



Sun™ SAM-FS 和 Sun™ SAM-QFS 存储及存档管理指南

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 U.S.A.
650-960-1300

部件号: 816-7695-10
2002 年 10 月, 修订版 A

请将有关本文档的意见发送至: docfeedback@sun.com

版权所有 ©2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, CA 95054 U.S.A.。保留所有权利。

Sun Microsystems, Inc. 拥有本产品或文档所含技术的知识产权。重点来讲（但不限于此），这些知识产权包括 <http://www.sun.com/patents> 网站列出的一个或多个美国专利，以及一个或多个在美国或其它国家/地区的其它专利或正在申请中的专利。

本产品或文档根据限制其使用、复制、分发和反编译的许可证条款进行分发。没有 Sun 及其许可证发行者（如果有）事先书面授权，不得以任何形式、任何方式复制本产品或文档的任何部分。

第三方软件，包括字体技术，均已从 Sun 供应商获得版权和许可。产品的某些部件可能源于 Berkeley BSD 系统，Sun 已从 University of California 获得使用许可。UNIX 是在美国及其它国家/地区的注册商标，Sun 已从 X/Open Company, Ltd. 获得独家使用授权。

Sun、Sun Microsystems、Sun 徽标、AnswerBook2、docs.sun.com、Sun Enterprise、Ultra、Java、OpenWindows、Solaris、SunSolve 和 Sun StorEdge 是 Sun Microsystems, Inc. 在美国以及其它国家/地区的商标、注册商标或服务商标。所有 SPARC 商标均为 SPARC International, Inc. 在美国和其它国家/地区的商标或注册商标，必须经过许可才能使用。带有 SPARC 商标的产品以 Sun Microsystems, Inc. 开发的体系结构为基础。“能源之星”徽标是 EPA 的注册商标。Adobe 是 Adobe Systems, Incorporated. 的注册商标。

OPEN LOOK 和 Sun™ Graphical User Interface 是 Sun 专门为其用户和许可证获得者开发的。Sun 感谢 Xerox 在用户界面形象化和图形化研发方面为计算机行业所做的先导性贡献。Sun 已从 Xerox 获得对图形用户界面 (GUI) 的非独占使用许可。该许可也涵盖实施 OPEN LOOK GUI 的 Sun 许可获得者，而其它情况则应符合 Sun 的书面许可协议。

文档以“原样”提供。除非有关的免责声明在法律上无效，否则我们拒绝承担任何明确或暗示的条件、表示和担保，包括任何对适销性、特定用途的适用性或非侵犯性的暗示担保。



请回收
利用



Adobe PostScript

目录

序言 xix

本书的结构安排 xix

相关文档 xx

访问 Sun 联机文档 xx

使用许可 xxi

诊断程序 xxi

安装帮助 xxii

使用 UNIX 命令 xxii

印刷约定 xxii

Shell 提示符 xxiii

Sun 欢迎您发表意见 xxiii

1. 概述 1

性能 1

 存档 2

 释放 2

 登台 2

 回收 3

存储设备 3

命令 3

用户命令	4
一般系统管理员命令	5
文件系统命令	6
自动化库命令	7
存档程序命令	8
专用维护命令	8
应用软件程序员界面	9
可操作实用程序	9
2. 基本操作	11
自动化库硬件和软件	12
配置自动化库	12
参数文件	13
mcf 文件	13
defaults.conf 文件	14
条码	15
驱动器计时值	15
共享驱动器	16
载入通知	17
目录操作	18
▼ 建立库目录	18
▼ 检查库目录	19
▼ 导入和导出卡盒	20
跟踪导出的介质 — Historian	20
约定	21
命令变量	21
术语	22
自动化库操作	22
▼ 自动清洁和自动载入	23

▼ 停止可移动介质操作	23
▼ 启动可移动介质操作	24
▼ 打开自动化库	24
▼ 关闭自动化库	25
▼ 将卡盒载入自动化库	25
▼ 从驱动器中卸载卡盒	26
标记卡盒	26
▼ 标记或重新标记磁带	27
▼ 标记或重新标记光盘	28
▼ 核查卷	28
▼ 核查自动化库（仅限于直接连接）	29
使用清洁卡盒	30
▼ 重设清洁循环次数	30
▼ 使用带有条码的清洁卡盒	31
▼ 使用不带条码的清洁卡盒	31
▼ 清洁磁带驱动器	32
▼ 清除介质错误	33
▼ 从驱动器中取出卡住的卡盒	34
导入和导出卡盒（配有邮箱的系统）	35
▼ 使用邮箱导入卡盒	36
▼ 使用邮箱导出卡盒	37
导入和导出卡盒（未配邮箱的系统）	37
▼ 在没有邮箱的情况下导入卡盒	37
▼ 在没有邮箱的情况下导出卡盒	38
手动载入驱动器操作	39
▼ 载入卡盒	39
▼ 卸载卡盒	39
▼ 查看库目录	40

3. 特定自动化库的基本操作 41

ADIC/Grau 自动化库 42

配置 42

mcf 文件 42

ADIC/Grau 参数文件 42

建立目录 44

操作 45

▼ 导入卡盒 45

▼ 导出卡盒 46

诊断信息 47

ADIC Scalar 系列库 47

配置 47

操作 47

Ampex 自动化库 47

▼ 配置 Ampex 库 48

▼ 格式化 D2 磁带 49

操作 49

Fujitsu LMF 自动化库 49

配置 50

mcf 文件 50

Fujitsu LMF 参数文件 50

▼ 建立目录 51

操作 52

▼ 导入卡盒 52

▼ 导出卡盒 53

IBM 3584 UltraScalable 磁带库 54

导入 54

清洁 54

分区	54
IBM 3494 库	55
Sony 直接连接 B9 和 B35 自动化库	55
配置	55
操作	55
Sony 直接连接 8400 PetaSite 自动化库	56
▼ 导入磁带	56
▼ 导出磁带	57
▼ 在邮箱端口未用作存储端口时导出磁带	57
▼ 在邮箱端口用作存储端口时导出磁带	58
▼ 如何将卡盒移至另一个端口	58
Sony 网络连接自动化库	59
配置	59
mcf 文件	59
Sony 参数文件	59
▼ 建立目录	61
操作	62
▼ 导入卡盒	62
▼ 导出卡盒	63
StorageTek ACSLS 连接自动化库	63
配置	64
mcf 文件	64
StorageTek 参数文件	64
ssi.sh 脚本	67
▼ 建立目录	67
常见问题和错误消息	68
操作	70
▼ 导入磁带	70
▼ 使用邮箱导出磁带	71

4. 存档 73

存档程序 — 操作理论 73

存档组 74

存档操作 75

步骤 1: 扫描要存档的文件 75

步骤 2: 编辑存档请求 76

步骤 3: 安排存档请求 77

步骤 4: 对存档请求中的文件进行存档 79

默认输出范例 80

存档后台程序 80

存档日志文件和事件日志 81

archiver.cmd 文件说明 82

archiver.cmd 文件 83

archiver.cmd 文件示例 84

archiver.cmd 指令 86

全局存档指令 86

interval 指令: 指令存档时间间隔 86

bufsize 指令: 设置存档程序缓冲器大小 87

drives 指令: 控制用于存档活动的驱动器数量 88

archmax 指令: 控制存档文件的大小 88

ovflmin 指令: 控制卷溢出功能 89

wait 指令: 推迟存档程序启动 89

notify 指令: 重命名事件通知脚本 90

logfile 指令: 指定存档程序日志文件 90

用于控制特定文件系统存档活动的指令 91

fs 指令: 指定文件系统 91

其它文件系统指令 91

- 存档组分配指令 91
 - 文件大小 *search_criteria*: -minsize 和 -maxsize 93
 - 所有者和群组 *search_criteria*: -user 和 -group 93
 - 使用样式匹配的文件名 *search_criteria*: -name regex 93
 - 释放及登台 *file_attributes*: -release 和 -stage 96
 - 存档组成员关系冲突 97
- 存档副本指令 98
 - 存档之后释放磁盘空间: -release 98
 - 推迟释放磁盘空间: -norelease 98
 - 设置存档时限 99
 - 自动取消存档 100
 - 为元数据指定多份副本 100
- 存档组参数 101
 - 设置存档程序缓冲器大小 101
 - 指定存档缓冲器锁定 102
 - 将多个驱动器分配给存档组 102
 - 在各个驱动器之间分割存档请求 103
 - 指定回收 104
 - 联合存档 104
 - 控制取消存档 105
 - 控制存档文件的写入方式 106
 - 保留卷 106
 - 设置存档属性 109
- VSN 关联指令 111
- VSN 池指令 112
- 磁盘存档 114
 - 定义磁盘 VSN 115
 - diskvols.conf 文件 115
 - 配置原则 116

定义磁盘存档组	116
示例 1	117
示例 2	118
存档程序示例	119
示例 1	120
示例 2	121
示例 3	124
示例 4	129
存档程序原则	132
排除存档程序故障	133
文件未被存档的原因	134
其它存档程序诊断方法	135
文件未被释放的原因	135

5. 释放 137

释放程序概述	138
操作原理	138
定义	139
部分释放和部分登台	140
系统管理员选项概述	141
用户选项概述	142
releaser.cmd 文件	142
指定与时段和大小相关的释放优先级指令	143
文件时段	143
文件大小	144
指定用于单个文件系统的指令	145
指定调试指令	146
指定最短驻留时间	146

指定日志文件	146
限制释放已存档的文件	148
archiver.cmd 文件在登台过程中的作用	148
配置释放程序	149
手动运行释放程序	150
排除释放程序的故障	151
6. 登台	153
stager.cmd 文件	153
指定驱动器数量	154
设置登台缓冲器大小	155
指定日志文件	156
指定登台请求的数量	157
stager.cmd 文件示例	157
archiver.cmd 文件在登台过程中的作用	158
使用 preview.cmd 文件对预备请求进行排序	158
VSN 和时段指令（全局）	159
界限指令（全局或文件系统专用）	160
计算预备请求的总优先级	161
如何设置预备请求的优先级方案	161
示例 1：强制执行登台请求	162
示例 2：强制执行存档请求	163
示例 3：按介质确定请求的优先级	163
示例 4：确定复杂请求的优先级	164
7. 回收	165
概述	165
回收磁盘存档副本	166
回收可移动介质存档副本	167

回收指令 167

logfile 指令 167

no_recycle 指令 168

库指令 168

配置回收程序 169

▼ 步骤 1: 配置 recycler.cmd 文件 (可选) 170

recycler.cmd 文件示例 171

▼ 步骤 2: 编辑 archiver.cmd 文件 (可选) 172

▼ 步骤 3: 运行回收程序 174

▼ 步骤 4: 为回收程序创建 crontab 文件 (可选) 175

▼ 步骤 5: 删除 -recycle_ignore 和 ignore 指令 175

▼ 步骤 6: 创建 recycler.sh 文件 176

排除回收程序的故障 179

8. 图形用户界面 (GUI) 工具 181

操作员权限级别 182

使用 libmgr(1M) 182

▼ 启动库管理器 182

▼ 重设库管理器的显示屏幕、图像和标题 183

库管理器显示屏幕 183

▼ 执行传输器操作 184

▼ 执行介质操作 184

▼ 执行介质驱动器操作 185

▼ 查看文件系统状态及属性 185

▼ 查看介质驱动器状态及属性 187

▼ 查看 VSN 目录显示屏幕 189

使用 samtool(1M) 191

▼ 启动和退出 samtool(1M) 191

选择工具	191
▼ 更新显示屏幕	192
▼ 更改刷新率	192
▼ 更新工具显示屏幕	192
▼ 控制自动刷新功能	192
▼ 管理屏幕资源	192
▼ 显示联机帮助	192
使用 robottool(1M)	193
▼ 启动 robottool(1M)	195
传输器	195
VSN 目录	196
设备	196
查看状态信息	197
管理传输器	198
▼ 执行全面核查	198
▼ 更改传输器的状态	198
导入和导出介质	199
▼ 将介质导入传输器	199
▼ 从传输器导出卡盒	199
载入和卸载存储匣	200
▼ 载入存储匣	200
▼ 卸载存储匣	200
使用卷	200
▼ 显示条码而不显示访问时间	200
▼ 查找 VSN	200
▼ 查找特定端口编号中的 VSN	201
▼ 选择 VSN	201
▼ 核查 VSN	201

- ▼ 导出卷 201
- ▼ 载入卷 202
- ▼ 卸载卷 202
- ▼ 标记卷 202
- ▼ 移动介质 204
- 查看 VSN 状态信息 205
- 查看设备信息 205
- 使用 devicetool(1M) 205
 - ▼ 启动 devicetool(1M) 206
 - ▼ 更改显示格式 208
 - ▼ 查看显示字段 208
 - 控制设备 209
 - ▼ 选择设备 209
 - ▼ 更改设备状态 210
 - ▼ 卸载设备 210
 - ▼ 核查设备 210
 - ▼ 标记设备中的卷 211
 - ▼ 设置阈值 212
 - ▼ 设置读前区和写后区 212
- 使用 previewtool(1M) 212
 - ▼ 更改显示格式 214
 - previewtool(1M) 显示屏幕字段 214
 - ▼ 清除安装请求 215
- 9. 使用 samu(1M) 操作员实用程序 217**
 - 概述 217
 - ▼ 启动 samu(1M) 218
 - ▼ 停止 samu(1M) 218
 - 与 samu(1M) 交互操作 218

输入设备	220
获取联机帮助	220
操作员显示屏幕	222
(a) — 存档程序状态显示屏幕	223
显示屏幕范例	223
字段说明	223
(c) — 设备配置显示屏幕	224
显示屏幕范例	224
字段说明	224
(d) — 后台跟踪控制显示屏幕	225
显示屏幕范例	225
(f) — 文件系统显示屏幕	226
显示屏幕范例	226
字段说明	226
(l) — 许可证显示屏幕	227
显示屏幕范例	227
(m) — 大容量存储设备状态显示屏幕	228
显示屏幕范例	228
字段说明	228
(n) — 登台状态显示屏幕	229
显示屏幕范例	229
(o) — 光盘状态显示屏幕	230
显示屏幕范例	230
字段说明	230
(p) — 可移动介质载入请求显示屏幕	231
显示屏幕范例 1	231
显示屏幕范例 2	231
字段说明	232
标记	232

(r) — 可移动介质状态显示屏幕	233
显示屏幕范例	233
字段说明	233
(s) — 设备状态显示屏幕	235
显示屏幕范例	235
字段说明	235
(t) — 磁带驱动器状态显示屏幕	236
显示屏幕范例	236
字段说明	236
(u) — 登台队列显示屏幕	237
显示屏幕范例	237
字段说明	237
(v) — 传输器目录显示屏幕	239
显示屏幕范例	239
字段说明	239
标记	240
(w) — 待定登台队列显示屏幕	241
显示屏幕范例	241
字段说明	241
操作员显示屏幕状态代码	242
可移动介质设备显示屏幕状态代码	242
文件系统显示屏幕状态代码	243
操作员显示屏幕设备状态	244
操作员命令	245
存档程序命令	245
设备命令	246
显示控制命令	247

文件系统命令 249

:meta_timeo *eq interval* 命令 249

:notrace *eq* 命令 249

:partial *eq size* 命令 249

:readahead *eq contig* 命令 249

:thresh *eq high low* 命令 250

:trace *eq* 命令 250

:writebehind *eq contig* 命令 250

传输器命令 251

:audit [-e] *eq* [:slot [:side]] 命令 251

:export *eq:slot* 和 :export *mt.vsn* 命令 251

:import *eq* 命令 251

:load *eq:slot* [:side] 和 :load *mt.vsn* 命令 251

其他命令 252

:clear *vsu* [*index*] 命令 252

:dtrace 命令 252

:mount *mntpt* 命令 252

:open *eq* 命令 253

:read *addr* 命令 253

:snap [*filename*] 命令 253

:! *shell_command* 命令 253

10. 升级环境 255

在自动化库中添加端口 256

▼ 在库中添加端口 256

升级或更换库 257

▼ 更换或升级库 257

升级 DLT 磁带驱动器 260

▼ 升级磁带驱动器 260

11. 高级内容	263
设备日志	263
何时使用设备日志	264
启用设备日志	265
第 1 种方法	265
第 2 种方法	266
可移动介质文件	266
卷溢出	267
分段文件	268
存档	269
故障恢复	269
系统错误工具报告	269
▼ 启用 SEF 报告	270
SEF 报告输出	270
sefreport(1M) 命令	273
管理 SEF 日志文件	273
词汇表	275
索引	287

序言

本手册 *Sun SAM-FS* 和 *Sun SAM-QFS 存储及存档管理指南* 介绍 Sun SAM-FS 和 Sun SAM-QFS 4.0 版本中支持的存储及存档管理软件。该软件可以自动将文件从联机磁盘复制到存档介质中。存档介质可由联机磁盘或可移动介质卡盒组成。

Sun Solaris™ 操作环境 (OE) 7、8 和 9 版本支持 Sun SAM-FS 和 Sun SAM-QFS 4.0 版本。

本手册适用于负责配置和维护 Sun SAM-FS 和 Sun SAM-QFS 软件的系统管理员。身为系统管理员，您必须通晓各种 Sun Solaris 过程，包括创建帐户、执行系统备份和其它基本的 Sun Solaris 系统管理员任务。

本书的结构安排

本手册包括以下章节：

- 第 1 章，产品概述。
- 第 2 章，介绍基本操作。本章所述的信息适用于大多数自动化库和手动装载的设备。
- 第 3 章，介绍如何依据各类库的专用操作说明来管理相应库中的卡盒。本章介绍了这些库及其相应的基本操作步骤。
- 第 4 章，说明存档过程。
- 第 5 章，说明释放过程。
- 第 6 章，说明登台过程。
- 第 7 章，说明回收过程。
- 第 8 章，介绍如何使用可用于 Sun SAM-FS 和 Sun SAM-QFS 软件的图形用户界面 (GUI) 工具。
- 第 9 章，介绍如何使用 `samu(1M)` 操作员实用程序。
- 第 10 章，介绍 Sun SAM-FS 和 Sun SAM-QFS 环境专用的升级过程。

- 第 11 章，介绍 Sun SAM-FS 和 Sun SAM-QFS 操作中的高级主题。

词汇表，定义了本手册及其它 Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 文档中使用的术语。

相关文档

本手册是 Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 软件产品操作文档系列的一部分。表 P-1 列出了这些 4.0 版产品的完整文档系列。

表 P-1 相关文档

书名	部件号
<i>Sun SAM-Remote 管理员指南</i>	816-7837
<i>Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 故障恢复指南</i>	816-7680
<i>Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 文件系统管理员指南</i>	816-7685
<i>Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 安装与配置指南</i>	816-7690
<i>Sun SAM-FS 和 Sun SAM-QFS 存储及存档管理指南</i>	816-7695
<i>Sun QFS、Sun SAM-FS 和 Sun SAM-QFS README 文件</i>	816-7700

请注意，《*Sun SAM-Remote 管理员指南*》尚未更新到 4.0 版。该手册的更新版本将在以后推出。

访问 Sun 联机文档

Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 软件销售套件中附带了这些产品文档的 PDF 文件。用户可从下列位置查看这些 PDF 文件：

1. Sun Network Storage 文档网站

此网站包含许多存储软件产品的文档。

a. 若要访问该网站，请输入下面的 URL：

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

屏幕上会出现 Storage Software（存储设备软件）网页。

b. 从下面的列表中单击适当的链接:

Sun QFS Software (Sun QFS 软件)

Sun SAM-FS and Sun SAM-QFS Software (Sun SAM-FS 和 Sun SAM-QFS 软件)

2. docs.sun.com

此网站包含 Solaris 和其它多个 Sun 软件产品的文档。

a. 若要访问该网站，请输入下面的 URL:

docs.sun.com

屏幕上会出现 docs.sun.com 网页。

b. 通过在搜索框中搜索下列项目之一来查找适用的产品文档:

- Sun QFS
- Sun SAM-FS
- Sun SAM-QFS

查看 PDF 文件需要使用 Acrobat Reader 软件。用户可从下面的网站免费下载该软件:

www.adobe.com/products/acrobat/readstep.html

使用许可

有关获取 Sun QFS、Sun SAM-FS 或 Sun SAM-QFS 软件许可证的信息，请与 Sun 销售代表或授权的服务供应商 (ASP) 联系。

诊断程序

Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 软件中包含 info.sh(1M) 脚本。此诊断脚本对您和 Sun 客户支持人员非常有用。它不仅可以生成服务器配置的诊断报告，而且还可收集日志信息。安装软件之后，您可以访问 info.sh(1M) 联机资料来了解此脚本的详细信息。

安装帮助

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

使用 UNIX 命令

本文档没有介绍基本的 UNIX® 命令和操作过程，如关闭系统、启动系统和配置设备等。

有关此类信息的说明，请参阅下列一个或多个文档：

- *Solaris Handbook for Sun Peripherals* (用于 Sun 外围设备的 Solaris 手册)
- 用于 Sun Solaris OE 的 AnswerBook2™ 联机文档
- 系统附带的其它软件文档

印刷约定

表 P-2 列出了本手册采用的印刷约定。

表 P-2 印刷约定

字样或符号	含义	示例
AaBbCc123	命令、文件和目录的名称；计算机屏幕输出信息。	编辑 .login 文件。 使用 <code>ls -a</code> 列出所有文件。 % You have mail.
AaBbCc123	键入的内容（相对于计算机屏幕输出信息）。	% su Password:
AaBbCc123	书名；新词或术语；需要强调的字词；以及命令行中需用实名或实际值替换的变量。	参阅 <i>用户指南</i> 的第 6 章。 这些称为 <i>class</i> 选项。 您必须为超级用户 (root) 才能执行此操作。 若要删除文件，请键入 <code>rm 文件名</code> 。

表 P-2 印刷约定 (续)

字样或符号	含义	示例
[]	在命令语句中，方括号内的参数表示可选参数。	scmadm [-d sec] [-r n[:n][,n]...] [-z]
{ arg arg }	在命令语句中，大括号和竖线表示必须指定其中一个参数。	sndradm -b {phost shost}
\	命令行末尾的反斜杠 (\) 表示此命令续接下一行。	atm90 /dev/md/rdisk/d5 \ /dev/md/rdisk/d1

Shell 提示符

表 P-3 列出了本手册中使用的 shell 提示符。

表 P-3 Shell 提示符

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

Sun 欢迎您发表意见

Sun 十分注重改进自身文档的质量，欢迎您提出宝贵的意见和建议。您可以使用下面的电子邮件地址将您的意见发送到 Sun:

docfeedback@sun.com

请在电子邮件主题行中注明文档的部件号 (816-7695-10)。

概述

Sun SAM-FS 和 Sun SAM-QFS 环境可以提供具有存储、存档管理和检索性能的可配置文件系统。Sun SAM-FS 和 Sun SAM-QFS 软件通过将联机磁盘高速缓存中的文件复制到存档介质来存档文件。存档介质可以是另一文件系统中的磁盘位片，也可以是自动化库或手动装载存储设备中的可移动磁带或磁光盘卡盒。此外，Sun SAM-FS 和 Sun SAM-QFS 软件产品还可以根据站点指定的使用阈值来保留相应的联机磁盘空间。这些产品可以释放与已存档的文件数据相关联的磁盘空间，并在需要时将文件恢复到联机磁盘。

本章简要介绍了 Sun SAM-FS 和 Sun SAM-QFS 组件的技术性能，其中包括以下主题：

- 第 1 页的“性能”
- 第 3 页的“存储设备”
- 第 3 页的“命令”

性能

Sun SAM-FS 环境包括文件系统和存储及存档管理软件。Sun SAM-QFS 环境包括 Sun QFS 文件系统。所有文件系统均是位于服务器磁盘高速缓存中的高性能 UNIX 文件系统。这些文件系统的主要区别在于 Sun SAM-QFS 文件系统可以提供分布式共享文件系统等更高级的性能。有关文件系统自身的详细信息，请参阅《*Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 文件系统管理员指南*》。

Sun SAM-FS 和 Sun SAM-QFS 环境中的其它组件包括：

- 存档程序，自动将联机磁盘高速缓存中的文件复制到存档介质中。存档介质可由联机磁盘文件或可移动介质卡盒组成。
- 释放程序，通过释放那些符合条件的已存档文件占用的磁盘块，自动将文件系统的联机磁盘高速缓存大小维持在站点指定的使用百分比阈值上。

- 登台程序，将文件数据恢复到磁盘高速缓存中。用户或进程请求已从磁盘高速缓存中释放的文件数据时，登台程序会自动将文件数据复制回联机磁盘高速缓存中。
- 回收程序，清除包含已过期存档副本的存档卷，以使这些卷可重新使用。

以下几节简要介绍了这些功能。这些功能的详细信息将在后面的章节中加以介绍。

存档

默认情况下，存档程序会自动创建 Sun SAM-FS 或 Sun SAM-QFS 文件系统中所有文件的存档副本，然后将这些副本复制到存档介质中。您可以配置存档程序，使其在不同存档介质上创建多达 4 份存档副本。如果文件分成几段，则每段会被视为一个文件单独进行存档。当基于磁盘的文件符合站点定义的一套选择标准后，存档进程将会启动。

有关存档程序的详细信息，请参阅第 73 页的“存档”。有关分段文件的详细信息，请参阅第 263 页的“高级内容”。

释放

释放是指释放由已存档文件的数据占用的主（磁盘）存储空间的过程。它使用两个阈值（由磁盘总空间的百分比表示）来管理联机磁盘高速缓存的可用空间。这两个阈值分别是上限和下限。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存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存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间。另外，文件的第一部分可保留在磁盘上，以加快访问速度和掩饰登台过程的延迟。如果某个文件被分段存档，则可以单独释放该文件的各个部分。有关释放程序的详细信息，请参阅第 137 页的“释放”。

登台

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

Sun SAM-FS 和 Sun SAM-QFS 软件可以自动处理登台请求错误。当收到返回的登台错误时，系统将尝试查找下一个可用的文件存档副本（如果存在副本和可以读取该副本所在介质的设备）。系统可自动处理的登台错误包括：介质错误，介质不可用，自动化库不可用，以及其它错误。有关登台的详细信息，请参阅第 153 页的“登台”。

回收

用户修改文件之后，存档介质上与这些文件的旧版本相关联的存档副本会被视为 *已过期*，并且可以从系统中清除。回收程序可以识别那些其中绝大部分是过期存档副本的存档卷，并将这些卷中的非过期副本移动到其它卷中。

当某个给定的卷中只包含过期副本时，即可执行站点定义的操作。例如，您可以重新标记某个可移动介质卷以立即重新使用该卷，或将其导出至离站存储设备，作为文件更改的历史记录。系统管理员可以使用标准的 UNIX 实用程序从过期存档副本中恢复文件的以前版本。由于回收过程与终端用户的数据文件有关，因此它对终端用户是完全透明的。有关回收的详细信息，请参阅第 165 页的“回收”。

存储设备

Sun SAM-FS 和 Sun SAM-QFS 环境支持多种磁带存储设备和磁光盘设备。有关支持的存储设备列表，请联系 Sun Microsystems 销售代表或授权的服务供应商 (ASP)。

Sun SAM-FS 或 Sun SAM-QFS 环境中管理的各个设备之间的关系在主配置文件 `/etc/opt/SUNWSamfs/mcf` 中进行定义。`mcf` 文件指定了 Sun SAM-FS 和 Sun SAM-QFS 环境中包含的可移动介质设备、自动化库及文件系统。在 `mcf` 文件中，每一个设备均分配有唯一的设备标识。`mcf` 文件中的条目还可定义手动装载的存档设备和自动化库目录文件。

系统将尽可能使用标准的 Sun Solaris 磁盘和磁带设备驱动程序。对于 Sun Solaris OE 不能直接支持的设备，例如某个特定的库和光盘设备，Sun SAM-FS 和 Sun SAM-QFS 软件包中附带了此类特殊设备的驱动程序。

命令

Sun SAM-FS 和 Sun SAM-QFS 环境由文件系统、后台程序、进程、各种命令（用户命令和管理员命令等）和工具组成。《*Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 文件系统管理员指南*》中简要介绍了后台程序的有关信息，各个后台程序将在相应的文档系列中加以说明。本部分向用户介绍各种命令，有关这些命令的详细信息，请参阅软件销售套件中随附的联机资料。

本部分包括下列主题：

- 第 4 页的“用户命令”
- 第 5 页的“一般系统管理员命令”
- 第 6 页的“文件系统命令”

- 第 7 页的“自动化库命令”
- 第 8 页的“存档程序命令”
- 第 8 页的“专用维护命令”
- 第 9 页的“应用软件程序员界面”
- 第 9 页的“可操作实用程序”

用户命令

默认情况下，文件系统操作对终端用户是完全透明的。但是，根据您的站点实际运作情况，您可能希望站点上的用户可以使用一些命令，以便更好地调整某些操作。表 1-1 汇总了这些命令。

表 1-1 用户命令

命令	说明	使用环境
archive(1)	将文件存档并设置文件的存档属性。	Sun SAM-FS, Sun SAM-QFS
release(1)	释放磁盘空间并设置文件的释放属性。	Sun SAM-FS, Sun SAM-QFS
request(1)	创建可移动介质文件	Sun SAM-FS, Sun SAM-QFS
sdu(1)	汇总磁盘使用情况。sdu(1) 命令基于 du(1) 命令的 GNU 版本。	Sun QFS, Sun SAM-FS, Sun SAM-QFS
segment(1)	设置分段文件属性。	Sun SAM-FS, Sun SAM-QFS
setfa(1)	设置文件属性。	Sun QFS, Sun SAM-FS, Sun SAM-QFS
sfind(1)	在目录层次结构中搜索文件。sfind(1) 命令基于 find(1) 命令的 GNU 版本，并且包含用于显示文件系统选项的选项。	Sun QFS, Sun SAM-FS, Sun SAM-QFS
s1s(1)	列出目录内容。s1s(1) 命令基于 1s(1) 命令的 GNU 版本，并且包含用于显示文件系统属性和信息的选项。	Sun QFS, Sun SAM-FS, Sun SAM-QFS

表 1-1 用户命令 (续)

命令	说明	使用环境
squota(1)	报告限额信息。	Sun QFS, Sun SAM-FS, Sun SAM-QFS
ssum(1)	设置文件的校验和属性。	Sun SAM-FS, Sun SAM-QFS
stage(1)	设置文件的登台属性，并且将脱机文件复制到联机磁盘中。	Sun SAM-FS, Sun SAM-QFS

一般系统管理员命令

表 1-2 汇总了用于管理和维护系统的命令。

表 1-2 一般系统管理员命令

命令	说明	使用环境
samcmd(1M)	执行一个 samu(1M) 操作员界面实用程序命令。	Sun QFS, Sun SAM-FS, Sun SAM-QFS
samd(1M)	启动或停止传输设备及可移动介质后台程序。	Sun SAM-FS, Sun SAM-QFS
samset(1M)	更改 Sun SAM-FS 或 Sun SAM-QFS 的设置。	Sun SAM-FS, Sun SAM-QFS
samu(1M)	调用全屏、基于文本的操作员界面。此界面基于 curses(3X) 软件库。samu 实用程序显示设备的状态，并且允许操作员控制自动化库。	Sun QFS, Sun SAM-FS, Sun SAM-QFS

文件系统命令

表 1-3 汇总了文件系统命令。这些命令用于执行文件系统维护操作。

表 1-3 文件系统命令

命令	说明	使用环境
mount(1M)	安装文件系统。此命令的联机资料名为 mount_samfs(1M)。	Sun QFS, Sun SAM-FS, Sun SAM-QFS
qfsdump(1M) qfsrestore(1M)	创建或恢复转储文件。此文件中包含与 Sun QFS 文件系统相关联的文件数据和元数据。	Sun QFS
sambcheck(1M)	列出文件系统的块使用情况。	Sun QFS, Sun SAM-FS, Sun SAM-QFS
samchaid(1M)	更改文件管理集标识属性。用于限额。	Sun QFS, Sun SAM-FS, Sun SAM-QFS
samfsck(1M)	检查并修复文件系统中的元数据冲突，并且收回已分配但尚未使用的磁盘空间。	Sun QFS, Sun SAM-FS, Sun SAM-QFS
samfsconfig(1M)	显示配置信息。	Sun QFS, Sun SAM-FS, Sun SAM-QFS
samfsdump(1M) samfsrestore(1M)	创建或恢复与 Sun SAM-FS 或 Sun SAM-QFS 文件系统相关联的元数据的转储文件。	Sun SAM-FS, Sun SAM-QFS
samfsinfo(1M)	显示 Sun QFS、Sun SAM-FS 或 Sun SAM-QFS 文件系统布局的有关信息。	Sun QFS, Sun SAM-FS, Sun SAM-QFS
samgrowfs(1M)	通过增加磁盘设备来扩展文件系统。	Sun QFS, Sun SAM-FS, Sun SAM-QFS
sammkfs(1M)	在磁盘设备中初始化新的文件系统。	Sun QFS, Sun SAM-FS, Sun SAM-QFS
samncheck(1M)	返回完整的目录路径名（包括安装点和索引节点编号）。	Sun QFS, Sun SAM-FS, Sun SAM-QFS
samquota(1M)	报告、设置或重设限额信息。	Sun QFS, Sun SAM-FS, Sun SAM-QFS

表 1-3 文件系统命令 (续)

命令	说明	使用环境
samquotastat(1M)	报告已激活和未激活文件系统的限额。	Sun QFS, Sun SAM-FS, Sun SAM-QFS
samsharefs(1M)	管理 Sun QFS 共享文件系统配置信息。	Sun QFS, Sun SAM-QFS
samtrace(1M)	转储跟踪缓冲器。	Sun QFS, Sun SAM-FS, Sun SAM-QFS
samunhold(1M)	释放 SANergy 文件保持。	Sun QFS, Sun SAM-FS, Sun SAM-QFS
trace_rotate.sh(1M)	循环更新跟踪文件。	Sun QFS, Sun SAM-FS, Sun SAM-QFS

自动化库命令

表 1-4 总结了用于在 Sun SAM-FS 和 Sun SAM-QFS 环境中配置、初始化和维护自动化库及设备的自动化库命令。

表 1-4 自动化库命令

命令	说明
auditslot(1M)	核查指定自动化库中的单个介质卡盒端口。
build_cat(1M)	建立自动化库的介质目录文件。此外，它还可用于生成目录文件。
chmed(1M)	设置或清除特定卡盒上的库目录标记和数值。
cleandrive(1M)	请求载入附带清洁磁带的磁带驱动器。
dump_cat(1M)	以各种 ASCII 格式显示二进制目录文件的内容。
import(1M)	通过将卡盒放入邮箱中以从库中导入或导出卡盒。对于通过网络连接的库，此命令用于更新库目录，而不是物理移动卡盒。
samexport(1M)	
move(1M)	将卡盒从一个端口移动到另一个端口。
odlabel(1M)	标记光盘，以便用于 Sun SAM-FS 和 Sun SAM-QFS 系统。
samdev(1M)	添加 /dev/samst 逻辑设备条目。用于传递自动化库、光盘和磁带驱动器信息。
tplabel(1M)	标记磁带，以便用于 Sun SAM-FS 和 Sun SAM-QFS 系统。

存档程序命令

表 1-5 汇总了用于在 Sun SAM-FS 和 Sun SAM-QFS 环境中控制存档程序操作的命令。

表 1-5 存档程序命令

命令	说明
archiver(1M)	评估存档程序命令文件的语法完整性和语义准确性。
showqueue(1M)	显示存档程序队列文件的内容。
reserve(1M)	保留卷和取消保留卷。
unreserve(1M)	

专用维护命令

表 1-6 汇总了各种维护命令。

表 1-6 专用维护命令

命令	说明
archive_audit(1M)	生成每个卡盒中所有存档文件的报告。
dev_down.sh(4)	当设备被标记为 down (禁用) 或 off (关闭) 时, 向超级用户 (root) 发送电子邮件。
dmpshm(1M)	转储共享的内存段。
exarchive(1M)	管理 (交换) 存档副本。
itemize(1M)	编制光盘目录。
load_notify.sh(1M)	当 Sun SAM-FS 或 Sun SAM-QFS 软件请求位于库外的卡盒时, 通知操作员。
samload(1M)	装载或卸载指定设备的卡盒。
unload(1M)	
rearch(1M)	标记或取消标记要重新存档的存档条目。
unrearch(1M)	
sam-recycler(1M)	收回过期存档副本在存档介质中占用的空间。
sam-releaser(1M)	从联机磁盘高速缓存文件系统中释放磁盘空间。
samdev(1M)	在 /dev/samst 目录中创建指向 Sun SAM-FS 或 Sun SAM-QFS 文件系统实际所用的设备的符号链接。此命令的功能与 UNIX makedev(1M) 命令相似。

表 1-6 专用维护命令 (续)

命令	说明
samset(1M)	更改或显示 Sun SAM-FS 或 Sun SAM-QFS 操作中所用的变量。
set_admin.sh(1M)	添加或删除管理员群组执行管理员命令的权限。
set_state(1M)	设置 Sun SAM-FS 或 Sun SAM-QFS 设备的状态。
star(1M)	创建磁带存档, 并且添加或抽取文件。它是 tar(1) 命令的 GNU 版本, 但其性能已被扩充以用于 Sun SAM-FS 或 Sun SAM-QFS 文件系统。当需要从存档磁带读取数据时, 可以使用此命令进行故障恢复。
unarchive(1M)	删除一个或多个文件的存档条目。
undamage(1M)	将一个或多个文件或目录的存档条目标记为“未损坏”。

应用程序程序员界面

应用程序程序员界面 (API) 允许从用户应用程序中发出文件系统请求。用户可以从本地或远程向此文件系统所运行的计算机发送请求。API 由 `libsam` 和 `libsamrpc` 库组成。这两个库包括一些库例行程序, 它们用于获取文件状态; 设置文件的存档、释放、登台属性; 以及管理自动化库的库目录。`sam-rpcd` 服务器进程用于处理远程请求。若要自动启动 `sam-rpcd` 服务器进程, 用户必须在 `defaults.conf` 文件中设置 `samrpc=on`。

有关 API 的详细信息, 请参阅 `intro_libsam(3)` 联机资料。此联机资料简要介绍了如何在 `libsam` 和 `libsamrpc` 中运行库例行程序。

可操作实用程序

在 Sun SAM-FS 和 Sun SAM-QFS 环境中, `samu(1M)` 操作员实用程序和 GUI 工具可以帮助您执行基本的操作。表 1-7 汇总了这些 GUI 工具。

表 1-7 可操作工具

GUI 工具	说明
devicetool(1M)	显示设备的状态, 并且允许您更改其状态。
libmgr(1M)	显示有助于在 Sun SAM-FS 或 Sun SAM-QFS 环境中管理自动化库、设备和安装请求的信息。此工具基于 Java 技术。
previewtool(1M)	显示并管理等待执行的卡盒安装请求。

表 1-7 可操作工具 (续)

GUI 工具	说明
robottool(1M)	显示自动化库的状态；安装和卸载文件系统；导入和导出卡盒；以及显示库目录。
samtool(1M)	一种应用程序启动器，提供用于访问 devicetool(1M)、robottool(1M) 和 previewtool(1M) 的起始点。
samu(1M)	提供用于访问 samu(1M) 操作员实用程序的起始点。

基本操作

自动化库是一个自动控制的设备，它可在无操作人员参与的情况下，自动载入和卸载可移动卡盒。卡盒可以导入自动化库或从中导出。系统能够自动载入或卸载卡盒。存档和登台进程使用站点定义的方案分配要使用的驱动器数量。自动化库也可称为介质更换器、自动光盘存储器、传输器、资料库或介质库。

以下各部分介绍了在 Sun SAM-FS 和 Sun SAM-QFS 环境中使用自动化库的各个方面。《Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 安装与配置指南》中提供了初始配置说明，本章将介绍自动化库和手动载入设备的操作说明。此外，本章还介绍在被请求的卷不在库中时，用于提示操作员载入该卷的通知工具。

注 – Sun SAM-FS 和 Sun SAM-QFS 软件可以与多家制造商生产的自动化库兼容。有关库模型编号、固件级别和其它兼容性信息，请参阅此版本附带的 README 文件。

某些自动化库的功能所产生的操作可能不同于本章介绍的操作。要确定您的自动化库在 Sun SAM-FS 或 Sun SAM-QFS 环境中使用时是否还要遵守供应商提供的其它专用操作说明，请参阅第 41 页的“特定自动化库的基本操作”。

本章包括下列主题：

- 第 12 页的“自动化库硬件和软件”
- 第 13 页的“mcf 文件”
- 第 14 页的“defaults.conf 文件”
- 第 17 页的“载入通知”
- 第 18 页的“目录操作”
- 第 21 页的“约定”
- 第 22 页的“自动化库操作”
- 第 39 页的“手动载入驱动器操作”

自动化库硬件和软件

自动化库通过以下方式之一连接到 Sun SAM-FS 或 Sun SAM-QFS 主机系统：

- 直接连接。自动化库使用小型计算机系统接口 (SCSI) 直接连接到主机系统。这种连接可以是直接连接，也可以是光纤信道连接。例如，Sun StorEdge 库使用直接连接方式。Sun SAM-FS 和 Sun SAM-QFS 系统使用自动化库的 SCSI 标准来直接控制它们。
- 网络连接。使用这种连接时，Sun SAM-FS 或 Sun SAM-QFS 软件被配置成库主机系统的客户机。网络连接自动化库包括 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。在这些情况下，Sun SAM-FS 或 Sun SAM-QFS 软件使用专为自动化库设计的后台程序与供应商软件通信。

表 2-1 列出了不同自动化库专用的后台程序。

表 2-1 自动化库后台程序

后台程序	说明
sam-robotd	监视传输器控制后台程序的执行情况。sam-robotd 后台程序由 sam-initd 后台程序自动启动。
sam-genericd	控制直接连接的库和介质更换器。
sam-stkd	通过 ACSAPI 接口控制 StorageTek 介质更换器。
sam-ibm3494d	通过 lmcpd 接口控制 IBM 3494 磁带库。
sam-sonyd	通过 DZC-8000S 接口控制 Sony 网络连接自动化库。

配置自动化库

您可以通过编辑 `/etc/opt/SUNWsamfs/mcf` 中的主配置文件来配置自动化库。mcf 文件中的 `Additional Parameters` (其它参数) 字段中含有库目录文件的路径名。该库目录文件中包含自动化库存储端口中每个卡盒内容的有关信息。默认情况下，库目录文件写入至 `/var/opt/SUNWsamfs/family_set_name`。

某些自动化库，例如 IBM 3494 库、StorageTek 网络连接库、ADIC/Grau 库 和 Sony 网络连接库，需要附加的配置信息。此类信息包含在一个单独的文件中，而该文件的名称也将指定在 mcf 文件中。

除非您安装了供应商提供的全部软件（如果有）并确定它们可以正常运行，否则，请不要将 Sun SAM-FS 或 Sun SAM-QFS 软件配置为用于自动化库。参阅供应商提供的文档，确保熟悉供应商的软件。

参数文件

如果自动化库使用的供应商软件包按前面所述运行某个接口，则您必须设置附加的参数文件来定义库及其驱动器的系统特征。参数文件因供应商提供的自动化库而异。有关各家供应商专用的参数文件，请参阅 `stk(7)`、`ibm3494(7)`、`ibm3584(7)`、`grauaci(7)`、`fujitsulmf(7)` 和 `sony(7)` 联机资料。

mcf 文件

《*Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 安装与配置指南*》和《*Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 文件系统管理员指南*》中介绍了 mcf 文件的格式化方法。驱动器通过使用系列集名与自动化库关联。每个驱动器均分配有唯一的设备序数和其它参数。

Equipment Identifier（设备标识）字段因您的库是直接连接还是网络连接而异，如下所述：

- 对于直接连接的库，设备标识字段是自动化库自身的 `/dev/samst` 条目。有关磁盘自动化库的配置示例，请参阅《*Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 安装与配置指南*》。
- 对于网络连接的 Sony、StorageTek、ADIC/Grau 或 Fujitsu 等自动化库，设备标识字段必须包含参数文件的完整路径名。有关参数文件的详细信息，请参阅第 13 页的“参数文件”。

对于既直接连接又网络连接的库，可在 Additional Parameters（其它参数）字段中填写库目录的完整路径名。

代码示例 2-1 所示的 mcf 文件中定义了 StorageTek 直接连接自动化库。在该示例中，两个驱动器与自动化库相关联。

代码示例 2-1 mcf 文件示例

```
# Equipment      Eq  Eq Family Dev Additional
# Identifier      Ord Ty Set   St  Parameters

# Define a file system

samfs1           10  ms samfs1
/dev/dsk/c1t1d0s0 11  md samfs1 on
/dev/dsk/c2t1d0s0 12  md samfs1 on

# Define a library

/dev/samst/c0t3u0 50  s9 stk   on
/dev/rmt/3cbn    51  sg stk   on
/dev/rmt/0cbn    52  sg stk   on
```

您可能希望定期更改 mcf 文件以添加新库、更改文件系统磁盘或执行其它任务。如果更改了 mcf 文件，则必须输入命令以重新初始化 mcf 文件，从而使 Sun SAM-FS 或 Sun SAM-QFS 软件能够应用这些更改。《Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 文件系统管理员指南》中介绍了初始化 mcf 文件的过程。

defaults.conf 文件

您可以在 /etc/opt/SUNWsamfs/defaults.conf 文件中设置数个指令以控制自动化库的操作。以下几节介绍了其中一些指令。有关这些指令的详细信息，请参阅 defaults.conf(4) 联机资料。

如果您在 defaults.conf 文件中更改了信息，（例如，为适应站点自动化库信息的更改），则必须输入命令以重新初始化该文件。《Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 文件系统管理员指南》中介绍了初始化 defaults.conf 文件的过程。

以下 defaults.conf 示例文件中的行显示了影响自动化库配置的数个参数：

```
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 中显示了另一个示例文件。有关 defaults.conf 文件的详细信息，请参阅 defaults.conf(4) 联机资料。

您可以使用 defaults.conf(4) 文件配置您的环境中的许多功能。以下几节介绍了其中一些功能。

条码

如果您配有一个使用条码读取器的磁带库，则可以配置系统，以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。您可以通过在 `defaults.conf` 文件中设置 `labels` 指令来完成此任务（如表 2-2 所示）。

表 2-2 `defaults.conf` 文件中的 `labels` 指令

指令	操作
<code>labels = barcodes</code>	将条码的前六个字符作为标签。此设置可以使存档程序在已选定磁带的情况下自动标记新介质。默认设置。
<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。此设置可以防止存档程序自动标记新介质。

如果设置了 `labels = barcodes` 或 `labels = barcodes_low` 指令，则对于已启用写操作、未标记且具有可读条码的磁带，（假设该磁带由于需要进行写入操作而安装），Sun SAM-FS 或 Sun SAM-QFS 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

您可以使用 `dev_delay` 和 `dev_unload` 指令分别设置设备的载入时间、卸载时间和卸载等待时间。这两个指令允许您将这段时间设置为符合站点要求的间隔时间。您可以在 `defaults.conf` 文件中设置这两个指令。

`dev_delay` 指令的格式如下：

```
dev_delay = seconds
```

其中：

`dev` `mcf(4)` 联机资料中指定的设备类型。

`seconds` 一个整数，用于指定从载入卡盒到卸载卡盒至少允许的时间。默认值为 30 秒。

`dev_unload` 参数的格式如下:

```
dev_unload = seconds
```

其中:

`dev` `mcf(4)` 联机资料中指定的设备类型。

`seconds` 一个整数, 用于指定输入 `unload` 命令之后需要等待的时间。此处指定的时间供自动化库在卸载卡盒之前弹出卡盒、打开挡门和执行其它操作。默认值为 0 秒。

例如:

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

有关这些指令的详细信息, 请参阅 `defaults.conf(4)` 联机资料。

共享驱动器

一般而言, Sun SAM-FS 或 Sun SAM-QFS 进程单独控制主机系统的 `mcf` 文件中所述的库驱动器。在许多情况下, 驱动器在单个 `mcf` 文件中进行定义, 以供单个 Sun SAM-FS 和 Sun SAM-QFS 进程使用。在进程不使用驱动器时, 驱动器将处于空闲状态。

共享驱动器功能允许两个以上的 `mcf` 文件定义同一个驱动器, 从而使多个 Sun SAM-FS 或 Sun SAM-QFS 进程可以使用该驱动器。虽然共享驱动器功能允许每一个 Sun SAM-FS 或 Sun SAM-QFS 进程共享驱动器, 但是它并不允许进程共享介质。每一个 Sun SAM-FS 或 Sun SAM-QFS 进程必须仍然保留其自身的 VSN 集。多个进程不能共享单个介质。

该功能有时非常有用。例如, 在某个库连接到 Sun SAM-FS 或 Sun SAM-QFS 环境中的多个主机系统时。共享驱动器功能可使库中的驱动器保持繁忙状态。Sun SAM-FS 或 Sun SAM-QFS 进程之间协调使用驱动器。

某些库经过配置, 可以将其中的一个或全部介质驱动器共享给多个主机系统上的多个 Sun SAM-FS 或 Sun SAM-QFS 进程。若要使用共享驱动器, 库必须采用网络连接方式。以下类型的库支持共享驱动器功能:

- 使用 ACSLS 接口的 StorageTek 库
- 使用 `lmcpd` 接口的 IBM 3494 库
- 使用 DZC-8000S 接口的 Sony 库

上述所有库均要求创建参数文件。若要设置一个或多个共享驱动器，请确保在要共享的每个驱动器的参数文件中指定 `shared`（共享）关键字。共享关键字的放置位置因制造商生产的库而有所不同。下面的参数示例文件显示了如何为 StorageTek 网络连接库指定共享关键字：

```
#
# This is file: /etc/opt/SUNWsamfs/stk50
#
hostname = acsls_server_name
portnum = 50004
access = some_user # No white space allowed in the user_id field
/dev/rmt/0cbn = (acs=0, lsm=1, panel=0, drive=1) #a comment
/dev/rmt/1cbn = (acs=0, lsm=1, panel=0, drive=2) shared
capacity = (0=215040, 1=819200, 5=10485760)
```

如上例所示，您可以在此参数文件中加入注释，但前提是它们必须以井字符 (#) 开头。

共享驱动器的配置过程因库的不同而有所差异。有关如何为网络连接库配置共享驱动器的详细信息，请参阅以下联机资料：[ibm3494\(7\)](#)、[sony\(7\)](#) 或 [stk\(7\)](#)。

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。若要更改此计时，请在 `defaults.conf` 文件中指定 `shared_unload` 指令。有关此指令的详细信息，请参阅 [defaults.conf\(4\)](#) 联机资料。

载入通知

为满足存档和登台的需要，Sun SAM-FS 和 Sun SAM-QFS 软件经常会请求载入卡盒。如果被请求的卡盒位于库中，则系统会自动处理该请求。如果被请求的卡盒不在库中，则需要操作员载入卡盒。

需要从库外获取卡盒时，`load_notify.sh(1M)` 脚本（如果已启用）将向有关人员发送电子邮件。此脚本位于下面所示的位置：

```
/opt/SUNWsamfs/examples/load_notify.sh
```

使用该脚本之前，您必须将其复制到以下位置：

```
/opt/SUNWsamfs/sbin/load_notify.sh
```

要使用脚本通知功能，`defaults.conf` 文件中的某些指令必须处于生效状态。这些指令包括：

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

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

默认情况下，脚本将向超级用户（`root`）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、拨叫录呼机或提供其它通知方式。

目录操作

库目录是一个信息中央仓库，它包含 Sun SAM-FS 或 Sun SAM-QFS 环境在自动化库中查找卡盒时需要的全部信息。库目录文件是一个二进制 UFS 驻留文件，其中包括自动化库中每一个端口的有关信息。该文件中的信息包括：与端口中的卡盒相关联的一个或多个卷序列名（VSN）；该卡盒的容量和剩余空间；以及指示卡盒只读、写保护和其它状态信息的标记。

Sun SAM-FS 和 Sun 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，如下所述：

- 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端口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端口。需要某个卡盒时，系统将首先查询库目录以确定哪一个端口包含该 VSN，然后发出命令以将该端口中的卡盒载入驱动器。
- 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端口不是直接的对应关系。它是自动化库中已知 VSN 的列表。请求某个卡盒时，系统将向供应商的软件发送请求以将该 VSN 载入驱动器。供应商的软件可以确定包含该 VSN 的存储端口。

您需要执行一些操作以便在 Sun SAM-FS 或 Sun SAM-QFS 环境中初始化和维护自动化库。以下几节介绍了如何建立库目录、检查库目录以及在自动化库中添加和移出卡盒。

▼ 建立库目录

当 `sam-catserverd` 后台程序启动时，它将检查每一个自动化库的库目录文件是否存在。如果某个自动化库的目录文件不存在，则必须按以下方式为其创建目录文件：

- 对于直接连接的 IBM 3494 自动化库，如果它包含编有条码的介质，Sun SAM-FS 或 Sun SAM-QFS 将直接建立目录文件。
- 对于直接连接的 IBM 3494 自动化库，如果它包含未编条码的介质，Sun SAM-FS 或 Sun SAM-QFS 将先核查该库，然后建立目录文件。核查过程将显示该自动化库中所有卡盒的位置。

- 对于网络连接的 StorageTek、ADIC/Grau、Fujitsu 和 Sony 自动化库，您必须通过输入一系列 `import(1M)` 命令或输入一个 `build_cat(1M)` 命令来建立目录。如果自动化库中的卡盒不太多，则输入一系列 `import(1M)` 命令是可行的。如果自动化库中有许多卡盒，则您可以提供一个 VSN 列表作为 `build_cat(1M)` 命令的变量。`build_cat(1M)` 命令将使用此 VSN 列表生成初始目录。

有关 `import(1M)` 命令的详细信息，请参阅 `import(1M)` 联机资料。有关 `build_cat(1M)` 命令的详细信息，请参阅 `build_cat(1M)` 联机资料。

▼ 检查库目录

`dump_cat(1M)` 命令以文本格式书写自动化库的目录文件。该命令的格式如下：

```
dump_cat -o -V catalog_name
```

其中：

- o 列出库目录中不再存在的卡盒。即未设置“使用中”标记但存在目录条目的卡盒。
- v 显示标记和标签时间，作为注释。此选项将打印每一个 VSN 的行，这些行显示了标签时间、上次修改时间和载入时间。它显示标记的格式与 `samu(1M)` 实用程序的 `v` 显示屏幕相同。

下面是 `dump_cat` 命令的输出清单示例：

```
# dump_cat stk
# audit_time Wed Dec 31 18:00:00 1969
# version 350 count 100 mediatype sg
# Index VSN Barcode Type PTOC Access Capacity Space Status Sector
Label time Eq Slot Part
#
    0 004974 004974 sg 0xaaa6 27 20971520 15379712 0x7a010000
262144 0x386ccfad 300 0 0
    1 004971 004971 sg 0xe003 45 20971520 13631104 0x6a000000
262144 0x3793b707 300 1 0
    3 004973 004973 sg 0xb8c 7 20971520 20593152 0x6a000000
262144 0x38c94d17 300 3 0
    4 004975 004975 sg 0x660 5 20971520 20762624 0x6a000000
262144 0x3693a80d 300 4 0
    5 004977 004977 sg 0x5421 27 20971520 17243264 0x6a000000
262144 0x384eb998 300 5 0
    6 004970 004970 sg 0x1468e 1 20971520 20971520 0x6a000000
262144 0x38c94da1 300 6 0
```

▼ 导入和导出卡盒

在自动化库中物理添加（导入）或取出（导出）卡盒可以执行数项功能。例如，您可以更换卡盒、将故障恢复磁带移至离站存储设备等。导入和导出卡盒还会更新库目录。Sun SAM-FS 或 Sun SAM-QFS 系统使用 `import(1M)` 和 `samexport(1M)` 命令来完成这些任务。另外，您还可使用 `libmgr(1M)` 或 `robottool(1M)` 命令来执行这些任务。

但是请注意，自动化库处理卡盒导入和导出操作的方式因系统特性和供应商提供的软件而有所不同。例如，对于 ACL 4/52 库，您需要先输入移动命令以将卡盒移至导入/导出设备中，然后才能从自动化库中导出卡盒。ADIC/Grau、StorageTek 和 Fujitsu 自动化库等设备使用自身的实用程序来导入和导出卡盒。因此，`import(1M)` 和 `samexport(1M)` 命令只是更新 Sun SAM-FS 和 Sun SAM-QFS 系统使用的库目录条目。

有关导入和导出卡盒的详细信息，请参阅第 35 页的“导入和导出卡盒（配有邮箱的系统）”和第 37 页的“导入和导出卡盒（未配邮箱的系统）”。

跟踪导出的介质 — Historian

Sun SAM-FS 和 Sun SAM-QFS 的 **Historians**（历史记录）可以跟踪从自动化库或手动载入设备中导出的卡盒。**Historian** 类似于一个虚拟库，但它没有定义的硬件设备。与自动化库一样，它也在 `mcf` 文件中进行配置；具有用于记录与其关联的所有卡盒的目录；可以导入和导出卡盒；以及在 `robottool(1M)` 中显示为另一自动化库。

您可以在 `mcf` 文件中使用 `hy` 设备类型对 **Historian** 进行配置。如果未在 `mcf` 文件中配置 **Historian**，则可以使用以下命令行进行创建：

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

在上述条目中，`n+1` 表示 `mcf` 文件中的最后一个设备序数 + 1。若要使用不同的设备序数或其它目录路径名，只需在 `mcf` 文件中定义 **Historian**。

Historian 第一次启动时，**Historian** 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。您的站点可能需要跟踪已从自动化库中导出的现有 Sun SAM-FS 和 Sun SAM-QFS 卡盒。在此情况下，您需要按 `build_cat(1M)` 联机资料中的说明在现有的卡盒中建立 **Historian** 目录。

`defaults.conf` 文件中的以下两个配置指令将影响 **Historian** 的操作：

- 如果设置了 `exported_media = unavailable` 指令，则从自动化库中导出的任何卡盒均会被标记为不能用于 **Historian**。请求此类卡盒将产生 EIO 错误。
- 如果设置了 `attended = no` 指令，它将通知 **Historian** 无任何操作员处理载入请求。如果已通知 **Historian** 载入卡盒，但实际并未载入，则会产生 EIO 错误。

有关配置的详细信息，请参阅 `historian(7)` 和 `defaults.conf(4)` 联机资料。

约定

执行本章所述的基本操作过程通常需要使用 `samu(1M)` 操作员实用程序和下列命令：

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

但是，在许多情况下，您可以选用多种方法来执行所述的任务。除了使用 `samu(1M)` 和上述命令之外，您还可以从图形用户界面 (GUI, graphical user interface) 工具（包括 `devicetool(1M)`、`libmgr(1M)` 和 `robottool(1M)`）中执行这些任务。有关 GUI 工具的详细信息，请参阅第 181 页的“图形用户界面 (GUI) 工具”。

命令变量

许多命令接受一组通用的变量。表 2-3 中列出了这些变量。

表 2-3 命令变量

变量	含义
<i>eq</i>	所访问的设备在 <code>mcf</code> 文件中定义的序数。可以识别的设备包括自动化库、驱动器或文件系统。
<i>slot</i>	自动化库中存储端口的编号，与库目录中标识的编号相同。
<i>partition</i>	磁光盘的一面。 <code>partition</code> 必须为 1 或 2。
<i>media_type</i>	介质类型。有关有效介质类型的列表，请参阅 <code>mcf(4)</code> 联机资料。
<i>vsn</i>	分配给卷的卷序列名。

某些命令可以接受多种变量组合，视您的环境而定。例如，在 `samu(1M)` 操作员实用程序中，`load` 命令具有以下两种格式：

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

注意以下细节：

- 第一种格式使用冒号 (:) 隔开 `eq` 和 `slot`。
- 第二种格式使用句号 (.) 隔开 `media_type` 和 `vsn`。

术语

本章所用的某些术语可能是您第一次碰到。表 2-4 列出了一些最常用的术语及其含义。

表 2-4 术语

术语	含义
自动化库 (<i>automated library</i>)	用于存储磁带或光盘卡盒的自动设备。
卡盒 (<i>cartridge</i>)	磁带或磁光盘卡盒。磁光盘卡盒可以包含一个或多个卷或分区。
分区 (<i>partition</i>)	整个磁带或磁光盘的一面。一个分区只能包含一个卷。
卷 (<i>volume</i>)	卡盒中用于存储数据的命名区域。一个卡盒中可以有一个或多个卷。双面卡盒有两个卷，每一面为一个卷。系统采用卷序列名 (VSN) 来标识卷。

自动化库操作

有几项基本操作在所有自动化库中基本上是相同的。本节介绍了下列基本操作：

- 第 23 页的“自动清洁和自动载入”
- 第 24 页的“启动可移动介质操作”
- 第 23 页的“停止可移动介质操作”
- 第 24 页的“打开自动化库”
- 第 25 页的“关闭自动化库”
- 第 25 页的“将卡盒载入自动化库”

- 第 26 页的“从驱动器中卸载卡盒”
- 第 26 页的“标记卡盒”
- 第 28 页的“核查卷”
- 第 29 页的“核查自动化库（仅限于直接连接）”
- 第 30 页的“使用清洁卡盒”
- 第 32 页的“清洁磁带驱动器”
- 第 33 页的“清除介质错误”
- 第 34 页的“从驱动器中取出卡住的卡盒”

▼ 自动清洁和自动载入

如果您的库采用直接连接方式，并且具有自动清洁和自动载入功能，则应禁用这两项功能。

如果您的库采用网络连接方式，则应根据库制造商的建议启用自动清洁和自动载入功能。

自动清洁和自动载入的具体操作因制造商而异。如果您遇到这方面的问题，请参阅第 41 页的“特定自动化库的基本操作”以确定是否需要为您的设备执行特殊步骤。

有关如何启用或禁用自动清洁和自动载入功能的信息，请参阅库制造商提供的文档。

▼ 停止可移动介质操作

有时，您可能需要停止可移动介质的操作，而仍保持 Sun SAM-FS 或 Sun SAM-QFS 系统的安装状态。例如，在您希望手动操作库中的卡盒时，可能需要执行此操作。恢复介质操作之后，未完成的登台操作将被重新输入，而存档操作也将随之恢复。

若要停止可移动介质操作，请输入下面的命令：

```
samcmd idle eq  
samd stop
```

其中：

eq 所访问的设备在 *mcf* 文件中定义的序数。要将驱动器置于空闲状态，请对 *mcf* 文件中配置的每一个 *eq* 运行 `samcmd idle eq` 命令：

此外，您还可以使用 `samu(1M)` 操作员实用程序或者使用 `robottool(1M)` 或 `libmgr(1M)` GUI 工具将驱动器置于空闲状态。有关 `samcmd(1M)` 命令的详细信息，请参阅 `samcmd(1M)` 联机资料。有关 `samd(1M)` 命令的详细信息，请参阅 `samd(1M)` 联机资料。

注 – 输入 `samd stop` 命令之前，请确保已将 Sun SAM-FS 或 Sun SAM-QFS 环境中的驱动器置于空闲状态。其目的是使存档程序、登台程序和其它进程结束当前的任务。如果未输入 `samd stop` 命令，则在恢复存档、登台或其它活动时可能导致意外的结果。

▼ 启动可移动介质操作

安装 Sun SAM-FS 或 Sun SAM-QFS 文件系统时，通常会启动可移动介质操作。要手动启动可移动介质操作，而不安装文件系统，请输入下面的 `samd(1M)` 命令：

```
# samd start
```

如果在输入上述命令时可移动介质操作已在运行，则会生成下面的消息：

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

有关 `samd(1M)` 命令的详细信息，请参阅 `samd(1M)` 联机资料。

▼ 打开自动化库

自动化库处于 on（打开）状态时，它由 Sun SAM-FS 或 Sun SAM-QFS 系统控制，并可继续处理一般操作。当您打开库时，Sun SAM-FS 或 Sun SAM-QFS 软件将执行以下操作：

- 根据设备的内部状态来查询设备。它可以发现磁带的位置、是否使用条码等。
- 更新目录和其它内部结构。

`samu(1M)` 实用程序的 `s` 显示屏幕可以显示自动化库的状态。若要打开自动化库，请使用 `samu(1M)` 实用程序的 `:on` 命令。该命令的格式如下：

```
:on eq
```

其中：

`eq` 所访问的自动化库在 `mcf` 文件中定义的设备序数。

此外，您还可以使用 GUI 工具 `robottool(1M)` 和 `libmgr(1M)` 来执行此任务。有关这些工具的详细信息，请参阅第 181 页的“图形用户界面 (GUI) 工具”或 `robottool(1M)` 和 `libmgr(1M)` 联机资料。

▼ 关闭自动化库

将库置入 `off` (关闭) 状态可以停止 I/O 操作并使自动化库脱离 Sun SAM-FS 或 Sun SAM-QFS 的控制。此时, 卡盒将不能自动移动。请注意, 自动化库中的驱动器仍处于 `on` (打开) 状态。若要执行以下任务, 您可能需要关闭自动化库:

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

若要关闭自动化库, 请使用 `samu(1M)` 实用程序的 `:off` 命令。该命令的格式如下:

```
:off eq
```

其中:

`eq` 所访问的设备在 `mcf` 文件中定义的序号。

此外, 您还可以使用 GUI 工具 `robottool(1M)` 和 `libmgr(1M)` 来执行此任务。有关这些工具的详细信息, 请参阅第 181 页的“图形用户界面 (GUI) 工具”或 `robottool(1M)` 和 `libmgr(1M)` 联机资料。

▼ 将卡盒载入自动化库

当请求某个 `VSN` 以进行存档或登台操作时, 系统会自动将卡盒载入驱动器。载入是指将卡盒从存储端口移入驱动器并使之处于就绪状态的过程。

若要手动载入卡盒, 请使用 `samu(1M)` 实用程序的 `:load` 命令。即使驱动器处于 `unavail` (不可用) 状态时也可使用此命令。该命令具有以下两种格式:

```
:load eq:slot[:partition]
:load media_type.vsn
```

其中:

`eq` 所访问的驱动器在 `mcf` 文件中定义的设备序号。

`slot` 自动化库中存储端口的编号, 与库目录中标识的编号相同。

`media_type` 介质类型。有关有效介质类型的列表, 请参阅 `mcf(4)` 联机资料。

`partition` 磁光盘的一面。`partition` 必须为 1 或 2。此变量不适用于磁带卡盒。

`vsu` 分配给卷的卷序列名。

此外，您还可以使用 GUI 工具 `robottool(1M)` 和 `libmgr(1M)` 来执行此任务。有关这些工具的详细信息，请参阅第 181 页的“图形用户界面 (GUI) 工具”或 `robottool(1M)` 和 `libmgr(1M)` 联机资料。

当您手动载入卡盒时，它通常会载入库中的下一个可用驱动器。如果您不想让某个驱动器用于此目的，请使用 `samu(1M)` 实用程序的 `:unavail` 命令，或使用 `devicetool(1M)` 更改该设备的状态。例如，在故障恢复操作或对磁带进行分析期间，您可能会执行此操作。

▼ 从驱动器中卸载卡盒

当不再需要使用卷时，系统会自动卸载卡盒。您也可手动卸载驱动器。卸载是指从驱动器中卸载卡盒。

若要手动卸载驱动器，请使用 `samu(1M)` 实用程序的 `:unload` 命令。即使驱动器处于 `unavail` (不可用) 状态时也可使用此命令。该命令的格式如下：

```
:unload eq
```

其中：

`eq` 所访问的驱动器在 `mcf` 文件中定义的设备序号。

此外，您还可以使用 GUI 工具 `robottool(1M)` 和 `libmgr(1M)` 来执行此任务。有关这些工具的详细信息，请参阅第 181 页的“图形用户界面 (GUI) 工具”或 `robottool(1M)` 和 `libmgr(1M)` 联机资料。

标记卡盒

如果您使用的自动化库没有安装条码读取器，则在使用 Sun SAM-FS 或 Sun SAM-QFS 系统之前，必须标记所有卡盒。

如果您的自动化库使用条码，则 `labels = barcodes` 为默认设置，并且其结果是 VSN 的标签为条码的前六个字符。

如果您的自动化库使用条码，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，加入下面的行：

```
labels = barcodes_low
```

当载入编有条码的卡盒以便进行写入操作时，系统会先将标签写至此卡盒，然后开始写入操作。卡盒必须已启用写操作、未标记且具有可读条码。

标记卡盒的过程取决于您是标记磁带还是标记光学卡盒。以下两节介绍了这些过程。



注意 – 标记和重新标记卡盒会造成 Sun SAM-FS 和 Sun SAM-QFS 软件无法访问卡盒中的数据。

▼ 标记或重新标记磁带

下面的 `tplabel(1M)` 命令行格式显示了在标记或重新标记磁带时最常用的选项：

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

其中：

vsn 卷序列名。如果您重新标记磁带，新 VSN 名可以与旧 VSN 名相同。

eq 所访问的自动化库或手动载入驱动器在 `mcf` 文件中定义的设备序数。

slot 自动化库中存储端口的编号，与库目录中标识的编号相同。该变量不适用于手动载入的驱动器。

若要标记新磁带，请使用带以下选项的 `tplabel` 命令：

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

若要重新标记现有的磁带，请使用带以下选项的 `tplabel` 命令：

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

运行上述用以标记或重新标记磁带的命令后，系统会载入磁带并确定其位置，然后写入磁带标签。有关 `tplabel(1M)` 命令的详细信息，请参阅 `tplabel(1M)` 联机资料。

此外，您还可以使用 GUI 工具 `robottool(1M)` 和 `libmgr(1M)` 来执行此任务。有关这些工具的详细信息，请参阅第 181 页的“图形用户界面 (GUI) 工具”或 `robottool(1M)` 和 `libmgr(1M)` 联机资料。

▼ 标记或重新标记光盘

下面的 `odlabel(1M)` 命令行格式显示了在标记或重新标记光盘时最常用的选项：

```
odlabel [ -new | -old vsn ] -vsn vsn eq:slot:partition
```

其中：

- vsn* 卷序列名。如果您重新标记磁带，新 VSN 名可以与旧 VSN 名相同。
- eq* 所访问的自动化库或手动载入驱动器在 `mcf` 文件中定义的设备序数。
- slot* 自动化库中存储端口的编号，与库目录中标识的编号相同。该变量不适用于手动载入的驱动器。
- partition* 磁光盘的一面。`partition` 必须为 1 或 2。此变量不适用于磁带卡盒。

若要标记新光盘，请使用带以下选项的 `odlabel(1M)` 命令：

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

若要重新标记现有的光盘，请使用带以下选项的 `odlabel(1M)` 命令：

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

输入上述用以标记或重新标记光盘的命令后，系统会载入光盘并确定其位置，然后写入光盘标签。有关 `odlabel(1M)` 命令的详细信息，请参阅 `odlabel(1M)` 联机资料。

此外，您还可以使用 GUI 工具 `robottool(1M)` 和 `libmgr(1M)` 来执行此任务。有关这些工具的详细信息，请参阅第 11 页的“基本操作”或 `robottool(1M)` 和 `libmgr(1M)` 联机资料。

▼ 核查卷

有时，以前报告的磁带或光学卡盒上的剩余空间需要在库目录中进行更新。`auditslot(1M)` 命令用于载入包含卷的卡盒、读取标签并更新存储端口的库目录条目。

该命令的格式如下：

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

其中：

- e 如果指定 -e 选项，将会更新剩余空间。否则，将不会进行更新。
- eq 所访问的自动化库或手动载入驱动器在 mcf 文件中定义的设备序号。
- slot 自动化库中存储端口的编号，与库目录中标识的编号相同。该变量不适用于手动载入驱动器。
- partition 磁光盘的一面。partition 必须为 1 或 2。此变量不适用于磁带卡盒。

有关 auditslot(1M) 命令的详细信息，请参阅 auditslot(1M) 联机资料。

另外，您还可以使用 samu(1M) 实用程序的 :audit 命令或者使用 GUI 工具 robottool(1M) 和 libmgr(1M) 来执行此任务。有关这些工具的详细信息，请参阅第 181 页的“图形用户界面 (GUI) 工具”、第 217 页的“使用 samu(1M) 操作员实用程序”或下列其中一种联机资料：samu(1M)、robottool(1M) 或 libmgr(1M)。

▼ 核查自动化库（仅限于直接连接）

注 – 此任务不能在网络连接的自动化库上执行。

全面核查命令用于将每一个卡盒载入驱动器、读取标签并更新库目录。在下列情况下应核查库：

- 在未使用 Sun SAM-FS 或 Sun SAM-QFS 命令的情况下将卡盒移入自动化库之后。
- 如果您怀疑库目录的状态有问题，且希望更新库目录（例如在断电后）。
- 如果您在没有配备邮箱的自动化库中添加、取出或移动了卡盒。

要对自动化库进行全面核查，请使用 samu(1M) 实用程序的 :audit 命令。该命令的格式如下：

```
:audit eq
```

其中：

- eq 所访问的驱动器在 mcf 文件中定义的设备序号。

另外，您还可以使用 `samu(1M)` 实用程序的 `:audit` 命令或者使用 GUI 工具 `robottool(1M)` 和 `libmgr(1M)` 来执行此任务。有关这些工具的详细信息，请参阅第 181 页的“图形用户界面 (GUI) 工具”、第 217 页的“使用 `samu(1M)` 操作员实用程序”或下列其中一种联机资料：`samu(1M)`、`robottool(1M)` 或 `libmgr(1M)`。

使用清洁卡盒

Sun SAM-FS 和 Sun SAM-QFS 系统允许您导入清洁卡盒以清洁磁带驱动器。本过程的具体步骤取决于清洁卡盒是否编有条码。以下几节介绍了使用清洁卡盒的不同情况。

清洁卡盒的具体操作因制造商而异。如果您遇到这方面的问题，请参阅第 41 页的“特定自动化库的基本操作”以确定是否需要为您的设备执行特殊步骤。

注 – 此任务不能在网络连接的自动化库上执行。

▼ 重设清洁循环次数

清洁磁带仅在一定的清洁循环次数内有效。您可以使用 `samu(1M)` 实用程序的 `:v` 显示屏幕或通过 GUI 工具 `robottool(1M)` 或 `libmgr(1M)` 来查看剩余循环的次数。有关这些工具的详细信息，请参阅第 181 页的“图形用户界面 (GUI) 工具”、第 217 页的“使用 `samu(1M)` 操作员实用程序”或下列其中一种联机资料：`samu(1M)`、`robottool(1M)` 或 `libmgr(1M)`。

Sun SAM-FS 和 Sun SAM-QFS 系统可以跟踪用于每个清洁磁带的清洁循环次数，并在剩余循环次数等于零时弹出磁带。例如，DLT 清洁磁带的循环次数设置为 20，Exabyte 清洁磁带的循环次数设置为 10。每次导入清洁磁带时，系统均会自动将清洁循环次数重设至已为该类型磁带设定的最高循环次数。

如果您的系统支持自动清洁功能，但自动化库中所有清洁磁带的剩余循环次数均为零，则系统会将驱动器设置为 `off`（关闭），并在 Sun SAM-FS 或 Sun SAM-QFS 日志中输入一则消息。您可以使用 `chmed(1M)` 命令重设剩余循环次数为零的清洁磁带。该命令的格式如下：

```
chmed -count count media_type.vsn
```

其中：

count 您为清洁磁带重设的清洁循环次数。

media_type 介质类型。有关有效介质类型的列表，请参阅 `mcf(4)` 联机资料。

vsn 分配给卷的卷序列名。

▼ 使用带有条码的清洁卡盒

如果清洁卡盒编有条码，则您可使用 `import(1M)` 命令导入清洁卡盒。条码必须为 `CLEAN` 或必须以字母 `CLN` 开头。

`import(1M)` 命令采用以下格式：

```
import eq
```

其中：

`eq` 所访问的驱动器在 `mcf` 文件中定义的设备序号。

Sun SAM-FS 和 Sun SAM-QFS 系统将卡盒从邮箱移至存储端口，并更新每个卡盒的库目录。

例如，下面的命令将一个在 `mcf` 文件中编号为 50 的清洁磁带导入自动化库：

```
# import 50
```

发出此命令后，系统将设置清洁介质标记，并根据介质的类型将访问计数设置为适当的清洁循环次数。每使用介质清洁一次驱动器，访问计数便减少一次。

另外，您也可以通过 `samu(1M)` 实用程序或使用 `robottool(1M)` 或 `libmgr(1M)` 这两种 GUI 工具之一来执行此过程。有关这些工具的详细信息，请参阅 `samu(1M)`、`robotool(1M)` 或 `libmgr(1M)` 联机资料。

▼ 使用不带条码的清洁卡盒

如果清洁卡盒未编有条码，则必须首先将其导入。此时，它未被标记为清洁卡盒。请执行以下步骤：

1. 使用 `import(1M)` 命令导入卡盒。

`import(1M)` 命令采用以下格式：

```
import eq
```

其中：

`eq` 所访问的驱动器在 `mcf` 文件中定义的设备序号。

2. 使用 `chmed(1M)` 命令将设备类型更改为清洁卡盒。

您必须知道自动化库的设备序数以及清洁卡盒被载入的存储端口。

在下面的命令行示例中，自动化库的设备序数为 50，且清洁卡盒位于存储端口 77 中：

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

上述命令将卡盒类型更改为清洁卡盒类型。

3. 再次使用 `chmed(1M)` 命令，设置清洁循环次数。

下面的命令示例为上一步骤中的清洁卡盒设置了清洁循环次数：

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

有关 `chmed(1M)` 命令的详细信息，请参阅 `chmed(1M)` 联机资料。

▼ 清洁磁带驱动器

注 – Sun SAM-FS 和 Sun SAM-QFS 系统不能自动清洁网络连接的库。您应使用供应商提供的库管理软件来自动清洁此类库。

在硬件支持清洁磁带的情况下，Sun SAM-FS 和 Sun SAM-QFS 环境支持使用清洁磁带。如果某个磁带驱动器要求清洁，系统会自动载入清洁磁带。

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

注 – 某些驱动器错误可能会导致重复载入清洁卡盒，直到执行完所有的清洁循环。为防止发生此类情况，您可以使用 `chmed(1M)` 命令限制清洁卡盒的清洁循环次数。例如：

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

当系统不支持自动清洁功能且使用条码时，您可以使用 `cleandrive(1M)` 命令人工请求清洁驱动器。该命令的格式如下：

```
cleandrive eq
```

其中：

eq 所访问的驱动器在 `mcf` 文件中定义的设备序号。这是要载入清洁卡盒的驱动器。

▼ 清除介质错误

当卡盒发生硬件或软件错误时，Sun SAM-FS 系统会在 VSN 目录中设置 `media error`（介质错误）标记。对于任何生成 `media error`（介质错误）信号的给定卡盒，您均可使用 `chmed(1M)` 命令清除其错误，然后尝试使用该卡盒。`media error`（介质错误）标记会出现在 `samu(1M)` 实用程序的 `v` 显示屏幕、`robottool(1M)` VSN 目录面板和 `libmgr(1M)` VSN 显示屏幕中。

1. 输入 `chmed(1M)` 命令清除 `media error`（介质错误）标记。

下面的格式显示了用于清除 `media error`（介质错误）标记的 `chmed(1M)` 命令：

```
chmed -E media_type.vsn
```

其中：

media_type 介质类型。有关有效介质类型的列表，请参阅 `mcf(4)` 联机资料。

vsn 分配给卷的卷序列名。

2. 输入 `auditslot(1M)` 命令更新剩余空间信息。

下面的格式显示了用于完成此任务的 `auditslot(1M)` 命令选项：

```
auditslot -e
```

▼ 从驱动器中取出卡住的卡盒

如果卡盒卡在驱动器里面，请执行以下步骤：

1. 关闭自动化库中的驱动器。

若要关闭驱动器，请使用 `samu(1M)` 实用程序的 `:off` 命令。该命令的格式如下：

```
:off eq
```

其中：

`eq` 所访问的设备在 `mcf` 文件中定义的设备序数。

此外，您还可以使用 GUI 工具 `robottool(1M)` 和 `libmgr(1M)` 来执行此任务。有关这些工具的详细信息，请参阅第 181 页的“图形用户界面 (GUI) 工具”或 `robottool(1M)` 和 `libmgr(1M)` 联机资料。

2. 关闭自动化库。

若要关闭自动化库，请使用 `samu(1M)` 实用程序的 `:off` 命令。该命令的格式如下：

```
:off eq
```

其中：

`eq` 所访问的设备在 `mcf` 文件中定义的序数。

此外，您还可以使用 GUI 工具 `robottool(1M)` 和 `libmgr(1M)` 来执行此任务。有关这些工具的详细信息，请参阅第 181 页的“图形用户界面 (GUI) 工具”或 `robottool(1M)` 和 `libmgr(1M)` 联机资料。

3. 从驱动器中物理取出卡盒。

确保不要损坏卡盒或驱动器。

4. 打开自动化库和驱动器。

若要打开自动化库或驱动器，请使用 `samu(1M)` 实用程序的 `:on` 命令。该命令的格式如下：

```
:on eq
```

其中：

`eq` 所访问的设备在 `mcf` 文件中定义的序数。

如果自动化库在打开后执行了核查，则本过程结束。如果未执行核查，请执行下一步骤。

5. 如果您将卡盒放回其存储端口，请使用 `chmed(1M)` 命令调整库目录，以便为损坏的磁带设置占用标记。

该命令的格式如下：

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

其中：

`eq` 所访问的自动化库或驱动器在 `mcf` 文件中定义的设备序数。

`slot` 自动化库中存储端口的编号，与库目录中标识的编号相同。该变量不适用于手动载入的驱动器。

有关 `chmed(1M)` 命令的详细信息，请参阅 `chmed(1M)` 联机资料。

如果您取出卡盒，并希望在以后将其放回，则必须将卡盒导入自动化库。

导入和导出卡盒（配有邮箱的系统）

本节介绍如何为使用邮箱的直接连接自动化库导入和导出介质。这些库包括 StorageTek 9714、StorageTek 9710、StorageTek 9740 和 ADIC Scalar 系列等。

注 – 如果您使用网络连接的自动化库，请参阅第 41 页的“特定自动化库的基本操作”了解有关导入和导出卡盒的信息。

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

导入和导出的具体操作过程因制造商而异。如果您遇到这方面的问题，请参阅第 41 页的“特定自动化库的基本操作”以确定是否需要为您的设备执行特殊步骤。

如果您的系统没有安装邮箱，则此处介绍的操作不适用于您的系统。您应参阅第 37 页的“导入和导出卡盒（未配邮箱的系统）”。

▼ 使用邮箱导入卡盒

要将卡盒导入使用邮箱的自动化库，请执行以下步骤：

1. 使用制造商建议的操作打开邮箱。

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

2. 将卡盒手动放入邮箱。

3. 合上邮箱。

4. 使用 `import(1M)` 命令导入卡盒。

该命令的格式如下：

```
import eq
```

其中：

`eq` 所访问的自动化库在 `mcf` 文件中定义的设备序数。

系统将卡盒从邮箱移至存储端口，并更新每个卡盒的库目录。

另外，您也可以通过 `samu(1M)` 实用程序或使用 `robottool(1M)` 或 `libmgr(1M)` 这两种 GUI 工具之一来执行此步骤。有关这些工具的详细信息，请参阅 `samu(1M)`、`robottool(1M)` 或 `libmgr(1M)` 联机资料。

▼ 使用邮箱导出卡盒

本过程用于将卡盒从存储端口移至邮箱或邮槽。要从使用邮箱的自动化库中导出（弹出）卡盒，请执行以下步骤：

1. 使用 `samexport(1M)` 命令将卡盒从存储端口移至邮箱。该命令有以下两种格式：

```
samexport eq:slot  
samexport media_type.vsn
```

其中：

<i>eq</i>	所访问的自动化库在 <code>mcf</code> 文件中定义的设备序数。
<i>slot</i>	自动化库中存储端口的编号，与库目录中标识的编号相同。
<i>media_type</i>	介质类型。有关有效介质类型的列表，请参阅 <code>mcf(4)</code> 联机资料。
<i>vsn</i>	分配给卷的卷序列名。

另外，您也可以通过 `samu(1M)` 实用程序或使用 `robottool(1M)` 或 `libmgr(1M)` 这两种 GUI 工具之一来执行此步骤。有关这些工具的详细信息，请参阅第 181 页的“图形用户界面 (GUI) 工具”、第 217 页的“使用 `samu(1M)` 操作员实用程序”或下列其中一种联机资料：`samu(1M)`、`robottool(1M)` 或 `libmgr(1M)`。

2. 使用制造商建议的操作打开邮箱或邮槽。

邮箱的旁边通常配有一个按钮。

导入和导出卡盒（未配邮箱的系统）

本节介绍如何在自动化库未配邮箱的情况下操作卡盒。

▼ 在没有邮箱的情况下导入卡盒

1. 启动 `samu(1M)` 实用程序，然后输入 `:unload` 命令。

该命令的格式如下：

```
:unload eq
```

其中：

<i>eq</i>	所访问的设备在 <code>mcf</code> 文件中定义的序数。
-----------	------------------------------------

等待系统完成其当前任务，将自动化库的状态设置为 off（关闭），然后将当前活动的目录转送至 Historian。

2. 解除自动化库挡门的锁定，打开挡门。
3. 将卡盒装入可用的端口。
4. 合上自动化库的挡门，锁定挡门。

自动化库将重新初始化并扫描库中的卡盒。Sun SAM-FS 或 Sun SAM-QFS 软件通过将导入卡盒的 VSN 来添加至目录来更新库目录。此时，自动化库的状态会设置为 on（打开）。

▼ 在没有邮箱的情况下导出卡盒

1. 启动 **samu(1M)** 实用程序，然后输入 **:unload** 命令。

该命令的格式如下：

```
:unload eq
```

其中：

eq 所访问的设备在 *mcf* 文件中定义的序数。

等待系统完成其当前任务，将自动化库的状态设置为 off（关闭），然后将当前活动的目录转送至 Historian 文件。

2. 解除自动化库挡门的锁定，打开挡门。
3. 从各个端口中取出所需的卡盒。
4. 合上自动化库的挡门，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。系统使用库端口中当前卡盒的 VSN 来更新库目录。已取出的卡盒的 VSN 会从库目录中删除，并且现在只记录在 Historian 文件中。此时，自动化库的状态会设置为 on（打开）。

手动载入驱动器操作

本部分介绍在您配有手动载入的独立驱动器（而不是自动化库）时所执行的操作。每一个手动载入驱动器都有自身的单端口库目录。

▼ 载入卡盒

- 要将卡盒载入手动载入设备，请根据制造商的说明将卡盒放入驱动器。

Sun SAM-FS 和 Sun SAM-QFS 系统将识别载入的卡盒，读取其标签，然后更新手册式单端口目录。无需进行其它操作。

▼ 卸载卡盒

- 使用 `samu(1M)` 实用程序的 `:idle` 命令将驱动器置入空闲状态。

`:idle` 命令可以确保无任何存档或登台进程处于活动状态。该命令的格式如下：

```
:idle eq
```

其中：

`eq` 所访问的驱动器在 `mcf` 文件中定义的设备序数。

完成所有 I/O 活动后，驱动器的状态将从 `idle`（空闲）切换为 `off`（关闭），并且会弹出磁带。

如果是磁带，则先进行倒带，然后才能取出卡盒。如果是光学卡盒，则会自动弹出。有关取出特定卡盒的说明，请参阅制造商提供的说明。

您可以通过 GUI 工具 `libmgr(1M)` 或 `devicetool(1M)` 来执行本过程的全部步骤。有关这些工具的详细信息，请参阅第 181 页的“图形用户界面 (GUI) 工具”或者 `libmgr(1M)` 或 `devicetool(1M)` 联机资料。

▼ 查看库目录

- 使用 `samu(1M)` 实用程序的 `:v` 命令。

该命令的格式如下：

```
:v eq
```

其中：

`eq` 所访问的驱动器在 `mcf` 文件中定义的设备序数。

特定自动化库的基本操作

大多数库可以使用第 11 页的“基本操作”中所述的操作过程与 Sun SAM-FS 或 Sun SAM-QFS 软件配合工作。但是，某些库需要执行供应商提供的特定操作过程。本章介绍了这些过程。

注 – Sun SAM-FS 和 Sun SAM-QFS 软件可以与多家制造商生产的自动化库兼容。有关库模型编号、固件级别和其它兼容性信息，请参阅此版本附带的 README 文件。

本章包括下列主题：

- 第 42 页的“ADIC/Grau 自动化库”
- 第 47 页的“ADIC Scalar 系列库”
- 第 47 页的“Ampex 自动化库”
- 第 49 页的“Fujitsu LMF 自动化库”
- 第 54 页的“IBM 3584 UltraScalable 磁带库”
- 第 55 页的“IBM 3494 库”
- 第 55 页的“Sony 直接连接 B9 和 B35 自动化库”
- 第 56 页的“Sony 直接连接 8400 PetaSite 自动化库”
- 第 59 页的“Sony 网络连接自动化库”
- 第 63 页的“StorageTek ACSLS 连接自动化库”

ADIC/Grau 自动化库

ADIC/Grau 自动化库通过 `grauaci` 接口在 Sun SAM-FS 和 Sun SAM-QFS 环境中操作。`grauaci` 接口是 Sun SAM-FS 和 Sun SAM-QFS 软件与 GRAU ABBA 库之间的接口程序。它使用 ADIC/GRAU 提供的 DAS/ACI 3.02 接口。有关 DAS/ACI 的详细信息，请参阅《*DAS/ACI 3.02 Interfacing Guide*》和《*DAS Administration Guide*》，它们均由 ADIC/Grau 提供。

配置

开始 Sun SAM-FS 或 Sun SAM-QFS 配置过程之前，您必须确保 ADIC/Grau 自动化库可以正常操作，并且已在 DAS 服务器上运行 ABBA 库。在该客户机的 DAS 配置文件中，必须将 `avc`（消除卷争用，`avoid volume contention`）和 `dismount` 参数设置为 `true`。

以下几节介绍了仅适用于 ADIC/Grau 自动化库操作的其它配置信息。

mcf 文件

在创建 `mcf` 文件以加入一个或多个 ADIC/Grau 自动化库条目时，必须在 `mcf` 文件的 `Equipment Identifier`（设备标识）字段中填写 `grauaci` 接口所用参数文件的完整路径名。

有关 `mcf` 文件的详细信息，请参阅 `mcf(4)` 联机资料。

ADIC/Grau 参数文件

ADIC/Grau 参数文件列出了一系列关键字 = 值参数行。不同关键字值用于标识 ADIC/Grau 自动化库、与库关联的驱动器以及服务器名。所有关键字和值条目均区分大小写，因此，此处列出的关键字和值必须与 DAS 配置文件和 Sun SAM-FS 或 Sun SAM-QFS `mcf` 文件中所用的条目完全相同。ADIC/Grau 参数文件中必须包括以下类型的关键字 = 值参数：

- `client = client_id`。对于 `client_id`，指定 DAS 配置文件中定义的客户机的名称。这是必需的参数。
- `server = server_id`。对于 `server_id`，指定运行 DAS 服务器代码的服务器的主机名。这是必需的参数。

- `acidrive drive_id = path`。对于 `drive_id`，指定 DAS 配置文件中配置的驱动器的名称。对于 `path`，指定驱动器的路径，此处的路径必须与您在 Sun SAM-FS 或 Sun SAM-QFS mcf 文件的 `Equipment Identifier`（设备标识）字段中输入的路径相同。每一个指定至客户机的驱动器均必须有一个 `acidrive` 参数行。

注释可以出现在任何行内，但它们必须以井字符 (#) 开头。井字符右侧的字符会被忽略。

如果 ABBA 库中包含多种介质类型，则每一种介质类型均应有介质更换器。每一个介质更换器在 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。

有关 mcf 文件的详细信息，请参阅 mcf(4) 联机资料。

示例：本示例显示了 Sun SAM-FS mcf 文件和两个 ADIC/Grau 参数文件。下面显示的是 mcf 文件：

```
#
# Sample mcf file entries for a GRAU library - DLT
#
/etc/opt/SUNWsamfs/grau50 50 gr gr50 -
/var/opt/SUNWsamfs/catalog/gr50
/dev/rmt/0cbn 51 lt gr50 - /dev/samst/c2t5u0
/dev/rmt/1cbn 52 lt gr50 - /dev/samst/c2t6u0
#
# Sample mcf file entries for a GRAU library - HP optical
#
/etc/opt/SUNWsamfs/grau60 60 gr gr60 -
/var/opt/SUNWsamfs/catalog/gr60
/dev/samst/c1t1u0 61 od gr60 -
```

下面显示的是两个由上述 mcf 文件引用的参数文件。这两个文件分别定义了一个支持 DLT 磁带的 ADIC/Grau 自动化库和一个支持 Hewlett Packard 光盘驱动器的 ADIC/Grau 自动化库。

代码示例 3-1 显示了第一个参数文件。

代码示例 3-1 文件 `/etc/opt/SUNWsamfs/grau50`

```
# 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 # a comment
#
```

代码示例 3-1 文件 /etc/opt/SUNWsamfs/grau50 (续)

```
# the name "drive2" is from the DAS configuration file
#
acidrive drive2 = /dev/rmt/lc1bn      # a comment
```

代码示例 3-2 显示了第二个参数文件。

代码示例 3-2 文件 /etc/fs/samfs/grau60

```
# This is file: /etc/opt/SUNWsamfs/grau60
#
client = DASclient
server = DAS-server
acidrive DH03 = /dev/samst/clt1u0
#
# the name "DH03" is from the DAS configuration file
```

建立目录

Sun SAM-FS 和 Sun SAM-QFS 系统可以自动为 ADIC/Grau 自动化库建立库目录。但是，您必须在该库目录中添加条目。有数种方法可以完成此任务。具体采用的方法取决于您所需的目录大小。方法如下：

- **方法 1:** 使用现有的 VSN 条目创建目录。（请注意，此方法只适用于磁带，而不适用于编有条码的光学介质。）您可以使用 `build_cat(1M)` 命令建立包含多个磁带条目的目录。您需要创建一个包含端口编号、VSN、条码和介质类型的文件，以作为 `build_cat(1M)` 命令的输入项。例如下面的 `input_vsns` 文件：

```
0 TAPE01 TAPE01 lt
1 TAPE02 TAPE02 lt
2 TAPE03 TAPE03 lt
```

`input_vsns` 文件可以用作 `build_cat(1M)` 命令的输入项，如下所示：

```
# build_cat input_vsns /var/opt/SUNWsamfs/grau50cat
```

- **方法 2:** 创建一个空目录并导入 VSN 条目。您可以创建一个空目录，然后在其中添加条目。若要创建一个包含 1000 个端口的目录，请使用 `build_cat(1M)` 命令，如下所示：

```
# build_cat -s 1000 /dev/null /var/opt/SUNWsamfs/catalog/grau50cat
```

使用 `import(1M)` 命令将 VSN 添加到此目录，如下所示：

```
# import -v TAPE01 50
```

- **方法 3：**使用默认目录并导入 VSN 条目。如果未在 `mcf` 文件中指定目录路径名，则在初始化 Sun SAM-FS 或 Sun SAM-QFS 时，系统会自动在 `/var/opt/SUNwsamfs/catalog/family_set_name` 下创建默认目录。初始化之后，您必须将 VSN 条目导入此目录。若要完成此任务，请使用 `import(1M)` 命令，如下所示：

```
# import -v TAPE01 50
```

在上述 `import(1M)` 命令中，50 是 `mcf` 文件中指定的自动化库的设备序号。

`audit(1M)` 命令不能用于 ADIC/Grau 自动化库。

操作

第 11 页的“基本操作”介绍了可通过自动化库执行的大多数基本操作，其中包括标记卡盒和载入卡盒等。在 Sun SAM-FS 或 Sun SAM-QFS 环境中，这些基本操作几乎适用于所有自动化库。但是，ADIC/Grau 自动化库的某些基本操作在导入和导出卡盒方面与第 11 页的“基本操作”中所述的基本操作不同。

由于您使用供应商提供的实用程序在 ADIC/Grau 自动化库中物理添加或取出卡盒，因此 Sun SAM-FS `import(1M)` 和 `samexport(1M)` 命令以及 Sun SAM-FS `libmgr(1M)` 导入和导出菜单只对库目录有效。

此外，您也可使用 `samu(1M)`、`robottool(1M)` 或 `libmgr(1M)` 来执行导入和导出过程。有关这些工具的详细信息，请参阅 `samu(1M)`、`robotool(1M)` 或 `libmgr(1M)` 联机资料。有关 `import(1M)` 和 `samexport(1M)` 命令的详细信息，请参阅 `import(1M)` 和 `samexport(1M)` 联机资料。

以下两节介绍了导入和导出过程。

▼ 导入卡盒

若要导入卡盒，请执行以下步骤。

1. 使用 ADIC/Grau 命令将卡盒物理移入自动化库。

2. 使用 Sun SAM-FS 和 Sun SAM-QFS import(1M) 命令更新库目录。

import(1M) 命令的语法如下所示：

```
import -v volser eq
```

其中：

volser 要添加的 *volser* (卷序)。grauaci 接口在使用新条目更新库目录之前，将验证 ADIC/Grau 自动化库是否具有 *volser* (卷序) 信息。

eq 所访问的设备在 mcf 文件中定义的序数。

▼ 导出卡盒

若要导出卡盒，请执行以下步骤。

1. 使用 Sun SAM-FS 和 Sun SAM-QFS samexport(1M) 命令从库目录中删除卡盒的条目。

samexport(1M) 命令有以下两种语法格式：

```
samexport eq:slot  
samexport media_type.vsn
```

其中：

eq 所访问的设备在 mcf 文件中定义的序数。

slot 自动化库中存储端口的编号，与库目录中标识的编号相同。

media_type 介质类型。有关有效介质类型的列表，请参阅 mcf(4) 联机资料。

vsn 分配给卷的卷序列名。

导出每一个 VSN 后，samexport(1M) 命令将更新库目录，并将每一个 VSN 的库目录条目从库目录移至 Historian (历史记录) 目录。

2. 使用 ADIC/Grau 命令将卡盒物理移出自动化库。

诊断信息

用于故障排除的诊断信息位于下面的目录中：

```
/var/opt/SUNWsamfs/.grau
```

系统在此目录中创建的文件将命名为 `grau-log-eq`，其中 *eq* 表示在 `mcf` 文件中定义的设备序数。有关这方面的详细信息，请参阅 `grauaci(7)` 和 `mcf(4)` 联机资料。

ADIC Scalar 系列库

ADIC Scalar 100、Scalar 224 和 Scalar 448 系列自动化库包含 2 个或 4 个 DLT 驱动器，最多可以容纳 48 个 DLT 磁带。这些自动化库可以选用邮箱和条码读取器。

配置

在运行 Sun SAM-FS 或 Sun SAM-QFS 软件时，切勿为这些 ADIC 库配置自动清洁或自动载入功能。您可以在初始载入卡盒时使用自动载入功能，但条件是 Sun SAM-FS 或 Sun SAM-QFS 系统未正在运行。在运行 Sun SAM-FS 或 Sun SAM-QFS 系统时，请不要忘记禁用自动载入功能。

操作

使用第 11 页的“基本操作”中所述的实用程序执行导入、导出及其它任务。

Ampex 自动化库

Ampex 自动化库（例如 DST 812）使用 Ampex D2 磁带和驱动器。本部分介绍如何配置和初始化这些系统。

Ampex 自动化库的配置方法与其它直接连接库相似。磁带驱动程序和运行时库软件由 Ampex 提供，在安装 Sun SAM-FS 或 Sun SAM-QFS 软件之前，您必须安装它们。Sun SAM-FS 和 Sun SAM-QFS 系统要求使用 3.4 版或更新版本的 Ampex DST 磁带设备驱动程序。

▼ 配置 Ampex 库

若要配置 Ampex 库以便与 Sun SAM-FS 或 Sun SAM-QFS 系统配合使用，您需要在 `/etc/opt/SUNWsamfs/mcf` 文件中正确标识系统。代码示例 3-3 显示了用于配置单个 D2 磁带驱动器和自动化库的 mcf 文件条目。

代码示例 3-3 mcf 文件中的 Ampex 库条目

# Equipment	Eq	Eq	Family	Dev	Additional
# Identifier	Ord	Ty	Set	St	Parameters
#					
<code>/dev/samst/c5t6u0</code>	55	rb	am55	on	
<code>/dev/rdst6,1</code>	56	d2	am55	on	<code>/dev/rdst6,7</code>

下面的过程介绍了如何配置两个自动化库。

1. **确定指向 `/devices` 文件的 `/dev/samst` 符号链接。**它与 `/var/adm/messages` 文件中所示的 Sun Solaris 硬件路径相同。

本示例中为 `/dev/samst/c5t6u0`。

2. **确定设备序数。**

本示例中为 55。

3. **定义设备类型。**

本示例中为 `rb`。

4. **使用一个通用系列集名来关联自动化库和驱动器。**

本示例中为 `am55`。

5. **将设备状态设置为 `on`。**

6. **修改 `/usr/kernel/drv/dst.conf` 文件。**

您必须按以下所示设置 `DST_ZERO_ON_EW` 设备驱动程序位：

```
(set dst_dev_options = 0x00004001)
```

7. **修改 `/usr/kernel/drv/dst.conf` 文件之后，重新引导系统。**

Ampex D2 驱动器的 Equipment Type（设备类型）必须为 `d2`。对于这些系统，请不要使用通用 Equipment Identifier（设备标识）`tp`。系列集名与您为库选择的名称相同（`am55`）。

▼ 格式化 D2 磁带

由于 Sun SAM-FS 系统可以自动格式化磁带，因此您不必使用 Ampex 提供的命令对其进行格式化。

Sun SAM-FS 和 Sun SAM-QFS 系统使用 `tplabel(1M)` 命令及其 `-erase` 选项来格式化磁带。格式化期间，系统会在磁带中写入数个系统区域。这些区域包含的卷 ID 必须与磁带的 ANSI 标准卡盒标签相同。这些区域可使系统识别载入的磁带，而不必将磁带倒回至起始处以读取其 ANSI 标准标签。

Sun SAM-FS 和 Sun SAM-QFS 系统可以使用以下两种方法重新标记已格式化的 D2 磁带：使用 `tplabel(1M)` 命令的 `-erase` 选项；或在运行 `devicetool(1M)` 时从标签弹出式菜单中选择 `erase` 选项。使用 `-erase` 选项重新格式化磁带。新 VSN 将用作系统区域中的 `vol_id`。如果未选定 `-erase` 选项，您仍可进行重新标记，但必须使用磁带的上一个标签以保持系统区域 `vol_id` 和 ANSI 标签同步。

有关详细信息，请参阅 `tplabel(1M)` 或 `devicetool(1M)` 联机资料。

操作

当 Sun SAM-FS 或 Sun SAM-QFS 软件控制系统时，自动化库前面板上的按钮不会被禁用，因为它们用于其它自动化库。使用第 11 页的“基本操作”中所述的实用程序执行导入、导出及其它任务。



注意 – 不要使用 Ampex 库前面板上的按钮来载入和卸载磁带。使用这些按钮可能在 Sun SAM-FS 和 Sun SAM-QFS 环境或自动化库中产生意外的操作。

Fujitsu LMF 自动化库

Fujitsu LMF 自动化库通过 Fujitsu 提供的 LMF 接口在 Sun SAM-FS 和 Sun SAM-QFS 环境中操作。`fujitsulmf` 接口是 Sun SAM-FS 或 Sun SAM-QFS 软件与 Fujitsu LMF 自动化库之间的接口程序。有关 LMF 的详细信息，请参阅《*LMF MTL Server/Client User's Guide*》或《*LMF SAM-FS Linkage Operations Guide*》。这两本书均由 Fujitsu Corporation 提供。

配置

只有在安装 Fujitsu LMF 软件包并确定它们可以正常运行之后，才可以开始 Sun SAM-FS 或 Sun SAM-QFS 软件配置过程。以下几节介绍了仅适用于 Fujitsu LMF 自动化库操作的其它配置信息。

mcf 文件

在创建 mcf 文件以加入一个或多个 Fujitsu LMF 自动化库条目时，必须在 mcf 文件的 **Equipment Identifier**（设备标识）字段中填写 `fujitsulmf` 接口所用参数文件的完整路径名。

Sun SAM-FS 或 Sun SAM-QFS 环境中的每一个自动化库均必须在 mcf 文件中具有自身的标识行。要在 **Equipment Identifier**（设备标识）字段中指定多个路径名，请将这些路径分别放置在单独的行内。

Fujitsu LMF 参数文件

Fujitsu LMF 参数文件用于标识自动化库中的驱动器。每一个自动化库均必须有一个参数文件。参数文件的名称必须为 `/etc/opt/SUNWsamfs/eq`，其中 `eq` 表示 mcf 文件中定义的设备序号。

参数文件由一系列 `lmfdrive drivename = value` 定义行和注释行组成。每一个指定至自动化库的驱动器均必须有一个 `lmfdrive` 行。注释可以出现在任何行内，但它们必须以井字符 (#) 开头。井字符右侧的字符会被忽略。

所有 *drivename* 和 *value* 信息均区分大小写。表 3-1 列出了您可以为 *drivename* 和 *value* 指定的值。

表 3-1 *drivename* 和 *value* 变量

变量	定义
<i>drivename</i>	驱动器的名称，视 LMF 配置而定。
<i>value</i>	驱动器的路径。此路径必须与 mcf 文件中的 Equipment Identifier （设备标识）字段相符。

有关 mcf 文件的详细信息，请参阅 `mcf(4)` 联机资料。

示例：以下 mcf 条目定义了 Fujitsu LMF 自动化库：

```
#
# Sample mcf file entries for an LMF library
#
/etc/opt/SUNWsamfs/lmf50 50  fj  fj50  -
/etc/opt/SUNWsamfs/fj50_cat
    /dev/rmt/0cbn      51  fd  fj50  -  /dev/samst/c2t5u0
    /dev/rmt/1cbn      52  fd  fj50  -  /dev/samst/c2t6u0
The following is the parameters file referenced by the preceding
mcf file:
#
# This is file /etc/opt/SUNWsamfs/lmf50
#
# The name "LIB001DRV000" is from the LMF configuration.
#
lmfdrive LIB001DRV000 = /dev/rmt/0cbn  # defines first drive
#
# the name "LIB001DRV001" is from the LMF configuration
#
lmfdrive LIB001DRV001 = /dev/rmt/1cbn  # defines second drive
```

▼ 建立目录

Sun SAM-FS 和 Sun SAM-QFS 系统可以自动为 Fujitsu LMF 自动化库建立库目录。但是，您必须在该库目录中添加条目。有两种方法可以完成此任务。

- 方法 1：使用一系列 `import(1M)` 命令。请注意，只有卡盒实际位于 Fujitsu LMF 自动化库中时，才能成功运行 `import(1M)` 命令。如果没有卡盒，则此命令会在 `Historian`（历史记录）文件中添加一个条目。

下面的一系列命令将三个 VSN 示例的条目添入库目录中：

```
# import -v vsn1 50
# import -v vsn2 50
# import -v vsn3 50
```

- 方法 2：使用 `build_cat(1M)` 命令。此备用方法适用于具有许多卡盒的大型自动化库。您可以创建一个包含 VSN 列表的文件，然后对该文件运行 `build_cat(1M)` 命令。此命令使用 VSN 列表在初始的目录中添加条目。有关此命令的详细信息，请参阅 `build_cat(1M)` 联机资料。

请注意，磁带在 Fujitsu LMF 自动化库中的端口位置与 Sun SAM-FS 或 Sun SAM-QFS 库目录中 VSN 的端口编号无关。

下面的示例文件显示了 `build_cat(1M)` 命令使用的文件格式。该示例文件中包含一个用于在库目录中添入条目的 `VSN` 列表。第一列为 `Sun SAM-FS` 或 `Sun SAM-QFS` 库目录端口编号，随后依次是标签、条码和介质类型。此示例文件如下所示：

```
0 VSN186 VSN186 fd
1 VSN187 VSN187 fd
2 VSN188 VSN188 fd
3 VSN189 VSN189 fd
```

`audit(1M)` 命令不能用于 Fujitsu LMF 自动化库。

操作

第 11 页的“基本操作”介绍了可通过自动化库执行的大多数基本操作，其中包括标记卡盒、载入卡盒等。在 `Sun SAM-FS` 或 `Sun SAM-QFS` 环境中，这些基本操作几乎适用于所有自动化库。但是，Fujitsu LMF 自动化库的某些基本操作与这些典型的基本操作略有不同，其区别主要体现在导入和导出卡盒方面。

由于您使用供应商提供的实用程序在 Fujitsu LMF 自动化库中物理添加或取出卡盒，因此，`Sun SAM-FS` 和 `Sun SAM-QFS` 的 `import(1M)` 和 `samexport(1M)` 命令以及 `libmgr(1M)` 导入和导出菜单只对库目录有效。

您可以使用 `samu(1M)`、`robottool(1M)` 或 `libmgr(1M)` 来执行导入和导出过程。有关这些工具的详细信息，请参阅 `samu(1M)`、`robotool(1M)` 或 `libmgr(1M)` 联机资料。有关 `import(1M)` 和 `samexport(1M)` 命令的详细信息，请参阅 `import(1M)` 和 `samexport(1M)` 联机资料。

以下两节介绍了导入和导出过程。

▼ 导入卡盒

若要导入卡盒，请执行以下步骤。

1. 使用 `Fujitsu` 命令将卡盒物理移入自动化库。

2. 使用 Sun SAM-FS 和 Sun SAM-QFS `import(1M)` 命令更新库目录。

`import(1M)` 命令的语法如下所示：

```
import -v volser eq
```

其中：

volser 要添加的 *volser*（卷序）。`fujitsulmf` 接口在使用新条目更新库目录之前，将验证 LMF 接口是否具有 *volser*（卷序）信息。

eq 所访问的设备在 `mcf` 文件中定义的序数。

▼ 导出卡盒

若要导出卡盒，请执行以下步骤。

1. 使用 Sun SAM-FS 和 Sun SAM-QFS `samexport(1M)` 命令从库目录中删除卡盒的条目。

`samexport(1M)` 命令有以下两种语法格式：

```
samexport eq:slot  
samexport media_type.vsn
```

其中：

eq 所访问的设备在 `mcf` 文件中定义的序数。

slot 自动化库中存储端口的编号，与库目录中标识的编号相同。

media_type 介质类型。有关有效介质类型的列表，请参阅 `mcf(4)` 联机资料。

vsn 分配给卷的卷序列名。

导出每一个 VSN 后，`samexport(1M)` 命令将更新库目录，并将每一个 VSN 的库目录条目从 Sun SAM-FS 或 Sun SAM-QFS 库目录移至 Sun SAM-FS 或 Sun SAM-QFS 的 Historian（历史记录）目录。

2. 使用 `Fujitsu` 命令将卡盒物理移出自动化库。

IBM 3584 UltraScalable 磁带库

Sun SAM-FS 和 Sun SAM-QFS 环境支持 IBM 3584 UltraScalable 磁带库。以下几节说明了在 Sun SAM-FS 或 Sun SAM-QFS 环境中使用该库时如何进行清洁以及如何处理其分区功能。

导入

Sun SAM-FS 或 Sun SAM-QFS 软件启动时，不会自动导入邮箱中的卡盒。

清洁

要在 Sun SAM-FS 或 Sun SAM-QFS 环境中使用此库，您需要禁用自动清洁功能而启用主机清洁功能。有关此过程的信息，请参阅 *IBM 3584 UltraScalable Tape Library Planning and Operator Guide*, IBM publication GA32-0408-01。此外，您也可参阅 Sun Microsystems 提供的 `ibm3584(7)` 联机资料。

分区

此库可以包含数个磁带驱动器。如果您使用多个驱动器，则可能需要将这—物理库划分成两个、三个或四个逻辑库。如果您已将库划分成两个以上的逻辑库，则在将 IBM 3584 库添加到 Sun SAM-FS 或 Sun SAM-QFS 环境之前，应确保这些逻辑库可以正常工作。

从已分区的库中导出卡盒时，只有从中导出卡盒的逻辑库可以访问该抽屉端口。如果您手动取出卡盒并将其重新插入，则任何逻辑分区均可访问该卡盒。以下步骤说明了在该情况下执行的操作：

1. 打开挡门
2. 取出卡盒
3. 合上挡门
4. 等待挡门锁定，然后取消锁定
5. 打开挡门
6. 放回卡盒
7. 合上挡门

有关在 Sun SAM-FS 或 Sun SAM-QFS 环境中将此库用作逻辑分区库的详细信息，请参阅您的 IBM 文档或 ibm3584(7) 联机资料。

IBM 3494 库

IBM 3494 自动化磁带库通过 IBM 1mcpd 软件包在 Sun SAM-FS 和 Sun SAM-QFS 环境中操作。此软件包必须由 IBM 提供，并且在安装 Sun Microsystems 提供的 SUNWsamfs 软件包之前，您必须安装此软件包并确保其可以正常运行。

除安装上述软件包之外，您还必须配置该库的 /etc/ibmat1.conf 文件和参数文件。

有关如何配置 IBM 3494 库才能在 Sun SAM-FS 或 Sun SAM-QFS 环境中操作的详细信息，请参阅 ibm3494(7) 联机资料。

Sony 直接连接 B9 和 B35 自动化库

Sony B9 和 B35 系列自动化库包含 DTF 驱动器。这些叠式存储库可以选用条码读取器。

注 – 本部分所述的内容仅适用于 Sony 直接连接 B9 和 B35 自动化库，而与第 56 页的“Sony 直接连接 8400 PetaSite 自动化库”和第 59 页的“Sony 网络连接自动化库”无关。

配置

Sony B9 和 B35 系列自动化库在 Sun SAM-FS 或 Sun SAM-QFS 环境中运行时，不应配置自动清洁或自动载入功能。有关详细信息，请参阅第 32 页的“清洁磁带驱动器”。

您可以在初始载入卡盒时使用自动载入功能，但条件是 Sun SAM-FS 或 Sun SAM-QFS 系统未正在运行。在运行 Sun SAM-FS 或 Sun SAM-QFS 系统时，请不要忘记禁用此功能。

操作

Sony 自动化库的操作取决于是否配有邮箱。邮箱用于将卡盒导入自动化库和从中导出卡盒。使用第 11 页的“基本操作”中所述的实用程序执行导入、导出及其它任务。

Sony 直接连接 8400 PetaSite 自动化库

Sony 8400 PetaSite 系列自动化库不同于其它 Sony 型号产品，因为它配有八端口导入和导出邮箱（端口编号 400-407）。因此，该系统可以方便快捷地执行导入和导出操作。此自动化库使用条码读取器。

由于邮箱端口可以用作存储端口，因此 Sun SAM-FS 和 Sun SAM-QFS 库目录会反映邮箱端口的信息。

注 – 本部分所述的内容仅适用于 Sony 直接连接 8400 PetaSite 自动化库，而与第 55 页的“Sony 直接连接 B9 和 B35 自动化库”和第 59 页的“Sony 网络连接自动化库”无关。

▼ 导入磁带

若要导入磁带，请执行以下步骤。

1. 按下自动化库前面板上的打开/合上按钮，打开自动化库的挡门。
2. 将卡盒装入邮箱端口。
3. 按下自动化库前面板上的打开/合上按钮，手动合上邮箱的挡门。

自动化库将在合上挡门后检查邮箱端口，以获取卡盒条码。如果条码有问题，该端口的 in（进）和 out（出）指示灯均会闪烁。

4. 运行 `import(1M)` 命令以使 Sun SAM-FS 或 Sun SAM-QFS 系统识别导入的卡盒。

该命令的格式如下：

```
import eq
```

其中：

`eq` 所访问的设备在 `mcf` 文件中定义的序数。

另外，您还可使用 `libmgr` 或 `robottool` 来执行此步骤。有关这些工具的详细信息，请参阅 `libmgr(1M)` 或 `devicetool(1M)` 联机资料。

▼ 导出磁带

导出磁带卡盒的过程取决于您是否将邮箱端口用作存储端口。

▼ 在邮箱端口未用作存储端口时导出磁带

在邮箱端口未用作存储端口时，请使用下面的过程导出卡盒。

1. 运行 `move(1M)` 命令将卡盒移至邮箱端口（端口编号 400-407）。

该命令的格式如下：

```
move source_slot destination_slot eq
```

其中：

<i>source_slot</i>	卡盒当前所在端口的端口编号。
<i>destination_slot</i>	卡盒将要移至端口的端口编号。
<i>eq</i>	所访问的设备在 <code>mcf</code> 文件中定义的序数。

2. 按下自动化库前面板上的打开/合上按钮。

此时，挡门会打开。

3. 从邮箱端口中取出卡盒。

4. 按下自动化库前面板上的打开/合上按钮，手动合上邮箱的挡门。

5. 运行 `samexport(1M)` 命令以使 Sun SAM-FS 或 Sun SAM-QFS 系统识别导出的卡盒。

该命令的格式如下：

```
samexport eq
```

其中：

<i>eq</i>	所访问的设备在 <code>mcf</code> 文件中定义的序数。
-----------	------------------------------------

另外，您还可使用 `libmgr(1M)` 或 `robottool(1M)` 来执行此步骤。有关这些工具的详细信息，请参阅 `libmgr(1M)` 或 `devicetool(1M)` 联机资料。

▼ 在邮箱端口用作存储端口时导出磁带

如果您将邮箱端口用作存储端口，并且您要导出的卡盒位于其中一个邮箱端口时，请使用下面的过程导出卡盒。

1. 按下自动化库前面板上的打开/合上按钮。
此时，挡门会打开。
2. 从邮箱端口中取出卡盒。
3. 按下自动化库前面板上的打开/合上按钮，手动合上邮箱的挡门。
4. 运行 `samexport(1M)` 命令以使 Sun SAM-FS 或 Sun SAM-QFS 系统识别导出的卡盒。
该命令的格式如下：

```
samexport eq
```

其中：

`eq` 所访问的设备在 `mcf` 文件中定义的序数。

另外，您也可使用 `libmgr(1M)` 或 `robottool(1M)` 来执行此步骤。有关这些工具的详细信息，请参阅 `libmgr(1M)` 或 `devicetool(1M)` 联机资料。

▼ 如何将卡盒移至另一个端口

要将卡盒移至另一个端口，请执行以下步骤：

1. 确保来源端口中具有卡盒，而目标端口中没有卡盒。
2. 运行 `move(1M)` 命令。
该命令的格式如下：

```
move eq:source_slot destination_slot
```

其中：

`eq` 所访问的设备在 `mcf` 文件中定义的序数。

`source_slot` 卡盒当前所在端口的端口编号。

`destination_slot` 卡盒将要移至端口的端口编号。

另外，您也可使用 libmgr(1M) 或 robottool(1M) 来执行此步骤。有关这些工具的详细信息，请参阅 libmgr(1M) 或 devicetool(1M) 联机资料。

Sony 网络连接自动化库

Sony 网络连接自动化库通过 DZC-8000S 应用接口库软件包在 Sun SAM-FS 和 Sun SAM-QFS 环境中操作。该软件包用于向 PetaSite 控制器 (PSC) 提供应用程序接口 (API)。有关 DZC-8000S 接口的详细信息，请参阅 Sony 提供的 *Sony PetaSite Application Interface Library DZC-8000S*。

注 – 本分部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，而不适用于直接连接的 Sony 自动化库。有关直接连接的 Sony 库的详细信息，请参阅第 55 页的“Sony 直接连接 B9 和 B35 自动化库”或第 56 页的“Sony 直接连接 8400 PetaSite 自动化库”。

配置

以下几节介绍了仅适用于 Sony 自动化库操作的其它配置信息。

mcf 文件

在创建 mcf 文件以加入一个或多个使用 DZC-8000S 接口的 Sony 网络连接自动化库条目时，必须在 mcf 文件的 `Equipment Identifier` (设备标识) 字段中填写 DZC-8000S 接口所用参数文件的完整路径名。

Sony 参数文件

Sony 参数文件中列出了一系列关键字 = 值参数行。不同关键字值用于标识 Sony 自动化库、与库关联的驱动器以及主机名。所有关键字和值条目均区分大小写，因此，此处列出的关键字和值必须与配置文件和 Sun SAM-FS 或 Sun SAM-QFS mcf 文件中所用的条目完全相同。Sony 参数文件中必须包括以下类型的关键字 = 值参数：

- `userid = user_id`。对于 `user_id`，指定介于范围 $0 \leq user_id \leq 65535$ 之间的编号。`userid` 参数用于在初始化 PetaSite 自动化库功能期间标识用户。这是必需的参数。
- `server = server_id`。对于 `server_id`，指定运行 PSC 服务器代码的服务器的主机名。这是必需的参数。

- `sonydrive drive_id = path [shared]`。这是必需的参数；`mcf` 文件中定义的每一个驱动器均必须有一个 `sonydrive` 行。

对于 `drive_id`，指定 PSC 配置文件中配置的驱动器盒编号。

对于 `path`，指定驱动器的路径，此处的路径必须与您在 Sun SAM-FS 或 Sun SAM-QFS `mcf` 文件的 Equipment Identifier (设备标识) 字段中输入的路径相同。

`shared` 关键字为可选项。该库经过配置，可将其介质驱动器共享给多个主机上的多个 Sun SAM-FS 或 Sun SAM-QFS 进程。有关设置共享驱动器的详细信息，请参阅第 16 页的“共享驱动器”或 `sony(7)` 联机资料。

注释可以出现在任何行内，但它们必须以井字符 (#) 开头。井字符右侧的字符会被忽略。

有关 `mcf` 文件的详细信息，请参阅 `mcf(4)` 联机资料。

示例：本示例显示了 Sun SAM-FS `mcf` 文件和 Sony 参数文件。下面显示的是 `mcf` 文件：

```
#
# Sample mcf file entries for a Sony network-attached library
#
/etc/opt/SUNWsamfs/sonyfile 100 pe psc on
/dev/rmt/1cbn                101 so psc on
/dev/rmt/2cbn                102 so psc on
```

下面是由上述 `mcf` 文件引用的参数文件 `/etc/opt/SUNWsamfs/sonyfile`：

```
#
# 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/1cbn
sonydrive 1002 = /dev/rmt/2cbn shared
```

▼ 建立目录

与直接连接的自动化库类似，Sun SAM-FS 和 Sun SAM-QFS 系统可以自动为 Sony 自动化库建立库目录。但是，您必须在该库目录中添加条目。有两种方法可以完成此任务。

- 方法 1：使用一系列 `import(1M)` 命令。请注意，只有卡盒实际位于 Sony 自动化库中时，才能成功运行 `import(1M)` 命令。如果没有卡盒，则此命令会在 `Historian`（历史记录）文件中添加一个条目。

如果 VSN 名称中包含一个或多个空格，则 VSN 名称必须括在引号 (“ ”) 内。

下面的一系列命令将三个 VSN 示例的条目添入库目录中：

```
# import -v "SEG 99001" 50
# import -v vsn2 50
# import -v vsn3 50
```

- 方法 2：使用 `build_cat(1M)` 命令。此备用方法适用于具有许多卡盒的大型自动化库。您可以创建一个包含卷序列名 (VSN) 列表的文件，然后对该文件运行 `build_cat(1M)` 命令。此命令使用 VSN 列表在初始的目录中添加条目。有关此命令的详细信息，请参阅 `build_cat(1M)` 联机资料。

请注意，磁带在 Sony 自动化库中的端口位置与库目录中 VSN 的端口编号无关。

如果 VSN 名称中包含一个或多个空格，则 VSN 名称必须括在引号 (“ ”) 内。

示例：下面的示例文件显示了 `build_cat(1M)` 命令使用的文件格式。该示例文件中包含一个用于在库目录中添加条目的 VSN 列表。第一列为 Sun SAM-FS 或 Sun SAM-QFS 库目录端口编号，随后依次是标签、条码和介质类型。此示例文件如下所示：

```
0 A00001 "2000 B00001" so
1 A00002 B00002 so
2 TEST01 TEST01 so
3 TEST02 TEST02 so
```

`build_cat(1M)` 命令不能用于 Sony 网络连接自动化库。

操作

第 11 页的“基本操作”介绍了可通过自动化库执行的大多数基本操作，其中包括标记卡盒、载入卡盒等。在 Sun SAM-FS 或 Sun SAM-QFS 环境中，这些基本操作几乎适用于所有自动化库。但是，Sony 自动化库的某些基本操作与这些典型的基本操作略有不同，其区别主要体现在导入和导出卡盒方面。

由于您使用供应商提供的实用程序在 Sony 自动化库中物理添加或取出卡盒，因此 Sun SAM-FS `import(1M)` 和 `samexport(1M)` 命令以及 Sun SAM-FS `libmgr(1M)` 导入和导出菜单只对库目录有效。

您可以使用 `samu(1M)`、`robottool(1M)` 或 `libmgr(1M)` 来执行导入和导出过程。有关这些工具的详细信息，请参阅 `samu(1M)`、`robotool(1M)` 或 `libmgr(1M)` 联机资料。有关 `import(1M)` 和 `samexport(1M)` 命令的详细信息，请参阅 `import(1M)` 和 `samexport(1M)` 联机资料。

以下两节介绍了导入和导出过程。

▼ 导入卡盒

若要导入卡盒，请执行以下步骤。

1. 使用 Sony 命令将卡盒物理移入自动化库。
2. 使用 Sun SAM-FS 和 Sun SAM-QFS `import(1M)` 命令更新库目录。

`import(1M)` 命令的语法如下所示：

```
import -v [ " ] volser [ " ] eq
```

其中：

- | | |
|---------------|--|
| " " | 引号。如果 <i>volser</i> 包含空格，则必须括在引号内。 |
| <i>volser</i> | 要添加的 <i>volser</i> （卷序）。PSC API 接口在使用新条目更新库目录之前，将验证 Sony 自动化库是否具有 <i>volser</i> （卷序）信息。如果卡盒并没有位于库中，则会在 <i>Historian</i> （历史记录）目录中添入一个条目。 |
| <i>eq</i> | 所访问的库在 <i>mcf</i> 文件中定义的设备序数。 |

▼ 导出卡盒

若要导出卡盒，请执行以下步骤。

1. 使用 **Sun SAM-FS** 和 **Sun SAM-QFS** `samexport(1M)` 命令从库目录中删除卡盒的条目。
`export(1M)` 命令有以下两种语法格式：

```
samexport eq:slot  
samexport media_type.vsn
```

其中：

<i>eq</i>	所访问的设备在 <code>mcf</code> 文件中定义的序数。
<i>slot</i>	自动化库中存储端口的编号，与库目录中标识的编号相同。
<i>media_type</i>	介质类型。有关有效介质类型的列表，请参阅 <code>mcf(4)</code> 联机资料。
<i>vsn</i>	分配给卷的卷序列名。

导出每一个 VSN 后，`samexport(1M)` 命令将更新库目录，并将每一个 VSN 的库目录条目从库目录移至 **Historian**（历史记录）目录。

2. 使用 **Sony** 命令将卡盒物理移出自动化库。

StorageTek ACSLS 连接自动化库

Sun SAM-FS 和 Sun SAM-QFS 系统与 StorageTek ACSLS 连接自动化库之间的相互操作在许多方面与直接连接自动化库相似。但是，与直接连接自动化库不同的是，StorageTek ACSLS 连接自动化库在安装和配置过程中还需执行附加的步骤。

StorageTek 提供的 ACSLS 软件包用于控制自动化库。后台程序软件通过 ACSAPI 接口控制 StorageTek 自动化库。

以下部分介绍了如何管理 StorageTek ACSLS 连接自动化库，主要内容包括配置、基本操作、错误消息和其它内容。

配置

只有在安装 StorageTek ACSLS 软件包并确定它们可以正常运行之后，才可以开始 Sun SAM-FS 或 Sun SAM-QFS 软件配置过程。以下几节介绍了 StorageTek ACSLS 连接自动化库专用的其它配置信息。

mcf 文件

StorageTek ACSLS 连接自动化库和直接连接自动化库的 mcf 文件的不同之处在于以下字段：

- **Equipment Identifier** (设备标识) 字段包含由 stk 后台程序使用的参数文件的完整路径名。该参数文件定义了 Sun SAM-FS 或 Sun SAM-QFS 环境中 StorageTek 自动化库及其驱动器的系统特性。第 64 页的“StorageTek 参数文件”定义了此文件的内容。
- **Equipment Type** (设备类型) 字段包含设备类型的两字符记忆码。对于 ACSLS 连接库，必须使用 sk 设备类型。有关其它设备类型，请参阅 mcf(4) 联机资料。
- **Family Set** (系列集) 字段包含与自动化库连接的驱动器的名称。

代码示例 3-4 中所示 mcf 文件的条目定义了连接有两个 9840 驱动器的 StorageTek 自动化库。

代码示例 3-4 mcf 文件中的 StorageTek 库条目

# Equipment Identifier	Eq Ord	Eq Ty	Family Set	Dev St	Additional Parameters
#					
/etc/opt/SUNWsamfs/stk50	50	sk	sk50	on	
/dev/rmt/0cbn	51	sg	sk50	on	
/dev/rmt/1cbn	52	sg	sk50	on	

StorageTek 参数文件

配置期间，您必须为每一个 StorageTek ACSLS 连接自动化库创建参数文件。参数文件的每一行均必须以关键字或注释开头。所用的关键字如下所示：

- **access = *user_id***
指定用户标识。*user_id* 用于输入 StorageTek 用以控制访问的 *user_id*。这是参数文件的可选条目。如果未指定 **access =** 参数，则访问控制字符串为空字符串，表示无任何 *user_id*。
- **hostname = *host_name***
指定服务器的主机名。*host_name* 用于输入运行 StorageTek ACSLS 接口的服务器的主机名。有关 *host_name* 值的信息，请参阅版本附带的 README 文件。

- `portnum = port_number`

指定用于在 ACSLS 与 Sun SAM-FS 或 Sun SAM-QFS 软件之间通信的端口号。有关 `port_number` 值的信息，请参阅版本附带的 README 文件。

- `capacity = (index = value[, index = value] ...)`

设置 StorageTek 支持的卡盒容量。`index = value` 赋值对必须由逗号隔开，并且括在括号内。

对于 `index`，指定 StorageTek 提供的 `media_type` 文件的索引，该文件位于下面的 ACSLS 目录中：

```
/export/home/ACSSS/data/internal/mixed_media/media_types.dat
```

对于 `value`，输入卡盒类型的容量，以 1024 个字节为单位。Sun SAM-FS 和 Sun SAM-QFS 系统提供了 Sun SAM-FS 和 Sun SAM-QFS 4.0 版本的最新 `index` 默认值。通常，它仅用于为新的卡盒类型提供容量条目，或用于改写 StorageTek 所支持的容量。

表 3-2 显示了这些默认值。

表 3-2 容量默认值

索引	类型	容量
0	3480	210 MB (215040)
1	3490E	800 MB (819200)
2	DD3A	10 GB (10485760)
3	DD3B	25 GB (26214400)
4	DD3C	50 GB (52428800)
5	DD3D	0 (DD3 cleaning tape)
6	DLTIII	10 GB (10485760)
7	DLTIV	20 GB (20971520)
8	DLTIIIEXT	15 GB (15728640)
9	STK1R (9840)	20 GB (20971520)
10	STK1U	0 (STK1R cleaning tape)
11	EECART	1.6 GB (16777216)
12	JCART	0 GB (foreign label)
13	STK2P (T9940A)	60 GB (62914560)
14	STK2W	0 GB (T9940A cleaning tape)
15	KLABEL	0 GB (unsupported)
16	LTO-100G	100 GB (104857600)
17	LTO-50G	50 GB (52428800)

表 3-2 容量默认值 (续)

索引	类型	容量
18	LTO-35G	35 GB (36700160)
19	LTO-10G	10 GB (10485760)
20	LTO-CLN2	0 GB (cleaning tape)
21	LTO-CLN3	0 GB (cleaning tape)
22	LTO-CLN1	0 GB (cleaning tape)
23	SDLT	110 GB (115343360)

- `device_path_name = (acs = value, lsm = value, panel = value, drive = value) [shared]`

指定客户机上设备的路径。每一个连接至此客户机的驱动器均必须有一个 `device_path_name =` 条目。“`device_path_name =`”关键字后面是对 StorageTek 自动化库中驱动器的说明。该说明以左括号开始，随后是四个 `keyword = value` 赋值对，最后是右括号。

这四个 `keyword = value` 赋值对可由逗号（如上所示）、冒号或空格隔开。请使用 ACSLS 驱动器查询命令提供的信息来配置 `device_path_name` 参数。表 3-3 显示了 `value` 参数规格。

表 3-3 `value` 参数规格

value	内容
<code>acs</code>	驱动器在 StorageTek 库中配置的 ACS 编号
<code>lsm</code>	驱动器在 StorageTek 库中配置的 LSM 编号
<code>panel</code>	驱动器在 StorageTek 库中配置的 PANEL 编号
<code>drive</code>	驱动器在 StorageTek 库中配置的 DRIVE 编号

您可以在设备路径名的参数规格后面添加 `shared` 关键字。该关键字指定将驱动器共享给多个主机上的多个 Sun SAM-FS 或 Sun SAM-QFS 进程。有关设置共享驱动器的详细信息，请参阅第 16 页的“共享驱动器”或 `stk(7)` 联机资料。

下面是 StorageTek 自动化库参数文件的示例：

```
#
# This is file: /etc/opt/SUNWsamfs/stk50
#
hostname = baggins
portnum = 50014
access = some_user # No white space allowed in user_id
capacity = ( 7 = 20971520, 9 = 20971520 )
/dev/rmt/0cbn = (acs=0, lsm=1, panel=0, drive=1) shared
/dev/rmt/1cbn = (acs=0, lsm=1, panel=0, drive=2)
```

ssi.sh 脚本

sam-stkd 后台程序使用 ssi.sh 来确保 SSI 后台程序副本 ssi_so 的运行。如果 ssi_so 退出，该后台程序会启动另一个副本。如果您的站点拥有自身的 ssi.sh 版本，则应修改此脚本，使其在收到 SIGTERM 信号后退出。SIGTERM 是后台程序发出的用于停止进程的信号。

您可以在 /opt/SUNWsamfs/examples/ssi.sh 中找到 ssi.sh 脚本的示例。如果 /etc/opt/SUNWsamfs/ssi.sh 中没有副本，ssi.sh 脚本会在安装期间自动复制到该目录中。

▼ 建立目录

Sun SAM-FS 和 Sun SAM-QFS 系统可以自动为 StorageTek 自动化库建立库目录。但是，您必须在该库目录中添入条目。有两种方法可以完成此任务。

- 方法 1：使用一系列 import(1M) 命令。请注意，只有卡盒实际位于 StorageTek ACSLS 连接自动化库中时，才能成功运行 import(1M) 命令。如果没有卡盒，则此命令会在 Historian（历史记录）目录中添加一个条目。

下面的一系列命令将三个 VSN 示例的条目添入库目录中：

```
# import -v vsn1 50
# import -v vsn2 50
# import -v vsn3 50
```

- **方法 2:** 使用 `build_cat(1M)` 命令。此备用方法适用于具有许多卡盒的大型自动化库。您可以创建一个包含卷序列名 (VSN) 列表的文件，然后对该文件运行 `build_cat(1M)` 命令。此命令使用 VSN 列表在初始的目录中添加条目。有关此命令的详细信息，请参阅 `build_cat(1M)` 联机资料。

请注意，磁带在 StorageTek ACSLS 连接自动化库中的端口位置与库目录中 VSN 的端口编号无关。

下面的示例文件显示了 `build_cat(1M)` 命令使用的文件格式。该示例文件中包含一个用于在库目录中添加条目的 VSN 列表。第一列为库目录端口编号，随后依次是标签、条码和介质类型。此示例文件如下所示：

```
0 DLT186 DLT186 lt
1 DLT187 DLT187 lt
2 DLT188 DLT188 lt
3 DLT189 DLT189 lt
```

`audit(1M)` 命令不能用于 StorageTek ACSLS 连接自动化库。

常见问题和错误消息

下面的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。

示例 1: 当 StorageTek 参数文件中存在语法错误时，系统会生成下面的消息。检查您的 StorageTek 参数文件，看看其中是否存在语法错误。请注意，每一行必须以关键字或注释开头。有关 StorageTek 参数文件的详细信息，请参阅 `stk(7)` 联机资料。

```
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.
```

示例 2: 假设您收到两组错误消息。第一组消息如下所示：

```
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
```

第二组消息如下所示：

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

samu(1M) 实用程序的 :r 显示屏幕的输出如下所示：

```
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
```

正处于初始化状态或未初始化的驱动器通常会指示配置错误。验证 ACSLS 是否已启动并且正在运行。验证主机名。确定是否可以对主机名成功运行 ping(1M) 命令。

检查 StorageTek 参数文件中的 portnum。例如，在 ACSLS 5.3 中，用于其它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。

示例 3：在本示例中，使用 import(1M) 命令将某个 VSN 导入至库目录，但 StorageTek 自动化库中没有该 VSN。只有在 ACSLS 管理的自动化库中存在该卡盒时，才能成功运行 import(1M) 命令。系统会生成以下消息：

```
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
```

操作

第 11 页的“基本操作”介绍了可通过自动化库执行的大多数基本操作，其中包括标记卡盒、载入卡盒等。在 Sun SAM-FS 或 Sun SAM-QFS 环境中，这些基本操作几乎适用于所有自动化库。以下几节仅就导入和导出卡盒介绍了 StorageTek ACSLS 连接自动化库基本操作的不同。

邮箱是自动化库中的一个区域，它用于在自动化库中添加和取出卡盒。某些 StorageTek 自动化库一次只能导入和导出一个卡盒。Sun SAM-FS 和 Sun SAM-QFS 环境中支持的配有邮箱的 StorageTek 自动化库示例包括 StorageTek 9714 和 StorageTek 9710。StorageTek 9730 使用邮槽。在 StorageTek 文档中，邮箱和邮箱端口通常称为 CAP。

将卡盒导入 ACSLS 自动化库或从中导出时，Sun SAM-FS 和 Sun SAM-QFS 命令只对库目录有效。Sun SAM-FS 和 Sun SAM-QFS 命令并不能将卡盒物理导入自动化库或从中导出。您必须使用 ACSLS 命令才能物理移动卡盒。您应采取必要的措施，使 ACSLS 库存清单与 Sun SAM-FS 或 Sun SAM-QFS 目录保持一致。

此外，您也可使用 `samu(1M)`、`robottool(1M)` 或 `libmgr(1M)` 来执行导入和导出过程。有关这些工具的详细信息，请参阅 `samu(1M)`、`robottool(1M)` 或 `libmgr(1M)` 联机资料。

▼ 导入磁带

- 若要导入磁带卡盒，请使用 `import(1M)` 命令。

该命令的格式如下：

```
import -v vsn eq
```

其中：

`vsn` 分配给卷的卷序列名。

`eq` 所访问的设备在 `mcf` 文件中定义的序号。

`import(1M)` 命令将在库目录中添加一个新 VSN。如果 Historian（历史记录）目录中包含此 VSN，则 Sun SAM-FS 或 Sun SAM-QFS 软件会将 VSN 信息从 Historian（历史记录）目录移至库目录。

▼ 使用邮箱导出磁带

您既可以按端口导出磁带卡盒，也可以按 VSN 导出磁带卡盒。

- 若要导出磁带卡盒，请使用 `samexport(1M)` 命令。

该命令具有以下两种格式：

```
samexport eq:slot  
samexport media_type.vsn
```

其中：

<i>eq</i>	所访问的设备在 <code>mcf</code> 文件中定义的序数。
<i>slot</i>	自动化库中存储端口的编号，与库目录中标识的编号相同。
<i>media_type</i>	介质类型。有关有效介质类型的列表，请参阅 <code>mcf(4)</code> 联机资料。
<i>vsn</i>	分配给卷的卷序列名。

导出每一个 VSN 后，`samexport(1M)` 命令将更新库目录，并将每一个 VSN 的库目录条目从库目录移至 **Historian**（历史记录）目录。

存档

存档是指将文件从 Sun SAM-FS 或 Sun SAM-QFS 文件系统复制到可移动介质卡盒上的卷或另一个文件系统中磁盘分区的过程。在本章中，术语存档介质是指向其中写入存档卷的各种卡盒或磁盘位片。Sun SAM-FS 和 Sun SAM-QFS 存档能力包括许多功能，例如用于指定哪些文件需要立即存档的功能、用于指定哪些文件永不需要存档的功能以及用于执行其它任务的功能。

本章介绍存档程序的操作理论、制订站点存档策略的一般原则以及如何通过创建 archiver.cmd 文件来实施这些存档策略。

本章包括下列主题：

- 第 73 页的“存档程序 — 操作理论”
- 第 82 页的“archiver.cmd 文件说明”
- 第 86 页的“archiver.cmd 指令”
- 第 114 页的“磁盘存档”
- 第 119 页的“存档程序示例”
- 第 132 页的“存档程序原则”
- 第 133 页的“排除存档程序故障”

存档程序 — 操作理论

存档程序可以自动将 Sun SAM-FS 和 Sun SAM-QFS 文件存档至存档介质。存档和登台文件并不需要操作人员的参与。文件存档至存档介质上的卷，每个卷由称为卷序列名 (VSN, *volume serial name*) 的唯一标识来识别。存档介质中可以包含一个或多个卷。识别单个卷时，必须指定介质类型和 VSN。

存档程序在安装 Sun SAM-FS 或 Sun SAM-QFS 文件系统时自动启动。通过在下面的文件中插入存档指令，您可以为您的站点自定义存档程序的操作：

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

执行存档时，并不是一定要用 `archiver.cmd` 文件。如果此文件不存在，Sun SAM-FS 和 Sun SAM-QFS 环境将执行以下默认操作：

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

以下几节介绍存档组的概念，并说明在存档过程中执行的操作。

存档组

存档组表示一组要存档的文件。用户可以在任意多个文件系统中定义存档组。存档组中的文件共同遵守与大小、所有权、群组 and 目录位置有关的通用标准。存档组控制存档副本的目的地、保留存档副本的时间以及存档数据之前等待的时间。存档组中的所有文件均被复制到与该存档组关联的卷。文件系统中的某个文件可以且只能属于一个存档组。

创建或修改文件之后，存档程序会将其复制到存档介质中。存档文件与标准 UNIX `tar(1)` 格式兼容。这确保了 Sun Solaris 操作环境 (OE) 和其它 UNIX 系统之间的数据兼容性。该格式包括文件访问数据（即索引节点）和文件的路径。如果您的 Sun SAM-FS 或 Sun SAM-QFS 环境数据完全丢失，则 `tar(1)` 格式允许您使用标准 UNIX 工具和命令来恢复文件。另外，存档进程还会复制用于 Sun SAM-FS 和 Sun SAM-QFS 文件系统操作的数据。此类数据包括目录、符号链接、分段文件的索引和存档介质信息。

在本节的以后部分，术语文件是指文件数据和元数据。仅在需要加以区分时，才会使用术语文件数据和元数据。术语文件系统是指已安装的 Sun SAM-FS 或 Sun SAM-QFS 文件系统。

通常，管理员可为存档组取任意名称，但以下情况除外：

- 系统存在两个保留的存档组名：`no_archive` 和 `allsets`。
`no_archive` 存档组是系统默认定义的存档组。选入此存档组的文件永远不会被存档。临时目录（例如 `/saml/tmp`）中的文件可能会包括在 `no_archive` 存档组中。
`allsets` 存档组用于定义适用于所有存档组的参数。
- 系统保留每一个 Sun SAM-FS 或 Sun SAM-QFS 文件系统的存档组的名称，以便保存控制结构信息。Sun SAM-FS 和 Sun SAM-QFS 文件系统为每一个文件系统提供了默认存档组。对于每一个文件系统，存档程序不仅为其存档元数据，而且还存档文件数据。文件系统存档组包括目录和链接信息以及不属于其它存档组的任何文件。默认存档组的名称与其关联的文件系统的名称相同，并且不能更改。例如，如果一个文件系统的名称为 `samfs1`，则其存档组的名称应为 `samfs1`。
- 存档组名最长不得超过 29 个字符。所用的字符仅限于 26 个大小写字母、数字 0 至 9 以及下划线字符 (`_`)。

存档操作

默认情况下，存档程序将为每一个存档组创建一份副本，但您可以要求为每一个存档组创建多达四份副本。存档组和副本份数意味着占用一系列卷。存档副本在其它卷中提供文件的复件。

为确保完整无缺地存档文件，存档程序会在文件修改之后等待一段指定的时间，然后再对其进行存档。正如前面所述，这段时间称为*存档时限*。

文件中的数据须经更改后，该文件才能视为存档或重新存档的对象。如果只是访问文件，则不会对其存档。例如，对某个文件运行 `touch(1)` 或 `mv(1)` 命令并不会使系统存档或重新存档该文件。运行 `mv(1)` 命令只是改变文件的名称，而并没有更改文件数据。在故障恢复时，如果通过 `tar(1)` 文件进行恢复，则运行此命令可能会产生其它后果。有关故障恢复的详细信息，请参阅《*Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 故障恢复指南*》。

存档程序依据文件的存档时限来选择要存档的文件。用户可以为每一个存档副本定义存档时限。

用户可以使用 `touch(1)` 命令将其文件的默认时间参考更改成过去较早的时间或将来较晚的时间。但是，这可能会导致意外的存档结果。为了避免发生此类问题，存档程序会调节时间参考以使它们始终处于下面所示的范围：

```
creation_time < time_ref < time_now
```

以下几节介绍存档程序执行的步骤，包括从最初的文件扫描进程到文件复制进程。

步骤 1：扫描要存档的文件

每一个安装的文件系统均有单独的 `sam-arfind` 进程。`sam-arfind` 进程将定期扫描每一个文件系统，以确定哪些文件需要存档。`sam-arfind` 进程执行的第一个扫描过程是目录扫描。在此扫描期间，`sam-arfind` 按降序方式逐层搜寻目录树。它将检查每一个文件，如果文件不需要存档，则为其设置 `archdone`（已存档）文件状态标记。某些文件可能不需存档，例如，在已创建这些文件的所有存档副本时，或这些文件位于 `no_archive` 存档组中时。

连续扫描期间，`sam-arfind` 将扫描 `.inodes` 文件。它不会扫描标有 `archdone`（已存档）标记的索引节点。许多活动可以改变文件的存档状态，其中包括：重新存档文件、取消存档文件和更改文件等。这些活动将清除 `archdone`（已存档）标记。

`sam-arfind` 进程使用文件的属性说明来确定文件所属的存档组。用于确定文件所属存档组的特性包括：文件名的目录路径部分和使用标准表达式的完整文件名（可选）；文件所有者的用户名；文件所有者所属的群组名；最小文件大小；以及最大文件大小。

如果文件的存档时限到期或过期（即需要创建一份或多份副本），`sam-arfind` 会将文件添加到存档组的存档请求中。存档请求是属于同一个存档组的所有文件的集合。存档请求是一个文件，它位于下面的目录中：

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

此类文件是二进制文件，可以使用 `showqueue(1M)` 命令显示。

存档请求有时也称 *ArchReq*。

如果文件处于脱机状态，`sam-arfind` 进程将选择用作存档副本来源的卷。如果文件副本需要重新存档，`sam-arfind` 进程将选择包含这个需要重新存档的存档副本的卷。

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存档。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存档组的成员而单独进行存档。

存档优先级是通过文件属性特征以及与存档组关联的文件属性乘数计算出来的。事实上，其计算公式如下所示：

$archive_priority = (file_property_value \times property_multiplier)$ 之和

大多数 *file_property_value* 数值为 1 或 0，相当于属性是 TRUE 或 FALSE。例如，如果正在创建第 1 个存档副本，则第 1 个属性副本的值为 1，而第 2 个副本、第 3 个副本和第 4 个副本的属性值为 0。

存档时间和文件大小等其它文件属性的值可以是 0 或 1 之外的其它数值。

property_multiplier 数值由存档组的 `-priority` 参数决定。由于可为文件的各个方面（例如存档时间或大小）设定值，因此您的站点可以改变存档请求的优先级。有关 `-priority` 参数的详细信息，请参阅 `archiver.cmd(4)` 联机资料。

archive_priority 和属性乘数是浮点数值。所有属性乘数的默认值均为 0.0。此时，此存档请求被设置为存档请求中的最高文件优先级。

完成文件系统扫描之后，`sam-arfind` 进程会将每一个存档请求发送至存档后台程序 `sam-archiverd`，以安排将文件复制到存档介质。`sam-arfind` 进程随后进入休眠期，时间长度为已设定的存档时间间隔。在时间间隔结束时，`sam-arfind` 进程将重新开始扫描。

如果 `sam-arfind` 进程发现某个文件需要存档，但存档组的上一个存档请求尚未完成，则会忽略刚才发现的文件。

因此会出现文件系统扫描和文件复制同时存在的情况。

步骤 2：编辑存档请求

`sam-archiverd` 后台程序收到存档请求后，将对其进行编辑。本步骤介绍了编辑进程。

存档请求中的所有文件一次可能无法全部存档。这取决于存档介质的容量或存档程序命令文件中指定的控制条件。编辑是指一次从存档请求中选择多少文件进行存档的过程。完成存档请求的存档副本操作后，如果仍有需要存档的文件，则会重新编辑存档请求。

`sam-archiverd` 后台程序根据某些默认标准和站点特定标准来排列存档请求中的文件的顺序。默认操作是根据在文件系统扫描期间发现文件的顺序，依次将存档请求中的所有文件存档到同一个存档卷中。站点特定标准可以使您控制文件的存档顺序以及如何将它们分布到不同的卷中。这些标准称为 *存档组* 参数，其估算顺序为：`-reserve`、`-join`、`-sort` 和 `-drives`。有关这些参数的详细信息，请参阅 `archiver.cmd(4)` 联机资料。

如果存档请求属于已指定 `-reserve owner` 的存档组，则 `sam-archiverd` 后台程序将依据文件的目录路径、用户名或群组名来排列存档请求中的文件的顺序。此操作由存档组的 `-reserve` 参数控制。属于第一个 *owner* 的文件首先被选择进行存档。剩余文件将在以后进行存档。

如果存档请求属于已指定 `-join method` 的存档组，则 `sam-archiverd` 后台程序将根据指定的 `-join method` 对文件进行分组。如果同时还指定了 `-sort method`，则 `sam-archiverd` 后台程序将根据 `-sort method` 对每个组中的文件进行排序。此时，后台程序会将存档请求标记为“已组合”和“已排序”。

对于以后的编辑和安排进程，每一个组内的所有组合文件均被视为单个文件。

如果存档请求属于已指定 `-sort method` 的存档组，则 `sam-archiverd` 后台程序将根据 `-sort` 参数指定的排序方法对文件进行排序。`sam-archiverd` 后台程序通常会根据排序方法、存档时间、大小或目录位置将文件组合在一起，视排序方法而定。此时，`sam-archiverd` 后台程序会将存档请求标记为“已排序”。默认情况下，存档请求不会进行排序，因此存档程序将根据在文件系统扫描期间发现文件的顺序对文件进行存档。

`sam-archiverd` 后台程序可以确定文件是处于联机状态还是脱机状态。如果存档请求中既包含联机文件，也包含脱机文件，则会首先选择联机文件进行存档。

如果存档请求未要求按排序方法进行组合或排序，则会按存档副本所在的卷来排列脱机文件的顺序。这可确保同一卷中每一个存档组内的所有文件可以按它们在介质上的排列顺序同时登台。为脱机文件创建多份存档副本期间，脱机文件不会系统被释放，直到创建所有要求的副本。从同一卷中登台的所有文件将被选作首先进行存档的文件。

请注意，在存档脱机文件时，使用 `-sort` 或 `-join` 参数可能会对性能造成负面影响。这是因为要存档的文件的顺序可能与脱机文件所需的存档卷顺序不符。我们建议您仅在创建第一个存档副本时使用 `-join` 或 `-sort` 参数。开始创建其它副本时，如果存档介质有足够的空间，则其它副本会尽可能地采用第一个副本的存档顺序进行存档。

存档请求将被输入到 `sam-archiverd` 后台程序的安排队列中。

步骤 3: 安排存档请求

`sam-archiverd` 后台程序中的安排程序将在出现下列情况时立即执行：

- 存档请求被输入到安排队列中。
- 已完成某个存档请求的存档。
- 从目录服务器收到介质状态发生变化的消息。
- 收到更改存档程序状态的消息。

安排队列中的存档请求按优先级排列。安排程序每次执行时，均会检查所有存档请求，以确定是否可以将它们分配到 `sam-arcopy` 进程，从而将文件复制到存档介质中。

此时，必须存在用于创建文件副本的驱动器。必须存在可供存档组使用的卷，并且它们有足够的空间来容纳存档请求中的文件。

驱动器

如果存档组指定了 `-drives` 参数，则 `sam-archiverd` 后台程序会将存档请求中选定的文件分配到多个驱动器中。如果此时可用的驱动器数量少于 `-drives` 参数指定的数量，则使用实际可用的数量。

如果存档请求中的文件总容量小于 `-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 `-archmax` 参数指定的值或为介质定义的值。有关 `-archmax` 参数和 `archmax=` 指令的详细信息，请参阅 `archiver.cmd(4)` 联机资料。

如果存档请求中的文件总容量大于 `-drivemin` 值，则 `drive_count` 的值等于 `total_size / drivemin` 的结果。如果 `drive_count` 的值小于计算得出的驱动器数量，则所用的驱动器数量为 `drive_count`。

卷

系统必须存在一个或多个具有足够空间的卷来容纳存档请求中的全部文件或至少一部分文件。存档程序将使用最近用于存档组的卷（如果它有足够的空间）。另外，该卷不应是存档程序正在使用的卷。

如果可用于存档组的卷正在使用中，则存档程序会选择其它卷。只有在未指定 `-fillvsns` 参数时，此原则才适用。指定该参数时，存档请求不能另行安排。

如果存档请求太大，无法装入一个卷中，则系统会选择将适于该卷的文件存档至该卷中。如果存档请求包含的文件太大，无法装入一个卷中，并且未为存档请求选择卷溢出功能，则无法存档该文件。此时，系统会将一则说明此情况的消息发送到日志中。

您可以使用 `-ovflmin` 参数为存档组指定卷溢出功能或使用 `ovflmin=` 指令为介质指定卷溢出功能。有关 `-ovflmin` 参数和 `ovflmin=` 指令的详细信息，请参阅 `archiver.cmd(4)` 联机资料。此参数规格 `ovflmin` 用于确定文件溢出介质时的最小容量。为存档组指定的 `ovflmin` 优先于为介质定义的 `ovflmin`。如果文件的大小小于 `ovflmin`，则无法存档文件。此时，系统会将一则说明此情况的消息发送到日志中。

如果文件的大小大于 `ovflmin`，则会根据需要分配其它卷。系统将按容量逐渐减少的顺序选择其它卷，以尽可能地减少文件所占用的卷数量。

如果没有可用于存档请求的卷，则存档请求会等待。

在确定某个存档请求的安排优先级时，除使用步骤 1 中得出的存档优先级之外，系统还将使用其它某些属性，例如文件是处于联机状态还是处于脱机状态。有关自定义属性乘数的详细信息，请参阅 `archiver.cmd(4)` 联机资料中所述的 `-priority` 参数。

对于每一个存档请求，`sam-archiverd` 后台程序均会通过计算存档优先级和各种系统资源属性关联的乘数之和，来确定其安排优先级。这些资源属性与以下各项相关：存档请求排队的时间（秒数）；存档进程中使用的第一个卷是否已载入驱动器；以及其它方面。

使用经调整的优先级，`sam-archiverd` 后台程序指定每一个准备复制的存档请求。

步骤 4：对存档请求中的文件进行存档

准备对存档请求进行存档时，`sam-archiverd` 后台程序将查询每一个存档请求，标记存档文件 (`tarball`) 的界限以使每一个存档文件的大小不超过 `-archmax target_size` 参数规格。如果单个文件大于 `target_size`，它将成为存档文件中的唯一文件。

对于每一个存档请求和要使用的每一个驱动器，`sam-archiverd` 后台程序均会将存档请求分配至 `sam-arcopy` 进程，以便将文件复制到存档介质。如果单个文件大于 `target_size`，它将成为存档文件中的唯一文件。存档信息会被输入至索引节点。

如果已启用存档日志功能，则会创建存档日志条目。

如果文件已登台，系统会释放其磁盘空间。此进程会持续运行，直到列表中的所有文件存档完毕。

许多错误会导致文件复制失败，其中包括读取磁盘高速缓存或向卷写入数据时发生的错误。另外，文件状态的改变也会导致文件复制失败，其中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，以及文件被删除等。

`sam-arcopy` 进程退出后，`sam-archiverd` 后台程序将检查存档请求。如果某些文件尚未存档，则会重新编辑存档请求。

默认输出范例

下面是运行 `archiver -l` 时的输出范例：

```
# archiver

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

存档后台程序

`sam-archiverd` 后台程序负责安排存档活动。`sam-arfind` 进程用于将需要存档的文件分配至存档组。`sam-arcopy` 进程用于将需要存档的文件复制到选定的卷。

Sun SAM-FS 或 Sun SAM-QFS 开始运行时，`sam-fsd` 会启动 `sam-archiverd` 后台程序。`sam-archiver` 后台程序将执行 `archiver(1M)` 命令以读取 `archiver.cmd` 文件并建立用于控制存档操作的表。它将为每一个已安装的文件系统启动 `sam-arfind` 进程；同样，如果未安装文件系统，则会停止相关的 `sam-arfind` 进程。然后，`sam-archiverd` 进程将监控 `sam-arfind` 的运行状况并处理来自操作员或其它进程的信号。

存档日志文件和事件日志

`sam-arfind` 和 `sam-arcopy` 进程可以生成日志文件，该文件中包含每一个已存档或自动取消存档的文件的有关信息。日志文件连续记录了存档操作。您可以使用日志文件来查找文件的用于一般备份目的的先前副本。

默认情况下，系统不会生成此文件。您可以在 `archiver.cmd` 文件中插入 `logfile=` 指令以指定创建日志文件及其名称。您可以确定此文件的名称。有关日志文件的详细信息，请参阅本章中第 86 页的“`archiver.cmd` 指令”和 `archiver.cmd(4)` 联机资料。

存档程序使用 `syslog` 工具和 `archiver.sh` 在日志文件中记录警告及参考性消息。

下面是一些摘自存档程序日志文件的示例行，这些行包括每个字段的定义：

```
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
```

表 4-1 按从左至右的顺序列出了上表中字段的内容。

表 4-1 存档程序日志文件字段

字段	内容
1	存档活动，如下所示： <ul style="list-style-type: none">• A 表示已存档。• R 表示已重新存档。• U 表示已取消存档。
2	存档活动发生的日期，格式为 <code>yyyy/mm/dd</code> 。
3	存档活动持续的时间，格式为 <code>hh:mm:ss</code> 。
4	存档介质类型。有关介质类型的详细信息，请参阅 <code>mcf(4)</code> 联机资料。
5	卷序列名。
6	存档组和副本份数。
7	存档文件（ <code>tar(1)</code> 文件）在介质上的起始物理位置和存档文件中的文件偏置（采用十六进制表示）。

表 4-1 存档程序日志文件字段 (续)

字段	内容
8	文件系统名。
9	索引节点编号和世代编号。世代编号是在索引节点编号被重新使用后生成的一个附加编号，它与索引节点编号一起用来标识此使用的唯一性。
10	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，表示文件部分的大小。
11	相对于文件系统安装点的文件路径以及名称。
12	文件类型，如下所示： <ul style="list-style-type: none">• d 表示目录。• f 表示常规文件。• l 表示符号链接。• R 表示可移动介质文件。• I 表示段索引。• s 表示数据段。
13	溢出文件的一部分或段。如果文件既未溢出，也未分段，则此数字为 0。
14	文件存档至的驱动器的设备序数。

archiver.cmd 文件说明

默认情况下，无论 sam-fsd 是否启动，还是 Sun SAM-FS 或 Sun SAM-QFS 文件系统是否安装，存档程序均会运行。存档程序的默认设置如下：

- 将所有文件存档至所有可用的卷。
- 所有文件的存档时限为 4 分钟。
- 存档时间间隔为 10 分钟。

您很可能希望更改存档程序的活动以满足站点的特殊存档要求。这些活动由存档程序命令文件 (archiver.cmd) 中的指令控制。此文件的路径名如下所示：

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

如果此文件不存在，存档程序将执行默认活动。

archiver.cmd 文件

archiver.cmd 文件由以下类型的指令组成:

- 全局指令
- 存档组分配指令
- 存档组指令
- VSN 池指令
- VSN 关联指令

指令由从 archiver.cmd 文件读取的文本行组成。每一个指令包含一个或多个字段，它们由空格或制表符隔开。井字符 (#) 后面的任何文本均被视为注释，并且不会被检查。通过在某行的末尾添加一个反斜杠 (\)，可使该行续接至下一行。

archiver.cmd 文件中的某些指令可能要求您指定时间或字节单位。若要指定这些单位，请将第 83 页的表 4-2 “archiver.cmd 文件指令单位”中列出的字母作为数字的后缀，以表示其单位。

表 4-2 archiver.cmd 文件指令单位

单位后缀	含义
s	秒。
m	分钟，60 秒。
h	小时，3,600 秒。
d	天，86,400 秒。
w	周，604,800 秒。
y	年，31,536,000 秒。
b	字节。
k	千字节，2**10 或 1,024 字节。
M	兆字节，2**20 或 1,048,576 字节。
G	十亿字节，2**30 或 1,073,741,824 字节。
T	兆兆字节，2**40 或 1,099,511,627,776 字节。

archiver.cmd 文件示例

代码示例 4-1 显示了 archiver.cmd 文件范例。右侧的注释指出了第 83 页的“archiver.cmd 文件”中列出的各种指令类型。

存档程序每隔一分钟检查一次 archiver.cmd 文件的状态。如果 archiver.cmd 文件在存档程序执行期间发生了更改，则存档程序会停止安排存档副本，并等待当前进行中的存档副本结束。然后，它读取修改过的 archiver.cmd 文件并重新启动。

注 – 如果存档程序在 archiver.cmd 文件中发现错误，则会记录错误的数量并显示下面的消息：

```
Errors in archiver commands - no archiving will be done.
```

显示上面的消息后，存档程序会停止运行。在您更改 archiver.cmd 文件中的错误或存档程序收到来自 samu(1M) 的 arrun 或 arrestart 命令后，存档程序将重新启动。

无论您何时更改 archiver.cmd 文件，均应使用 archiver(1M) 命令检查其中是否存在语法错误。指定下面的 archiver(1M) 命令可以在当前 Sun SAM-FS 或 Sun SAM-QFS 系统上检验 archiver.cmd 文件：

```
# archiver -lv
```

上面的命令将列出所有选项，并将 archiver.cmd 文件、卷、文件系统内容和错误的列表写入至标准输出文件 (stdout)。出现错误时，存档程序将停止运行。您可以先对要使用的存档程序命令文件运行 archiver(1M) 命令，然后再将其移至 /etc/opt/SUNWsamfs/archiver.cmd。如果您在运行 archiver(1M) 命令时未提供输入文件，则系统将通过 archiver.cmd 文件生成存档程序信息。如果没有 archiver.cmd 文件，则会返回系统默认值。有关详细信息，请参阅 archiver(1M) 联机资料。

以下部分简要介绍了存档程序命令文件中的指令。有关这些指令的详细信息，请参阅 archiver.cmd(4) 联机资料。

代码示例 4-1 archiver.cmd 文件示例

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

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

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

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

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

archiver.cmd 指令

以下小节说明了 archiver.cmd 指令，如下所述：

- 第 86 页的“全局存档指令”
- 第 91 页的“用于控制特定文件系统存档活动的指令”
- 第 91 页的“存档组分配指令”
- 第 98 页的“存档副本指令”
- 第 101 页的“存档组参数”
- 第 111 页的“VSN 关联指令”
- 第 112 页的“VSN 池指令”

全局存档指令

全局指令控制存档程序的整体操作。archiver.cmd 文件中的全局指令可以通过第二个字段中的等号 (=) 或其它字段不存在来识别。这些指令可以使您根据站点的配置来优化存档程序的操作。

在 archiver.cmd 文件中，全局指令必须位于任何 fs= 指令前面。fs= 指令是专用于特定文件系统的指令。如果存档程序检测到全局指令位于 fs= 指令后面，则会发出一则消息。

interval 指令：指令存档时间间隔

存档程序将定期检查所有已安装的 Sun SAM-FS 和 Sun SAM-QFS 文件系统的状态。计时由存档时间间隔控制。存档时间间隔是指在每一个文件系统中执行扫描操作的时间间隔。若要更改时间间隔，请使用 interval 指令。该指令的格式如下：

```
interval=time
```

默认时间间隔为 10 分钟。如果存档程序收到 samu(1M) 实用程序的 :arrun 命令，将会立即扫描所有文件系统。

如果为文件系统设置 hwm_archive 安装选项，则可以自动缩短存档时间间隔。该安装选项指定存档程序在文件系统充满且超过空间占用上限时开始进行扫描。high=percent 安装选项用于设置文件系统的空间占用上限。

有关指定存档时间间隔的详细信息，请参阅 archiver.cmd(4) 联机资料。有关设置安装选项的详细信息，请参阅 mount_samfs(1M) 联机资料。

bufsize 指令：设置存档程序缓冲器大小

默认情况下，系统使用内存缓冲器将需要存档的文件复制到存档介质。您可以使用 `bufsize` 指令指定非默认的缓冲器大小和（可选）锁定缓冲器。这些操作可以改善系统的性能。您可以试用不同的 `buffer_size` 值以确定最适合的缓冲器大小。

该指令的格式如下：

```
bufsize=media buffer_size [ lock ]
```

其中：

- media* 指定 `mcf(4)` 联机资料中列出的存档介质类型。
- buffer_size* 指定介于 2 至 32 之间的数字（包括 2 和 32）。默认值为 4。此值乘以相应介质类型的 `dev_blksize` 值等于所用的缓冲器大小。用户可以在 `defaults.conf` 文件中指定 `dev_blksize` 的值。有关此文件的详细信息，请参阅 `defaults.conf(4)` 联机资料。
- lock* `lock` 变量指明存档程序在创建存档副本时是否使用锁定的缓冲器。如果指定 `lock`，存档程序将在 `sam-arcopy(1M)` 操作期间在内存中的存档缓冲器上设置文件锁定。这可以避免由于为每一个 I/O 请求锁定和取锁定缓冲器而造成的开销，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` 变量。如果内存不足，则可能会造成内存用尽的情况。
- 只有已为需要存档的文件启用直接 I/O 时，`lock` 变量才有效。默认情况下，不会指定 `lock` 变量，并且文件系统会在所有直接 I/O 缓冲器上设置锁定（包括用于存档的缓冲器）。有关启用直接 I/O 的详细信息，请参阅 `setfa(1)` 联机资料、`sam_setfa(3)` 库例程联机资料或 `mount_samfs(1M)` 联机资料中介绍的 `-O forcedirectio` 选项。

例如，您可以按以下方式在 `archiver.cmd` 文件的指令行中指定该指令：

```
bufsize=od 7 lock
```

此外，您还可为单个存档组设置 `bufsize` 和 `lock` 参数。有关上述指令的详细信息，请参阅第 101 页的“存档组参数”。

drives 指令：控制用于存档活动的驱动器数量

默认情况下，存档程序使用自动化库中的所有驱动器进行存档活动。若要限制存档程序使用的驱动器数量，请使用驱动器指令。

该指令的格式如下：

```
drives=auto_lib count
```

其中：

auto_lib *mcf* 文件中定义的自动化库的系列集名。

count 用于存档活动的驱动器数量。

archmax 指令：控制存档文件的大小

archmax 指令用于指定存档文件的最大容量。将用户文件组合在一起，便形成存档文件。达到 *target_size* 值后，将不能再向存档文件中添加用户文件。大容量用户文件将写入单个存档文件中。

存档文件的最大容量与介质有关。默认情况下，写入光盘的存档文件不得超过 5 MB。对于磁带，存档文件的最大默认容量为 512 MB。

若要更改默认值，请使用下面的指令：

```
archmax=media target_size
```

大容量的存档文件和小容量的存档文件均有优缺点。例如，在使用磁带进行存档活动时，将 *archmax* 设置成较大的值可以降低磁带驱动器停止和启动的次数。但是，在写入大量存档文件时，有可能提前到达磁带末尾，因而浪费了大量磁带空间。一般而言，*archmax* 的设置值不应超过介质容量的百分之五。例如，对于容量为 20G 的磁带，您可以使用下面的 *archmax* 指令：

```
archmax=sg 1G
```

此外，您还可为单个存档组设置 *archmax* 指令。

ovflmin 指令：控制卷溢出功能

Volume overflow（卷溢出）是指允许在多个卷上存档文件的过程。有关卷溢出功能的详细信息，请参阅第 267 页的“卷溢出”。

使用卷溢出功能之前，请务必充分理解卷溢出的含义。只有在全面评估卷溢出对您的站点所产生的影响后，才可以使用卷溢出功能。存档在多个卷上的文件会给故障恢复操作和回收操作带来很大的困难。

存档程序通过 `ovflmin` 指令来控制卷溢出功能。`ovflmin` 指令用于指定允许文件溢出卷的最小容量。默认情况下，存档程序会禁用卷溢出功能。

该指令的格式如下：

```
ovflmin = media minimum_file_size
```

其中：

media 介质类型。有关有效介质类型的列表，请参阅 `mcF(4)` 联机资料。

minimum_file_size 指定文件溢出时的最小容量。

例如，假设许多文件存档在 `mo` 介质卡盒中，并且产生严重的碎片（例如 25%）。由于这些文件不能完全充满卷的空间，因而导致每个卷中存有大量的未用空间。为了更好地压缩卷的未用空间，您需要将 `mo` 介质的 `ovflmin` 设置成略微小于最小文件的值。下面的指令将 `ovflmin` 设置为 150 MB：

```
ovflmin=mo 150m
```

请注意，在本示例中启用卷溢出功能会造成在存档和登台文件时需要载入两个卷。

此外，您还可为单个存档组设置 `ovflmin` 指令。

wait 指令：推迟存档程序启动

`wait` 指令使存档程序等待从 `samu(1M)` 发出的启动信号。当收到此信号后，存档程序才开始典型的存档操作。默认情况下，存档程序在由 `sam-fsd(1M)` 启动后开始存档操作。若要推迟存档，请使用 `wait` 指令。该指令的格式如下：

```
wait
```

此外，您还可为单个存档组设置 `wait` 指令。

notify 指令：重命名事件通知脚本

notify 指令用于将存档程序的事件通知脚本文件的名称设置为 *filename*。该指令的格式如下：

```
notify=filename
```

默认文件名为 `/opt/SUNWsamfs/sbin/archiver.sh`。

该文件由存档程序执行，允许您以站点特定的方式处理各种事件。脚本通过第一个变量的关键字来调用。可用的关键字包括：`emerg`、`alert`、`crit`、`err`、`warning`、`notice`、`info` 和 `debug`。

其它变量在默认脚本中加以说明。

logfile 指令：指定存档程序日志文件

存档程序可以生成包含每一个文件的存档、重新存档或自动取消存档信息的日志文件。日志文件连续记录了存档操作。若要指定日志文件，请使用 `logfile` 指令。该指令的格式如下：

```
logfile=pathname
```

其中：

pathname 指定日志文件的绝对路径名。默认情况下，系统不会生成此文件。

示例：假定您希望通过将前一天的日志文件复制到其它位置来每天备份存档程序日志文件。如果您确保在存档程序日志文件关闭时执行复制，则可以完成此任务。换言之，在系统打开存档程序日志文件以向其中写入信息时，您不能执行复制操作。您需要执行以下步骤：

1. 使用 `mv(1)` 命令在 UFS 中移动存档程序日志文件。这可给予 `sam-arfind(1M)` 或 `sam-arcopy(1M)` 一些操作时间以结束向存档程序日志文件写入信息。
2. 使用 `mv(1)` 命令将前一天的存档程序日志文件移至 Sun SAM-FS 或 Sun SAM-QFS 文件系统。

此外，您还可为单个存档组设置 `logfile` 指令。

用于控制特定文件系统存档活动的指令

在 `archiver.cmd` 文件中，您可以在全局指令后面添加专用于特定文件系统的指令。这些指令用于指定仅适于单个文件系统的活动。

fs 指令：指定文件系统

默认情况下，存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单个文件系统上。若要指定单个文件系统，请使用 `fs` 指令。该指令的格式如下：

```
fs=fsname
```

其中：

fsname 指定 `mcfs` 文件中定义的文件系统名。

这些指令后面出现的全局指令和存档组相关指令仅适用于指定的文件系统，直到出现下一个 `fs=` 指令。例如，您可以使用此指令为每一个文件系统指定不同的日志文件。

其它文件系统指令

数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单个文件系统的指令。无论在何处指定，它们的效用都是相同的。这些指令包括：

- `interval` 指令。有关此指令的详细信息，请参阅第 86 页的“`interval` 指令：指令存档时间间隔”。
- `logfile` 指令。有关此指令的详细信息，请参阅第 90 页的“`logfile` 指令：指定存档程序日志文件”。
- `wait` 指令。有关此指令的详细信息，请参阅第 89 页的“`wait` 指令：推迟存档程序启动”。

存档组分配指令

默认情况下，文件将作为（已为文件系统命名的）存档组的一部分进行存档。不过，您可以指定其它存档组以包含具有类似特征的文件。如果某个文件不属于您指定的任何存档组，则它将作为（已为文件系统命名的）默认存档组的一部分进行存档。

存档组成员指令可以将具有类似特征的文件分配至同一个存档组。这些指令的语句位于 `find(1)` 命令后面。每一个存档组分配指令均采用以下格式：

```
archive_set_name path [search_criteria1 search_criteria2 ... ] [file_attributes]
```

其中：

<i>archive_set_name</i>	站点为存档组定义的名称。它必须是存档组分配指令中的第一个字段。存档组名通常暗示属于该存档组的文件的特征。指定存档组名时，只能使用 26 个英文字母、数字 (0-9) 和下划线字符 (<code>_</code>)，而不得使用其它特殊字符或空格。存档组名的第一个字符必须是字母。若要防止存档各种文件，请将 <code>no_archive</code> 指定为 <i>archive_set_name</i> 。
<i>path</i>	相对于文件系统安装点的路径。这可以使存档组成员指令应用于多个 Sun SAM-FS 和 Sun SAM-QFS 文件系统。如果该路径包括文件系统的所有文件，请在路径字段中输入句点 (<code>.</code>)。不允许在路径的开头使用斜杠 (<code>/</code>)。 <i>path</i> 所指定目录中的文件及其子目录均视为属于该存档组。
<i>search_criteria</i>	可以指定零个、一个或多个 <i>search_criteria</i> 变量。指定搜索标准的目的是根据文件大小、文件所有权和其它要素限制存档组的范围。有关可用的 <i>search_criteria</i> 变量的信息，请参阅以下几节。
<i>file_attributes</i>	可以指定零个、一个或多个 <i>file_attributes</i> 参数。当 <code>sam-arfind</code> 进程在存档期间扫描文件系统时，将为文件设置这些文件属性。

示例 1：下面是典型的存档组成员指令：

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

示例 2：通过将文件归入名为 `no_archive` 的存档组，可以防止存档文件。下面的命令将防止存档 `tmp` 目录下的各级文件，而不论 `tmp` 目录处于文件系统中的哪一个目录下：

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

以下几节介绍了可以指定的 *search_criteria* 参数。

文件大小 *search_criteria*: -minsize 和 -maxsize

使用 `-minsize size` 和 `-maxsize size` 特征，可以通过文件大小来确定存档组的成员。对于 *size*，指定一个整数并在其后附上表 4-3 中所示的字母之一。

表 4-3 大小后缀

字母	含义
b	字节
k	千字节
M	兆字节
G	十亿字节
T	兆兆字节

示例：本示例中的命令行指定所有介于 500 KB 和 100 MB 之间的文件均属于存档组 `big_files`。超过 100 MB 的文件属于存档组 `huge_files`。命令行如下所示：

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

所有者和群组 *search_criteria*: -user 和 -group

使用 `-user name` 和 `-group name` 特征，可以通过所有权和群组关系来确定存档组的成员。例如：

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

所有属于用户 `sysadmin` 的文件均属于存档组 `adm_set`，所有具有群组名称 `marketing` 的文件均属于存档组 `mktng_set`。

使用样式匹配的文件名 *search_criteria*: -name *regex*

使用标准表达式可以指定属于某个存档组的文件的名称。作为一个 *search_criteria*，`-name regex` 规格指定任何与标准表达式 *regex* 相匹配的完整路径均为存档组的成员。

regex 变量后跟 `regexp(5)` 联机资料中列出的惯例。请注意，标准表达式不能后跟像 UNIX 通配符这样的惯例。

在系统内部，所有位于选定目录之下的文件（及其相对于文件系统安装点的指定路径）均会被列出并一起传送，以进行样式匹配。这可以使您在 `-name regex` 字段中创建样式以匹配文件名和路径名。

示例

1. 下面的指令将存档组 `images` 中的文件限定为那些以 `.gif` 结尾的文件。

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

2. 下面的指令选择以字符 `GEO` 开头的文件：

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

3. 您可以将标准表达式与 `no_archive` 存档组结合使用。下面的指令可防止存档任何以 `.o` 结尾的文件：

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

4. 假定您的 `archiver.cmd` 文件中包含下面所示的行：

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

对于此 `archiver.cmd` 文件，存档程序不会存档用户目录或子目录下的 `fred.*`。文件的存档情况如下所述：

- 下面所示的文件不会被存档：

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

- 下面所示的文件会被存档:

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

5. 假定您的 archiver.cmd 文件中包含下面所示的行:

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

对于此 archiver.cmd 文件, 存档程序不会存档用户根目录下的 fred.*, 但可以存档用户子目录和目录 share/marketing 下的 fred.*。在此情况下, 用户根目录刚好是 first_user。在本示例中, 从 share/marketing/ 到下一个斜杠字符 (/) 之间的任何目录均被视为用户的根目录。文件的存档情况如下所述:

- 下面所示的文件不会被存档:

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

- 下面所示的文件会被存档:

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

释放及登台 *file_attributes*: -release 和 -stage

使用 -release 和 -stage 选项可以分别设置与存档组中的文件相关联的释放和登台属性。这两种设置可以取代用户以前可能设置的释放或登台属性。如果在文件存档之后设置这些属性，则这些设置将从下一个存档的文件开始生效。

-release 选项的格式如下所示：

```
-release attributes
```

-release 指令的 *attributes* 后跟的惯例与 release(1) 命令相同，表 4-4 中列出了这些惯例。

表 4-4 -release 选项

属性	含义
a	完成第一个存档副本后，释放文件占用的磁盘空间。
n	永不释放文件占用的磁盘空间。
p	释放文件占用的部分磁盘空间。

-stage 选项的格式如下所示：

```
-stage attributes
```

-stage 指令的 *attributes* 后跟的惯例与 stage(1) 命令相同，表 4-5 中列出了这些惯例。

表 4-5 -stage 选项

属性	含义
a	联合登台文件。
n	永不登台文件。

下面的示例显示了如何使用文件名规格和文件属性来部分释放 Macintosh 资源目录：

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


存档组成员关系冲突

有时，在选择路径和其它文件特征以将文件归入存档组时，可能会造成混乱的存档组成员关系。系统采用下列方式来解决此类情况：

1. 选择存档组中最初的成员关系定义。
2. 首先选择局限于文件系统的成员关系定义，然后选择全局的成员关系定义。
3. 如果某个成员关系定义与先前的成员关系定义完全相同，则系统会发出错误通知。

作为这些规则的进一步措施，用户应在指令文件的前部设置更加严格的成员关系定义。

使用 `fs=fsname` 指令控制特定文件系统的存档活动时，系统首先评估指令对该文件系统的局部效用，然后评估其全局效用。这样，文件可以分配至局部存档组（包括 `no_archive` 存档组），而不是分配至全局存档组。在设置 `no_archive` 等全局存档组分配时，系统会默认采用此规则。

例如，假设 `archiver.cmd` 文件中包含下面所示的指令：

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

从上面的行看出，管理员本不准备存档两个文件系统中的 `.o` 文件。但是，由于局部存档组分配 `allfiles` 的评估先于全局存档组分配 `no_archive`，因此系统将存档 `samfs1` 和 `samfs2` 文件系统中的 `.o` 文件。

要确保不存档两个文件系统中的 `.o` 文件，可以使用下面的指令：

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

存档副本指令

如果不指定存档副本份数，则存档程序将为存档组中的文件创建一份存档副本。默认情况下，创建文件副本的存档时限为四分钟。如果您需要多份副本，则必须使用存档副本指令指定所有副本（包括第一个副本）。

存档副本指令以数字开头。该数字（1、2、3 或 4）是副本编号。数字后面是一个或多个用于指定该副本存档特征的变量。

存档副本指令必须紧跟在与它们相关的存档组分配指令的后面。每一个存档副本指令均采用以下格式：

```
copy_number [ -release | -norelease ] [archive_age] [unarchive_age]
```

以下几节介绍这些存档副本指令变量。

存档之后释放磁盘空间：-release

您可以使用副本编号后面的 `-release` 指令指定在创建存档副本之后自动释放文件所占用的磁盘空间。该选项的格式如下：

```
-release
```

在本示例中，属于群组 `images` 的文件在其存档时间达到 10 分钟时进行存档。在创建第一个存档副本后，系统会释放文件占用的高速缓存空间。

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

推迟释放磁盘空间：-norelease

您可能希望在创建多份存档副本之后释放文件占用的磁盘空间。`-norelease` 选项可以使系统在创建所有标记 `-norelease` 的副本之后自动释放磁盘高速缓存。该选项的格式如下：

```
-norelease
```

下面的示例指定了一个名为 `vault_tapes` 的存档组。系统将为其创建两份副本，但它在创建这两份副本之后再释放与该存档组相关的磁盘高速缓存。这种方案可用于需要在创建离站 `vault` 磁带之前联机访问文件的站点。

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

请注意，仅需创建单个存档副本时，`-norelease` 参数规格对自动释放磁盘空间无效，因为只有创建至少一个副本后才能释放磁盘空间。另外，`-norelease` 和 `-release` 这两个参数规格相互排斥，不能同时使用。

设置存档时限

您可以通过在该指令的下一个字段中指定存档时限来设置文件的存档时限。您可以使用表 4-6 中所示的后缀字符来指定存档时限。

表 4-6 时限后缀

时间后缀	含义
s	秒
m	分钟
h	小时
d	天
w	周
y	年

在下面的示例中，目录 `data` 中的文件将在其存档时间到达 1 个小时后进行存档：

```
ex_set data
  1 1h
```

自动取消存档

如果您为文件指定多个存档副本，则可能需要只保留一个存档副本，而自动取消存档其它所有副本。当使用不同存档时限将文件存档至不同介质时，可能发生此类情况。

下面的示例指定了取消存档时限：

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

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存档。当文件的存档时间达到 10 周时，系统会创建第二个和第三个存档副本，并且会取消存档第一个副本。

有关控制取消存档的其它方法，请参阅第 105 页的“控制取消存档”。

为元数据指定多份副本

如果需要多份元数据的副本，您可以在指令文件中 `fs=` 指令的后面直接添加副本定义。例如：

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

在本示例中，系统将在四小时后创建 `samfs7` 文件系统的元数据的第一份副本，然后在 12 小时后创建第二份副本。

文件系统元数据包括对文件系统中路径名的更改。因此，如果您经常更改目录，则系统会创建新的存档副本。这会造成系统频繁地载入您为元数据指定的卷。

存档组参数

`archiver.cmd` 文件中的存档组参数部分始于 `params` 指令，止于 `endparams` 指令。存档组指令的格式如下：

```
params
archive_set_name.copy_number [ -param1 -param2 ...]
.
.
.
endparams
```

伪存档组 `allsets` 提供了一种为所有存档组设置默认存档组指令的方法。所有 `allsets` 指令必须位于实际存档组副本指令的前面，其原因是为单独存档组副本设置的参数可以取代由 `allsets` 指令设置的参数。有关 `allsets` 存档组的详细信息，请参阅 `archiver.cmd(4)` 联机资料。

本部分介绍除 `-disk_archive` 参数之外的其它所有存档组处理参数。有关 `-disk_archive` 参数的详细信息，请参阅第 114 页的“磁盘存档”。

设置存档程序缓冲器大小

默认情况下，系统首先将需要存档的文件存储在缓冲器的内存中，然后再将它写入存档介质。您可以使用 `-bufsize` 参数指定非默认的缓冲器大小。此类操作可以改善系统的性能。您可以试用不同的 `buffer_size` 值以确定最适合系统的缓冲器大小。

该参数的格式如下：

```
-bufsize=buffer_size
```

`buffer_size` 指定介于 2 至 32 之间的数字（包括 2 和 32）。默认值为 4。此值乘以相应介质类型的 `dev_blksize` 值等于所用的缓冲器大小。您可以在 `defaults.conf` 文件中指定 `dev_blksize` 的值。有关此文件的详细信息，请参阅 `defaults.conf(4)` 联机资料。

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

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

此外，通过指定全局 `bufsize=media buffer_size` 指令，也可以达到与该指令同样的效果。有关此主题的详细信息，请参阅第 87 页的“`bufsize` 指令：设置存档程序缓冲器大小”。

指定存档缓冲器锁定

默认情况下，系统首先将需要存档的文件存储在缓冲器的内存中，然后再将它写入存档介质。如果已启用直接 I/O，则您可以使用 `-lock` 参数锁定此缓冲器。此操作可以改善系统的性能，并且您可以使用该参数进行实验。

该参数的格式如下：

```
-lock
```

`-lock` 参数指明存档程序在创建存档副本时是否使用锁定的缓冲器。如果指定 `-lock`，存档程序将在 `sam-arcopy(1M)` 操作期间在内存的存档缓冲器上设置文件锁定。这样可以避免对缓冲器进行分页，因而提高了系统性能。

仅在配有大量内存的大型系统上，才有必要指定 `-lock` 参数。如果内存不充足，则可能会造成内存用尽的情况。

只有已为需要存档的文件启用直接 I/O 时，`-lock` 参数才有效。默认情况下，`-lock` 参数不会被指定，并且文件系统会在所有直接 I/O 缓冲器上设置锁定（包括用于存档的缓冲器）。有关启用直接 I/O 的详细信息，请参阅 `setfa(1)` 联机资料、`sam_setfa(3)` 库例程联机资料或 `mount_samfs(1M)` 联机资料中介绍的 `-O forcedirectio` 选项。

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

```
yourset.3 -lock
```

另外，通过指定 `bufsize=media buffer_size [lock]` 指令的 `lock` 变量，您还可以指定适用于全局的与该参数等效的参数。有关此主题的详细信息，请参阅第 87 页的“`bufsize` 指令：设置存档程序缓冲器大小”。

将多个驱动器分配给存档组

默认情况下，存档程序通常只使用一个介质驱动器对存档组中的文件进行存档。如果存档组中的文件数量众多或比较大，则使用多个驱动器可以帮助您更有效地存档。您可以使用 `-drives` 参数来指定多个驱动器。该参数的格式如下：

```
-drives number
```

number 指定用于对选定存档组进行存档的驱动器数量。

例如：

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

当存档组 `huge_files.2` 中的文件总容量等于或大于介质的 `drivemin` 的两倍时，应使用两个驱动器来存档文件。

在各个驱动器之间分割存档请求

您可以将 `-drivemin min_size` 参数与 `-drives` 指令结合使用，以设置在各个驱动器上分割存档请求的最小空间。例如，如果您希望在各个驱动器上分割存档请求，但是您又不希望让较小的存档请求占用全部驱动器，则可以使用 `-drivemin` 参数。它适用于大容量文件的操作。

`-drivemin` 参数的默认值是 `-archmax` 参数的设置。`-archmax` 参数的默认设置是正在使用的特定卷的 `target_size`。

`-drivemin` 选项的格式如下：

```
-drivemin min_size
```

`-drivemin` 参数将多个驱动器用于存档组的最小容量设置为 `min_size`。当使用 `-drives` 参数时，仅在一次存档的数据量大于 `min_size` 时才有必要使用多个驱动器。实际并行使用的驱动器数是 $arch_req_total_size/min_size$ 和 `-drives` 参数指定的驱动器数两者当中的较小者。

存档请求根据 `-drives` 和 `-drivemin` 这两个参数进行评估，如下所述：

- 如果存档请求小于 `min_size`，则只使用一个驱动器写入存档请求。
- 如果存档请求大于 `min_size`，则根据 `min_size` 评估存档请求，并安排适当数量的驱动器，最多不超过指定的全部驱动器数。
- 如果 `min_size` 为零，则尝试在所有指定的驱动器上分割存档请求。

例如，假设您在五个驱动器上分割一个名为 `big_files` 的存档组。根据存档组的大小，表 4-7 中显示了可能的分割方法。

表 4-7 存档组分割示例

存档组大小	驱动器数
小于 20 G	1
大于或等于 20 G 而小于 30 G	2
大于或等于 30 G 而小于 40 G	3
大于或等于 40 G 而小于 50 G	4
大于或等于 50 G	5

对于本示例，应在 `/etc/opt/SUNWsamfs/archiver.cmd` 文件中输入下面所示的行：

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

指定回收

回收进程可以使您收回由过期存档映像占用的存档卷。默认情况下，系统不会执行回收进程。不过，如果您想回收，则可以在 `archiver.cmd` 文件中指定回收指令以控制此进程。

有关 `archiver.cmd` 文件中支持的回收指令的详细信息，请参阅第 165 页的“回收”。

联合存档

在您指定 `-join path` 参数时，将使用联合存档功能。如果您希望将整个目录存档至一个卷，并且您知道存档文件实际只占用一个卷，则使用联合存档可以满足您的要求。否则，如果您想将目录保存在一起，请使用 `-sort path` 参数将文件连续地保存在一起。

将需要存档的文件写入卷时，存档程序会先将文件写入某个存档文件，以便高效地将用户文件组装至卷。将来，当从同一目录中访问文件时，您会遇到一段延迟，因为登台进程需要对卷进行重新定位以读取下一个文件。为了缩短延迟，您可以将来自同一目录路径的文件连续地存档至存档文件。联合存档进程将取代空间效率运算法则，从而将来自同一目录的文件存档在一起。`-join path` 参数可以使这些文件在存档组副本中连续地存档在一起。

当无需更改文件内容但需要始终同时访问一组文件时，使用联合存档可以满足此类要求。例如在医院里，用户可能使用联合存档来访问医疗图片。与同一位患者关联的图像可以保存在同一个目录中，以便医生可以同时访问这些图片。如果根据静态图片的目录位置连续存档图片，医生可以方便快捷地访问它们。例如：

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

注 – `-join path` 参数可将来自相同目录的数据文件写入至同一个存档文件。如果存在很多目录，但这些目录中的文件数量较少且容量很小，存档程序会创建很多较小的存档文件。由于数据文件对每个存档文件的 `tar(1)` 标题来说相对较小，因此这些较小的分散文件会降低系统的写入性能。当向高速磁带驱动器写入此类数据文件时，这会损害驱动器的写入性能。

另外，由于 `-join path` 参数指定将所有来自同一目录的文件存档至单个卷中，因此，一组文件可能不适合存档至任何可用的卷。在这种情况下，文件可能无法进行存档，直到为存档组分配了更多的卷。另一种情况可能是一组要存档的文件太大而永远不能存档至单个卷。在这种情况下，文件永远无法存档。

对于大多数情况，如果 `-join path` 参数的操作不受限制，则使用 `-sort path` 参数比使用 `-join path` 参数更为有效。

另外，您也可以按文件存档时间、大小或路径对存档组中的文件进行排序。`age` 和 `size` 变量相互排斥，不能同时使用。若要对存档组中的文件进行排序，请将 `-sort` 参数与变量 `age` 或 `size` 结合使用，如下所示：

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

第一个示例行将强制存档程序按路径名对存档请求进行排序。第二个示例行将强制存档程序按文件的存档时间（从最新到最早）对名为 `cardiac.2` 的存档组副本进行排序。第三个示例行将强制存档程序按文件的大小（从最大到最小）对名为 `catscans` 的存档组副本进行排序。

控制取消存档

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

示例 1

假定您的 Sun SAM-FS `archiver.cmd` 文件中包含下面所示的行：

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

如果由上述 `archiver.cmd` 文件控制的文件常被访问，它将始终保留在磁盘中，即使它的存档时间超过 60 天。只有在该文件未被访问的时间超过 60 天时才会删除副本 1 信息。

如果存档程序因文件未被访问的时间超过 60 天而删除其副本 1 信息，则有关人员从副本 2 登台文件时，需要从磁带中读取文件。该文件恢复联机后，存档程序会在磁盘上为其创建新的副本 1，并且为期 60 天的访问周期重新开始。如果该文件再次被访问，则 Sun SAM-FS 存档程序会为其重新生成新的副本 1。

示例 2

假设某位患者在医院里进行为期四周的疗养。在此期间，该患者的所有文件均位于快速介质中（副本 1 = mo）。四周后，该患者从医院出院。在该患者出院后的 60 天内，如果该患者的数据从未被访问过，则会取消存档索引节点中的副本 1 条目，而只保留副本 2 和副本 3 条目。此时，用户可以回收卷以便为当前患者腾出空间（因而不必增加磁盘库）。如果该患者六个月后到医院复查，则首先从磁带（副本 2）访问数据。现在，存档程序会在磁盘中自动创建一个新的副本 1，以确保在复查期间（可能持续数天或数周）可从快速介质中访问数据。

控制存档文件的写入方式

默认情况下，存档程序会首先写入一个磁带标记（EOF 标签），然后在存档文件之间写入多个磁带标记。当启动下一个存档文件时，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` 参数可以指示存档程序只写入初始的磁带标记。这样，驱动程序只需返回到上一个磁带标记（而不是第一个磁带标记）后面的位置，因而提高了性能。另外，如果指定 `-tapenonstop` 参数，存档程序将在复制操作的末尾处输入存档信息。

有关 `-tapenonstop` 参数的详细信息，请参阅 `archiver.cmd(4)` 联机资料。

保留卷

默认情况下，存档程序会将存档组副本写入由 `archiver.cmd` 文件中卷关联部分所述的标准表达式指定的卷。但是，有时可能需要存档组卷只包含来自同一存档组的文件。保留卷进程可以满足这一数据存储要求。

注 - `-reserve` 参数用于保留专供一个存档组使用的卷。站点使用保留卷时，可能导致频繁的卡盒载入和卸载操作。

使用 `-reserve` 参数可为存档组保留卷。在设置 `-reserve` 参数并将一个卷分配给某个存档组副本后，该卷的标识不会分配至其它任何存档组副本，即使某个标准表达式与之相匹配。

选择供某个存档组使用的卷后，系统会为该卷分配一个保留名称。此保留名称是联结存档组和卷的唯一标识。

`-reserve` 参数的格式如下所示：

<code>-reserve 关键字</code>

此处指定的关键字取决于您所使用的格式。可用的格式包括存档组格式、所有者格式和文件系统格式，如下所述：

- 存档组格式。此格式使用 `set` 关键字，如下所示：`-reserve set`
- 所有者格式。此格式使用下列其中一个关键字：`dir`、`user` 和 `group`。这些关键字的格式如下：

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

上述三种所有者格式相互排斥。也就是说，只能对存档组和副本使用这三种所有者格式的其中一种。

- 文件系统格式。此格式使用 `fs` 关键字，如下所示：`-reserve fs`

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

例如，在下面的 `archiver.cmd` 文件片段中，以 `arset.1` 开头的行依据存档组、群组和文件系统创建保留名称：

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

保留卷的信息存储在库目录中。库目录中的行包括介质类型、VSN、保留信息和保留日期及时间。保留信息包括存档组组件、路径名组件和文件系统组件，由双斜杠 (//) 隔开。

双斜杠并不表示路径名；它们仅仅是为显示保留名称的三个组件而设定的隔离符。如下面的库目录示例片段所示，说明保留卷的行以 `#R` 字符开头：

```
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 hgml.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
```

请注意，为符合页宽，上述代码片段中的行已作了删节。

一个或多个保留信息字段可以保留空白，视 `archiver.cmd` 文件中定义的选项而定。所列的日期和时间是指创建保留的时间。保留行添加到每一个保留卷（即在存档期间保留用于某个存档组的卷）的文件。

您可以使用 `samu(1M)` 实用程序的 `v` 显示屏幕来显示保留信息，也可使用以下一种 `archiver(1M)` 或 `dump_cat(1M)` 命令格式来进行显示：

```
archiver -lv
dump_cat -V catalog_name
```

下面的格式说明了每一种显示参数、关键字和分配至卷的保留名示例的格式。

- 存档组格式。 `set` 关键字用于激活表 4-8 所示保留名称中的存档组组件。

表 4-8 存档组格式示例

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

例如，在下面的 `archiver.cmd` 文件片段中，以存档组名 `allsets` 开头的行将按存档组为所有存档组设置保留卷：

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

- 所有者格式。 `dir`、`user` 和 `group` 关键字用于激活保留名称中的所有者组件。`dir`、`user` 和 `group` 关键字相互排斥，不能同时使用。`dir` 关键字使用紧跟在存档组定义的路径参数规格后面的目录路径组件。`user` 和 `group` 是自我说明式的关键字。表 4-9 中显示了示例。

表 4-9 所有者格式示例

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

注 - `-reserve` 参数用于保留专供一个存档组使用的卷。如果存在很多目录，但这些目录中的文件数量较少且容量很小，则会造成很多较小的存档文件写入至每一个保留卷。由于数据文件对于每个存档文件的 `tar(1)` 标题来说相对较小，因此这些较小的分散文件会降低系统的性能。

- 文件系统格式。 `fs` 关键字用于激活保留名称中的文件系统组件。表 4-10 中显示了示例。

表 4-10 文件系统格式示例

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

本章末尾第 129 页的“示例 4”中介绍了使用保留卷的完整存档示例。

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

此外，您还可以使用 `reserve(1M)` 和 `unreserve(1M)` 命令分别保留及取消保留卷。有关这些命令的详细信息，请参阅 `reserve(1M)` 和 `unreserve(1M)` 联机资料。

设置存档属性

Sun SAM-FS 和 Sun SAM-QFS 文件系统为文件存档提供了可以配置的优先级体系。每一个文件均分配有优先级。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` 文件中为每一个存档组进行设置）计算出来的。文件属性包括联机/脱机、存档时间、创建副本的数量和大小。

默认情况下，存档程序不会对存档请求中的文件进行排序，并且所有属性乘数均为零。这将使存档程序按先发现先存档的顺序对文件进行存档。有关优先级的详细信息，请参阅 `archiver(1M)` 和 `archiver.cmd(4)` 联机资料。

通过设置优先级和排序方法，您可以控制文件的存档顺序。下面是您可以设置的优先级示例：

- 选择 `priority` 排序方法可以按优先级的顺序对存档请求中的存档文件进行存档。
- 更改 `archive_loaded` 优先级可以减少介质载入次数。
- 更改 `offline` 优先级可以使联机文件先于脱机文件存档。
- 更改 `copy#` 优先级可以按副本顺序创建存档副本。

表 4-11 中列出了存档优先级。

表 4-11 存档优先级

存档优先级	定义
-priority age	存档时间属性乘数
-priority archive_immediate	立即存档属性乘数
-priority archive_overflow	多个存档卷属性乘数
-priority archive_loaded	已载入存档卷属性乘数
-priority copy1	副本 1 属性乘数
-priority copy2	副本 2 属性乘数
-priority copy3	副本 3 属性乘数
-priority copy4	副本 4 属性乘数
-priority copies	创建副本属性乘数
-priority offline	文件脱机属性乘数
-priority queuwait	队列等待属性乘数
-priority rearchive	重新存档属性乘数
-priority reqrelease	请求释放属性乘数
-priority size	文件大小属性乘数
-priority stage_loaded	已载入登台卷属性乘数
-priority stage_overflow	多个登台卷属性乘数

VSN 关联指令

archiver.cmd 文件中的 VSN 关联部分用于将卷分配给存档组。此 VSN 关联部分始于 vsns 指令，止于 endvsns 指令。

通过下面格式的指令可以将一组卷分配给存档组：

```
archive_set_name.copy_num media_type vsn_expr ... [ -pool vsn_pool_name ... ]
```

其中：

<i>archive_set_name</i>	您的站点为存档组定义的名称。它必须是存档组分配指令中的第一个字段。存档组名通常暗示属于该存档组的文件的特征。指定存档组名时，只能使用 26 个英文字母、数字 (0-9) 和下划线字符 (_)，而不得使用其它特殊字符或空格。存档组名的第一个字符必须是字母。
<i>copy_num</i>	数字后面是一个或多个用于指定该副本存档特征的变量。存档副本指令以数字开头。该数字 (1、2、3 或 4) 是副本编号。
<i>media_type</i>	介质类型。有关有效介质类型的列表，请参阅 mcf(4) 联机资料。
<i>vsn_expr</i>	标准表达式。请参阅 regex(5) 联机资料。
<i>-pool vsn_pool_name</i>	已命名的 VSN 组。

VSN 关联部分至少应有三个字段：*archive_set_name* 和 *copy_number*、*media_type* 以及至少一个卷。*archive_set_name* 和 *copy_number* 通过一个句点 (.) 相连。

下面的几个示例以不同的方法指定了相同的 VSN。

示例 1： 下面的示例显示了两行 VSN 参数规格：

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

示例 2： 下面的示例显示了 VSN 参数规格，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 3: 下面的示例使用简单的标准表达式指定 VSN:

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

卷可由一个或多个 *vsu_expression* 关键字指定。这些关键字是 `regexp(5)` 联机资料中说明的标准表达式。请注意，这些标准表达式不能后跟像通配符这样的惯例。除标准表达式之外，您还可以指定从哪些 VSN 池中选择卷。VSN 池通过 `-pool vsu_pool_name` 指令和 VSN 关联来表示。

存档程序需要使用卷对存档组中的文件进行存档时，将检查自动化库和手动载入驱动器中选定介质类型的每一个卷，以确定它们是否符合 VSN 表达式的要求。存档程序将选择第一个符合表达式要求且包含足够空间以进行存档副本操作的卷。例如：

- 下面的指令指定将属于存档组 `ex_set` 的文件的第 1 个副本复制到介质类型 `mo`（它使用 20 个卷，名称从 `optic20` 至 `optic39`）：

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

- 下面的指令将属于存档组 `ex_set` 的文件的第 2 个副本复制到介质类型 `lt`（其中包含任何以 `TAPE` 开头的卷）：

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

如果您的 Sun SAM-FS 或 Sun SAM-QFS 环境被配置为按存档组进行回收，则不要将 VSN 分配给多个存档组。

注 – 在设置 `archiver.cmd` 文件时，请确保将卷分配给用于存档元数据的存档组。每一个文件系统均有一个与其自身名称相同的存档组。有关保存元数据的详细信息，请参阅 `samfsdump(1M)` 联机资料或《*Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 故障恢复指南*》。

VSN 池指令

`archiver.cmd` 文件中的 VSN 池部分始于 `vsnpools` 指令，止于 `endvsnpools` 指令或 `archiver.cmd` 文件的末尾。该部分命名了一组卷。

VSN 池是一个已命名的卷组。VSN 池非常适用于为某个存档组定义可用的卷，因为它可以提供有一个有益于为存档组分配和保留卷的缓冲器。

您可以使用 VSN 池定义多个单独的卷组，以供公司中的部门、群组中的用户、某类数据和其它传统的分组使用。系统会为 VSN 池分配名称、介质类型和一组卷。暂用池是指当 VSN 关联中的特定卷或另一个 VSN 池用尽时，系统临时使用的一组卷。有关 VSN 关联的详细信息，请参阅第 111 页的“VSN 关联指令”。

如果某个卷被保留用于存档组，则该卷将不能再供它所属的 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。您可以按以下格式输入 archiver(1M) 命令来查看 VSN 池：

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

VSN 池定义至少需要三个由空格分隔的字段：池名称、介质类型和至少一个卷。语法格式如下：

```
vsn_pool_name media_type vsn_expression
```

其中：

<i>vsn_pool_name</i>	指定 VSN 池。
<i>media_type</i>	由 2 个字符表示的介质类型。有关有效介质类型的列表，请参阅 mcf(4) 联机资料。
<i>vsn_expression</i>	标准表达式；可以有一个或多个 <i>vsn_expression</i> 变量。请参阅 regcmp(3G) 联机资料。

下面的示例使用了四个 VSN 池：users_pool、data_pool、proj_pool 和 scratch_pool。对于前面三个特定池，如果任何一个池的卷已用尽，则会选用暂用池 VSN。该示例的行如下所示：

```
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
```

磁盘存档

存档是指将文件从联机磁盘复制到存档介质的过程。通常，存档程序将存档副本写入至自动化库中磁光盘或磁带卡盒上的卷中；但在应用磁盘存档功能时，它将文件系统上的联机磁盘用作存档介质。

应用磁盘存档功能，不仅可以将 Sun SAM-FS 或 Sun SAM-QFS 文件系统中的文件存档至同一计算机主机系统中的另一个文件系统，而且还可将源文件存档至其它 Sun Solaris 系统中的另一文件系统。如果在两个主机系统上应用磁盘存档功能，则其中一个系统为客户机，另一个为服务器。客户机系统是指作为源文件宿主的系统。服务器系统是指作为存档副本宿主的系统。

存档文件写入至的文件系统可以是任何一个 UNIX 文件系统，而并不一定是 Sun SAM-FS 或 Sun SAM-QFS 文件系统。如果将磁盘存档副本写入至其它服务器系统，则该服务器系统至少应安装一个 Sun SAM-FS 或 Sun SAM-QFS 文件系统。

存档程序对待存档至磁盘卷的文件的方式与对待存档至自动化库中卷的文件相同。您仍然可以创建一份、二份、三份或四份存档副本。如果您创建多份存档副本，则可以将其中一个存档副本写入至磁盘卷，而将其它存档副本写入至可移动介质卷。另外，如果您通常将文件存档至 Sun SAM-FS 或 Sun SAM-QFS 文件系统中的磁盘卷，则存档程序将根据该文件系统的 `archiver.cmd` 文件中的规则对存档文件副本自身进行存档。

下面列出了存档至联机磁盘与存档至可移动介质的相似点和不同点：

- 与写入至磁光盘或磁带的存档副本不同，写入至磁盘的存档副本不会记录在目录中。另外，磁盘卷中的存档文件不会出现在 `Historian`（历史记录）目录中。
- 如果您要将文件存档至可移动介质卷，则在安装文件系统后，无需更改 `archiver.cmd` 文件中的任何默认值便可开始存档。但是，如果您要将文件存档至磁盘卷，则在安装文件系统之前必须编辑 `archiver.cmd` 文件以定义磁盘存档组。
- 磁盘存档不能使用 `mcf(4)` 文件中的条目。您需要在 `archiver.cmd` 文件中指定 `-disk_archive` 参数，并且需要在 `/etc/opt/SUNWsamfs/diskvols.conf` 文件中定义磁盘卷。后者是一个附加的配置文件，如果您仅将文件存档至可移动介质卷，则无需使用此文件。

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

- 如果存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。
- 如果存档副本写入至另一个 Sun Solaris 系统，则 `diskvols.conf` 文件将包含该服务器系统的主机名。在此情况下，该服务器系统中也必须有一个 `diskvols.conf` 文件，以定义那些有权向该服务器系统写入数据的客户机。

以下几节说明了如何配置文件以启用磁盘存档功能。

定义磁盘 VSN

磁盘卷在 `/etc/opt/SUNWsamfs/diskvols.conf` 文件中进行定义。该文件是您定义用于接收存档副本的磁盘卷的文件。以下几节对 `diskvols.conf` 文件作了说明，并提供了一些配置磁盘存档的原则。

diskvols.conf 文件

`diskvols.conf` 文件必须与需要存档的源文件同处于一个 Solaris 系统中。如果将源文件写入至另一 Solaris 系统中的存档副本，则该服务器系统中也必须有一个 `diskvols.conf` 文件。

`diskvols.conf` 文件可以包含注释。井字符 (#) 表示注释，# 右侧的所有文本均会被忽略。

`diskvols.conf` 文件可以包含延续行。若要延续某一行，请在该行的末尾添加一个撇号 (')。

`diskvols.conf` 文件中的每一行均分成以下几个字段：

- **VSN Name (VSN 名称)** 字段中包含磁盘存档卷的唯一名称。该名称最多可以包含 31 个字符。
- **Host Name (主机名)** 是一个可选字段。如果您将文件从客户机系统存档至服务器系统，则必须指定服务器系统的主机名。Host Name (主机名) 字段必须是存档副本写入至的文件系统所在的服务器主机系统的名称。存档副本写入至的 Sun Solaris 系统必须安装了 Sun SAM-FS 或 Sun SAM-QFS 文件系统。如果指定 Host Name (主机名) 字段，则其后必须跟一个冒号 (:)。如果在 Host Name (主机名) 字段中输入了信息，则还必须在该服务器系统中创建一个 `diskvols.conf` 文件。
- **Path (路径)** 字段指定存档副本写入至的位置 (目录)。开始存档之前，必须在此字段中已指定目录结构，必须已安装该目录所在的文件系统。如果已卸载文件系统，则不会进行存档。此处指定的 Path (路径) 必须是相对于安装点的路径。例如，在将存档副本写入至 `archivefs1` 文件系统的 `vsns` 目录时，应在 Path (路径) 字段中指定 `/archivefs1/vsns`。

在创建目的地目录时，Sun Microsystems 建议将该目录的写入权限只授予给超级用户 (root)。

示例：下面的 `diskvols.conf` 文件位于客户机系统 `pluto` 中：

```
# 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
```

在上面的 `diskvols.conf` 文件中，名为 `disk01` 和 `disk02` 的 VSN 将写入至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。

下面的 `diskvols.conf` 文件位于服务器系统 `mars` 中：

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

下面是客户机系统 `pluto` 中的 `archiver.cmd` 文件的片段：

```
params
arset1.2 -disk_archive disk01
arset2.2 -disk_archive disk02
arset3.2 -disk_archive disk03
endparams
```

配置原则

虽然磁盘存档卷的位置不受限制，但是，建议您不要将该卷设置在源文件所在的磁盘上。如果将客户机系统中的存档副本写入至服务器系统中的磁盘卷，则更加理想。我们建议您创建多个存档副本，并将它们写入至不同类型的存档介质。例如，您可以将第 1 个副本写入至磁盘卷，将第 2 个副本写入至磁带，而将第 3 个副本写入至磁光盘。

如果您将文件存档至服务器上的文件系统，则存档文件自身还会被存档至与目标服务器连接的库中的可移动介质卡盒。

定义磁盘存档组

您必须编辑作为源文件宿主的系统中的 `archiver.cmd` 文件，以便添入磁盘存档组的有关信息。

`archiver.cmd` 文件中的 `-disk_archive` 参数用于定义磁盘存档组。当数据写入存档磁盘的安装点时，存档程序使用此参数来维护数据的文件系统结构。如其它所有存档组处理参数一样，该参数也必须指定在 `params` 和 `endparams` 之间。

该指令的格式如下：

```
params
archive_set.copy_number -disk_archive VSN_Name
endparams
```

用于 *VSN_Name*，指定 `diskvols.conf` 文件中定义的 VSN。

注 – 将文件存档至联机磁盘时，只有 `archiver.cmd` 指令的一部分可用于磁盘存档组。这些用于定义存档组和配置回收的指令包括：

```
-disk_archive
-recycle_hwm
-recycle_ignore
-recycle_mailaddr
-recycle_mingain
```

有关回收指令的详细信息，请参阅 `archiver.cmd(4)` 联机资料。

示例 1

在本示例中，文件 `/sam1/testdir0/filea` 位于存档组 `arset0.1` 中，并且存档程序将文件 `/sam1/testdir0/filea` 的内容复制到称为 `/sam_arch1` 的目标路径。

有关此情况的 `diskvols.conf` 文件如下：

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

archiver.cmd 文件中包含以下有关磁盘存档的行:

```
.  
.br/>.br/>params  
arset0.1 -disk_archive disk01  
endparams  
.br/>.br>.
```

下面是对已存档至磁盘的 filea 文件运行 sls(1) 命令时的输出。在此输出示例中, dk 是磁盘存档介质的介质类型, disk02 是 VSN:

```
# sls -D filea  
mode: -rw-r----- links: 1 owner: root group: other  
length: 5766596 inode: 988  
archdone;  
copy 1: ---- Jan 11 08:34 0.1 dk disk02  
access: Jan 11 08:26 modification: Jan 11 08:26  
changed: Jan 11 08:26 attributes: Jan 11 08:26  
creation: Jan 11 08:26 residence: Jan 11 08:26
```

示例 2

在本示例中, 文件 /sam2/my_proj/fileb 位于客户机主机 snickers 的存档组 arset0.1 中, 存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。

客户机主机 snickers 上的 diskvols.conf 文件如下所示:

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

服务器主机 mars 上的 diskvols.conf 文件如下所示:

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

archiver.cmd 文件中有关本示例的指令如下所示:

```
.
.
.
params
arset0.1 -disk_archive disk01
endparams
.
.
.
```

存档程序示例

假设本部分中的所有存档程序示例采用下面的目录结构:

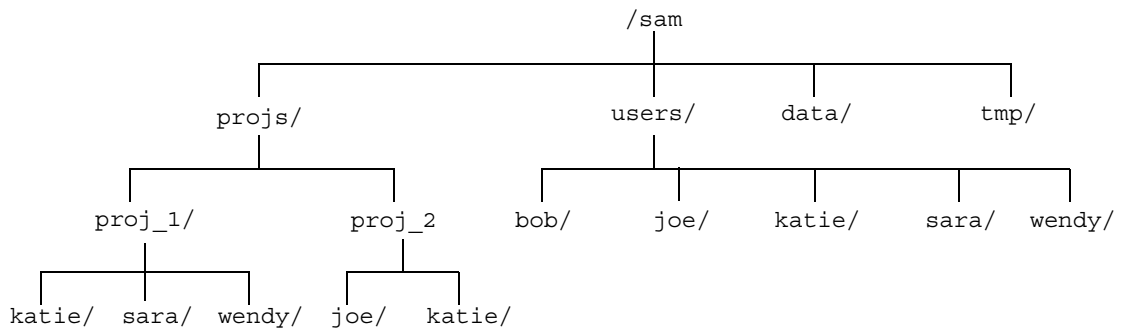


图 4-1 存档程序示例目录结构

示例 1

本示例介绍存档程序在无 `archiver.cmd` 文件可用时的活动。在本示例中，Sun SAM-FS 环境包括一个文件系统、配有两个驱动器的光学自动化库和六个卡盒。本示例显示的输出由下面的命令产生：

```
# archiver -lv
```

下面的输出表明存档程序选择的默认介质类型为 `mo`。只有 `mo` 介质可用：

```
Notify file: /opt/SUNWsamfs/sbin/archiver.sh

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

下面的输出表明存档程序使用两个驱动器。它列出了 12 个卷的存储容量和可用空间。

```
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-----
```


下面的输出表明元数据和数据文件均包括在存档组 `samfs` 中。当文件的存档时间到达默认的四分钟（240 秒）时，存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
```

下面的输出表明存档组中的文件按指定的顺序存档至卷中：

```
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 2

本示例说明如何将数据文件和元数据划分至两个不同的存档组。除了第 117 页的“示例 1”中所述的光学自动化库之外，文件系统环境还配备了手动安装 DLT 磁带驱动器。较大的文件存档至磁带，而较小的文件存档至光学卡盒。

下面介绍了示例 2 的 `archiver.cmd` 文件。该文件通过下面的命令来输出：

```
# archiver -lv -c example2.cmd
```

下面的输出部分是 archiver.cmd 文件的内容:

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

再次显示要使用的介质和驱动器，但没有添加 DLT 及其默认值的信息。

```
Notify file: /opt/SUNWsamfs/sbin/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-----
```

下面是文件系统的组织信息。大于 512000 字节 (500 KB) 的文件在四分钟后存档；其它所有文件在 30 秒后存档。

```
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
```

在下面的输出中，注意查看在可移动介质上划分存档组的情况：

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

示例 3

在本示例中，用户文件和项目数据文件存档至不同的介质。目录数据的文件按大小分别存储至光盘介质和磁带介质。分配至组标识 `pict` 的文件将被分配至另一个卷组。存档程序不对目录 `tmp` 和 `users/bob` 中的文件进行存档。存档程序每隔 15 分钟进行一次存档，并且保存存档记录。

代码示例 4-2 中显示的输出由下面的命令产生:

```
# archiver -lv -c example3.cmd
```

代码示例 4-2 archiver 命令输出

```
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]$
```

代码示例 4-2 archiver 命令输出 (续)

```
Reading archiver command file "example3.cmd"
41: prod_big.2 lt 0[5-6]$
42: endvsns

Notify file: /opt/SUNWsamfs/sbin/archiver.sh

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

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

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

Archive file selections:
Filesystem samfs Logfile: /var/opt/SUNWsamfs/archiver/log
samfs Metadata
copy:1 arch_age:240
no_archive Noarchive path:users/bob
prod_big path:data minsize:50.2k
copy:1 arch_age:60 unarch_age:2592000
copy:2 arch_age:180
prod path:data
copy:1 arch_age:60
proj_1 path:projs/proj_1
```

代码示例 4-2 archiver 命令输出 (续)

```
Reading archiver command file "example3.cmd"
  copy:1  arch_age:60
  copy:2  arch_age:60
joe  path:. uid:10006
  copy:1  arch_age:60
  copy:2  arch_age:60
pict path:. gid:8005
  copy:1  arch_age:60
  copy:2  arch_age:60
no_archive Noarchive path:tmp
samfs  path:.
  copy:1  arch_age:240

Archive sets:
allsets

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

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

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

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

prod.1
media: mo
Volumes:
  optic20
  optic21
  optic22
```

代码示例 4-2 archiver 命令输出 (续)

```
Reading archiver command file "example3.cmd"
  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 4

在本示例中，用户文件和项目数据文件存档至光学介质。请注意，代码示例 4-3 不使用图 4-1 中所示的目录结构。

本示例定义了四个 VSN 池；其中三个池分别用于用户池、数据池和项目池，另一个是暂用池。当 `proj_pool` 用尽介质时，它使用 `scratch_pool` 来保留卷。本示例介绍如何依据存档组组件、所有者组件和文件系统组件来为每一个存档组保留卷。存档程序每隔 10 分钟进行一次存档，并且保存存档日志。

代码示例 4-3 显示了 `archiver.cmd` 文件及存档程序输出。

代码示例 4-3 archiver.cmd 文件及存档程序输出

```
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
```

代码示例 4-3 archiver.cmd 文件及存档程序输出 (续)

```
Reading archiver command file "example4.cmd"
34: proj.1 mo -pool proj_pool -pool scratch_pool
35: endvsns

Notify file: /opt/SUNWsamfs/sbin/archiver.sh

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

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

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

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

代码示例 4-3 archiver.cmd 文件及存档程序输出 (续)

```
Reading archiver command file "example4.cmd"
proj_pool media: mo Volumes:
    optic12
    optic13
Total space available: 981.9M

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

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

Archive sets:
allsets

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

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

samfs.1
media: mo
```

代码示例 4-3 archiver.cmd 文件及存档程序输出 (续)

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

存档程序原则

存档程序使用 archiver.cmd 文件自动进行存储管理操作。编写此文件之前，请复习一些可以改善 Sun SAM-FS 或 Sun SAM-QFS 文件系统和存档程序性能的通用原则，这对您编辑文件非常有益，并可确保以最安全的方式存储您的数据。

每一个站点在计算应用、数据存储硬件和软件方面都是各不相同的。下面的建议是 Sun Microsystems 根据多年的经验总结出来的。为您的站点编写 archiver.cmd 文件时，请确保通过考虑以下方面来反映站点的数据存储要求。

1. 保存存档日志。存档日志提供了有关恢复数据的基本信息。即使在 Sun SAM-FS 或 Sun SAM-QFS 软件无法使用时，也可使用它进行数据恢复。我们建议您将这些日志保存在安全的地方，以防发生灾难性故障而造成 Sun SAM-FS 或 Sun SAM-QFS 软件无法使用。
2. 使用标准表达式指定卷。允许系统将文件存储在许多不同的卷上。通过标准表达式指定的卷范围可以使系统连续不断地运行。如果您为存档组副本指定特定的卷名，则会造成数据很快充满卷。因此，您不得不频繁地更换介质，从而导致不应有的工作流程问题。
3. 根据文件创建和修改的频率以及您是否需要保存从前保存的所有修改副本来设置您的存档时间间隔。请注意，存档时间间隔是指对文件系统执行扫描操作的时间间隔。如果将存档时间间隔设置得太短，则会使存档程序几乎不间断地执行扫描。

4. 考虑您要使用的文件系统数量。与单个 Sun SAM-FS 和 Sun SAM-QFS 文件系统相比，多个 Sun SAM-FS 或 Sun SAM-QFS 文件系统通常可以提高存档程序的性能。存档程序为每一个文件系统运行单独的进程。多个文件系统的扫描时间要比单个文件系统少得多。
5. 同 UNIX 文件系统一样，请使用目录结构来组织 Sun SAM-FS 或 Sun SAM-QFS 文件系统中的文件。出于性能考虑，Sun Microsystems 建议您不要将 10,000 个以上的文件放入同一个目录中。
6. 始终制作两份文件副本，并将它们存储至不同的卷上。如果将数据存储至同一个介质类型，则在介质出现物理问题时，您会面临数据丢失的风险。如果条件允许，请不要仅制作一份存档副本。
7. 务必定期运行 `samfsdump(1M)` 来转储您的元数据。元数据（包括目录结构和文件名等）存储在与文件系统同名的存档组中。在出现故障时，您可以使用此类信息来恢复文件系统。如果您不想执行此操作，可以通过将此存档组分配至不存在的 VSN，以防止存档此类数据。有关保存元数据的详细信息，请参阅《Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 故障恢复指南》或《Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 安装与配置指南》。

排除存档程序故障

存档程序执行任务的结果可能不尽人意，视最初的设置而定。请确保使用下列工具来监视系统的存档活动：

- `samu(1M)` 实用程序的 `a` 显示屏幕。此显示屏幕可以显示每一个文件系统的存档程序活动。另外，它还可以显示错误和警告消息，如下所示：

```
Errors in archiver commands - no archiving will be done
```

它将显示每一个文件系统的消息，其中包括存档程序何时再次扫描 `.inodes` 文件以及当前正在存档的文件。

- 存档日志。这些日志在 `archiver.cmd` 文件中进行定义，您应定期检查以确保将文件存档至卷中。存档日志可能会变得非常大，您应定期手动或使用 `cron(1)` 作业来减少其大小。为安全起见，请存档这些日志文件，因为这些信息能够恢复数据。
- `sfind(1)`。使用此命令定期检查未被存档的文件。如果发现未被存档的文件，请确保了解它们未被存档的原因。
- `s1s(1)`。除非文件存在有效的存档副本，否则系统不会释放该文件占用的磁盘空间。使用 `s1s -D` 命令可以显示文件的索引节点信息（包括副本信息）。

注 – 对于某个文件，`s1s -D` 命令的输出可能会显示 `archdone` 字样。这并不表示文件已具有存档副本。它只是表示存档程序已扫描该文件并且已完成所有与其自身相关的工作。只有通过查看 `s1s(1)` 命令显示的副本信息，才能确定文件是否具有存档副本。

有时，您可能会看到一些表明存档程序已用尽卡盒空间或卡盒不存在的消息，如下所述：

- 当存档程序未发现分配给存档组的卡盒时，会发出下面的消息：

```
No volumes available for Archive Set setname
```

- 当存档程序发现分配给存档组的卡盒无可用空间时，会发出下面的消息：

```
No space available on Archive Set setname
```

`archiver.sh` 脚本将在 `/var/opt/SUNWsamfs/archiver` 下创建两个目录来分别处理无可用空间或存档组无可用卷的例外情况。这两个目录的名称分别是 `NoSpace` 和 `NoVSNs`，它们列出了大小为零的文件以及相应存档组的名称。在特定的条件消失后，您应删除这些文件。有关详细信息，请参阅 `archiver.sh(4)` 联机资料。

文件未被存档的原因

下面的核对表列出了 Sun SAM-FS 或 Sun SAM-QFS 环境中的文件未被存档的原因。

1. `archiver.cmd` 文件存在语法错误。运行 `archiver -lv` 命令查找错误，然后更正标有错误标记的行。
2. `archiver.cmd` 文件中存在 `wait` 指令。删除此 `wait` 指令或运行 `samu(1M)` 实用程序的 `:arrun` 命令来取代该指令。
3. 无可用的卷。使用 `archiver -lv` 命令也可显示这种情况。根据需要添加更多的卷。您可能需要导出现有的卡盒以腾出自动化库中的端口。
4. 存档组的卷已满。您可以导出卡盒并导入新的替换卡盒（确保已标记新卡盒），也可对卡盒执行回收操作。有关回收的详细信息，请参阅第 165 页的“回收”。
5. `archiver.cmd` 文件中的 VSN 部分列出的介质不正确。检查标准表达式和 VSN 池，确保正确定义了它们。

6. 可用卷上无足够的空间来存档任何文件。如果您的文件较大，并且卷似乎已满，则卡盒的利用率可能已达到 Sun SAM-FS 或 Sun SAM-QFS 环境规定的界限。如果出现这些情况，请添加新卡盒或执行回收操作。

如果您已指定 `-join path` 参数，并且无足够的空间将目录中的所有文件存档至任何卷，则不会进行存档。您应添加卡盒，执行回收操作，或者使用 `-sort path` 参数。有关这些参数的详细信息，请参阅第 104 页的“联合存档”。

7. `archiver.cmd` 文件为包含大容量文件的目录或文件系统设置了 `no_archive` 指令。
8. 使用 `archive -n`（永不存档）命令设置了太多的目录，导致其中的文件无法存档。
9. 大容量文件使用频繁。因此，它们永远达不到存档时限而无法存档。
10. 自动化库存在硬件问题或配置问题。
11. 客户机与服务器之间的网络连接有问题。确保客户机和服务器之间已建立稳定的通信。

其它存档程序诊断方法

在排除存档程序的故障时，除检查上表列出的项目之外，还应检查以下项目。

1. `syslog` 文件（默认位于 `/var/adm/sam-log` 目录）。此文件包含的存档程序消息可以指出问题的根源。
2. 卷容量。确保所有必需的卷可用，并且它们具有足够的存档空间。
3. 如果存档程序产生过多的原因不明的卡盒活动，或似乎未执行任何活动，请打开跟踪工具并检查跟踪文件。有关跟踪文件的详细信息，请参阅 `defaults.conf(4)` 联机资料。
4. 您可以对存档程序进程 (`sam-archiverd`) 运行 `truss -p pid` 命令，以确定不正确的系统调用。有关 `truss(1)` 命令的详细信息，请参阅 `truss(1)` 联机资料。
5. `showqueue(1M)` 命令可以显示存档程序队列文件的内容。您可以使用此命令来检查正在安排或存档的存档请求的状态。任何无法安排的存档请求均会生成一则说明原因的消息。另外，此命令还可以显示存档的过程。

文件未被释放的原因

存档程序和释放程序相互合作来协调磁盘高速缓存中的可用数据量。未从磁盘高速缓存自动释放文件的主要原因是它们尚未存档。

有关文件未被释放的详细信息，请参阅第 151 页的“排除释放程序的故障”。

释放

释放是指释放程序通过识别已存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用的过程。这可以为其它从存档介质中创建或登台的文件腾出空间。释放程序只释放已存档的文件。释放文件后，系统会从磁盘高速缓存中清除该文件的任何数据。

当达到站点指定的磁盘阈值时，Sun SAM-FS 和 Sun SAM-QFS 文件系统会启动释放程序进程。与释放程序不同，`release(1)` 命令可以使用户立即释放文件占用的磁盘空间或为文件设置释放参数。有关释放程序进程的详细信息，请参阅 `sam-releaser(1M)` 联机资料。

释放程序所包含的功能允许您指定哪些文件在存档之后立即释放、哪些文件永不释放以及哪些文件可以部分释放。由于某些实用程序（如 `filemgr(1)`）只读取文件的起始部分，因此部分释放功能特别有用。采用部分释放功能时，文件的一部分保留在磁盘高速缓存中，而文件的剩余部分会被释放。读取仍保留在磁盘高速缓存中的文件的第一部分时，并不会导致系统从存档介质中将文件的剩余部分重新登台至磁盘高速缓存。本章将介绍上述功能和其它一些功能。

本章包括下列主题：

- 第 138 页的“释放程序概述”
- 第 138 页的“操作原理”
- 第 139 页的“定义”
- 第 140 页的“部分释放和部分登台”
- 第 142 页的“`releaser.cmd` 文件”
- 第 148 页的“`archiver.cmd` 文件在登台过程中的作用”
- 第 149 页的“配置释放程序”
- 第 150 页的“手动运行释放程序”
- 第 151 页的“排除释放程序的故障”

释放程序概述

当文件系统的利用率超过配置的上限时，文件系统管理软件将启动释放程序。首先，释放程序读取 `releaser.cmd` 文件并收集用于控制释放进程的指令。其次，它扫描文件系统并收集每一个文件的有关信息。最后，在扫描整个文件系统后，释放程序开始按优先级顺序释放文件。

只要文件系统的利用率高于配置的下限，释放程序就会继续释放文件。通常，释放程序会释放足够的空间以使文件系统的利用率低于配置的下限。如果释放程序没有发现任何需要释放的文件，系统会强制退出释放程序。以后，当更多的文件可以释放时，释放程序即会运行。当高于上限时，文件系统会每隔一分钟启动一次释放程序。

上限和下限通过文件系统的安装选项 `high=percent` 和 `low=percent` 进行设置。有关这些安装选项的详细信息，请参阅 `mount_samfs(1M)` 联机资料。

操作原理

文件系统可以包含成千上万个文件。由于只需释放几个大文件便有可能使文件系统的利用率降至下限，因此没有必要记录所有文件的释放优先级。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。释放程序通过只确定前 10,000 个备选文件来解决这一问题。

确定前 10,000 个备选文件之后，如果随后的备选文件优先级不高于前 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，释放程序便会检查一次文件系统的高速缓存利用率是否低于下限。如果是，释放程序将停止释放文件。如果否，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，释放程序将重新确定 10,000 个新备选文件。

如果找不到任何适合的备选文件，释放程序会退出。例如，在文件没有存档副本时，就会出现这种情况。一分钟后，Sun SAM-FS 和 Sun SAM-QFS 文件系统会再次启动释放程序。

定义

表 5-1 介绍了本章中使用的术语。

表 5-1 释放程序术语

术语	定义
时段	<p>给定事件从发生到现在所持续的时间。文件的索引节点将记录释放程序所用的下列时间：驻留更改时间、数据修改时间和数据访问时间。</p> <p>您可以使用 <code>sls(1)</code> 命令的 <code>-D</code> 选项来查看这些时间。每一种时间都有对应的时段。例如，如果当前时间是上午 10 点 15 分，则在上午 10 点 10 分更改的文件的数据修改时段为 5 分钟。有关 <code>sls(1)</code> 命令的详细信息，请参阅 <code>sls(1)</code> 联机资料。</p>
备选文件	<p>适合释放的文件。某个文件不是备选文件的原因包括：</p> <ul style="list-style-type: none">• 该文件已脱机。• 该文件尚未存档。• <code>archiver.cmd</code> 命令文件为该文件指定了 <code>-norelease</code> 属性，并且尚未为该文件创建完要求的副本份数。• 该文件标记为“已损坏”。• 该文件不是标准文件。它是目录文件、块文件、特殊字符文件或管道文件。• 存档程序正在登台该文件以创建另一副本。登台之后，该文件便成为适合释放的文件。• 该文件的时段为负数。这种情况通常会发生在未正确设置时钟的 NFS 客户机上。• 该文件被标记“<code>release -n</code>”。• 该文件在过去登台的时间小于最短驻留时间设置。有关详细信息，请参阅第 146 页的“指定最短驻留时间”。• 已用 <code>release(1)</code> 命令的 <code>-p</code> 选项将该文件标记为“部分释放”，并且释放程序已部分释放该文件。• 该文件太小。
优先级	<p>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数。总优先级是以下两种优先级之和：时段优先级和大小优先级。</p> <p>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。</p>
权数	<p>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）的数值。例如，如果将文件的大小权数设置为零，则在计算优先级时，不会考虑文件的大小属性。权数是介于 0.0 和 1.0 之间的浮点值。</p>
部分释放	<p>通过指定在磁盘高速缓存中只保留文件的起始部分而释放文件的剩余部分，可以部分释放文件。例如，在使用 <code>filemgr(1)</code> 等实用程序读取文件的起始部分时，部分释放功能非常有益。</p>

部分释放和部分登台

释放和登台是两个互为补充的进程。站点在存档文件后，可以从联机磁盘高速缓存中完全释放文件，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。这一部分释放文件的功能可以使系统在不登台文件的情况下立即访问文件存根中的数据。

系统管理员可以在安装文件系统时指定默认部分释放大小和保持联机的存根的最大值。系统管理员可以在 `mount(1M)` 命令中设置这些大小，如下所述：

- `-o partial=n` 选项可将保持联机的文件存根设置为默认大小 (n)。 `-o partial=n` 设置必须小于或等于 `-o maxpartial=n` 设置。最小的设置为 `-o partial=8 KB`。默认设置为 `-o partial=16 KB`。
- `-o maxpartial=n` 选项可将保持联机的文件存根设置为最大值 (n)。若要限制可以保持联机的文件存根的大小，请使用 `-o maxpartial=n` 选项，并将其值指定为可以保持联机的最大存根大小。若要禁用部分释放功能，请指定 `-o maxpartial=0`。

通过在 `release(1)` 命令中指定 `-p` 选项或在 `sam_release(3)` 库例程中指定 `p` 选项，用户可以指定文件的默认存根大小。要为多种类型的文件或不同的应用程序指定不同大小的文件存根，用户可以在 `release(1)` 命令中指定 `-s` 选项或在 `sam_release(3)` 库例程中指定 `s` 选项。`-s` 和 `s` 的值必须小于您在使用 `mount(1M)` 命令安装文件系统时指定的 `-o maxpartial` 值。

另一个安装选项，即 `-o partial_stage=n`，可以使系统管理员确定在登台文件的剩余部分之前应读取部分释放存根的数据量。也就是说，读完 `-o partial_stage=n` 大小规格之后开始登台文件。

默认情况下，系统将 `-o partial_stage=n` 选项设置为部分释放存根的大小。虽然用户可以配置该值，但应注意该值对文件登台的影响，如下所述：

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

示例：假定您设置了下列选项：

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

filemgr(1) 是所用的程序，它首先读取文件的前 8 KB。此时，系统不会登台文件。video-on-demand 程序读取同一个文件，并且当它读完文件的前 16 KB 时，系统开始登台文件。在安装并定位存档磁带后，该应用程序会继续读取 2G 的磁盘数据。当 video-on-demand 程序读完 2G 的文件数据后，它会在登台活动之后立即进行读取。由于在该应用程序读取部分文件数据时已经安装并定位了磁带，因此它不必等待。

数个命令行选项将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性。

系统管理员选项概述

系统管理员可以在安装文件系统时更改部分释放的最大值和默认值。表 5-2 中列出了影响部分释放的 mount(1M) 选项。有关 mount(1) 命令的详细信息，请参阅 mount_samfs(1M) 联机资料。

表 5-2 影响部分释放的安装选项

mount(1M) 选项	作用
-o maxpartial= <i>n</i>	确定在文件标记为部分释放时，可以在联机磁盘高速缓存中保留的最大空间 (KB)。最大值为 2,097,152 KB，即 2G。最小值为 0，即不允许部分释放任何文件。 如果指定 -o maxpartial=0，则会禁用部分释放功能。所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。一旦安装文件系统，用户便不能改写此选项指定的值。 默认情况下，系统将 <i>n</i> 变量设置为 16。此设置可以使用户将文件标记为部分释放，并将文件保留在磁盘中的最大空间设置为 16 KB。
-o partial= <i>n</i>	当用户使用 release(1) 命令的 -p 选项将文件标记为部分释放时，该选项用于设置要在磁盘高速缓存中保留的默认空间 (KB)。 <i>n</i> 变量的值至少为 8，但它可以与您为 -o maxpartial= <i>n</i> 选项指定的值一样大。 由于某些应用程序不必访问整个文件便可完成其工作，因此该选项可用于确保应用程序能够从文件的起始部分获得所需的信息。同时，使用此选项还可防止系统登台不必要的文件。 默认值为 -o partial=16。
-o partial_stage= <i>n</i>	指定在访问部分释放的文件时，从存档介质登台整个文件之前应读取 <i>n</i> 字节的文件数据。为此选项设置的值通常小于 -o partial 的设置值。 <i>n</i> 指定介于 0 和 -o maxpartial 规格之间的整数。默认情况下，系统将它设置为 16 或您为 -o partial 选项指定的任何值。
-o stage_n_window= <i>n</i>	将一次可以登台的数据量指定为 <i>n</i> 。 <i>n</i> 指定介于 64 和 2,048,000 之间的整数。默认值为 256 KB。此选项仅适用于已设置 stage -n 属性的文件。

用户选项概述

系统管理员可以设置文件在释放后可以保留在磁盘高速缓存中的文件存根的最大值和默认值。此外，系统管理员还可以确定是否为特定的文件系统启用部分释放功能。

不过，通过使用 `release(1)` 命令和 `sam_release(3)` 库例程，用户可以设置其它释放属性以及指定要标记为部分释放的文件。表 5-3 中列出了可以确定部分释放属性的命令和库选项。有关 `release(1)` 命令的详细信息，请参阅 `release(1)` 联机资料。有关 `sam_release(3)` 库例程的详细信息，请参阅 `sam_release(3)` 联机资料。

表 5-3 用户释放选项

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

releaser.cmd 文件

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

以下几节介绍了 `releaser.cmd` 文件中的指令：

- 第 143 页的“指定与时段和大小相关的释放优先级指令”
- 第 145 页的“指定用于单个文件系统的指令”
- 第 146 页的“指定调试指令”
- 第 146 页的“指定最短驻留时间”
- 第 146 页的“指定日志文件”
- 第 148 页的“限制释放已存档的文件”

有关这些指令的详细信息，请参阅 `releaser.cmd(4)` 联机资料。

指定与时段和大小相关的释放优先级指令

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

有关释放程序指令的其它信息，请参阅 `releaser.cmd(4)` 联机资料。

文件时段

释放程序在确定与时段相关的文件释放优先级要素时将考虑下列可能的时段：

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

在某些情况下，您可能希望文件的访问时段优先于修改时段。而在其它情况下，可能优
先考虑由最近访问时间、修改时间和驻留状态更改时间得出的简单时段。

默认情况下，文件时段是指下列三个文件时段中最小的一个：

- 文件访问时段
- 文件修改时段
- 文件驻留时段

您可以使用指令指定在计算文件的释放优先级时要使用的加权时段优先级。

这些时段优先级指令的格式如下：

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

- `weight_age` 指令指定要为其分配加权因子的文件默认时段（文件访问时段、修改时
段和驻留时段三者中的较小者）。`float` 指定处于以下范围内的浮点数：
 $0.0 \leq float \leq 1.0$ 。默认情况下，`float = 1.0`。

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

- `weight_age_residence`、`weight_age_modify` 和 `weight_age_access` 指令
指定由这些可能时段（一个、二个或三个）的组合确定的文件时段。`float` 指定处于以
下范围内的浮点数： $0.0 \leq float \leq 1.0$ 。默认情况下，`float = 1.0`。

这些指令不能与 `weight_age` 指令结合使用。

如果使用 `weight_age_residence`、`weight_age_modify` 和 `weight_age_access` 指令，则依据这三个时段的组合来计算文件的与时段相关的优先级。首先，收集每一个文件的可能时段数据。其次，将文件时段数据与 `releaser.cmd` 文件中指定的加权因子相乘。最后，通过将时段数据乘以每一个加权因子的结果相加（如下所示），计算文件的与时段相关的优先级：

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

示例：`releaser.cmd` 文件中的下列行指定在计算文件的释放优先级时，只考虑文件的驻留时段（而忽略修改时段和访问时段）：

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

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

文件大小

释放程序在确定与大小相关的文件释放优先级要素时将考虑文件的大小。文件的大小（在 4 KB 块中）乘以您为 `weight_size` 指令指定的权数，得出与大小相关的文件释放优先级要素。

`weight_size` 指令的格式如下所示：

```
weight_size = float
```

`float` 指定处于以下范围内的浮点数： $0.0 \leq float \leq 1.0$ 。默认情况下，`float = 1.0`。

示例：下面的 `releaser.cmd` 文件指定在计算文件的释放优先级时，将忽略 `samfs1` 和 `samfs2` 文件系统中所有文件的大小。

```
# releaser.cmd file
logfile = /var/adm/default.releaser.log
weight_size = 0.0
#
fs = samfs1
weight_age = 1.0
logfile = /var/adm/samfs1.releaser.log
#
fs = samfs2
weight_age_modify = 0.3
weight_age_access = 0.03
weight_age_residence = 1.0
logfile = /var/adm/samfs2.releaser.log
```

指定用于单个文件系统的指令

您可以使用 `releaser.cmd` 文件中的 `fs = family_set_name` 指令来指明 `fs =` 指令后面的指令仅应用于指定的文件。该指令的格式如下：

```
fs = family_set_name
```

family_set_name 指定 `mcf` 文件中的系列集名称。

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

`releaser.cmd(4)` 联机资料中包含 `fs =` 指令的示例。

指定调试指令

在调节或调试释放程序时，`no_release` 和 `display_all_candidates` 指令非常有用。下面介绍了这两种指令：

- `no_release` 指令可以防止文件从联机磁盘高速缓存中删除。您可以在不真正释放文件的情况下，使用此指令来检查 `releaser.cmd` 文件中的指令。该指令的格式如下：

```
no_release
```

- `display_all_candidates` 指令可以将所有释放备选文件的名称写入至日志文件。该指令的格式如下：

```
display_all_candidates
```

由于释放程序可将释放备选文件的名称写入至日志文件，而不是真正从文件系统中释放这些文件，因此这些指令对调试大有帮助。

指定最短驻留时间

`min_residence_age` 指令可以使您指定文件在成为释放备选文件之前必须在文件系统中驻留的最短时间。该指令的格式如下：

```
min_residence_age = time
```

对于 *time*，指定驻留时间，以秒为单位。默认时间为 600 秒，即 10 分钟。最短或最长 *time* 设置并没有具体规定。

指定日志文件

如果在 `releaser.cmd` 文件中指定了 `logfile` 指令，释放程序会将其活动添加至指定的文件名，或创建文件名（如果不存在文件名）。该指令的格式如下：

```
logfile = filename
```

对于 *filename*，指定日志文件的名称。

下面是一个日志文件范例（请注意，为适应页宽，某些行已被换行）：

```
Releaser begins at Wed Apr 28 17:29:06 1999
inode pathname      /sam1/.inodes
low-water mark      24%
weight_size         1
weight_age          1
fs equipment ordinal 1
family-set name     samfs1
started by sam-initd? yes
release files?      yes
display_all_candidates? no
---before scan---
blocks_now_free:    3481504
lwm_blocks:         3729362
---scanning---
10501 (R: Wed Apr 21 18:47:50 CDT 1999) 10001 min, 500 blks /sam1/testdir0/filevp
10500 (R: Wed Apr 21 18:48:10 CDT 1999) 10000 min, 500 blks /sam1/testdir0/filewq
...
---after scan---
blocks_now_free:    3730736
lwm_blocks:         3729362
archnodrop: 0
already_offline: 0
bad_inode_number: 0
damaged: 0
extension_inode: 0
negative_age: 0
nodrop: 1
not_regular: 9
number_in_list: 675
released_files: 202
too_new_residence_time: 0
too_small: 2
total_candidates: 675
total_inodes: 1376
wrong_inode_number: 0
zero_arch_status: 689
zero_inode_number: 0
zero_mode: 0
CPU time: 2 seconds.
Elapsed time: 10 seconds.
Releaser ends at Wed Apr 28 17:29:16 1999
```

releaser(1M) 联机资料介绍了日志文件中包含的信息。随着释放程序的每一次运行，日志文件的大小会不断增加，因此，请务必注意减少日志文件的大小，或删除 logfile 关键字。

下面的数学关系式位于 ---after scan--- 行下面显示的统计项中：

```
total_inodes = wrong_inode_number +
zero_inode_number +
zero_mode +
not_regular +
extension_inode +
zero_arch_status +
already_offline +
damaged +
nodrop +
archnodrop +
too_new_residence_time +
too_small +
negative_age +
total_candidates
released_files = total_candidates
```

限制释放已存档的文件

默认情况下，系统会释放标记为重新存档的文件。如果在 releaser.cmd(4) 文件中指定 `research_no_release` 指令，则不会释放标记为重新存档的文件。该指令的格式如下：

```
research_no_release
```

archiver.cmd 文件在登台过程中的作用

archiver.cmd 文件中的大多数指令会影响存档过程，但存档组分配指令可以使您指定应用于存档组中所有文件的释放属性。

存档组分配指令的格式如下：

```
archive_set_name path [search_criteria ...] directives ...
```

表 5-4 显示了与释放有关的 *directives*（指令）。

表 5-4 存档组分配 *directives*（指令）

指令	作用
-release a	指定释放程序应在创建存档组中文件的第一个副本后释放文件。如果您需要为每一个文件创建多份存档副本，请勿使用此选项。在此情况下，系统会登台第一个副本以创建第二个副本。
-release n	指定存档组中的文件永不释放。
-release p	指定释放程序应在存档文件后部分释放存档组中的文件。

有关这些及其它 `archiver.cmd` 指令的详细信息，请参阅第 73 页的“存档”。

配置释放程序

您有必要为您的站点确定高速缓存中文件的特性。在只需登台很小的文件时载入磁带是很不经济的，因此您可能更愿意让您的系统在高速缓存中保留少量的文件。为使释放程序首先释放最大的文件，请在 `releaser.cmd` 文件中使用下面的指令：

```
weight_size = 1.0
weight_age = 0.0
```

此外，您可能希望将最近修改过的文件保留在高速缓存中，因为您不久可能会重新修改它们。这可以避免因登台文件以进行修改时产生的开销。在此情况下，请使用第二组时段权数。要使释放程序严格按从最早修改到最近修改的顺序加权文件，请在 `releaser.cmd` 文件中使用下面的指令：

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

不过，大多数情况并不采用这种简单直接的方式，如下面的示例所述。

示例 1：假定您希望首先释放最大的文件。文件系统中包括成百上千个大小相同的小文件和数个大文件。小文件的大小之和可能超过一个最大文件的大小。最终，释放程序将释放所有大文件。如果指定 `weight_age = 0.0`，释放程序基本上按任意顺序释放小文件，因为它们的大小相同，并且具有相同的释放优先级。

在此方案中，您可以设置 `weight_age = 0.01` 以作为一种应急的选择方法。此时，对于大小相等的两个文件，释放程序会首先释放较旧的文件。

示例 2: 本示例介绍了一种指定如何首先释放最大文件的更好方法。

设置 `weight_size = 1.0` 和 `weight_age = 0.01`。

这两个指令通过优先选择较小的不经常访问的文件（而不是较大的经常访问的文件）作为释放备选文件，因而它们违背了首先释放最大文件的原則。通过使 `weight_age` 的值更小于 `weight_size` 的值，您可以将这一影响减少到您所需的程度。例如，根据上面的设置，已登台 100 分钟的 4 KB 文件与刚刚登台的 8 KB 文件具有相同的释放优先级。

此时，释放程序会选择释放其中任何一个文件。如果它选择 4 KB 的文件，则违背了首先释放最大文件的意向。为了降低这一影响，请将 `weight_age` 设置为更小的值，例如 0.001。如果 4 KB 文件登台的时间为 1,000 分钟，则它与刚刚登台的 8 KB 文件具有相同的优先级。

您可以使用 `no_release` 和 `display_all_candidates` 指令以及手动运行释放程序以获得按优先级顺序排列的备选文件列表，以便根据该表来调整优先级权数。

手动运行释放程序

有时，您可能需要手动运行释放程序。若要进行此项操作，您需要知道文件系统的安装点以及释放程序试图达到的下限。

例如，在 `/sam1` 文件系统的充满程度达到 47% 时开始释放其中的文件。以超级用户 (`root`) 的身份登录并输入下面所示的行：

```
# /opt/SUNWsamfs/sbin/sam-releaser /sam1 47 1.0
```

最终的变量 `weight-size` 将由 `releaser.cmd` 文件中的 `weight_size` 命令改写。

当释放程序运行时，它将在屏幕上显示有关信息并写入至释放程序日志文件（如果已在 `releaser.cmd` 文件中指定）。

排除释放程序的故障

释放程序未能释放文件的原因有许多种。下面列出了一些可能的原因：

- 只有在存档文件后才能释放文件。文件可能没有存档副本。有关此主题的详细信息，请参阅第 134 页的“文件未被存档的原因”。
- 存档程序请求不要释放文件。这可能会发生在下列情况下：
 - 存档程序刚刚登台某个脱机文件以创建另一副本。
 - 已在 `archiver.cmd` 文件中设置 `-norelease` 指令，并且所有标记为 `-norelease` 的副本尚未存档。请注意，释放程序汇总输出显示了设定 `archnodrop` 标记的文件的总数。
- 文件已设置为部分释放，但文件的大小等于或小于向上圆整至磁盘分配单元 (DAU) 大小（块大小）的部分释放大小。
- 文件在最后的 `min_residence_age` 分钟内更改了驻留状态。
- 已使用 `release -n` 命令以防止释放目录和文件。
- `archiver.cmd` 文件中为太多的目录和文件设置了 `-release n` 选项。
- 释放程序上限设置得太高，而造成自动释放过程发生得太迟。在 `samu(1M)` 实用程序的 `m` 显示屏幕中或使用 `libmgr(1M)` 命令验证这一情况，并根据需要减少该值。
- 释放程序下限设置得太高，而造成自动释放过程停止得太早。在 `samu(1M)` 实用程序的 `m` 显示屏幕中或使用 `libmgr(1M)` 命令验证这一情况，并根据需要减少该值。
- 大容量文件使用频繁。它们永远不会达到存档时限，永远不会存档，因此也永远不会释放。

登台

登台是将文件数据从近线或离线存储设备复制到在线存储设备的过程。登台功能可以使您立即登台文件、永不登台文件、指定部分登台以及指定其它登台操作。例如，永不登台功能可供随机从大文件访问一小段记录的应用程序使用；启用该功能时，系统不需将文件登台为联机文件便可直接从存档介质中访问数据。

本章介绍 Sun SAM-FS 和 Sun SAM-QFS 文件登台功能。它包括下列主题：

- 第 153 页的“`stager.cmd` 文件”
- 第 158 页的“`archiver.cmd` 文件在登台过程中的作用”
- 第 158 页的“使用 `preview.cmd` 文件对预备请求进行排序”
- 第 161 页的“计算预备请求的总优先级”
- 第 161 页的“如何设置预备请求的优先级方案”

`stager.cmd` 文件

您可以使用 `stager.cmd` 文件指定登台程序的操作。该文件的完整路径名为 `/etc/opt/SUNWsamfs/stager.cmd`。默认情况下，登台程序将执行下列操作：

- 登台程序尝试使用库中的所有驱动器来登台文件。
- 登台缓冲器大小由介质类型决定，并且不锁定登台缓冲器。
- 不写日志文件。
- 一次最多可以激活 1000 个登台请求。

您可以使用 `stager.cmd` 文件指定有关指令来改写这些默认操作。本节的以后部分将介绍登台程序的指令。有关登台程序指令的其它信息，请参阅 `stager.cmd(4)` 联机资料。

第 157 页的“`stager.cmd` 文件示例”显示了编写完毕的已设置所有可能指令的 `stager.cmd` 文件。

本节中的示例假定已在 `mcf` 文件中进行了以下配置：

```
#
# Sun SAM-FS file system configuration example
#
# Equipment      Eq Eq Family Dev Additional
# Identifier     Or Tp Set  St  Parameters
# -----
samfs1           60 ms samfs1
/dev/dsk/c1t1d0s6 61 md samfs1 on
/dev/dsk/c2t1d0s6 62 md samfs1 on
/dev/dsk/c3t1d0s6 63 md samfs1 on
/dev/dsk/c4t1d0s6 64 md samfs1 on
/dev/dsk/c5t1d0s6 65 md samfs1 on
#
samfs2           2 ms samfs2
/dev/dsk/c1t1d0s0 15 md samfs2 on
/dev/dsk/c1t0d0s1 16 md samfs2 on
#
/dev/samst/c0t2d0 20 od -      on
/dev/samst/c1t2u0 30 rb dog  on /var/opt/SUNWsamfs/catalog/dogcat
/dev/samst/c1t5u0 31 od dog  on
/dev/samst/c1t6u0 32 od dog  on
/dev/rmt/0cbn     40 od -      on
/dev/samst/c1t3u1 50 rb bird on /var/opt/SUNWsamfs/catalog/birdcat
/dev/rmt/2cbn     51 tp bird  on
```

指定驱动器数量

默认情况下，登台程序在登台文件时使用所有可用的驱动器。如果登台程序使所有驱动器处于繁忙状态，这会影响存档程序的活动。`drives` 指令用于指定登台程序可用的驱动器数量。该指令的格式如下：

```
drives = library count
```

其中

library 库的系列集名，与它在 Sun SAM-FS 或 Sun SAM-QFS `mcf` 文件中的系列集名称相同。

count 可用驱动器的最大数量。默认情况下，此数量与该库在 `mcf` 文件中配置的驱动器数量相同。

例如，下面的指令行表示只将 dog 系列集库中的一个驱动器用于登台文件：

```
drives = dog 1
```

有关 mcf 文件的详细信息，请参阅 mcf(4) 联机资料。

设置登台缓冲器大小

默认情况下，登台程序在将文件从存档介质恢复至联机磁盘高速缓存之前，首先将需要登台的文件读入缓冲器中的内存。您可以使用 `bufsize` 指令指定非默认的缓冲器大小和（可选）锁定缓冲器。这些操作可以改善系统的性能。您可以试用不同的 `buffer_size` 值来确定最适合的缓冲器大小。该指令的格式如下：

```
bufsize=media buffer_size [ lock ]
```

其中：

- media** 指定 mcf(4) 联机资料中列出的存档介质类型。
- buffer_size** 指定介于 2 至 32 之间的数字（包括 2 和 32）。默认值为 4。此值乘以相应介质类型的 `dev_blksize` 值等于所用的缓冲器大小。您可以在 `defaults.conf` 文件中指定 `dev_blksize` 的值。为 `buffer_size` 指定的数值越大，可用的内存就越多。有关此文件的详细信息，请参阅 `defaults.conf(4)` 联机资料。
- lock** `lock` 变量指明存档程序在登台存档副本时是否使用锁定的缓冲器。如果指定 `lock`，存档程序将在复制操作期间在内存中的存档缓冲器上设置文件锁定。这可以避免由于为每一个 I/O 请求锁定和取消锁定缓冲器而造成的开销，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` 变量。如果内存不足，则可能会造成内存用尽的情况。
- 只有已为需要存档的文件启用直接 I/O 时，`lock` 变量才有效。默认情况下，不会指定 `lock` 变量，并且文件系统会在所有直接 I/O 缓冲器上设置锁定（包括用于存档的缓冲器）。有关启用直接 I/O 的详细信息，请参阅 `setfa(1)` 联机资料、`sam_setfa(3)` 库例程联机资料或 `mount_samfs(1M)` 联机资料中介绍的 `-O forcedirectio` 选项。

例如，您可以按以下方式在 `stager.cmd` 文件的指令行中指定该指令：

```
bufsize=od 8 lock
```

指定日志文件

您可以请求 Sun SAM-FS 或 Sun SAM-QFS 文件系统收集文件登台事件的信息，并将其写入至日志文件。logfile 指令用于指定登台程序可在其中写入记录信息的日志文件。该指令的格式如下：

```
logfile=filename
```

filename 指定完整的路径名。

当指定日志文件时，登台程序会将每一个登台文件的有关信息写入至日志文件中的行。该行中包括文件名、登台日期和时间以及 VSN 等信息。例如，下面的指令行指定了文件 /var/adm/stage.log：

```
logfile=/var/adm/stage.log
```

代码示例 6-1 显示了登台程序日志文件的示例。

代码示例 6-1 登台程序日志文件示例

```
S 10/24 09:30:51 mo opt02b 29405.fa7d 24.47 4699763 /saml/testdir0/filec
S 10/24 09:30:52 mo opt02b 29405.11e5a 25.47 1452980 /saml/testdir0/fileb
S 10/24 09:30:58 mo opt02b 29405.12971 26.47 4194084 /saml/testdir0/filea
S 10/24 09:31:04 mo opt02b 29405.2 13.47 4121178 /saml/testdir0/filel
S 10/24 09:31:08 mo opt02b 29405.1f75 14.47 2532411 /saml/testdir0/filek
S 10/24 09:31:11 mo opt02b 29405.32c9 15.47 2919620 /saml/testdir0/filej
S 10/24 09:31:16 mo opt02b 29405.4911 16.47 4173923 /saml/testdir0/filei
S 10/24 09:31:21 mo opt02b 29405.68eb 17.47 4714187 /saml/testdir0/fileh
S 10/24 09:31:24 mo opt02b 29405.8ce4 19.47 2595485 /saml/testdir0/fileg
S 10/24 09:31:28 mo opt02b 29405.a0b3 20.47 3952040 /saml/testdir0/filef
S 10/24 09:31:33 mo opt02b 29405.bedb 21.47 4344648 /saml/testdir0/filee
S 10/24 09:31:37 mo opt02b 29405.e002 22.47 3470154 /saml/testdir0/filed
```

如代码示例 6-1 所示，登台程序日志文件由一些分成九个字段的信息行组成。表 6-1 说明了登台程序日志文件中各个字段的内容。

表 6-1 登台程序日志文件字段

字段	内容说明
----	------

- | | |
|---|----------------------------------|
| 1 | 登台活动。S 表示已经登台。C 表示取消登台。E 表示登台出错。 |
| 2 | 登台活动的日期，格式为 <i>mm/dd</i> 。 |
| 3 | 登台活动的时间，格式为 <i>hh:mm:ss</i> 。 |

表 6-1 登台程序日志文件字段 (续)

字段	内容说明
4	存档介质类型。有关介质类型的详细信息，请参阅 mcf(4) 联机资料。
5	卷序列名。
6	存档文件 (tar(1) 文件) 在介质上的起始物理位置和存档文件中的文件偏置 (采用十六进制表示)。
7	索引节点编号和世代编号。世代编号是在索引编号被重新使用后生成的一个附加编号，它与索引编号一起用来标识使用的唯一性。
8	文件的大小。
9	文件的名称。

指定登台请求的数量

您可以使用 `maxactive` 指令来指定一次可以执行的登台请求的数量。该指令的格式如下：

```
maxactive=number
```

默认情况下，`number` 为 1000。允许的最小数量为 1。

例如，下面的指令行指定队列中最多可以同时存在 500 个登台请求。

```
maxactive=500
```

stager.cmd 文件示例

下面是一个 `stager.cmd` 文件示例：

```
# This is stager.cmd file /etc/opt/SUNWsamfs/stager.cmd
drives=dog 1
bufsize=od 8 lock
logfile=/var/adm/stage.log
maxactive=500
```

archiver.cmd 文件在登台过程中的作用

archiver.cmd 文件中的大多数指令会影响存档过程，但存档组分配指令可以使您指定应用于存档组中所有文件的登台属性。存档组分配指令的格式如下：

```
archive_set_name path [search_criteria ...] directives ... ]
```

表 6-2 显示了与登台有关的 *directives* (指令)。

表 6-2 登台 *directives* (指令)

指令	作用
-stage a	指定存档组中的文件应联合登台。
-stage n	指定存档组中的文件永不登台。

有关这些及其它 archiver.cmd 指令的详细信息，请参阅第 73 页的“存档”。

使用 preview.cmd 文件对预备请求进行排序

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

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

预备队列中的条目数量取决于 defaults.conf 文件中的 previews= 指令。有关更改该指令值的信息，请参阅 defaults.conf(4) 联机资料。

您可以为各个预备请求分配不同的优先级。您可以通过在预备命令文件中输入指令来改写 FIFO 默认值。该文件写入至下面所示的位置：

```
/etc/opt/SUNWsamfs/preview.cmd
```

该文件根据请求是用于登台还是存档来安排预备请求。您也可增加特定 VSN 的优先级。此外，`preview.cmd` 文件中的设置也可根据上限（HWM）或下限（LWM）设置来重新排列所有或特定文件系统的预备请求的顺序。

`sam-initd` 后台程序会在启动时读取预备指令。每个指令行只允许有一个指令。如果在 `sam-initd` 后台程序运行期间对该文件进行了更改，则这些更改在重新启动 `sam-initd` 后台程序才能生效。注释行以井字符（#）开头，并且延伸至行的末尾。有关此文件的详细信息，请参阅 `preview.cmd(4)` 联机资料。

`preview.cmd` 文件可以包含以下两种类型的指令：

- 全局指令，应用于所有文件系统。这些指令必须出现在第一个 `fs =` 行的前面。
- 专用于文件系统的指令，它位于全局指令的后面。与 `archiver.cmd` 文件相似，`preview.cmd` 文件可以包含专用于单个文件系统的指令。在此文件中，专用于单个文件系统的指令必须出现在所有全局指令的后面。

文件系统指令必须以 `fs = file_system_name` 指令开头。该指令用于命名文件系统，其后的所有指令均属于该文件系统。文件中可以包含多个文件指令块。文件系统指令的应用范围到出现下一个 `fs =` 行或到达文件末尾为止。

注 – 当多个指令影响文件系统时，专用于文件系统的指令将取代全局指令。

VSN 和时段指令（全局）

VSN 和时段优先级指令均是全局指令。如果在 `preview.cmd` 文件中指定这些指令，则它们必须出现在任何专用于文件系统的指令的前面。也就是说，它们必须出现在所有 `fs =` 指令的前面。VSN 优先级指令的格式如下：

```
vsn_priority = value
```

该指令是一个静态的优先级因子。它表示标记为高优先级 VSN 的 VSN 的总优先级将要增加的值。`vsn_priority` 的默认值为 1000.0。当 VSN 被安排为预备请求时，它们必须已被设置优先级标记才能获得此增加值。使用 `chmed(1M)` 命令及其 `p` 选项可以设置优先级标记（例如 `chmed +p lt.AAA123`）。只有提交的 VSN 请求尚不是预备请求时，对它们设置此优先级标记才会有效。时段优先级指令的格式如下：

```
age_priority = factor
```

该指令是一个静态的优先级因子，但它的整体影响是动态的。`age_priority` 因子与请求成为预备请求的秒数（即请求等待的时间）相乘，其结果添加到请求的总优先级。请求等待满足的时间越长，时段因子就越大。设置该因子有助于确保较旧的请求不会被具有其它较高优先级因子的较新请求无限期地取代。

如果该因子大于 1.0，则它可以增加时间因子在计算总优先级中的重要性。如果小于 1.0，则会降低时间因子的重要性。如果将该因子设置为 0.0，则在计算总优先级时不考虑时间因子。

对于没有设置优先级标记的 VSN，系统将根据它在队列中等待的时间提高其优先级。这样，该 VSN 的优先级可能会高于以后进入队列且已设置优先级标记的 VSN。

界限指令（全局或文件系统专用）

预备请求的界限指令既可以用作全局指令，也可用作文件系统的专用指令。界限优先级指令用于确定预备请求的界限优先级 (`wm_priority`)。`wm_priority` 因子是下列设置的总和：

```
wm_priority = lwm_priority + lhwm_priority + hlwm_priority +  
hwm_priority
```

当 `wm_priority` 因子是正数时，计算出的总优先级所产生的影响是：增加存档请求的优先级而减少登台请求的优先级。不过，`wm_priority` 因子还可以是负数。在此情况下，将会减少存档请求的总优先级，这导致系统优先处理登台请求，然后处理存档请求。如果将此因子设置为 0.0（或根本不指定命令），则表示在文件系统遇到此情况时，系统不会对存档请求采取任何特殊的操作。有关此因子的详细信息，请参阅第 162 页的“示例 1：强制执行登台请求”中的示例。

表 6-3 显示了四个界限优先级指令及其变量。

表 6-3 界限优先级指令

优先级指令	变量
<code>lwm_priority = value</code>	<i>value</i> 用于在文件系统低于下限 (LWM) 时，指定存档请求的 <code>wm_priority</code> 因子的改变量。默认值为 0.0。
<code>lhwm_priority = value</code>	<i>value</i> 用于在文件系统超过下限 (LWM) 但持续低于上限 (HWM) 时，指定存档请求的 <code>wm_priority</code> 因子的改变量。这通常表示文件系统中的文件正在增加。默认值为 0.0。
<code>hlwm_priority = value</code>	<i>value</i> 用于在文件系统超过下限 (LWM) 但持续高于上限 (HWM) 时，指定存档请求的 <code>wm_priority</code> 因子的改变量。这通常表示释放程序不能释放足够的磁盘空间以使文件系统低于下限 (LWM)。默认值为 0.0。
<code>hwm_priority = value</code>	<i>value</i> 用于在文件系统高于上限 (HWM) 时，指定存档请求的 <code>wm_priority</code> 因子的改变量。默认值为 0.0。

总之，这四个界限设置用于创建包括百分比值（表示文件系统的充满程度）以及 HWM 和 LWM 设置级别的动态优先级因子。分配给预备请求的值取决于优先级因子是全局性的，文件系统专用的，还是未设置。

当文件系统的情况发生变化时，系统将根据以下条件重新计算与该文件系统关联的每一个 VSN 的优先级：相应的界限优先级设置；是否使用 `chmed(1M)` 命令的 `p` 选项设置了优先级。

界限优先级仅用于计算与存档有关的介质请求，而不能用于计算与登台有关的介质请求。

下面的指令示例显示了如何在文件系统处于 HWM 状态时略微增加存档请求的优先级。这些设置示例可以使释放程序释放足够的磁盘空间以使文件系统低于 LWM。

```
lhwm_priority = -200.0
hlwm_priority = 100.0
```

计算预备请求的总优先级

预备请求的优先级数值由数个静态和动态因子共同决定。数值越大，优先级就越高。静态优先级因子在生成请求时进行设置。一旦请求生成并进入等待执行状态，静态优先级因子对总优先级的影响将不会发生变化。在请求等待执行期间，动态优先级因子可以增加或减少请求的总优先级。

预备请求的总优先级是所有优先级因子的总和，其计算方式如下：

```
vsn_priority
+ wm_priority
+ (age_priority * time_in_sec_as_preview_request)
= priority
```

如何设置预备请求的优先级方案

除非绝对需要，否则请勿更改默认的预备请求 FIFO 方案。在下列条件下，可能需要更改默认的预备请求 FIFO 方案：

- 条件 1：确保首先处理登台请求，然后处理存档请求。
- 条件 2：确保存档请求在文件系统将要充满时获得最高优先级。
- 条件 3：将使用特定介质组的请求排在预备请求列表的顶部。

对于用户对数据访问要求极高、VSN 驱动器限制使用或将文件存档作为后台功能的环境，您可以使用 `preview.cmd` 文件来改变存储系统资源执行登台请求的方式。您可以自定义 `preview.cmd` 文件中的设置以支持前面所述的任何方案并改变已配置 Sun SAM-FS 或 Sun SAM-QFS 环境。

由于该文件中的设置对数据并无影响，因此建议您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，能够在存档请求和登台请求之间找到适当的平衡点。

下面的 `preview.cmd` 文件示例用于解决前面列出的三种条件：

```
# 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
```

示例 1：强制执行登台请求

下面的设置示例提供了一种可以确保登台请求先于存档请求处理的方法。本示例假定符合以下条件：

- 数个请求在队列中等待的时间为 100 秒。
- `vsn_priority` 的默认值为 1000。

表 6-4 显示了如何计算请求的总优先级。

表 6-4 请求优先级示例

优先级	计算
具有优先级的存档 VSN, LWM:	$1000 + (-200) + (1 \times 100) = 900$
具有优先级的登台 VSN, LWM:	$1000 + 0 + (1 \times 100) = 1100$
无优先级的登台 VSN, LWM:	$0 + 0 + (1 \times 100) = 100$

本示例表明在其它因子都相同时，具有负值的 `wm_priority` 会使系统优先处理登台请求，然后处理存档请求。

示例 2：强制执行存档请求

当在环境中权衡将文件登台回用户与将新文件存档至介质这二者的重要性时，最为关心的是超过上限 (HWM)。在此情况下，如果没有足够的满足存档要求的文件来降低文件系统的充满程度，则完成待定的存档请求是下一个防止文件系统充满的最佳方法。

在此情况下，只需在 `preview.cmd` 文件中进行以下设置：

```
hwm_priority = 500.0
```

示例 3：按介质确定请求的优先级

在面向项目的环境中，特定用户可能只处理占用特定 VSN 的文件组，并且其数据与其他用户的数据相互分开。在这种环境中，某些项目有时可能具有较高的优先级；因此，它们需要具有优先使用可用系统存储资源的权利。您可以使用下面的指令配置 `preview.cmd` 文件，以适当给予用户及其介质优先使用介质驱动器的权利：

```
hwm_priority = 5000.0
```

然后，对于每一个属于优先级用户群组的 VSN，输入下面的信息：

```
chmed +p lt.AAA123 ## or whatever VSN is used
```

以后，每一个要求访问 VSN AAA123（或使用的任何 VSN）的请求均优先于预备队列中的其它待定安装请求。

将来，若要降低用户介质的优先级，请为每一个 VSN 输入相反的命令：

```
chmed -p lt.AAA123 ## or whatever media type is used
```

示例 4：确定复杂请求的优先级

假设存在两个具有以下要求的 Sun SAM-FS 文件系统：

- 请求在队列中的等待时间不能太长 (`age_priority`)。
- 当文件系统低于下限 (LWM) 时，优先处理登台请求。
- 当文件系统高于下限 (LWM) 且低于上限 (HWM) 时，无需区分存档请求或登台请求的优先级。在此情况下，相关的指令如下所示：

```
lwm_priority = -200.0
lhwm_priority = 0.0
hlwm_priority = 0.0
```

在此情况下，其它指令保持不变。

当文件系统超过上限 (HWM) 时，优先处理存档请求。

如果两个文件系统同时超过了上限 (HWM)，则应首先防止第二个文件系统（例如 `samfs2`）充满。例如，当 `samfs1` 是一个用户工作文件系统而 `samfs2` 是一个关键文件系统时，便会出现这种情况。

在任何情况下（不管出现何种情形），如果已设置 `chmed(1M)` 命令的 `p` 标记，则要求访问选定 VSN 组的请求将优先于预备请求队列中的其它请求。

下面的 `preview.cmd` 文件将根据前面列出的要求排列请求的顺序。

```
age_priority = 100.0
vsn_priority = 20000.0
lhwm_priority = -200.0
hlwm_priority = -200.0
fs = samfs1
hwm_priority = 1000.0
fs = samfs2
hwm_priority = 5000.0
```

回收

*回收*是指从存档卷中收回空间的过程。回收程序与存档程序配合工作，以收回由无用的存档副本占用的空间。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存档副本。回收程序可以识别那些其中绝大部分是过期存档副本的存档卷，并将这些卷中的非过期副本移动到其它卷中。当某个给定的卷中只包含过期副本时，即可执行站点定义的操作。例如，您可以重新标记此类卷以便立即重新使用此类卷，或将其中的数据导出至离站存储设备，从而单独保存文件更改的历史记录。由于回收过程只与用户的数据文件有关，因此用户不会觉察到回收过程。

本章包括下列主题：

- 第 165 页的“概述”
- 第 167 页的“回收指令”
- 第 169 页的“配置回收程序”
- 第 179 页的“排除回收程序的故障”

概述

回收程序负责将过期存档副本占用的空间保持在由站点指定参数定义的最低水平。在任何时候，给定存档卷的空间均由以下各项组成：

- *当前数据空间*，由当前有效的存档映像占用的空间。
- *过期数据空间*，由当前不再有效的存档映像占用的空间。
- *可用空间*，未被当前有效或过期的存档映像占用的空间。

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 3G 数据的 10G 磁带卷来说，它的容量为 10G，可用空间为 7G。

对于全新的存档介质或新标记的存档介质，其容量等于可用空间。当将数据存档至该介质时，可用空间会减少，而当前数据空间会增加。

当更改或删除文件系统中已存档的文件时，这些文件的存档映像会过期，并且其类别由当前数据类别变为过期数据类别。这些映像占用的物理空间并没有发生变化；只是文件系统中已没有指向该空间的文件。

这些过期的映像（即过期数据）最终会占满全部可用空间。只有回收空间，才能删除这些映像并使它们占用的空间变为可用空间。回收程序的目标是将过期数据占用的空间转变为可用空间，而丝毫不损任何当前数据。

您只能向磁带卡盒添加数据，而不能在其中重新写入数据。重新使用磁带卡盒的唯一方法是从卡盒中移走所有当前数据，重新标记卡盒，然后从头开始使用卡盒。为达到这一目的，存档程序将识别卷中所有的当前存档映像，然后标记这些映像，以便它可以使另一卷中的副本替换要回收的卷中的副本。此操作过程称为**重新存档**。您可以使用 `sls(1)` 命令及其 `-D` 选项来显示某个文件的有关信息。`sls(1)` 命令的输出中显示了该文件是否已安排重新存档。

实际上，回收程序永远不会将文件移至新介质。它只是为选定卷中的所有文件设置“重新存档”文件属性，而后由存档程序挑选这些文件。回收程序将在选定介质上设置 `recycle` 属性，从而使该介质在存档期间不能接收任何新数据。余下的工作由存档程序完成。存档程序处理具有 `rearchive`（重新存档）属性的文件，并将这些文件的存档副本从选定介质实际移至新介质。

重新存档 VSN 中的所有存档映像之后，该 VSN 中只包含可用空间和过期数据空间。此时，您可以安全地重新标记卡盒。

您可以通过输入 `sam-recycler(1M)` 命令来启动回收过程。回收过程既可手动执行，也可通过 `cron(1)` 作业来执行。您可以采用以下方式执行回收：

- 按自动化库利用率
- 按存档组利用率

回收过程因存档介质的不同而略有差异。以下几节介绍了回收磁盘存档副本和回收可移动介质卷的过程。

回收磁盘存档副本

与回收可移动介质卷相比，回收已写入至磁盘的过期存档副本的过程较为简单。当您回收磁盘卷时，其中的文件不会重新存档至其它介质。由于不需要重新存档，因此过期存档副本会从磁盘中删除。

回收磁盘存档副本的唯一方法是按存档组进行回收。您不能按库进行回收。执行回收过程时，无需使用 `recycler.cmd` 文件，因为磁盘存档副本的所有回收活动均由 `archiver.cmd` 文件中的指令控制。

回收可移动介质存档副本

回收程序可以定期运行。每次启动该程序后，它会尽力完成所有工作。在两次运行之间，回收程序将状态信息保留在库目录和索引节点中。

当回收程序运行时，它在存档程序实际将数据移至新介质之前早就结束了工作。事实上，回收程序必须提早结束才能使存档程序顺利完成重新存档任务。如果随后存档程序并未运行、介质不可用或其它任何存档程序异常启动，则具有 `rearchive` 属性的文件不会被重新存档至新介质。在此情况下，旧介质中的数据将无法清除干净。此外，在下次运行回收程序（查找需要重新标记的介质 - 回收程序和存档程序在上一次组合运行时已将该介质中的数据清除干净）时，如果存档程序未能重新存档所有文件，则您不能重新标记和重新使用该介质，因为其中仍存在有效的存档副本。

回收指令

`recycler.cmd` 文件接受以下几节中所述的指令：

- 第 167 页的“`logfile` 指令”
- 第 168 页的“`no_recycle` 指令”
- 第 168 页的“库指令”

logfile 指令

`logfile` 指令用于指定回收程序日志文件。该指令的格式如下：

```
logfile = filename
```

其中：

filename 指定日志文件的路径。

下面是 `logfile=` 指令行的示例：

```
logfile=/var/adm/recycler.log
```

no_recycle 指令

`no_recycle` 指令可以使您防止回收卷。要指定 VSN，您可以使用标准表达式以及一个或多个特定介质类型。该指令的格式如下：

```
no_recycle media_type VSN_regex [ VSN_regex ... ]
```

其中：

<i>media_type</i>	指定 <code>mcf(4)</code> 联机资料中列出的介质类型。
<i>VSN_regex</i>	指定一个或多个由空格隔开的标准表达式以说明卷。有关标准表达式的格式信息，请参阅 <code>regexp(5)</code> 联机资料或第 93 页的“使用样式匹配的文件名 <code>search_criteria: -name regex</code> ”。

通过指定 *media_type*，您可以防止回收存储在特定类型介质上的卷。一个或多个 *VSN_regex* 规格可以使您使用标准表达式来标识那些免于回收的特定卡盒。

例如，对于其 VSN 标识以 DLT 开头的磁带卷，下面的指令行可防止回收程序回收这些磁带卷：

```
no_recycle lt DLT.*
```

库指令

库指令可以使您为那些与特定库关联的 VSN 指定各种不同的回收参数。该指令的格式如下：

```
library parameter [ parameter ... ]
```

其中：

<i>library</i>	指定库的名称，该名称与在 <code>mcf(4)</code> 文件的 Family Set（系列集）字段中指定的名称相同。
<i>parameter</i>	指定表 7-1 中列出的一个或多个以空格隔开的 <i>parameter</i> （参数）关键字。

表 7-1 库指令 *parameter* (参数) 值

参数	操作
-dataquantity <i>size</i>	限制回收程序可以安排重新存档的数据量，以免清除有用数据所在的卷。默认值为 1G。
-hwm <i>percent</i>	库利用率的上限。默认值为 95。
-ignore	防止回收库中的卷。在测试 <code>recycler.cmd</code> 文件时，此指令十分有用。
-mail [<i>email_address</i>]	向指定的 <i>email_address</i> 发送电子邮件消息。默认情况下，系统不发送电子邮件。如果已指定 <code>-mail</code> 但未带任何变量，则会将电子邮件发送至超级用户 (<code>root</code>)。
-mingain <i>value</i>	最小 VSN 增益百分比。默认值为 50。
-vsncount <i>count</i>	限制可回收卷的数量。默认值为 1。

例如，查看下面的指令行：

```
gr47 -hwm 85 -ignore -mail root -mingain 40
```

它为库 `gr47` 指定了以下各项：

- 当库中卷的充满程度达到 85% 时，应考虑对库执行回收操作。
- 最小增益百分比为 40%。
- 重新存档的数据量最多为 1G。这是一个默认值，因此未在 `recycler.cmd` 文件中指定。
- 只能回收一个卷。这也是一个默认设置。
- 将回收消息发送至超级用户 (`root`)。

配置回收程序

配置回收程序之前，请注意以下事项：

- `archiver.cmd` 文件中的指令按存档组控制回收过程。`recycler.cmd` 文件中的指令按库控制回收过程。此外，`recycler.cmd` 文件还控制一般回收程序操作。有关回收程序指令的信息，请参阅第 167 页的“回收指令”。
- 不要对包含可移动介质文件的卷运行回收程序。可移动介质文件通过 `request(1)` 命令创建。回收程序不会保留通过 `request(1)` 命令创建的可移动介质文件。包含可移动介质文件的卷永远不能清除干净。

- 对 Sun SAM-FS 或 Sun SAM-QFS 文件系统进行维护时，切勿运行回收程序。回收程序使用 `.inodes` 文件和 `mcf` 文件来识别当前文件或过期文件以及与文件系统关联的设备。如果没有这些文件中的正确信息，回收程序可能会将当前存档数据视为过期数据而对其执行回收操作。
- 在运行回收程序时，必须安装所有 Sun SAM-FS 和 Sun SAM-QFS 文件系统。如果您要对联机磁盘执行回收操作，则必须安装包含此磁盘卷的文件系统，并且可以访问主机系统。

默认情况下，系统不会启用回收程序。您必须通过输入 `sam-recycler(1M)` 命令来启动回收程序。启动回收程序后，第 168 页的“库指令”中指定的默认回收程序设置将会生效。有关回收程序进程的详细信息，请参阅 `sam-recycler(1M)` 联机资料。

以下几节介绍配置回收程序的过程。该过程包括下列几个步骤：

- 第 170 页的“步骤 1: 配置 `recycler.cmd` 文件（可选）”
- 第 172 页的“步骤 2: 编辑 `archiver.cmd` 文件（可选）”
- 第 174 页的“步骤 3: 运行回收程序”
- 第 175 页的“步骤 4: 为回收程序创建 `crontab` 文件（可选）”
- 第 175 页的“步骤 5: 删除 `-recycle_ignore` 和 `ignore` 指令”
- 第 176 页的“步骤 6: 创建 `recycler.sh` 文件”

如果您要回收库中的卡盒，则此过程还包括创建 `recycler.cmd` 文件以及（可选）编辑 `archiver.cmd` 文件。如果您将文件存档至磁盘，则只能按存档组进行存档，因此要启用这些磁盘卷的回收过程，您需要编辑 `archiver.cmd` 文件。下面的过程说明了如何为存档介质配置回收程序。

▼ 步骤 1: 配置 `recycler.cmd` 文件（可选）

如果您要回收库中卡盒上的存档副本，请执行本步骤。

如果您要回收磁盘卷上的存档副本，则不必执行本步骤，因为回收过程由 `archiver.cmd` 文件中的指令控制。有关在 `archiver.cmd` 文件中配置回收过程的详细信息，请参阅第 172 页的“步骤 2: 编辑 `archiver.cmd` 文件（可选）”。

`recycler.cmd` 文件中包含一般的回收指令，另外，它还可包含用于 Sun SAM-FS 或 Sun SAM-QFS 环境中每一个库的指令。有关回收指令的信息，请参阅第 167 页的“回收指令”。

即使您按存档组执行回收操作，也应在 `recycler.cmd` 文件中配置每一个库。这可以确保回收程序能够回收不属于存档组的 VSN（如有必要）。

典型的 `recycler.cmd` 文件包含以下命令行：

- `logfile=` 命令行用于指定回收程序日志文件。系统将回收消息和回收报告写入至该文件。
- 一个或多个命令行，用于要回收的卷所在的每一个库。该命令行必须包含要回收库的系列集名（来自 `mcf` 文件）。这使回收程序可以识别库。

由于您仍在创建 `recycler.cmd` 行，并且尚未经过测试，因此应使用 `ignore` 关键字。`ignore` 关键字将在本过程后面的步骤中删除。

recycler.cmd 文件示例

代码示例 7-1 显示了 `recycler.cmd` 文件的示例。

代码示例 7-1 `recycler.cmd` 文件示例

```
logfile = /usr/tmp/recycler.log
stk30 -hwm 51 -mingain 60 -ignore -mail root
```

以下部分介绍代码示例 7-1 中指定的参数。

-hwm 51 参数

通过指定使用率上限，您可以设置开始执行回收过程时的介质使用率。此百分率是库中已用空间与总容量的比率。例如，某个库包括 10 个 20G 的磁带，其中三个磁带的充满程度为 100%，另外七个磁带的充满程度均为 30%，则介质利用率为：

$$\frac{(3 * 1.00 + 7 * 0.30) * 20G}{(10 * 20G)} * 100\% = 51\%$$

请注意，该计算方法并不区分当前数据和过期数据，它只考虑已用介质的量。

在本示例中，如果上限为 51% 或更小，回收程序不会自动选择自动化库的任何 VSN 进行回收。

注 – 通过运行下面的命令设置回收标记，您可以强制回收某个 VSN：

```
# chmed +c lt.AAA123
```

当设置 `+c` 标记时，存档程序不会再向该卷写入存档映像。您可以通过 `samu(1M)` 实用程序来查看 `+c` 标记。有关详细信息，请参阅 `chmed(1M)` 和 `samu(1M)` 联机资料。

`-mingain=60` 参数

minimum VSN gain percentage 用于设置通过回收卡盒所获得的空间量的下限。例如，在自动化库的某个卡盒中，95% 是当前数据空间，另外 5% 是过期数据空间，因此通过回收该卡盒所获得的增益仅为 5%。为获得此空间（即 5%）而移动另外 95% 的空间，这可能是不值得的。将最小增益设置为 6% 或更大值可以防止回收程序自动选择此 VSN 示例进行回收。

另一个示例是某个卡盒中具有 90% 的过期数据空间，5% 的当前数据空间，以及 5% 的可用空间。如果回收此卡盒，则会获得 90% 的增益。

`-ignore` 参数

`ignore` 关键字用于防止回收程序回收某个特定的库。在您配置回收程序时，应使用此关键字。

`mail root` 参数

`mail` 关键字用于指定回收程序在回收给定的库后发送邮件。邮件消息的主题行如下所示：

```
Robot robot-name recycle
```

消息正文范例如下所示：

```
I will recycle VSN vsn.
```

```
Cannot find any candidate VSN in this media changer.
```

```
Previously selected VSN vsn is not yet finished recycling.
```

```
Previously selected VSN vsn is now finished recycling. It will  
now be post-recycled.
```

▼ 步骤 2: 编辑 `archiver.cmd` 文件（可选）

如果您按存档组执行回收，请执行本步骤。如果您将文件存档至磁盘，则按存档组执行回收是唯一可行的方法，因此您必须完成本步骤才能进行回收。

如果您按库执行回收，请继续下一步骤。

要按存档组执行回收，请编辑 `/etc/opt/SUNWsamfs/archiver.cmd` 文件并添加您要回收的存档组的有关信息。回收指令必须出现在 `params` 和 `endparams` 指令之间。表 7-2 显示了存档组回收指令。

表 7-2 存档组回收指令

指令	功能
<code>-recycle_dataquantity size</code>	限制回收程序可以安排重新存档的数据量，以免清除有用数据所在的卷。
<code>-recycle_hwm percent</code>	设置上限百分比。
<code>-recycle_ignore</code>	防止回收存档组。
<code>-recycle_mailaddr mail_address</code>	将回收程序消息发送到 <code>mail_address</code> 。
<code>-recycle_mingain percent</code>	限制回收程序回收那些不能获得指定可用空间（即 <code>percent</code> 或更多）的 VSN。
<code>-recycle_vsncount count</code>	限制可重新存档的卷的数量（即 <code>count</code> ）。

有关上述指令的详细信息，请参阅 `archiver.cmd(4)` 联机资料。

如前所述，请在全局指令部分加入 `-recycle_ignore` 指令以防止回收程序在尚未测试配置之前进行回收操作。

代码示例 7-2 显示了用于回收磁盘存档的 `archiver.cmd` 示例。

代码示例 7-2 `archiver.cmd` 文件中的磁盘存档规格

```
fs = samfs1
  1 2m

arset0 testdir0
  1 2m
  2 4m

arset1 testdir1
  1 2m
  2 4m

params
arset0.1 -disk_archive disk01 -recycle_hwm 5 -recycle_mingain 2
arset1.1 -disk_archive disk02 -recycle_hwm 5 -recycle_mingain 2
endparams
```

▼ 步骤 3: 运行回收程序

运行 `sam-recycler(1M)` 命令。回收程序将读取 `recycler.cmd` 文件。检查标准输出、日志、SAM 日志和 `/var/adm/messages`，看看是否存在来自回收程序的错误消息。代码示例 7-3 显示了回收可移动介质卡盒时的回收程序日志文件范例。

代码示例 7-3 可移动介质卡盒的回收程序日志文件

```
===== 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
```

代码示例 7-3 可移动介质卡盒的回收程序日志文件 (续)

```
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 =====
```

代码示例 7-4 显示了回收磁盘存档文件时的回收程序日志文件范例。

代码示例 7-4 磁盘存档文件的回收程序日志文件示例

```
---Archives---  -----Percent-----
----Status-----      Count      Bytes      Use Obsolete Free      Library:Type:VSN
new candidate          0          0          0    41      59 <none>:dk:disk01

677 files recycled from VSN disk01 (mars:/sam4/copy1)
0 directories recycled from VSN disk01 (mars:/sam4/copy1)
```

▼ 步骤 4: 为回收程序创建 crontab 文件 (可选)

如果系统运行正常, 您即可为超级用户创建 crontab 条目以定期运行回收程序。您可能希望每隔 2 个小时运行一次回收程序, 视您的站点条件而定。

下面是超级用户 (root) 的 crontab 文件中的条目示例, 它确保 cron 后台程序在每个奇数小时内每隔 5 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
```

▼ 步骤 5: 删除 -recycle_ignore 和 ignore 指令

从 archiver.cmd 文件中删除 -recycle_ignore 指令, 并从 recycler.cmd 文件中删除 ignore 指令。现在, 回收程序即可执行回收过程。

▼ 步骤 6: 创建 `recycler.sh` 文件

如果您要回收可移动介质卡盒上的存档副本，请执行本步骤。

如果您只将文件存档至磁盘，请不要执行本步骤。

在存档程序将 VSN 中的所有当前映像重新存档至另一个 VSN 后，回收程序将执行 `recycler.sh` 脚本。您可以在 `/opt/SUNWsamfs/examples/recycler.sh` 中找到这些操作的示例。代码示例 7-5 显示了如何重新标记已回收的 VSN 并向超级用户发送邮件。

代码示例 7-5 `recycler.sh` 文件示例

```
#!/bin/csh -f
#
# /opt/SUNWsamfs/sbin/recycler.sh - post-process a VSN after recycler has
# drained it of all known active archive copies.
#
# Arguments are:
#   $1 - generic media type "od" or "tp" - used to construct the name
#       of the appropriate label command: odlabel or tplabel
#
#   $2 - VSN being post-processed
#
#   $3 - Slot in the library where the VSN is located
#
#   $4 - equipment number of the library where the VSN is located
#
#   $5 - actual media type ("mo", "lt", etc.) - used to chmed
#       the media if required
#
#   $6 - family set name of the physical library, or the string
#       "hy" for the historian library. This can be used to
#       handle recycling of off-site media, as shown below.
#
#   $7 - VSN partition, used for optical and D2 media
#
# $Id: recycler.sh,v 2.7 2000/04/10 14:51:45 ram Dev $
#
# It is a good idea to log the calls to this script
#echo `date` $* >> /var/opt/SUNWsamfs/recycler.sh.log
#
# As an example, if uncommented, the following lines will relabel the VSN,
# if it exists in a physical library. If the VSN is in the historian
# catalog (e.g., it's been exported from a physical library and moved
# to off-site storage), then email is sent to "root" informing that the
```


代码示例 7-5 recycler.sh 文件示例 (续)

```
#!/bin/csh -f
# medium is ready to be returned to the site and reused.
#
#set stat=0
#if ( $6 != hy ) then
# /opt/SUNWsamfs/sbin/chmed -R $5.$2
# /opt/SUNWsamfs/sbin/chmed -W $5.$2
# if ( $5 != "d2" ) then
#     if ( $1 != "od" ) then
#         /opt/SUNWsamfs/sbin/${1}label -w -vsn $2 -old $2 $4\:$3
#         if ( $status != 0 ) then
#             set stat = 1
#         endif
#     else
#         /opt/SUNWsamfs/sbin/${1}label -w -vsn $2 -old $2 $4\:$3\:$7
#         if ( $status != 0 ) then
#             set stat = 1
#         endif
#     endif
# else
#     /opt/SUNWsamfs/sbin/${1}label -w -vsn $2 -old $2 $4\:$3\:$7
#     if ( $status != 0 ) then
#         set stat = 1
#     endif
# endif
#endif
# else
#     /opt/SUNWsamfs/sbin/${1}label -w -vsn $2 -old $2 $4\:$3\:$7
#     if ( $status != 0 ) then
#         set stat = 1
#     endif
# endif
#endif
# else
#     mail root <</eof
#VSN $2 of type $5 is devoid of active archive
#images. It is currently in the historian catalog, which indicates that
#it has been exported from the on-line libraries.
#
#You should import it to the appropriate library, and relabel it using
#${1}label.
#
#This message will continue to be sent to you each time the recycler
#runs, until you relabel the VSN, or you use the Sun SAM-FS samu or
#robottool programs to export this medium from the historian catalog to
#suppress this message.
#/eof
#endif
#echo `date` $* done >> /var/opt/SUNWsamfs/recycler.sh.log
#if ( $stat != 0 ) then
# exit 1
#else
# exit 0
#endif
#
```

代码示例 7-5 recycler.sh 文件示例 (续)

```
#!/bin/csh -f
#
# These lines would inform "root" that the VSN should be removed from the
# robotic library:
#
#mail root <</eof
#VSN $2 in library $4 is ready to be shelved off-site.
#/eof
#echo `date` $* done >> /var/opt/SUNWsamfs/recycler.sh.log
#exit 0

# The default action is to mail a message reminding you to set up this
# file. You should comment out these lines (through and including the /eof
# below) after you've set up this file.
#
mail root <</eof
The /opt/SUNWsamfs/sbin/recycler.sh script was called by the Sun SAM-FS recycler
with the following arguments:

    Media type: $5($1)  VSN: $2  Slot: $3  Eq: $4
    Library: $6

/opt/SUNWsamfs/sbin/recycler.sh is a script which is called when the recycler
determines that a VSN has been drained of all known active archive
copies. You should determine your site requirements for disposition of
recycled media - some sites wish to relabel and reuse the media, some
sites wish to take the media out of the library for possible later use
to access historical files. Consult the recycler(1m) man page for more
information.
/eof
#echo `date` $* done >> /var/opt/SUNWsamfs/recycler.sh.log
exit 0
```

回收程序使用以下变量来调用 /opt/SUNWsamfs/sbin/recycler.sh 脚本:

```
Media type: $1  VSN: $2  Slot: $3  Eq: $4
```

当回收程序确定已清除 VSN 中所有已知的有效存档副本后, 它将调用 /opt/SUNWsamfs/sbin/recycler.sh 脚本。您应确定您的站点对分配已回收卡盒的要求。某些站点选择重新标记或重新使用卡盒; 而其它站点选择从自动化库中取出卡盒以便将来用于访问历史文件。有关详细信息, 请参阅 recycler(1M) 和 recycler.sh(4) 联机资料。

排除回收程序的故障

回收程序的最常见问题是显示类似于下面的消息。回收程序在启动时可能会生成该消息：

```
Waiting for VSN mo:OPT000 to drain, it still has 123 active archive
copies.
```

此消息可能由以下情况引起：

- 情况 1：存档程序未能重新存档卷中的 123 个有效存档副本。
- 情况 2：这 123 个存档副本不指向文件系统中的文件，而是指向 123 个元数据存档副本。

造成第 1 种情况的原因可能包括：

- 需要重新存档的文件被标记为 `no_archive`。
- 需要重新存档的文件位于 `no_archive` 存档组中。
- 由于没有可用的 VSN 而无法存档文件。
- `archiver.cmd` 文件中包含 `wait` 指令。

要确定引起问题的情况，请运行回收程序并选择 `-v` 选项。该选项将显示与回收程序日志文件中的 123 个存档副本相关联的文件的完整路径名。回收程序日志文件中包含类似于下面的消息：

```
Archive copy 2 of /sam/fast/testA resides on VSN LSDAT1
Archive copy 1 of /sam3/tmp/dir2/filex resides on VSN LSDAT1
Archive copy 1 of Cannot find pathname for file system /sam3
inum/gen 30/1 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gunk/tstfilA00 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gunk/tstfilF82 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gunk/tstfilV03 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gink/tstfilA06 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gink/tstfilA33 resides on VSN LSDAT1
Waiting for VSN dt:LSDAT1 to drain, it still has 8 active archive
copies.
```

本输出示例中显示了包含七个路径名的消息和一则包含 `Cannot find pathname...` 文本的消息。若要纠正未清除 LSDAT1 的问题，您需要确定未重新存档这七个文件的原因。重新存档七个文件之后，只有一个存档副本未与文件关联。请注意，只有在系统崩溃而造成 `.inodes` 文件部分损坏时才会发生这种情况。

若要解决查找路径名的问题，请运行 `samfsck(1M)` 以收回遗留索引节点。如果您不选择运行 `samfsck(1M)`，或您无法卸载文件系统以运行 `samfsck(1M)`，请检查 `recycler -v` 的输出以确保清除了有效存档副本，然后手动重新标记卡盒。不过，由于回收程序会再次遇到仍保留在 `.inodes` 文件中的无效索引节点，因此当该 VSN 再次成为回收对象时，会发生同样的问题。

当回收程序未能选择任何 VSN 以进行回收时，会发生另一个回收程序问题。要确定每一个 VSN 遭到拒绝的原因，请运行回收程序并选择其 `-d` 选项。它将显示有关回收程序如何选择 VSN 进行回收的信息。

图形用户界面 (GUI) 工具

本章介绍用于管理 Sun SAM-FS 或 Sun SAM-QFS 环境中设备的 GUI 工具。您可以使用以下两个 GUI 工具来管理传输器、设备和介质安装请求：

- `libmgr(1M)` — 提供了用于管理所有自动化库和设备的单个界面，用户可以自定义该界面以适合站点操作。
- `samtool(1M)` — 它由三个界面组成：`robottool`、`devicetool` 和 `previewtool`。

若要使用这些工具，用户必须熟悉 Window 系统和鼠标键。

注 – 除这些 GUI 工具之外，用户还可使用全屏操作员工具 `samu(1M)` 来管理 Sun SAM-FS 或 Sun SAM-QFS 环境中的设备。有关使用 `samu(1M)` 命令的说明，请参阅第 217 页的“使用 `samu(1M)` 操作员实用程序”。

本章包括下列主题：

- 第 182 页的“操作员权限级别”
- 第 182 页的“使用 `libmgr(1M)`”
- 第 191 页的“使用 `samtool(1M)`”
- 第 193 页的“使用 `robottool(1M)`”
- 第 205 页的“使用 `devicetool(1M)`”
- 第 212 页的“使用 `previewtool(1M)`”

操作员权限级别

本章介绍的 GUI 工具只供超级用户使用。libmgr(1M) 只能由超级用户运行。不过，samtool(1M)、robottool(1M)、devicetool(1M) 和 previewtool(1M) 界面可由超级用户以及操作员群组中的个体成员运行。

作为站点管理员，您可以定义无超级用户权限的操作授权。但是，这种特殊授权所授予的权限只能用于执行操作员类型的功能，例如清除磁带载入请求和更改设备状态等。您可以在 `/etc/opt/SUNWsamfs/defaults.conf` 文件中设置操作员群组并定义允许操作员执行的任务。具有超级用户 (root) 权限的用户可以完全访问 samtool 中的功能。属于操作员群组的用户不具有完全访问权限，并且只能执行某些功能。在尝试使用 robottool、devicetool 和 previewtool 中的功能时，用户会明显地体会到这一点。

用户可在 `defaults.conf` 文件中使用 `operator` 关键字来定义单个操作员群组，并使用 `oper_privileges` 关键字定义操作员群组有权执行的任务。标记介质、执行核查、在自动化库中移动卡盒以及更改设备状态等，这些都是可以定义的操作员任务示例。

有关操作员有权执行的任务的完整列表，请参阅 `defaults.conf(4)` 联机资料。

使用 libmgr(1M)

库管理器（通过 libmgr(1M) 命令启动）是一个用于管理自动化库的 GUI 工具。用户可以使用 libmgr(1M) 检查自动化库和卡盒的状态、导入和导出卡盒以及回应卡盒载入请求。

▼ 启动库管理器

若要启动库管理器，请在操作系统提示符下输入下面的命令：

```
# libmgr&
```

▼ 重设库管理器的显示屏幕、图像和标题

libmgr 中的显示屏幕、图像和标题具有高度的可配置性。libmgr 在启动时将会读取 /etc/opt/SUNWsamfs/SamGUI.rsc 资源文件。如果不进行任何更改，libmgr 将根据 mcf 文件中定义的设备产品标识、供应商标识和设备号来显示设备标题和图像。

您可以使用 SamGUI.rsc 文件设置下列各项：

- 设备和介质的标题及图像
- 目录设置
- 安装请求设置
- 屏幕设置，包括高度、宽度和字体大小

有关资源设置的完整列表，请参阅 SamGUI.rsc(4) 联机资料。

若要重新配置 libmgr 显示屏幕，您必须编辑 SamGUI.rsc 文件、退出 libmgr 并重新启动 libmgr。

库管理器显示屏幕

库管理器显示屏幕分为以下三个水平面板：

- 库面板，位于顶部。如果没有配置任何传输器，则不会显示此面板。
- 目录面板，位于中部。
- 文件系统和安装请求面板，位于底部。

显示屏幕由一些可通过鼠标操作的对象组成。表 8-1 列出了大多数对象回应鼠标的方式。

表 8-1 鼠标操作

鼠标操作	作用
左击	选择对象。
右击	显示活动的下拉菜单。
双击	显示有关对象的详细信息。

▼ 执行传输器操作

在此面板中，将光标放在所需的传输器图像上。表 8-2 列出了您可以执行的操作。

表 8-2 传输器操作

所需操作	鼠标键	菜单选择
将自动化库的状态转换为打开、关闭或禁用。	右击	选择 On（打开）、Off（关闭）、Unavailable（不可用）或 Down（禁用）。
导入介质。	右击	选择 Import（导入）。
从传输器目录中卸载 VSN。	右击	选择 Unload（卸载）。传输器的目录会被清空并将传输器设置为 Off（关闭）。依次将传输器设置为 On（打开）和 Reset（重设）。
全面核查传输器。	右击	选择 Audit（核查）。

▼ 执行介质操作

在目录面板中，选择所需的传输器。单击鼠标左键选择介质，然后选择表 8-3 中列出的操作。

表 8-3 介质操作

所需操作	鼠标键	菜单选择
标记或重新标记介质。	右击	选择 Label（标签）。输入 VSN、块大小，并（可选）重新标记或删除。
核查 VSN。	右击	选择 Audit（核查）。
安装 VSN。	右击	选择 Mount（安装）。
移动 VSN。	右击	选择 Move（移动）。输入目标端口编号。
导出 VSN。	右击	选择 Export（导出）。

▼ 执行介质驱动器操作

将光标放在所需的介质驱动器图像上，然后选择表 8-4 中列出的操作。

表 8-4 介质驱动器操作

所需操作	鼠标键	菜单选择
标记或重新标记介质。	右击	选择 Label（标签）。输入 VSN、块大小，并（可选）重新标记或删除。
将驱动器的状态转换为打开、关闭、不可用或禁用。	右击	选择 On（打开）、Off（关闭）、Unavailable（不可用）或 Down（禁用）。

▼ 查看文件系统状态及属性

若要查看文件系统状态及属性并进行适当的更改，请双击所需的文件系统。此时，屏幕上会显示该文件的详细信息窗口。



图 8-1 libmgr(1M) — 文件系统状态及属性

libmgr(1M) 联机资料中的 Icon Attributes（图标属性）部分介绍了文件系统的设备状态及设备属性。您可以通过选择按钮并单击 OK（确定）来更改属性或状态。

注 – 在此界面中对安装选项所做的任何更改均在卸载文件系统后停止生效。

▼ 查看介质驱动器状态及属性

若要查看介质状态及属性并进行适当的更改，请双击所需的介质驱动器图像。此时，屏幕上会显示该设备的设备状态及属性窗口。



图 8-2 libmgr(1M) — 介质驱动器状态及属性屏幕

libmgr(1M) 联机资料中的 Icon Attributes（图标属性）部分介绍了介质驱动器的设备状态及设备属性。您可以通过选择适当的按钮并单击 OK（确定）来启用对属性或状态所做的更改。

▼ 查看 VSN 目录显示屏幕

若要查看 VSN 的目录设置，请根据需要双击目录面板中的 VSN。此时，屏幕上会显示选定 VSN 的完整属性列表。此类信息源自 mcf 文件中定义的传输器目录。

状态: *slot number 1

VSN 000038 类型 lt 条码 000035

容量 106457399296

已用存储空间 0%

块大小 128 访问次数 23

介质属性

已标记 已破坏 正在清除

已擦除 写保护 只读

回收

标签时间 2002-8-30 20:37:16

安装时间 2002-8-30 20:37:16

修改时间 2002-8-30 20:37:49

插槽编号 0

目录条目属性

半写状态 正在禁用

已占用 不可用

写入/导出

确定 取消

图 8-3 libmgr(1M) — VSN 目录显示屏幕

使用 samtool(1M)

samtool(1M) 是 robottool(1M)、devicetool(1M) 和 previewtool(1M) 工具的初始启动窗口。以下几节介绍了这些工具。

▼ 启动和退出 samtool(1M)

若要启动 samtool，请在操作系统提示符下输入下面的命令：

```
# samtool&
```

此时，系统会显示 samtool 工具组。该显示屏幕中包括 robottool(1M)、devicetool(1M) 和 previewtool(1M) 工具的图标。

若要退出 samtool，请右击窗口顶部的栏，然后选择 QUIT（退出）或 Close（关闭）。

选择工具

samtool(1M) 显示屏幕中包括每一个工具的图标。若要激活工具，请左击您要使用的工具的图标。这些工具包括：

- robottool(1M)，可以使您查看和管理 Sun SAM-FS 或 Sun SAM-QFS 环境中配置的传输器的有关信息。
- devicetool(1M)，可以使您查看和管理 Sun SAM-FS 或 Sun SAM-QFS 环境中配置的设备的有关信息。
- previewtool(1M)，可以使您查看和管理 Sun SAM-FS 或 Sun SAM-QFS 环境中的待定安装请求。

注 – 此外，您可以通过在命令行中输入工具的名称来启动工具。例如，要启动 robottool(1M)，请在命令行中输入 robottool。要在后台启动工具，请在工具名后面输入 “&” 字符。例如，要在后台启动 robottool(1M)，请输入 robottool&。

▼ 更新显示屏幕

默认情况下，系统会每隔五秒钟自动刷新一次 `samtool(1M)` 显示屏幕。您可以更改刷新率或禁用自动刷新功能。此外，您还可以在必要时强制显示屏幕进行自我更新。更新按钮、刷新复选框和刷新字段用于控制更新显示屏幕。

▼ 更改刷新率

1. **确保已启用自动刷新功能。**
也就是说，确保刷新复选框中包含表示已启用刷新的复选标记。
2. **在刷新字段中输入新的刷新率，或使用增加/减少设置按钮。**

▼ 更新工具显示屏幕

- **若要立即更新工具显示屏幕，请单击窗口右上侧的 `Update`（更新）按钮。**

▼ 控制自动刷新功能

- **若要启用或禁用自动刷新功能，请单击刷新。**
当刷新复选框中包含复选标记时，表示已启用自动刷新功能。

▼ 管理屏幕资源

通过使用 `fontfamily` 资源设置，您可以更改 `samtool(1M)` 显示屏幕中面板列表的字体。下面是来自 `.Xdefaults` 资源文件的示例，它定义了用于 `robottool` 的字体集：

```
robottool.fontfamily: fixed
```

▼ 显示联机帮助

1. **左击 `Help`（帮助）按钮可以显示有关 `samtool(1M)` 及其操作的一般帮助信息。**
2. **右击 `Help`（帮助）按钮可以显示包含每一个工具项目的菜单。**
选择与您要获得帮助的工具对应的菜单项目。

使用 robottool(1M)

robottool 实用程序显示了已配置的传输器以及与选定传输器关联的 VSN 目录和设备。默认情况下，系统在启动 robottool 时会选择第一个 Sun SAM-FS 或 Sun SAM-QFS 传输器。在传输器显示屏幕中选择传输器会使系统显示选定传输器的 VSN 目录和设备。

图 8-4 显示了 robottool 显示屏幕的范例。

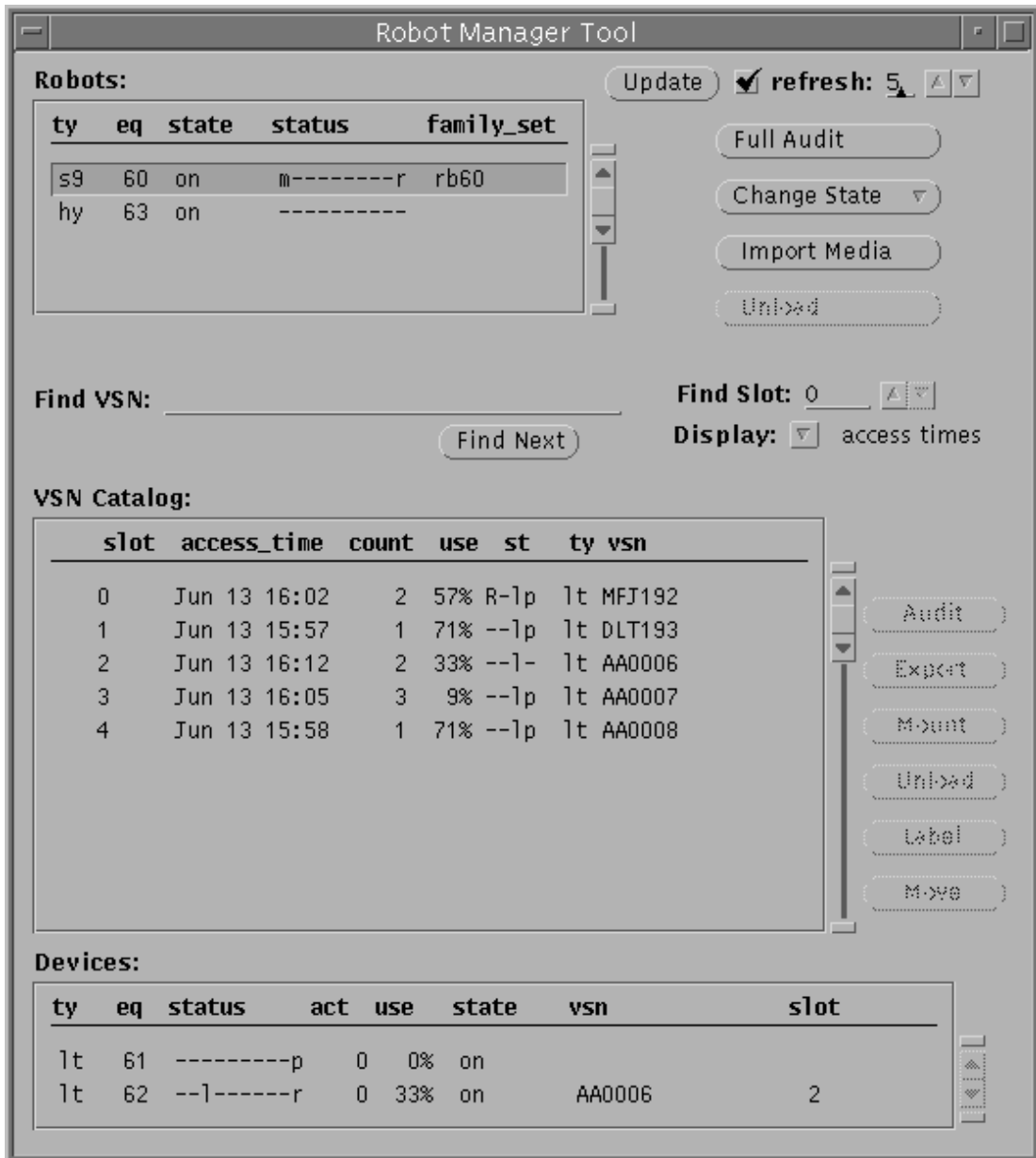


图 8-4 robottool(1M) — 初始屏幕

robottool 显示屏幕包括三个区域：

- 传输器
- VSN 目录
- 设备

下面几节介绍了这些显示屏幕区域的内容。

▼ 启动 robottool(1M)

在操作系统提示符下输入下面的命令：

```
# robottool&
```

传输器

Robots（传输器）区域列出了 Sun SAM-FS 或 Sun SAM-QFS 环境中配置的所有传输器。表 8-5 列出了系统为每一个传输器显示的信息。

表 8-5 传输器显示区域

信息	说明
ty (类型)	设备类型。
eq (设备序数)	设备序数。
state (操作状态)	设备的操作状态。请参阅第 198 页的“更改传输器的状态”。
status (设备状态)	传输器的状态。请参阅第 197 页的“查看状态信息”。
family_set (系列集)	传输器所属的系列集的名称。

您可以使用传输器按钮来全面核查所有介质、更改传输器的状态、导入介质和卸载介质。

VSN 目录

VSN Catalog (VSN 目录) 区域列出了选定传输器的 VSN。表 8-6 列出了可用于每一个传输器的信息。

表 8-6 VSN 目录显示区域

信息	说明
slot (端口)	介质的端口编号。
access_time (访问时间)	上次访问该介质的时间。
barcode (条码)	介质的条码。
count (次数)	介质被访问的次数。
use (占用)	该卷已用空间的百分比。
st (VSN 状态)	VSN 的状态。有关详细信息，请参阅第 197 页的“查看状态信息”。
ty (类型)	介质类型。
vsn	卷序列名。

VSN 显示区域中还包括所有 VSN 的访问时间或条码信息。您可以使用 VSN 操作按钮来核查、导出、安装、卸载、标记和移动卷。

设备

Devices (设备) 区域显示了选定传输器的设备信息。表 8-7 列出了显示的信息。

表 8-7 设备显示区域

信息	说明
ty (类型)	设备类型。
eq (设备序数)	设备序数。
status (设备状态)	设备的状态。请参阅第 197 页的“查看状态信息”。
act (计数)	活动计数。
use (占用)	装入设备的卷中已用空间的百分比。

表 8-7 设备显示区域 (续)

信息	说明
state (操作状态)	设备的操作状态。
vsn	介质的卷序列名。
slot (端口)	介质的端口编号。

若要控制设备，请参阅第 205 页的“使用 devicetool(1M)”。

查看状态信息

表 8-8 介绍了各个状态位。

表 8-8 状态位

状态位	对于设备的含义	对于文件系统的含义
s-----	正在扫描介质。	
m-----	当前已安装文件系统。	
M-----	维护模式。	
-E-----	设备在扫描期间收到无法恢复的错误。	
-a-----	设备处于核查模式。	正在存档文件系统。
--l-----	介质具有标签。	
--N-----	Sun SAM-FS 或 Sun SAM-QFS 环境无法识别介质。	
---I-----	正在等待设备空闲。	
---A-----	需要操作员看管。	
----C-----	正在清洁卡盒。	
----U-----	已请求卸载。	
----R-----	已保留设备。	
-----w---	进程正在介质上写入数据。	
-----o--	设备已打开。	
-----P-	设备正在定位 (仅适用于磁带)。	
-----F-	已占用所有存储端口 (仅适用于传输器状态)。	

表 8-8 状态位 (续)

状态位	对于设备的含义	对于文件系统的含义
-----w	设备就绪，并且介质已写保护。	
-----R	设备就绪，并且介质只读。	
-----r	设备已装载并准备就绪。	正在释放文件系统占用的磁盘空间。
-----p	设备已存在。	

管理传输器

本节介绍您可以对选定传输器执行的操作。表 8-9 列出了可行的操作。

表 8-9 管理传输器

操作	说明
全面核查	全面核查选定传输器中的所有卷。
更改状态	更改传输器的状态。
导入介质	将介质导入选定传输器。
卸载	从选定传输器中卸载所有介质。

▼ 执行全面核查

1. 从可用传输器列表中选择传输器。
2. 左击 **Full Audit (全面核查)** 按钮。系统会提示您确认操作。
系统将全面核查传输器中的每一个卷。

▼ 更改传输器的状态

1. 从可用传输器列表中选择传输器。
2. 执行以下其中一项操作：
 - 左击 **Change State (更改状态)** 按钮，将状态更改为 ON (打开)。

- 右击 **Change State**（更改状态）按钮，显示状态列表。表 8-10 中列出了某些可能的状态。

表 8-10 更改传输器状态

当前状态	可能的下一个状态
ON（打开）	IDLE（空闲），OFF（关闭）
IDLE（空闲）	IDLE（空闲）时自动转换为 OFF（关闭）
OFF（关闭）	DOWN（禁用），ON（打开）
DOWN（禁用）	OFF（关闭）

导入和导出介质

▼ 将介质导入传输器

1. 从可用传输器列表中选择传输器。
2. 单击 **Import Media**（导入介质）按钮。
3. 将卡盒放入传输器的邮箱中。

系统将通知传输器接受放入传输器邮箱中的卡盒。选择 **Import Media**（导入介质）之后，您随后可以将卡盒放入邮箱中。如果在 30 秒内未放入卡盒，系统将终止导入操作。

▼ 从传输器导出卡盒

1. 从可用传输器列表中选择传输器。
2. 选择要从中导出卡盒的端口。
3. 单击 **Export Media**（导出介质）按钮。

系统会通知传输器将选定的卡盒放入传输器邮箱中。

注 – 只有在传输器设备配有邮箱时，才能导入和导出卡盒。

载入和卸载存储匣

▼ 载入存储匣

1. 从可用传输器列表中选择传输器。

选定的传输器中当前不可包含存储匣。

2. 单击 Load（载入）按钮。

系统将通知传输器载入存储匣。

▼ 卸载存储匣

1. 从可用传输器列表中选择传输器。

选定的传输器中当前必须包含存储匣。

2. 单击 Unload（卸载）按钮。

系统将通知传输器卸载存储匣。

注 – 只有在选定的传输器支持载入和卸载存储匣时，才能载入和卸载存储匣。

使用卷

选定传输器之后，此传输器的所有卷将显示在屏幕中部的 VSN 目录中。本部分提供了卷的使用说明。

▼ 显示条码而不显示访问时间

目录显示区域包含选定传输器中每个端口的信息。您可以在目录显示区域中显示访问时间或条码。默认情况下，系统显示访问时间。

● **若要显示条码而不显示访问时间，请右击“显示”按钮并选择条码。**

此时，系统将显示条码而不显示访问时间。

▼ 查找 VSN

若要按 VSN 搜索并选择卷，请执行以下步骤。

- 在 **Find VSN (查找 VSN)** 字段中键入 **VSN 名称或起始样式**，然后按回车键进行查找。

如果系统找到您指定的 VSN 样式，则会选择第一个符合指定样式的 VSN。若要查找符合指定样式的下一个 VSN，请单击 **Find Next (查找下一个)** 按钮。如果未找到 VSN，则会生成一则错误消息。

使用样式匹配方法搜索 VSN 时，如果输入的长度为 n 个字符的样式与某个 VSN 的前 n 个字符完全匹配，则系统会选择该 VSN。

▼ 查找特定端口编号中的 VSN

- 在 **Find Slot (查找端口)** 字段中键入 **端口编号**。

您还可以使用向上和向下按钮来增加或减少端口编号。如果未找到 VSN，则会生成一则错误消息。

若要转至下一个端口编号，请单击 **Find Next (查找下一个)** 按钮。

▼ 选择 VSN

- 左击您想选择的 VSN。

▼ 核查 VSN

若要核查选定的 VSN，请执行以下步骤。

1. 选择您想核查的 VSN。
2. 左击 **Audit (核查)** 按钮。

系统将读取该 VSN 并更新端口的目录条目。

注 – 若要核查传输器中的每个 VSN，请在传输器显示区域中选择传输器，然后单击 **Full Audit (全面核查)** 按钮。

▼ 导出卷

1. 选择您想导出的 VSN。您所选择的 VSN 当前必须在传输器中。
2. 左击 **Export (导出)** 按钮。

传输器将移除 VSN 并将其放入传输器邮箱中。

▼ 载入卷

1. 选择您想载入的 VSN。
2. 左击 **Mount (安装)** 按钮。

传输器会将选定的 VSN 装入其中一个传输器设备。

▼ 卸载卷

1. 选择您想卸载的 VSN。
2. 左击 **Unload (卸载)** 按钮。

传输器将从传输器设备中卸载选定的 VSN，并将其放回端口中。

▼ 标记卷

在 Sun SAM-FS 和 Sun SAM-QFS 环境中，已标记的卷用于区分不同的卡盒。软件标签为 Sun SAM-FS 和 Sun SAM-QFS 软件提供重要信息，其中包括 VSN 的名称以及在卡盒上开始写入数据的位置。



注意 – 由于标记卷会导致先前写入该卷的数据丢失，因此，请在继续之前确认您的操作。

若要标记卷，请执行以下步骤。

1. 选择您想标记的 VSN。
2. 左击 **Label (标签)** 按钮。

图 8-5 显示了出现的对话框。

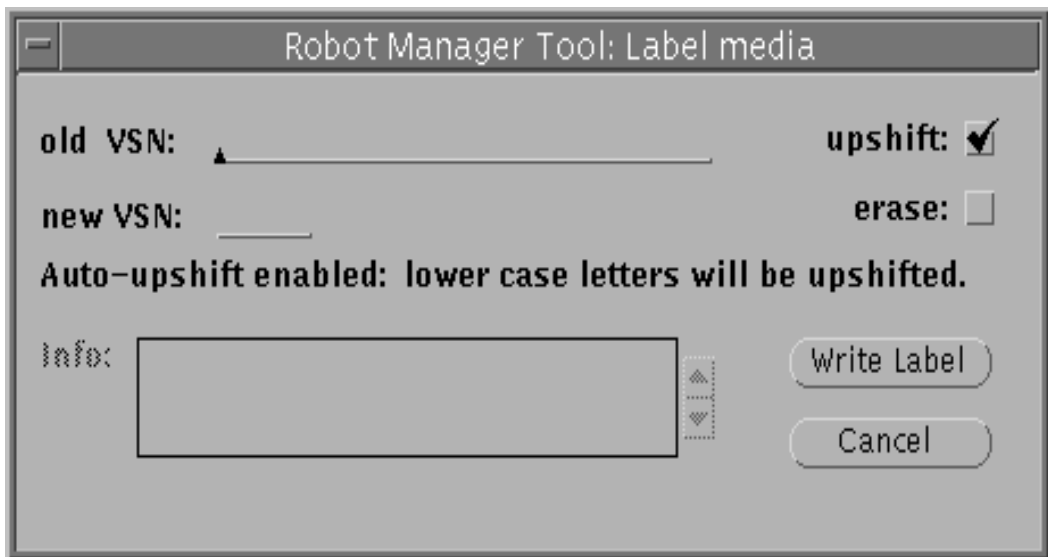


图 8-5 robottool(1M) — 介质标记屏幕

3. 执行以下其中一项操作：

对于旧 VSN，如果您要重新标记卷，请键入旧 VSN。旧 VSN 必须与卷的当前 VSN 完全匹配。如果您希望工具自动将小写字母转换为大写字母，请单击 Upshift（大写转换）框。如果您要重新标记磁带并选择了 Upshift（大写转换），则旧 VSN 的大小写可以与磁带当前的 VSN 不同。

对于新 VSN，请键入一个新 VSN。对于光学介质，VSN 标签可以包含多达 31 个字符。对于其它所有介质，VSN 标签可以包含多达 6 个字符。对于光学介质，您可以在 Info（信息）窗口中键入多达 128 个字符以包括在标签中。

4. 如果您希望在标记操作期间清除介质，请单击 erase（清除）框。

清除介质可能需要很长时间。请注意，系统会在标记操作期间清除数据。清除将覆盖卷上的每个扇区。

5. 单击 Write Label（写入标签）按钮。

6. 如果检测到错误，则 Label media（标记介质）窗口 Info（信息）框的上方会出现错误检查框及有关消息。

要确认错误，请单击检查框，错误消息即会被删除。

可能的错误包括：VSN 无效，或旧 VSN 与选定端口中卷的 VSN 不匹配。

▼ 移动介质

要将卷移至另一端口，请执行以下步骤。

1. 选择您想移动的卷。
2. 左击 **Move（移动）** 按钮。

图 8-6 显示了出现的对话框。



图 8-6 robottool(1M) — 介质移动屏幕

3. 在此框中输入以下内容。

对于 **Source slot**（来源端口），请通过双击端口编号并键入新编号，或通过使用退格键删除编号并键入新编号，输入新的来源端口编号。您指定的来源端口中必须包含卷。默认情况下，**Source slot**（来源端口）字段中包含您所选定卷的端口编号。

对于 **Destination slot**（目标端口），请键入新的目标端口编号。您指定的端口编号必须可用。

4. 单击 **Move（移动）** 按钮。
5. 如果检测到错误，**Move media（移动介质）** 窗口中会出现错误检查框及有关消息。

要确认错误，请单击检查框，错误消息即会被删除。

可能的错误包括：未指定来源或目标端口，或指定的端口无效。有效端口编号必须是一个整数，并且必须大于或等于零且小于传输器目录中的条目数。

查看 VSN 状态信息

st 栏用于显示目录条目的状态。表 8-11 列出了可能的状态位。

表 8-11 VSN 状态位

状态位	含义
A---	卷需要核查。
R---	卷标记为回收。
W---	卷已经写保护。
-E--	介质错误。
-X--	这是导出端口。
-r--	卷标记为只读。
--u-	端口不可用。
--l-	已标记卷。
--N-	Sun SAM-FS 或 Sun SAM-QFS 环境无法识别卷。
---c	正在清洁。
---p	端口已占用。

查看设备信息

robottool 底部的第三个面板显示了与选定传输器相关的设备。其中显示的只是参考性信息，您不能在设备上执行操作。若要管理单个非传输器设备，请使用 devicetool。其中显示的信息与 devicetool 特定介质显示屏幕中显示的信息相同。

使用 devicetool(1M)

devicetool(1M) 程序是一个用于查看 Sun SAM-FS 和 Sun SAM-QFS 关联设备的信息并管理这些设备的 GUI 工具。

▼ 启动 devicetool(1M)

要启动 devicetool，请在操作系统提示符下输入下面的命令：

```
# devicetool&
```

图 8-7 显示了 devicetool(1M) 的初始显示屏幕。

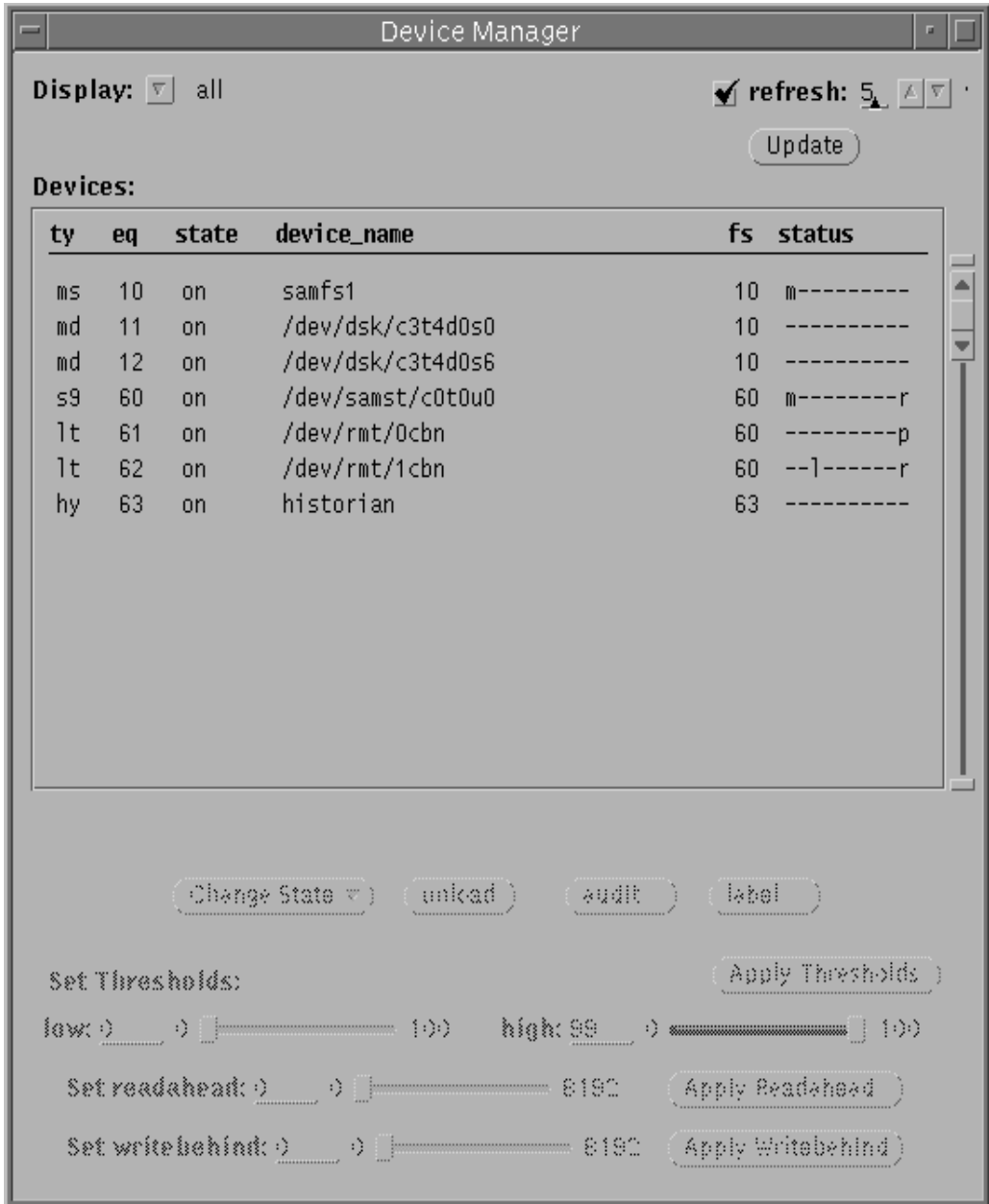


图 8-7 devicetool(1M) — 初始屏幕

▼ 更改显示格式

devicetool(1M) 显示了 Sun SAM-FS 或 Sun SAM-QFS 环境中配置的各个设备。这些设备显示在屏幕中央的滚动列表中。默认情况下，系统会显示所有设备。

若要更改显示格式，请执行以下步骤。

1. 右击 **Display (显示)** 按钮上的 **MENU (菜单)**。
屏幕上会显示下拉菜单。
2. 选择某个显示选项。
3. 如果已选择 **Removable Media (可移动介质)** 或 **Manual Only (仅限于手动载人设备)**，则会显示 **Media (介质)** 下拉菜单。
若要进一步限制显示的内容，请右击 **Media (介质)** 按钮。屏幕上会显示下拉菜单。
4. 选择要显示的介质类型。

▼ 查看显示字段

某些显示字段显示在所有显示屏幕中，而其它一些字段只显示在特定格式的数据显示屏幕中。表 8-12 按字母顺序列出了所有可能显示的字段。

表 8-12 显示字段

字段	说明
act (计数)	活动次数。打开卷的次数。
device_name (设备名)	分配给驱动器的名称。
eq (设备序号)	装置的设备序号。
family_set (系列集)	设备所属系列集的名称。
free (可用空间)	可用磁盘空间包含的 1024 字节块数。
fs (文件系统)	设备所属文件系统的名称。
low/high (下限/上限)	磁盘使用阈值百分比的下限和上限。
ord (序号)	磁盘设备在存储系列集中的序号。
ra (读前区)	此文件系统的最大读前区，以 KB 为单位。它被圆整至 8 KB 的倍数。

表 8-12 显示字段 (续)

字段	说明
state (操作状态)	设备的当前操作状态。有效设备状态包括： <ul style="list-style-type: none"> • ready (就绪) — 设备已打开，并且传输设备中载入的磁盘或磁带可供访问。 • notrdy (未就绪) — 设备已打开，但是传输设备中无磁盘和磁带。 • idle (空闲) — 设备不能用于新请求。进行中的操作会继续进行，直到完成。 • off (关闭) — 设备不可进行访问。 • down (禁用) — 设备只能进行维护访问。
status (设备状态)	设备状态。
ty (类型)	设备类型。
used (已用空间)	已用空间的百分比。
vsn	分配给卷的卷序列号，如果未标记卷，则为卷分配 nolabel。如果传输设备中无卡盒或设备已关闭，则此字段为空。
wb (写后区)	此文件系统的最大写后区，以 KB 为单位。

控制设备

以下几节介绍控制设备的说明。它们包含以下主题：

- 更改设备状态
- 卸载设备
- 核查设备中的卷
- 标记设备中的卷
- 设置阈值
- 设置读前区和写后区

▼ 选择设备

- 要从显示屏幕中选择某个设备，请单击包含该设备的行中的 **SELECT (选择)**。

选择设备之后，显示屏幕的下部将显示适用于该设备类型的操作按钮。您可以更改设备的状态以及卸载、核查和标记设备，这取决于所选的设备。

▼ 更改设备状态

您可以使用 **Change State**（更改状态）按钮更改设备的状态。如果单击此按钮上的 **SELECT**（选择），则会导致选定设备的默认状态，即打开状态。如果单击此按钮上的 **MENU**（菜单），屏幕上将显示 **Change State**（更改状态）菜单，您可以从中选择设备状态。可能的操作状态包括打开、空闲、关闭和禁用。若要更改设备的状态，请执行以下步骤。

1. 在可用设备列表中选择设备。
2. 执行以下其中一项操作：
 - 左击 **Change State**（更改状态）按钮，将状态更改为 **ON**（打开）。
 - 右击 **Change State**（更改状态）按钮，显示状态列表。表 8-13 列出了可能的设备状态。

表 8-13 可能的设备状态

当前状态	可能的下一个状态
ON（打开）	IDLE（空闲），OFF（关闭）
IDLE（空闲）	IDLE（空闲）时自动转换为 OFF（关闭）
OFF（关闭）	DOWN（禁用），ON（打开）
DOWN（禁用）	OFF（关闭）

▼ 卸载设备

1. 选择您要卸载的设备。
2. 左击 **Unload**（卸载）按钮。

传输器将卸载选定的设备。

▼ 核查设备

1. 选择您要核查的设备。
2. 左击 **Audit**（核查）按钮。

系统将读取设备中的卷并更新库目录条目。

注 – 若要核查传输器中的每个 **VSN**，请在传输器显示区域中选择传输器，然后单击 **Full Audit**（全面核查）按钮。

▼ 标记设备中的卷

1. 选择您要标记介质的设备。
2. 左击 **Label**（标签）按钮。

图 8-8 显示了出现的对话框。

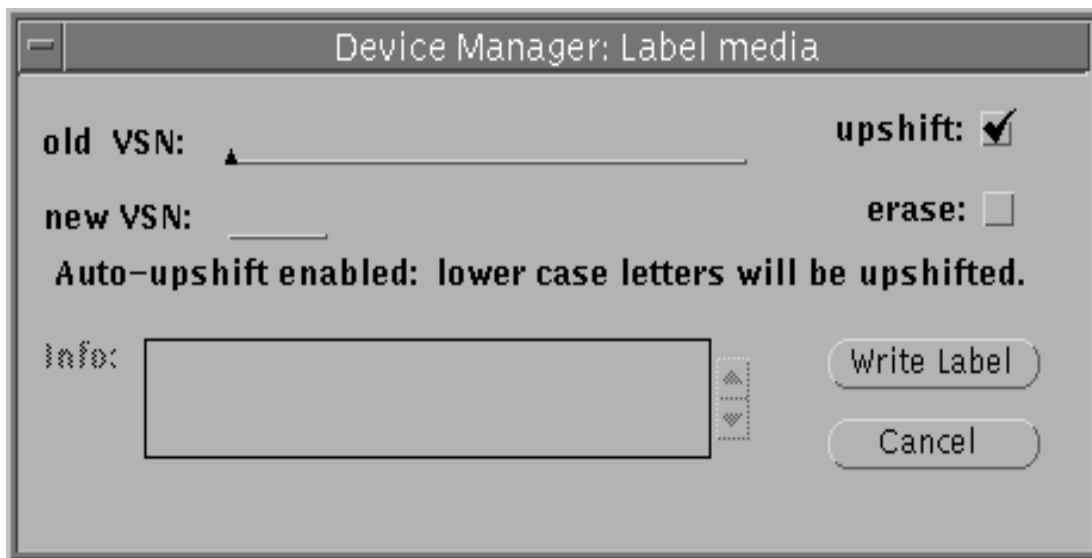


图 8-8 devicetool(1M) — 标记介质屏幕

3. 执行以下其中一项操作：

- 对于旧 VSN，如果您要重新标记卷，请键入旧 VSN。旧 VSN 必须与卷的当前 VSN 完全匹配。如果您希望工具自动将小写字母转换为大写字母，请单击 **upshift**（大写转换）框。如果您重新标记磁带并选择了 **upshift**（大写转换），则旧 VSN 的大小写可以与磁带当前的 VSN 不同。
- 对于新 VSN，请键入一个新 VSN。对于光学介质，VSN 标签可以包含多达 31 个字符，并且您可以在 **Info**（信息）窗口中键入多达 128 个字符以包括在标签中。对于其它所有介质，VSN 标签可以包含多达 6 个字符。

4. 如果您希望在标记操作期间清除卷，请单击 **erase**（清除）框。

清除介质可能需要很长时间。

5. 单击 **Write Label**（写入标签）按钮。

如果检测到错误，则 Label media（标记介质）窗口 Info（信息）框的上方会出现错误检查框及有关消息。要确认错误，请单击错误检查框，错误消息即会被删除。

可能的错误包括：VSN 无效，旧 VSN 与选定端口中介质的 VSN 不匹配。

注意 – 标记卷会造成写入该卷的所有数据丢失。

▼ 设置阈值

您可以通过执行以下步骤来设置磁盘集的高阈值和低阈值。

1. 选择您要设置阈值的磁盘集。
2. 键入所需的数值，分别指定低阈值或高阈值的使用百分比。
此外，您还可以使用滑标条来增加或减少数值。
3. 单击 **Apply Thresholds（应用阈值）** 按钮上的 **SELECT（选择）**。
除非发生更改或重新安装文件系统，否则新阈值会一直有效。

▼ 设置读前区和写后区

您可以在磁盘集上为读前区和写后区设置连续的 1KB 数据块的最大数量。若要设置读前区和写后区，请执行以下步骤。

1. 选择您要设置读前区或写后区的磁盘集。
2. 键入所需的数值，分别为读前区或写后区指定连续的 1KB 数据块的数量。
此外，您还可以使用相应的滑标条来增加或减少数值。
3. 单击 **Apply Readahead（应用读前区）** 或 **Apply Writebehind（应用写后区）** 按钮上的 **SELECT（选择）**。
除非发生更改或重新安装文件系统，否则读前区或写后区的新设置会一直有效。

使用 previewtool(1M)

previewtool(1M) 程序可以使您查看和管理未完成的安装请求。起初，显示屏幕的安装请求窗口中将显示所有未完成的安装请求。信息显示在滚动列表中。通过抓取窗口的边角并进行拉伸或收缩，您可以调整窗口的大小，使其显示 1 至 18 个安装请求。

图 8-9 显示了 previewtool(1M) 的初始显示屏幕。

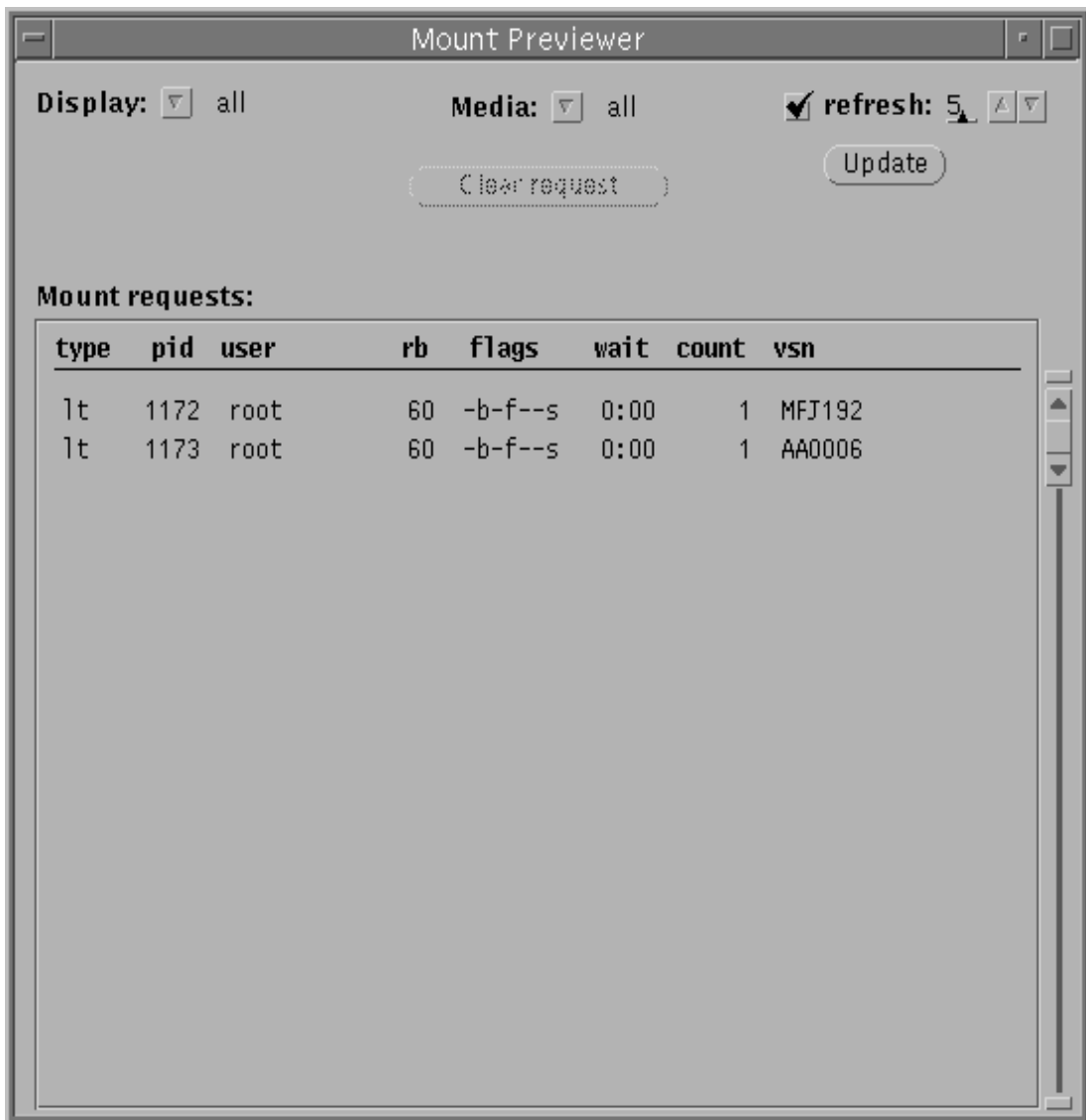


图 8-9 previewtool(1M) — 初始屏幕

▼ 更改显示格式

1. 右击 **Display (显示)** 按钮。
2. 选择其中一个显示选项。
3. 选择显示类型，如下所述：
 - 如果选择特定的传输器，屏幕上会显示可用传输器的菜单。选择您要显示的传输器。
 - 如果未选择特定的传输器，您可以使用介质类型来进一步限制显示的内容。右击 **Media (介质)** 按钮。选择您要限制显示的介质类型。

previewtool(1M) 显示屏幕字段

表 8-14 列出了 `previewtool(1M)` 显示屏幕的字段。

表 8-14 `previewtool(1M)` 字段

字段	说明
slot (端口)	卷的端口编号。
type (类型)	分配给卷的设备类型代码。
pid	UNIX 进程标识。进程标识 1 表示 NFS 访问。
user (用户)	请求安装的用户姓名。
rb	请求的 VSN 所在的传输器的设备序数。
flags (标记)	有关 flags (标记) 字段的说明，请参阅表 8-15。
wait (等待时间)	从收到安装请求到现在所经历的时间。如果等待时间超过一天，则以天为单位显示等待时间；否则以 <i>hh:mm</i> 格式显示等待时间。
count (计数)	如果请求是登台安装，则显示请求此 VSN 的次数。
vsn	分配给介质的卷序列号。

表 8-15 说明了 flags (标记) 字段。

表 8-15 标记定义

标记	含义
W-----	已请求写访问
-b-----	条目正忙
--C----	已请求清除 VSN
---f---	已请求文件系统
----B--	使用块 I/O 进行数据传输
-----S-	已安装另一面
-----s	登台请求标记

▼ 清除安装请求

1. 选择您要清除请求的 VSN。
2. 左击 **Clear request** (清除请求) 按钮。

使用 samu(1M) 操作员实用程序

本章介绍如何通过 samu(1M) 操作员实用程序来控制 Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 环境中配置的设备。虽然 Sun QFS 环境并非支持所有的 samu(1M) 显示屏幕，但为了完整起见，本章介绍了三类环境。

本章包括下列主题：

- 第 217 页的“概述”
- 第 222 页的“操作员显示屏幕”
- 第 242 页的“操作员显示屏幕状态代码”
- 第 244 页的“操作员显示屏幕设备状态”
- 第 245 页的“操作员命令”

概述

samu(1M) 操作员实用程序需要一个最低分辨率为 80 字宽乘以 24 行的显示终端。该实用程序包括以下功能：

- 显示屏幕，可以使您监控 Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 设备以及文件系统活动
- 一组命令，可以使您选择显示屏幕、设置显示选项、控制对设备的访问、控制设备活动以及抓取显示窗口的快照

本章所示的显示窗口都是典型的示例。显示终端具体显示的格式和信息量取决于显示终端机型以及 Sun QFS、Sun SAM-FS 或 Sun SAM-QFS 环境中配置的设备。

在 samu(1M) 中执行的操作也可以使用 samcmd(1M) 命令执行。有关 samcmd(1M) 的详细信息，请参阅 samcmd(1M) 联机资料。

以下几节介绍如何启动和停止 samu(1M)、与实用程序交互操作、访问帮助窗口以及查看操作员显示屏幕。

▼ 启动 samu(1M)

1. 若要启动 `samu(1M)`，请在 UNIX 命令行中输入 `samu(1M)` 命令，如下所示：

```
# samu
```

系统将启动 `samu(1M)` 并显示帮助显示屏幕。

2. 按 **CTRL-f** 移动到下一个帮助屏幕，其中列出了用于控制显示的按键。

`samu(1M)` 命令接受其命令行中的选项，其中包括用于选择初始显示屏幕的选项。有关 `samu(1M)` 命令行选项的详细信息，请参阅 `samu(1M)` 联机资料。

注 – 与 `vi(1)` 编辑器相似，`samu(1M)` 基于 `curses(3X)` 库例程。启动 `samu(1M)` 之前，您必须正确定义您的显示终端类型。

▼ 停止 samu(1M)

- 若要退出 `samu(1M)`，请输入以下一项：
 - 按 `q` 键
 - 输入 `:q`

`samu(1M)` 操作员实用程序会退出并返回至命令 `shell`。

与 samu(1M) 交互操作

与 `samu(1M)` 交互操作在以下方面类似于与 UNIX `vi(1)` 编辑器交互操作：向前或向后调页、输入命令、刷新显示屏幕以及退出实用程序。

查看操作员显示屏幕时，您可以使用表 9-1 中列出的按键来控制显示。这些按键的具体功能取决于所查看的显示屏幕。有关在特定显示屏幕上操作按键的信息，请参阅 samu(1M) 联机资料。

表 9-1 samu(1M) 显示控制组合键

按键	功能	显示屏幕
CTRL-b	上一个文件系统	: a, a
	向前翻页	c, h, o, p, s, t, u, v, w
CTRL-d	向后翻半页	c, p, s, u, w
	下一个传输器目录	v
	向后翻页 (顶部)	h
	向后翻页 (底部)	a
CTRL-f	下一个文件系统	: a, a
	向后翻页	c, h, o, p, s, t, u, v, w
CTRL-k	选择 (手动, 自动, 两者全选, 优先级)	p
	高级排序键	v
	切换路径显示	n, u, w
CTRL-u	向前翻半页	c, p, s, u, w
	上一个传输器目录	v
	向前翻页 (顶部)	h
	向前翻页 (底部)	a
CTRL-i	2 行详细显示格式	v
1-7	选择排序键, 如下: <ul style="list-style-type: none"> • 1, 按端口排序。 • 2, 按计数排序。 • 3, 按使用排序。 • 4, 按 VSN 排序。 • 5, 按访问时间排序。 • 6, 按条码排序。 • 7, 按标签时间排序。 	v
/	搜索 VSN	v
%	搜索条码	v

命令和显示错误消息显示在显示窗口的最后一行。如果出现命令错误，系统会中止自动刷新显示屏幕功能，直到操作员进行下一项操作。

输入设备

Sun QFS、Sun SAM-FS 或 Sun SAM-QFS 环境中的每一个设备在 `mcf` 文件中均分配有一个设备序数（如 10）。许多 `samu(1M)` 命令只应用于特定设备。

示例 1: `:off` 命令的语法格式如下：

```
:off eq
```

对于 `eq`，输入您要指定的设备的设备序数。

示例 2: 有时，`samu(1M)` 会提示您输入设备。当您访问传输器目录显示屏幕（本章随后会加以说明）时，它会提示您输入传输器的设备序数：

```
Enter robot:
```

在提示时，请输入设备序数，或按回车键，选择上一个使用的设备。

获取联机帮助

当您启动 `samu(1M)` 时，系统会自动显示第一个帮助屏幕。该帮助屏幕可能会有所不同，具体取决于您是否具有 Sun QFS、Sun SAM-FS 或 Sun SAM-QFS 文件系统。共有五页帮助屏幕，但本手册只显示第一页帮助屏幕。随后的帮助屏幕显示了 `samu(1M)` 命令。

代码示例 9-1 显示了用于 Sun SAM-FS 和 Sun SAM-QFS 文件系统的初始帮助屏幕。

代码示例 9-1 Sun SAM-FS 和 Sun SAM-QFS `samu(1M)` 初始帮助屏幕

```
Help information           page 1/5   samu 4.0-x Thu Oct 11 13:22:30

Displays:
  a Archiver status          v Robot catalog
  c Device configuration     w Pending stage queue
  d Daemon trace controls   C Memory
  f File systems             F Optical disk label
  h Help information         I Inode
  l License information      J Preview shared memory
  m Mass storage status     L Shared memory tables
  n Staging status          M Shared memory
  o Optical disk status     N File System Parameters
  p Removable media load requests R SAM-Remote
  r Removable media         S Sector data
```

代码示例 9-1 Sun SAM-FS 和 Sun SAM-QFS samu(1M) 初始帮助屏幕 (续)

```
s Device status                T SCSI sense data
t Tape drive status           U Device table
u Staging queue

more (ctrl-f)
```

代码示例 9-2 显示了用于 Sun QFS 文件系统的初始帮助屏幕。

代码示例 9-2 Sun QFS samu(1M) 初始帮助屏幕

```
Help information                page 1/5  samu 4.0-x Thu Oct 11 13:58:20

Displays:
  d Daemon trace controls      m Mass storage status
  f File systems                C Memory
  h Help information            I Inode
  l License information         N File System Parameters

more (ctrl-f)
```

要从一个屏幕向前或向后移动到下一个屏幕，请输入下面的组合键：

- 按 CTRL-f 向后翻至下一显示屏幕。
- 按 CTRL-b 向前翻至上一显示屏幕。

您随时可以按 h 键返回至帮助显示屏幕。

注 – 本手册并未介绍大写 samu(1M) 显示项目 (A、C、F、I、J、L、M、N、R、S、T 和 U)，因为它们只有在技术支持人员的协助下才能在客户站点上使用。

操作员显示屏幕

您可以通过按与每个显示屏幕对应的按键来查看各个 `samu(1M)` 操作员显示屏幕。从 `a` 到 `w` 的小写字母键用于显示操作信息。

对于溢出屏幕区域的显示，屏幕的底部会显示字样 `more`，表示显示还包含其它信息。代码示例 9-3 包含字样 **more**，表示随后的屏幕上还有其它信息。

代码示例 9-3 表示还存在其它文本的 `samu(1M)` 屏幕

```
xb54 54  exb8505  pt03  0  yes  2  0  on
lt55 55  dlt2000  pt02  1  yes  4  0  on  ml65
hp56 56  hpc1716  pt01  1  yes  3  0  on  hp70
hp57 57  hpc1716  pt01  1  yes  4  0  on  hp70
more
```

如果 `samu(1M)` 提示您输入设备，请输入相关的设备序数。配置显示屏幕 (c) 中显示了所有设备的设备序数。若要控制所有显示，请使用控制键。

以下几节介绍了操作员显示屏幕。这些小节不仅提供了示例，而且还在显示屏幕示例的后面附带了用于说明显示屏幕中字段的表格（如有必要）。

(a) — 存档程序状态显示屏幕

存档程序显示屏幕按文件系统显示了存档程序的状态。

显示屏幕范例

代码示例 9-4 显示了单个文件系统的活动和统计信息。

代码示例 9-4 samu(1M) a 显示屏幕

```
Archiver status samu 4.0.x Fri Jan 04 14:08:45

sam-archiverd: Archiving files

sam-arfind: samfs1 mounted at /sam1
Sleeping until Fri Jan 04 14:10:26 2002

sam-arcopy: samfs1 arset0.2.9360 mo.opt06a
Copying file testdir0/filewh
```

字段说明

若要查看存档程序的详细显示屏幕，请输入 `:a filesystem`。表 9-2 列出了详细显示屏幕中的字段。

表 9-2 samu(1M) a 显示屏幕字段说明

字段	说明
samfs1 mounted at	安装点。
regular files	标准文件的数量和大小
offline files	脱机文件的数量和大小
archdone files	archdone 文件的数量和大小表示存档程序已完成处理过程，并且无需进一步处理 archdone 文件。但是，请注意 archdone 文件尚未存档。
copy1	文件的数量和存档副本 1 的总容量。
copy2	文件的数量和存档副本 2 的总容量。
copy3	文件的数量和存档副本 3 的总容量。
copy4	文件的数量和存档副本 4 的总容量。
Directories	目录的数量和总容量。
sleeping until	表示存档程序再次运行的时间。

(c) — 设备配置显示屏幕

配置显示屏幕显示了配置的连通性。若要查看配置显示屏幕，请按 **c** 键。

显示屏幕范例

代码示例 9-5 显示了设备配置显示屏幕。

代码示例 9-5 samu(1M) c 显示屏幕

```
Device configuration:      samu 4.0.x Thu Oct 11 13:10:23

ty  eq  state  device_name      fs family_set
ae  60  on    /dev/samst/c0t0u0  60 m160
at  61  on    /dev/rmt/0cbn     60 m160
at  62  on    /dev/rmt/1cbn     60 m160
at  63  on    /dev/rmt/3cbn     60 m160
at  64  on    /dev/rmt/4cbn     60 m160
hy  65  on    historian          65
```

字段说明

表 9-3 列出了该显示屏幕的字段说明。

表 9-3 samu(1M) c 显示屏幕字段说明

字段	说明
ty	设备类型。
eq	设备的设备序数（主配置文件中定义的唯一编号）。
state	设备的当前操作状态。有效设备状态包括： <ul style="list-style-type: none">• on — 设备可以进行访问。• ro — 设备可以进行只读访问。• off — 设备不可访问。• down — 设备只能进行维护访问。• idle — 设备不可用于新连接。进行中的操作会继续进行，直到完成。
device_name	设备的路径。
fs	系列集的设备序数。
family_set	设备所属的存储系列集或库的名称。

(d) 一 后台跟踪控制显示屏幕

后台跟踪控制显示屏幕显示了 `defaults.conf` 文件中指定跟踪的事件。有关启用跟踪文件的详细信息，请参阅 `defaults.conf(4)` 联机资料。

显示屏幕范例

代码示例 9-6 显示了跟踪文件信息。它包括所跟踪的后台程序的有关信息、跟踪文件的路径、所跟踪的事件以及有关跟踪文件大小和时段的信息。

代码示例 9-6 samu(1M) d 显示屏幕

```
Daemon trace controls      samu    4.0.5816 Fri Jan 18 10:42:02

sam-archiverd  /var/opt/SUNWsamfs/trace/archiver
                cust err misc files date module
                size    0    age 0

sam-catserverd /var/opt/SUNWsamfs/trace/catserver
                cust err fatal ipc misc proc queue ftp debug date module
                size    0    age 0

sam-fsd        /var/opt/SUNWsamfs/trace/fsd
                cust err fatal ipc misc proc queue ftp debug date module
                size    0    age 0

sam-ftpd       /var/opt/SUNWsamfs/trace/ftp
                cust err fatal ipc misc proc queue ftp debug date module
                size    0    age 0

sam-recycler   /var/opt/SUNWsamfs/trace/recycler
                cust err fatal ipc misc proc queue ftp debug date module
                size    0    age 0

sam-sharefsd   off

sam-stagerd    /var/opt/SUNWsamfs/trace/stager
                cust err misc proc files debug date module
                size    0    age 0
```

(f) — 文件系统显示屏幕

文件系统显示屏幕显示了 Sun QFS、Sun SAM-FS 或 Sun SAM-QFS 文件系统的组件。若要查看文件系统显示屏幕，请按 **f** 键。

显示屏幕范例

代码示例 9-7 显示了文件系统显示屏幕。

代码示例 9-7 samu(1M) f 显示屏幕

```
File systems                               samu   4.0.x Thu Oct 11 13:12:07

ty eq state      device_name      status high low mountpoint server
ms 1   on          samfs1 m----2----d  80% 70%   /samfs1
md 11  on /dev/dsk/c2t5d0s5
md 12  on /dev/dsk/c2t6d0s5
```

字段说明

表 9-4 列出了该显示屏幕的字段说明。

表 9-4 samu(1M) f 显示屏幕字段说明

字段	说明
ty	设备类型。
eq	设备的设备序数（主配置文件中定义的唯一编号）。
state	设备的当前操作状态。有效设备状态包括： <ul style="list-style-type: none">• on — 设备可以进行访问。• ro — 设备可以进行只读访问。• off — 设备不可访问。• down — 设备只能进行维护访问。• idle — 设备不能用于新的操作。进行中的操作会继续进行，直到完成。
device_name	文件系统名或设备的路径。
status	设备状态。有关状态代码的说明，请参阅第 242 页的“操作员显示屏幕状态代码”。
high	磁盘使用上限阈值百分比。
low	磁盘使用下限阈值百分比。
mountpoint	文件系统的安装点。
server	安装文件系统的主机系统的名称。

(1) — 许可证显示屏幕

许可证显示屏幕显示了 Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 软件的许可证和到期日期。若要查看许可证显示屏幕，请按 1 键。

显示屏幕范例

代码示例 9-8 显示了许可证显示屏幕的示例。

代码示例 9-8 samu(1M) 1 显示屏幕

```
License Information samu 4.0.x Thu Oct 11 13:13:11

hostid = xxxxxxxx

License never expires
Remote sam server feature enabled
Remote sam client feature enabled
Migration toolkit feature enabled
Fast file system feature enabled
Data base feature enabled
Direct media access feature enabled
Shared SAN filesystem support enabled
Segment feature enabled
Robot type ADIC 100 Library is present and licensed
    100 at slots present and licensed
Robot type DLT Tape Library is licensed
    100 lt slots licensed
Robot type IBM 3570 Changer is licensed
    100 i7 slots licensed
Robot type IBM 3584 Library is licensed
    100 li slots licensed
```

上面的显示屏幕范例显示了 Sun SAM-FS 文件系统的许可证信息。许可证信息来自以下文件中的许可证密钥：

```
/etc/opt/SUNWsamfs/LICENSE.4.0
```

其中显示了以下系统信息：

- 到期信息
- 主机 ID
- 已启用的 Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 产品及功能
- 设备/介质组合

(m) — 大容量存储设备状态显示屏幕

大容量存储设备状态显示屏幕显示了大容量存储文件系统及其成员驱动器的状态。若要查看大容量存储设备状态显示屏幕，请按 m 键。

显示屏幕范例

代码示例 9-9 显示了成员驱动器如何缩进一位并直接位于它们所属文件系统的下面。

代码示例 9-9 samu(1M) m 显示屏幕

```
Mass storage status                                samu 4.0.x Thu Oct 11 13:13:42

ty  eq  status      use  state  ord  capacity   free   ra  part  high  low
ms   1  m----2----d  21%  on                    8.402G  6.644G  1024  16   80%  70%
  md  11                21%  on           0   4.251G  3.372G
  md  12                21%  on           1   4.151G  3.272G
```

字段说明

表 9-5 列出了该显示屏幕的字段说明。

表 9-5 samu(1M) m 显示屏幕字段说明

字段	说明
ty	设备类型。
eq	大容量存储设备的设备序号。
status	设备状态。有关状态代码的说明，请参阅第 242 页的“操作员显示屏幕状态代码”。
use	已用磁盘空间的百分比。
state	大容量存储设备的当前操作状态。
ord	磁盘设备在存储系列集中的序号。
capacity	磁盘存储空间包含的 1024 字节块数。
free	可用磁盘空间包含的 1024 字节块数。
ra	读前区大小 (KB)。
part	部分登台大小 (KB)。
high	磁盘使用上限阈值百分比。
low	磁盘使用下限阈值百分比。

(n) — 登台状态显示屏幕

登台状态显示屏幕为所有介质显示了登台程序的状态。若要查看登台状态显示屏幕，请按 `n` 键。若要查看特定设备类型的状态，请输入 `:n media`，此处 *media* 是指介质类型。

显示屏幕范例

代码示例 9-10 samu(1M) n 显示屏幕

```
Staging status                samu   4.0.x Thu Oct 11 13:14:23

Log output to:

Stage request: at.000004
Copying file /samfs1/testdir3/fileia

Stage request: at.000002
Copying file /samfs1/testdir1/fileei

Stage request: at.000003
Positioning for file /samfs1/testdir2/fileaa
```

(o) — 光盘状态显示屏幕

光盘状态显示屏幕显示了 Sun SAM-FS 或 Sun SAM-QFS 环境中配置的所有光盘驱动器的状态。若要查看光盘状态显示屏幕，请输入 :o。

显示屏幕范例

代码示例 9-11 samu(1M) o 显示屏幕

```
Optical disk status          samu      4.0.x Thu Oct 11 13:15:40

ty  eq  status      act  use  state  vsn
mo  35  --l---wo-r   1  29%  ready  oper2
```

字段说明

表 9-6 列出了该显示屏幕的字段说明。

表 9-6 samu(1M) o 显示屏幕字段说明

字段	说明
ty	设备类型。
eq	光盘的设备序号。
status	设备状态。有关状态代码的说明，请参阅第 242 页的“操作员显示屏幕状态代码”。
act	活动次数。
use	已用卡盒空间的百分比。
state	光盘的当前操作状态。有效设备状态包括： <ul style="list-style-type: none">• ready — 设备已打开，已将磁盘载入传输设备；可以进行访问。• notrdy — 设备已打开，但传输设备中没有磁盘。• idle — 设备不可用于新连接。进行中的操作会继续进行，直到完成。• off — 设备不可访问。• down — 设备只能进行维护访问。
vsn	分配给光盘的卷序列名，如果没有标记卷，则为卷分配关键字 nolabel。

(p) — 可移动介质载入请求显示屏幕

可移动介质载入请求显示屏幕列出了等待载入可移动介质的请求的有关信息。您既可以选择特定类型的介质（如 DLT 磁带），也可以选择介质系列（如磁带）。优先级显示屏幕列出了预备队列中的优先级（而不是用户的优先级），并按优先级对条目排序。

安装请求采用三种格式显示：手动和自动请求、仅手动请求或仅自动请求。

如果仅输入 `:p`，则显示当前选定的所有可移动设备的安装请求。

如果输入 `:p media_type`，则显示给定类型的可移动介质设备的安装请求。

若要选择手动 / 传输器显示屏幕或优先级显示屏幕，请按 `CTRL-k` 组合键。

显示屏幕范例 1

代码示例 9-12 samu(1M) p 显示屏幕 1

```
Removable media mount requests all both samu 4.0.x Fri Feb 9 11:21:42
count: 1
count type pid user rb flags wait count vsn
0 1t 473 root 40 Wb-f--- 0:00 TAPE0
```

显示屏幕范例 2

代码示例 9-13 samu(1M) p 显示屏幕 2

```
Removable media load requests all priority samu 4.0.x Mon Apr 26 21:44:27
License: License never expires. count: 3
index type pid priority rb flags wait count vsn
0 i7 0 3007 70 ---f--- 0:00 TAPE5
2 i7 0 0 70 ---f--- 0:00 TAPE1
99 i7 1383 -49607 70 W--f--- 0:06 TAPE14
```

字段说明

表 9-7 列出了该显示屏幕的字段说明。

表 9-7 samu(1M) p 显示屏幕字段说明

字段	说明
index	预备表中的索引号。
type	分配给可移动介质的设备类型代码。
pid	UNIX 进程标识。进程标识 1 表示 NFS 访问。
user	请求载入介质的用户的姓名。
priority	请求的优先级。
rb	被请求的 VSN 所在的传输器的设备序数。
flags	设备的标记。请参阅表 9-8。
wait	从收到安装请求到现在所经历的时间。
count	如果是一个登台请求，则表示请求此 VSN 的次数。
vsn	卷的卷序列名。

标记

表 9-8 列出了各个标记。

表 9-8 samu(1M) p 显示屏幕的标记字段

字段	说明
w-----	已请求写访问
-b-----	条目正忙
--C----	已请求清除 VSN
---f---	已请求文件系统
-----S-	已安装另一面
-----s	登台请求标记

(r) — 可移动介质状态显示屏幕

可移动介质状态显示屏幕可以使您监控磁带驱动器等可移动介质的活动。您既可以监控特定类型的设备（如录像带），也可以监控设备系列（如所有磁带设备）。

若要查看所有可移动介质设备的状态，请输入 `:r`。若要查看特定设备的状态，请输入 `:r dt`，此处 `dt` 是指设备。

显示屏幕范例

代码示例 9-14 samu(1M) r 显示屏幕

```
Removable media status: all      samu 4.0.x      Thu Oct 11 13:17:06

ty eq status      act use state vsn
at 61 --l----o-r  1 73% ready 000002
      0x541 blocks transferred
at 62 --l----o-r  1 70% ready 000004
      0x7da blocks transferred
at 63 --l----o-r  1 90% ready 000003
      0x2a0 blocks transferred
at 64 --l-----r  0 54% ready 000001
      idle
```

字段说明

表 9-9 列出了该显示屏幕的字段说明。

表 9-9 samu(1M) r 显示屏幕字段说明

字段	说明
ty	设备类型。
eq	驱动器的设备序数。
status	设备状态。有关状态代码的说明，请参阅第 242 页的“操作员显示屏幕状态代码”。
act	活动次数。

表 9-9 samu(1M) r 显示屏幕字段说明 (续)

字段	说明
use	已用卡盒空间的百分比 (仅限于光盘)。
state	可移动介质的当前操作状态。有效设备状态包括: <ul style="list-style-type: none">• ready — 设备已打开, 磁盘或磁带已载入传输设备; 可以进行访问。• notrdy — 设备已打开, 但传输设备中没有磁盘和磁带。• idle — 设备不可用于新连接。进行中的操作会继续进行, 直到完成。• off — 设备不可访问。• down — 设备只能进行维护访问。
vsn	分配给卷的卷序列名, 如果没有标记卷, 则为卷分配关键字 nolabel。如果传输设备中没有卷或设备已关闭, 则为空白。

(s) — 设备状态显示屏幕

设备状态显示屏幕显示了 Sun SAM-FS 或 Sun SAM-QFS 环境中配置的所有设备的状态。若要查看设备状态汇总显示屏幕，请输入 `:s`。

显示屏幕范例

代码示例 9-15 samu(1M) s 显示屏幕

```
Device status                samu  4.0.x      Thu Oct 11 13:18:18

ty  eq  state  device_name      fs  status      pos
ae  60  on    /dev/samst/c0t0u0  60  m-----r
      move complete
at  61  on    /dev/rmt/0cbn     60  --l----o-r
      0x70d blocks transferred
at  62  on    /dev/rmt/1cbn     60  --l----o-r
      0x986 blocks transferred
at  63  on    /dev/rmt/3cbn     60  --l----o-r
      0x46d blocks transferred
at  64  on    /dev/rmt/4cbn     60  --l-----r
      idle
hy  65  on    historian          65  -----
```

字段说明

表 9-10 列出了该显示屏幕的字段说明。

表 9-10 samu(1M) s 显示屏幕字段说明

字段	说明
ty	设备类型。
eq	设备的设备序数。
state	设备的当前操作状态。
device_name	设备的路径。对于文件系统设备，这是文件系统名称。
fs	设备所属的系列集的设备序数。
status	设备状态。有关状态代码的说明，请参阅第 242 页的“操作员显示屏幕状态代码”。
pos	设备的位置编号。

(t) — 磁带驱动器状态显示屏幕

磁带驱动器状态显示屏幕显示了 Sun SAM-FS 或 Sun SAM-QFS 环境中配置的所有磁带驱动器的状态。若要查看磁带驱动器状态显示屏幕，请按 t 键。

显示屏幕范例

代码示例 9-16 samu(1M) t 显示屏幕

```
Tape drive status          samu  4.0.x Thu Oct 11 13:18:48

ty  eq  status      act  use  state  vsn
at  61  --l----o-r   1  73% ready  000002
      0x7b7 blocks transferred
at  62  --l----o-r   1  70% ready  000004
      0xa35 blocks transferred
at  63  --l----o-r   1  90% ready  000003
      0x518 blocks transferred
at  64  --l----o-r   1  54% ready  000001
      0x20 blocks transferred
```

字段说明

表 9-11 列出了该显示屏幕的字段说明。

表 9-11 samu(1M) t 显示屏幕字段说明

字段	说明
ty	设备类型。
eq	驱动器的设备序数。
status	设备状态。有关状态代码的说明，请参阅第 242 页的“操作员显示屏幕状态代码”。
act	活动次数。
use	已用卡盒空间的百分比（仅限于光盘）。
state	可移动介质的当前操作状态。有效设备状态包括： <ul style="list-style-type: none">• ready — 设备已打开，磁盘或磁带已载入传输设备；可以进行访问。• notrdy — 设备已打开，但是传输设备中无磁盘和磁带。• idle — 设备不可用于新连接。进行中的操作会继续进行，直到完成。• off — 设备不可访问。• down — 设备只能进行维护访问。
vsn	分配给卷的卷序列名，如果没有标记卷，则为卷分配关键字 nolabel。如果传输设备中没有卷或设备已关闭，则为空白。

(u) — 登台队列显示屏幕

samu(1M) 实用程序的 u 显示屏幕列出了登台队列中的所有文件。若要选择该显示屏幕，请输入 u。按 CTRL-k 组合键可以列出每个条目第二行中的文件路径名。

显示屏幕范例

代码示例 9-17 samu(1M) u 显示屏幕

```
Staging queue by media type: all samu 4.0.x Thu Oct 11 13:19:34
volumes 2 files 827

ty      length  fseq   ino   position  offset  vsn
at      1.674M   1      2513   389d4     7e70b   000004
at      1.875M   1      2640   389d4     7f470   000004
at      1.643M   1      1536   389d4     80372   000004
at      1.063M   1      248    389d4     81099   000004
at      562.037k 1      595    389d4     8191b   000004
at      1.000M   1      142    389d4     81d81   000004
at      1.264M   1      442    389d4     82582   000004
at      599.014k 1      2237   389d4     82fa0   000004
at      816.685k 1      2435   389d4     83450   000004
at      1.429M   1      2701   389d4     83ab3   000004
at      1.752M   1      439    389d4     84623   000004
at      1.089M   1      565    389d4     85428   000004
at      975.326k 1      121    389d4     85ce1   000004
at      1.014M   1      28     389d4     86481   000004
at      683.581k 1      419    389d4     86c9f   000004
at      1.562M   1      1608   389d4     871f8   000004
more
```

字段说明

表 9-12 列出了该显示屏幕的字段说明。

表 9-12 samu(1M) u 显示屏幕字段说明

字段	说明
ty	设备类型。
length	文件的大小。
fseq	文件系统设备编号。

表 9-12 samu(1M) u 显示屏幕字段说明 (续)

字段	说明
ino	索引节点编号。
position	存档文件在特定介质上的位置 (以十进制格式表示)。
offset	存档文件在特定介质上的偏移量。
vsn	卷的卷序列名。

(v) — 传输器目录显示屏幕

传输器目录显示屏幕显示了当前编入传输器目录的所有磁盘或磁带的位置和 VSN。若要查看库 VSN 目录显示屏幕，请按 v 键。如果操作员实用程序提示您输入传输器名称，请输入设备名或设备序号。如果不输入任何条目，则显示上一次显示的库。有关所有设备的名称及设备序号，请按 c 键查看配置显示屏幕。

按 CTRL-k 组合键可以更改该显示屏幕的排序键。按 CTRL-i 组合键可以更改为显示时间和条码的双行显示屏幕。再次按 CTRL-i 组合键会在第二行显示卷保留信息。

显示屏幕范例

代码示例 9-18 samu(1M) v 显示屏幕

```
Robot VSN catalog by slot : eq 60 samu 4.0.x Thu Oct 11 13:20:04
                                count 32
slot      access time  count  use  flags          ty  vsn
   0  none                70   0%  -il-oCb----- at  CLN005
   1  2001/10/11 08:31    10  90%  -il---b----- at  000003
   2  2001/10/11 13:07    17  73%  -il---b----- at  000002
   3  2001/10/11 12:48    16  70%  -il---b----- at  000004
   4  2001/10/11 12:55    30  54%  -il---b----- at  000001
   5  none                 0   0%  -il-o-b----- at  000005
   6  none                 0   0%  -il-o-b----- at  000044
   7
  13  2001/10/11 13:05    61   0%  -il-o-b----- at  000033
```

字段说明

表 9-13 列出了该显示屏幕的字段说明。

表 9-13 samu(1M) v 显示屏幕字段说明

字段	说明
Robot VSN catalog	指定传输器的名称以及刷新显示屏幕的时间。
count	库中端口的数量。
slot	指定库中的端口编号。
access time	上次访问光盘的时间。
count	自从上次核查之后访问该卷的次数。
use	该卷已用空间的百分比。

表 9-13 samu(1M) v 显示屏幕字段说明 (续)

字段	说明
flags	设备的标记。有关标记的信息，请参阅表 9-14。
ty	设备类型。
vsn	卷的卷序列名。

标记

某些情况下，一个字段中可以出现不同的标记，并且一个标记可以改写另一个标记。表 9-14 显示了表 9-13 中 flags 字段的标记。

表 9-14 samu(1M) v 显示屏幕的标记字段

标记	说明
A-----	卷需要核查。
-i-----	端口在使用中。
--l-----	已标记。改写 N 标记。
--N-----	未标记。Sun SAM-FS 或 Sun SAM-QFS 环境无法识别该卷。
---E-----	介质错误。当 Sun SAM-FS 或 Sun SAM-QFS 软件在卡盒上检测到写错误时设置此标记。
----o-----	端口已被占用。
-----C-----	卷是清洁磁带。改写 p 标记。
-----p-----	优先 VSN。
-----b-----	已检测到条码。
-----w-----	写保护。当在卡盒上启用物理写保护机制时设置此标记。
-----R---	只读。
-----c--	回收。
-----d-	VSN 重复。改写 U 标记。
-----U-	卷不可用。
-----X	导出端口。

(w) — 待定登台队列显示屏幕

待定登台队列显示屏幕显示了尚未为其载入卷的登台请求队列。按 CTRL-k 组合键会在每个条目的第二行列出路径名。

显示屏幕范例

代码示例 9-19 samu(1M) w 显示屏幕

```
Pending stage queue by media type: all      samu      4.0.x Thu Oct 11 13:20:27
                                              volumes 1 files 13

ty      length  fseq  ino  position  offset  vsn
at      1.383M   1    42    3a786    271b   000002
at      1.479M   1    56    3a786    5139   000002
at     1018.406k  1    60    3a786    6550   000002
at      1.000M   1    65    3a786    7475   000002
at      1.528M   1    80    3a786    99be   000002
at      1.763M   1    92    3a786    ce57   000002
at      1.749M   1   123    3a786   11ece   000002
at      556.559k  1   157    3a786   1532f   000002
at      658.970k  1   186    3a786   17705   000002
at      863.380k  1   251    3a786   1dd58   000002
at      1.268M   1   281    3a786   1f2b7   000002
at      1.797M   1   324    3a786   23dfa   000002
at      1.144M   1   401    3a786   2bb6d   000002
```

字段说明

表 9-15 列出了该显示屏幕的字段说明。

表 9-15 samu(1M) w 显示屏幕字段说明

字段	说明
ty	设备类型。
length	文件的大小。
fseq	文件系统设备序号。
ino	索引节点编号。
position	存档文件在特定介质上的位置（以十进制格式表示）。
offset	存档文件在特定介质上的偏移量。
vsn	卷的卷序列名。

操作员显示屏幕状态代码

对于可移动介质设备显示屏幕和文件系统显示屏幕，操作员显示屏幕为它们提供了不同的状态代码。以下几节说明了这些显示屏幕。

可移动介质设备显示屏幕状态代码

c、o、r、s 和 t 操作员显示屏幕显示了可移动介质的状态代码。状态代码采用 10 位格式显示，按照从左（第 1 位）至右（第 10 位）的顺序读取。

本节介绍的状态代码不适用于 samu(1M) f、m 和 v 显示屏幕。有关 f 和 m 显示屏幕的状态代码的信息，请参阅第 243 页的“文件系统显示屏幕状态代码”。有关 v 显示屏幕的状态代码的信息，请参阅第 239 页的“(v) — 传输器目录显示屏幕”。

表 9-16 定义了每一状态位的有效状态代码。

表 9-16 可移动介质设备显示屏幕状态代码

状态位	对于设备的含义
s-----	正在扫描介质。
M-----	维护模式。
-E-----	设备在扫描期间收到无法恢复的错误。
-a-----	设备处于核查模式。
--l-----	介质具有标签。
--N-----	无法识别的介质。
---I-----	正在等待设备空闲。
---A-----	需要操作员看管。
----C-----	需要清洁。
----U-----	已请求卸载。
-----R----	已保留设备。
-----w---	进程正在介质上写入数据。
-----o--	设备已打开。
-----P-	设备正在定位（仅适用于磁带）。
-----F-	对于传输器，表示所有的存储端口均被占用。对于磁带和磁光盘驱动器，表示介质已满。
-----R	设备就绪，并且介质只读。

表 9-16 可移动介质设备显示屏幕状态代码 (续)

状态位	对于设备的含义
-----r	设备已装载并准备就绪。
-----p	设备已存在。
-----w	设备已被写保护。

文件系统显示屏幕状态代码

f 和 m 操作员显示屏幕显示了文件系统的状态代码。状态代码采用 11 位格式显示，按照从左（第 1 位）至右（第 11 位）的顺序读取。

本节所述的状态代码不适用于 samu(1M) c、o、r、s、t 或 v 显示屏幕。有关 c、o、r、s 和 t 显示屏幕的状态代码的信息，请参阅第 242 页的“可移动介质设备显示屏幕状态代码”。有关 v 显示屏幕的状态代码的信息，请参阅第 239 页的“(v) — 传输器目录显示屏幕”。

表 9-17 定义了每一状态位的有效状态代码。

表 9-17 文件系统显示屏幕状态代码

状态位	对于文件系统的含义
m-----	当前已安装文件系统。
M-----	当前正安装文件系统。
-u-----	当前正卸载文件系统。
--A-----	正在存档文件系统数据。
---R-----	正在释放文件系统数据。
----S-----	正在登台文件系统数据。
-----1----	Sun SAM-FS 或 Sun SAM-QFS 文件系统第 1 版。
-----2----	Sun SAM-FS 或 Sun SAM-QFS 文件系统第 2 版。
-----C----	Sun QFS 共享文件系统
-----W---	单个写入器。
-----R--	多个阅读器。
-----r-	mr 设备。
-----d	md 设备。

操作员显示屏幕设备状态

c、m、o、r、s 和 t 操作员显示屏幕显示了设备状态代码。这些代码表示设备的当前访问状态。

您可以使用 `samu(1M)` 更改设备的状态。以下示例显示了将驱动器的状态从 `down` 更改为 `on` 以及从 `on` 更改为 `down` 的过程：

- **示例 1：** 以下过程用于将设备的状态从 `down` 更改为 `on`：

`down -> off -> [unavail] -> on`

在本过程中，方括号表示不需要经过 `unavail` 状态。

- **示例 2：** 以下过程用于将设备的状态从 `on` 更改为 `down`：

`on -> [idle] -> [unavail] -> off -> down`

在本过程中，方括号表示不需要经过 `idle` 或 `unavail` 状态。

表 9-18 定义了有效的状态代码。

表 9-18 操作员显示屏幕设备状态

设备状态	说明
on	设备可以访问。对于某些显示屏幕，该状态可能由状态 <code>ready</code> 或 <code>notrdy</code> 代替。
ro	设备可以进行只读访问。与 <code>on</code> 类似，对于某些显示屏幕，该状态可能由 <code>ready</code> 或 <code>notrdy</code> 代替。
off	设备不可访问。对于磁带和光盘驱动器，造成设备处于 <code>off</code> 状态的可能原因包括： <ul style="list-style-type: none">• 设备需要清洁，但是未在自动化库中找到清洁卡盒。• 清洁卡盒无法载入驱动器，或者不能从驱动器卸载。• 在初始化期间发现驱动器已满，并且尝试清洁驱动器失败。• 系统无法从驱动器清除卡盒。• 未能在驱动器加速旋转期间打开驱动器以进行 I/O 操作。• 减慢驱动器转速以卸载驱动器时收到除 <code>NOT READY</code> 之外的错误。• 未能在驱动器加速旋转期间打开驱动器上的标准磁带驱动程序。
down	设备只能进行维护访问。
idle	设备不能用于新连接。进行中的操作会继续进行，直到完成。
ready	设备已打开，并且载入传输设备中的磁盘或磁带可供访问。
notrdy	设备已打开，但传输设备中没有磁盘或磁带。
unavail	设备不可访问并且不能用于自动 Sun SAM-FS 或 Sun SAM-QFS 操作。当设备处于不可用状态时，您可以继续使用装载和卸载命令在设备中放入和移出介质。

操作员命令

本部分介绍以下类型的操作员命令：

- 第 245 页的“存档程序命令”
- 第 246 页的“设备命令”
- 第 247 页的“显示控制命令”
- 第 249 页的“文件系统命令”
- 第 251 页的“传输器命令”
- 第 252 页的“其他命令”

注 – 如果您要在 Sun Solaris 操作环境 (OE) 命令行中输入任何操作员命令，则必须把它们当作 `samcmd(1M)` 命令的变量使用。有关 `samcmd(1M)` 命令的详细信息，请参阅 `samcmd(1M)` 联机资料。

当输入的 `samu(1M)` 命令以冒号 (:) 开头时，则表示该命令是命令行命令，而不是一组热键。

存档程序命令

表 9-19 列出了存档程序命令及其作用。

表 9-19 存档程序命令作用

命令	作用
<code>aridle</code>	在下一个方便的位置停止所有存档操作。例如，在当前 <code>tar(1)</code> 文件的末尾停止 <code>sam-arcopy</code> 操作。例如，在卸载文件系统之前，用户可以使用该命令停止所有文件系统的全部存档活动。
<code>arrestart</code>	中断存档程序的运行并重新启动存档程序。如果输入此命令，则无论存档程序处于何种状态，均会产生此效用。因此，应谨慎使用 <code>arrestart</code> 命令。输入此命令时，某些对存档介质的复制操作可能会中断，您必须重新进行复制操作。这会浪费介质的空间。
<code>arrun</code>	使存档程序开始存档操作。该命令将改写 <code>archiver.cmd</code> 文件中现有的全局 <code>wait</code> 命令。
<code>arstop</code>	立即停止所有存档操作。

存档程序命令的格式如下：

```
:aridle [ dk | rm | fs.fsname ]
:arrestart
:arrun [ dk | rm | fs.fsname ]
:arstop [ dk | rm | fs.fsname ]
```

这些命令的变量是可选的。如果不指定变量，则命令应用于所有文件系统。如果指定了变量，则命令将根据指定的存档文件类型（`dk` 或 `rm`）和指定的文件系统来采取不同的操作。表 9-20 列出了存档程序命令的变量。

表 9-20 存档程序命令的变量

变量	说明
<code>dk</code>	指明该命令适用于磁盘存档文件。
<code>rm</code>	指明该命令适用于可移动介质文件。
<code>fs.fsname</code>	指明该命令适用于特定文件系统。输入用于替换 <code>fsname</code> 的文件系统名。

设备命令

表 9-21 列出了设备命令及其作用。

表 9-21 设备命令作用

命令	作用
<code>devlog</code>	设置设备日志选项。
<code>down</code>	停止设备 <code>eq</code> 上的操作。
<code>idle</code>	通过阻止建立设备 <code>eq</code> 的新连接来限制访问设备。现有操作会继续进行，直到完成。
<code>off</code>	逻辑关闭设备 <code>eq</code> 。
<code>on</code>	逻辑打开设备 <code>eq</code> 。
<code>unavail</code>	选择设备 <code>eq</code> 并使其不可用于 Sun SAM-FS 或 Sun SAM-QFS 文件系统。
<code>unload</code>	从指定的可移动介质设备 <code>eq</code> 卸载已安装的介质。对于存储匣设备， <code>unload</code> 命令可以卸载已安装的卡盒并弹出存储匣。

设备控制命令的格式如下：

```
:devlog eq [ option ... ]
:down eq
:idle eq
:off eq
:on eq
:unavail eq
:unload eq
```

表 9-22 列出了设备命令的变量。

表 9-22 设备命令的变量

变量	说明
<i>eq</i>	设备在 mcf 文件中的设备序号。
<i>option</i>	零个或多个事件类型。可能的事件类型包括：all、date、default、detail、err、event、label、mig、module、msg、none、retry、stage、syserr 和 time。有关这些选项的信息，请参阅 defaults.conf(4) 联机资料。

显示控制命令

表 9-23 列出了显示控制命令及其作用。

表 9-23 显示控制命令的作用

命令	作用
:a [<i>filesystem</i>]	显示存档程序的状态。
:n [<i>media</i>]	为可移动介质 I/O 活动显示屏幕选择介质类型。
:p [<i>media</i>]	为安装请求显示屏幕选择介质类型。
:q	退出 samu 操作员实用程序。
:r [<i>media</i>]	为可移动介质状态显示屏幕选择设备类型。
:refresh <i>i</i>	设置刷新显示窗口的时间间隔并启用显示刷新功能。按 CTRL-r 组合键可以交替打开和关闭显示刷新功能。
:u [<i>media</i>]	显示登台队列。该命令适用于当前已安装的卷。
:v [<i>eq</i>]	为显示屏幕选择库 VSN 目录。若要查看 Historian（历史记录）目录中的 VSN，请用 historian 替换关键字 <i>eq</i> 。
:w [<i>media</i>]	显示预置队列。此命令适用于尚未安装的卷。

显示控制命令的格式如下：

```
:a [ filesystem ]
:n [ media ]
:p [ media ]
:q
:r [ media ]
:refresh i
:u [ media ]
:v [ eq ]
:w [ media ]
```

这些命令变量两侧的括号表示这些变量在许多情况下是可选项。其中一些命令的变量可以将 `samu(1M)` 显示输出的范围限制在特定文件系统、介质类型或设备序数上。如果未指定参数，则命令会显示当前所有选定的或配置的文件系统、介质类型和设备序数的信息。

表 9-24 列出了显示控制命令的变量。

表 9-24 显示控制命令的变量

变量	说明
<i>filesystem</i>	指定 Sun SAM-FS 或 Sun SAM-QFS 文件系统的名称。 如果指定 <i>filesystem</i> 变量，存档程序状态显示屏幕会列出标准文件的数量；脱机文件的数量；已存档文件的数量；存档副本和目录的数量；文件系统；安装点；索引节点活动；以及时间间隔。 如果未指定 <i>filesystem</i> 变量，则存档程序状态显示屏幕会列出文件系统和安装点的名称；索引节点扫描活动；并列出存档程序下一次扫描文件系统的时间。
<i>media</i>	指定介质类型。有关所支持的介质类型列表，请参阅 <code>mcf(4)</code> 联机资料。另外，用户也可以指定关键字 <code>all</code> ，以表示所有介质类型或可移动介质设备。
<i>eq</i>	设备在 <code>mcf</code> 文件中的设备序数。
<i>i</i>	时间间隔，以秒为单位。

文件系统命令

`:meta_timeo eq interval` 命令

`metatimeo` 命令用于设置 Sun QFS 共享文件系统元数据在高速缓存中的超时值。

对于 *eq*，指定文件系统的设备序数。

对于 *interval*，指定时间间隔（秒）。默认 *interval* 是 15 秒。时间间隔到期之后，客户机主机系统将从元数据服务器主机获取新的元数据副本。

`:notrace eq` 命令

`notrace` 命令用于禁用跟踪功能。

对于 *eq*，指定文件系统的设备序数。

`:partial eq size` 命令

`partial` 命令用于设置文件在释放之后保持联机的千字节数。

对于 *eq*，指定文件系统的设备序数。

对于 *size*，指定保持联机的千字节数。默认 *size* 是 16。

`:readahead eq contig` 命令

`readahead` 命令用于指定文件系统可提前读取的最大字节数。

对于 *eq*，指定文件系统的设备序数。

对于 *contig*，指定 1 千字节块的数量。*contig* 必须是介于 1 和 8192 之间的整数。指定的 *contig* 将被圆整为 8 千字节的倍数。默认 *contig* 是 8（131072 字节）。

例如，下面的命令将设备序数为 3 的文件系统的最大连续块大小设置为 262,144 字节。

```
:readahead 3 256
```

此外，用户还可以通过在 `samfs.cmd` 文件中指定 `readhead` 指令来配置该数值。有关详细信息，请参阅 `samfs.cmd(4)` 联机资料。

`:thresh eq high low` 命令

`thresh` 命令用于设置文件系统的阈值上限和下限，以便控制文件存档活动。

对于 `eq`，指定存储系列集的设备序数。

对于 `high`，指定阈值上限。

对于 `low`，指定阈值下限。

例如，下面的命令将文件系统设备序数为 10 的存储系列集的阈值上限设置为 50%，阈值下限设置为 40%。

```
:thresh 10 50 40
```

`:trace eq` 命令

`trace` 命令用于启用文件系统跟踪功能。

对于 `eq`，指定文件系统的设备序数。

`:writebehind eq contig` 命令

`writebehind` 命令用于指定文件系统可以后写的最大字节数。

对于 `eq`，指定文件系统的设备序数。

对于 `contig`，指定 1 千字节块的数量。`contig` 必须是介于 1 和 8192 之间的整数。默认 `contig` 是 8 (131072 字节)。

例如，下面的命令将设备序数为 50 的文件系统的最大连续块大小设置为 262,144 字节。

```
:writebehind 50 256
```

此外，用户还可以通过在 `samfs.cmd` 文件中指定 `writebehind` 指令来配置该数值。有关详细信息，请参阅 `samfs.cmd(4)` 联机资料。

传输器命令

`:audit [-e] eq [:slot [:side]]` 命令

`audit` 命令可使指定的传输设备安装每一个卷、读取 VSN 并重建库目录。

对于 `eq`，指定传输设备的设备序数。

`:export eq:slot` 和 `:export mt.vsn` 命令

`export` 命令可使指定的传输设备将卷导入邮槽。卷按其其在传输器中的端口位置进行标识。

- 如果按设备序数和端口编号导出，则指定的传输设备会将卷载入驱动器。对于 `eq`，指定设备序数或设备名称。对于 `slot`，指定包含您要载入的卷的端口编号。
- 如果按逻辑标识导出，则指定的传输设备会将已标记的卷装入驱动器。对于 `mt`，指定介质类型；有关有效介质类型的信息，请参阅 `mcf(4)` 联机资料。对于 `vsni`，指定要安装的卷。

`:import eq` 命令

`import` 命令可以使您使用指定的传输设备添加卡盒。对于 `eq`，指定传输设备的设备序数。

`:load eq:slot [:side]` 和 `:load mt.vsn` 命令

`load` 命令可以使您按物理或逻辑标识执行载入操作，如下所述：

- 如果按设备序数和端口编号执行载入操作，则指定的传输设备会将卷载入驱动器。对于 `eq`，指定设备序数或设备名称。对于 `slot`，指定包含您要载入的卷的端口编号。
- 如果按逻辑标识执行载入操作，则指定的传输设备会将已标记的卷装入驱动器。对于 `mt`，指定介质类型；有关有效介质类型的信息，请参阅 `mcf(4)` 联机资料。对于 `vsni`，指定要安装的卷。

其他命令

:clear *vsn* [*index*] 命令

clear 命令用于从可移动介质安装请求显示屏幕中清除指定的 VSN（参阅第 231 页的“(p) — 可移动介质载入请求显示屏幕”）。输入此命令时，系统将中止任何等待 VSN 安装的进程。如果指定 *index*，则 *index* 是 VSN 在可移动介质显示屏幕中的十进制序数。

:dtrace 命令

dtrace 命令的格式如下：

- :dtrace *daemon_name* on
- :dtrace *daemon_name* off
- :dtrace *daemon_name.variable value*

dtrace 命令用于指定各种跟踪选项。表 9-25 列出了跟踪控制命令的变量。

表 9-25 跟踪命令的变量

变量	说明
<i>daemon_name</i>	指定关键字 all 或进程名称。如果指定关键字 all，则跟踪命令会影响所有后台程序。如果指定以下进程名称之一，则跟踪命令只影响指定的进程： sam-archiverd、sam-catserverd、sam-fsd、sam-ftpd、 sam-recycler、sam-sharefsd 和 sam-stagerd。用户可在进程名称之后指定关键字 on 或 off。如果指定 on 或 off，则会为所有指定的进程启用或禁用跟踪功能。
<i>variable value</i>	可以指定许多不同的 <i>variable</i> 和 <i>value</i> 变量。defaults.conf(4) 联机资料包含有这些变量的详细信息。指定下列一种 <i>variable</i> 和 <i>value</i> 组合： <ul style="list-style-type: none">• <i>file value</i>。对于 <i>value</i>，指定跟踪文件可以写入的文件名。它可以是一个完整的路径名。• <i>options value</i>。对于 <i>value</i>，指定一系列由空格隔开的跟踪选项。• <i>age value</i>。对于 <i>age</i>，指定跟踪文件的循环更新时间。• <i>size value</i>。对于 <i>value</i>，指定跟踪文件开始循环更新时的大小。

:mount *mntpt* 命令

mount 命令用于选择 Sun QFS、Sun SAM-FS 或 Sun SAM-QFS 文件系统。

:open *eq* 命令

`open` 命令可以使您访问指定的磁盘设备。使用 `read` 命令、磁盘扇区显示选项 (S) 或文件标签显示选项 (F) 之前，您必须运行该命令。*eq* 表示设备序数。

:read *addr* 命令

`read` 命令用于从当前打开的磁盘设备中读取指定的扇区。读取之前您必须打开设备。对于 *addr*，指定十六进制扇区地址。

:snap [*filename*] 命令

`snap` 命令用于将显示窗口的快照发送至 *filename*，后者是接受显示信息的文件名称。

为便于报告问题，您可以抓取所有 `samu(1M)` 实用程序显示屏幕的快照。每一个新的快照均会添加到快照文件的末尾。默认文件是当前工作目录中的 `snapshots`。该文件可以打印，通过 `vi(1)` 进行检查，或传真给 Sun Microsystems 客户支持人员。

:!*shell_command* 命令

!`shell_command` 命令可以使您在不退出 `samu(1M)` 操作员实用程序的情况下运行 `shell` 命令。

升级环境

本章介绍如何升级现有 Sun SAM-FS 或 Sun SAM-QFS 环境中的硬件。它包括下列主题：

- 第 256 页的“在自动化库中添加端口”
- 第 257 页的“升级或更换库”
- 第 260 页的“升级 DLT 磁带驱动器”

另外，用户还可能需要在 Sun SAM-FS 或 Sun SAM-QFS 环境中执行其它类型的操作和升级过程。以下出版物介绍了这些类型的升级过程：

- 《*Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 文件系统管理员指南*》介绍了以下类型的操作和升级过程：
 - 如何初始化文件系统
 - 如何初始化或重新初始化 mcf 或 defaults.conf 文件
 - 如何安装文件系统
 - 如何卸载文件系统
 - 如何检验文件系统完整性
 - 如何修复文件系统
 - 如何保存升级信息
 - 如何准备升级硬件
 - 如何添加磁盘高速缓存到文件系统
 - 如何在文件系统中更换磁盘
 - 如何升级主机系统
 - 如何在 Sun QFS、Sun SAM-FS 或 Sun SAM-QFS 环境中升级 Sun Solaris 操作环境 (OE)
 - 如何在 Sun QFS 环境中升级 Sun Solaris 操作环境 (OE)
- 《*Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 安装与配置指南*》介绍了如何升级 Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 软件。

在自动化库中添加端口

Sun SAM-FS 或 Sun SAM-QFS 系统可以管理的卡盒端口数量由许可证密钥控制。若要增加端口的数量，请执行本节中所述的步骤。

▼ 在库中添加端口

1. 确定您是否需要通过授权的服务供应商 (ASP) 获取一组新的许可证密钥，如果您的帐户未分配 ASP，请通过 Sun Microsystems 获取许可证密钥。（可选）

2. 使用新的许可证密钥取代现有的许可证密钥。（可选）

如果您已获得新的许可证密钥，请执行本步骤。

许可证密钥从下面文件的第一列开始：

```
/etc/opt/SUNWsamfs/LICENSE.4.0
```

其中不可包含其它任何关键字、主机 ID 或其它信息。

3. 运行 `samd config` 命令以使 Sun SAM-FS 或 Sun SAM-QFS 软件识别新的许可证密钥。（可选）

如果您已获得新的许可证密钥，请执行本步骤。

例如：

```
# samd config
```

4. 卸载库目录。

您可以使用 `samu(1M)` 实用程序的 `:unload` 命令，如下所示：

```
:unload eq
```

`eq` 变量表示自动化库在 `mcf` 文件中定义的设备序数。此命令可将库目录条目移入 `Historian`（历史记录）目录，并保存每一个卡盒的目录信息。

在 `samu(1M)` 中输入 `:unload` 命令之后，自动化库的 `v` 显示屏幕会清空，而 `Historian` 的 `v` 显示屏幕则充满了曾位于自动化库中的 `VSN`。

5. 停止运行 Sun SAM-FS 或 Sun SAM-QFS 系统。

有关如何执行本步骤的信息，请参阅第 11 页的“基本操作”。

6. 根据制造商建议的过程关闭主机系统和库的电源。

7. 请库硬件工程师将端口添加至自动化库。

8. 使用标准启动过程打开系统的电源。

9. 启动 Sun SAM-FS 和 Sun SAM-QFS 系统。

有关如何执行本步骤的信息，请参阅第 11 页的“基本操作”。新的许可证信息将出现在 samu(1M) 实用程序的 1 显示屏幕中。

升级或更换库

断开连接并安装另一自动化库之前，请根据《Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 文件系统管理员指南》中的“准备升级硬件”部分所述，为升级做好准备。

▼ 更换或升级库

1. 使用 samu(1M) 实用程序的 `:unload` 命令卸载库目录。

例如：

```
:unload eq
```

`eq` 变量表示自动化库在 `mcf` 文件中定义的设备序数。此命令可将库目录条目移入 Historian（历史记录）目录，并保存每一个卡盒的目录信息。

输入 `:unload` 命令之后，自动化库的 `v` 显示屏幕会清空，而 Historian 的 `v` 显示屏幕则充满了曾位于自动化库中的 VSN。

2. 更新 `/etc/opt/SUNWsamfs/inquiry.conf` 文件。（可选）

新库应在此文件中按供应商、自动化库型号和 Sun SAM-FS 或 Sun SAM-QFS 内部名标识出来。

例如，释放的 `inquiry.conf` 文件中包含下面的行：

```
"HP", "C1710T", "hpoplib" # HP optical library
```

该行表示如果系统检测到由 HP 生产的型号为 C1710T 的 SCSI 设备，则系统将它作为 hpoplib 进行驱动。前两个字段（供应商/产品）由硬件设备得出。最后一个字段 hpoplib 是系统内部使用的名称，系统使用该名称来确定如何与该设备进行通信。如果 inquiry.conf 文件需要更改，则所做的更改在重新启动 sam-initd 后台程序后才能生效。

3. 将当前 /etc/vfstab 文件另存为 /etc/vfstab.cur。
4. 编辑 /etc/vfstab 文件。
将 Sun SAM-FS 或 Sun SAM-QFS 安装选项从 yes 更改为 no。
5. 将 /etc/opt/SUNWsamfs/archiver.cmd 文件另存为 archiver.cmd.cur。
6. 编辑 /etc/opt/SUNWsamfs/archiver.cmd 文件。
将 wait 指令添加在第一行。
7. 根据制造商建议的过程关闭主机系统和外围设备的电源。
8. 断开自动化库的连接。
9. 将连接电缆连接至新的自动化库。
10. 根据制造商建议的开机顺序打开外围设备和主机系统的电源。
11. 确保主机系统可以识别新的自动化库。

输入下面的命令：

```
> probe-scsi-all
```

继续下一步骤之前，该命令必须显示了新的自动化库及其驱动器。如果系统不能识别自动化库及其驱动器，则它们可能存在连接问题。

12. 引导系统。

输入下面的命令以在新配置下引导系统：

```
> boot -rv
```

13. 如果驱动器或自动化库的目标号发生了变化，或者驱动器在自动化库中的顺序或编号发生了变化，请修改 /etc/opt/SUNWsamfs/mcf 文件以反映新的配置信息。（可选）

这类似于《Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 安装与配置指南》中所述的初始安装过程。

14. 创建新的 /dev/samst 条目。（可选）

如果您添加了新设备，请执行本步骤。输入下面的命令：

```
# samdev
```

15. 初始化 Sun SAM-FS 和 Sun SAM-QFS 系统。

您可以通过安装文件系统或运行下面的命令来执行本步骤。

```
# samd start
```

系统在初始化期间会检测到自动化库中的端口数量发生了变化。系统将对自动化库运行全面的核查以更新库目录。恢复存档之前，必须完成全面核查过程。

如果在核查期间出现问题，则很可能是因为驱动器在自动化库中的顺序与其在 /etc/opt/SUNWsamfs/mcf 文件中的顺序不符。请记住，驱动器有两个属性：SCSI 目标 ID 以及它在自动化库中的位置。无论是在升级之前还是在升级之后，这两个属性均必须正确无误。

如果核查顺序完成，请继续下一步骤。

16. 使用升级之前的版本替换 /etc/vfstab 和 /etc/opt/SUNWsamfs/archiver.cmd 文件。

分别使用保存的 /etc/vfstab.cur 和 /etc/opt/SUNWsamfs/archiver.cur 文件。

17. 重新引导系统，确保配置不存在任何错误。

自动化库按位置编号调用驱动器。例如，当系统希望将卡盒载入驱动器时，它必须向自动化库发出命令以将卡盒从端口 123 载入至驱动器 3。

根据第三个 mcf 条目，驱动器 3 可能是 SCSI 目标 6。系统之所以知道它是驱动器 3 是因为它是 mcf 文件中的第三个驱动器条目。自动化库之所以知道它是驱动器 3 是因为它在自动化库中占据的物理位置。

请求自动化库将卡盒载入驱动器之后，系统将测试驱动器是否处于设备就绪状态。此时，系统将使用 mcf 文件中 /dev/samst/scsi-target 条目定义的 SCSI 目标 ID。因此，这些条目与刚才载入卡盒的驱动器相符是非常重要的。

目前，没有好的方法来确定此类信息。通常，制造商在自动化库出厂时为驱动器设置了升序的 SCSI ID，但这一点并没有保证。确定此类信息的一种方法是运行 samu(1M) 实用程序的 :load 命令以载入卡盒，然后观察 samu(1M) 实用程序的 s 显示屏幕，从而确定哪一个驱动器在 t 显示屏幕的状态标记中显示 r 而不是 p。

升级 DLT 磁带驱动器

为了利用拥有更高密度和更快速度的磁带技术，通常需要升级自动化库中的 DLT 磁带驱动器或独立磁带驱动器。例如，您可以将 DLT 4000 驱动器升级至 DLT 7000 驱动器。

在 Sun SAM-FS 或 Sun SAM-QFS 环境中，您需要在启动 Sun SAM-FS 环境之前进行一系列操作，添加新驱动器、重新引导新配置以及更新 mcf 文件（如有必要）。此外，如果您要升级端口的数量，则还需联系授权的服务供应商 (ASP) 或 Sun Microsystems，因为您可能需要升级许可证。

升级驱动器之前，请注意以下限制和一般信息：

- Sun SAM-FS 和 Sun SAM-QFS 环境不支持在同一个直接连接自动化库中混用不同的 DLT 磁带驱动器。例如，Sun SAM-FS 系统不能区分同一个自动化库中的 DLT 4000 磁带驱动器和 DLT 7000 磁带驱动器。因此，您必须同时使用新的驱动器更换所有的旧 DLT 驱动器。
- 低密度磁带不能与高密度磁带或磁带驱动器混合使用。不过，您可以继续使用高密度驱动器来读取低性能的磁带或向其中写入数据。
- 为了充分利用高密度 DLT 磁带，您可能希望回收现有的文件并将它们移植到高密度磁带中。若要执行此操作，请将所有低密度磁带标记为只读，然后标记要回收的磁带。有关回收磁带的信息，请参阅第 165 页的“回收”。
- 当标记每一个磁带时，系统将确定磁带的密度并将其记录至库目录中。

▼ 升级磁带驱动器

1. 确定您的当前转储文件是否完整无缺。（可选）

如果不是，请在继续下一步骤之前对文件系统执行 `samfsdump(1M)`。

2. 更新 `/kernel/drv/st.conf` 文件以标识新驱动器。

磁带驱动器在此文件中按供应商、磁带型号和 Sun SAM-FS 或 Sun SAM-QFS 内部名进行标识。例如，释放的 `st.conf` 文件中包含下面的行：

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

`/opt/SUNWsamfs/examples/st.conf_changes` 中提供了文件示例。您可以将整个文件读入 `/kernel/drv/st.conf`，也可合并所需的更改。有关更新 `st.conf` 文件的详细信息，请参阅《*Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 安装与配置指南*》。

3. 根据制造商建议的过程关闭主机系统和外围设备的电源。

4. 使用新的磁带驱动器更换旧的磁带驱动器。
5. 根据制造商建议的开机顺序打开外围设备和主机系统的电源。
6. 确保主机系统可以识别新的驱动器。

输入下面的命令：

```
> probe-scsi-all
```

继续下一步骤之前，该命令必须显示了自动化库及其新的驱动器。如果未显示这些驱动器，可能存在连接问题，请予以纠正。该命令返回预期的信息之后，您可以继续下一步骤。

7. 引导系统。

输入下面的命令以在新配置下引导系统：

```
> boot -rv
```

8. 修改 `/etc/opt/SUNwsamfs/mcf` 文件以反映新的配置信息。（可选）

如果驱动器或自动化库的目标号发生了变化，或者驱动器在自动化库中的顺序或编号发生了变化，请执行本步骤。这类似于《*Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 安装与配置指南*》中所述的初始安装过程。

9. 为新设备创建新的 `/dev/samst` 条目。（可选）

若要创建这些条目，请输入下面的命令：

```
# samdev
```

10. 启动 Sun SAM-FS 和 Sun SAM-QFS 系统。

11. 安装文件系统。

现在，即可继续使用现有的 Sun SAM-FS 或 Sun SAM-QFS 磁带。

高级内容

本章介绍超出基本系统管理和使用范围的高级内容。

它包括以下内容：

- 设备日志
- 可移动介质文件
- 卷溢出
- 分段文件
- 系统错误工具 (SEF) 报告

设备日志

设备日志工具可以提供设备专用的错误信息，您可以使用这些信息来分析某些类型的设备问题。它可帮助您确定自动化库、磁带驱动器或光盘驱动器发生故障事件的顺序。请注意，设备日志工具并不能收集软介质错误，如可恢复的读取错误。

系统将设备日志消息分别写入至各个不同的日志文件。每一个自动化库、磁带和光盘驱动器设备均有一个日志文件，并且 **Historian**（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。每一个日志文件的名称分别与相应设备的 **Equipment Ordinal**（设备序数）相同。

示例：假设您的环境中具有 Sun SAM-FS 文件系统和一个配有两块光盘驱动器的 Hewlett Packard 光盘库。

代码示例 11-1 显示了 mcf 文件。

代码示例 11-1 mcf 文件示例

```
/dev/samst/c1t5u0 40 hp hp40 - etc/opt/SUNWsamfs/hp40_cat
/dev/samst/c1t4u0 41 mo hp40 -
/dev/samst/c1t6u0 42 mo hp40 -
```

/var/opt/SUNWsamfs/devlog 文件如下所示：

```
# pwd
/var/opt/SUNWsamfs/devlog
# ls
40      41      42      43
#
```

设备 43 是 Historian（历史记录）。

何时使用设备日志

设备日志可以方便地生成许多日志消息，特别适用于已打开所有设备的全部日志选项且具有大量设备活动的场合。设备日志设置的最初设置为下面的默认值：

```
err, retry, syserr
```

如果您怀疑 Sun SAM-FS 或 Sun SAM-QFS 环境中配置的某个设备出现问题，则应为该设备启用额外的日志事件。另外，如果您的服务供应商建议您这样做，也应启用设备日志。在这些情况下，请将事件设置为 detail（详细）。在特别情况下，您的服务供应商可能建议您将某个设备的事件设置为 all（全部）。这可以添加额外的日志信息。不过，一般而言，在系统运行时设置过多的日志并无益处，甚至是不可行的。

当运行 info.sh(1M) 命令时，系统会自动收集设备日志信息。这样，作为问题分析活动的一部分，文件系统服务人员可以复查任何可能的设备错误信息。

启用设备日志

您可以使用两种方法来启用设备日志。

对于第 1 种和第 2 种方法：

- *eq* 是设备在 *mcf* 文件中定义的设备序数，或对于所有设备，它为关键字 *all*。
- `samset(1M)` 联机资料中列出了设备日志事件。另外，第 265 页的“启用设备日志”也列出了这些设备日志事件。请注意，设备日志消息仅以英文文本的格式提供。*event* 是下表中的一种或多种事件类型：
 - *all*
 - *date*
 - *default*
 - *detail*
 - *err*
 - *event*
 - *label*
 - *mig*
 - *module*
 - *msg*
 - *none*
 - *retry*
 - *stage*
 - *stage_ck*
 - *syserr*
 - *time*

第 1 种方法

第 1 种方法按以下格式使用 `samset(1M)` 命令：

```
samset devlog eq event
```

有关 `samset(1M)` 命令的详细信息，请参阅 `samset(1M)` 联机资料。

第 2 种方法

第 2 种方法要求在 `/etc/opt/SUNwsamfs/defaults.conf` 文件中输入指令。编辑 `defaults.conf` 文件并添加下面的指令：

```
devlog eq event
```

当 Sun SAM-FS 或 Sun SAM-QFS 文件系统启动时，它会自动将每一个可用设备的事件类型设置为 `default`。另外，您还可以使用 `samset(1M)` 命令来确定每一个设备日志的当前设置。

可移动介质文件

您可以使用 `request(1)` 命令手动创建、写入和读取那些不使用磁盘高速缓存来缓冲数据的文件。采用这种方式创建的文件称为 *可移动介质文件*。

与典型的 Sun SAM-FS 或 Sun SAM-QFS 文件类似，可移动介质文件也具有权限、用户名、群组名和大小属性。但是，这些文件的数据并不保存在磁盘高速缓存中。因此，大于磁盘高速缓存的文件可以创建并写入至介质中。对于 `request(1)` 命令中指定的文件，系统将在 `.inodes` 文件中为其创建索引节点条目。用户并不需要知道文件在可移动介质中的起始处。（这与磁盘高速缓存中的数据相同。）Sun SAM-FS 和 Sun SAM-QFS 文件系统从索引节点条目读取有关信息。多个可移动介质文件可以保存在同一个卡盒中。

可移动介质文件必须按顺序读取和写入。用户必须为介质指定介质类型和至少一个 VSN。用户可以指定多个卷（最多 256 个）来处理卷溢出功能（参阅第 267 页的“卷溢出”）。如果所请求的卷位于 `mcf` 文件中定义的自动化库中，则 Sun SAM-FS 或 Sun SAM-QFS 文件系统可以自动安装该卷。

用于 `request(1)` 命令的卷不应与那些在 Sun SAM-FS 或 Sun SAM-QFS 环境中用于自动存档的卷相同。每次存档时，系统均会将下一个要存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。

如果卷中存在可移动介质文件，则回收程序不能对该卷进行回收。回收程序希望只有已存档的文件位于指定用于存档的特定卷中。另外，存档程序永远不会存档可移动介质文件。

NFS 不支持可移动介质文件。

如果在 `request(1)` 命令或 `sam_request(3)` 库例程中指定 `-N` 选项，则所读取的文件可以是 Sun SAM-FS 或 Sun SAM-QFS 环境无法识别的文件。在读取未标记、编有条码且已启用写保护的磁带时，用户可以使用此选项。用户可为卷溢出文件最多指定 256 个卷。

有关说明如何创建可移动介质文件的示例，请参阅 `request(1)` 联机资料。

卷溢出

卷溢出功能允许系统将单个文件写入至多个卷中。对于超过选定介质容量的大文件来说，这是一个非常有用的功能。请注意，如果使用卷溢出功能，则在恢复出现故障的文件时，恢复卷溢出数据将会很困难。有关详细信息，请参阅 request(1) 联机资料。

在 archiver.cmd 文件中使用 ovflmin 指令即可启用卷溢出功能。当文件的大小超过 ovflmin 的值时，存档程序会将文件的另一部分写入至另一个同类型的可用卷中（如有必要）。写入至每一个卷的文件部分称为片段。有关为卷溢出功能设置 ovflmin 指令的说明，请参阅第 73 页的“存档”中有关控制卷溢出功能的信息。

用户可直接使用 request(1) 命令创建卷溢出可移动介质文件。请注意，使用 request(1) 命令将会忽略存档程序的一般功能。将文件写入至各个不同的卷时，必须用斜杠将 VSN 隔开。使用 request(1) 命令的 -l 选项，可在文件中列出 VSN。有关完整的语法结构，请参阅 request(1) 联机资料。

示例 1: 下面的 request(1) 命令示例使用 Ampex D2 磁带中的三个卷创建可移动介质文件：

```
# request -m d2 -v TAPE01/TAPE02/TAPE03 large.file
```

示例 2: sls(1) 命令列出的存档副本可以指明文件在每一个 VSN 上的各个片段。本示例显示了存档程序日志文件以及对写入至多个卷上的大文件 file50 运行 sls -D 命令时的输出。

此处的存档程序日志文件表明文件 file50 写入至三个卷中，它们的 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，它们的值与另外显示的 sls -D 输出中的值相匹配。有关存档日志条目的详细说明，请参阅 archiver(1M) 联机资料。

file50 的存档日志文件条目如下所示：

```
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
```

sls -D 的输出如下所示:

```
# sls -D file50
file50:
mode: -rw-rw----  links: 1  owner: gmm  group: sam
length: 1500000172  admin id: 7  inode: 1407.5
offline;  archdone;  stage -n
copy1: ---- Jan 13 15:55 1e4b1.1 1t DLT001
  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
```

用户可为卷溢出文件最多指定 256 个卷。

请注意，卷溢出文件不能生成校验和。有关使用校验和的详细信息，请参阅 `ssum(1)` 联机资料。

如果您使用卷溢出功能，并且您要恢复的文件位于多个卷上，请参阅《*Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 故障恢复指南*》中的示例来了解如何恢复此类文件。

分段文件

Sun SAM-FS 和 Sun 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。使用分段文件，您可以只将文件的一部分时刻保留在磁盘高速缓存中。

`segment(1)` 命令可以使您指定分段大小。您所设置的分段大小不能大于当前文件的大小。

分段文件支持磁带拆分功能。将文件分段后，可将文件同时拆分到多个磁带设备中，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存档，因此分段还可以提高存档效率。文件的各个分段可以并行存档，并且分段文件可以并行登台。这提高了系统的存档和恢复性能。

用户可以为文件、目录或整个文件系统启用分段功能。分段文件支持所有其它 Sun SAM-FS 和 Sun SAM-QFS 功能。

以下几节说明了分段文件与非分段文件之间的差异。有关分段文件的详细信息，请参阅 `segment(1)` 或 `sam_segment(3)` 联机资料。

存档

对于分段文件，存档单位是文件段自身，而不是整个文件。所有存档属性和优先级均应用于单个文件段，而不应用于整个文件。

所存档的单位是文件段。通过在 `archiver.cmd` 文件中为存档组设置 `-drives` 参数，可以拆分文件段。

例如，假定文件系统中有一个大小为 100MB 的分段文件，其段大小为 10MB。如果 `archiver.cmd` 中使用 `-drives 2` 指令定义存档组，则该分段文件将并行存档至 2 个驱动器。段 1、3、5、7 和 9 存档在第一个驱动器中，而段 2、4、6、8 和 10 存档在第二个驱动器中。

存档程序只存档已发生更改的文件段 — 而不是整个文件。每一个文件段最多可以创建四个存档副本。文件段支持卷溢出功能。

注 – 分段文件的索引不含用户数据。索引被视为元数据，由系统分配至文件系统存档组。

故障恢复

有关在出现故障时恢复分段文件的详细信息，请参阅《*Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 故障恢复指南*》。

系统错误工具报告

系统错误工具 (SEF) 报告系统用于收集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至日志文件并转换成可读的格式。它包括以下项目：

- 日志文件，包含从磁带设备日志检测页收集的数据。
- `sefreport(1M)` 命令，以可读的格式将日志文件写入至 `stdout`。该日志文件可以作为用户的分析脚本的输入项。

`sefreport(1M)` 命令用于读取 Sun SAM-FS 或 Sun SAM-QFS SEF 日志文件的内容。日志文件中包含从 Sun SAM-FS 和 Sun SAM-QFS 环境中所用外围磁带设备的日志检测页收集的数据。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，请参阅每一个特定设备的供应商文档。

独立磁带驱动器不支持 SEF 报告功能。

▼ 启用 SEF 报告

1. 以超级用户 (root) 的身份登录系统。

您必须是超级用户才能完成本过程。

2. 使用 `mkdir(1)` 命令创建 SEF 目录。

例如：

```
# mkdir /var/opt/SUNWsamfs/sef
```

3. 使用 `touch(1)` 命令启用 SEF 报告功能。

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。您可以使用下面的命令来创建日志文件：

```
# touch /var/opt/SUNWsamfs/sef/sefdata
```

上面的命令示例表明所创建的 SEF 日志文件位于 `/var/opt/SUNWsamfs/sef/sefdata` 目录中。这是默认位置。

生成的 SEF 数据将添加至日志文件的末尾。

您可以对 SEF 报告功能进行配置，以便在其它位置记录和读取日志检测数据。有关从其它位置读取日志检测文件的详细信息，请参阅 `sefreport(1M)` 联机资料。

SEF 报告输出

使用 `sefreport(1M)` 命令之前，请确保 `/opt/SUNWsamfs/sbin` 位于您的命令路径中。SEF 报告输出的内容由标题行和日志检测数据组成。

记录中每一页的日志检测数据将打印在标题行的后面。对于每一日志检测数据页，系统均会打印用于标识页码的行，随后是一行列标题。接下来打印数据，每行三列，各列的标题分别为：`param code`、`control` 和 `param value`。所有生成的数据均采用十六进制格式。

例如，下面的 SEF 命令从默认位置读取 SEF 日志文件，写入每一个设备的设备编号和路径名，并生成输出：

```
# sefreport -d /var/opt/SUNWsamfs/sef/sefdata > sef.output
```

代码示例 11-2 显示了 sef.output 文件的内容。

代码示例 11-2 sef.output 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
```

代码示例 11-2 sef.output 内容 (续)

```
PAGE CODE 3
param code control param value
    00h      74h      0x0
    01h      74h      0x0
    02h      74h      0x0

    03h      74h      0x0
    04h      74h      0x0
    05h      74h      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
```


代码示例 11-2 sef.output 内容 (续)

```
00h      74h      0x0  
.   
.   
. 
```

注 – 受本手册的内容所限，上面的输出已作了删节。

有关 SEF 日志文件的详细信息（包括内容和格式），请参阅 `sefdata(4)` 联机资料。
有关可选 SEF 报告格式的详细信息，请参阅 `sefreport(1M)` 联机资料。

sefreport(1M) 命令

下面是 `sefreport(1M)` 命令的两个最常用选项：

- `-d` 选项。`-d` 选项用于生成附加的设备信息。它将为每一条记录写入附加的包含设备序数和路径名的标题行。这便于用户搜索和查找与特定设备相关的 SEF 记录。
- `-v` 选项。`-v` 选项用于生成详细的信息。它将与设备序数、页码和 VSN 的信息添加至记录的每一行。这使用户可以只选择与特定设备或特定卷相关的行。

管理 SEF 日志文件

SEF 日志文件的管理方式与其它任何 Sun SAM-FS 或 Sun SAM-QFS 日志文件相同。用户可以定期运行 `cron(1)` 作业以将当前日志文件保存至另一位置、删除旧 SEF 文件、创建新（空）的 SEF 文件或执行其它任务。

另外，用户还可使用 `log_rotate.sh(1M)` 实用程序来循环更新该日志文件。

有关用于管理 SEF 日志文件的工具的详细信息，请参阅 `cron(1)` 或 `log_rotate.sh(1M)` 联机资料。

词汇表

字母

- DAU**
(disk allocation unit) 联机存储设备的基本单位。也称“块大小”。
- Sun SAM-FS 和 Sun SAM-QFS 文件系统既支持小 DAU，也支持大 DAU。小 DAU 是指 4 KB (2^{14} 或 4096 字节)。大 DAU 是指 16、32 或 64 KB。有效的 DAU 大小配对包括 4/16、4/32 和 4/64。
- 此外，Sun QFS 和 Sun SAM-QFS 文件系统还支持大小完全可调的 DAU，范围从 16 KB 到 65,528 KB 不等。用户指定的 DAU 必须是 8 KB 的倍数。
- FDDI** 光纤分布式数据接口 (Fiber distributed data interface) 的缩写。一种运行速度为 100 MB/s 的光纤局域网。
- FTP** 文件传输协议 (File Transfer Protocol) 的缩写。一种通过 TCP/IP 网络在两个主机之间传送文件的网际协议。
- LAN** 局域网 (Local area network) 的缩写。
- LUN** 逻辑单元编号 (Logical unit number) 的缩写。
- mcf** 主配置文件 (master configuration file) 的缩写。系统在初始化期间读取的文件，它定义了 Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 环境中各个设备之间的关系。
- NFS** 网络文件系统 (Network file system) 的缩写。一种 Sun 分布式文件系统，可以对不同网络上的远程文件系统进行完全透明的访问。
- NIS** SunOS 4.0 (最小) 网络信息服务 (SunOS 4.0 (minimum) Network Information Service) 的缩写。一种分布式网络数据库，包含与网络中系统和用户有关的关键信息。NIS 数据库存储在主服务器和所有从属服务器上。
- RAID** 廉价/单独磁盘冗余阵列 (Redundant array of inexpensive/independent disks) 的缩写。一种使用多个独立磁盘来可靠存储文件的磁盘技术。它可以在单个磁盘出现故障时防止数据丢失；提供容错磁盘环境；以及提供比单个磁盘更高的吞吐量。

RPC	远程过程调用 (remote procedure call) 的缩写。NFS 用以实施用户网络数据服务器的基本数据交换机制。
samfsdump	一个程序，用于为给定的文件组创建控制结构转储文件并复制所有控制结构信息。它与 UNIX <code>tar(1)</code> 实用程序类似，但它通常不复制文件数据。
samfsrestore	一个程序，用于从控制结构转储文件中恢复索引节点和目录信息。
SCSI	小型计算机系统接口 (Small Computer System Interface) 的缩写。一种电子通信技术规格，通常适用于磁盘驱动器、磁带驱动器和自动化库等外围设备。
Sun SAM-FS	Sun 存储及存档管理器文件系统 (Sun Storage and Archive Manager File System) 的缩写。Sun SAM-FS 软件可以控制对所有存储的文件以及主配置文件 (<code>mcf</code>) 中配置的所有设备的访问。
Sun SAM-QFS	即 Sun SAM-QFS 软件，由 Sun 存储及存档管理器与 Sun QFS 文件系统组合而成。Sun SAM-QFS 不仅为用户和管理员提供了高速的标准 UNIX 文件系统接口，而且还提供了存储及存档管理实用程序。它可以使用 Sun SAM-FS 命令集中的许多命令和标准 UNIX 文件系统命令。
Sun SAM-Remote 服务器 (Sun SAM-Remote server)	Sun SAM-Remote 服务器不仅是功能完备的 Sun SAM-FS 或 Sun SAM-QFS 存储管理服务器，而且还是可定义各个 Sun SAM-Remote 客户机共享库的 Sun SAM-Remote 服务器后台程序。
Sun SAM-Remote 客户机 (Sun SAM-Remote client)	Sun SAM-Remote 客户机可以是 Sun SAM-FS 系统，也可以是 Sun SAM-QFS 系统，用于建立包含多个伪设备的 Sun SAM-Remote 客户机后台程序。它可能有（也可能没有）自己的库设备。客户机用来存储一个或多个存档副本的存档介质由 Sun SAM-Remote 服务器决定。
tar	磁带存档 (Tape archive) 的缩写。它是 Sun SAM-FS 和 Sun SAM-QFS 软件用来存储存档图像的标准文件 / 数据记录格式。
TCP/IP	传输控制协议 / 网际协议 (Transmission Control Protocol / Internet Protocol) 的缩写。网际协议负责主机之间的寻址和路由以及数据信息包传递 (IP)，而传输控制协议负责在各个应用点之间可靠地传递数据 (TCP)。
VSN	卷序列名 (Volume serial name) 的缩写。如果您将数据存档到可移动介质卡盒，则 VSN 是指写入卷标中的磁带和光盘的逻辑标识。如果您将数据存档到磁盘高速缓存，则它表示该磁盘存档组的唯一名称。
WORM	单写多读 (Write once read many) 的缩写。一种介质存储类别，只能写入一次，但可以多次读取。

A

- 安装点 (mount point)** 安装文件系统的目录。
- 按优先级排列预备请求 (prioritizing preview requests)** 为不能立即满足的存档和登台请求分配优先级。

B

- 备份存储 (backup storage)** 一组文件的快照，旨在防止意外丢失。备份不仅包括文件的属性，而且还包括关联的数据。

C

- 拆分 (striping)** 一种以交叉方式将文件同时写入到多个逻辑磁盘的数据存取方法。对于每一个单独的文件系统，所有 Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 文件系统均允许用户声明是采用拆分存取方法，还是循环存取方法。Sun QFS 和 Sun SAM-QFS 文件系统允许用户在每一个文件系统中声明拆分组。另请参阅“循环”词汇条目。
- 拆分大小 (stripe size)** 移至拆分的下一个设备之前，要分配的磁盘分配单元 (DAU) 的数量。如果 `stripe=0`，文件系统将采用循环存取方式，而不采用拆分存取方式。
- 拆分组 (striped group)** Sun QFS 或 Sun SAM-QFS 文件系统及设备集合。在 `mcf` 文件中，它被定义成一个或多个（通常为两个）`gXXX` 设备。拆分组作为一个逻辑设备使用，并且始终拆分成若干个大小等于磁盘分配单元 (DAU) 的空间。一个文件系统中可以指定多达 128 个拆分组，但在整个系统中，最多不能超过 252 个设备。
- 超级块 (superblock)** 文件系统中的一种数据构造，用于定义文件系统的基本参数。它由系统写入到存储设备系列集中的所有分区，以识别该系列集中的各个分区成员。
- 传输器 (robot)** 自动化库的一部分，用于在存储端口和驱动器之间移动卡盒。也称传输设备。
- 磁盘拆分 (disk striping)** 指在数个磁盘上记录同一文件的过程，因此可以提高存取性能和增加整体存储容量。另请参阅“拆分”条目。
- 磁盘分配单元 (disk allocation unit)** 参阅 DAU。

**磁盘高速缓存
(disk cache)**

Sun SAM-FS 和 Sun SAM-QFS 文件系统软件的磁盘驻留部分。它用于在联机磁盘高速缓存和存档介质之间创建和管理数据文件。单个磁盘分区或整个磁盘均可用作磁盘高速缓存。

**磁盘缓冲器
(disk buffer)**

使用 Sun SAM-Remote 软件时，磁盘缓冲器是指在将数据从客户机存档到服务器时所用的服务器系统上的缓冲器。

**磁盘空间阈值
(disk space thresholds)**

管理员定义的供用户使用的磁盘空间数量。此项功能用于定义理想磁盘高速缓存利用率的范围。阈值上限表示磁盘高速缓存利用率的最高级别。阈值下限表示磁盘高速缓存利用率的最小级别。释放程序依据这些预定义的磁盘高速缓存空间阈值来控制磁盘高速缓存的利用率。

**存储端口
(storage slots)**

自动化库中的存储位置，当卡盒不在驱动器中使用时，将会存储在存储端口内。如果是直接连接的库，则存储端口的内容保存在自动化库的目录中。

**存储系列集
(storage family set)**

由一系列磁盘组成，整体表现为单个磁盘系列设备。

存档程序 (archiver)

一种可以自动将文件复制到可移动卡盒的存档软件程序。

**存档存储
(archive storage)**

已在存档介质中创建的文件数据副本。

**存档介质
(archive media)**

写入存档文件的介质。存档介质可以是库中的可移动磁带或磁光盘卡盒。此外，它还可以是另一系统中的安装点。

D

登台 (staging)

是指将近线或离线文件从存档存储设备恢复到在线存储设备的过程。

**多读取器文件系统
(Multireader file
system)**

Sun QFS 多读文件系统是一项单写入器 - 多读取器功能，使您可以指定能够安装在多台主机上的文件系统。多台主机可以读取该文件系统，但只有一台主机可以向该文件系统写入数据。多个读取器通过 `mount(1M)` 命令的 `-o reader` 选项指定。单一写入器主机通过 `mount(1M)` 命令中的 `-o writer` 选项指定。有关 `mount(1M)` 命令的详细信息，请参阅 `mount_samfs(1M)` 联机资料。

F

- 范围阵列 (extent array)** 文件索引节点中的阵列，用于定义分配给文件的每个数据块在磁盘上的位置。
- 分区 (partition)** 设备的一部分或磁光盘卡盒的一面。

G

- 光纤分布式数据接口 (fibre-distributed data interface)** 参阅 FDDI。
- 光纤信道 (fibre channel)** 由 ANSI 提出的标准，规定在设备之间实行高速串行通信。光纤信道是 SCSI-3 中使用的其中一个总线结构。

H

- 核查 (audit) (全面)** 载入卡盒并验证其 VSN 的过程。对于磁光盘卡盒，用于确定其容量和空间，然后输入到自动化库的目录中。
- 回收程序 (recycler)** 一种 Sun SAM-FS 和 Sun SAM-QFS 实用程序，用于回收由过期存档副本占用的卡盒空间。

J

- 计时器 (timer)** 一种限额软件，用于跟踪用户已在为其设定的软限制和硬限制之间经历的时间。
- 间接块 (indirect block)** 包含存储块列表的磁盘块。Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 文件系统最多可以具有三级间接块。第一级间接块包含用于存储数据的块的列表。第二级间接块包含第一级间接块的列表。第三级间接块包含第二级间接块的列表。
- 介质 (media)** 磁带或光盘卡盒。
- 介质回收 (media recycling)** 回收或重新使用一些利用率较低（即包含较少的存档文件）的存档介质的过程。

**近线存储设备
(nearline storage)**

一种可移动介质存储设备，访问此类设备之前需要启用自动安装功能。近线存储设备通常比在线存储设备便宜，但访问时间相对长一些。

**镜像写入
(mirror writing)**

在互不相连的磁盘组中保存两份副本的过程，用于防止因单个磁盘损坏而导致数据丢失的情况。

卷 (volume)

卡盒中用于存储共享数据的命名区域。一个卡盒中可以有一个或多个卷。双面卡盒有两个卷，每一面为一个卷。

**卷溢出
(volume overflow)**

一种可以使系统在多个卷上存储单个文件的功能。对于使用大容量文件（超过单个卷的容量）的场合，卷溢出功能非常有用。

K

卡盒 (cartridge)

一种包含数据记录介质的物品。例如，磁带或光盘。有时称为介质、卷或媒体。

**可寻址存储设备
(addressable storage)**

存储空间包括在线、近线、离站和离线存储设备，用户可以根据不同的情况在 Sun QFS、Sun SAM-FS 或 Sun SAM-QFS 文件系统中使用这些空间。

**可移动介质文件
(removable media file)**

一种特殊类型的用户文件，可以直接从它所在的可移动介质卡盒（如磁带或光盘卡盒）中进行访问。也可用于写入存档和登台文件数据。

**客户机 - 服务器
(client-server)**

分布式系统中的交互模式，在此模式下，一个站点上的程序向另一站点上的程序发送请求并等待回应。发送请求的程序称为“客户机”。提供回应的程序称为“服务器”。

库 (library)

参阅“自动化库”。

**库目录 (library
catalog)**

参阅“目录”。

块大小 (block size)

参阅 DAU。

**块分配图
(block allocation map)**

一个显示磁盘上每个可用存储块的位图，它指出了每个块的状态，即是在使用中还是未使用。

**宽限期
(grace period)**

对于磁盘限额，是指用户在达到为其设定的软限制后，系统允许用户继续创建文件和 / 或分配存储空间的时间期限。

L

**离线存储设备
(offline storage)**

一种要求操作员参与才能载入的存储设备。

**离站存储设备
(offsite storage)**

远离服务器的用于故障恢复的存储设备。

连接 (connection)

两个协议模块之间的通道，用于提供稳定可靠的数据流传输服务。TCP 连接可以从一台计算机上的 TCP 模块扩展到另一台计算机上的 TCP 模块。

M

名称空间 (name space)

一组文件的元数据部分，用于标识文件及其属性和存储位置。

目录 (catalog)

自动化库中 VSN 的记录。每个自动化库均有一个目录，并且一个站点有一个 historian，用于记录所有自动化库。

目录 (directory)

指向文件系统中其它文件和目录的文件数据结构。

N

内核 (kernel)

提供基本系统功能的中央控制程序。UNIX 内核可以创建和管理进程；提供存取文件系统的功能；提供基本安全性能；以及提供通信功能。

Q

驱动器 (drive)

用于在可移动介质卷中存取数据的机械装置。

**全局指令
(global directives)**

应用于所有文件系统的存档程序和释放程序指令，位于第一个 fs = 行之前。

R

软限制 (soft limit) 对于磁盘限额，是指用户可以临时超量使用的文件系统资源（块或索引节点）的阈值限制。超过软限制时，系统会启动一个计时器。当超过软限制的时间大于指定时间（默认值为一星期）时，用户将不能再超量使用系统资源，直到减少文件系统的使用至低于软限制的水平。

S

设备日志 (device logging) 一项可配置的功能，用于提供设备专用的错误信息，以供分析设备问题。

设备扫描程序 (device scanner) Sun SAM-FS 或 Sun SAM-QFS 文件系统中的一种软件，用于定期监视所有手动安装的可移动设备，并检测是否存在可由用户或其它进程请求的安装卡盒。

设备系列集 (family device set) 参阅“系列集”。

释放程序 (releaser) 一个 Sun SAM-FS 和 Sun SAM-QFS 组件，用于识别已存档的文件并释放它们的磁盘高速缓存副本，从而腾出更多的磁盘高速缓存空间。释放程序可以自动将在线磁盘存储量调整到阈值上限和阈值下限。

释放优先级 (release priority) 一种计算文件系统中文件的释放优先级的方法，它通过各种加权数乘以相应的文件属性并得出各个结果之和，从而确定文件的优先级。

数据设备 (data device) 对于 Sun QFS、Sun SAM-FS 或 Sun SAM-QFS 文件系统来说，是指用于存储文件数据的单个设备或设备组。

索引节点 (inode) 索引节点 (Index node) 的缩写。文件系统用于描述文件的数据结构。索引节点描述了与文件关联的所有属性（名称属性除外）。属性包括所有权、存取、权限、大小和文件在磁盘系统上的位置。

索引节点文件 (inode file) 文件系统中的特殊文件 (.inodes)，包含该文件系统中存储的所有文件的索引节点结构。所有 Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 索引节点的大小均为 512 字节。索引节点文件是一个元数据文件，它不同于 Sun QFS 和 Sun SAM-QFS 文件系统中的其它数据文件。

T

通过网络连接的自动化库 (network-attached automated library)

许多厂家生产的库，如 StorageTek、ADIC/Grau、IBM 或 Sony，它们分别由各自制造商提供的软件包进行控制。通过使用专为这些自动化库设计的 Sun SAM-FS 或 Sun SAM-QFS 介质更换器后台程序，Sun SAM-FS 和 Sun SAM-QFS 文件系统可以与制造商软件相互通信。

W

伪设备 (pseudo device)

未关联任何硬件的软件子系统或驱动程序。

文件系统 (file system)

由分层的文件和目录组成的集合。

文件系统专用指令 (file system specific directives)

位于全局指令后面的存档程序和释放程序指令，专用于特定的文件系统，以 fs = 开头。文件系统专用指令的应用范围到出现下一个 fs = 指令行或文件末尾（如果遇到）结束。如果多个指令影响到一个文件系统，则文件系统专用指令会取代全局指令。

X

系列集 (Family Set)

由一组独立物理设备组成的存储设备，例如自动化库中的磁盘组或驱动器组。另请参阅“磁盘高速缓存系列集”。

限额 (quota)

允许用户使用的系统资源量。限额不适用于可移动介质或磁盘存档资源。

小型计算机系统接口 (Small Computer System Interface)

参阅 SCSI。

循环 (round robin) 一种按顺序将全体文件写入到多个逻辑磁盘的数据存取方法。当将单个文件写入磁盘时，这个文件的全部内容将写入第一个逻辑磁盘。第二个文件将写入下一个逻辑磁盘，依次类推。每个文件的大小决定 I/O 的大小。

默认情况下，Sun QFS、Sun SAM-FS 和 Sun SAM-QFS 文件系统均采用拆分数据存取方法（除非存在拆分组）。如果指定循环存取方法，则采用循环方法存取文件。如果文件系统包含不匹配的拆分组，系统将不支持拆分功能，并且会强制实行循环存取方法。

另请参阅“磁盘拆分”和“拆分”条目。

Y

以太网 (Ethernet) 一种局域包交换网络技术。它的最初设计是使用同轴电缆，随着技术的进步，价格便宜的屏蔽双绞线目前已逐渐取代了同轴电缆。以太网是指运行速度为 10 MB/s 或 100 MB/s 的局域网。

硬限制 (hard limit) 对于磁盘限额，是指用户不能超量使用的文件系统资源（块或索引节点）的最大限制。

预分配 (preallocation) 在磁盘高速缓存上预先保留一定数量的连续空间以备写入文件的过程。这可以确保获得连续的空间。只能对大小为零的文件执行预分配操作。也就是说，您只能对大小为零的文件运行 `setfa -l` 命令。有关详细信息，请参阅 `setfa(1M)` 联机资料。

阈值 (thresholds) 一种为在线存储设备定义适当的可用存储空间的机制。阈值用于设置释放程序的存储目标。另请参阅“磁盘空间阈值”。

元数据 (metadata) 与数据有关的数据。元数据是指用于在磁盘上查找某个文件的具体数据位置的索引信息。它由以下各项的有关信息组成：文件、目录、访问控制列表、符号链接、可移除介质、分段文件和分段文件索引。元数据用于确定数据的位置。在丢失数据时，您必须先恢复元数据才能恢复丢失的数据，因此，元数据必须得到保护。

**元数据设备
(metadata device)**

用于存储 Sun QFS 和 Sun SAM-QFS 文件系统元数据的独立设备，如固态硬盘或镜像设备等。将文件数据和元数据单独存放可以提高系统的性能。在 `mcf` 文件中，元数据设备被声明为 `ma` 文件系统系统中的 `mm` 设备。

**远程过程调用
(remote procedure call)**

参阅 RPC。

Z

在线存储设备 (online storage)

可以随时访问的存储设备，如磁盘高速缓存等。

直接 I/O (direct I/O)

一种用于大型块对齐连续 I/O 的属性。setfa(1) 命令的 -D 选项是设置直接 I/O 属性的选项。它用于为文件或目录设置直接 I/O 属性。如果应用到目录，则直接 I/O 属性可以继承。

直接访问 (direct access)

一个文件属性（永远不必登台），表示可从存档介质直接访问近线文件，而无需检索磁盘高速缓存。

直接连接库 (direct-attached library)

是指通过 SCSI 接口直接连接到服务器的自动化库。Sun SAM-FS 或 Sun SAM-QFS 软件可以通过使用自动化库的 SCSI 标准来直接控制 SCSI 连接库。

自动化库 (automated library)

一种自动控制的设备，它可在无操作人员参与的情况下，载入和卸载可移动介质卡盒。自动化库包括两个部分：一个或多个驱动器；以及用于将卡盒移入或移出存储端口和驱动器的传输装置。

租借 (lease)

在 Sun QFS 共享文件系统中，租借用于向客户机主机授予权限，使其可在租借有效期内对文件进行操作。元数据服务器向每一个客户机主机发放租借。它随时可以根据需要更新租借，以使客户机主机能够继续对文件进行操作。

索引

符号

- ! 命令 (samu(1M) 命令), 253
- .Xdefaults 文件, 192
- /dev/samst 条目, 13
- /etc/opt/SUNWsamfs/archiver.cmd, 参阅 archiver.cmd 文件
- /etc/opt/SUNWsamfs/defaults.conf, 参阅 defaults.conf 文件
- /etc/opt/SUNWsamfs/diskvols.conf, 参阅 diskvols.conf 文件
- /etc/opt/SUNWsamfs/LICENSE.4.0 参阅 LICENSE.4.0 文件
- /etc/opt/SUNWsamfs/mcf, 参阅 mcf 文件
- /etc/opt/SUNWsamfs/preview.cmd, 参阅 preview.cmd 文件
- /etc/opt/SUNWsamfs/recycler.cmd, 参阅 recycler.cmd 文件
- /etc/opt/SUNWsamfs/SamGUI.rsc, 参阅 SAMGUI.rsc 文件
- /opt/SUNWsamfs/examples/recycler.sh, 参阅 recycler.sh 文件
- /opt/SUNWsamfs/sbin/archiver.sh, 参阅 archiver.sh 脚本
- /var/opt/SUNWsamfs/devlog 文件, 参阅 devlog 文件
- /var/opt/SUNWsamfs/sef/sefdata, 参阅 sefdata 文件

字母

- a 显示控制命令 (samu(1M) 命令), 247
- ACSAPI 接口, 12, 63
- ADIC Scalar 系列库
 - 操作, 47
 - 概述, 47
 - 配置, 47
- ADIC/Grau 自动化库
 - 参数文件, 42
 - 操作, 45
 - 概述, 42
 - 配置, 42
 - 添入目录, 44
 - 诊断信息, 47
- age_priority preview.cmd 指令, 159
- allsets 存档组, 74, 101
- Ampex 自动化库
 - 操作, 49
 - 概述, 47
 - 格式化 D2 磁带, 49
 - 配置, 48
- API 例程, 9
- archive(1) 命令, 4
- archive_audit(1M) 命令, 8
- archiver(1M) 命令, 8, 84
- archiver.cmd 文件, 73, 80, 82, 114, 132, 173
- archiver.sh(4) 脚本, 90
- archmax 存档程序指令, 88

- aridle samu(1M) 命令, 245
- arrestart samu(1M) 命令, 245
- arrun samu(1M) 命令, 245
- arstop samu(1M) 命令, 245
- audit 传输器命令 (samu(1M) 命令), 251
- audit(1M) 命令, 52, 61, 68
- auditslot(1M) 命令, 7, 28
- bufsize 存档程序指令, 101
- bufsize 存档程序指令, 87
- bufsize 登台程序指令, 155
- build_cat(1M) 命令, 7, 19, 44, 51, 61, 68
- chmed(1M) 命令, 7, 33, 171
- cleandrive(1M) 命令, 7, 33
- clear 命令 (samu(1M) 命令), 252
- crontab 条目, 175
- d2format(1M) 命令, 49
- DAS 服务器, 42
- defaults.conf 文件, 14, 15, 182, 266
- dev_down.sh(4) 命令, 8
- devicetool(1M) 命令, 9, 205
- devlog
 - samu(1M) 命令, 246
 - 文件, 264
- disk_archive 存档程序指令, 116
- diskvols.conf 文件, 114
- display_all_candidates 释放程序指令, 146
- DLT 磁带驱动器, 260
- dmpshm(1M) 命令, 8
- down samu(1M) 命令, 246
- drivemin 存档程序指令, 103
- drives 存档程序指令, 88
- drives 存档组参数指令, 102
- drives 登台程序指令, 154
- dtrace 命令 (samu(1M) 命令), 252
- du(1) 命令, 4
- dump_cat(1M) 命令, 7, 19
- DZC-8000S 接口, 12, 59
- endparams 存档程序指令, 101
- endvsnpools 存档程序指令, 112
- endvsns 存档程序指令, 111
- exarchive(1M) 命令, 8
- export 传输器命令 (samu(1M) 命令), 251
- exported_media 指令, 20
- find(1) 命令, 另请参阅 sfind(1) 命令, 4
- fs 存档程序指令, 91
- fs 释放程序指令, 145
- Fujitsu LMF 自动化库
 - fujitsulmf 接口, 49
 - 参数文件, 50
 - 操作, 52
 - 概述, 49
 - 配置, 50
 - 添入目录, 51
- Grau 自动化库, 参阅 ADIC/Grau 自动化库
- GRAU/ABBA 库, 参阅 ADIC/Grau 自动化库
- grauaci 接口, 42
- group 存档程序指令, 93
- GUI 工具
 - 概述, 181
 - 另请参阅 libmgr(1M)、samtool(1M)、robottool(1M)、devicetool(1M) 或 previewtool(1M)
 - 命令, 9
- hlwm_priority previewer 指令, 160
- hwm_priority previewer 指令, 160
- I/O 活动, 247
- IBM 3494 库, 55
- IBM 3584 自动化库
 - 导入, 54
 - 分区, 54
 - 概述, 54
 - 清洁, 54
- IBM 自动化库, 12
- idle samu(1M) 命令, 246
- ignore 回收程序指令, 175
- import 传输器命令 (samu(1M) 命令), 251
- import(1M) 命令, 7, 20, 31, 36, 44, 46, 51, 53, 56, 61, 62, 67, 70
- interval 存档程序指令, 86
- itemize(1M) 命令, 8

- join path 存档程序指令, 104
- Jukebox, 参阅自动化库
- lhwm_priority previewer 指令, 160
- libmgr(1M)
 - 传输器显示屏幕, 184
 - 概述, 182
 - 介质操作显示屏幕, 185
 - 介质驱动器操作显示屏幕, 185
 - 命令, 9, 182
 - 驱动器状态, 187
 - 设备状态, 187
 - 显示, 183
- libsam, 9
- libsamrpc, 9
- LICENSE.4.0 文件, 227
- lmcpcd 接口, 12
- load 传输器命令 (samu(1M) 命令), 251
- load_notify.sh(1M) 命令, 8
- lock 存档程序指令, 102
- logfile
 - 存档程序指令, 90
 - 登台程序指令, 156
 - 回收程序指令, 167
 - 释放程序指令, 146
- ls(1) 命令, 参阅 sls(1) 命令
- lwm_priority previewer 指令, 160
- madev(1M) 命令, 参阅 samdev(1M) 命令
- maxactive 登台程序指令, 157
- maxsize 存档程序指令, 93
- mcf 文件
 - libmgr(1M) 中的显示, 183
 - 概述, 13
 - 库历史记录, 20
 - 用途, 3
 - 自动化库配置, 12
- meta_timeo 文件系统命令 (samu(1M) 命令), 249
- min_residence_age 释放程序指令, 146
- minsize 存档程序指令, 93
- mount 命令 (samu(1M) 命令), 252
- mount(1M) 命令, 6
- move(1M) 命令, 7, 57, 58
- n 显示控制命令 (samu(1M) 命令), 247
- name 存档程序指令, 93
- no_archive 存档组, 74, 92
- no_recycle 回收程序指令, 168
- no_release 释放程序指令, 146
- norelease 存档程序指令, 98
- notify 存档程序指令, 90
- notrace 文件系统命令 (samu(1M) 命令), 249
- odlabel(1M) 命令, 7, 28
- off samu(1M) 命令, 246
- on samu(1M) 命令, 246
- open 命令 (samu(1M) 命令), 253
- ovflmin 存档程序指令, 89
- p 显示控制命令 (samu(1M) 命令), 247
- params 存档程序指令, 101
- partial 文件系统命令 (samu(1M) 命令), 249
- PetaSite 控制器, 59
 - pool 存档程序指令, 111
- preview.cmd 文件, 另请参阅预备请求, 158, 162
- previewtool(1M) 命令, 9, 212
 - priority 存档程序指令, 109
- PSC 控制器, 59
- q 显示控制命令 (samu(1M) 命令), 247
- qfsdump(1M) 命令, 6
- qfsrestore(1M) 命令, 6
- r 显示控制命令 (samu(1M) 命令), 247
- read 命令 (samu(1M) 命令), 253
- readahead
 - 文件系统命令 (samu(1M) 命令), 249
- research(1M) 命令, 8
- research_no_release 释放程序指令, 148
- recycle_dataquantity 存档程序指令, 173
- recycle_hwm 存档程序指令, 173
- recycle_ignore 存档程序指令, 173, 175
- recycle_mailaddr 存档程序指令, 173
- recycle_mingain 存档程序指令, 173
- recycle_vsncount 存档程序指令, 173
- recycler.cmd 文件, 170

- recycler.sh 文件, 176
- release 存档程序指令, 96, 98
- release(1) 命令, 4, 137
- releaser.cmd 文件, 142, 149
- request(1) 命令, 4, 169, 266
- reserve 存档程序指令, 106
- reserve(1M) 命令, 8
- robottool(1M)
 - 概述, 193
 - 命令, 10, 195
 - 设备显示屏幕, 196
 - 显示区域, 195
 - 状态位, 197
- sam_segment(3), 268
- sam-archiverd 后台程序, 80
- sam-arcopy 后台程序, 80
- sam-arfind 后台程序, 80
- sambcheck(1M) 命令, 6
- sam-catservd 后台程序, 18
- samchaid(1M) 命令, 6
- samcmd(1M) 命令, 5, 23
- samd(1M) 命令, 5, 23
- samdev(1M) 命令, 7, 8
- samexport(1M) 命令, 7, 20, 37, 46, 53, 57, 63, 71
- samfsck(1M) 命令, 6
- samfsconfig(1M) 命令, 6
- sam-fsd 后台程序, 80
- samfsdump(1M) 命令, 6
- samfsinfo(1M) 命令, 6
- samfsrestore(1M) 命令, 6
- sam-genericd 后台程序, 12
- samgrowfs(1M) 命令, 6
- SAMGUL.rsc 文件, 183
- sam-ibm3494d 后台程序, 12
- sam-initd 后台程序, 12
- samload(1M) 命令, 8
- sammkfs(1M) 命令, 6
- samncheck(1M) 命令, 6
- samquota(1M) 命令, 6
- samquotastat(1M) 命令, 7
- sam-recycler(1M) 命令, 8, 174
- sam-releaser(1M) 命令, 8, 137
- sam-robotsd 后台程序, 12
- sam-rpcd 后台程序, 9
- samset(1M) 命令, 5, 9, 265
- samsharefs(1M) 命令, 7
- sam-sonyd 后台程序, 12
- sam-stkd 后台程序, 12
- samtool(1M)
 - 概述, 191
 - 更新显示, 192
 - 命令, 10, 191
 - 启动, 191
 - 刷新字段, 192
 - 退出, 191
 - 选择工具, 191
- samtrace(1M) 命令, 7
- samu(1M)
 - meta_timeo 文件系统命令, 249
- samu(1M)
 - ! 命令, 253
 - a 显示控制命令, 247
 - aridle 命令, 245
 - arrestart 命令, 245
 - arrun 命令, 245
 - arstop 命令, 245
 - audit 传输器命令, 251
 - clear 命令, 252
 - devlog 命令, 246
 - down 命令, 246
 - dtrace 命令, 252
 - export 传输器命令, 251
 - idle 命令, 246
 - import 传输器命令, 251
 - load 传输器命令, 251
 - mount 命令, 252
 - n 显示控制命令, 247
 - notrace 文件系统命令, 249
 - off 命令, 246
 - on 命令, 246
 - open 命令, 253
 - p 显示控制命令, 247
 - partial 文件系统命令, 249

- q 显示控制命令, 247
- r 显示控制命令, 247
- read 命令, 253
- readahead 文件系统命令, 249
- snap 命令, 253
- thresh 文件系统命令, 250
- trace 文件系统命令, 250
- u 显示控制命令, 247
- unavail 命令, 246
- unload 命令, 246
- v 显示控制命令, 247
- w 显示控制命令, 247
- writebehind 文件系统命令, 250
- 操作员显示屏幕, 222
- 查看 I/O 活动, 247
- 传输器命令, 251
- 后台程序跟踪命令, 252
- 界面, 218
- 启动命令, 5, 10, 217
- 使用 samu(1M) 查看状态, 245
- 刷新显示控制命令, 247
- 文件系统命令, 249
- 显示控制 samu(1M) 命令, 245
- 显示屏幕键, 218
- 与 mcf 文件交互作用, 220
- 状态代码, 242
- samu(1M) 中的待命登台, 241
- samunhold(1M) 命令, 7
- sdu(1) 命令, 4
- SEF, 269
- sefdata 文件, 270
- sefreport(1M) 命令, 269
- segment(1) 命令, 4, 268
- set_admin.sh(1M) 命令, 9
- set_state(1M) 命令, 9
- setfa(1) 命令, 4
- sfind(1) 命令, 4
- showqueue(1M) 命令, 8
- sls(1) 命令, 4
- snap 命令 (samu(1M) 命令), 253
- Sony B9 和 B35 自动化库
 - 操作, 55
 - 概述, 55
 - 配置, 55
- Sony PetaSite 自动化库
 - 操作, 56
 - 概述, 56
- Sony 网络连接自动化库
 - 参数文件, 59
 - 操作, 62
 - 概述, 59
 - 配置, 59
 - 添入库目录, 61
- Sony 自动化库, 12
 - sort 存档程序指令, 104
 - squota(1) 命令, 5
 - ssi.sh 脚本, 67
 - ssum(1) 命令, 5
 - stage 存档程序指令, 96
 - stage(1) 命令, 5
 - star(1M) 命令, 9
- StorageTek ACSLS 连接自动化库
 - 参数文件, 64
 - 操作, 70
 - 概述, 63
 - 配置, 64
 - 添入库目录, 67
- StorageTek 自动化库, 12, 63
- Sun SAM-FS
 - GUI 工具概述, 181
 - 停止, 23
- Sun SAM-QFS
 - 参阅 Sun SAM-FS
- tapenonstop 存档程序指令, 106
- tar(1) 命令, 参阅 star(1M)
- thresh 文件系统命令 (samu(1M) 命令), 250
- tplabel, 7
- tplabel(1M) 命令, 27, 49
- trace 文件系统命令 (samu(1M) 命令), 250
- trace_rotate.sh(1M) 命令, 7
- u 显示控制命令 (samu(1M) 命令), 247
- unarchive(1M) 命令, 9
- unavail samu(1M) 命令, 246

- undamage(1M) 命令, 9
- unload samu(1M) 命令, 246
- unload(1M) 命令, 8
- unresearch(1M) 命令, 8
- unreserve(1M) 命令, 8
- user 存档程序指令, 93
- v 显示控制命令 (samu(1M) 命令), 247
- VSN
 - pool 指令, 112
 - 保留, 106
 - 操作库目录, 200
 - 磁盘存档指令, 114
 - 关联指令, 111
 - 目录
 - libmgr(1M), 189
 - robottool(1M), 196
 - 使用标准表达式, 112
 - 最小增益 /VSN, 172
- vsn_priority preview.cmd 指令, 159
- vsnpools 存档程序指令, 112
- vsns 存档程序指令, 111
- w 显示控制命令 (samu(1M) 命令), 247
- wait 存档程序指令, 89
- weight_age 释放程序指令, 143
- weight_age_access 释放程序指令, 143
- weight_age_modify 释放程序指令, 143
- weight_age_residence 释放程序指令, 143
- weight_size 释放程序指令, 144
- wm_priority 因子, 160
- writebehind
 - 文件系统命令 (samu(1M) 命令), 250

B

- 标记介质
 - 命令, 26
 - 使用 devicetool(1M), 211
 - 使用 robottool(1M), 202
- 标准表达式, 93
- 部分释放和部分登台功能, 140

C

- 参数文件, 13, 42, 50, 59, 64
- 操作员权限级别, 182
- 操作员实用程序, 参阅 samu(1M)
- 重新存档, 定义, 166
- 处理登台请求时出错, 2
- 处于, 24
- 传输器
 - samu(1M) 中的命令, 251
 - 另请参阅自动化库
- 磁带
 - 标签, 15
 - 使用 samu(1M) 显示驱动器状态, 236
- 磁盘存档, 114, 115, 166
- 磁盘高速缓存
 - 上限, 138
 - 释放优先级, 2
 - 下限, 138
 - 阈值, 2
- 存档程序
 - archiver.cmd 示例, 84
 - samu(1M) 显示屏幕, 223
 - VSN 池, 112
 - VSN 关联指令, 111
 - 保留 VSN, 107
 - 操作理论, 73
 - 磁盘存档, 参阅磁盘存档
 - 存档介质定义, 1, 73
 - 存档时间间隔定义, 74
 - 存档时限定义, 74
 - 存档组, 74
 - 存档组成员关系冲突, 97
 - 存档组成员指令, 91
 - 存档组处理指令, 101
 - 定义, 1, 73
 - 防止存档, 92
 - 副本创建, 2
 - 副本定义指令, 98
 - 概述, 2, 83, 86
 - 后台程序, 80
 - 控制文件大小, 88
 - 联合存档, 104
 - 命令, 8

- 默认值, 74
- 排除故障, 133
- 取消存档, 105
- 确定存档时间, 105
- 日志文件, 81, 133
- 删除条目, 9
- 设置存档时限, 99
- 设置优先级, 109
- 设置自动取消存档, 100
- 使用标准表达式, 93
- 示例, 119
- 原则, 132
- 在 archiver.cmd 中指定文件系统, 91
- 指定文件系统数据的副本份数, 100
- 指令, 另请参阅指令, 83, 86

D

- 大容量存储设备状态显示屏幕 (samu(1M)), 228
- 当前数据空间, 定义, 165
- 导出介质
 - 概述, 20
 - 命令, 35, 37
 - 使用 robottool(1M), 199, 201
 - 使用 samu(1M), 251
- 导入介质
 - 概述, 20
 - 命令, 35, 37
 - 使用 robottool(1M), 199
 - 使用 samu(1M), 251
- 登台程序
 - bufsize 指令, 155
 - drives 指令, 154
 - logfile 字段, 156
 - maxactive 指令, 157
 - samu(1M) 中的待定登台, 241
 - 部分登台, 140
 - 处理登台请求时出错, 2
 - 存档在登台过程中的作用, 158
 - 定义, 2
 - 概述, 2, 153
 - 日志活动, 156

- 使用 samu(1M) 显示状态, 229, 237
- 指令, 153

读前区

- 使用 devicetool(1M) 设置, 212

- 端口, 添加, 256

F

- 分段文件, 268

G

- 共享驱动器, 16
- 管理员
 - 权限级别, 182
 - 实用程序, 参阅 samu(1M)
- 光盘状态显示屏幕, 230
- 过期数据空间, 定义, 165

H

核查

- 卷, 28
- 使用 devicetool(1M) 核查设备, 210
- 自动化库, 29

后台程序

- sam-archiverd, 80
- sam-arcopy, 80
- sam-arfind, 80
- sam-catserverd, 18
- sam-fsd, 80
- sam-genericd, 12
- sam-ibm3494d, 12
- sam-initd, 12
- sam-robotd, 12
- sam-rpcd, 9
- sam-sonyd, 12
- sam-stkd, 12
- samu(1M) 显示屏幕, 225
- 跟踪命令 (samu(1M) 命令), 252
- 自动化库后台程序, 12

恢复数据

另请参阅故障恢复, 268

回收程序

crontab 条目, 175

ignore 选项, 172

logfile 指令, 167

mail 选项, 172

no_recycle 指令, 168

recycler.cmd 示例, 171

recycler.sh 文件, 176

操作理论, 167

磁盘存档副本, 166

定义, 2

概述, 3, 165

排除故障, 179

配置, 169

使用 chmed(1M) 命令强制, 171

指令, 167

J

将光盘编入目录, 8

校验和属性, 5

界限指令, 160

介质

标记, 26

错误, 33

更换器, 参阅自动化库

基本操作, 181, 217

库, 参阅自动化库

类型, 21

使用 samu(1M) 显示状态, 233

移动, 20

载入, 25

载入请求显示屏幕, samu(1M), 231, 247

卷序列名, 参阅 VSN

卷溢出

ovflmin 存档程序指令, 89

概述, 267

K

卡盒

标记, 26

导出, 37

导入, 36

清洁, 30

取出, 34

卸载, 8, 26, 39

载入, 8, 25, 39

看管指令, 20

可移动介质文件, 266

可用空间, 定义, 165

库, 参阅自动化库

库管理器, 参阅 libmgr(1M)

库历史记录, 20

库目录

查看, 40

概述, 18

建立, 12, 18

L

联合存档, 104

M

命令

archive(1), 4

archive_audit(1M), 8

archiver(1M), 8, 84

audit(1M), 52, 61, 68

auditslot(1M), 7, 28

build_cat(1M), 7, 19, 44, 51, 61, 68

chmed(1M), 7, 33, 171

cleandrive(1M), 7, 33

d2format(1M), 49

dev_down.sh(4), 8

devicetool(1M), 9

dmpshm(1M), 8

du(1), 4

dump_cat(1M), 7, 19

exarchive(1M), 8
 find(1), 另请参阅 sfind(1) 命令, 4
 GUI, 9
 import(1M), 7, 20, 31, 36, 44, 46, 51, 53, 56, 61, 62, 67, 70
 itemize(1M), 8
 libmgr(1M), 9
 load_notify.sh(1M), 8
 ls(1), 另请参阅 sls(1) 命令, 4
 makedev(1M), 参阅 samdev(1M)
 mount(1M), 6
 move(1M), 7, 57, 58
 odlabel(1M), 7, 28
 previewtool(1M), 9, 212
 qfsdump(1M), 6
 qfsrestore(1M), 6
 rearch(1M), 8
 release(1), 4, 137
 request(1), 4, 169, 266
 reserve(1M), 8
 robottool(1M), 10
 sambcheck(1M), 6
 samchaid(1M), 6
 samcmd(1M), 5, 23
 samd(1M), 5, 23
 samdev(1M), 7, 8
 samexport(1M), 7, 20, 37, 46, 53, 57, 63, 71
 samfsck(1M), 6
 samfsconfig(1M), 6
 samfsdump(1M), 6
 samfsinfo(1M), 6
 samfsrestore(1M), 6
 samgrowfs(1M), 6
 samload(1M), 8
 sammkfs(1M), 6
 samncheck(1M), 6
 samquota(1M), 6
 samquotastat(1M), 7
 sam-recycler(1M), 8, 174
 sam-releaser(1M), 8, 137
 samset(1M), 5, 9, 265
 samsharefs(1M), 7
 samtool(1M), 10
 samtrace(1M), 7
 samu(1M), 5, 10, 217
 samunhold(1M), 7
 sdu(1), 4
 sefreport(1M), 269
 segment(1), 4, 268
 set_admin.sh(1M), 9
 set_state(1M), 9
 setfa(1), 4
 sfind(1), 4
 showqueue(1M), 8
 sls(1), 4
 squash(1), 5
 ssum(1), 5
 stage(1), 5
 star(1M), 9
 tar(1), 参阅 star(1M) 命令
 tplabel, 7
 tplabel(1M), 27, 49
 trace_rotate.sh(1M), 7
 unarchive(1M), 9
 undamage(1M), 9
 unload(1M), 8
 unrearch(1M), 8
 unreserve(1M), 8
 文件系统, 6
 一般系统管理员, 5
 用户, 4
 自动化库, 7
 默认值
 存档程序, 74
 设置系统默认值, 9
P
 排除故障
 存档程序, 133
 回收程序, 179
 释放程序, 151
 配置
 samu(1M) 显示屏幕, 224
 回收程序, 170
 文件, 参阅 mcf
 自动化库, 12

Q

- 启动 Sun SAM-FS, 24
- 清洁
 - 磁带驱动器, 32
- 请求文件, 参阅可移动介质文件
- 驱动器
 - 共享, 16
 - 清洁, 30, 32
- 取消存档, 100, 105
- 权限级别, 182

R

- 日志文件
 - 存档程序, 81, 90, 133
 - 登台程序, 156
 - 回收程序, 174
 - 设备日志, 263
 - 释放程序, 146
- 容量, 定义, 165
- 软件升级, 255

S

- 删除已损坏的文件, 9
- 上限
 - previewer 指令, 160
 - 回收程序, 171
- 设备
 - devicetool(1M) 命令, 206
 - devlog samu(1M) 命令, 247
 - samu(1M) 中的命令, 246
 - 创建 Sun SAM-FS, 8
 - 代码, 使用 samu(1M) 查看, 242, 243
 - 计时值, 15
 - 链接, 8
 - 日志, 参阅日志文件
 - 设置状态, 9
 - 使用 samu(1M) 显示状态, 235
 - 卸载值, 16

- 状态, 24, 25
- 状态, 使用 samu(1M) 查看, 244

- 升级硬件和软件, 255
- 使用 devicetool(1M) 设置阈值, 212
- 使用 robottool(1M) 移动介质, 204

释放程序

- fs 指令, 145
- 备选文件, 139
- 部分释放, 140
- 定义, 1
- 概述, 2, 137, 138
- 命令文件, 142
- 排除故障, 135, 151
- 配置, 149
- 权数, 139
- 日志文件, 146
- 时段, 139
- 手动操作, 150
- 优先级, 139
- 指令, 142

刷新显示控制命令 (samu(1M) 命令), 247

T

- 添加端口, 256
- 条码, 15
- 停止 Sun SAM-FS, 23
- 图形用户界面
 - 参阅 GUI 工具

W

- 网络连接自动化库, 参阅自动化库
- 文件, 设置属性, 96
- 文件系统
 - samu(1M) 中的命令, 249
 - 概述, 1
 - 命令, 6
 - 使用 samu(1M) 显示, 227
 - 数据, 100

X

系统错误工具, 参阅 SEF

下限指令, 160

小型计算机系统接口

写后区

 使用 devicetool(1M) 设置, 212

卸载介质

 使用 devicetool(1M), 210

 使用 robottool(1M), 200,202

 使用 samu(1M), 26,39,246

许可证

 samu(1M) 显示屏幕, 227

 一般信息, xxi

 在自动化库中添加端口, 256

Y

一般系统管理员命令, 5

应用程序接口, 参阅 API

硬件升级, 255

用户命令, 4

预备请求

 age_priority 指令, 159

 hlwm_priority 指令, 160

 hwm_priority 指令, 160

 lhwm_priority 指令, 160

 lwm_priority 指令, 160

 vsns_priority 指令, 159

 初始化, 161

 计划, 161

 计算优先级, 161

 排列优先顺序, 158

Z

载入介质

 使用 robottool(1M), 200,202

 使用 samu(1M), 25

 使用 samu(1M) 查看安装请求, 247

 手动载入驱动器, 39

直接连接自动化库, 参阅自动化库

指令

 previewer

 age_priority, 159

 hlwm_priority, 160

 hwm_priority, 160

 lhwm_priority, 160

 lwm_priority, 160

 vsns_priority, 159

 存档程序

 archmax, 88

 -bufsize, 101

 bufsize, 87

 -disk_archive, 114,116

 -drivemin, 103

 -drives, 102

 drives, 88

 endparams, 101

 endvsnpools, 112

 endvsns, 111

 fs, 91

 -group, 93

 interval, 86

 -join path, 104

 -lock, 102

 logfile, 90

 -maxsize, 93

 -minsize, 93

 -name, 93

 -norelease, 98

 notify, 90

 ovflmin, 89

 params, 101

 -pool, 111

 -priority, 109

 -recycle_dataquantity 指令, 173

 -recycle_hwm 指令, 173

 -recycle_ignore 指令, 173,175

 -recycle_mailaddr 指令, 173

 -recycle_mingain 指令, 173

 -recycle_vsncount 指令, 173

 -release, 96,98

 -reserve, 106

 -sort, 104

 -stage, 96

 -tapenonstop, 106

 -user, 93

 vsnpools, 112

 vsns, 111

- wait, 89
- 存档组成员, 91
- 存档组副本编号, 98
- 回收指令, 104
- 设置存档时限, 99
- 设置自动取消存档, 100
- 指定文件系统数据的副本份数, 100
- 登台程序
 - bufsize, 155
 - drives, 154
 - logfile, 156
 - maxactive, 157
- 回收程序
 - ignore 指令, 175
 - logfile, 167
 - no_recycle, 168
- 释放程序
 - display_all_candidates, 146
 - fs, 145
 - logfile, 146
 - min_residence_age, 146
 - no_release, 146
 - rearch_no_release, 148
 - weight_age, 143
 - weight_age_access, 143
 - weight_age_modify, 143
 - weight_age_residence, 143
 - weight_size, 144
 - 释放优先级指令, 143
- 主配置文件
 - 参阅 mcf 文件
- 状态代码
 - 使用 robottool(1M) 查看, 205
 - 使用 samu(1M) 查看, 242, 243
- 自动化库
 - IBM, 12
 - SCSI 连接, 参阅自动化库, 直接连接
 - Sony, 12
 - StorageTek, 12
 - 参数文件, 13
 - 操作, 18
 - 打开, 24
 - 定义, 11
 - 关闭, 25
 - 核查, 29
 - 后台程序, 12
 - 历史记录
 - 参阅库历史记录, 20
 - 另请参阅各个供应商的条目, 12
 - 命令, 7
 - 目录, 12
 - 配置, 12
 - 网络连接, 12, 13, 18
 - 直接连接, 12, 13, 18
- 自动清洁, 23
- 自动载入, 23