另外，还可使用 `samu` 或 SAM-QFS Manager 执行此步骤。

▼ 如何从没有邮箱的库导入卡盒

- 1 卸载卡盒。

```
# samcmd unload eq
```

其中的 `eq` 用于指定在 `mcf` 文件中为库定义的设备编号。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。

- 2 解除自动化库挡门的锁定，打开挡门。

- 3 将卡盒装入可用的插槽。

- 4 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。SAM-QFS 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`。

▼ 如何从没有邮箱的库导出卡盒

- 1 请使用以下命令：

```
# samcmd unload eq
```

其中的 `eq` 用于指定在 `mcf` 文件中为所访问的库定义的设备编号。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。

2 解除自动化库挡门的锁定，打开挡门。

3 从各个插槽中取出所需的卡盒。

4 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。

▼ 如何启用装入通知

SAM-QFS 软件会定期请求装入卡盒，以满足归档和回写需求。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，则需要操作员装入卡盒。如果启用了装入通知，则当必须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 将装入通知脚本从其安装位置复制到其可操作位置。

例如：

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

3 检查 `defaults.conf` 文件。

确保文件中有以下默认指令，并且这些指令未进行更改。

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

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

如果要文件归档到具有加密功能的驱动器，请根据以下注意事项规划您的归档操作：

- 请勿在库中混用不具备加密功能和具备加密功能的驱动器。
- 驱动器启用了加密之后，便无法禁用。
- 请勿在磁带上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。
- 具有加密功能的驱动器可以读取非加密数据。

手动装入驱动器操作

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

▼ 如何将卡盒装入手动装入的设备

- 按生产商提供的说明将卡盒放到驱动器中。

SAM-QFS 系统会识别已装入的卡盒，读取标签并更新单插槽目录。无需进行其他操作。

▼ 如何卸载卡盒

- 使驱动器空闲。

```
# samcmd idle eq
```

其中的 *eq* 用于指定在 *mcf* 文件中为驱动器定义的设备编号。

此命令可以确保无任何归档或回写进程处于活动状态。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信息，请参见生产商提供的说明。另外，还可使用 *samu* 或 SAM-QFS Manager 执行此任务。

▼ 如何查看库目录

- 使用 *samu* 命令查看库目录。

```
# samu :v eq
```

其中的 *eq* 用于指定在 *mcf* 文件中为库定义的设备编号。

管理特定于供应商的库

SAM-QFS 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，请使用第 7 章，[管理自动化库和手动装入的驱动器](#)中所述的操作过程进行操作。但以下库使用特定于供应商的操作过程进行操作：

- 第 73 页中的“StorageTek ACSLS 连接自动化库”
- 第 75 页中的“ADIC/Grau 自动化库”
- 第 76 页中的“IBM 3584 UltraScalable 磁带库”
- 第 77 页中的“IBM 3494 库”
- 第 78 页中的“Sony 8400 PetaSite 直接连接自动化库”
- 第 80 页中的“Sony 网络连接自动化库”

注 - 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 Oracle 的销售代表或授权服务提供商联系。

StorageTek ACSLS 连接自动化库

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 7 章，[管理自动化库和手动装入的驱动器](#)中所述的过程。

某些 StorageTek 自动化库（如 StorageTek 9730 库）使用邮槽一次只能导入和导出一个卡盒。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。带有邮箱的 StorageTek 自动化库的示例有 StorageTek 9714 和 StorageTek 9710 库。

在 StorageTek 文档中，邮箱和邮槽都被称为卡盒访问端口 (cartridge access port, CAP)。

在向 ACSLS 连接自动化库中导入及从中导出卡盒时，ACSLs 清单必须与 SAM-QFS 目录保持一致。

- 导入卡盒时，`samimport` 命令并没有将卡盒实际插入到自动化库中。您还必须运行 ACSLS 命令才能完成该操作。

- 导出卡盒时，请运行 `samexport -f` 命令指示 SAM-QFS 系统将卡盒置入 CAP 并更新目录。如果不使用 `-f` 选项，卡盒不会在 CAP 中，因此之后必须使用 ACSLS 命令完成该操作。

另外，还可使用 `samu` 或 SAM-QFS Manager 来执行导入和导出过程。有关更多信息，请参见 SAM-QFS Manager 联机帮助。

▼ 如何导入磁带

- 导入磁带。

```
# samimport -v vsn eq
```

参数	定义
<i>vsn</i>	指定分配给卷的卷序列名 (volume serial name, VSN)。
<i>eq</i>	指定在 <code>mcf</code> 文件中为所访问的设备定义的设备序号。

`samimport` 命令将导致在库目录中添加一个新的 VSN。如果历史记录中包含此 VSN，则 SAM-QFS 软件会将 VSN 信息从历史记录移至库目录。

▼ 如何使用邮箱导出磁带

- 通过使用插槽或 VSN 信息，可导出磁带卡盒。

使用以下格式之一导出磁带：

- `samexport [-f] eq: slot`
- `samexport [-f] media-type .vsn`

参数	定义
<code>-f</code>	指定 SAM-QFS 系统将卷放入卡盒访问端口 (cartridge access port, CAP) 中，并相应地更新目录。
<i>eq</i>	指定在 <code>mcf</code> 文件中为所访问的设备定义的设备序号。
<i>slot</i>	指定自动化库中存储插槽的编号，与在库目录中识别的编号相同。
<i>media-type</i>	指定介质类型。有关有效介质类型的列表，请参见《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“ <code>mcf(4)</code> ”。
<i>vsn</i>	指定分配给卷的卷序列名 (volume serial name, VSN)。

ADIC/Grau 自动化库

注 - x64 硬件平台上的 SAM-QFS 软件不支持 ADIC/Grau 网络连接库。

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 7 章，[管理自动化库和手动装入的驱动器](#)中所述的过程。

由于使用供应商提供的实用程序在 ADIC/Grau 自动化库中实际添加及从中移除卡盒，因此 SAM-QFS 接口（`samimport`、`samexport` 和 SAM-QFS Manager）仅影响库目录。

▼ 如何导入卡盒

- 1 使用 ADIC/Grau 命令将卡盒从物理上移入自动化库。
- 2 使用 `samimport` 命令更新库目录。

```
# samimport -v volser eq
```

参数	定义
<code>volser</code>	指定要添加的 <code>volser</code> 。 <code>grauaci</code> 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 <code>volser</code> 信息。
<code>eq</code>	指定在 <code>mcf</code> 文件中为所访问的设备定义的设备序号。

▼ 如何导出卡盒

- 1 使用 `samexport` 命令从库目录中删除条目。

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

参数	定义
<code>eq</code>	指定在 <code>mcf</code> 文件中为所访问的设备定义的设备序号。
<code>slot</code>	指定自动化库中存储插槽的编号，与在库目录中识别的编号相同。
<code>media-type</code>	指定介质类型。 有关有效介质类型的列表，请参见《 Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual 》中的“ <code>mcf(4)</code> ”。

参数	定义
<i>vsn</i>	指定给卷的卷序列名 (volume serial name, VSN)。

`samexport` 命令在导出每个 VSN 时更新库目录。此外，还会将每个 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。

IBM 3584 UltraScalable 磁带库

注 - x64 硬件平台上的 SAM-QFS 软件不支持 IBM 3584 UltraScalable 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 7 章，管理自动化库和手动装入的驱动器中所述的过程。

导入卡盒

启动 SAM-QFS 软件时，邮箱中的卡盒不会自动导入。

清洁驱动器

禁用自动清洁，而启用托管清洁。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。《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`ibm3584(7)`”中也对此过程进行了介绍。

分区

此库可以包含多个磁带驱动器。如果使用多个驱动器，可以将这一个物理库划分成两个、三个或四个逻辑库。如果您将此库划分成两个或更多的逻辑库，则在将 IBM 3584 库添加到 SAM-QFS 环境中之前，请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。如果手动取出卡盒并将其重新插入，则任何逻辑库均可访问该抽屉插槽。

有关在 SAM-QFS 环境中将此库用作逻辑分区库的更多信息，请参见 IBM 文档或《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`ibm3584(7)`”。

▼ 如何从逻辑库中取出卡盒

- 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 替换卡盒。
- 7 关闭挡门。

IBM 3494 库

注 - x64 硬件平台上的 SAM-QFS 软件不支持 IBM 3494 网络连接库。

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 7 章，[管理自动化库和手动装入的驱动器](#)中所述的过程。

▼ 如何导入卡盒

- 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。

 - 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=shared`，请运行 `samimport` 命令将介质添加到目录中。

▼ 如何导出卡盒

- 1 使用 `export` 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

注 - x64 硬件平台上的 SAM-QFS 软件不支持 Sony 8400 PetaSite 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）。由于邮箱插槽可用作存储插槽，因此 SAM-QFS 库目录可以跟踪邮箱插槽。此自动化库使用条码读取器。

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 直接连接自动化库，也不适用于 [第 80 页](#)中的“Sony 网络连接自动化库”。

▼ 如何导入磁带

- 1 按下自动化库前面板上的 "Open/Close"（打开/关闭）按钮，打开自动化库的挡门。
- 2 将卡盒装入邮箱插槽。
- 3 按下自动化库前面板上的 "Open/Close"（打开/关闭）按钮。
- 4 手动关闭邮箱挡门。
自动化库将检查邮箱插槽，以获取卡盒条码。如果库检测到条码存在问题，该插槽的 in 和 out 指示灯都会闪烁。
- 5 使用 `samimport` 命令，使 SAM-QFS 系统识别导入的卡盒。

```
# samimport eq
```

其中的 `eq` 用于指定在 `mcf` 文件中为所访问的设备定义的设备序号。

另外，还可使用 SAM-QFS Manager 来执行此步骤。有关更多信息，请参见 SAM-QFS Manager 联机帮助。

▼ 如何在邮箱插槽未用作存储插槽时导出磁带

- 1 将卡盒移至邮箱插槽（插槽 400-407）。

```
# move source-slot destination-slot eq
```

参数	定义
<i>source-slot</i>	指定卡盒当前所在插槽的插槽编号。
<i>destination-slot</i>	指定卡盒将要移至的插槽编号。
<i>eq</i>	指定在 mcf 文件中为所访问的设备定义的设备序号。

- 2 按下自动化库前面板上的 "Open/Close"（打开/关闭）按钮以打开挡门。
- 3 从邮箱插槽中取出卡盒。
- 4 按下自动化库前面板上的 "Open/Close"（打开/关闭）按钮。
- 5 手动关闭邮箱挡门。
- 6 运行 `samexport` 命令，以使 SAM-QFS 系统能够识别导出的卡盒。

```
# samexport eq
```

其中的 *eq* 用于指定在 mcf 文件中为所访问的设备定义的设备序号。

另外，还可使用 SAM-QFS Manager 来执行此步骤。有关更多信息，请参见 SAM-QFS Manager 联机帮助。

▼ 如何在邮箱插槽用作存储插槽时导出磁带

- 1 按下自动化库前面板上的 "Open/Close"（打开/关闭）按钮以打开挡门。
- 2 从邮箱插槽中取出卡盒。
- 3 按下自动化库前面板上的 "Open/Close"（打开/关闭）按钮。
- 4 手动关闭邮箱挡门。
- 5 运行 `samexport` 命令，以使 SAM-QFS 系统能够识别导出的卡盒。

```
# samexport eq
```

其中的 *eq* 用于指定在 mcf 文件中为所访问的设备定义的设备序号。

另外，还可使用 SAM-QFS Manager 来执行此步骤。有关更多信息，请参见 SAM-QFS Manager 联机帮助。

▼ 如何将卡盒移至另一个插槽

- 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` 命令。

```
# move eq:source-slot destination-slot
```

参数	定义
<code>eq</code>	指定在 <code>mcf</code> 文件中为所访问的设备定义的设备序号。
<code>source-slot</code>	指定卡盒当前所在插槽的插槽编号。
<code>destination-slot</code>	指定卡盒将要移至的插槽编号。

另外，还可使用 SAM-QFS Manager 来执行此步骤。有关更多信息，请参见 SAM-QFS Manager 联机帮助。

Sony 网络连接自动化库

注 - x64 硬件平台上的 SAM-QFS 软件不支持 Sony 网络连接库。

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 7 章，[管理自动化库和手动装入的驱动器](#)中所述的过程。

由于使用供应商提供的实用程序添加及移除卡盒，因此 SAM-QFS 接口（`samimport`、`samexport` 和 SAM-QFS Manager）仅影响库目录。

▼ 如何导入卡盒

- 1 使用 `Sony` 命令将卡盒从物理上移入自动化库。
- 2 使用 `samimport` 命令更新库目录。

```
# samimport -v "volser" eq
```

参数	定义
<i>volser</i>	指定要添加的 <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。 如果 <i>volser</i> 值包含空格，则必须括在引号中。
<i>eq</i>	指定在 <i>mcf</i> 文件中为所访问的库定义的设备序号。

▼ 如何导出卡盒

1 从库目录中删除条目。

导出每一个 VSN 后，`samexport` 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。

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

参数	定义
<i>eq</i>	指定在 <i>mcf</i> 文件中为所访问的设备定义的设备序号。
<i>slot</i>	指定自动化库中存储插槽的编号，与在库目录中识别的编号相同。
<i>media-type</i>	指定介质类型。有关有效介质类型的列表，请参见《 Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual 》中的“ <i>mcf(4)</i> ”。
<i>vsn</i>	指定分配给卷的卷序列名 (volume serial name, VSN)。

2 使用 Sony 命令将卡盒从物理上移出自动化库。

关于归档

归档是指将文件从文件系统复制到以下位置之一的过程：

- 驻留在可移除介质卡盒上的卷
- 另一文件系统的磁盘分区

使用 SAM-QFS 时，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。

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。本节提供了为您的站点制定归档策略的一般性指导原则，并介绍了归档程序的工作原理。有关如何配置归档程序的详细信息，请参见第 10 章，[配置归档程序](#)。

计划归档操作

归档程序使用 `archiver.cmd` 文件中的信息自动进行存储管理操作。编写此文件之前，请查看以下一般性指导原则：

- **保存归档日志。** 归档日志对于恢复数据至关重要，即使 SAM 软件无法使用时也是如此。请将这些日志保存在安全的位置，以防出现毁灭性灾难。
- **对卷范围使用正则表达式。** 允许系统将文件放置在许多不同的卷中。卷范围可使系统不间断地运行。如果使用特定卷名，归档集副本会很快充满卷。在更换介质时，这会导致 workflow 问题。
- **设置最佳归档时间间隔。** 归档时间间隔是各次文件系统扫描之间的时间。根据文件的创建和修改频率以及是否需要保存所有修改的副本，设置您的归档时间间隔。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。** 与单个文件系统相比，多个文件系统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。
- **使用目录结构组织文件系统中的文件。** 为了最大程度地提高性能，放入一个目录的文件不要多于 10,000 个。

- 始终至少制作两个文件副本，并将它们存储在不同的卷上。如果将数据放到一种类型的介质上，则在此介质出现物理问题时，您会面临数据丢失的风险。
- 备份元数据（目录结构、文件名等）。定期使用 `samfsdump`。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。如果不希望归档程序备份元数据，请在 `archiver.cmd` 文件中设置 `archivemeta=off`，然后预定 `samfsdump` 命令在 `cron` 文件中运行。

上述指导原则有助于改进文件系统和归档程序的性能，确保以最安全的方式存储数据。

预备队列

归档程序和回写程序进程均可请求装入及卸载介质。如果请求的数量超过可用于介质装入的驱动器数量，则多余的请求会发送至预备队列。

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。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`。有关使用此文件的更多信息，请参见第 169 页中的“确定预备请求的优先级”。

归档程序守护进程

SAM-QFS 具有以下归档程序守护进程和其他进程：

- `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。

如果启动 SAM-QFS，其 `sam-fsd` 守护进程将启动 `sam-archiverd` 守护进程。

`sam-archiverd` 守护进程执行 `archiver` 命令，以读取 `archiver.cmd` 文件，然后建立用于控制归档操作的表。

`sam-archiverd` 守护进程将对所挂载的每个文件系统都启动一个 `sam-arfind` 进程。卸载某个文件系统后，它会停止所关联的 `sam-arfind` 进程。

`sam-archiverd` 进程监视 `sam-arfind`，并处理来自操作员或其他进程的信号。

归档操作

对文件进行归档不需要操作人员的操作。归档程序将文件写入归档介质上的卷中。归档介质中可以包含一个或多个卷。每个卷都由称为卷序列名 (volume serial name, VSN) 的唯一标识符来标识。

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。归档集的副本在不同的卷中提供文件的复件。

要成为备选的归档或重新归档文件，必须更改文件中的数据。如果只是访问文件而未执行更改，则不会对文件归档。例如，对某个文件运行 `touch` 命令，并不会导致系统对该文件进行归档或重新归档。如果使用 `mv` 命令重命名某个文件，该文件可能会移动到不同的归档集。在这种情况下，归档程序软件将重新评估归档副本，以确定是否需要对该文件进行归档或重新归档。

注 – 由于运行 `mv` 命令只是更改文件名称，而没有更改文件数据，如果计划通过 `tar` 文件进行恢复，则此操作可能会导致灾难恢复。

归档操作受以下因素的影响：

- **归档时限。** 自上次修改文件以来所经过的时间段。用户可以为每一个归档副本定义归档时限。

使用 `touch` 命令可将其文件的默认时间基准更改成过去或将来的时间。请注意，这种做法可能导致意外的归档结果。为了避免发生问题，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的某个点。

- **归档优先级。** 此值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得出，如下所示：

$archive-priority = file-property-value \times property-multiplier$

- 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，因此，副本 2、3 和 4 的值为 0。其他属性（如归档时限和文件大小）可具有 0 或 1 之外的值。
- *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。所有属性乘数的默认值均为 0.0。有关 `-priority` 参数的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`archiver.cmd(4)`”。

`archive-priority` 和 `property-multiplier` 的值是浮点数值。

以下几节介绍了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：

- 第 86 页中的“步骤 1：确定要归档的文件”
- 第 87 页中的“步骤 2：组合归档请求”

- 第 88 页中的“步骤 3：预定归档请求”
- 第 90 页中的“步骤 4：对归档请求中的文件进行归档”
- 第 91 页中的“archiver -l 命令默认输出样例”

步骤 1：确定要归档的文件

单独的 `sam-arfind` 进程将监视每个挂载的文件系统，确定哪些文件需要归档。当某个文件的更改影响它的归档状态时，文件系统都会向此 `sam-arfind` 进程发出通知。此类更改包含文件修改、重新归档、取消归档以及重命名。

`sam-arfind` 进程将检查文件，确定该文件所属的归档集以及所要执行的操作。

`sam-arfind` 进程使用以下文件属性说明确定文件的归档集：

- 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

`sam-arfind` 进程执行以下评估来确定归档操作：

- 如果文件的一个或多个副本到达或超出归档时限，`sam-arfind` 进程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 `/var/opt/SUNWsamfs/archiver/file_sys/ArchReq` 目录中。此目录中的文件为二进制文件。要显示它们，请使用 `showqueue` 命令。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。这允许您独立地控制这两种类型文件的预定。
- 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。

可使用以下方法完成归档操作：

- 第 86 页中的“连续归档”
- 第 87 页中的“扫描归档”

连续归档

使用连续归档方法时，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。

连续归档为默认归档方法（`archiver.cmd` 文件参数为 `examine=noscan`），并按以下默认开始条件操作：

- 每两个小时进行一次归档。
- 归档一直等待，直到数据至少 90% 的 `archmax` 值已准备好进行归档。
- 归档一直等待，直到有至少 500,000 个文件要归档。

满足任一项预定开始条件时，`sam-arfind` 进程都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。

要控制归档操作的预定，可使用 `-startage`、`-startcount` 和 `-startsize` 参数为每个归档集指定开始条件。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：

- 如果创建您想要同时归档的文件需要花费 1 个小时，请将 `-startage` 参数设置为 1 小时 (`-startage 1h`)，以确保在发生归档请求前，所有文件均已创建完毕。
- 如果希望所有归档请求的数据不得小于 150 GB，请将 `-startsize` 指定为 150 GB (`-startsize 150g`)，以指示归档程序在可以归档的数据达到 150 GB 时，才进行归档。
- 如果知道可归档 3000 个文件，请指定 `-startcount 3000`，确保同时对这些文件进行归档。

有关归档参数的更多信息，请参见第 119 页中的“全局归档指令”。

扫描归档

使用扫描方法时，归档程序将定期检查文件系统，然后选择要归档的文件。要启用扫描方法和禁用连续扫描，请在 `archiver.cmd` 文件中指定 `examine=scan`。

`sam-arfind` 进程将定期扫描每个文件系统，确定哪些文件需要归档，并将它们放置在归档请求中。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，则将其文件状态标志设置为 `archdone`。在连续扫描期间，`sam-arfind` 进程将扫描 `.inodes` 文件，并只检查无 `archdone` 标志的 `inode`。

完成扫描之后，`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。

有关控制 `archdone` 标志设置的信息，请参见第 125 页中的“`setarchdone` 指令：控制 `archdone` 标志的设置”。

步骤 2：组合归档请求

无论使用的是哪种归档方法，`sam-arfind` 进程都会将每个归档请求发送至归档程序守护进程 `sam-archiverd` 进行组合。本节将介绍此组合过程。

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。可能无法同时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。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条件和特定于站点的条件，来放置归档请求中的文件。默认操作是按照全路径名将一个归档请求中的所有文件归档，这样，相同目录中的文件也会全部位于归档介质上。您可以通过指定特定于站点的条件，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。这些条件称为**归档集参数**，其检验顺序为：`-reserve`、`-sort`、`-rsort`（反向排序）和 `-drives`。有关更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`archiver.cmd(4)`”。

如果指定 `-reserve owner` 参数，`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排序。将选择属于 `owner` 的文件进行归档。剩余文件将在以后进行归档。

如果指定 `-sort` 或 `-rsort` 参数，`sam-archiverd` 守护进程将依据指定排序方法（如时限、大小或目录位置）对文件排序。

如果归档请求同时包含联机 and 脱机文件，则会首先选择联机文件进行归档。

如果没有指定的排序方法，脱机文件将按照归档副本所在的卷进行排序。此规则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回写。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回写的所有文件都将被选作首先进行归档的文件。

注 - 归档脱机文件期间，使用 `-sort` 和 `-rsort` 参数会对性能产生负面影响。如果要归档文件的顺序与脱机文件所需卷的顺序不一致，则会产生这一负面影响。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，则其他副本必须采用第一个副本的归档顺序进行归档。

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

步骤 3：预定归档请求

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

- 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，都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：

- 必须存在可用于创建文件副本的驱动器。
- 必须存在可供使用的卷，且有足够的空间来容纳归档请求中的文件。

驱动器使用量

如果为归档集指定了 `-drives` 参数，`sam-archiverd` 守护进程会将归档请求中的选定文件分配至多个驱动器。如果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` 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` 参数指定的值或为介质定义的值。有关 `-archmax` 参数和 `archmax=` 指令的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`archiver.cmd(4)`”。

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。

不同的驱动器归档文件所花的时间各不相同。`-drivemax` 参数指定在重新预定驱动器接受更多数据前，将写入该驱动器的最大字节数。可使用 `-drivemax` 参数来获得最佳的驱动器利用率。

卷使用量

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，无法装入一个卷中，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，无法装入一个卷中，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，或使用 `ovflmin=` 指令为介质指定卷溢出。有关 `-ovflmin` 参数和 `ovflmin=` 指令的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`archiver.cmd(4)`”。`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。有关定制优先级乘数的更多信息，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` 守护进程都会通过将归档优先级和与各种系统资源属性相关联的乘数相加，来确定预定优先级。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装入驱动器；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。

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。

对于每一个已回写的文件，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。这些错误包括从磁盘高速缓存读取时和向卷写入数据时发生的错误。状态的改变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。该指令如下所示：

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

`operation` 为以下各项之一：

- `read`—从磁盘读取文件。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`—回写要归档的文件。默认超时时间为 0 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分别指定写入操作的 `timeout` 值。

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。如果归档请求在该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。

archiver -l 命令默认输出样例

以下示例显示了 `archiver -l` 命令的输出样例。

```
# archiver -l
Archive media:
default:mo
media:mo archmax:5000000
media:lt archmax:5000000
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
```

归档操作的日志文件和事件日志记录

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。`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和信息性消息，该日志文件包含每一个已归档文件或自动取消归档文件的相关信息。

默认情况下，禁用该日志文件。在 `archiver.cmd` 文件中使用 `logfile=` 指令可启用日志记录及指定日志文件的名称。有关日志文件的更多信息，请参见第 95 页中的“关于 `archiver.cmd` 文件”和《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`archiver.cmd(4)`”。

以下示例显示了归档程序日志文件中的样例行。

示例 9-1 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
```

下表定义了日志中的每个字段。

表 9-1 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—已归档 ■ R—已重新归档 ■ U—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《 Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual 》中的“ mcf(4) ”。
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 <code>tar(1)</code> 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（ <code>tar</code> 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。

表 9-1 归档程序日志文件的字段 (续)

字段	示例值	内容
12	f	文件类型，具体如下： <ul style="list-style-type: none"> ■ d-目录 ■ f-常规文件 ■ l-符号链接 ■ R-可移除介质文件 ■ I-段索引 ■ S-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

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。该进程使用 `ssum` 选项标记要验证的文件和目录。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。子目录继承其父目录的数据验证属性。此命令会强制生成并使用校验和来进行归档和回写。它还会阻止释放文件，直到所有归档副本均已创建完毕，并验证了它们的校验和。只有超级用户才能针对文件或目录设置此属性。

注 - 数据验证会给回写程序资源带来额外的负担，因为数据验证请求将与常规回写请求一起放入回写程序队列中。数据验证还会导致挂载更多的磁带，因此将会影响归档程序和回写程序的性能。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。

配置归档程序

本章介绍了如何通过修改 `archiver.cmd` 文件来配置归档程序。

关于 `archiver.cmd` 文件

`archiver.cmd` 文件用于控制归档程序的行为。

归档程序将文件从文件系统复制到可移除介质卡盒的卷中，或复制到另一文件系统的磁盘分区中。您可以根据站点上的文件类型以及站点数据的保护需求，调整归档程序的操作，方法是创建一个名为 `/etc/opt/SUNWsamfs/archiver.cmd` 的归档程序命令文件。系统并不要求您创建 `archiver.cmd` 文件，但您可以根据站点的具体情况调整归档程序，从而改善归档程序的效率和性能。

默认情况下，只要启动 `sam-fsd` 进程并挂载文件系统，归档程序便开始运行。

如果 `archiver.cmd` 文件不存在，归档程序则进入 `wait` 状态。要重新启动归档程序，请使用 `samcmd arrun` 命令。如果在重新启动后未发现 `archiver.cmd` 文件，归档程序将使用以下默认设置继续运行：

- 将所有文件归档至所有已配置库的所有可用卷中。
- 为每个文件创建一个副本。
- 将所有文件的归档时限设置为 4 分钟。归档时限是指自上次修改文件后经过的时间量。
- 将归档时间间隔设置为 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。

要针对您的站点调整归档程序的操作，请在 `archiver.cmd` 中设置相应的指令。指令类似于一个命令参数，由 `archiver.cmd` 文件中的文本行组成。以下规则适用于 `archiver.cmd` 文件：

- 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。
- 井号 (`#`) 后面的任何文本均被视为注释，并且不会被检查。

- 以反斜杠字符 (\) 结尾的行将会与下一行相连接。

archiver.cmd 文件使用两种类型的指令。

有关 archiver.cmd 指令的详尽信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“archiver.cmd(4)”。

无论何时更改了 archiver.cmd 文件，均应使用 archiver(1M) 命令检查是否存在语法错误。此命令将生成所有选项的列表，并将卷、文件系统内容和错误的列表写入标准输出文件 (stdout)。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。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。

归档指令

归档指令用于指定常规归档操作，由 archiver.cmd 文件中的两大区域组成：

- 在文件顶部，**全局指令**影响在 mcf 文件中定义的所有文件系统。
- 文件下半部分包含**特定于文件系统的指令**。这些指令必须位于全局指令之后。对于任何文件系统，这些指令都优先于全局指令。特定于文件系统的指令以用于标识文件系统的 `fs= name` 指令开头。

在文件系统的指令中，还可以指定归档副本指令，以定制归档副本的数量和频率。

有关详细信息，请参见第 11 章，[归档指令 \(archiver.cmd\)](#)。

归档集指令

归档集标识一组要归档的文件，不考虑其文件系统。文件系统中的文件只能属于一个归档集。同一个归档集中的文件在大小、所有权、组和目录位置等方面符合相同的条件。

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。

归档集的指令包括：

- 分配指令—定义归档集。
- 副本参数—定义每个归档集的归档方式。
- VSN 关联指令—为归档集指定卷。
- VSN 池指令—定义卷集合。

有关详细信息，请参见第 12 章，[归档集指令 \(archiver.cmd\)](#)。

▼ 如何使用命令行创建 archiver.cmd 文件

- 如果您的站点有 /etc/opt/SUNWsamfs/archiver.cmd 文件，且系统已经在归档文件，请不要对该文件进行更改。将文件复制到可以编辑的位置，然后对其进行测试。如果该文件通过检验，请用新文件替换现有文件。
- 如果您的站点没有 archiver.cmd 文件，可在 /etc/opt/SUNWsamfs 目录中编辑该文件。

1 编辑 archiver.cmd 文件，以添加或更改用于控制站点中的归档的指令。

有关可包括在此文件中的指令的信息，请参见第 11 章，归档指令 (archiver.cmd) 和第 12 章，归档集指令 (archiver.cmd)。

2 保存并关闭 archiver.cmd 文件。

3 针对当前 SAM-QFS 环境检验该文件。

如果正在使用测试 archiver.cmd 文件，则可在 archiver(1M) 命令中使用 -c 选项并提供文件名。

```
# archiver -lv
```

4 如果您正在使用测试文件，请将它移动到 /etc/opt/SUNWsamfs/archiver.cmd。

5 使用 samd config 命令传播文件更改。

```
# samd config
```

▼ 如何使用 SAM-QFS Manager 创建 archiver.cmd 文件

当您在 SAM-QFS Manager 界面内为文件系统创建或编辑归档策略时，系统会自动创建或编辑 archiver.cmd 文件。

1 在 "Servers" (服务器) 页面上，选择您要为其创建归档策略的服务器的名称。

屏幕上将显示 "File Systems Summary" (文件系统摘要) 页面。

2 在导航树中选择 "Archive Administration" (归档管理) 节点。

屏幕上将显示 "Archive Policies Summary" (归档策略摘要) 页面。

3 单击 "New" (新建)。

屏幕上将显示 "New Archive Policy" (新建归档策略) 向导。

4 按照向导中的步骤执行操作。

有关使用 "New Archive Policy" (新建归档策略) 向导的详细说明，请参见 SAM-QFS Manager 联机帮助。

保存新建的归档策略时，系统会自动将其写入 archiver.cmd 文件中。

archiver.cmd 文件示例

本节提供了一些归档配置示例。

示例 10-1 简单 archiver.cmd 文件

以下示例显示了一个可修改的简单 archiver.cmd 文件。仅为容纳更多的归档集、副本和 VSN 使用来添加指令。

```
archiver.cmd# One file system = samfs1
archmax = sg 2G
examine = noscan
fs = samfs1
logfile = /var/opt/SUNWsamfs/log/samfs1.log
all_archset .
  1 -norelease 10m
  2 -norelease 10m
params
allsets -sort path -offline_copy stageahead -reserve set
allsets -recycle hwm 50 -recycle_mingain 90 \
-recycle_vsncount 5 -recycle_dataquantity 40G
allsets.1 -startage 6h -startsize 6G -startcount 30000
allsets.2 -startage 10h -startsize 12G \
-startcount 60000 -archmax 12G
endparams
vsns
all.1 li .*
all.2 li .*
endvsns
```

示例 10-2 高级 archiver.cmd 文件

以下示例显示了一个复杂的 archiver.cmd 文件。注释指出了各种指令类型。

```
# Global directives

archmax = li 8G
examine = noscan
scanlist_squash = on

# Limit the drives

drives = stk50 3

# File selection

fs = samfs1
logfile = /var/adm/samfs1.log
archive_archset .
  1 -norelease 5m
  2 -norelease 5m
```

示例 10-2 高级 archiver.cmd 文件 (续)

```

fs = samfs2
logfile = /var/adm/samfs2.log
archive_archset .
    1 -norelease 5m
    2 -norelease 5m

fs = samfs3
logfile = /var/adm/samfs3.log
archive_archset .
    1 -norelease 5m
    2 -norelease 5m

fs = samfs4
logfile = /var/adm/samfs4.log
archive_archset .
    1 -norelease 5m
    2 -norelease 5m

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

# The following 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 3 -drivemin 10G
allsets.2 -startage 10h -startsize 12G -startcount 90000
-archmax 12G -drives 3 -drivemin 10G
endparams

# Define VSNS for archive sets

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

```

示例 10-3 无 archiver.cmd 文件

本示例介绍了在某个 SAM-QFS 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。

以下 archiver -lv 输出表明归档程序所选的默认介质类型为 mo。只有 mo 介质可用。

```

# 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

```

示例 10-3 无 archiver.cmd 文件 (续)

以下 archiver -lv 输出表明归档程序使用了两个驱动器。它列出了 12 个卷、存储容量和具有可用空间的 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-----
```

以下 archiver -lv 输出表明归档集 samfs 既包括元数据，也包括数据文件。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。

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

以下 archiver -lv 输出表明归档集中的文件按指定的顺序归档至卷中。

```
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10-4 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。

以下示例显示了 archiver.cmd 文件的内容。

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

以下 archiver -lv 输出显示了要使用的介质和驱动器。

```
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-----
```

示例 10-4 数据文件和元数据分开归档 (续)

以下 archiver -lv 输出显示了文件系统的组织状态。超过 512000 个字节 (500 KB) 的文件将在四分钟后归档。所有其他文件在 30 秒后归档。

```
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
```

以下 archiver -lv 输出表明归档集在可移除介质之间的分配。

```
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
```

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

示例10-5 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。

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

示例 10-5 将用户文件和数据文件归档至不同的介质 (续)

```

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
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

```

示例 10-5 将用户文件和数据文件归档至不同的介质 (续)

```
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
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 10-6 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。

示例 10-6 将用户文件和数据文件归档至光盘介质 (续)

定义了四个 VSN 池。其中三个池分别用于用户、数据和项目，另一个是临时池。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 10 分钟进行一次归档，并且保存归档日志。

以下示例显示了 archiver.cmd 文件和归档程序输出。

```

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-----

```

示例 10-6 将用户文件和数据文件归档至光盘介质 (续)

```

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

```

示例 10-6 将用户文件和数据文件归档至光盘介质 (续)

```
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
```

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主机中的另一文件系统。如果为两个主机系统配置了磁盘归档，这两个系统分别充当客户机和服务器。源文件所在的主机系统为 *client*，写入归档副本的主机系统为 *server*。

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 QFS 或 SAM-QFS 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。可以创建一个、两个、三个或四个归档副本。如果要创建多个归档副本，可将其中一个归档副本写入至磁盘卷，将其他归档副本写入至可移除介质卷。另外，如果将文件归档至归档文件系统中的磁盘卷，则归档程序也会根据该文件系统的 `archiver.cmd` 文件规则对归档副本进行归档。

如果计划对站点进行磁盘归档，请考虑以下指导原则：

- 创建磁盘卷的磁盘要不同于原始文件所在的磁盘。
- 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，将第 1 个归档副本创建至磁盘卷，将第 2 个副本创建至磁带，而将第 3 个副本创建至磁光盘。

- 如果要将文件归档至服务器系统上的文件系统，则归档副本也会被写入与该服务器系统连接的库中的可移除介质卡盒。

下面的列表概述了归档至磁盘卷与归档至可移除介质的相似点和不同点：

- 磁盘卷中的归档副本不会记录在目录中。
- 磁盘卷中的归档副本不会出现在历史记录中。
- 归档至磁盘卷不依赖于 `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。
- 要归档至磁盘卷，在挂载文件系统之前，必须在 `archiver.cmd` 文件中定义磁盘归档集。如果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。

关于 `diskvols.conf` 文件

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。

在源文件所在的系统上创建 `diskvols.conf` 文件。根据归档副本写入位置的不同，`diskvols.conf` 文件可能包含以下信息：

- 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主机系统，则 `diskvols.conf` 文件将包含该服务器系统的主机名。在这种情况下，在服务器系统上创建另外一个用于指定客户机系统主机名的 `diskvols.conf` 文件。



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要格外小心。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 SAM-QFS 服务器共享任何目标目录，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。

`diskvols.conf` 文件必须包含以下信息：

`VSN-name [host-name:] path`

字段	内容
<code>VSN-name</code>	接收归档副本的磁盘 VSN 的唯一字母数字名（最多 31 个字符）。

字段	内容
<i>host-name</i>	写入归档副本的主机的名称，后跟冒号字符(:)。 如果要归档至另一主机中的磁盘，请指定目标服务器的名称。 如果接收归档副本的文件系统与源文件系统处于同一台服务器中，请勿指定主机名。
<i>path</i>	接收归档文件的目录的完整路径（相对于挂载点）。开始归档之前，此目录必须存在，并且已挂载目标文件系统。例如，如果将归档副本写入 <code>archivefs1</code> 文件系统的 <code>vsns</code> 目录，请在 <code>path</code> 字段中指定 <code>/archivefs1/vsns</code> 。创建目标目录，并且只有 <code>root</code> 用户具有对该目录的写权限。

以下附加规则适用于 `diskvols.conf` 文件：

- 每个注释行以井号 (#) 开头。系统将忽略位于 # 右侧的所有文本。
- 要延续某一行，请在该行的末尾添加一个撇号 (')。

有关更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`diskvols.conf(4)`”。

磁盘归档指令

归档至磁盘卷时，归档程序可识别用于定义归档集和回收的指令，并忽略与可移除介质卡盒相关的指令。归档程序可识别以下各节介绍的指令：

- [第 110 页](#)中的“归档集副本参数例外”
- [第 111 页](#)中的“归档指令例外”
- [第 111 页](#)中的“回收指令例外”
- [第 111 页](#)中的“`vsns` 指令”
- [第 111 页](#)中的“`clients` 和 `endclients` 指令”
- [第 112 页](#)中的“`-recycle minobs` 回收程序指令”

归档集副本参数例外

[第 137 页](#)中的“归档集副本参数”中介绍的以下参数之外的所有其他参数均有效：

- `-reserve method`
- `-tapenonstop`

要配置归档集以写入多个同步磁盘归档流，请使用 `-drives` 参数。在此配置中，以循环方式选择卷，首先选择可用空间百分比最高的卷。不过，如果指定了参数 `-fillvsns`，将首先选择剩余空间百分比最低的卷。

归档指令例外

第 11 章：归档指令 (`archiver.cmd`) 中介绍的以下指令之外的所有其他指令均有效：

- `ovflmin min-size`

回收指令例外

表 16-3 中介绍的以下指令之外的所有其他指令均有效：

- `-recycle_dataquantity size`
- `-recycle_vsncount count`
- `recycle_hwm`

vsn 指令

以下指令有效：

- `vsns` 和 `endvsns`
- `vsnpools` 和 `endvsnpools`

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 表达式关键字予以标识。还可以指定从哪些 VSN 池中选择磁盘卷，如以下示例中所示。

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

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

可在 StorageTek 5800 系统上执行磁盘归档。StorageTek 5800 是一种联机存储设备，具有集成的硬件和软件体系结构，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。`vsns` 指令中 StorageTek 5800 磁盘归档的介质缩写是 `cb`。

注 - 如果在 StorageTek 5800 上使用磁盘卷进行归档，请注意 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 StorageTek 5800 文档。

clients 和 endclients 指令

`clients` 和 `endclients` 指令有效。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。以下示例显示了这些指令的格式。其中的 `client-system`，用于指定包含源文件的客户机系统的主机名。

```
clients
client-system1
```

```
client-system2
...
endclients
```

-recycle minobs 回收程序指令

-recycle_minobs_percent 回收程序指令有效。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。默认阈值为 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。

▼ 如何在客户机主机上启用磁盘归档

在包含待归档文件的客户机主机系统上执行此过程。另外，您可以使用 SAM-QFS Manager 界面来指定归档至磁盘卷的归档策略。此操作将同时更新 archiver.cmd 文件和 diskvols.conf 文件。

注 - 当您第一次在站点上配置 Sun QFS 文件系统，并且尚未在其他主机上安装 SAM-QFS 软件时，您必须将归档副本写入到与源文件位于同一主机上的文件系统上的磁盘卷。如果您稍后在另一台主机上配置 Sun QFS 文件系统，则可以对配置文件进行相应修改。

- 1 成为客户机主机系统上的超级用户。
- 2 创建或打开文件 `/etc/opt/SUNWsamfs/archiver.cmd`。
- 3 按照以下示例中所添加磁盘归档集指令：

```
#
vsns
archset1.1 dk disk01
archset2.1 dk disk02
archset3.1 dk disk03
endvsns
```

还可在 StorageTek 5800 系统上执行磁盘归档。StorageTek 5800 是一种联机存储设备，具有集成的硬件和软件体系结构，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。vsns 指令中 StorageTek 5800 磁盘归档的介质缩写是 cb。

有关指定归档集的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“[archiver.cmd\(4\)](#)”或第 12 章，归档集指令 (archiver.cmd)。

- 4 保存并关闭 archiver.cmd 文件。
- 5 创建名为 diskvols.conf 的文件。

6 指定归档副本将要写入到的目录。

以下示例显示了用于归档三个归档集中文件的 `diskvols.conf` 文件。名为 `disk01` 和 `disk02` 的磁盘卷所在的文件系统位于名为 `otherserver` 的服务器系统中。磁盘卷 `disk03` 位于待归档文件所在的主机上。

```
# This is file sourceserver:/etc/opt/SUNWsamfs/diskvols.conf
# on the client
#
# VSN_name [host_name:] path
#
disk01 otherserver:/sam/archset1
disk02 otherserver:/sam/archset2
disk03 /sam/archset3
```

7 保存并关闭 `diskvols.conf` 文件。

8 在文件系统中创建接收归档副本的目录。

例如：

```
# mkdir sam
# cd sam
# mkdir archset1
# mkdir archset2
```

9 检验 `archiver.cmd` 文件中的语法：

```
# archiver -lv
```

10 如果发现错误，请先纠正错误，再执行以下步骤。

▼ 如何在服务器主机上配置磁盘归档

仅当用于接收归档副本的主机系统不是源文件所在的主机系统时，才需执行本过程。必须在该主机上创建至少一个 QFS 或 SAM-QFS 文件系统。如果您在同一个主机系统中创建源文件和写入归档副本，则不需要执行此过程。

注 - 您可以使用 SAM-QFS Manager 界面通过指定归档到磁盘 VSN 所使用的归档策略来启用磁盘归档。此操作将同时更新 `archiver.cmd` 文件和 `diskvols.conf` 文件。

这时，您需要创建客户机/服务器环境：

- 客户机为包含源文件的主机。
- 服务器为接收归档副本的主机。

1 成为该服务器的超级用户。

2 创建或打开文件 `/etc/opt/SUNWsamfs/archiver.cmd`。

3 编辑 `archiver.cmd` 文件以按照以下示例中所示添加磁盘归档集指令：

```
#
vsns
archset1.1 dk disk01
archset2.1 dk disk02
archset3.1 dk disk03
endvsns
```

有关指定归档集的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`archiver.cmd(4)`”或第 12 章，归档集指令 (`archiver.cmd`)。

4 保存并关闭该文件。

5 切换到想要写入归档副本的文件系统。

例如：

```
# cd /ufs1
```

6 在文件系统中创建目录。

例如：

```
# mkdir sam
# cd sam
# mkdir archset1
# mkdir archset2
```

7 创建 `/etc/opt/SUNWsamfs/diskvols.conf` 文件。

8 指定 `clients` 和 `endclients` 指令以及客户机的名称。

以下示例中客户机的名称为 `sourceserver`。

```
# This is
# file destination_server:/etc/opt/SUNWsamfs/diskvols.conf
# on the server
#
clients
sourceserver
endclients
```

9 保存并关闭 `diskvols.conf` 文件。

▼ 如何启用磁盘归档

磁盘归档功能可随时启用。该过程假定您要将磁盘归档添加到现有归档配置。

开始之前 请确保在要写入磁盘归档副本的服务器主机上，至少安装了一个 QFS 或 SAM-QFS 文件系统。

1 以超级用户的身份登录到要归档的文件所在的主机系统。

- 2 在客户机和服务器主机上启用或配置磁盘归档。

按照第 112 页中的“如何在客户机主机上启用磁盘归档”过程或第 113 页中的“如何在服务器主机上配置磁盘归档”过程操作。

- 3 在客户机主机上，传播配置文件更改，然后重新启动系统。

```
# samd config
```

- 4 如果要归档至另一台主机的磁盘，请执行以下步骤：

- a. 成为该服务器主机的超级用户。
- b. 使用 `samd config` 命令传播配置文件更改并重新启动目标系统。

- 5 如果要归档至 StorageTek 5800 系统，请升级 StorageTek 5800 元数据模式配置。

按照 StorageTek 5800 System 管理指南中介绍的过程进行操作并使用以下示例中的 XML 覆盖来定义 SAM-QFS 使用的元数据。

```
<<?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"/>
<field name="fileName" type="string" indexable="true"/>
<field name="modTime" type="time" indexable="true"/>
</namespace>
</namespace>
</namespace>
</schema>

<fsViews>
</fsViews>

</metadataConfig>
```

磁盘归档示例

以下示例显示了磁盘归档配置。

示例 1

在本示例中，标识为 `disk01`、`disk02` 和 `disk04` 的 VSN 将写入至 `pluto`，这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。

以下示例显示了驻留在客户机系统 `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
```

以下示例显示了服务器系统mars上的diskvols.conf文件。

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

以下示例显示了pluto上archiver.cmd文件的片段。

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

示例 2

在此示例中，文件/sam1/testdir0/filea位于arset0.1的归档集中，归档程序将该文件的内容复制到目标路径/sam_arch1。

以下示例显示了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
```

以下示例显示了与磁盘归档有关的archiver.cmd文件行：

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

以下示例显示了已归档至磁盘的filea文件的sls(1)命令输出。请注意copy 1的以下相关信息：

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

```
# 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。

以下示例显示了 `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
```

以下示例显示了 `mars` 上的 `diskvols.conf` 文件。

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

以下示例显示了 `archiver.cmd` 文件中与此示例相关的指令。

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


归档指令 (archiver.cmd)

本章提供了有关归档指令的详细信息。

全局归档指令

全局指令用于控制归档程序的整体操作，并可用于优化站点的操作。可以直接在 `archiver.cmd` 文件中添加全局指令，也可以使用 SAM-QFS Manager 软件指定全局指令。有关使用 SAM-QFS Manager 设置全局指令的更多信息，请参见 SAM-QFS Manager 联机帮助。

首先指定全局指令，然后再指定任何文件系统指令（`fs=` 指令）。如果归档程序检测到全局指令位于 `fs=` 指令之后，则会发出一条消息。

在 `archiver.cmd` 文件中，可通过第二个字段中是否存在等号 (=) 或者是否缺少其他字段来识别全局指令。支持以下全局指令：

- 第 120 页中的“`archivemeta` 指令：控制是否对元数据进行归档”
- 第 120 页中的“`archmax` 指令：控制归档文件的大小”
- 第 121 页中的“`bufsize` 指令：设置归档程序缓冲区大小”
- 第 121 页中的“`drives` 指令：控制用于归档的驱动器数”
- 第 122 页中的“`examine` 指令：控制归档扫描”
- 第 122 页中的“`interval` 指令：指定归档时间间隔”
- 第 123 页中的“`logfile` 指令：指定归档程序日志文件”
- 第 123 页中的“`notify` 指令：重命名事件通知脚本”
- 第 124 页中的“`ovflmin` 指令：控制卷溢出”
- 第 125 页中的“`scanlist_squash` 指令：控制扫描列表合并”
- 第 125 页中的“`setarchdone` 指令：控制 `archdone` 标志的设置”
- 第 126 页中的“`wait` 指令：延迟归档程序启动”

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，从而减少可移除介质驱动器所执行的操作。默认情况下，不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`。默认设置为 `off`。

元数据的归档过程取决于您使用的超级块是第 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，不再向归档文件添加任何用户文件。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
```

参数	定义
<code>media</code>	介质类型。有关有效介质类型的列表，请参见《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“ <code>mcf(4)</code> ”。
<code>target-size</code>	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，提前到达磁带末尾会浪费大量磁带空间。最佳做法是，不要将 `archmax` 指令设置为超过介质容量的 5%。

此外，您还可以为单个归档集设置 `archmax` 指令。

注 – 对于要归档至 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）。这些操作可以改善性能。您可以尝试不同的 `buffer-size` 值。此指令的格式如下：

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

参数	定义
<i>media</i>	介质类型。有关有效介质类型的列表，请参见《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“ <code>mcf(4)</code> ”。
<i>buffer-size</i>	一个介于 2 到 1024 的数字。默认值为 4。此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《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“ <code>defaults.conf(4)</code> ”。
<i>lock</i>	指明归档程序在创建归档副本时是否可以使用锁定的缓冲区。如果指定 <code>lock</code> ，则归档程序将在 <code>sam-arcopy</code> 操作期间在内存中的归档缓冲区上设置文件锁定。此操作可以避免由于为每个 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，才必须指定 <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，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《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“ <code>setfa(1)</code> ”、《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“ <code>sam_setfa(3)</code> ”或《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“ <code>mount_samfs(1M)</code> ”上的 <code>-O forcedirectio</code> 选项。

您可以使用归档集副本参数 `-bufsize` 和 `-lock`，以每个归档集为单位指定缓冲区大小和锁定。有关更多信息，请参见第 137 页中的“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
```

参数	定义
<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 139 页中的“为归档请求指定驱动器数：-drivemax、-drivemin 和 -drives”中对归档集副本参数 -drivemax、-drivemin 和 -drives 的说明。

examine 指令：控制归档扫描

新文件和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以立即检测文件更改。
- 扫描归档。当进行扫描归档时，归档程序会定期扫描文件系统，寻找需要更改的文件。

用于扫描归档的 examine 指令的格式如下所示：

```
examine=method
```

<i>method</i> 值	定义
noscan	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 inode 信息进行扫描。这种归档方法的性能要好于扫描归档，尤其是当文件系统中文件的数量大于 1,000,000 时。默认设置。
scan	指定扫描归档。第一次的文件系统扫描是执行目录扫描。接下来对 inode 进行扫描。
scandirs	指定仅对目录执行扫描归档。如果归档程序查找到设置了 no_archive 属性的目录，将不对该目录进行扫描。如果有未更改的文件，将它们放置在此类型目录中，可缩短归档扫描时间。
scaninodes	指定仅对 inode 执行扫描归档。

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启用归档的已挂载文件系统的状态。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且未指定 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。

如果归档程序接收到 `samu` 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《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`archiver.cmd(4)`”。有关设置挂载选项的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`mount_samfs(1M)`”。

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。默认情况下，系统不会生成此文件。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。此外，还可以为单个文件系统设置 `logfile` 指令。

示例 11-1 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来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` 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` 或 `sam-arcopy` 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` 命令将前一天的归档程序日志文件移至文件系统。

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。默认文件名为 `/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。通过对第一个参数使用以下关键字之一来调用该脚本：`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《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`archiver.sh(1M)`”。

ovflmin 指令：控制卷溢出

如果启用卷溢出功能，归档程序可创建存储在多个卷上的归档文件。如果文件大小超出指定的最小大小，归档程序会将此文件的剩余部分写入该同一类型的另一个卷上。写入至每一个卷的文件部分称为片段。

`sls` 命令将列出归档副本，显示该文件在每个卷上的每一个片段。

注 - 请慎用卷溢出功能，在检验卷溢出对您的站点所产生的影响后，再使用该功能。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。

归档程序通过 `ovflmin` 指令来控制卷溢出功能。默认情况下，归档程序会禁用卷溢出功能。要启用卷溢出功能，请在 `archiver.cmd` 文件中使用 `ovflmin` 指令。此指令的格式如下：

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

参数	定义
<i>media</i>	介质类型。有关有效介质类型的列表，请参见《 Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual 》中的“ <code>mcf(4)</code> ”。
<i>minimum-file-size</i>	触发卷溢出的最小文件大小。

此外，您还可为单个归档集设置 `ovflmin` 指令。

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《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`ssum(1)`”。

卷溢出示例

一个站点有许多文件归档在 `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。为了有效地利用卷内空间，请将 `mo` 介质的 `ovflmin` 设置成略小于最小文件大小的值。以下指令将该值设置为 150 MB：

```
ovflmin=mo 150m
```

在本示例中，在归档和回写文件时装入两个卷，因为每个文件都将溢出到另一个卷上。

以下示例显示了启用卷溢出功能时的归档程序日志文件。文件 `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分别显示在第七个和第十个字段中（第一个卷的 `7eed4.1` 和 `477609472`）。

有关归档程序日志条目的完整说明，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`archiver(1M)`”。

```
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
```

归档程序日志文件的此部分与文件 `file50` 的 `sfs -D` 输出相匹配，如以下示例中所示。

```
# sfs -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
```

scanlist_squash 指令：控制扫描列表合并

`scanlist_squash` 参数用于控制扫描列表合并。默认设置为 `off`。此参数可以是全局参数，也可以是特定于文件系统的参数。

当启用了此选项时，稍后需要使用 `sam-arfind` 扫描的具有同一父目录的两个或多个子目录中文件的扫描列表条目将被合并。这些目录向上合并为公共的父目录，这将会导致对许多子目录进行深度递归扫描。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，此合并会引发严重的性能降低问题。

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。在目录扫描过程中，还将为从不归档的文件设置 `archdone` 标志。由于评估某个文件是否进行了归档可能会影响到性能，利用 `setarchdone` 指令可控制此项活动。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。

如果将 `examine` 指令设置为 `scandirs` 或 `noscan`，则该指令的默认设置为 `off`。

wait 指令：延迟归档程序启动

`wait` 指令会促使归档程序等待来自 `samu(1M)` 或 `SAM-QFS 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，还可以为单个文件系统设置 `wait` 指令。

文件系统指令

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中。当归档程序遇到 `archiver.cmd` 文件中的 `fs=` 指令时，会对该特定文件系统应用所有后续指令。因此，请在常规指令之后放置 `fs=` 指令。

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

fs 指令：指定文件系统

使用 `fs=` 指令为特定文件系统指定操作。

例如，您可以使用此指令为每一个文件系统创建不同的日志文件。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。

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

全局指令作为文件系统指令

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。以下各节对这些指令进行了说明：

- 第 122 页中的“`interval` 指令：指定归档时间间隔”
- 第 123 页中的“`logfile` 指令：指定归档程序日志文件”
- 第 125 页中的“`scanlist_squash` 指令：控制扫描列表合并”
- 第 126 页中的“`wait` 指令：延迟归档程序启动”

归档副本指令

默认情况下，当归档集中文件的归档时限为四分钟时，归档程序将为这些文件编写一个归档副本。要更改默认行为，请使用归档副本指令。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。

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。每一个归档副本指令均采用以下格式：

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

可以按照以下几节所述通过编辑 `archiver.cmd` 文件，或者通过使用 SAM-QFS Manager 软件来指定归档副本指令。有关更多信息，请参见 SAM-QFS Manager 联机帮助。

以下几节介绍这些归档副本指令参数。

-release 指令：归档后释放磁盘空间

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

```
-release
```

示例 11-2 使用 `-release` 指令的 `archiver.cmd` 文件

在以下示例中，组 `images` 中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。

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

-norelease 指令：延迟磁盘空间释放

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。`-norelease` 指令使归档集在所有副本都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。此指令的格式如下：

```
-norelease
```

对单个副本使用 `-norelease` 指令对自动释放没有效果。

示例 11-3 使用 `-norelease` 指令的 `archiver.cmd` 文件

以下示例指定一个名为 `vault_tapes` 的归档集。系统将为其创建两个副本，然后释放与该归档集相关的磁盘高速缓存。

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

同时使用 -release 和 -norelease

要确保在完成归档集所有副本的归档后立即释放其占用的磁盘空间，请同时使用 `-release` 和 `-norelease` 指令。如果同时使用 `-release` 和 `-norelease`，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，无需等待调用释放程序。

设置归档时限

通过指定归档时限更改用于归档文件的计时。时间的后缀字符可以指定为 `h`（代表小时）或 `m`（代表分钟）等，如表 12-1 中所示。

示例 11-4 指定归档时限的 `archiver.cmd` 文件

在以下示例中，目录 `data` 中的文件将在其归档时限达到一个小时后进行归档。

```
ex_set data
1 1h
```

自动取消归档

如果为某个文件指定多个归档副本，可以只保留一个副本，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，您可以执行此操作。

示例 11-5 指定取消归档时限的 `archiver.cmd` 文件

以下示例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。如果文件已存在 10 周，归档程序将创建第二和第三个归档副本，并取消第一个副本的归档。

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

有关控制取消归档的更多方法，请参见第 12 章，[归档集指令 \(archiver.cmd\)](#)。

为元数据指定多个副本

如果需要多个元数据副本，可在 `archiver.cmd` 文件中放置副本定义，其位置是紧跟 `fs=` 指令之后。

示例 11-6 指定多个元数据副本的 `archiver.cmd` 文件

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。

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

文件系统元数据包括文件系统的路径名。如果您经常更改目录，新路径名将导致创建新归档副本，并会频繁装入为元数据指定的卷。

归档集指令 (archiver.cmd)

本章提供了有关归档集指令的信息：

- 第 131 页中的“关于归档集指令”
- 第 137 页中的“归档集副本参数”
- 第 147 页中的“VSN 关联指令”
- 第 149 页中的“VSN 池指令”

关于归档集指令

默认情况下，有以下归档集：

- 保留归档集：`no_archive` 和 `allsets`。
 - `no_archive` 归档集是系统默认定义的归档集。指定给此归档集的文件永远不会被归档。例如，可将临时目录中的文件指定给 `no_archive` 归档集。
 - `allsets` 归档集定义适用于所有归档集的参数。
- 每个文件系统都有一个同名的默认归档集，且名称无法更改。这些归档集是为控制结构信息保留的。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，您可以为符合相同条件（如大小、所有权、组或目录位置）的文件创建归档集。如果某个文件不属于您指定的任何归档集，则将其作为默认归档集的一部分进行归档。文件系统中的文件只能属于一个归档集。归档集中的所有文件均被复制到与该归档集关联的卷。

归档文件与标准 UNIX `tar` 格式兼容，以便实现与 Oracle Solaris OS 和其他 UNIX 系统的数据兼容性。由于采用了 `tar` 格式，所以当 SAM-QFS 环境中的所有数据都丢失时，可以使用标准的 UNIX 工具和命令来恢复文件。

归档集的特征由归档集指令来控制。这些指令分为以下几类：

- **分配指令**定义归档集的以下特征：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。
- **副本参数**定义如何归档每个归档集：归档进程将复制执行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。
- **VSN 关联指令**为归档集指定卷。
- **VSN 池指令**定义卷集合。

可以按照以下几节所述通过编辑 `archiver.cmd` 文件，或者通过使用 SAM-QFS Manager 软件来创建归档集。在 SAM-QFS Manager 中，**归档策略**定义归档集。有关更多信息，请参见 SAM-QFS Manager 联机帮助。

归档集分配指令

归档集分配指令按照共同特征选择要包括在归档集中的文件。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 ...]
```

参数	定义
<i>archive-set-name</i>	您的站点为归档集定义的名称。最佳做法是定义一个名称，用于标识属于归档集的各文件的共同特征。该名称需满足以下要求： <ul style="list-style-type: none"> ■ 最多 29 个字符。 ■ 包括大小写字母字符、数字 0-9 以及下划线字符 (<code>_</code>)。 ■ 而不得使用其他特殊字符或空格。 ■ 第一个字符必须是字母。 ■ 所创建的归档集不能与保留归档集 <code>no_archive</code> 或 <code>allsets</code> 的名称相同。要阻止对归档集中的文件进行归档，请将名称指定为 <code>no_archive</code>
<i>path</i>	指定文件系统挂载点的相对路径，以便将 <i>path</i> 指定的目录及其子目录中的文件包含在此归档集中。要包含文件系统的所有文件，请使用句点 (<code>.</code>)。不允许在路径的开头使用斜杠 (<code>/</code>)。
<i>search-criterion1</i> <i>search-criterion2</i>	可指定零个、一个或多个搜索条件，以将归档集限定为符合条件的文件。搜索条件包括文件时限、文件大小、文件所有权和文件名。
<i>file-attribute1</i> <i>file-attribute2</i>	可指定零个、一个或多个文件属性。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。

示例 12-1 归档集分配指令。

在本示例中，用户帐户 `hmk` 中的所有文件均归档在一个单独的归档集中。`xray` 组的各目录中所有超过 1 MB 的文件均归档在 `datafiles` 归档集中。所有其他文件均归档在 `system` 归档集中。

示例 12-1 归档集分配指令。 (续)

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

以下示例显示了阻止对 `tmp` 目录各级子目录下的文件进行归档的指令，而且不管 `tmp` 目录在文件系统中的位置如何，都是如此。

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

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

要使用上次打开文件的时间来定义归档集分配，请使用 `-access age` 特征作为其中的一个 *search-criterion* 参数。

此特征会使得在 *age* 值内未获得访问的文件重新归档到价格相对便宜的其他介质。其中的 *age* 用于指定一个整数，这个整数后跟下表所示的后缀之一。

表 12-1 文件时限后缀

后缀	定义
s	秒
m	分钟
h	小时
d	天
w	周
y	年

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。在此类情况下，使用 `-nftv` (`no file time validation`，非文件时间验证) 参数阻止对文件的访问时间和修改时间进行验证。

文件时限 *search-criterion* 参数：-after

使用 `-after date-time` 特征将最近修改或创建的文件加入到同一归档集。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。按照以下格式指定日期和时间：

```
YYYY-MM-DD[Thh:mm>:ss][Z]
```

如果未指定时间，默认时间为 00:00:00。如果包含 Z，时间为国际协调时间 (Coordinated Universal Time, UTC)。如果未包含 Z，时间为本地时间。

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

使用 **-minsize** 和 **-maxsize** 特征将归档集中的成员限定为大于或小于指定大小的那些成员。其中的 *size* 用于指定一个整数，这个整数后跟下表所示的字母之一。

表 12-2 **-minsize** 和 **-maxsize** 的后缀

字母	定义
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)

示例 12-2 使用 **-minsize** 和 **-maxsize** 特征

此示例指定所有介于 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`。

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

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

要将归档组中的成员限定为所有权和组关系，请使用 **-user name** 和 **-group name** 特征。

示例 12-3 使用 **-user** 和 **-group** 指令

在以下示例中，属于用户 `sysadmin` 的所有文件均属于归档集 `adm_set`，组名为 `marketing` 的所有文件均在归档集 `mktng_set` 中。

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

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

要指定将文件名用于归档集分配，请使用 **-name *regex*** 特征，它指定与正则表达式 *regex* 匹配的任意完整路径均为归档集的成员。

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。因此，您可以在 **-name *regex*** 字段中指定模式，以匹配文件名和路径名。

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

示例 12-4 -name 参数

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$
```

以下指令为 `satellite` 归档集选择以字符 `GEO` 开头的文件：

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

以下指令可阻止对以 `.o` 结尾的任何文件进行归档：

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

示例 12-5 使用正则表达式的模式匹配

以下示例中的归档集分配指令表示不归档用户目录或子目录中的 `fred.*`。

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

因此，不会对以下文件进行归档：

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

而将归档以下文件：

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

示例 12-6 使用正则表达式的模式匹配

以下示例中的归档集分配指令表示不归档用户主目录中的 `fred.*`，但会归档用户子目录和目录 `share/marketing` 中的 `fred.*`。在这种情况下，用户起始目录是从 `share/marketing/` 到下一个斜杠字符 `/` 之间的字符串部分。

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

不会对以下文件进行归档：

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

而将归档以下文件：

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

释放和回写 *file-attributes* 参数：**-release** 和 **-stage**

通过分别使用 `-release` 和 `-stage` 选项，可设置与归档集内文件相关联的释放和回写属性。这两项设置都优先于任何现有回写或释放属性。

`-stage` 选项的格式如下所示：

```
-stage attribute
```

属性	定义
a	联合回写本归档集中的文件。
d	重置为默认设置。
n	从不回写本归档集中的文件。

示例 12-7 归档集和文件属性

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
```

归档集中的成员冲突

无法清楚判断是否应选择某个文件将其包含在归档集中时，归档程序将采用以下规则：

- 选择归档集中最初的成员定义。先将限制最严格的分配指令放置在 `archiver.cmd` 文件中。
- 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。
- 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。

归档程序先检验特定于文件系统的指令，然后再检验全局指令。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，就默认采用此结果。

示例 12-8 存在成员冲突的 `archiver.cmd` 文件

在以下示例中，管理员不想归档两个文件系统中的所有 `.o` 文件。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。

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

示例 12-9 不存在成员冲突的 `archiver.cmd` 文件

以下示例显示了用于确保不归档两个文件系统中的任何 `.o` 文件的指令。

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

归档集副本参数

归档集副本参数定义如何归档每个归档集：数据文件、目录、符号链接、分段文件的索引和归档介质信息。

`archiver.cmd` 文件的归档集副本参数部分以 `params` 指令开头，以 `endparams` 指令结尾。

以下示例显示了归档集副本参数的格式。

```

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

```

表 12-3 归档集副本参数的参数

参数	定义
<i>archive-set-name</i>	您的站点为归档集定义的名称。最佳做法是定义一个名称，用于标识属于归档集的各文件的共同特征。该名称需满足以下要求： <ul style="list-style-type: none"> ■ 最多 29 个字符。 ■ 包括大小写字母字符、数字 0-9 以及下划线字符 (_)。 ■ 而不得使用其他特殊字符或空格。 ■ 第一个字符必须是字母。
<i>copy-number</i>	用于定义归档副本数的整数：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，可以通过使用 R 并在 <i>-param1</i> 参数中指定 VSN 来将重新归档的副本定向到特定的卷。
<i>-param1</i> <i>-param2</i>	一个或多个参数，如最大大小、缓冲区大小、驱动器数，等等。以下几小节介绍了 <code>params</code> 和 <code>endparams</code> 指令之间可以指定的参数。

要为所有归档集设置默认指令，请为归档集 `allsets` 指定指令。`allsets` 指令必须先于归档集副本的指令，因为为单个归档集副本设置的参数优先于为 `allsets` 指令设置的参数。有关 `allsets` 归档集的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`archiver.cmd(4)`”。

可以按照以下几节所述通过编辑 `archiver.cmd` 文件，或者通过使用 SAM-QFS Manager 软件来指定归档集副本参数。有关更多信息，请参见 SAM-QFS Manager 联机帮助。

以下各节介绍了除磁盘归档参数外的所有其他归档集处理参数。有关磁盘归档参数的信息，请参见第 108 页中的“关于磁盘归档”。

控制归档文件的大小：-archmax 参数

`-archmax` 参数用于为归档集设置最大文件大小。此参数的格式如下：

```
-archmax target-size
```

此参数与 `archmax` 全局指令非常相似。有关该指令以及为 `target-size` 输入的值的信
息，请参见第 138 页中的“控制归档文件的大小：-archmax 参数”。

设置归档程序缓冲区大小：-bufsize 参数

默认情况下，要归档的文件先存储在大小为该介质类型默认大小的缓冲区的内存中，然后再写入归档介质。使用 `-bufsize` 指令指定缓冲区大小。定制大小可以改善性能。此参数的格式如下：

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

默认缓冲区大小为 4，表示实际缓冲区大小为 4 与该介质类型的 `dev_blksize` 值的乘积。指定一个介于 2 和 32 之间的数。`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《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`defaults.conf(4)`”。

示例 12-10 缓冲区大小：-bufsize

```
mysset.1 -bufsize=6
```

此参数与 `bufsize=media buffer-size` 全局指令相似。有关该指令的更多信息，请参见第 139 页中的“[设置归档程序缓冲区大小：-bufsize 参数](#)”。

为归档请求指定驱动器数：-drivemax、-drivemin 和 -drives

默认情况下，归档程序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，则使用多个驱动器更为高效。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，并提高归档效率。驱动器指令的格式如下：

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

参数	定义
<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 <code>-drivemin min-size</code> 指令，则 SAM-QFS 软件仅在活动足够多时才使用多个驱动器。作为一个指导原则，可以将 <code>min-size</code> 设置得足够大，以使传输时间远远大于卡盒更改时间（装入、定位、卸载）。
<i>number</i>	用于对此归档集进行归档的驱动器数量。默认值为 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，可使用 `-drivemin` 和 `-drives` 参数。对于非常大的文件，可能会发生这种情况。

要设置这些参数，请考虑文件的创建速度、驱动器数目、装入和卸载驱动器所需的时间以及驱动器的传输速率。例如，站点将一个名为 `bigfiles` 的归档集分割至五个驱动器。可如下表中所示来分割此归档集。

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

示例 12-11 用于将归档请求分割至多个驱动器的指令

以下示例显示了如何将 10 GB 或更多的归档请求分割至五个驱动器上。

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

此外，下行可确保当归档集 `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。

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

最大化卷上的空间：-fillvsns 参数

默认情况下，归档程序在写入归档副本时，将选择一个空间足以容纳所有文件的卷。此操作将导致卷不会被完全占用。如果指定 `-fillvsns`，归档程序将归档请求拆分为较小的组，并可能使用不同的卷。

指定归档缓冲区锁定：-lock 参数

默认情况下，文件存储在一个缓冲区中，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。`-lock` 参数指明归档程序在创建归档副本时必须使用锁定的缓冲区。如果指定 `-lock`，则归档程序将在 `sam-arcopy` 操作期间在内存中的归档缓冲区上设置文件锁定。此操作可避免缓冲区进行分页，并能改善性能。

此参数的格式如下：

`-lock`

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

只有为文件启用了直接 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《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“`setfa(1)`”、《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“`sam_setfa(3)`”或《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“`mount_samfs(1M)`”上的 `-O forcedirectio` 选项。

此参数与 `bufsize` 全局指令的 `lock` 参数相似。有关此主题的更多信息，请参见第 139 页中的“设置归档程序缓冲区大小：-bufsize 参数”。

创建脱机文件的归档副本：-offline_copy 参数

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。如果在创建所有剩余归档副本前，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使用的方法。如果指定该方法，要考虑 SAM-QFS 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`

<i>method</i> 值	定义
<code>none</code>	在将文件复制到归档卷中之前，根据需要回写每个文件。默认设置。

<i>method</i> 值	定义
<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。将 <code>stage_n_window</code> 挂载选项的值增大，使其大于默认值 256 KB。有关挂载选项的信息，请参见《 Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual 》中的“ <code>mount_samfs(1M)</code> ”。
<code>stageahead</code>	在将某归档文件写入目标位置的同时，回写下一个归档文件。
<code>stageall</code>	在归档前，将所有文件回写到磁盘高速缓存。这种方法将使用一个驱动器，并假定磁盘高速缓存的空间足以容纳所有文件。

指定回收

使用回收过程来回收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。必须同时在 `archiver.cmd` 文件和 `recycler.cmd` 文件中指定指令。有关更多信息，请参见第 16 章，[配置回收程序](#)。

排序归档文件：-sort 和 -rsort 参数

默认情况下，归档集中的文件按照归档前的路径进行排序。您可以指定文件按照 `age`、`priority` 或 `size` 排序，也可以指定不排序 (`none`)。每个归档集只能使用一种排序方法。

可使用 `-rsort` 颠倒方法所指定的排序顺序。

示例 12-12 对归档集中的文件排序

示例中的第一行根据文件的时限，按从旧到新的顺序，对归档集副本 `cardiac.2` 进行排序。第二行强制根据文件的大小，按从大到小的相反顺序，对归档集副本 `catsscans` 进行排序。

```
cardiac.2 -sort age
catsscans.3 -rsort size
```

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。这一特性意味着，经常访问的数据可存储在磁盘等快速介质中，不经常访问的数据可存储在磁带中。默认情况下，文件永远不会被取消归档。

示例 12-13 控制取消归档的指令

以下示例指令指定 `arset1` 文件始终位于磁盘上，即使已超过 60 天。如果该文件未被访问的时间超过 60 天，将删除副本 1 信息。删除副本 1 信息后，所有访问请求将通过副本 2 实现，并从磁带中读取。归档程序将在磁盘上创建一个新的副本 1，60 天周期重新开始。

```
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]
```

这些示例指令同时满足以下方案中的访问和归档要求。

某位患者在医院里进行为期四周的治疗。在此期间，该患者的所有文件均位于快速介质上，其中的数据被频繁访问。这是副本 1（副本 1=mo）。两周后，该患者出院。对该患者文件的访问频率降低，并逐渐降为零。如果此患者的数据未被访问的时间超过 60 天，将取消归档 `inode` 中的副本 1 条目。仅副本 2 和副本 3 条目可用。此时，可以回收快速介质的卷供新患者使用，从而避免增加磁盘库。但六个月后，该患者又回到医院。首次访问该患者的文件时从磁带（副本 2）读取数据。为了在快速介质上获取数据，归档程序将在磁盘上创建一个新的副本 1，准备存储新信息。

控制归档文件的写入方式：`-tapenonstop` 参数

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。此外，归档程序在复制操作结束时输入归档信息。

有关 `-tapenonstop` 参数的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`archiver.cmd(4)`”。

保留卷：`-reserve` 参数

默认情况下，归档程序会将归档集副本写入由 `archiver.cmd` 文件中的卷关联部分描述的一个正则表达式指定的卷。不过，如果您要求归档集卷仅包含一个归档集中的文件，可为此保留一个卷。

请注意以下指导原则：

- 站点使用保留卷时，将会导致频繁的卡盒装入和卸载操作。

- 当文件系统的目录很多，但其中的文件较小且数量不多时，如果站点对这样的文件系统使用保留卷，则会造成归档程序将很多较小的归档文件写入每一个保留卷。这些较小的归档文件都有其自己的 tar 头文件，会降低性能。

-reserve 参数指定供某个归档集使用的卷，并为其指定一个唯一的标识符，用于将归档集与卷绑定在一起。即使某个正则表达式与该卷标识符相匹配，也不会将其指定给任何其他归档集副本。-reserve 参数的格式如下所示：

-reserve *keyword*

keyword 的值取决于您使用的格式。您可以指定同时使用一种、两种或全部三种格式。

格式	关键字	保留名称示例	说明
归档集	set	users.1// {}Data.1//	
属主	dir	proj.1/p105/ {}proj.1/p104/	dir、user 和 group 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。
	user	users.1/user5/ {}users.1/user4/	
	group	data.1/engineering/	
文件系统	fs	proj.1/p103/samfs1{ }proj.1/p104/samfs1	fs 关键字用于指定保留名称中的文件系统部分。

示例 12-14 按归档集保留卷

以下示例指定 allsets 归档集为每个归档集保留了一个卷。

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

示例 12-15 保留卷名称

以下示例指定 arset.1 归档集保留了一个卷，卷标识符通过归档集、组和文件系统创建。

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

保留卷的相关信息存储在库目录中。库目录中用于描述保留卷的那些行以 #R 字符开头，并显示介质类型、VSN、保留信息以及保留日期和时间。此信息还包括归档集、路径名和文件系统三个部分，它们之间由双斜杠 (//) 分隔。

注 - 斜杠字符并不表示路径名。它们用于分隔保留名称的各个组成部分。

示例 12-16 显示保留卷的库目录

为符合页宽，某些行已作了删减。

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

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。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。

此外，您还可以分别使用 `reserve` 和 `unreserve` 命令来保留及取消保留卷。有关这些命令的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`reserve(1M)`”和《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`unreserve(1M)`”。

重新标记某个卷后，由于卷中的归档数据实际上已被清除，因此归档程序会取消保留该卷。

可通过使用 `samu` 实用程序的 `v` 显示屏幕来显示保留信息，也可通过使用 `archiver` 或 `dump_cat` 命令以下例中所示的格式之一进行显示：

```
# archiver -lv
# dump_cat -V _catalog-name_
```

设置归档优先级：-priority 参数

启用归档的文件系统提供用于归档文件的优先级。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。结果是按先发现先归档的顺序对文件进行归档。要更改文件的归档顺序，请设置优先级和排序方法。新优先级的示例包括：

- 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，以减少介质装入次数。

- 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 12-4 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>	已装入归档卷属性乘数
<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>	已装入回写卷属性乘数
<code>-priority stage_overflow value</code>	多个回写卷属性乘数

其中的 *value*，用于指定一个以下范围之内的浮点数：

`-3.400000000E+38 <= _value_ <= 3.402823466E+38`

有关优先级的更多信息，请参见《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“archiver(1M)”和《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“archiver.cmd(4)”。

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

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 12-5 -startage、-startcount 和 -startsize 指令

指令	定义
-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。按照第 128 页中的“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。当归档请求中的文件数量达到该值时，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该值时，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。这些值优先于 `examine=method` 指定值（如果已指定）。

可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：

- 如果 `examine=noscan`，使用指令的默认值：`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=` 指令的值来预定归档请求。此方法将连续归档，为默认方法。
- 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)` 手册页中提供了有关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 SAM-QFS Manager 软件配置 VSN 关联。有关更多信息，请参见 SAM-QFS 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 ...]
```

关联需要至少三个字段：*archive-set-name*、*copy-num* 和 *media-type*，以及至少一个卷。*archive-set-name* 和 *copy-num* 的值通过一个句点 (.) 相连。

参数	定义
<i>archive-set-name</i>	您的站点为归档集定义的名称。
<i>copy-num</i>	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。
<i>media-type</i>	介质类型。有关有效介质类型的列表，请参见《 Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual 》中的“mcf(4)”。
<i>vsn-expr</i>	正则表达式。请参见 regex(5) 手册页。
<i>-pool vsn-pool-name</i>	已命名的 VSN 组。

注 - 如果将 SAM-QFS 环境配置为按归档集进行回收，则不要将一个 VSN 指定给多个归档集。

示例 12-17 VSN 规范（多行）

以下示例显示了两行的 VSN 规范。

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

示例 12-18 带有后续行的 VSN 规范

以下示例显示了一个 VSN 规范，它使用反斜杠字符“\”将上一行续接至下一行。

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

示例 12-19 使用以简化符号表示的 VSN 规范

以下示例使用以简化符号表示的正则表达式指定 VSN。

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

 示例 12-19 使用以简化符号表示的 VSN 规范 (续)

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文件系统元数据的归档集。每一个文件系统均有一个与其自身名称相同的归档集。有关保留元数据的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`samfsdump(1M)`”。

VSN 池指令

`archiver.cmd` 文件的 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。

还可以使用 SAM-QFS Manager 软件配置 VSN 池。有关更多信息，请参见 SAM-QFS 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。可以按以下格式运行 `archiver` 命令来查看 VSN 池：

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

定义 VSN 池的语法如下：

```
vsn-pool-name media-type vsn-expr
```

参数	定义
<i>vsn-pool-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《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“mcf(4)”。
<i>vsn-expr</i>	正则表达式。可以提供一个或多个 <i>vsn-expr</i> 参数。请参见 <code>regcmp(3G)</code> 手册页。

以下示例使用了四个 VSN 池：`users_pool`、`data_pool`、`proj_pool` 和 `scratch_pool`。临时池是指当 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。如果这三个特定池中有一个池的卷已消耗殆尽，归档程序将选择临时池 VSN。

示例 12-20 VSN 池

以下示例显示了使用四个 VSN 池的 `archiver.cmd` 文件。

```
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 147 页中的“VSN 关联指令”。

SAM-QFS 中的数据完整性验证

数据完整性验证 (Data Integrity Validation, DIV) 功能可以在 SAM 磁盘高速缓存与磁带归档之间提供端对端用户数据保护。

关于 DIV

Oracle 的 StorageTek T10000C 磁带机支持 DIV。如果在 SAM-QFS 文件系统中为归档副本启用了 DIV 模式，SAM-QFS 文件系统将使用 `mt DADP` 命令在 `st` 驱动程序中启用 DIV。之后，`st` 驱动程序将计算由归档程序写入磁带中的每个逻辑块的校验和（根据服务器平台由硬件或软件生成）。校验和由磁带存储驱动器进行检查，必须与要写入磁带中的数据匹配。StorageTek T10000C 驱动器将用户数据和校验和都放在介质上。如果校验和不匹配，归档程序中将返回一个错误，并且归档程序不会将文件标记为 `copy archdone`。`st` 驱动程序在读、写操作期间都将重试 DIV 错误。如果只是暂时错误，`st` 驱动程序将成功传输数据，而不会返回错误。

SAM-QFS 使用 `st` 驱动程序检查回写程序从磁带中读取的每个逻辑块的校验和。校验和必须匹配，数据才能返回给用户。如果检测到错误，回写程序将转向备选副本。DIV 在块级别提供直接反馈，以保证只将正确的数据写入磁带，并且只从磁带返回正确的数据。StorageTek T10000C 驱动器支持非常快速的用户数据检验选项，而不必将数据传输回 SAM-QFS。由于校验和存储在含有用户数据的介质上，因此可以由磁带存储设备直接检验。

对于 StorageTek T10000C 驱动器，SAM-QFS 支持新的 `tpverify` 命令，该命令将检验写入磁带的的数据，而不对数据进行回写。可以定期执行此项检查，以检验磁带上的数据是否正确。如果数据不正确，SAM-QFS 将回写备选副本，并自动将文件重新归档。此外，如果 `tpverify` 命令检测到不正确的数据，则将返回错误。另一项归档程序功能是写后检验，该功能使归档程序在文件写入到磁带上后对整个文件进行检验。写后检验功能可确保 DIV 循环冗余码 (Cyclic Redundancy Codes, CRC) 正确存储在磁带上。

对于 Sun SAM-Remote 软件，要使用 DIV 功能，必须为 SAM-Remote 服务器及客户机启用 `default.conf` 文件中的 `div` 设置。有关 SAM-Remote 软件的更多信息，请参见第 18 章，使用 Sun SAM-Remote 软件。

在 SAM-QFS 中使用 DIV 的最低硬件和软件要求

在 SAM-QFS 中启用 DIV 的最低硬件和软件要求：

- StorageTek T10000C 磁带机
最低固件发行版级别为 1.53.315
- Oracle Solaris 11

在 SAM-QFS 中配置 DIV

可以通过更改 `/etc/opt/SUNWsamfs/defaults.conf` 文件中的 `div` 值来启用或禁用 DIV。默认情况下 DIV 处于禁用状态。可以设置以下值在 SAM-QFS 中启用或禁用 DIV：

- `div = off`
禁用 DIV。这是默认值。
- `div = on`
对读、写和检验操作启用 DIV。
- `div = verify`
对读、写和检验操作启用 DIV。此外，启用归档程序的写后检验模式。
有关更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`defaults.conf(4)`”。

▼ 如何在 SAM-QFS 中配置 DIV

- 1 使用 `samd stop` 命令停止所有操作。

```
# samd stop
```

- 2 将 `/etc/opt/SUNWsamfs/defaults.conf` 文件中的 `div` 值设置为 `on` 或 `verify`。

```
div = on | verify
```

- 3 传播配置更改。

```
# samd config
```

- 4 启动所有操作。

```
# samd start
```

DIV 使用示例

下面的示例说明如何使用 `samu` 命令显示 `defaults.conf` 文件中的 DIV 设置，以及显示各种写后检验状态。

有关显示 DIV 设置的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`samu(1M)`”。

示例 13-1 显示 DIV 设置

使用以下语法显示 `defaults.conf` 文件中的 DIV 设置：

```
# samu -L
div VERIFY
```

示例 13-2 显示写后检验归档程序状态

启用 DIV 检验后，归档程序可以在检验归档文件时显示检验状态。使用以下语法显示写后检验归档状态：

```
# samu -a
sam-arcopy: qfs.arset1.2.344 ti.TKC960
Verifying archive file at position 1175
```

示例 13-3 显示写后检验设备状态

`samu -s` 命令显示有关执行归档文件检验操作的磁带机的信息。使用以下语法显示写后检验设备状态：

```
# samu -s
ti          91 on          /dev/rmt/4cbn          90 -l----oVr
Verify averaging 240.9 MB/s
```

可以使用 `itemize -2` 命令显示上次检验的时间和位置。例如：

```
# itemize -2 30
Robot VSN catalog: eq: 30 count: 2
slot access_time count use ty vsn lvtime lvpos
0 Jun 24 17:21 2 0% ti CET052 Jun 24 17:21 0
1 Jun 24 18:43 8 5% ti CET050 Jun 24 17:11 0x5ee
```

磁带检验 (tpverify) 命令

tpverify 命令将 SCSI Verify 命令与 StorageTek T10000C 磁带机的 DIV/DADP 功能结合使用，以检验介质。

tpverify 命令将指定的卷装入磁带机并进行检验。如果未指定设备，则卷将装入到介质转换器中的可用驱动器上。SAM-QFS 文件系统选择装入卷的驱动器。如果需要资源，您可以取消检验操作。如果取消 tpverify 命令，则上次检验的位置 (last verified position, LVPos) 将作为下次检验操作的起始位置保存在目录中。

注 - 如果需要资源，SAM-QFS 归档程序或回写程序可以取消处于 "on" 状态的磁带机的检验操作。请使用处于 "unavail" 状态的驱动器，以免检验操作被 SAM-QFS 取消。

tpverify 命令可以与非 DIV 磁带机一起使用，前提是该磁带机在块计数为 0x1000、0x100 或 0xff 的固定块中支持 SCSI Verify 命令。例如，HP LTO-4 磁带机支持 SCSI Verify 命令和块计数 0xff。处理非 DIV 磁带机的 SCSI Verify 错误的方式与处理 DIV 介质错误的方式相同，tpverify 命令都会设置错误介质，并终止检验运行。在 /var/opt/SUNWsamfs/devlog/eqord 文件中查找故障的详细信息。

下表列出了可用于 tpverify 命令的选项。

选项	说明
-a	覆盖目录中保存的上次检验的位置，从介质上的第一个归档文件开始检验操作。
-c	取消正在运行的 tpverify 命令操作。
-w	等待操作完成再将其终止。

有关磁带检验命令的更多信息，请参见 tpverify(1M) 手册页。

磁带检验 (tpverify) 命令返回值

下表列出了 tpverify 命令的返回值。

返回值	说明
0	成功
249	介质错误
250	驱动器设置为关闭

返回值	说明
251	DIV 错误；介质集错误
252	驱动器需要清洁
253	SAM-QFS 取消了检验操作
254	用户已取消检验操作
255	检验失败

示例 13-4 tpverify 命令的使用示例

下面的示例说明如何应用 `tpverify` 命令。

将 VSN CET050 装入可用驱动器：

```
# tpverify ti.CET050
```

将 VSN CET048 装入驱动器 31 并等待结果：

```
# tpverify -w -a ti.CET048 31; echo $?
```

取消在 VSN CET048 上运行的检验并等待结果：

```
# tpverify -w -c ti.CET048 31; echo $?
```


关于释放

本章介绍释放过程和释放程序的操作。

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。此操作可以为其他从归档介质中创建或回写的文件腾出空间。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。

当达到站点指定的磁盘阈值时，Sun Storage Archive Manager (SAM) 软件会调用释放程序进程。或者，可以使用 `release` 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。

释放程序进程

当文件系统的利用率超过所配置的上限 (**high-water mark**, HWM) 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：

- 读取 `releaser.cmd` 文件并收集用于控制释放进程的指令
- 扫描文件系统并收集每个文件的相关信息
- 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 (**low-water mark**, LWM)，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，当文件没有归档副本时，就会出现这种情况。这种情况下，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。有关这些挂载选项的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`mount_samfs(1M)`”。

释放程序概念

本节介绍释放程序进程的基本概念：

- 第 158 页中的“时限”
- 第 158 页中的“备选文件”
- 第 159 页中的“优先级”
- 第 159 页中的“权重”
- 第 159 页中的“部分释放”

时限

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` 可以跟踪以下时间：

- 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 `sls` 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。有关更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`sls(1)`”。

备选文件

备选文件是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”。
- 该文件不是目录、块、字符特殊文件或管道。
- 归档程序正在回写该文件以创建另一副本。回写之后，该文件便成为适合释放的文件。
- 该文件的时限为负数。对于未正确设置时钟的网络文件系统 (network file system, NFS) 客户机，会发生这种情况。
- 文件被标记为不释放。这可通过 `release -n` 命令来指定。
- 该文件在过去回写的时间小于最短驻留时间设置。
- 已通过 `release` 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

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

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` 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回写

释放和回写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。部分释放文件这一功能可以使系统在不回写文件的情况下，立即访问文件存根中的数据。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默认大小和最大大小。

您可以使用 `mount` 命令或在 SAM-QFS Manager 中设置部分释放和回写的值。有关更多信息，请参见 SAM-QFS Manager 联机帮助。

`mount` 命令选项如下所示：

- `-o partial=n` 选项—设置要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—设置要保持联机的文件存根的最大大小 (*n*)。

您可以通过在 `release` 命令中指定 `--p` 选项或在 `sam_release` 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，请在 `release` 命令中指定 `--s` 选项，或在 `sam_release` 库例程中指定 `-s` 选项。`-s` 和 `s` 值必须小于挂载文件系统时与 `mount` 命令一起使用的 `-o maxpartial` 值。

注—部分释放的文件所占用的磁盘空间相当于一个 DAU。例如，如果将部分释放文件存根设置为 16K，而 DAU 大小为 256K，则该文件实际占用的磁盘空间仍为 256K。

使用挂载选项 `-o partial_stage=n` 确定在回写文件的剩余部分之前，必须读取多少部分释放存根。当读取的数据量超过所指定的 `-o partial_stage=n` 大小后，将开始回写文件。

默认情况下，`-o partial_stage=n` 选项被设置为等于部分释放存根的大小。更改该值对文件回写的影响如下：

- 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回写文件。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，系统将回写文件。这可以加快应用程序对文件数据剩余部分的访问。

示例 14-1 部分回写

在此示例中，站点设置了以下选项：

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

示例 14-1 部分回写 (续)

- `-o maxpartial=2097152 (2 GB)`

`filemgr` 程序读取文件的前 8 KB。此时，系统不会回写文件。

视频点播应用程序将读取同一个文件。当读完文件的前 16 KB 后，将回写该文件。在挂载并定位归档磁带后，该应用程序会继续读取。

当视频点播应用程序读完 2 GB 的文件数据后，它会在回写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。

- 第 161 页中的“系统管理员选项摘要”
- 第 162 页中的“用户选项摘要”

系统管理员选项摘要

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。下表中的 `mount` 选项将影响部分释放。有关 `mount` 命令的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`mount_samfs(1M)`”。

选项	作用
<code>-o maxpartial= n</code>	指定在文件标记为部分释放时，可以在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，这会禁用部分释放功能，这样，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。默认情况下， <code>n</code> 参数设置为 16。
<code>-o partial= n</code>	当用户使用 <code>release</code> 命令的 <code>-p</code> 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)。 <code>n</code> 参数的值必须至少为 8，它可以等于为 <code>-o maxpartial= n</code> 选项指定的值。由于某些应用程序不必访问整个文件便可完成其工作，因此该选项可确保应用程序能够获得文件的起始部分。另外，使用此选项还可阻止对文件进行不必要的回写。默认情况下， <code>n</code> 为 16。已从磁盘部分释放的文件所占用的磁盘空间相当于一个 DAU。
<code>-o partial_stage= n</code>	指定在访问部分释放的文件时，从归档介质回写整个文件之前必须读取 <code>n</code> 字节的文件数据。将此值设置为小于 <code>-o partial</code> 的设置值。其中的 <code>n</code> ，用于指定一个介于 0 到 <code>-o maxpartial</code> 指定值之间的整数。默认情况下，此值为 16 或是为 <code>--o partial</code> 选项指定的值。

选项	作用
<code>-o stage_n_window=<i>n</i></code>	将一次可以回写的数据量指定为 <i>n</i> 。指定一个介于 64 和 2,048,000 之间的整数。默认大小是 256 KB。此选项仅适用于已设置 <code>stage -n</code> 属性的文件。

用户选项摘要

作为用户，您可以设置文件在被释放后，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。

但通过使用 `release` 命令和 `sam_release` 库例程，用户可以设置其他释放属性并可指定要标记为部分释放的文件。下表列出了用于确定部分释放属性的命令和库选项。有关更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“[release\(1\)](#)”和《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“[sam_release\(3\)](#)”。

表 14-1 用户释放选项

选项	作用
<code>release</code> 命令和 <code>--p</code> 选项或 <code>sam_release</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=<i>n</i></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</code> 命令和 <code>--s partial_size</code> 选项或 <code>sam_release</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在挂载文件系统时指定的 <code>-o maxpartial=<i>n</i></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，而忽略用户指定的值。

配置回写程序

本章介绍了如何通过 `stager.cmd` 文件配置回写程序。

关于回写

回写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。

回写程序在 `samd` 守护进程运行时启动。回写程序具有以下默认行为：

- 回写程序尝试使用库中的所有驱动器。
- 回写缓冲区的大小由介质类型决定，并且不锁定回写缓冲区。
- 不写日志文件。
- 一次最多可以激活 1000 个回写请求。

可以通过在 `/etc/opt/SUNWsamfs/stager.cmd` 文件中插入指令为站点定制回写程序的操作。

如果应用程序需要脱机文件，则其归档副本将回写到磁盘高速缓存中（如果未设置 `-n` 选项的 `-stage never`）。要使文件立即可供应用程序使用，读取操作在回写操作之后立即进行跟踪，这样在整个文件回写完毕之前就可以开始进行访问。

回写错误包括介质错误、介质不可用、自动化库不可用以及其他错误。返回回写错误时，如果存在文件副本并有用来读取归档副本的介质的设备，则 SAM-QFS 软件将尝试找到下一个可用的文件副本。

关于 stager.cmd 文件

在 stager.cmd 文件中，指定要覆盖默认行为的指令。您可以对回写程序进行配置，以便立即回写文件、从不回写文件、回写部分文件以及指定其他回写操作。例如，指定 never-stage 属性对访问大型文件中小型记录的应用程序有益，因为不必联机回写文件即可从归档介质直接访问数据。

本节将介绍回写程序指令。有关回写程序指令的更多信息，请参见《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“stager.cmd(4)”。如果您使用的是 SAM-QFS Manager 软件，则可以通过“File System Summary”（文件系统摘要）页或“File System Details”（文件系统详细信息）页来控制回写。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回写的特定文件。可以选择从哪个副本回写，也可以让系统选择该副本。

示例 15-1 stager.cmd 文件

以下示例显示了设置所有可能指令后的 stager.cmd 文件。

```
# This is stager.cmd file /etc/opt/SUNWsamfs/stager.cmd
drives=dog 1
bufsize=od 8 lock
logfile=/var/adm/stage.log
maxactive=500
```

▼ 如何创建 stager.cmd 文件

- 1 在 /etc/opt/SUNWsamfs/stager.cmd 文件中，根据以下各节的信息添加控制站点回写的指令：
 - 第 164 页中的“如何创建 stager.cmd 文件”
 - 第 165 页中的“drives 指令：指定进行回写的驱动器数量”
 - 第 165 页中的“bufsize 指令：设置回写缓冲区大小”
 - 第 166 页中的“logfile 指令：指定日志文件”
 - 第 168 页中的“maxactive 指令：指定回写请求的数量”
 - 第 168 页中的“copysel 指令：指定回写的副本顺序”
- 2 保存并关闭 stager.cmd 文件。
- 3 传播文件更改并重新启动系统。

```
# samd config
```

drives 指令：指定进行回写的驱动器数量

默认情况下，回写程序在回写文件时使用所有可用的驱动器。如果回写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 指令用于指定回写程序可用的驱动器数量。此指令的格式如下：

```
drives = library count
```

参数	定义
<i>library</i>	mcf 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 mcf 文件中为该库配置的驱动器数量相同。

示例 15-2 drives 指令

下面的示例指定，dog 系列集的库中只有一个驱动器用于回写文件：

```
drives = dog 1
```

bufsize 指令：设置回写缓冲区大小

默认情况下，要回写的文件先被读取到缓冲区的内存中，然后再从归档介质恢复到磁盘高速缓存。使用 bufsize 指令指定缓冲区大小，还可以选择锁定缓冲区。这些操作可以改善性能。您可以尝试各种 *buffer-size* 值。该指令的格式如下：

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

参数	定义
<i>media</i>	指定 mcf(4) 手册页中列出的某归档介质类型。
<i>buffer-size</i>	一个介于 2 到 8192 的数字。默认值为 16。此值乘以该介质类型的 <i>dev_blksize</i> 值，计算结果即为所使用的缓冲区大小。 <i>dev_blksize</i> 的值是在 defaults.conf 文件中指定的。为 <i>buffer-size</i> 指定的数值越大，所使用的内存就越多。有关更多信息，请参见《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“defaults.conf(4)”。

参数	定义
lock	lock 参数指明回写程序在回写归档副本时应当使用锁定的缓冲区。如果指定 lock，回写程序将在复制操作期间在内存中的回写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 lock 参数。如果内存不足，则可能会导致内存用尽。只有已为需要回写的文件启用直接 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 缓冲区上设置锁定（包括用于回写的缓冲区）。有关启用直接 I/O 的更多信息，请参见《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“setfa(1)”、《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“sam_setfa(3)”或《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“mount_samfs(1M)”上的 -O forcedirectio 选项。

logfile 指令：指定日志文件

可以请求 SAM-QFS 软件收集文件回写事件信息，并将此信息写入日志文件中。默认情况下，不写入任何日志文件。logfile 指令用于指定回写程序可在其中写入记录信息的日志文件。回写程序在日志文件中写入一行或多行有关每个已回写文件的信息。该行中包括文件名、回写日期和时间以及卷序列号 (volume serial number, VSN) 等信息。该指令的格式如下：

```
logfile=filename [event]
```

参数	定义
<i>filename</i>	指定一个全路径名。
<i>event</i>	指定一个或多个回写事件。如果要指定多个事件，请使用空格将各个事件分隔开。可能的事件规范如下： all — 记录所有回写事件。 start — 记录文件开始回写的时间。 finish — 记录文件结束回写的时间。默认启用。 cancel — 记录操作员取消回写操作的时间。默认启用。 error — 记录回写错误。默认启用。

示例 15-3 指定回写程序日志文件

以下指令将创建 /var/adm/stage.log 文件：

```
logfile=/var/adm/stage.log
```

示例 15-4 回写程序日志文件

```

S 2003/12/16 14:06:27 dk disk01 e.76d 2557.1759 1743132 /saml/testdir0/filebu 1 root other root 0 -
F 2003/12/16 14:06:27 dk disk01 e.76d 2557.1759 1743132 /saml/testdir0/filebu 1 root other root 0 -
S 2003/12/16 14:06:27 dk disk02 4.a68 1218.1387 519464 /saml/testdir1/fileaq 1 root other root 0 -
S 2003/12/16 14:06:43 dk disk01 13.ba5 3179.41 750880 /saml/testdir0/filecl 1 root other root 0 -
F 2003/12/16 14:06:43 dk disk01 13.ba5 3179.41 750880 /saml/testdir0/filecl 1 root other root 0 -
S 2003/12/16 14:06:59 dk disk01 17.167b 1155.1677 1354160 /saml/testdir0/filedb 1 root other root 0 /
-
F 2003/12/16 14:06:59 dk disk01 17.167b 1155.1677 1354160 /saml/testdir0/filedb 1 root other root 0 /
-
S 2003/12/16 14:06:59 dk disk02 f.f82 3501.115 1458848 /saml/testdir1/filecb 1 root other root 0 -
S 2003/12/16 14:07:15 dk disk01 1f.473 1368.1419 636473 /saml/testdir0/fileed 1 root other root 0 -
S 2003/12/16 14:07:15 dk disk02 16.f15 3362.45 1065457 /saml/testdir1/filecz 1 root other root 0 -
S 2003/12/16 14:07:31 dk disk01 23.201d 3005.1381 556807 /saml/testdir0/fileeq 1 root other root 0 -
S 2003/12/16 14:07:47 dk disk01 26.c4d 2831.1113 1428718 /saml/testdir0/fileez 1 root other root 0 -
S 2003/12/16 14:07:47 dk disk02 1b.835 3736.59 1787855 /saml/testdir1/filedp 1 root other root 0 -

```

下表介绍了回写程序日志文件字段的内容。

字段	示例值	内容描述
1	S	回写活动—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	回写操作发生的日期，格式为 <i>yyyymm/dd</i> 。
3	14:06:27	回写操作的时间，格式为 <i>hhmm:ss</i> 。
4	dk	归档介质类型。有关介质类型的信息，请参见《 Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual 》中的“mcf(4)”。
5	disk01	VSN。
6	e.76d	介质上归档文件（tar 文件）开头的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	/saml/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	待回写文件所在驱动器的设备序号。

字段	示例值	内容描述
15	-	此字段中的 v 表示针对文件使用数据验证。

maxactive 指令：指定回写请求的数量

可以使用 `maxactive` 指令指定一次可激活的回写请求的数量。该指令的格式如下：

```
maxactive=number
```

默认情况下，`number` 为 4000。允许的最小数量为 1。允许的最大数量为 500,000。

以下示例指定，可以同时存在于队列中的回写请求数量不超过 500 个：

```
maxactive=500
```

归档集分配指令：指定归档集中所有文件的回写属性

`archiver.cmd` 文件中的大多数指令只影响归档，但可以使用归档集分配指令指定适用于归档集中所有文件的回写属性。

第 12 章，[归档集指令 \(archiver.cmd\)](#) 完整介绍了归档集分配指令及其参数。下表列出了可能出现在 `archiver.cmd` 文件的归档集分配指令中的回写指令。

指令	作用
<code>-stage a</code>	指定归档集中的文件应联合回写。
<code>-stage d</code>	重置为默认设置。
<code>-stage n</code>	指定不回写归档集中的文件。

copysel 指令：指定回写的副本顺序

利用 `stager.cmd` 文件中的配置选项 `copysel`，可以根据文件系统更改回写程序副本选择顺序。

默认情况下，回写的副本选择项为副本编号 1 到 4。先选择 1 号副本，然后选择 2 号副本，接着选择 3 号副本，最后选择 4 号副本。为配置选项 `copysel = n1:n2:n3:n4` 中的 `n` 输入从 1 到 4 的值，即可更改每个文件系统的回写程序副本选择顺序。即使 `n1` 到 `n4` 这四个副本不可用，也必须进行定义。

以下日志文件示例显示了 `stager.cmd` 操作的结果。

示例 15-5 stager.cmd 操作的日志文件

```
logfile = /var/opt/SUNWsamfs/log/stager
drives = hp30 1
copysel = 4:3:2:1
fs = samfs1
copysel = 3:1:4:2
streams
dk -maxsize 2G -maxcount 10000
endstreams
```

上例中的日志文件解释如下：

- 日志文件位于 `/var/opt/SUNWsamfs/log/stager` 目录中。
- 只允许介质驱动器 HP30 使用一个驱动器回写文件。
- 默认情况下，回写程序在回写文件时按照 4、3、2、1 的顺序选择副本。
- 对于 `samfs1` 文件系统，回写程序在回写文件时按照 3、1、4、2 的顺序选择副本。
- `dk` 类型的介质的最大流大小为 2 GB，每个流的最大文件计数为 10,000 个。

有关更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`stager.cmd(4)`”。

确定预备请求的优先级

归档程序和回写程序进程均会请求装入和卸载介质。如果请求的数量超过可用于介质装入的驱动器数量，则多余的请求会发送至预备队列。

预备队列中的条目数量取决于 `defaults.conf` 文件中的 `previews=` 指令。有关更改此指令的值的的信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`defaults.conf(4)`”。

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。

预备请求的总优先级是由静态和动态因子共同决定的。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。

可以通过在 `/etc/opt/SUNWsamfs/preview.cmd` 命令文件中输入指令来覆盖 FIFO 默认值。

关于 `preview.cmd` 文件

`sam-amld` 守护进程在启动时读取 `preview.cmd` 文件。此文件根据请求是针对回写还是归档，对预备队列中的请求进行排序。您可以提高特定 VSN 的优先级，也可以控制特定文件系统预备请求的优先级。

以下规则适用于 `preview.cmd` 文件：

- 每行放置一条指令。
- 如果在 `sam-amld` 守护进程运行时更改此文件，应重新启动守护进程以使更改生效。
- 注释行以井号 (#) 开头。

有关此文件的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“[preview.cmd\(4\)](#)”。

`preview.cmd` 文件中使用以下类型的指令：

- 全局指令，应用于所有文件系统
- 文件系统指令，特定于单个文件系统

全局指令位于文件顶部，其设置适用于所有文件系统。

文件系统指令以 `fs =` 指令开头，该指令指定适用所有后续指令的文件系统。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。

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。

以下各节介绍了如何编辑 `preview.cmd` 文件，以控制预备队列：

- [第 170 页](#)中的“如何设置全局 VSN 和时限优先级指令”
- [第 171 页](#)中的“如何设置全局或特定于文件系统的界限指令”
- [第 172 页](#)中的“设置预备请求的优先级方案”

▼ 如何设置全局 VSN 和时限优先级指令



注意 - VSN 指令和时限优先级指令是全局指令，因此它们在 `preview.cmd` 文件中位于任何特定于文件系统的指令之前。

1 更新 `vsn_priority` 指令。

```
vsn_priority = value
```

该指令是一个静态优先级因子，表示在有高优先级卷时，总优先级将要增加的值。默认值为 `1000.0`。要使用该优先级因子，必须在将卷安排为预备请求之前设置其优先级标记。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`）。

2 更新 `age_priority` 指令。

```
age_priority = factor
```

该指令是静态优先级因子，尽管其影响是动态的。该因子将与请求作为预备请求的秒数（即请求等待的时间）相乘。其结果将与请求的总优先级相加。请求等待的时间越长，优先级就会变得越高。设置该因子可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。

如果将此因子设置为 1.0 以上，可以在计算总优先级时提高时间因子的重要性，如果将其设置为小于 1.0 的值，则会降低时间因子的重要性。如果将该因子设置为 0.0，则在计算总优先级级时不考虑时间因子。

对于没有设置优先级标志的卷，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。

▼ 如何设置全局或特定于文件系统的界限指令

- 界限预备请求指令可以用作全局指令，也可用作特定于文件系统的指令。这些指令决定了预备请求的界限优先级，如下面的等式所示。

$$lwm_priority + lhwm_priority + hlwm_priority + hwm_priority = \text{water mark priority (界限优先级)}$$

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。

界限优先级仅用于计算与归档有关的请求，而不能用于计算与回写有关的介质请求。当界限优先级因子是一个正数时，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回写请求的优先级。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回写请求，然后再处理归档请求。界限优先级因子为 0.0（或不指定指令）表示不采取任何特殊的操作。

有关更多信息，请参见示例 15-7—强制执行回写请求的方案。

界限指令具有以下格式：

```
wmtype_priority = value
```

界限指令	参数
<code>lwm_priority = value</code>	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。默认值为 0.0。
<code>lhwm_priority = value</code>	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。该变化表明文件系统中的文件正在增加。默认值为 0.0。
<code>hlwm_priority = value</code>	指定当文件系统从高于上限下降到低于上限，但仍高于下限时，希望归档请求的界限优先级因子更改的数量。该变化表明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。默认值为 0.0。

界限指令	参数
<code>hwm_priority = value</code>	指定当文件系统高于上限时，希望归档请求的界限优先级因子更改的数量。默认值为 0.0。

当文件系统的情况发生变化时，系统将根据相应的界限优先级设置来重新计算与该文件系统关联的每个卷的优先级（可能使用也可能不使用 `chmed` 命令的 `-p` 选项）。

下面的示例释放足够的磁盘空间，可以使文件系统低于下限。

```
lhwm_priority = -200.0
hlwm_priority = 100.0
```

设置预备请求的优先级方案

预备请求的总优先级是所有优先级因子的总和：

$$\text{total priority (总优先级)} = \text{vsn_priority} + \text{wm_priority} + (\text{age_priority} * \text{time_in_sec_as_preview_request})$$

仅出于如下原因更改默认的 FIFO 方案：

- 确保首先处理回写请求，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。

以下示例显示了可以满足这三个条件的 `preview.cmd` 文件。

示例 15-6 `preview.cmd` 文件样例

```
# 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
```

在用户访问数据的重要性极高、VSN 驱动器数量有限或者文件归档作为后台功能执行的环境中，使用 `preview.cmd` 文件可以影响存储系统资源处理回写请求的方式。可以在 `preview.cmd` 文件中定制设置，以支持上述所有方案，并影响所配置的 SAM-QFS 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，能够在归档请求和回写请求之间找到适当的平衡点。

示例 15-7 强制执行回写请求的方案

以下计算示例显示如何使用一个负 `wm_priority` 值来确保回写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。

下表显示了计算总请求优先级的方式，如下所示：

优先级	计算
具有优先级的归档 VSN，下限	$1000 + (-200) + (1 \times 100) = 900$
具有优先级的回写 VSN，下限	$1000 + 0 + (1 \times 100) = 1100$
无优先级的回写 VSN，下限	$0 + 0 + (1 \times 100) = 100$

示例 15-8 强制执行归档请求的方案

当在环境中权衡为用户回写文件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达到极限的最佳方法。

在此情况下，只需在 `preview.cmd` 文件中进行以下示例中的设置：

```
hwm_priority = 500.0
```

示例 15-9 按介质类型设定请求级别的方案

假设在站点所处的环境中，用户正在处理使用特定卷的多组文件，并且用户之间彼此隔离。在此环境中，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。以下示例为用户及其介质指定了适当的优先级：

```
vsn_priority = 5000.0
```

之后，对于优先级用户组中的每个卷，加入以下信息：

```
# chmed +p lt. VSN
```

现在，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定挂载请求。

示例 15-10 复杂优先级的方案

假设有两个具有以下要求的文件系统：

- 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，将优先处理回写请求。
- 如果其中的一个文件系统高于下限但低于上限，则不会确定请求的优先级。

示例 15-10 复杂优先级的方案 (续)

以下示例显示了受影响的指令。

```
lwm_priority = -200.0
lhwm_priority = 0.0
hlwm_priority = 0.0
```

当其中的一个文件系统超过上限时，将优先处理归档请求。

假设两个文件系统都超过上限，但必须阻止第二个文件系统 (samfs2) 达到其极限。以下示例显示的 `preview.cmd` 文件根据这些要求确定请求的优先级。

```
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
```

配置回收程序

本章介绍回收过程和回收指令。

关于回收

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他卷中。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，可以重新标记卷以便立即重新使用此卷，或将其中的数据导出至异地存储设备，从而单独保存文件更改的历史记录。用户不清楚回收过程。

在任何时候，归档卷的空间均由以下各项组成：

- 当前数据，由有效的归档映像组成
- 过期数据，由不再有效的归档映像组成
- 空闲空间，由未被有效或过期的归档映像占用的空间组成

回收程序负责将过期数据占用的空间保持在由站点指定参数定义的最低水平。

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 卷来说，它的容量为 10 GB，空闲空间为 7 GB。

对于全新的归档介质或新标记的归档介质，其容量等于空闲空间。当将数据归档至该介质时，空闲空间会减少，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变为过期数据。归档映像占用的物理空间不变。不过，文件系统中没有指向该空间的文件。回收空间后，这些映像被删除，它们占用的空间随即释放，可另作他用。回收程序的目标是将过期数据占用的空间转变为空闲空间，而丝毫不损坏任何当前数据。

回收过程

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` 属性来标记卷上存在的所有当前归档映像。
2. 如果要归档至可移除介质，回收程序用 `recycle` 属性来标记归档卷，阻止归档程序再将任何归档映像写入卷中。
3. 归档程序将所有已做标记的映像移至另一个卷。此操作称为**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，旧卷中只包含空闲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。

回收程序设计为定期运行，但您可以随时运行该程序。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 `inode` 中。在回收过程中，可以使用 `sls(1)` 命令及其 `-D` 选项来显示文件的相关信息。`sls(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。

规划回收

回收使用两种方法来完成，具体取决于介质的类型。

表 16-1 介质类型与回收方法

归档介质	回收方法
可移除介质卡盒	按自动化库
可移除介质卡盒	按归档集
磁盘	按归档集

有关按这些方法进行配置的信息，请参见第 16 章，[配置回收程序](#)。

回收方法

通过手动输入 `sam-recycler` 命令或通过 `cron` 作业启动回收过程。下表显示了回收方法。

表 16-2 回收方法与介质类型

回收方法	归档介质	配置位置
按自动化库	可移除介质卡盒	<code>recycler.cmd</code> 、 <code>recycler.sh</code>
按归档集	可移除介质卡盒	<code>recycler.cmd</code> （可选，对于未包含在归档集中的 VSN）、 <code>recycler.sh</code> 、 <code>archiver.cmd</code>
按归档集	磁盘	<code>archiver.cmd</code>

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。
`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 179 页中的“[创建 recycler.cmd 文件](#)”。
- 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，来确定是当前文件还是过期文件，并识别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。
- 所有文件系统必须都已挂载，方可运行回收程序。如果要对联机磁盘执行回收操作，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供访问。



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要格外小心。每个 SAM-QFS 服务器的 `diskvols.conf` 文件必须指向一组唯一的磁盘归档目标目录。如果由不同的 SAM-QFS 服务器共享其中任何目录，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。

控制回收

可使用命令文件中的参数启用及禁用回收过程。

配置回收程序及测试结果时，对相应的文件进行编辑，添加其参数。

回收方法	参数
归档集	archiver.cmd 文件中的 <code>-recycle_ignore</code> 参数。
自动化库	recycler.cmd 文件中的 <code>ignore</code> 参数。

准备好使用回收程序时，编辑文件，删除参数。

另外，也可以使用 SAM-QFS Manager。有关更多信息，请参见 SAM-QFS Manager 联机帮助。

启动回收程序

通过运行 `sam-recycler` 命令启动回收程序。

```
# sam-recycler
```

回收程序将读取 `recycler.cmd` 文件。

检查标准输出日志、SAM-QFS 日志和 `/var/adm/messages`，查看是否存在任何来自回收程序的错误消息。如果有错误，请更正文件。

启动回收程序后，在“`library` 指令：指定回收整个自动化库”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《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`sam-recycler(1M)`”。

如果系统以常规方式执行，可使用 `crontab` 条目定期运行回收程序。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 `root` 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
```

在可移除介质卡盒上配置回收

如果要回收库中卡盒上的归档副本，请创建 `recycler.cmd` 文件。

如果要按归档集回收，请在 `recycler.cmd` 文件中配置每个库。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）。

创建一个 `recycler.sh` 文件完成该操作。

创建 `recycler.cmd` 文件

`recycler.cmd` 文件包含常规回收指令。它还可包含针对 SAM-QFS 环境中的每个库的指令。典型的 `recycler.cmd` 文件中包含以下指令行：

- `logfile=` 指令用于指定回收程序日志文件。
- 针对每个含有待回收卷的库的一个或多个指令。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。
- 测试期间，加入 `ignore` 关键字。以后再删除 `ignore` 关键字。

作为 `root` 用户，将一个或多个指令添加到 `/etc/opt/SUNWsamfs/recycler.cmd` 文件。

另外，可使用 SAM-QFS Manager 软件创建一个 `recycler.cmd` 文件。有关更多信息，请参见 SAM-QFS Manager 联机帮助。

示例 16-1 `recycler.cmd` 文件样例

```
logfile = /usr/tmp/recycler.log
stk30 -hwm 51 -mingain 60 -ignore -mail root
```

`recycler.cmd` 文件可以接受以下各节介绍的指令：

- 第 179 页中的“`logfile` 指令：指定日志文件”
- 第 180 页中的“`no_recycle` 指令：阻止回收”
- 第 181 页中的“`library` 指令：指定回收自动化库”

logfile 指令：指定日志文件

`logfile` 指令用于创建回收程序日志文件。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
```

示例 16-2 可移除介质卡盒的回收程序日志文件样例

以下示例显示了回收可移除介质卡盒时的回收程序日志文件样例。

```

===== 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 =====

```

no_recycle 指令：阻止回收

no_recycle 指令禁用卷回收。此指令的格式如下：

```
no_recycle media-type VSN-regexP [VSN-regexP...]
```

参数	定义
<i>media-type</i>	mcf(4) 手册页中列出的某一介质类型。可禁用存储在特定类型介质上的卷的回收
<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。可禁用特定卡盒的回收。有关信息，请参见 regexp(5) 手册页。

以下示例排除了 VSN 标识符以 DLT 开头的所有磁带卷：

```
no_recycle lt DLT.*
```

library 指令：指定回收自动化库

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（见下表）。

参数值	定义
-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。默认值取决于介质： <ul style="list-style-type: none"> ■ 对于容量小于 200 GB 的卷，默认的 <i>mingain</i> 为 60%。 ■ 对于容量大于等于 200 GB 的卷，默认的 <i>mingain</i> 为 90%。
-vsncount count	可回收卷的最大数量。默认值为 1。

示例 16-3 library 指令

下面的示例为库 *gr47* 指定了以下内容：

- 当库中卷的占用率达到 85% 时，将对库执行回收操作。
- 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。
- 要重新归档的数据不超过 1 GB。该值是默认值，因此未在 *recycler.cmd* 文件中指定。

```
gr47 -hwm 85 -ignore -mail root -mingain 40
```

以下几节将介绍这些参数。

-hwm 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：

$$((3 * 1.00 + 7 * 0.30) * 20G) / (10 * 20G) * 100\% = 51\%$$

该计算方法并不区分当前数据和过期数据，它只考虑使用的介质质量。

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。

通过使用以下命令设置回收标志可强制回收 VSN：

```
# chmed +c lt. VSN
```

设置 +c 标志后，归档程序不会再向卷中写入任何归档映像。可以通过 samu 实用程序来查看 +c 标志。有关更多信息，请参见《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“chmed(1M)”和《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“samu(1M)”。有关使用 samu(1M) 操作员实用程序的信息，请参见《Sun QFS 文件系统 5.3 配置和管理指南》。

-mingain 参数

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，如果自动化库某个卡盒中当前数据占用 95% 的空间，可用空间为 5%，则将最小增益设置为 6% 或更大可禁止回收程序自动选择此 VSN。

-ignore 参数

-ignore 参数可禁止回收程序回收某个特定库。在配置及测试回收程序时，请使用此参数。

-mail 参数

-mail 参数指定回收程序在回收某个库时发送电子邮件。电子邮件的主题行如下所示：

```
Robot robot-name recycle
```

示例 16-4 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.
```

创建 `recycler.sh` 文件

如果要归档至可移除介质卡盒，请创建一个 `recycler.sh` 文件。

如果只将文件归档至磁盘，则请勿执行本步骤。

针对已回收卡盒确定站点策略。某些站点重新标记并重新使用卡盒；而某些站点从自动化库中取出卡盒以便将来用于访问历史文件。

在将 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。

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
```

有关脚本的示例，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`recycler.sh(1M)`”或查看 `/opt/SUNWsamfs/examples/recycler.sh` 脚本，其中显示如何重新标记已回收 VSN 及如何向超级用户发送邮件。

配置磁盘归档卷的回收

如果要归档至磁盘，必须编辑 `archiver.cmd` 文件才能进行回收。

如果要按归档集回收，必须在 `params` 指令和 `endparams` 指令之间添加归档集回收指令。

如果要按库进行回收，此步骤为可选项。

编辑 `archiver.cmd` 文件

要编辑 `archiver.cmd` 文件，请按照第 95 页中的“关于 `archiver.cmd` 文件”中介绍的步骤操作。

另外，也可以使用 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。

下表显示了您可以使用的归档集回收指令。

表 16-3 归档集回收指令

指令	功能
<code>-recycle_dataquantity size</code>	限制回收程序可安排进行重新归档的数据量，以清除含有有用数据的磁盘卷。默认情况下，将忽略磁盘归档回收的限制。

表 16-3 归档集回收指令 (续)

指令	功能
-recycle_ignore	阻止回收归档集。
-recycle_mailaddr <i>mail-address</i>	将回收程序消息发送至指定的电子邮件地址。
-recycle_mingain <i>percent</i>	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%。
-recycle_minobs <i>percent</i>	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%。
-rearch_stage_copy <i>copy-number</i>	设置从所选择的 (更快的) 副本进行回写来重新归档。

有关归档程序指令的更多信息, 请参见第 10 章, [配置归档程序](#)或 `archiver.cmd(4)` 手册页。

磁盘归档的回收程序日志记录

示例 16-5 磁盘归档文件的回收程序日志文件

```

---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)

```

回收以实现归档副本保留

作为常规回收过程的替代方法, 可将 `sam-nrecycler` 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 SAM-QFS 转储文件来实现归档保留的功能。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` 工具会扫描文件系统元数据和 SAM-QFS 转储文件, 以便确定哪些卷包含归档映像。在非高峰时间可通过 `crontab` 文件调用该工具, 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 SAM-QFS 转储文件, 来识别可移除介质卷或磁盘归档 tar 文件中的所有归档映像。随后, `nrecycler` 可确定卷是否包含任何归档映像。可回收这些卷的空间。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，则可以从磁盘归档目录中删除此 tar 文件。

当 `sam-nrecycler` 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。

通过在 `/etc/opt/SUNWsamfs/nrecycler.cmd` 文件中加入指令来控制 `sam-nrecycler` 工具的操作。还必须指定包含 SAM-QFS 转储文件的目录路径。目录列表必须完整，且必须包含所有的 SAM-QFS 转储文件。

还可以在 `nrecycler.cmd` 文件中加入 `logfile=` 指令行，以指定 `nrecycler` 日志文件。系统会将回收消息和回收报告写入此文件。

有关 `sam-nrecycler(1M)` 的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`sam-nrecycler(1M)`”。

SAM-QFS 高级主题

本章讨论了有关 Sun Storage Archive Manager (SAM-QFS) 系统管理和使用的高级主题。

使用设备日志记录

设备日志记录工具可以提供特定于设备的错误信息，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。系统将分别为每一个自动化库、磁带和光盘驱动器设备创建一个日志文件，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 QFS 文件系统和一个配有两个光盘驱动器的 Hewlett-Packard 光盘库。

以下示例显示了 `mcf` 文件。

```
/dev/samst/c1t5u0 40 hp hp40 - /etc/opt/SUNWsamfs/hp40_cat
/dev/samst/c1t4u0 41 mo hp40 -
/dev/samst/c1t6u0 42 mo hp40 -
```

以下示例显示了 `/var/opt/SUNWsamfs/devlog` 文件。设备 43 是历史记录。

```
# pwd
/var/opt/SUNWsamfs/devlog
# ls
40 41 42 43
#
```

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，特别是在已打开所有设备的全部日志记录选项且发生大量设备活动时。设备日志设置最初设置为下面的默认值：

```
err retry syserr date
```

如果您怀疑其中某个设备存在问题，请考虑为该设备启用其他日志记录事件。另外，如果您的服务供应商建议您这样做，请启用设备日志记录。在这些情况下，请将事件设置为 `detail`。在极为特殊的情况下，服务提供商可能建议您将某个设备的事件设置为 `all`。不过，一般而言，在系统运行时设置过多的日志记录是不可行的。

当运行 `samexplorer(1M)` 命令时，系统会自动收集设备日志信息。此进程支持文件系统服务人员在问题分析活动期间复查任何设备错误信息。

▼ 如何通过使用 `samset` 命令启用设备日志

- 运行 `samset` 命令。

```
# 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。

▼ 如何通过编辑 `defaults.conf` 文件启用设备日志

- 1 成为超级用户。
- 2 在 `/etc/opt/SUNWsamfs/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。

当 QFS 文件系统启动时，它会将每个可用设备的事件类型设置为 `default`。另外，还可以使用 `samset` 命令来确定每一个设备日志的当前设置。

- 3 保存并关闭 `defaults.conf` 文件。
- 4 使用 `samd config` 命令传播 `defaults.conf` 文件更改。

```
# samd config
```

使用可移除介质文件

您可以使用 `request` 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` 命令将会忽略归档程序的典型功能。

与典型的 QFS 文件类似，可移除介质文件也具有权限、用户名、组名和大小等特征。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，并将它们写入到可移除介质卡盒。

对于您在 `request` 命令指定的文件，系统会为它在 `.inodes` 文件中创建 `inode` 条目。QFS 文件系统会从 `inode` 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。

存储在多个卷中的可移除介质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 QFS 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

使用 `tplabel` 或 `odlabel` 命令分别标记磁带或磁光盘卡盒。有关详细信息，请参见第 60 页中的“标记卡盒及取消卡盒标记”。

要创建文件，请运行 `request` 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
```

参数	定义
<code>media-type</code>	可移除介质卡盒的介质类型。有关有效 <code>media-type</code> 规范的信息，请参见《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“mcf(4)”。

参数	定义
<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。使用正斜杠字符 (/) 分隔 <i>vsn</i> 参数。所指定的 VSN 不能是 SAM-QFS 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。如果拥有许多 VSN，请使用包含 VSN 列表的输入文件。
<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 QFS 文件系统中。

示例 17-1 创建可移除介质文件

以下命令用于创建可移除介质文件：

```
# request -m lt -v aaa rem1
```

有关如何创建可移除介质文件的详细示例，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“request(1)”。

示例 17-2 创建卷溢出文件

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
```

使用分段文件

SAM-QFS 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。这种情况下，无论何时，分段文件都只有一部分驻留在磁盘高速缓存中。

可以使用 `segment` 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回写。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。分段的文件支持所有其他 SAM-QFS 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《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`segment(1)`”或《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`sam_segment(3)`”。

归档分段文件

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，不应用于整个文件。

通过在 `archiver.cmd` 文件中为归档集同时指定 `-drives` 和 `-drivemin` 参数，可分散读写某个段。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，此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。最多可为每个段创建四个归档副本。SAM-QFS 还支持段的卷溢出。

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。

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。此实用程序包括以下项目：

- 日志文件，包含从磁带设备日志检测页收集的数据。
- `sefreport` 命令，以可读的格式将日志文件写入 `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` 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《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`tapealert(1M)`”。

▼ 如何启用 SEF 报告

- 1 成为超级用户。

- 2 运行 `mkdir` 命令创建 SEF 目录。

例如：

```
# mkdir /var/opt/SUNWsamfs/sef
```

- 3 使用 `touch` 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 stop` 和 `samd start` 命令初始化 SEF 报告。

```
# samd stop  
# samd config  
# samd start
```

- a. 等待至磁带机空闲。然后卸载磁带机。

对于 `eq`，请指定驱动器的设备编号。

- b. 使用 `samd unload` 命令卸载所有可移除介质：

```
# samd unload eq
```

- 5 对于每个磁带机，重复步骤 1 到 4。

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，请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。有关从备选位置读取日志检测数据的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`sefreport(1M)`”。

▼ 如何生成 SEF 报告输出

SEF 报告输出的内容由标题行和日志检测数据组成。

- 1 验证 `/opt/SUNWsamfs/sbin` 是否在命令路径中。

- 2 使用 `sefreport` 命令生成 SEF 输出。

下面是 `sefreport` 命令最常用的选项：

- `-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。利用此进程，能够更轻松地搜索和查找与某个特定设备相关的 SEF 记录。
- `-v` 选项或 `-t` 选项。不要在同一命令行中指定 `-t` 和 `-v` 选项。它们是互相排斥的。
 - `-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。利用此方法，可仅选择与某个特定设备或特定卷相关的那些行。
 - `-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。

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
```

示例 17-3 sef.output 文件的内容

```
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
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
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（包括其内容和格式），请参见 [《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》](#) 中的“sefdata(4)”。有关可选 SEF 报告格式的更多信息，请参见 [《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》](#) 中的“sefreport(1M)”。

管理 SEF 日志文件

SEF 日志文件的管理方式与任何其他 SAM-QFS 日志文件相同。您可以定期运行 cron 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。

此外，还可以使用 `log_rotate.sh` 实用程序来轮转此日志文件。

有关用于管理 SEF 日志文件的工具的更多信息，请参见 `cron(1M)` 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 Oracle 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，以便进行介质分析。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《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`defaults.conf(4)`”。

▼ 如何创建 SEF sysevent 处理程序

1 创建 `/var/tmp/xx` 文件。

例如：

```
# !/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+rxw /var/tmp/xx
```

通过添加以下信息，将 SEF sysevent 处理程序添加到 `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 处理程序 `/var/tmp/xx` 的 `/etc/sysevent/config/SUNW,SUNWsamfs,Device,sysevent.conf` 文件，并将该事件处理程序装入 `syseventd` 守护进程。

3 要装入 SEF sysevent 处理程序，请运行以下命令：

```
# pkill -HUP syseventd
```

有关更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`sfsysevent(4)`”。

使用 Sun SAM-Remote 软件

通过 Sun SAM-Remote 客户机和 Sun SAM-Remote 服务器构成的实现体系，您可以在 SAM-QFS 主机系统之间共享库和其他可移除介质设备。使用 SAM-Remote 软件可配置多个存储客户机，它们通过集中式磁带库或磁光盘库中归档和回写文件。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。

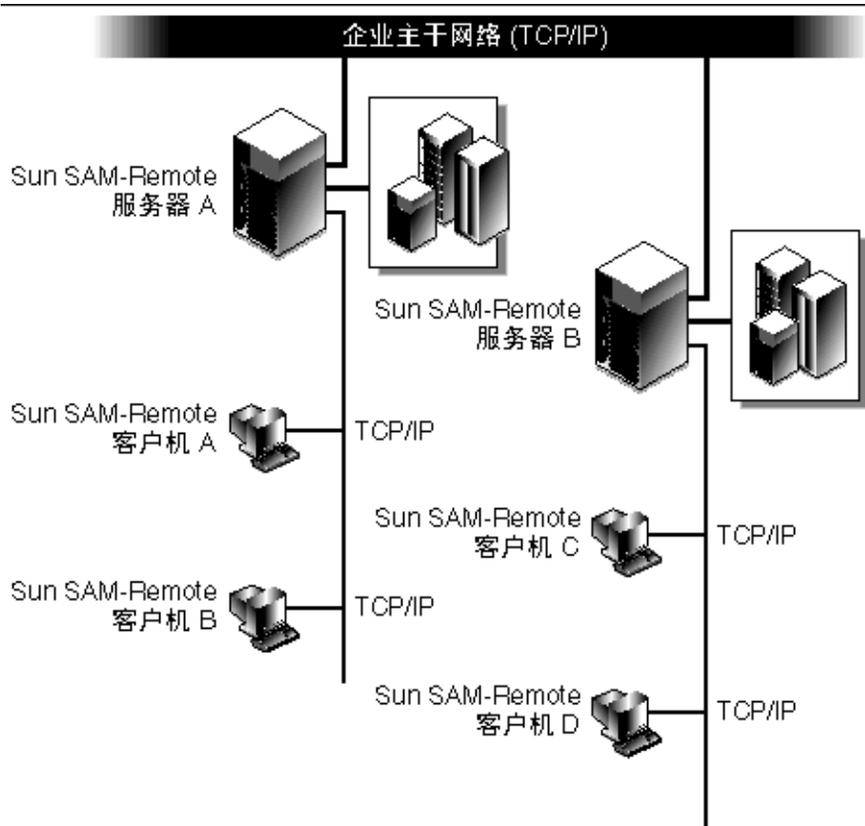
Sun SAM-Remote 软件概述

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迁移数据。
- 允许多个 SAM-QFS 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 Oracle 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 SAM-QFS 软件的 Oracle 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访问各自的数据。即使在他们的存储库不可用时，也是如此。

图 18-1 两台 Sun SAM-Remote 服务器，各有两个客户机



系统要求

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且可操作的 SAM-QFS 归档管理软件包。
- 主机系统应安装相同修订版本的 SAM-QFS 软件和相同的修补程序集。如果必须升级某些主机系统，请参见《[Sun QFS 和 Sun Storage Archive Manager 5.3 安装指南](#)》中的第 10 章“升级 Sun QFS 和 SAM-QFS”。
- 一个或多个要用作 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。
- 客户机和服务器之间应通过 TCP/IP 连接进行网络连接。

软件限制

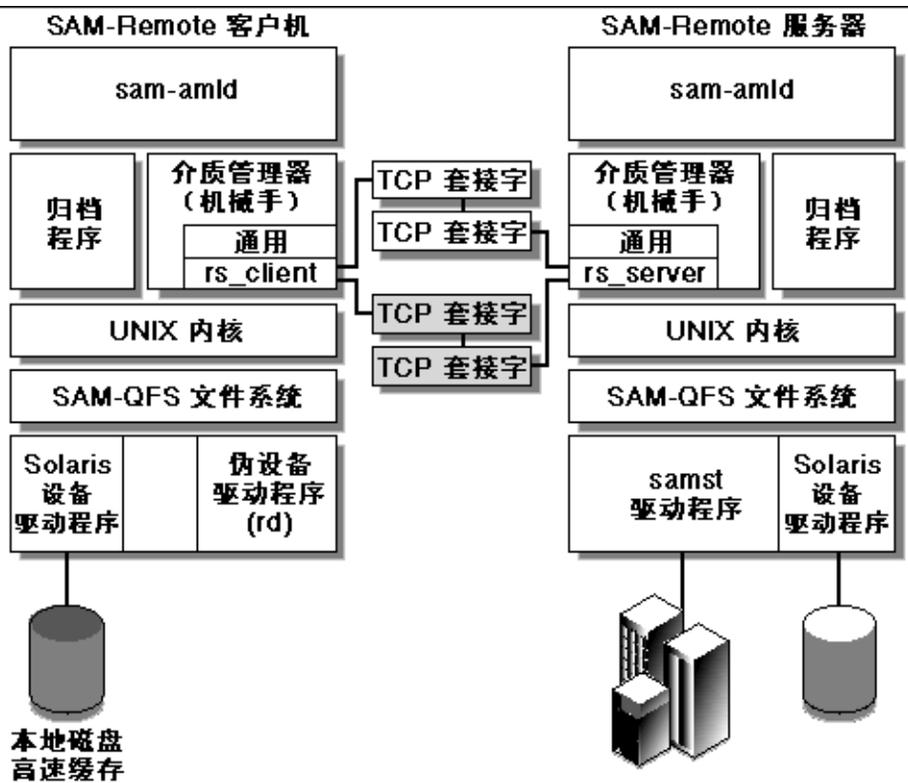
SAM-QFS 软件对待远程库中卡盒的方式与对待本地库中卡盒的方式完全一样。不过，Sun SAM-Remote 软件存在以下限制：

- 使用 Sun SAM-Remote 回收介质之前，务必全面测试您的环境。有关更多信息，请参见第 212 页中的“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。
- SAM-QFS 软件不能在共享 Sun QFS 文件系统中的 Sun QFS 客户机上工作，因此 Sun SAM-Remote 也不能。当运行软件的服务器既是某些 Sun QFS 文件系统的元数据服务器，又是其他 Sun QFS 文件的客户机时，SAM-QFS 软件和 Sun SAM-Remote 只能在由该服务器作为元数据服务器的文件系统上工作。

客户机与服务器的交互作用

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 Oracle 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 18-2 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 SAM-QFS 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《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`sam-remote(7)`”。

Sun SAM-Remote 客户机概述

Sun SAM-Remote 客户机是一个 SAM-QFS 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《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“[sam-remote\(7\)](#)”。

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。作为系统管理员，您可以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 上侦听客户机。您可以在 Oracle 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，在[第 202 页](#)中的“[配置 Sun SAM-Remote 软件](#)”中有更详细的介绍。

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。

初始化之后，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 v` 显示屏幕来查看介质是否可用。客户机上 `samu 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。最佳做法是通过 Sun SAM-Remote 服务器上的 `samu 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 202 页](#)中的“[配置 Sun SAM-Remote 软件](#)”。有关 `samu` 操作员实用程序的信息，请参见《[Sun QFS 文件系统 5.3 配置和管理指南](#)》中的[第 13 章](#)“[使用 samu 操作员实用程序](#)”。

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 SAM-QFS 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。

Sun SAM-Remote 软件中的数据完整性验证

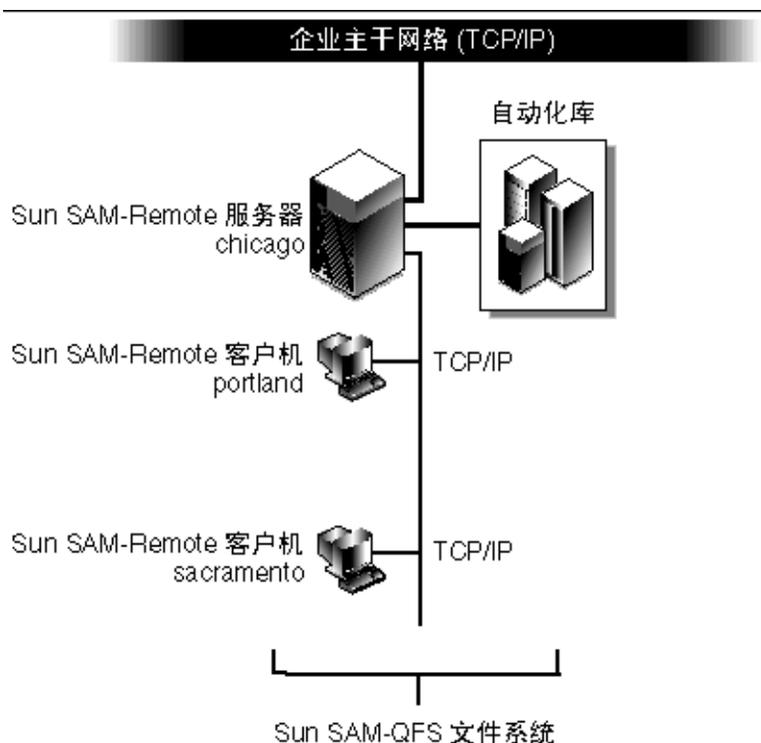
要能够使用 DIV 功能，Sun SAM-Remote 服务器以及客户机必须在 `defaults.conf` 文件中启用 `div`。有关 DIV 功能的更多信息，请参见第 13 章，[SAM-QFS 中的数据完整性验证](#)。

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。

在以下样例配置中，`portland` 和 `sacramento` 上的 SAM-QFS 文件系统是 `chicago` 上 Sun SAM-Remote 服务器的客户机。

图 18-3 Sun SAM-Remote 配置示例



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。这些过程使用图 18-3 中显示的配置。

- 按照所述的顺序执行这些过程。
- 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。
- 对于您要在其中安装 Sun SAM-Remote 软件的客户机系统，您必须具有超级用户访问权限。
- 必须在 Sun SAM-Remote 环境中所有客户机和服务器主机上安装相同的 SAM-QFS 软件发行版本和修订版本。
- 必须在 SAM-Remote 环境中的所有客户机和服务器主机上安装相同的修补程序集。

▼ 步骤 1：登录至潜在的服务器及客户机主机

- 1 以超级用户身份登录至 Sun SAM-Remote 服务器。
- 2 以超级用户身份登录至 Sun SAM-Remote 客户机。

▼ 步骤 2：检验客户机和服务器软件

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 `pkginfo -l` 命令，并检查其输出。

```
# pkginfo -l SUNWsamfsr
PKGINST: SUNWsamfsr
  PKGINST: SUNWsamfsr
  NAME: Sun SAM and Sun SAM-QFS software Solaris 11 (root)
CATEGORY: system
ARCH: i386
VERSION: 5.3.0,REV=5.11.2011.10.27
BASEDIR: /
VENDOR: Sun Microsystems, Inc.
DESC: Storage and Archive Manager File System
PSTAMP: build34-mn20111027150903
INSDATE: Oct 28 2011 02:57
HOTLINE: Please contact your local service provider
STATUS: completely installed
FILES: 641 installed pathnames
      7 shared pathnames
      23 linked files
      79 directories
      208 executables
      1 setuid/setgid executables
      96696 blocks used (approx)
```

```
chicago# pkginfo -l SUNWsamfsu
PKGINST: SUNWsamfsu
  NAME: Sun SAM and Sun SAM-QFS software Solaris 11 (usr)
CATEGORY: system
ARCH: i386
VERSION: 5.3.0,REV=5.11.2011.10.27
BASEDIR: /
VENDOR: Sun Microsystems, Inc.
DESC: Storage and Archive Manager File System
PSTAMP: build34-mn20111027150917
INSDATE: Oct 28 2011 02:58
HOTLINE: Please contact your local service provider
STATUS: completely installed
FILES: 57 installed pathnames
      9 shared pathnames
      20 directories
      19 executables
      12584 blocks used (approx)
```

```
chicago#
```

输出显示服务器 (chicago) 正在运行软件版本 5.3。在此服务器所在环境中的所有系统也必须运行版本 5.3。

2 运行 `showrev -p` 命令，并检查其输出。

```
chicago# showrev -p | grep SUNWsamfs
```

```
chicago#
```

输出显示服务器未安装 SAM-QFS 修补程序。在此服务器所在环境中的所有系统也必须运行相同的版本和修补程序级别。

如果需要执行任何软件升级，请参见《[Sun QFS 和 Sun Storage Archive Manager 5.3 安装指南](#)》中的第 10 章“升级 Sun QFS 和 SAM-QFS”。

▼ 步骤 3：在每台客户机上编辑 `mcf` 文件

`mcf` 文件用于定义文件系统。此过程将主机定义添加为 Sun SAM-Remote 客户机。

1 在 Sun SAM-Remote 服务器上，停止 SAM-QFS 功能。

2 运行 `samcmd` 命令使每个可移除介质都处于空闲状态。有关 `samcmd` 命令的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`samcmd(1M)`”。

```
# samcmd idle eq1
# samcmd idle eq2.
.
# samcmd idle eqn
```

其中的 `eq` 用于指定在 `mcf` 文件中为可移除介质驱动器定义的设备序号。

另外，还可使用 `samu` 操作员实用程序使驱动器处于空闲状态。有关信息，请参见《[Sun QFS 文件系统 5.3 配置和管理指南](#)》中的第 13 章“使用 `samu` 操作员实用程序”。

注 – 在运行下一个命令 `samd stop` 之前，所有驱动器必须均处于空闲状态，这样，归档程序、回写程序及其他进程能够完成当前任务，并能够卸载卡盒，将它们放置到存储插槽中。

3 运行带有 `stop` 选项的 `samd` 命令，以停止 `sam-amld` 守护进程及其子进程。

```
# /opt/SUNWsamfs/sbin/samd stop
```

4 在每台客户机上，打开 `/etc/opt/SUNWsamfs/mcf` 文件。

5 将每个系统定义为 Sun SAM-Remote 服务器的客户机。

示例代码显示了对客户机 `portland` 的 `mcf` 文件进行编辑使 `portland` 成为 Sun SAM-Remote 服务器 `chicago` 的客户机之后的该文件。

- 6 打开另一客户机的 `mcf` 文件。
在示例中，对客户机 `sacramento` 的 `mcf` 文件进行了编辑。
- 7 将第一台客户机 `mcf` 文件中的最后几行复制到下一台客户机的 `mcf` 文件。
在示例中，将 `portland` 的 `mcf` 文件中的最后几行复制到 `sacramento` 的 `mcf` 文件。
- 8 保存并关闭 `mcf` 文件。

示例 18-1 客户机上的 `mcf` 文件

```
# mcf file on portland
#
# Define a Sun QFS file system
#
# Equipment                               Eq Eq Family Dev Additional
# Identifier                               Ord Ty Set   St Parameters
# =====                               === == ===== == =====
samfs1                                     10 ms samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE5F00048F2Dd0s6 11 md samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE23000B24C2d0s6 12 md samfs1 on

# 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` 条目包括 Sun SAM-Remote 客户机的单行条目，以及您要配置的每个设备的伪设备条目，由 `rd` 设备类型指示。伪设备定义 Sun SAM-Remote 服务器上实际设备的网络连接。每个条目都使用下表中列出的字段。

字段	说明
Equipment Identifier	客户机配置文件的全路径名。在步骤 4 中创建客户机配置文件。
Eq Ord	用于标识此客户机系统的编号。使用此编号指定 <code>eq</code> 参数。
Eq Ty	由两个字母组成的助记符，用于标识客户机系统的类型。
Family Set	要在此服务器上使用的守护进程的系列集名称。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。 注 - Sun SAM-remote 客户机上的系列集名称必须与 Sun SAM-remote 服务器上的系列集名称一致。
Dev St	设备状态：on 或 off
Additional Parameters	可选字段。在本示例中为目录文件的路径。

▼ 步骤 4：创建 Sun SAM-Remote 客户机配置文件

Sun SAM-Remote 客户机的配置文件包含一个条目：Sun SAM-Remote 服务器的名称。

- 1 在每台客户机上，在您在 `mcf` 文件中指定的位置创建要用作 Sun SAM-Remote 客户机配置文件的文件。

例如：

```
portland# vi /etc/opt/SUNWsamfs/rmt200
```

- 2 键入 Sun SAM-Remote 服务器的名称。

以下示例显示了客户机 `portland` 的客户机配置文件，指定其 Sun SAM-Remote 服务器为 `chicago` 系统。

以下示例显示了客户机配置文件。

```
portland# cat /etc/opt/SUNWsamfs/rmt200
chicago
```

▼ 步骤 5：编辑服务器的 `mcf` 文件

- 1 在 Sun SAM-Remote 服务器上，编辑 `/etc/opt/SUNWsamfs/mcf` 文件。
- 2 添加用于将 SAM-QFS 文件系统和此系统定义为 Sun SAM-Remote 服务器的条目。您必须至少有一个 SAM-QFS 文件系统。以下示例显示了 `chicago` 上的 `mcf` 文件。

以下示例显示了服务器上的 `mcf` 文件

```
# mcf file on Sun SAM-Remote server chicago:
# Define a SAM-QFS file system
#

# Equipment                               Eq Eq Family Dev Additional
# Identifier                               Ord Ty Set   St Parameters
# =====                               === == ===== == =====
samfs1                                     10 ms samfs1 on
/dev/dsk/c6t600A0B80004850A600000F8048EF90ADd0s6 11 md samfs1 on
/dev/dsk/c6t600A0B800048505600000E9D48EF91EE0s6 12 md samfs1 on

# Define a tape library that client portland can use:
/dev/samst/c4t500104F0009C2F6Fu0 100 sn rb100 on /var/opt/SUNWsamfs/catalog/rb100.cat
/dev/rmt/0cbn                        101 li rb100 on
/dev/rmt/lcbn                         102 li rb100 on

# Define Sun SAM-Remote server chicago
#
/etc/opt/SUNWsamfs/rmt200             50 ss chicagoss on
```

注 - Sam-Remote 服务器上的系列集名称必须与 Sam-Remote 客户机上的系列集名称一致。

▼ 步骤 6：创建服务器的配置文件

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。要支持多个客户机，请配置另外一个 Sun SAM-Remote 服务器守护进程。

- 1 在服务器上，创建 Sun SAM-Remote 服务器配置文件。
- 2 使用以下格式，添加每个客户机的定义：

```
client-name
[ parameter1 ]
  media
    eq media-type regex
    [eq media-type regex ]
    [ . . . ]
  endmedia
```

下表介绍了可用于客户机的字段。

字段	定义和要求
<i>client-name</i>	标识调用 Sun SAM-Remote 守护进程所服务的系统。使用网络名称、其 IP 地址或全限定域名。 <i>client-name</i> 中的第一个字符必须是该行中的首个字符。
<i>parameter</i> (可选)	使用 <i>keyword = value</i> 对定义客户机的属性。例如，可使用 <i>parameter</i> 指定此客户机的套接字所用的网络块大小（以 KB 为单位）。此参数的格式为 <i>net_blk_size = size</i> ，其中 <i>size</i> 是一个整数，其范围为 $4 \leq \text{size} \leq 64$ 。默认值为 4，指定 4096 个字节。 <i>parameter</i> 所在的行必须以空格或制表符开头。
<i>media</i> 和 <i>endmedia</i>	包含介质定义的关键字。这两个关键字中的定义用于定义客户机可使用的介质归档卷。 注 - 这两个关键字是必需的，且必须 缩进 一个空格或制表符。

字段	定义和要求
<code>eq media-type (regex)</code>	<p>定义此客户机可使用的介质归档卷。将 <code>regex</code> 数据置于圆括号内。由于网络连接库中混用不同的介质，请在单独的行中指定每种介质类型。</p> <p>注-使用空格或制表符缩进介质定义。</p> <ul style="list-style-type: none"> ■ <code>eq</code>: mcf 文件中所示的设备编号。 ■ <code>media-type</code>: 由两个字符表示的特定介质类型，如 <code>li</code>。有关有效介质类型的信息，参见《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“<code>mcf(4)</code>”，但不要使用通用介质类型。 ■ <code>(regex)</code>: 在其中归档文件的卡盒的卷序列名 (volume serial name, VSN)，以扩展正则表达式的形式表示。有关扩展正则表达式的信息，请参见 <code>egrep(1)</code> 手册页。有关正则表达式的信息，请参见 <code>regcomp(3C)</code> 手册页。

例如，以下是有效的介质类型定义：

```
media
  100 li (VSN1)
  100 li (VSN2)
endmedia
```

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，则客户机和服务器不能共用卡盒。

以下示例显示了 Sun SAM-Remote 服务器 `chicago` 的服务器配置文件 `/etc/opt/SUNWsamfs/rmt200`。其中定义了客户机 `portland` 和 `sacramento`。

```
# Sun SAM-Remote server config file /etc/opt/SUNWsamfs/rmt200
#
portland
  media
    100 li (100031|100032|100034|100035|100037|100038)
    100 li (200001|200002|200003|200004|200005|200006)
  endmedia
#
#
sacramento
  media
    100 li (300141|300142|300143|300145|300146|300147)
    100 li (400001|400002|400003|400005|400006|400007)
  endmedia
```

▼ 步骤 7：启用归档

- 1 验证每台客户机上的 `archiver.cmd` 文件。您可能需要执行以下任务，具体取决于您的配置：

- a. 确保服务器配置文件中定义的 VSN 已指定给正确的归档集。
- b. 如果以下指令应用于要在与 Sun SAM-Remote 服务器连接的库中进行归档的归档集，请将其删除：
 - `-tapenonstop`
 - `-offline_copy direct`

- 2 在服务器和客户机上启动 SAM-QFS 进程。在客户机和服务器上运行以下命令：

```
server# samd start
```

- 3 检查 Sun SAM-Remote 连接的状态。

- a. 在服务器和客户机上运行 `samu` 命令。

有关更多信息，请参见《Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual》中的“`samu(1M)`”或《Sun QFS 文件系统 5.3 配置和管理指南》中的第 13 章“使用 `samu` 操作员实用程序”。

- b. 在每个客户机上，查看 `samu` 实用程序的 `s` 设备状态显示屏幕。

以下示例显示了 Sun SAM-Remote 客户机 `portland` 上的状态 `s` 显示屏幕。设备类型 `sc` 用于标识 Sun SAM-Remote 客户机。该行以后的消息表示已建立与服务器 `chicago` 的连接。

```
Device status samcmd      5.3.0 20:44:09 Jul 30 2009
samcmd on portland

ty   eq state  device_name                fs status
sc   200 on    /etc/opt/SUNWsamfs/rmt200  200  -----r
      Remote server 10.1.229.92 connected
rd   201 on    /dev/samrd/rd0             200  -----
rd   202 on    /dev/samrd/rd1             200  -----
hy   203 on    historian                    203  -----
```

- c. 在服务器上，查看 `samu` 实用程序的 `s` 设备状态显示屏幕。

以下示例显示了 Sun SAM-Remote 服务器 `chicago` 上的 `samu` 状态 `s` 显示屏幕。设备类型 `ss` 用于标识 Sun SAM-Remote 服务器。

```
Device status samcmd      5.3.0 20:40:05 Jul 30 2009
samcmd on chicago

ty   eq state  device_name                fs status
sk   1 on     /etc/opt/SUNWsamfs/SL500_SAM  1  m-----r
```

```

running
li      2 on      /dev/rmt/0cbn      1  -----p
empty
li      3 on      /dev/rmt/1cbn      1  -----p
empty
ss      50 on     /etc/opt/SUNWsamfs/rmt200  50  -----o-r
hy      51 on     historian            51  -----

```

d. 在服务器上，查看 **samu** 实用程序的 **R SAM-Remote** 显示屏幕。

以下示例显示了 Sun SAM-Remote 服务器 **chicago** 中的 **samu Sun SAM-Remote R** 显示屏幕。

```
Remote server eq: 50   addr: 00003858 samu      5.3.0 20:41:38 Jul 30 2009
```

message:

```
Client IPv4: jimmy 192.10.10.3                port - 5000
          client index - 0 port - 0 flags - 0004
```

```
Client IPv4: portland 10.1.229.97            port - 5000
          client index - 1 port - 32848 flags - 0005 connected
```

如果 Sun SAM-Remote 配置包括多个客户机，请按 **CONTROL-f** 组合键滚动查看每个客户机。每个客户机由名称及其 **client index** 字段（0 到 9 之间的一个整数，表示该客户机在为此服务器守护进程定义的 10 个客户机中的顺序）来标识。网络块大小、最大文件大小及最小文件大小的单位均为“字节”。标志用于表示连接的状态。

标志	编号	定义
0x00000000	0004	没有连接。
0xc0000000	0005	已建立连接。

4 在服务器上，使用 **samu** 实用程序的 **v** 自动化库目录显示屏幕，以确保客户机上有 **Sun SAM-Remote** 目录。从 **samu** 命令提示符下，运行以下命令：

```
: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）。

该示例显示了 **portland** 可以访问的卷。

```
Robot VSN catalog by slot : eq 200 samu 5.3.0 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 在每一台客户机上，运行带有 **-A** 选项的 **archiver** 命令，检验是否在进行从客户机到服务器的归档。

此命令将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《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“**archiver(1M)**”。

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。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 - SAM-QFS 软件不强制实施这些限制。

为了避免丢失数据，确保遵循以下限制：

- 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** 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，而不得相互使用对方的卡盒。

您可以在 Sun SAM-Remote 服务器的 `/etc/opt/SUNWsamfs/recycler.cmd` 文件中创建一个 `no_recycle` 列表，以防止意外回收 Sun SAM-Remote 客户机使用的 VSN。不过，对 `no_recycle` 列表中的卷使用 `chmed` 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，才允许在 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。服务器配置文件介质定义行（*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。

在 Sun SAM-Remote 环境中回收（方法 1）

本节中的过程介绍了一种在 Sun SAM-Remote 环境中启用回收的方法。这些过程假定服务器名为 *sky*，客户机名为 *zeke*。

注 - 要在 Sun SAM-Remote 环境中使用回收程序，您必须严格执行本过程，且您的配置经测试可以正确执行回收过程。

服务器的配置文件

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。以下示例显示了这些文件。

示例 18-2 服务器 jimmy 上的 *mcf* 文件

```
# This is the mcf file for the server (jimmy).
# The server parameters file (rmt2000) points
# back to the correct automated library's equipment number
# (1000) for the SL8500 tape library.
#

# Equipment                               Eq Eq Family Dev Additional
# Identifier                               Ord Ty Set   St  Parameters
# =====                               === == ===== == =====
samfs1
/dev/dsk/c6t600A0B80004850A600000F8048EF90ADd0s0 101 mm samfs1 on
/dev/dsk/c6t600A0B800048505600000E9D48EF91EEd0s6 102 mr samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE5F00048F2Dd0s6 103 mr samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE23000B24C2d0s6 104 mr samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE50000ADAECd0s6 104 mr samfs1 on

samfs2
/dev/dsk/c7t60003BA13F71500048EDCE720001B17Fd0s6 201 md samfs2 on
/dev/dsk/c7t60003BA13F71500048EDCE7200014BEAd0s6 202 md samfs2 on

# SL8500
/etc/opt/SUNWsamfs/T10K      1000  sk    T10K  on    /etc/opt/SUNWsamfs/T10K_cat
/dev/rmt/4cbn               1001  ti    T10K  on
/dev/rmt/5cbn               1002  ti    T10K  on
/dev/rmt/0cbn               1003  ti    T10K  on
```

示例 18-2 服务器 jimmy 上的 mcf 文件 (续)

```

/dev/rmt/1cbn          1004   ti    T10K   on
/dev/rmt/6cbn          1005   ti    T10K   on
/dev/rmt/7cbn          1006   ti    T10K   on
/dev/rmt/2cbn          1007   ti    T10K   on
/dev/rmt/11cbn         1008   ti    T10K   on
/dev/rmt/10cbn         1009   ti    T10K   on
/dev/rmt/12cbn         1010   ti    T10K   on

# Define Sun SAM-Remote server jimmy
/etc/opt/SUNWsamfs/rmt2000 2000 ss jimmy on

```

示例 18-3 服务器 jimmy 上的服务器配置文件

```

# Server configuration file /etc/opt/SUNWsamfs/rmt2000 on jimmy.
# The eq of the automated library MUST match the eq of the
# automated library that you want to use in the mcf file.
tera
  media
    1000 ti 00002[0-9]
  endmedia

```

客户机的配置文件

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。以下示例显示了这些文件。

示例 18-4 客户机 tera 上的 mcf 文件

```

# mcf file for client (tera)
#

# Equipment                               Eq Eq Family Dev Additional
# Identifier                               Ord Ty Set   St  Parameters
# =====                               === == ===== == =====
samfs1                                     100 ms samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE5F00048F2Dd0s6 101 md samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE23000B24C2d0s6 102 md samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE50000ADAECd0s6 103 md samfs1 on

# Define a L500 with 2 drives
/dev/samst/c4t500104F0009C2F6Fu0 300 rb L500 on
/dev/rmt/0cbn                       301 li L500 on
/dev/rmt/1cbn                       302 li L500 on

# Define tera as a Sun SAM-Remote client using jimmy as the server
/etc/opt/SUNWsamfs/rmt2000          2000 ss jimmy on
/dev/samrd/rd0                      2001 rd jimmy on
/dev/samrd/rd1                      2002 rd jimmy on
/dev/samrd/rd2                      2003 rd jimmy on
/dev/samrd/rd3                      2004 rd jimmy on
/dev/samrd/rd4                      2005 rd jimmy on
/dev/samrd/rd5                      2006 rd jimmy on
/dev/samrd/rd6                      2007 rd jimmy on

```

示例 18-4 客户机 tera 上的 mcf 文件 (续)

```
/dev/samrd/rd7                2008 rd jimmy on
```

示例 18-5 客户机 tera 上的客户机配置文件

```
# cat /etc/opt/SUNWsamfs/rmt2000
# File /etc/opt/SUNWsamfs/rmt2000 on Sun SAM-Remote client tera: jimmy
```

▼ 如何配置回收 (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。

注 - 请勿在服务器上使用 `chmed` 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。

开始本过程之前，请先阅读第 175 页中的“关于回收”。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。如果您尚不熟悉回收过程，请进行学习了解。

1 确保 Sun SAM-Remote 客户机和服务器配置正确，且可以进行归档。

有关更多信息，请参见第 202 页中的“配置 Sun SAM-Remote 软件”，其中包含有关配置 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。

以下示例显示了客户机 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。

```
# This is file /etc/opt/SUNWsamfs/archiver.cmd
# on Sun SAM-Remote client zeke.
#
archivemeta = off
archmax = li 12G

fs = samfs1
logfile = /var/adm/samfs1.archiver.log
no_archive tmp
all .
    1 -norelease 10m
    2 -norelease 10m

params
allsets -sort path -offline_copy stageahead -reserve set
```

```

allsets -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 2 -dataquantity 100G
allsets -ignore
allsets.1 -startage 8h -startsize 10G -drives 2
allsets.2 -startage 24h -startsize 20G -drives 2 -archmax 24G
endparams

vsns
all.1 li ^10.*
all.2 li ^20.*
endvsns

```

示例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_vsncount2` 指令指定回收程序一次清空两个 VSN。不要让回收过程造成系统崩溃。

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。

以下示例显示了客户机 `zeke` 上的 `recycler.cmd` 文件，已对此文件进行编辑，以指定回收程序日志文件。

```

#
# This is the /etc/opt/SUNWsamfs/recycler.cmd file
# on client zeke.
#
logfile = /var/adm/recycler.log

```

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。

以下示例显示了服务器 `sky` 上的 `archiver.cmd` 文件。此文件指定按归档集执行归档过程。

```

# 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
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，指定用于接收回收程序输出的回收程序日志文件和用于 Sun SAM-Remote 客户机 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。

以下示例显示了服务器 sky 上的 `recycler.cmd` 文件，已对此文件进行编辑，以指定回收程序日志文件。

```

#
# This is the /etc/opt/SUNWsamfs/recycler.cmd file
# on Sun SAM-Remote server sky.
#
logfile = /var/opt/SUNWsamfs/recycler/recycler.log
adicl -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` 命令测试回收程序。

在 Sun SAM-Remote 客户机系统上运行回收程序可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。Sun SAM-Remote 客户机与 SAM-QFS 服务器之间进行通信，告知对方是否具有归档副本。所有此类信息都是从本地 SAM-QFS 文件系统在本地提供的。



注意 – 标记卡盒将会破坏卡盒上的所有数据。

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` 命令的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`sam-recycler(1M)`”。

7 检查回收程序日志文件以查找消息。

Recycling is ignored on this archive set.

请参见示例 18-6。

8 在 Sun SAM-Remote 服务器上运行 `sam-recycler` 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` 命令的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`sam-recycler(1M)`”。另请参见示例 18-7。

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。

- 在客户机日志文件中，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 227 页中的“[如何回收 no-data VSN](#)”。
 - `partially full`。要回收 `partially full` VSN，请参见“[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。

关于如何配置回收（方法 1）的示例

示例 18-6 日志文件样例

以下示例显示了一个日志文件样例。

```
# recycler.log from client zeke.
===== Recycler begins at Mon Jun 4 09:49:41 2001 =====
Initial 7 catalogs:
```

示例 18-6 日志文件样例 (续)

```

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.

```

示例 18-6 日志文件样例 (续)

```

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

```

示例 18-6 日志文件样例 (续)

```

---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 skyr:at:000020
empty VSN 0 0 0 0 100 skyr:at:000022
empty VSN 0 0 0 0 100 skyr:at:000023
empty VSN 0 0 0 0 100 skyr:at:000024
in multiple sets 0 0 0 0 100 skyr:at:000021

---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 =====

```

示例 18-7 回收程序日志文件样例

以下示例显示了一个回收程序日志文件样例。

```

# 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

```

示例 18-7 回收程序日志文件样例 (续)

```
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
```

示例 18-7 回收程序日志文件样例 (续)

```

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.

```

示例 18-7 回收程序日志文件样例 (续)

```
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.
```

示例 18-7 回收程序日志文件样例 (续)

```
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.
```

示例 18-7 回收程序日志文件样例 (续)

```
VSN not in correct media changer.
Checking 000027. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000028. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000023. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
```

```
Checking 000024. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000016. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN001. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN002. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN004. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000000. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
No candidate was found in this media changer.
34 VSNs:
```

```
---Archives--- -----Percent----- testset.1
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 100 0 adic1:at:000010
no-data VSN 0 0 0 100 0 adic1:at:000014
```

```
---Archives--- -----Percent----- testset.2
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 100 0 adic1:at:000019
partially full 677 2.3G 5 93 2 adic1:at:000015
```

```
---Archives--- -----Percent----- allsam1.1
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
partially full 97 173.8M 1 9 90 adic1:at:000001
no-data VSN 0 0 0 2 98 adic1:at:000003
no-data VSN 0 0 0 2 98 adic1:at:000004
empty VSN 0 0 0 0 100 adic1:at:000005
empty VSN 0 0 0 0 100 adic1:at:000002
```

```
---Archives--- -----Percent----- allsam1.2
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
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
```

示例 18-7 回收程序日志文件样例 (续)

```

no_recycle VSN 0 0 0 51 49 adicl:at:000020
no-data VSN 0 0 0 17 83 adicl:at:000017
no-data VSN 0 0 0 10 90 adicl:at:000018
empty VSN 0 0 0 100 adicl:at:CLN003

no_recycle VSN 0 0 0 0 100 adicl:at:000021
no_recycle VSN 0 0 0 0 100 adicl:at:000022
no_recycle VSN 0 0 0 0 100 adicl:at:000027
no_recycle VSN 0 0 0 0 100 adicl:at:000028
no_recycle VSN 0 0 0 0 100 adicl:at:000023
no_recycle VSN 0 0 0 0 100 adicl:at:000024
empty VSN 0 0 0 0 100 adicl:at:000016
empty VSN 0 0 0 0 100 adicl:at:CLN001
empty VSN 0 0 0 0 100 adicl:at:CLN002
empty VSN 0 0 0 0 100 adicl:at:CLN004
partially full 12 88.3k 0 0 100 adicl:at:000000

Recycler finished.
===== Recycler ends at Mon Jun 4 09:51:05 2001 =====

```

▼ 如何回收 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 文件中是否存在任何 no-data VSN。

在本示例中，可以考虑回收客户机 zeke 中的 VSN 000029 和 000026，因为它们是 no-data VSN，如以下代码中所示。

```

# From the client zeke recycler.log file:
---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

```

2 如果找到 no-data VSN，检查服务器中的 recycler.log 文件，确定它们与在服务器回收程序日志文件中是否相同。

验证来自服务器的有效数据是否没有被归档在这些 VSN 上。

以下代码显示了服务器 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。

```

# From the Server log file:
---Archives--- -----Percent----- adicl
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 100 0 adicl: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 0 100 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` 或 `odlabel` 命令重新标记这些 VSN。



注意 - 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# tptlabel -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，请使用 `odlabel` 命令。有关更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`odlabel(1M)`”。

▼ 如何回收 partially full VSN

还可以回收那些报告为 `partially full` 状态的 VSN。

- 1 检查客户机的 `recycler.log` 文件中是否存在任何 `partially full VSN`。

在示例中，您可以考虑回收客户机 `zeke` 的 VSN 000025，因为其状态为 `partially full`，如下所示。

```
# From the client zeke recycler.log file:
---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 0 100 skyrs:at:000028
empty VSN 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。

- 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。

```
# 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
.
.
.
```

- 在服务器上，重新归档 VSN 上的活动文件。

```
sky# chmed +c at.000025
```

有关 `chmed` 命令的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`chmed(1M)`”。

- 在客户机上，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 的每一个活动文件。

- 启动归档程序。

您可以让归档程序正常运行，也可以通过客户机上的 `samu(1M)` 实用程序键入 `:arrun`。有关 `:arrun` 命令的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`samu(1M)`”。

- 归档完成后，发出 `sam-recycler` 命令在客户机上再次运行回收程序。

```
zeke# sam-recycler -dvx
```

这可确保所有活动文件均已重新归档。

- 如果 `Count`、`Bytes` 和 `Use` 字段值均为 0（零），请使用 `tplabel` 或 `odlabel` 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` 命令。有关 `odlabel` 命令的更多信息，请参见《[Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#)》中的“`odlabel(1M)`”。

在 Sun SAM-Remote 环境中回收（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。

▼ 如何配置回收（方法 2）

- 1 在 Sun SAM-Remote 客户机上，运行 `sam-recycler` 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“回收 no-data VSN”。
 - partially full。要回收 partially full VSN，请参见“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` 命令，在选定的 VSN 上设置回收标志。
例如：

```
server# chmed +c at.00025
```
- 4 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。
- 5 在 Sun SAM-Remote 服务器上，运行 `tplabel` 或 `odlabel` 命令，根据归档介质重新标记卷。
- 6 在 Sun SAM-Remote 服务器上，清除会阻止将卷用于在 Sun SAM-Remote 客户机上执行归档的任何标志，例如 `R` 或 `c`。