

Oracle® Hierarchical Storage Manager and StorageTek QFS Software

维护和管理指南

发行版 6.1

E56775-03

2016 年 3 月

Oracle® Hierarchical Storage Manager and StorageTek QFS Software
维护和管理指南

E56775-03

版权所有 © 2011, 2016, Oracle 和/或其附属公司。保留所有权利。

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

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

如果将本软件或相关文档交付给美国政府，或者交付给以美国政府名义获得许可证的任何机构，则适用以下注意事项：

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

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

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

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

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

目录

前言	11
文档可访问性	11
使用本文档的先决条件	11
约定	11
可用文档	12
1. 维护 Oracle HSM 解决方案	13
2. 监视文件系统操作	15
Oracle HSM Manager	15
samu	16
日志和跟踪文件	17
3. 管理 Oracle HSM 文件系统	19
管理 Oracle HSM 文件系统	19
管理文件系统配额	19
确定用户、组和组织单元的存储需求的特征	20
为由多个组使用的项目和目录创建管理集	23
配置新文件系统以使用配额	24
配置现有文件系统以使用配额	26
为组、项目、目录和用户设置配额	29
修复不一致的配额	31
检查配额	33
以文件系统管理员身份监视配额	34
监视您自己的用户配额	35
暂时延长或取消宽限期	36
将宽限期延长指定的时间量	36
重新启动宽限期	37
提前结束宽限期	39
停止新的资源分配	41
删除文件系统的配额	43
控制归档和回写操作	44
使归档和回写进程闲置	44

停止归档和回写进程	45
重新启动归档和回写进程	47
重命名文件系统	47
重命名文件系统	47
修复文件系统	50
修复文件系统	50
向文件系统添加设备	51
向已挂载的文件系统添加设备	51
完成对添加到共享文件系统的新设备的配置	55
向未挂载的文件系统添加设备	57
从文件系统中删除数据设备	60
确保对文件系统元数据和数据进行备份	60
运行 samexplorer	60
为文件系统创建恢复点文件	61
从已挂载的高性能文件系统中删除设备	62
管理 Oracle HSM 共享文件系统	66
挂载和卸载共享文件系统	66
挂载共享文件系统	66
卸载共享文件系统	68
更改共享文件系统的主机配置	69
配置额外的文件系统客户机	69
在共享文件系统配置中添加主机信息	69
在 Solaris 客户机上配置共享文件系统	71
在 Solaris 主机上挂载共享文件系统	75
在 Linux 客户机主机上配置共享文件系统	77
在 Linux 客户机主机上挂载共享文件系统	79
从共享文件系统配置中删除主机	81
从文件系统 Hosts 文件中删除主机	81
为分布式磁带 I/O 配置数据移动器客户机	84
配置数据移动器客户机	84
使用永久绑定连接磁带机	88
更新永久绑定以反映对硬件配置的更改	88
将新的文件系统主机永久绑定到可移除介质设备	91
从活动元数据服务器切换到潜在元数据服务器	95
激活潜在元数据服务器以替换有故障的活动元数据服务器	95
激活潜在元数据服务器以替换运行正常的活动元数据服务器	96
将非共享文件系统转换为共享文件系统	97
在活动元数据服务器和潜在元数据服务器上创建 Hosts 文件	97
共享非共享的文件系统并配置客户机	100

使用本地 Hosts 文件来路由网络通信	102
将共享文件系统转换为非共享文件系统	105
将共享元数据服务器转换为非共享系统	105
4. 管理文件和目录	109
设置 Oracle HSM 文件属性	109
恢复默认文件属性值	109
预分配文件系统空间	110
为文件指定循环分配或分散读写分配	111
在指定的分散读写组设备上分配文件存储	112
使用扩展文件属性	112
适应大型文件	113
管理特大型文件的磁盘高速缓存	113
文件分段	113
将文件分段	114
在多个卷上分散读写分段文件	114
将可移除介质文件用于大型数据集	116
创建可移除介质或卷溢出文件	116
读取外部磁带卷作为可移除介质文件	117
使用 Linear Tape File System (LTFS) 卷	118
将 LTFS 介质导入到磁带库	118
将 LTFS 目录和文件连接到 Oracle HSM 文件系统	118
使 LTFS 文件可按需访问	118
使 LTFS 文件在磁盘高速缓存中可立即访问	119
使用 Oracle HSM 软件访问 LTFS 介质	120
将 LTFS 卷装入磁带机并挂载 LTFS 文件系统	120
卸载 LTFS 文件系统，并从磁带机中卸载卷	121
使用 Oracle HSM 软件管理 LTFS 介质	121
将卷格式化为 LTFS 文件系统	121
清除 LTFS 数据，并从卷中删除格式和分区	122
检查 LTFS 文件系统的完整性	122
显示 LTFS 配置和状态信息	123
管理 SMB/CIFS 共享中的目录和文件	123
管理 SMB/CIFS 共享中的系统属性	123
Oracle HSM 支持的系统属性	123
显示系统属性	124
修改系统属性	124
管理访问控制列表	124

5. 管理库、介质和驱动器	127
管理自动化介质库	127
使库联机和脱机	127
使库脱机	127
使库联机	128
导入和导出可移除介质	129
导入可移除介质磁带	129
导出可移除介质磁带	129
维护库目录	130
查看库目录	131
审计库插槽的内容	132
审计整个直接连接的自动化库	133
从目录中清除介质错误	133
管理历史记录目录	135
查看历史记录目录	136
在历史记录目录中添加条目	136
从历史记录目录中删除条目	137
更新历史记录信息	137
确定驱动器在库中的安装顺序	137
为库和 Solaris 主机收集驱动器信息	138
将直接连接的库中的驱动器映射到 Solaris 设备名称	138
将 ACSLS 连接的库中的驱动器映射到 Solaris 设备名称	140
管理驱动器	142
装入和卸载驱动器	142
装入和卸载安装在自动化库中的驱动器	142
从指定的库位置装入驱动器	142
使用指定的介质类型和卷序列号装入驱动器	143
卸载库中的指定驱动器	143
手动装入和卸载独立驱动器	143
向独立驱动器中装入磁带	143
从独立驱动器中卸载磁带	143
必须手动装入卷时通知操作员	144
启用装入通知	144
清洗磁带机	145
提供足够的清洗磁带	146
启用自动化磁带机清洗（建议配置）	147
手动清洗磁带机	148
使用具备加密功能的驱动器	148

处理驱动器问题	148
使驱动器脱机以进行维护或修复	149
在解决驱动器问题后将介质返回到库中	149
将介质返回尚未执行自动审计的磁带库	149
自动审计后将介质返回磁带库	149
管理可移除介质	150
为可移除介质设置标签	150
基于条码生成标签	150
为新磁带设置标签或者为现有磁带重新设置标签	152
为新光盘设置标签或为现有光盘重新设置标签	154
维护数据完整性	155
显示数据完整性验证 (Data Integrity Validation, DIV) 设置和状态	155
显示 DIV 设置	155
监视归档文件的“写入后验证”状态	156
监视设备的“写入后验证”状态	156
检查给定磁带卷的完整性	156
验证通过磁带库位置指定的磁带上的数据	157
验证通过介质类型和卷序列号指定的磁带上的数据	157
验证使用指定驱动器的磁带上的数据	157
从磁带开头处重新开始数据验证	158
验证 T10000C/D 磁带上所有块的 ECC	158
验证 T10000C/D 磁带上所有块的 ECC 和 DIV 校验和	159
重新构建 T10000C/D 磁带的介质信息区域 (Media Information Region, MIR)	159
取消指定磁带的的数据验证	159
显示磁带的 DIV 状态和验证进度	160
监视自动化完整性验证	160
查看和验证 <code>verifyd.cmd</code> 配置文件	161
重新加载 <code>verifyd.cmd</code> 配置文件	161
显示所有定期介质验证磁带缺陷数据库中列出的缺陷	161
显示针对特定卷列出的缺陷	162
清除定期介质验证磁带缺陷数据库中列出的缺陷	162
6. 为数字保留管理归档	163
为保留配置文件系统	163
使用消息摘要 (校验和)	164
确保文件系统主机性能将足够高	165
为文件提供消息摘要并启用验证	166

为文件生成消息摘要并启用验证	168
为目录中的每个文件生成消息摘要并启用验证	170
在回写期间验证文件的消息摘要	172
在对文件归档之前更改消息摘要和验证属性	173
将文件设置为不可变	175
提供消息摘要并将文件设置为不可变	176
生成消息摘要并将文件设置为不可变	177
检查文件摘要和固定性属性	177
列出消息摘要和验证属性	177
使用 WORM 文件系统	178
了解 WORM 文件系统	179
为目录启用 WORM	180
为文件激活 WORM 保护	181
查找并列出的 WORM 文件	183
7. 备份配置和文件系统	185
备份文件系统	185
了解恢复点和归档日志	185
根据需要创建恢复点	186
备份归档程序日志	188
备份 Oracle HSM 配置	188
手动备份 Oracle HSM 配置	189
使用 samexplorer 收集配置和诊断信息	191
运行 samexplorer	191
8. 迁移到新存储介质	193
准备迁移	194
确保文件系统已备份	194
提供所需的介质	194
选择迁移方法	194
选择最符合您需求的迁移方法	194
迁移完整卷	195
创建 migrationd.cmd 文件	195
检查处于活动状态的迁移作业	202
迁移卷	204
回写文件并将其重新归档到替换介质中	208
估算可用资源	208
将归档过程配置为使用新介质	208

将数据迁移到替换介质	210
将数据从一个磁带迁移到另一个磁带	210
迁移之后处置旧介质	214
A. 设备类型词汇表	215
推荐的设备和介质类型	215
其他设备和介质类型	216
B. 介质状态标志	219
C. 共享文件系统中的挂载选项	221
共享文件系统挂载选项	221
bg : 在后台挂载	221
retry : 再次尝试文件系统挂载	221
shared : 声明 Oracle HSM 共享文件系统	221
minallopsz 和 maxallopsz : 调整分配大小	221
rdlease 、 wrlease 和 aplease : 在 Oracle HSM 共享文件系统中使用租约	222
mh_write : 启用多台主机读写	222
min_pool : 设置最小并发线程数	223
meta_timeo : 保留缓存属性	223
stripe : 指定分散读写分配	223
sync_meta : 指定元数据写入频率	224
worm_capable 和 def_retention : 启用 WORM 功能	224
D. 用于归档的配置指令	225
归档指令	225
全局归档指令	225
archivemeta : 控制是否对元数据进行归档	226
archmax : 控制归档文件的大小	226
bufsize : 设置归档程序缓冲区大小	226
drives : 控制用于归档的驱动器数	227
examine : 控制归档扫描	227
interval : 指定归档时间间隔	228
logfile : 指定归档程序日志文件	228
notify : 重命名事件通知脚本	229
ovflmin : 控制卷溢出	229
scanlist_squash : 控制扫描列表合并	229

setarchdone : 控制 archdone 标志的设置	230
wait : 延迟归档程序启动	230
文件系统指令	230
fs : 指定文件系统	230
copy-number [archive-age]: 指定文件系统元数据的多个副本	231
interval 、 logfile 和 scanlist 作为文件系统指令	231
archive-set-name : 归档集分配指令	231
归档副本指令	232
副本参数	233
卷序列号 (Volume Serial Number, VSN) 池指令	237
卷序列号 (Volume Serial Number, VSN) 关联指令	238
回写指令	238
stager.cmd 文件	239
drives : 指定进行回写的驱动器数量	239
bufsize : 设置回写缓冲区大小	240
logfile : 指定回写日志文件	240
maxactive : 指定回写请求的数量	241
copysel : 指定回写期间的副本选择顺序。	242
预览请求指令	242
全局指令	242
vs_n_priority : 调整卷优先级	243
age_priority : 针对在队列中的等待时间调整优先级	243
全局和/或特定于文件系统的指令	243
hwm_priority : 在磁盘高速缓存几乎填满时调整优先级	244
lwm_priority : 在磁盘高速缓存几乎为空时调整优先级	244
lhwm_priority : 根据磁盘高速缓存的填充情况调整优先级	244
hlwm_priority : 根据磁盘高速缓存为空的程度调整优先级	244
preview.cmd 文件样例	245
E. 产品辅助功能	247
词汇表	249
索引	259

前言

本文档用于满足执行以下任务的系统管理员、存储和网络管理员以及服务工程师的需求：使用 Oracle Hierarchical Storage Manager and Oracle StorageTek QFS Software（Oracle Hierarchical Storage Manager 以前称为 StorageTek Storage Archive Manager）监视、管理和维护文件系统和归档解决方案。

文档可访问性

有关 Oracle 对可访问性的承诺，请访问 Oracle Accessibility Program 网站：<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>。

获得 Oracle 支持

购买了支持服务的 Oracle 客户可通过 My Oracle Support 获得电子支持。有关信息，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>；如果您听力受损，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>。

使用本文档的先决条件

本文档假定您已熟悉 Oracle Solaris 操作系统、磁盘和磁带存储系统以及本地和存储区域网络的管理。有关相关任务、命令和过程的信息，请参阅 Solaris 文档和手册页以及存储硬件文档。

约定

本文档中使用了以下文本约定：

- 斜体类型表示书籍标题和强调。
- 等宽字体类型表示终端窗口中显示的命令和文本，以及配置文件、shell 脚本和源代码文件的内容。
- 等宽粗体类型表示用户输入和对命令行输出、终端显示或文件内容的显著更改。它还可用于强调某个文件或显示中特别重要的部分。
- 等宽粗体倾斜类型表示终端显示或文件中的变量输入和输出。
- 等宽倾斜类型表示终端显示或文件中的其他变量。
- ...（三点省略号）表示与示例无关、因而为简短或清晰起见已忽略的文件内容或命令输出。
- /（反斜杠）位于示例中某一行的末尾，用于将换行符转义，以便下一行是同一命令的一部分。
- [-]（方括号括住由连字符分隔的值）用于限定值的范围。
- []（方括号）位于命令语法描述中，用于指示可选参数。

- `root@solaris:~#` 和 `[hostname]:root@solaris:~#` 表示 Solaris 命令 shell 提示符。
- `[root@linux ~]#` 表示 Linux 命令 shell 提示符。

可用文档

《Oracle Hierarchical Storage Manager and StorageTek QFS Software 维护和管理指南》是多卷 Oracle HSM 客户文档库的一部分，可以从 <http://docs.oracle.com/en/storage/#sw> 获取。

可从 <http://docs.oracle.com/en/operating-systems/> 获取 Oracle Solaris 操作系统文档。

有关系统要求、新功能和错误修复的信息，请查询发行说明 `README.txt`，位于下载 ZIP 文件中，或文件系统服务器上的 `/opt/SUNWsamfs/doc/README.txt` 中。

第 1 章 维护 Oracle HSM 解决方案

在规划和部署 QFS 文件系统和 Oracle Hierarchical Storage Manager 软件过程中，始终贯彻了以下设计理念：将性能优化、数据保护和归档的复杂特性隐藏在简单易用的 UNIX 文件系统界面下。用户、应用程序以及管理员（大多数情况下为管理员）应该能够方便地使用在磁盘阵列和磁带库混合环境中实施的经过充分优化的 Oracle HSM 归档系统，就像它是在单一本地磁盘上实施的普通 UFS 文件系统一样。安装和配置 Oracle HSM 软件后，该软件自动以尽可能最高效、最可靠的方式管理数据和存储资源，所需的人为干预极少。

也就是说，与任何 UNIX 文件系统一样，仍然需要执行监视任务和定期管理任务。本书将概括介绍这些活动。

第 2 章 监视文件系统操作

正确配置的 Oracle HSM 文件系统只需要很少量的例行管理操作。但您需要监视每个系统以了解异常。通常需要监视两方面：可用性和使用率。

可用性概念明了，易于监视。主机系统、网络接口、文件系统或存储子系统等主要组件不可用时，核心功能会突然丧失或降级，管理界面和日志中会显示警报。

使用率问题较不易觉察，更需要自身判断。使用过度时，例如缺乏存储介质等资源，会像组件出现故障一样停止归档进程，即使系统在正常运行也是如此。同样的使用率级别在一个组织中可能预示即将出现问题，而在另一个组织中可能会多年都无问题地运行。因此监视使用率时，识别趋势和等级至关重要。对于使用了 80% 的资源，如果使用率每年增长 1%，这没有问题，但如果每周增长 1%，这就是非常重大的问题。

Oracle HSM 提供三个监视界面：

- [Oracle HSM Manager](#)
- [samu](#)
- [日志和跟踪文件](#)

每个界面各有特点，取决于您的工作方式和习惯。

Oracle HSM Manager

Oracle HSM Manager 是基于浏览器的图形用户界面，允许管理员监视和控制文件系统运行的所有方面。Oracle HSM Manager 浏览器界面页分为三个部分：

- 标题
- 导航树
- 内容窗格

标题显示应用程序的名称和状态信息，例如在浏览器界面中上次更新数据的时间，当前登录用户的名称和角色，托管 Oracle HSM Manager 软件的管理站的名称以及未确认故障的当前数量和类型。

位于界面左侧的导航树包含服务器菜单和可用显示的分层列表。单击导航树中的链接后相应的显示即会出现在内容窗格中。

导航树的 "Monitoring"（监视）节点是您的主要监视资源。您可以列出在受监视文件系统和设备上检测到的所有故障并进行排序。您可以配置自动电子邮件警报。您

可以列出当前运行的所有作业。"Monitoring" (监视) 节点还包含指向 "Monitoring Dashboard" (监视系统信息显示板) 的链接, 这是一个弹出窗口, 用于提示您注意问题, 并提供以下关注点的汇总信息的快速链接:

- 守护进程
- 文件系统
- 归档介质使用率
- 磁带库
- 磁带库驱动器
- 卷暂挂装入请求
- 不可用的卷
- 归档副本队列
- 回写队列

导航树的 "Metrics and Reports" (度和报告和) 节点提供全面的状态和使用率报告以及指向 "System Details" (系统详细信息) 的链接。在 "System Details" (系统详细信息) 页面可以快速查看 Oracle HSM 配置, 并可以快捷地访问日志和跟踪文件。

还有一个综合的帮助系统全面记录了 Oracle HSM Manager 的使用方法。

samu

samu 操作实用程序是基于文本的菜单驱动型配置和管理界面, 可以从命令行启动该界面。该界面是一个便捷工具, 使用它可以方便地监视 Oracle HSM 设备、文件系统活动及错误消息。

samu 实用程序在某些方面类似于 UNIX *vi* 编辑器。您可以使用类似的控制键序选择显示、设置显示选项、在显示内部和之间导航、输入命令、刷新显示以及退出实用程序。每个显示窗口的最后一行显示错误消息。显示会自动刷新, 除非出现错误, 此时显示会停止, 直至操作人员采取进一步的操作。如果需要, 可以提取显示窗口的快照供以后参考。

h 命令用于打开帮助屏幕, 其中列出所有键盘快捷键、命令及参数。有关其他信息, 您还可以参阅 *samu* 手册页和 Oracle HSM 客户文档库 (<http://docs.oracle.com/en/storage/#sw>) 中的《Oracle Hierarchical Storage Manager and StorageTek QFS Software *samu* 命令参考》。

下面是一个典型的 *samu* 监视显示:

```

Archiver status          samu          5.4          12:24:10 Mar 19 2014

sam-archiverd: Waiting for resources

sam-arfind:  samma1 mounted at /samma1
Files waiting to start 0   schedule 70,524   archiving 0
Monitoring file system activity.
```



```
sam-arfind: DISKVOL1 mounted at /diskvols/DISKVOL1
Files waiting to start 0  schedule 0  archiving 0
Monitoring file system activity.
```

```
samu on samqfshost1
```

日志和跟踪文件

Oracle HSM 软件可执行全面的日志记录，如果配置，还可执行全面的跟踪。因此您可能希望监视以下文件，尤其是当出现问题时：

- `/var/adm/messages`
- `/var/adm/sam-log`
- `/var/opt/SUNWsamfs/trace/`（守护进程和进程的跟踪文件）
- `/var/opt/SUNWsamfs/devlog/`（在 `/etc/opt/SUNWsamfs/mcf` 文件中配置的设备日志）
- `/var/opt/SUNWsamfs/archiver.log`
- `/var/opt/SUNWsamfs/stager.log`
- `/var/opt/SUNWsamfs/recycler.log`
- 特定于文件系统的其他归档日志（如果已配置）。

有关配置日志记录和跟踪的信息，请参见 *Oracle HSM 客户文档库* (<http://docs.oracle.com/en/storage/#sw>) 中的《Oracle Hierarchical Storage Manager and StorageTek QFS Software 安装和配置指南》。

第 3 章 管理 Oracle HSM 文件系统

本章介绍文件系统维护和重新配置任务。第一节“[管理 Oracle HSM 文件系统](#)”介绍所有 Oracle HSM 文件系统的维护，包括归档和非归档文件系统、共享和非共享（独立）文件系统。第二节“[管理 Oracle HSM 共享文件系统](#)”介绍影响共享文件系统的特殊注意事项。

管理 Oracle HSM 文件系统

本节概述以下任务：

- [管理文件系统配额](#)
- [控制归档和回写操作](#)
- [重命名文件系统](#)
- [修复文件系统](#)
- [向文件系统添加设备](#)
- [从文件系统中删除数据设备](#)

管理文件系统配额

设置文件系统配额以控制给定用户或用户集合在文件系统中可以占用的在线存储空间和总存储空间。可以按用户 ID、组 ID 或管理员定义的根据公共特征（如参与特定的项目）对用户分组的管理集 ID 设置配额。当项目包含来自多个组的用户并跨多个目录和文件时，管理集 ID 尤其有用。

可以通过使用 *quota* 挂载选项（默认设置）挂载文件系统来启用配额，通过使用 *noquota* 挂载选项挂载文件系统来禁用配额。通过在文件系统根目录中放置一个或多个配额文件可定义配额：*.quota_u*、*.quota_g* 和 *.quota_a*，分别为用户、组和管理集设置配额。每个文件中的第一个记录（即记录 0）设置默认值。后续记录设置特定于特定用户、组或管理集的值。

配额分配可用的文件系统空间，而不仅仅是存储空间。因此，配额同时针对在介质上分配的 512 字节块的数目和在文件系统中分配的 inode 的数目设置上限。块计数本质上度量存储空间。inode 计数度量可用于访问该存储的资源。因此，使用多个存储空间块而只有一个 inode 的单个文件所占用的文件系统空间量可能会与占用多个 inode 但没有块的多个长度为零的空文件所占用的文件系统空间量相同。

每个配额可以包括软限制和硬限制。硬限制定义给定所有者的所有文件可以暂时使用的文件系统资源上限。软限制定义所有者的文件可以无限期使用的文件系统资源上

限。资源使用量只能在短暂的间隔内增长到软限制与硬限制之间的数量，该间隔由配额中的宽限期定义。

本节介绍以下管理任务：

- 确定用户、组和组织单元的存储需求的特征
- 为由多个组使用的项目和目录创建管理集
- 配置新文件系统以使用配额
- 配置现有文件系统以使用配额
- 为组、项目、目录和用户设置配额
- 修复不一致的配额
- 检查配额
- 暂时延长或取消宽限期
- 停止新的资源分配。

确定用户、组和组织单元的存储需求的特征

要设置可持续的配额，必须以可管理和可伸缩的方式设置适应用户要求的限制。因此，在设置配额之前，请评估用户的存储需求。要使流程可管理，请首先尽可能宽泛地对用户需求进行分类，以便通过最少的管理工作满足尽可能多的需求。然后，您可以专门评估未包括在较宽泛类别中的少量用户需求。结果将提供关于要设置的配额和限制类型的粗略大纲。

下面概述的方法首先确定访问控制组的文件系统需求，因为大多数组织已定义了这些组。然后，定义需求与这些标准组不一致的特殊用户集。然后，这时才开始解决单个用户的独特需求。执行如下操作：

1. 由于您的现有访问控制组已经将具有类似资源需求的用户归在一起，因此请首先定义将使用文件系统的任何组的平均存储需求。评估所使用的平均存储空间量（以 512 千字节块为单位）和所存储的平均文件数，这等效于所使用的平均 inode 数。

由于组成员通常具有类似的组织角色和工作职责，因此他们经常需要访问相同的目录和文件并且通常对存储具有类似的需求。在示例中，确定了将使用文件系统 / *hsm/hsmfs1* 的三个组：*dev*（产品开发）、*cit*（公司信息技术）和 *pgmt*（程序管理）。随后在一个简单的电子表格中列出这些组、每个组的成员数及其个体和组的平均需求。

组	用户	每用户平均块数	每用户平均文件数	平均块数/组	平均文件数/组
dev	30	67108864	500	2013265920	15000
cit	15	10485760	50	157286400	750
pgmt	6	20971520	200	125829120	1200
总块数/ 文件数（平均）					

2. 接下来，针对任意给定时间的最大存储空间量和组成员将存储的最大文件数执行相同的计算。记录结果。

在示例中，将结果记录在一个新的电子表格中：

组	用户	每用户最大块数	每用户最大文件数	最大块数/组	最大文件数/组
dev	30	100663296	1000	3019898880	30000
cit	15	15728640	100	235929600	1500
pmgt	6	31457280	400	188743680	2400
总块数/ 文件数 (最大)					

3. 现在，确定属于不同的组但具有相同独特存储需求（此需求无法基于组来解决）的用户集。为所确定的每个组织执行与每个访问控制组相同的评估并执行相同的计算。

在示例中，确定了将需要存储分配的两个公司项目，代码为 *portal* 和 *lockbox*。工程、市场营销、合规、测试和文档组的成员将一起处理这些项目并使用相同的目录和许多相同的文件。因此，将其添加到需求电子表格中：

组	用户	每用户平均块数	每用户平均文件数	平均块数/组	平均文件数/组
dev	30	67108864	500	2013265920	15000
cit	15	10485760	50	157286400	750
pmgt	6	20971520	200	125829120	1200
portal	10	31457280	400	314572800	4000
lockbox	12	31457280	500	377487360	6000
总块数/ 文件数 (平均)					

组	用户	每用户最大块数	每用户最大文件数	最大块数/组	最大文件数/组
dev	30	100663296	1000	3019898880	30000
cit	15	15728640	100	235929600	1500
pmgt	6	31457280	400	188743680	2400
portal	10	37748736	700	377487360	7000
lockbox	12	45613056	600	547356672	7200
总块数/ 文件数 (最大)					

4. 现在，确定需求尚未解决的单个用户。为每个用户执行与每个访问控制组和非组组织相同的评估并执行相同的计算。

只要可能，就应将用户需求放在一起解决，以便统一策略并使管理开销降至最低。不过，当个别需求比较独特时，您需要单独解决它们。在示例中，确定了 *pmgt* 组

中的 *jr23547* 是具有特殊职责的用户，需要特殊的存储分配。因此，将他添加需求电子表格中：

组	每个集的用户	每用户平均块数	每用户平均文件数	平均块数	平均文件数
dev	30	67108864	500	2013265920	15000
cit	15	10485760	50	157286400	750
pmgt	6	20971520	200	125829120	1200
portal	10	31457280	400	314572800	4000
lockbox	12	31457280	500	377487360	6000
jr23547	1	10485760	600	10485760	600
总块数/ 文件数 (平均)					

组	用户	每用户最大块数	每用户最大文件数	最大块数/组	最大文件数/组
dev	30	100663296	1000	3019898880	30000
cit	15	15728640	100	235929600	1500
pmgt	6	31457280	400	188743680	2400
portal	10	37748736	700	377487360	7000
lockbox	12	45613056	600	547356672	7200
jr23547	1	100663296	2000	100663296	2000
总块数/ 文件数 (最大)					

5. 最后，计算所有用户需要的平均和最大块数和文件数。

组	用户	每用户平均块数	每用户平均文件数	平均块数/组	平均文件数/组
dev	30	67108864	500	2013265920	15000
cit	15	10485760	50	157286400	750
pmgt	6	20971520	200	125829120	1200
portal	10	31457280	400	314572800	4000
lockbox	12	31457280	500	377487360	6000
jr23547	1	10485760	600	10485760	600
总块数/ 文件数 (平均)				2998927360	27550

组	用户	每用户最大块数	每用户最大文件数	最大块数/组	最大文件数/组
dev	30	100663296	1000	3019898880	30000
cit	15	15728640	100	235929600	1500

组	用户	每用户最大块数	每用户最大文件数	最大块数/组	最大文件数/组
pmgt	6	31457280	400	188743680	2400
portal	10	37748736	700	377487360	7000
lockbox	12	45613056	600	547356672	7200
jr23547	1	100663296	2000	100663296	2000
总块数/文件数 (平均)				4470079488	50100

- 如果您需要管理基于项目的配额或者其他无法按访问控制组 and 用户 ID 定义的配额，请为多个组使用的项目和目录创建管理集。
- 如果您要在空的新创建的文件系统上设置配额，请转至[“配置新文件系统以使用配额”](#)。
- 如果您要在已包含文件的文件系统上设置配额，请转至[“配置现有文件系统以使用配额”](#)。

为由多个组使用的项目和目录创建管理集

管理集是用于配额的一个目录层次结构或单个目录或文件，由管理集 ID 予以标识。以指定管理集 ID 创建的或者存储在具有指定管理集 ID 的目录中的所有文件都具有相同的配额，无论实际拥有这些文件的用户或组 ID 是什么。要定义管理集，请执行如下操作：

- 以 *root* 用户身份登录到文件系统服务器。

在示例中，服务器名为 *server1*：

```
[server1]root@solaris:~#
```

- 如果要使用管理集为新项目或团队配置存储配额，请在文件系统内的某个位置为该项目或团队创建一个新目录。

在示例中，在 */hsm/hsmfs1* 文件系统中创建该目录并将其命名为 *portalproject/*，与项目同名。

```
[server1]root@solaris:~# mkdir /hsm/hsmfs1/portalproject
```

- 为您需要设置配额的目录或文件分配管理集 ID。使用命令 *samchaid [-fhr] admin-set-id directory-or-file-name*，其中：
 - f* 强制执行分配并且不报告错误。
 - h* 将管理集 ID 分配给符号链接。不使用此选项时，将更改符号链接引用的文件的组。
 - R* 以递归方式将管理集 ID 分配给子目录和文件。
 - admin-set-id* 是一个具有唯一性的整数值。
 - directory-or-file-name* 是您要向其分配管理集 ID 的目录或文件的名称。

在示例中，将管理集 ID 1 分配给目录 `/hsm/hsmfs1/portalproject/` 及其所有子目录和文件。

```
[server1]root@solaris:~# samchaid -R 1 /hsm/hsmfs1/portalproject/
```

4. 如果需要，您可以检查分配。使用命令 `command sls -D directory-path`，其中 `-D` 指定为 `directory-path` 中的文件和目录显示详细的 Oracle HSM 目录列表：

```
[server1]root@solaris:~# sls -D /hsm/hsmfs1/
/portalproject:
mode: drwxr-xr-x   links: 2   owner: root       group: root
length: 4096   admin id: 1   inode: 1047.1
project: user.root(1)
access: Feb 24 12:49   modification: Feb 24 12:44
changed: Feb 24 12:49   attributes: Feb 24 12:44
creation: Feb 24 12:44   residence: Feb 24 12:44
```

5. 如果您要在空的新创建的文件系统上设置配额，请转至["配置新文件系统以使用配额"](#)。
6. 如果您要在已包含文件的文件系统上设置配额，请转至["配置现有文件系统以使用配额"](#)。

配置新文件系统以使用配额

如果您正在创建新文件系统，并且该文件系统中当前没有任何文件，则可以执行此过程。

1. 以 `root` 用户身份登录到文件系统服务器。

在示例中，服务器名为 `server2`：

```
[server2]root@solaris:~#
```

2. 如果新文件系统当前未挂载，在继续操作前请先挂载它。
3. 如果必须为组设置配额，请在文件系统根目录中创建一个组配额文件 `.quota_g`。使用 Solaris 命令 `dd if=/dev/zero of=mountpoint/.quota_g bs=4096 count=number-blocks`，其中：
 - `if=/dev/zero` 指定将 UNIX 特殊文件 `/dev/zero` 中的空字符作为输入。
 - `of=mountpoint/.quota_g` 指定输出文件，其中 `mountpoint` 是文件系统的挂载点目录。
 - `bs=4096` 将用于写入的块大小设置为 4096 字节。
 - `count=number-blocks` 指定要写入的块数。该值取决于文件将存放的记录数。对于每个指定的配额都有一条 128 字节的记录，因此一个块可以包含 32 条记录。

在示例中，为挂载在 `/newsamfs` 的文件系统 `newsamfs` 创建组配额文件。在需求收集阶段中，确定了文件系统中需要配额的三个组：`dev`、`cit` 和 `pgmt`。由于不打算添加任何其他组配额，因此设置每个块上的文件大小：

```
[server2]root@solaris:~# dd if=/dev/zero of=/newsamfs/.quota_g bs=4096 count=1
```

4. 如果必须为管理集设置配额，请在文件系统根目录中创建一个管理集配额文件 `.quota_a`。使用 Solaris 命令 `dd if=/dev/zero of=mountpoint/.quota_a bs=4096`，其中：

- `mountpoint` 是文件系统的挂载点目录。
- `.quota_a` 是输出文件的名称。
- `4096` 是用于写入的块大小（以字节为单位）。
- `number-blocks` 是要写入的块数。

在示例中，为挂载在 `/newsamfs` 的文件系统 `newsamfs` 创建管理集配额文件。在需求收集阶段中，确定了文件系统中需要配额的两个项目：`portal`（管理集 ID 1）和 `lockbox`（管理集 ID 2）。由于不打算添加任何其他管理集配额，因此设置每个块上的文件大小：

```
[server2]root@solaris:~# dd if=/dev/zero of=/newsamfs/.quota_a bs=4096 count=1
```

5. 如果必须为用户设置配额，请在文件系统根目录中创建一个用户配额文件 `.quota_u`。使用 Solaris 命令 `dd if=/dev/zero of=mountpoint/.quota_u bs=4096 count=number-blocks`，其中：

- `mountpoint` 是文件系统的挂载点目录。
- `.quota_u` 是输出文件的名称。
- `4096` 是用于写入的块大小（以字节为单位）。
- `number-blocks` 是要写入的块数。

在示例中，为挂载在 `/newsamfs` 的文件系统 `newsamfs` 创建用户配额文件。在需求收集阶段中，确定了文件系统中需要特定配额的一个用户 `jr23547`。由于不打算添加任何其他单个用户配额，因此设置每个块上的文件大小：

```
[server2]root@solaris:~# dd if=/dev/zero of=/newsamfs/.quota_u bs=4096 count=1
```

6. 卸载文件系统。

您必须先卸载文件系统，然后重新挂载它才能启用配额文件。

```
[server2]root@solaris:~# umount /newsamfs
```

7. 执行文件系统检查。

```
[server2]root@solaris:~# samfsck -F newsamfs
```

8. 重新挂载文件系统。

当系统检测到在文件系统的根目录中有一个或多个配额文件时，会启用配额。

您不需要在 `/etc/vfstab` 或 `samfs.cmd` 文件中包括 `quota` 挂载选项，因为默认情况下挂载文件系统时会启用配额。

```
[server2]root@solaris:~# mount /newsamfs
```

9. 接下来，根据需要设置或更新配额。请参见“为组、项目、目录和用户设置配额”。

配置现有文件系统以使用配额

如果您要为已包含文件的文件系统创建配额，请使用此过程。

1. 以 `root` 用户身份登录到文件系统服务器。

在示例中，服务器名为 `server1`：

```
[server1]root@solaris:~#
```

2. 在文本编辑器中打开 `/etc/vfstab` 文件，并确保未设置 `noquota` 挂载选项。

在示例中，在 `vi` 文本编辑器中打开文件。`noquota` 挂载选项已设置：

```
[server1]root@solaris:~# vi /etc/vfstab
#File
#Device      Device  Mount          System  fsck  Mount  Mount
#to Mount    to fsck Point          Type    Pass  at Boot Options
#-----
/devices     -      /devices      devfs   -     no     -
/proc       -      /proc         proc    -     no     -
...
hsmfs1      -      /hsm/hsmfs1   samfs   -     no     noquota
```

3. 如果已在 `/etc/vfstab` 文件中设置了 `noquota` 挂载选项，请删除该选项并保存文件。

```
[server1]root@solaris:~# vi /etc/vfstab
#File
#Device      Device  Mount          System  fsck  Mount  Mount
#to Mount    to fsck Point          Type    Pass  at Boot Options
#-----
```

```

/devices      -      /devices      devfs  -      no      -
/proc         -      /proc         proc   -      no      -
...
hsmfs1       -      /hsm/hsmfs1   samfs  -      no      -
:wq
[server1]root@solaris:~#

```

4. 在文本编辑器中打开 `/etc/opt/SUNWsamfs/samfs.cmd` 文件，并确保未设置 `noquota` 挂载选项。

在示例中，在 `vi` 文本编辑器中打开文件。`noquota` 挂载选项未设置：

```

[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/samfs.cmd
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
#
#inodes = 0
#fs = hsmfs1
# forcedirectio (default no forcedirectio)
# high = 80
# low = 70
# weight_size = 1.
# weight_age = 1.
# readahead = 128
...
# dio_wr_ill_min = 0
# dio_wr_consec = 3
# qwrite      (ma filesystem, default no qwrite)
# shared_writer (ma filesystem, default no shared_writer)
# shared_reader (ma filesystem, default no shared_reader)

```

5. 如果已在 `/etc/opt/SUNWsamfs/samfs.cmd` 文件中设置了 `noquota` 挂载选项，请删除该选项并保存文件。
6. 如果您从 `/etc/vfstab` 文件和/或 `/etc/opt/SUNWsamfs/samfs.cmd` 文件中删除了 `noquota` 挂载选项，请卸载文件系统。

删除 `noquota` 挂载选项后，必须卸载文件系统，以便您可以在重新挂载后启用配额。

```

[server1]root@solaris:~# umount /hsm/hsmfs1

```

7. 如果文件系统当前未挂载，现在请挂载它。

必须先挂载文件系统，然后才能启用配额。

```
[server1]root@solaris:~# mount /hsm/hsmfs1
```

8. 转到文件系统的根目录并检查是否存在任何配额文件。使用 Solaris 命令 `ls -a` 并查找 `.quota_g`、`.quota_a` 和/或 `.quota_u` 文件。

在示例中，当前不存在配额文件。

```
[server1]root@solaris:~# cd /hsm/hsmfs1
[server1]root@solaris:~# ls -a /hsm/hsmfs1
.                .archive        .fuid           .stage         portalproject
..               .domain         .inodes        lost+found
```

9. 如果存在配额文件，不要修改它们。
10. 如果必须为组设置配额并且文件系统根目录中尚未存在组配额文件 `.quota_g`，现在请创建该文件。使用 Solaris 命令 `dd if=/dev/zero of=mountpoint/.quota_g bs=4096 count=number-blocks`，其中：
 - `if=/dev/zero` 指定将 UNIX 特殊文件 `/dev/zero` 中的空字符作为输入。
 - `of=mountpoint/.quota_g` 指定输出文件，其中 `mountpoint` 是文件系统的挂载点目录。
 - `bs=4096` 将用于写入的块大小设置为 4096 字节。
 - `count=number-blocks` 指定要写入的块数。该值取决于文件将存放的记录数。对于每个指定的配额都有一条 128 字节的记录，因此一个块可以包含 32 条记录。

在示例中，为在 `/hsm/hsmfs1` 处挂载的文件系统 `/hsm/hsmfs1` 创建组配额文件。在需求收集阶段中，确定了文件系统中需要配额的三个组：`dev`、`cit` 和 `pgmt`。由于不打算添加任何其他组配额，因此设置每个块上的文件大小：

```
[server1]root@solaris:~# dd if=/dev/zero of=/hsm/hsmfs1/.quota_g bs=4096 count=1
```

11. 如果必须为管理集设置配额并且文件系统根目录中尚未存在管理集配额文件 `.quota_a`，现在请创建该文件。使用 Solaris 命令 `dd if=/dev/zero of=mountpoint/.quota_a bs=4096 count=number-blocks`，其中：
 - `mountpoint` 是文件系统的挂载点目录。
 - `.quota_a` 是输出文件的名称。
 - `4096` 是用于写入的块大小（以字节为单位）。
 - `number-blocks` 是要写入的块数。

在示例中，为在 `/hsm/hsmfs1` 处挂载的文件系统 `/hsm/hsmfs1` 创建管理集配额文件。在需求收集阶段中，确定了文件系统中需要配额的两个项目：`portal`（管理集 ID 1）和 `lockbox`（管理集 ID 2）。由于不打算添加任何其他管理集配额，因此设置每个块上的文件大小：

```
[server1]root@solaris:~# dd if=/dev/zero of=/hsm/hsmfs1/.quota_a bs=4096 count=1
```

12. 如果必须为用户设置配额并且文件系统根目录中尚未存在用户配额文件 `.quota_u`，现在请创建该文件。使用 Solaris 命令 `dd if=/dev/zero of=mountpoint/.quota_u bs=4096 count=number-blocks`，其中：

- `mountpoint` 是文件系统的挂载点目录。
- `.quota_u` 是输出文件的名称。
- `4096` 是用于写入的块大小（以字节为单位）。
- `number-blocks` 是要写入的块数。

在示例中，为在 `/hsm/hsmfs1` 处挂载的文件系统 `/hsm/hsmfs1` 创建用户配额文件。在需求收集阶段中，确定了文件系统中需要特定配额的一个用户 `jr23547`。由于不打算添加任何其他单个用户配额，因此设置每个块上的文件大小：

```
[server1]root@solaris:~# dd if=/dev/zero of=/hsm/hsmfs1/.quota_u bs=4096 count=1
```

13. 卸载文件系统。

您必须先卸载文件系统，然后重新挂载它才能启用配额文件。

```
[server1]root@solaris:~# umount /hsm/hsmfs1
```

14. 执行文件系统检查。

```
[server1]root@solaris:~# samfsck -F /hsm/hsmfs1
```

15. 重新挂载文件系统。

当系统检测到在文件系统的根目录中有一个或多个配额文件时，会启用配额。

您不需要在 `/etc/vfstab` 或 `samfs.cmd` 文件中包括 `quota` 挂载选项，因为默认情况下挂载文件系统时会启用配额。

```
[server1]root@solaris:~# mount /hsm/hsmfs1
```

16. 接着，为组、项目、目录和用户设置配额。

为组、项目、目录和用户设置配额

可以使用 `samquota` 命令设置新配额和调整现有配额。请遵循以下过程：

1. 在确定存储需求的特征后，为每个组、用户和非组组织确定合适的配额。考虑以下因素并根据需要进行调整：
 - 文件系统大小与所有用户需要的平均和最大块数之比

- 文件系统中 inode 的数目与所有用户需要的平均和最大 inode 数之比
 - 在任意给定时间可能接近其最大需求的用户的数目和类型。
2. 以 *root* 用户身份登录到文件系统服务器。

在示例中，服务器名为 *server1*：

```
[server1]root@solaris:~#
```

3. 为需要限制的每个组设置限制。使用命令 *samquota -b number-blocks:type[:scope] -f number-files:type[:scope] -t interval[:scope] -G groupID [directory-or-file]*，其中：
 - *-b number-blocks* 将可以在文件系统中存储的 512 千字节块的最大数目设置为 *number-blocks*（一个整数）（有关用来指定大小的其他方法，请参见 *samquota* 手册页）。值为 0（零）指定块数不受限。
 - *:* 是字段分隔符。
 - *type* 指定限制类型，*h* 表示硬限制，*s* 表示软限制。
 - *scope*（可选）标识受限制管制的存储类型。该参数可以为 *o*（表示仅在线（磁盘高速缓存）存储）或 *t*（表示总存储，包含磁盘高速缓存和归档存储（默认范围））。
 - *-f number-files* 将文件系统中可存储的最大文件数设置为 *number-files*（一个整数）。值为 0（零）指定文件数不受限。
 - *-t number-seconds* 将宽限期（可以超出软限制的时间长度）设置为 *number-seconds*，这是一个表示秒数的整数（有关用来指定时间的其他方法，请参见 *samquota* 手册页）。
 - *-G groupID* 指定组名称或组的整数标识符。值为 0（零）将为所有组设置默认限制。
 - *directory-or-file*（可选）是您需要设置配额的特定文件系统或特定目录或文件的挂载点目录。

在示例中，使用需求收集阶段的估算结果对组 *dev* 可以使用的 */hsm/hsmfs1* 文件系统中的存储空间量以及可以存储的文件数量设置硬限制和软限制。我们仅针对在线存储将宽限期设置为 43200 秒（十二小时）（请注意，下面的命令是作为单行输入的一使用反斜杠字符对换行符进行转义）：

```
[server1]root@solaris:~# samquota -b 3019898880:h:t -f 30000:h:t /
-G dev /hsm/hsmfs1
[server1]root@solaris:~# samquota -b 2013265920:s:t -f 15000:s:t -t 43200:o /
-G dev /hsm/hsmfs1
[server1]root@solaris:~#
```

4. 为需要限制的每个管理集设置限制。使用命令 *samquota -b number-blocks:type[:scope] -f number-files:type[:scope] -t interval[:scope]*

`-A adminsetID [directory-or-file]`，其中 `-A adminsetID` 是唯一标识管理集的整数值。

将 `adminsetID` 设置为 0（零）将为所有管理集设置默认限制。

在示例中，使用需求收集阶段的评估对 `/hsm/hsmfs1` 文件系统中 `portal` 项目（管理集 ID 1）可以使用的存储空间量以及可以存储的文件数量设置硬限制和软限制。我们针对使用的总存储（这是默认范围）将宽限期设置为 43200 秒（十二小时）（请注意，下面的命令是作为单行输入的一使用反斜杠字符对换行符进行转义）：

```
[server1]root@solaris:~# samquota -b 377487360:h:t -f 7000:h:t -A 1 /hsm/hsmfs1
[server1]root@solaris:~# samquota -b 314572800:s:t -f 4000:s:t -t 43200 /
-A 1 /hsm/hsmfs1
[server1]root@solaris:~#
```

5. 为需要限制的每个单个用户设置限制。使用命令 `samquota -b number-blocks:type[:scope] -f number-files:type[:scope] -t interval[:scope] -U userID [directory-or-file]`，其中 `-U userID` 是用户名或用户的整数标识符。

将 `userID` 设置为 0（零）将为所有用户设置默认限制。

在示例中，使用需求收集阶段的估算结果对用户 `jr23547` 可以使用的 `/hsm/hsmfs1` 文件系统中的存储空间量以及 `jr23547` 可以存储的文件数量设置硬限制和软限制。我们针对使用的总存储（这是默认范围）将宽限期设置为 1209600 秒（两周）（请注意，下面的命令是作为单行输入的一使用反斜杠字符对换行符进行转义）：

```
[server1]root@solaris:~# samquota -b 100663296:h:t -f 600:h:t /
-U jr23547 /hsm/hsmfs1
[server1]root@solaris:~# samquota -b 10485760:s:t -f 2000:s:t -t 1209600 /
-U jr23547 /hsm/hsmfs1
[server1]root@solaris:~#
```

6. 在此处停止。

修复不一致的配额

当根目录中存在配额文件时，如果使用 `noquota` 挂载选项挂载 Oracle HSM 文件系统，则当分配或释放了块或文件时，配额记录将变得不一致。在这种情况下，请执行如下操作：

1. 以 `root` 用户身份登录到文件系统服务器。

在示例中，服务器名为 `server1`：

```
[server1]root@solaris:~#
```

2. 卸载受影响的文件系统。

在示例中，卸载文件系统 *samfs2*：

```
[server1]root@solaris:~# umount samfs2
[server1]root@solaris:~#
```

3. 在文本编辑器中打开 */etc/vfstab* 文件，并确保未设置 *noquota* 挂载选项。

在示例中，在 *vi* 文本编辑器中打开文件。*noquota* 挂载选项已设置：

```
[server1]root@solaris:~# vi /etc/vfstab
#File
#Device      Device  Mount          System  fsck  Mount  Mount
#to Mount    to fsck Point          Type    Pass  at Boot Options
#-----
/devices     -      /devices      devfs   -     no     -
/proc        -      /proc          proc    -     no     -
...
samfs2       -      /samfs2       samfs   -     no     noquota
```

4. 如果已在 */etc/vfstab* 文件中设置了 *noquota* 挂载选项，请删除该选项并保存文件。

```
[server1]root@solaris:~# vi /etc/vfstab
#File
#Device      Device  Mount          System  fsck  Mount  Mount
#to Mount    to fsck Point          Type    Pass  at Boot Options
#-----
/devices     -      /devices      devfs   -     no     -
/proc        -      /proc          proc    -     no     -
...
samfs2       -      /samfs2       samfs   -     no     -
:wq
[server1]root@solaris:~#
```

5. 在文本编辑器中打开 */etc/opt/SUNWsamfs/samfs.cmd* 文件，并确保未设置 *noquota* 挂载选项。

在示例中，在 *vi* 文本编辑器中打开文件。*noquota* 挂载选项未设置：

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/samfs.cmd
```



```

# These are the defaults. To change the default behavior, uncomment the
# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
#
#inodes = 0
#fs = hsmfs1
# forcedirectio (default no forcedirectio)
# high = 80
# low = 70
# weight_size = 1.
# weight_age = 1.
# readahead = 128
...
# dio_wr_ill_min = 0
# dio_wr_consec = 3
# qwrite      (ma filesystem, default no qwrite)
# shared_writer (ma filesystem, default no shared_writer)
# shared_reader (ma filesystem, default no shared_reader)

```

6. 如果已在 `/etc/opt/SUNWsamfs/samfs.cmd` 文件中设置了 `noquota` 挂载选项，请删除该选项并保存文件。
7. 修复不一致的配额记录。使用命令 `samfsck -F family-set-name`，其中 `family-set-name` 是 `/etc/opt/SUNWsamfs/mcf` 文件中文件系统的系列集名称。

```
[server1]root@solaris:~# samfsck -F samfs2
```

8. 重新挂载文件系统。

当系统检测到在文件系统的根目录中有一个或多个配额文件时，会启用配额。

您不需要在 `/etc/vfstab` 或 `samfs.cmd` 文件中包括 `quota` 挂载选项，因为默认情况下挂载文件系统时会启用配额。

```
[server1]root@solaris:~# mount /samfs2
[server1]root@solaris:~#
```

9. 在此处停止。

检查配额

管理员和用户都可以监视配额和资源使用量。`root` 用户可以使用 `samquota` 命令生成关于用户、组或管理集的配额报告。文件系统用户可以使用 `squota` 命令检查自己的配额。

请参见以下过程：

- [以文件系统管理员身份监视配额](#)
- [监视您自己的用户配额](#)

以文件系统管理员身份监视配额

1. 以 *root* 用户身份登录到文件系统服务器。

在示例中，服务器名为 *server1*：

```
[server1]root@solaris:~#
```

2. 要显示所有组的配额统计信息，请使用命令 *samquota -g [directory-or-file]*，其中可选的 *directory-or-file* 参数将报告的范围限定于指定目录上挂载的文件系统、指定的目录本身或指定的文件。

在示例中，我们请求有关 *hsmfs1* 文件系统的报告，该文件系统挂载在 */hsm/hsmfs1* 上：

```
[server1]root@solaris:~# samquota -g /hsm/hsmfs1
```

3. 要显示所有管理集的配额统计信息，请使用命令 *samquota -a [directory-or-file]*，其中可选的 *directory-or-file* 参数将报告的范围限定于指定目录上挂载的文件系统、指定的目录本身或指定的文件。

在示例中，我们请求有关 *hsmfs1* 文件系统的报告，该文件系统挂载在 */hsm/hsmfs1* 上：

```
[server1]root@solaris:~# samquota -a /hsm/hsmfs1
```

4. 要显示所有用户的配额统计信息，请使用命令 *samquota -u [directory-or-file]*，其中可选的 *directory-or-file* 参数将报告的范围限定于指定目录上挂载的文件系统、指定的目录本身或指定的文件。

在示例中，我们请求有关 *hsmfs1* 文件系统的报告，该文件系统挂载在 */hsm/hsmfs1* 上：

```
[server1]root@solaris:~# samquota -u /hsm/hsmfs1
```

5. 要显示特定组的配额统计信息，请使用命令 *samquota -G groupID [directory-or-file]*，其中 *groupID* 指定组名称或组的整数标识符，可选的 *directory-or-file* 参数将报告的范围限定于指定目录上挂载的文件系统、指定的目录本身或指定的文件。

在示例中，我们请求有关 *hsmfs1* 文件系统中 *dev* 组的配额的报告，该文件系统挂载在 */hsm/hsmfs1* 上：

```
[server1]root@solaris:~# samquota -G dev /hsm/hsmfs1
```

- 要显示特定管理集的配额统计信息，请使用命令 `samquota -A adminsetID [directory-or-file]`，其中 `adminsetID` 指定管理组的整数标识符，可选的 `directory-or-file` 参数将报告的范围限定于指定目录上挂载的文件系统、指定的目录本身或指定的文件。

在示例中，我们请求有关 `hsmfs1` 文件系统中 `1` 管理集的配额的报告，该文件系统挂载在 `/hsm/hsmfs1` 上：

```
[server1]root@solaris:~# samquota -A 1 /hsm/hsmfs1
```

- 要显示特定用户的配额统计信息，请使用命令 `samquota -U userID [directory-or-file]`，其中 `userID` 指定用户名或用户的整数标识符，可选的 `directory-or-file` 参数将报告的范围限定于指定目录上挂载的文件系统、指定的目录本身或指定的文件。

在示例中，我们请求有关 `hsmfs1` 文件系统中 `jr23547` 用户的配额的报告，该文件系统挂载在 `/hsm/hsmfs1` 上：

```
[server1]root@solaris:~# samquota -U jr23547 /hsm/hsmfs1
```

- 在此处停止。

监视您自己的用户配额

- 使用您的用户 ID 登录到文件系统主机。

在示例中，以 `od447` 用户身份登录到主机 `server1`：

```
[server1]od447@solaris:~#
```

- 要显示所有组的配额统计信息，请使用命令 `squota [directory-or-file]`，其中可选的 `directory-or-file` 参数将报告的范围限定于指定目录上挂载的文件系统、指定的目录本身或指定的文件。

在示例中，请求关于所有文件系统的报告：

```
[server1]od447@solaris:~# sqquota
```

				Limits	
Type	ID	In Use	Soft	Hard	
/hsm/hsmfs1					
Files group	101	1	1000	1200	
Blocks group	101	8	20000	30000	

```

Grace period                25920
No user quota entry.
[server1]od447@solaris:~#

```

3. 在此处停止。

暂时延长或取消宽限期

当需要暂时延长宽限期或需要缩短宽限期时，可以按以下方式操作：

- [将宽限期延长指定的时间量](#)
- [重新启动宽限期](#)
- [提前结束宽限期。](#)

将宽限期延长指定的时间量

如果组、用户或管理集已超出了其配额的指定软限制，但是超出软限制的时长需要暂时大于当前宽限期所允许的时间，则可以按以下方式延长：

1. 以 *root* 用户身份登录到文件系统服务器。

在示例中，登录到主机 *server1*：

```
[server1]root@solaris:~#
```

2. 检查需要延长的配额。使用命令 *samquota -quota-type ID [directory-or-file]*，其中：

- *quota-type ID* 是 *G* 加上组名称或 ID 编号，*A* 加上管理集 ID 编号，或者 *U* 加上用户名或 ID 编号。
- *directory-or-file* (可选) 是您需要延长宽限期的特定文件系统的挂载点目录或特定目录或文件。

在示例中，*dev* 组明显超出了软限制并且距离其宽限期只剩下几个小时：

```

[server1]root@solaris:~# samquota -G dev /hsm/hsmfs1

```

		Online Limits				Total Limits		
Type	ID	In Use	Soft	Hard	In Use	Soft	Hard	
/hsm/hsmfs1								
Files	group 101	323	15000	30000	323	15000	30000	
Blocks	group 101	3109330961	2013265920	3019898880	3109330961	2013265920	3019898880	
Grace period			4320			4320		

```

---> Warning: soft limits to be enforced in 2h21m16s
[server1]root@solaris:~#

```

3. 如果有必要，延长宽限期。使用命令 `samquota -quota-type ID -x number-seconds [directory-or-file]`，其中：
 - `quota-type ID` 是 `G` 加上组名称或 ID 编号，`A` 加上管理集 ID 编号，或者 `U` 加上用户名或 ID 编号。
 - `directory-or-file`（可选）是您需要延长宽限期的特定文件系统的挂载点目录或特定目录或文件。
 - `number-seconds` 是一个整数，表示延期的秒数（有关其他指定时间的方法，请参见 `samquota` 手册页）。

当提示继续时，输入 `y (yes)`。

在示例中，将 `dev` 组在 `hsmfs1` 文件系统中的文件宽限期延长到 `2678400` 秒（31 天）：

```
[server1]root@solaris:~# samquota -G dev -x 2678400 /hsm/hsmfs1
Setting Grace Timer:  continue? y
```

当重新检查 `dev` 组配额时，宽限期已延长：

```
[server1]root@solaris:~# samquota -G dev /hsm/hsmfs1
```

			Online Limits			Total Limits		
Type	ID	In Use	Soft	Hard	In Use	Soft	Hard	
/hsm/hsmfs1								
Files group	101	323	15000	30000	323	15000	30000	
Blocks group	101	43208	2013265920	3019898880	43208	2013265920	3019898880	
Grace period			2678400			2678400		
---> Warning: soft limits to be enforced in 31d								

```
[server1]root@solaris:~#
```

4. 如果组、管理集或用户经常需要延长，请重新评估存储需求并/或考虑永久增大宽限期。使用“为组、项目、目录和用户设置配额”过程。
5. 在此处停止。

重新启动宽限期

如果组、用户或管理集已超出了其配额的指定软限制，并且无法在当前宽限期到期之前尽快释放空间以降至软限制之下，则可以重新启动宽限期。执行如下操作：

1. 以 `root` 用户身份登录到文件系统服务器。

在示例中，登录到主机 `server1`：

```
[server1]root@solaris:~#
```

- 检查需要延长的配额。使用命令 `samquota -quota-type ID [directory-or-file]`，其中：
 - `quota-type ID` 是 `G` 加上组名称或 ID 编号，`A` 加上管理集 ID 编号，或者 `U` 加上用户名或 ID 编号。
 - `directory-or-file`（可选）是您需要延长宽限期的特定文件系统的挂载点目录或特定目录或文件。

在示例中，`cit` 组超出了 `hsmfs1` 文件系统的软限制，并且距离其宽限期仅剩一个多小时：

```
[server1]root@solaris:~# samquota -G cit /hsm/hsmfs1
                                     Online Limits
                                     Soft      Hard
Type  ID   In Use      Soft      Hard      In Use      Soft  Hard
/hsm/hsmfs1
Files group 119      762        750      1500      762        750  1500
Blocks group 119 3109330961 2013265920 3019898880 120096782 157286400
235929600
Grace period          4320          4320
---> Warning: soft limits to be enforced in 1h11m23s
[server1]root@solaris:~#
```

- 要在下次分配文件或块时将宽限期重置为其完整的起始大小，请清除宽限期计时器。使用命令 `samquota -quota-type ID -x clear [directory-or-file]`，其中：
 - `quota-type ID` 是 `G` 加上组名称或 ID 编号，`A` 加上管理集 ID 编号，或者 `U` 加上用户名或 ID 编号。
 - `directory-or-file`（可选）是您需要延长宽限期的特定文件系统的挂载点目录或特定目录或文件。

当提示继续时，输入 `y` (yes)。

在示例中，清除了 `hsmfs1` 文件系统上 `cit` 组的配额的宽限期计时器。

```
[server1]root@solaris:~# samquota -G cit -x clear /hsm/hsmfs1
Setting Grace Timer: continue? y
[server1]root@solaris:~#
```

当重新检查 `cit` 组的配额时，可以发现已经分配了一个文件并且宽限期已重置为 `12h`，即 12 个小时（4320 秒）：

```
[server1]root@solaris:~# samquota -G cit /hsm/hsmfs1
                                     Online Limits
                                     Soft      Hard
Type  ID   In Use      Soft      Hard      In Use      Soft  Hard
/hsm/hsmfs1
```

```

Files group 119          763          750          1500          763          750 1500
Blocks group 119 3109330961 2013265920 3019898880 120096782 157286400
235929600
Grace period          4320          4320
---> Warning: soft limits to be enforced in 12h
[server1]root@solaris:~#

```

4. 另外，要立即将宽限期重置为其完整的起始大小，请重置宽限期计时器。使用命令 `samquota -quota-type ID -x reset [directory-or-file]`。
 - `quota-type ID` 是 `G` 加上组名称或 ID 编号，`A` 加上管理集 ID 编号，或者 `U` 加上用户名或 ID 编号。
 - `directory-or-file`（可选）是您需要延长宽限期的特定文件系统的挂载点目录或特定目录或文件。

当提示继续时，输入 `y` (yes)。

在示例中，清除了 `hsmfs1` 文件系统上 `cit` 组的配额的宽限期计时器。

```

[server1]root@solaris:~# samquota -G cit -x reset /hsm/hsmfs1
Setting Grace Timer: continue? y
[server1]root@solaris:~#

```

当重新检查 `cit` 组的配额时，可以发现宽限期已重置为 `12h`，即 12 个小时（4320 秒）：

```

[server1]root@solaris:~# samquota -G cit /hsm/hsmfs1
                                     Online Limits          Total Limits
          Type  ID   In Use      Soft      Hard      In Use      Soft  Hard
/hsm/hsmfs1
Files group 119     762       750     1500       762     750 1500
Blocks group 119 3109330961 2013265920 3019898880 120096782 157286400
235929600
Grace period          4320          4320
---> Warning: soft limits to be enforced in 12h
[server1]root@solaris:~#

```

5. 在此处停止。

提前结束宽限期

1. 以 `root` 用户身份登录到文件系统服务器。

在示例中，登录到主机 `server1`：

```
[server1]root@solaris:~#
```

2. 检查您需要缩短的宽限期。使用命令 `samquota -quota-type ID [directory-or-file]`, 其中:
 - `quota-type ID` 是 `G` 加上组名称或 ID 编号, `A` 加上管理集 ID 编号, 或者 `U` 加上用户名或 ID 编号。
 - `directory-or-file` (可选) 是您需要延长宽限期的特定文件系统的挂载点目录或特定目录或文件。

在示例中, `cit` 组超出了软限制并且距离其宽限期还剩十一个小时, 但是需要提前结束宽限期:

```
[server1]root@solaris:~# samquota -G cit /hsm/hsmfs1
```

		Online Limits				Total Limits	
Type	ID	In Use	Soft	Hard	In Use	Soft	
Hard							
/hsm/hsmfs1							
Files	group 119	822	750	1500	822	750	
				1500			
Blocks	group 119	3109330961	2013265920	3019898880	120096782	157286400	
				235929600			
Grace period			4320			4320	

```

---> Warning: soft limits to be enforced in 11h
[server1]root@solaris:~#

```

3. 使宽限期到期。使用命令 `samquota -quota-type ID -x expire [directory-or-file]`, 其中:
 - `quota-type ID` 是 `G` 加上组名称或 ID 编号, `A` 加上管理集 ID 编号, 或者 `U` 加上用户名或 ID 编号。
 - `directory-or-file` (可选) 是您需要延长宽限期的特定文件系统的挂载点目录或特定目录或文件。

在示例中, 使 `cit` 组的宽限期到期:

```
root@solaris:~# samquota -G cit -x expire /hsm/hsmfs1
Setting Grace Timer: continue? y
```

当重新检查配额时, `cit` 组的软限制正被强制用作硬限制:

```
[server1]root@solaris:~# samquota -G cit /hsm/hsmfs1
```

		Online Limits				Total Limits	
Type	ID	In Use	Soft	Hard	In Use	Soft	Hard
/hsm/hsmfs1							


```

Files group 119          762          750          1500          762          750 1500
Blocks group 119 3109330961 2013265920 3019898880 120096782 157286400
235929600
Grace period          4320          4320
---> Online soft limits under enforcement (since 6s ago)
[server1]root@solaris:~#

```

4. 在此处停止。

停止新的资源分配

您可以通过创建不一致的配额值限制文件系统资源分配。当文件系统检测到用户、组或管理集的配额值不一致时，会阻止用户、组或管理集使用其他任何系统资源。因此，将配额的硬限制设置为低于对应的软限制将停止进一步的分配。要使用此方法，请执行如下操作：

1. 以 *root* 用户身份登录到文件系统服务器。

在示例中，登录到主机 *server1*：

```
[server1]root@solaris:~#
```

2. 备份配额以便以后可以恢复。导出当前配置，并将信息重定向到某个文件。使用命令 *samquota -quota-type ID [directory-or-file] > file*，其中：

- *quota-type ID* 是 *G* 加上组名称或 ID 编号，*A* 加上管理集 ID 编号，或者 *U* 加上用户名或 ID 编号。
- *directory-or-file*（可选）是您需要延长宽限期的特定文件系统的挂载点目录或特定目录或文件。
- *file* 是输出文件的名称。

在示例中，我们将 *cit* 组的配额导出到 *root* 用户起始目录中的文件 *restore.hsmfs1.quota_g.cit* 中（请注意，下面的命令是作为单行输入的一使用反斜杠字符对换行符进行转义）：

```
[server1]root@solaris:~# samquota -G cit -e /hsm/hsmfs1 > /
/root/restore.hsmfs1.quota_g.cit
[server1]root@solaris:~#
```

3. 检查输出。使用 Solaris 命令 *more < file*，其中 *file* 是输出文件的文件名。

```
[server1]root@solaris:~# more < /root/restore.hsmfs1.quota_g.cit
# Type ID
#
#           Online Limits
#           soft          hard
# Files
```

```
# Blocks
# Grace Periods
samquota -G 119 /
    -f      750:s:o  -f      1500:h:o      -f      750:s:t  -f      1500:h:t /
    -b 157286400:s:o  -b 235929600:h:o      -b 157286400:s:t  -b 235929600:h:t /
    -t 4320:o                      -t 4320:t
[server1]root@solaris:~#
```

4. 将配额的硬限制设置为 0（零），将软限制设置为 1（或任何非零值）。使用命令 `samquota -quota-type ID -f 1:s -f 0:h -b 1:s -b 0:h [directory-or-file]`。
 - `quota-type ID` 是 *G* 加上组名称或 ID 编号，*A* 加上管理集 ID 编号，或者 *U* 加上用户名或 ID 编号。
 - `directory-or-file`（可选）是您需要延长宽限期的特定文件系统的挂载点目录或特定目录或文件。

在示例中，使 `/hsm/hsmfs1` 文件系统中 `cit` 组的配额不一致，并借此停止新的资源分配。

```
[server1]root@solaris:~# samquota -G cit -f 1:s -f 0:h -b 1:s -b 0:h /hsm/hsmfs1
[server1]root@solaris:~#
```

当检查 `cit` 组的配额时，可以发现生效的是零配额。感叹号字符 (!) 将所有当前使用量显示为超出配额，因此将不会执行进一步的分配：

```
[server1]root@solaris:~# samquota -G cit /hsm/hsmfs1
Online Limits                Total Limits
Type  ID                        In Use   Soft    Hard      In Use   Soft    Hard
/sam6
Files  group 119                822!    1       0         822!    1       0
Blocks group 119 3109330961!    1       0  3109330961!    1       0
Grace period                        4320                      4320
---> Quota values inconsistent; zero quotas in effect.
[server1]root@solaris:~#
```

5. 当做好准备后，通过将修改的配额恢复为其原始状态来继续正常的分配。执行您创建为 shell 脚本的备份文件。使用 Solaris 命令 `sh file`，其中 `file` 是备份文件的文件名。

在示例中，通过执行 `/root/restore.hsmfs1.quota_g.cit` 文件恢复 `cit` 组的配额

```
[server1]root@solaris:~# sh /root/restore.hsmfs1.quota_g.cit
Setting Grace Timer:  continue? y
```

```
Setting Grace Timer: continue? y
[server1]root@solaris:~#
```

当检查配额时，可以发现已恢复了正常限制并且不再阻止分配：

```
[server1]root@solaris:~# samquota -G cit /hsm/hsmfs1
                                Online Limits
                                Soft      Hard
Type  ID   In Use      Soft      Hard      In Use      Soft  Hard
/hsm/hsmfs1
Files group 119      822        750      1500        822      750  1500
Blocks group 119 3109330961 2013265920 3019898880 120096782 157286400
235929600
Grace period          4320          4320
---> Warning: soft limits to be enforced in 11h
[server1]root@solaris:~#
```

6. 在此处停止。

删除文件系统的配额

要删除或禁用文件系统的配额，请在挂载过程中禁用配额。

1. 以 *root* 用户身份登录到文件系统服务器。

在示例中，登录到主机 *server1*：

```
[server1]root@solaris:~#
```

2. 在文本编辑器中打开 */etc/vfstab* 文件，向文件系统行的 "Mount Options" 列中添加 *noquota* 挂载选项，然后保存文件。

在示例中，在 *vi* 文本编辑器中打开文件，并为 *hsmfs1* 文件系统设置 *noquota* 挂载选项：

```
[server1]root@solaris:~# vi /etc/vfstab
#File
#Device      Device  Mount          System fsck  Mount  Mount
#to Mount    to fsck  Point          Type  Pass  at Boot  Options
#-----
/devices     -       /devices      devfs -    no     -
/proc       -       /proc         proc  -    no     -
...
hsmfs1      -       /hsm/hsmfs1   samfs -    no     noquota
:wq
[server1]root@solaris:~#
```

3. 如果文件系统已挂载，请将其卸载。

您必须卸载然后重新挂载文件系统，以便使操作系统重新加载 `/etc/vfstab` 文件并执行指定的更改。在示例中，卸载 `hsmfs1` 文件系统：

```
[server1]root@solaris:~# umount hsmfs1
[server1]root@solaris:~#
```

4. 挂载文件系统。

在示例中，挂载 `hsmfs1` 文件系统：

```
[server1]root@solaris:~# mount hsmfs1
[server1]root@solaris:~#
```

5. 如果您预计以后要恢复配额，请保留配额文件。

当准备好恢复配额时，只需要简单地卸载文件系统，在文件系统上运行命令 `samfsck -F`，删除 `noquota` 挂载选项，然后重新挂载文件系统。

6. 如果预计不会恢复配额或者如果需要回收配额文件占用的空间，请使用 Solaris 命令 `rm` 从文件系统的根目录中删除文件 `.quota_g`、`.quota_a` 和/或 `.quota_u`。

在示例中，从 `/hsm/hsmfs1` 文件系统根目录中删除所有配额文件：

```
[server1]root@solaris:~# rm /hsm/hsmfs1/.quota_g
[server1]root@solaris:~# rm /hsm/hsmfs1/.quota_a
[server1]root@solaris:~# rm /hsm/hsmfs1/.quota_u
[server1]root@solaris:~#
```

7. 在此处停止。

控制归档和回写操作

通常，管理归档文件系统的方法与非归档文件系统大致相同。不过，在执行大多数文件系统管理任务之前必须停止归档进程。当处于活动状态时，归档进程会对文件系统的主磁盘高速缓存进行更改。因此，在对磁盘高速缓存执行维护工作之前必须停止这些进程。本节介绍以下任务：

- [使归档和回写进程闲置](#)
- [停止归档和回写进程](#)
- [重新启动归档和回写进程。](#)

使归档和回写进程闲置

1. 以 `root` 用户身份登录到文件系统主机。

在示例中，登录到主机 *server1*：

```
[server1]root@solaris:~#
```

2. 使所有归档进程闲置。使用命令 *samcmd aridle*。

此命令将允许当前的归档和回写操作完成，但不会启动任何新作业：

```
[server1]root@solaris:~# samcmd aridle
[server1]root@solaris:~#
```

3. 使所有回写进程闲置。使用命令 *samcmd stidle*。

此命令将允许当前的归档和回写操作完成，但不会启动任何新作业：

```
[server1]root@solaris:~# samcmd stidle
[server1]root@solaris:~#
```

4. 等待活动的归档作业完成。使用命令 *samcmd a* 检查归档进程的状态。

当归档进程为 *Waiting for :arrun* 时，归档进程处于空闲状态：

```
[server1]root@solaris:~# samcmd a
Archiver status samcmd      5.4 10:20:34 May 20 2014
samcmd on samfs-mds
sam-archiverd: Waiting for :arrun
sam-arfind: ...
Waiting for :arrun
```

5. 等待活动的回写作业完成。使用命令 *samcmd u* 检查回写进程的状态。

当回写进程为 *Waiting for :strun* 时，回写进程处于空闲状态：

```
[server1]root@solaris:~# samcmd u
Staging queue samcmd      5.4 10:20:34 May 20 2014
samcmd on solaris.demo.lan
Staging queue by media type: all
sam-stagerd: Waiting for :strun
root@solaris:~#
```

6. 要完全停止系统，还需要停止归档和回写进程。

停止归档和回写进程

1. 如果尚未执行此操作，将归档和回写进程闲置。

2. 如果您尚未以 *root* 用户身份登录到文件系统主机，请执行此操作。

在示例中，登录到主机 *server1*：

```
[server1]root@solaris:~#
```

3. 在继续操作之前，使所有可移除的介质驱动器停工。针对每个驱动器，使用命令 *samcmd equipment-number idle*，其中 *equipment-number* 是在 */etc/opt/SUNWsamfs/mcf* 文件中分配给驱动器的设备序号。

此命令将允许当前的归档和回写作业在驱动器关闭之前完成，但不会启动任何新作业。在示例中，使序号分别为 *801*、*802*、*803* 和 *804* 的四个驱动器闲置：

```
[server1]root@solaris:~# samcmd 801 idle
[server1]root@solaris:~# samcmd 802 idle
[server1]root@solaris:~# samcmd 803 idle
[server1]root@solaris:~# samcmd 804 idle
[server1]root@solaris:~#
```

4. 等待正在运行的作业完成。

可以使用命令 *samcmd r* 检查驱动器的状态。当所有驱动器都处于 *notrdy* 和 *empty* 时，已准备好继续。

```
[server1]root@solaris:~# samcmd r
Removable media samcmd      5.4 18:37:09 Feb 17 2014
samcmd on hsmfs1host
ty  eq  status      act  use  state  vsn
li  801  -----p      0   0%  notrdy
      empty
li  802  -----p      0   0%  notrdy
      empty
li  803  -----p      0   0%  notrdy
      empty
li  804  -----p      0   0%  notrdy
      empty
[server1]root@solaris:~#
```

5. 当归档程序和回写程序进程处于空闲状态，并且磁带机都处于 *notrdy* 时，停止磁带库控制守护进程。使用命令 *samd stop*。

```
[server1]root@solaris:~# samd stop
[server1]root@solaris:~#
```

6. 继续进行文件系统维护。

7. 当维护完成时，重新启动归档和回写进程。

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

8. 在此处停止。

重新启动归档和回写进程

当准备就绪时，恢复正常的自动化操作，按下述方法继续操作：

1. 以 *root* 用户身份登录到文件系统主机。

在示例中，登录到主机 *server1*：

```
[server1]root@solaris:~#
```

2. 重新启动 Oracle HSM 磁带库控制守护进程。使用命令 *samd start*。

```
[server1]root@solaris:~# samd start  
[server1]root@solaris:~#
```

3. 在此处停止。

重命名文件系统

重命名文件系统是一个两步骤过程。首先，通过编辑 */etc/opt/SUNWsamfs/mcf* 文件更改文件系统的系列集名称。然后，使用 *samfsck -R -F* 命令读取新名称并更新对应磁盘设备上的超级块。要重命名文件系统，请使用以下过程：

重命名文件系统

1. 以 *root* 用户身份登录到文件系统服务器。

在示例中，登录到主机 *server1*：

```
[server1]root@solaris:~#
```

2. 如果要修复归档文件系统，请在继续操作前执行“[使归档和回写进程闲置](#)”过程。
3. 卸载需要重命名的文件系统。

在示例中，卸载文件系统 *hsmfs1*：

```
[server1]root@solaris:~# umount hsmfs1
```

4. 在文本编辑器中打开 */etc/opt/SUNWsamfs/mcf* 文件并找到需要重命名的文件系统。

在示例中，使用 *vi* 编辑器。需要更改 *hsmfs1* 文件系统的名称：

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment  Equipment  Family      Device  Additional
# Identifier     Ordinal   Type       Set         State   Parameters
#-----
hsmfs1          100      ms        hsmfs1     on
/dev/dsk/c1t3d0s3  101     md        hsmfs1     on
/dev/dsk/c1t4d0s5  102     md        hsmfs1     on
```

5. 在文件的第四列中，将该文件系统的系列集名称更改为新值。您还可以更改第一列中的文件系统设备标识符，但不要更改其他任何值。保存文件并关闭编辑器。

在示例中，将文件系统的设备标识符和系列集名称同时从 *hsmfs1* 更改为 *samqfs-hpcc*：

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment  Equipment  Family      Device  Additional
# Identifier     Ordinal   Type       Set         State   Parameters
#-----
samqfs-hpcc     100      ms        samqfs-hpcc on
/dev/dsk/c1t3d0s3  101     md        samqfs-hpcc on
/dev/dsk/c1t4d0s5  102     md        samqfs-hpcc on
:wq
root@solaris:~#
```

6. 重写文件系统超级块以反映新系列集名称。使用命令 *samfsck -R -F family-set-name*，其中 *family-set-name* 是刚刚在 */etc/opt/SUNWsamfs/mcf* 文件中指定的系列集名称。

当使用 *-R* 和 *-F* 选项发出 *samfsck* 命令时，将从 */etc/opt/SUNWsamfs/mcf* 文件中读取新系列集名称和对应的磁盘存储设备标识符。然后，使用新的系列集名称在指定的磁盘设备上重写超级块。在示例中，使用新的 *samqfs-hpcc* 系列集名称运行该命令：

```
[server1]root@solaris:~# samfsck -R -F samqfs-hpcc
```

7. 在文本编辑器中打开 */etc/vfstab* 文件并找到您要重命名的文件系统的条目。

在示例中，在 *vi* 文本编辑器中打开文件。需要将 *hsmfs1* 文件系统条目更改为使用新名称：

```
[server1]root@solaris:~# vi /etc/vfstab
#File
#Device      Device  Mount          System  fsck  Mount  Mount
```



```

#to Mount      to fsck  Point          Type    Pass  at Boot  Options
#-----
/devices        -        /devices       devfs   -     no      -
/proc          -        /proc          proc    -     no      -
...
hsmfs1         -        /hsm/hsmfs1    samfs   -     no      -

```

8. 在已重命名的文件系统的 `/etc/vfstab` 条目中，更改第一列中的文件系统名称和第三列中的挂载点目录名称（如果需要），然后保存该文件。

在示例中，将 `hsmfs1` 文件系统的名称更改为 `samqfs-hpcc` 并相应地更改挂载点：

```

[server1]root@solaris:~# vi /etc/vfstab
#File
#Device      Device  Mount          System  fsck  Mount  Mount
#to Mount    to fsck  Point          Type    Pass  at Boot  Options
#-----
/devices      -        /devices       devfs   -     no      -
/proc        -        /proc          proc    -     no      -
...
samqfs-hpcc -        /samqfs-hpcc  samfs   -     no      -
:wq
[server1]root@solaris:~#

```

9. 如果需要，将为新文件系统创建新的挂载点目录，并为挂载点设置访问权限。

用户必须具有执行 (x) 权限才能转到挂载点目录并访问挂载的文件系统中的文件。在示例中，创建 `/samqfs-hpcc` 挂载点目录并将权限设置为 `755 (-rwxr-xr-x)`：

```

[server1]root@solaris:~# mkdir /samqfs-hpcc
[server1]root@solaris:~# chmod 755 /samqfs-hpcc
[server1]root@solaris:~#

```

10. 运行 `sam-fsd` 命令检查 `mcf` 文件中是否有错误并纠正检测到的任何错误。

`sam-fsd` 是一个读取 Oracle HSM 配置文件的初始化命令。该命令会在遇到以下错误时停止：

```
[server1]root@solaris:~# sam-fsd
```

11. 指示 Oracle HSM 软件重新读取 `mcf` 文件并相应地重新配置自身。使用命令 `samd config`。

```
[server1]root@solaris:~# samd config
```

12. 如果 `samd config` 报告了错误，请更正错误直至没有错误，然后重新发出该命令。
13. 挂载文件系统。

在示例中，使用新的挂载点目录：

```
[server1]root@solaris:~# mount /samqfs-hpcc
```

14. 在此处停止。

修复文件系统

当文件系统通过 `samu`、Oracle HSM Manager 或 `/var/adm/sam-log` 文件报告了错误时，请遵循以下过程：

修复文件系统

1. 以 `root` 用户身份登录到文件系统服务器。

在示例中，登录到主机 `server1`：

```
[server1]root@solaris:~#
```

2. 如果要修复归档文件系统，请在继续操作前执行“[使归档和回写进程闲置](#)”过程。
3. 卸载受影响的文件系统。

如果在等待归档停止，则可能需要尝试多次。在示例中，卸载文件系统 `hsmfs1`：

```
[server1]root@solaris:~# umount hsmfs1
samfs umount: /hsm/hsmfs1: is busy
[server1]root@solaris:~# umount hsmfs1
[server1]root@solaris:~#
```

4. 修复文件系统。使用命令 `samfsck -F -V family-set-name`，其中 `family-set-name` 是 `/etc/opt/SUNWsamfs/mcf` 文件中为文件系统指定的系列集名称。

通常，将修复结果保存到带日期戳的文件中是个很好的主意，必要时可供以后参考和诊断。因此在本示例中，我们将 `samfsck` 输出通过管道传输到命令 `tee /var/tmp/samfsck-FV.family-set-name.`date '+%Y%m%d.%H%M%S'`` 中，从而保存结果（请注意，下面的命令是作为单行输入的一使用反斜杠字符对换行符进行转义）：

```
[server1]root@solaris:~# samfsck -F -V hsmfs1 | tee /
/var/tmp/samfsck-FV.hsmfs1.`date '+%Y%m%d.%H%M%S'`
```

```

name:      /hsm/hsmfs1      version:    2A
First pass
Second pass
Third pass
NOTICE: ino 2.2,  Repaired link count from 8 to 14
Inodes processed: 123392
total data kilobytes      = 1965952
total data kilobytes free = 1047680
total meta kilobytes      = 131040
total meta kilobytes free = 65568
INFO:  FS samma1 repaired:
        start:  May 19, 2014 10:57:13 AM MDT
        finish: May 19, 2014 10:57:37 AM MDT
NOTICE: Reclaimed 70057984 bytes
NOTICE: Reclaimed 9519104 meta bytes
[server1]root@solaris:~#

```

5. 重新挂载文件系统。

```

[server1]root@solaris:~# mount /hsm/hsmfs1
[server1]root@solaris:~#

```

6. 在此处停止。

向文件系统添加设备

在向现有文件系统添加设备之前，应当考虑您的需求和备选方案。确保对现有文件系统进行扩展是满足不断增长的容量需求的最佳方式。如果您需要更多的物理存储空间来容纳新项目或用户团体，则创建一个或多个新的 Oracle HSM 文件系统可能是一个较好的选择。多个较小的文件系统通常可以提供比一个较大的文件系统更好的性能，并且较小的文件可能更容易创建和维护。

在决定需要扩展文件系统后，请采取以下方法之一：

- [向已挂载的文件系统添加设备](#)（建议使用）
- [向未挂载的文件系统添加设备](#)

向已挂载的文件系统添加设备

执行如下操作：

1. 以 *root* 用户身份登录到文件系统服务器。

在示例中，登录到主机 *server1*：

```

[server1]root@solaris:~#

```

- 在文本编辑器中打开 `/etc/opt/SUNWsamfs/mcf` 文件并找到需要扩展的文件系统。

在示例中，使用 `vi` 编辑器。我们需要扩展两个文件系统，通用的 `samqfsms` 文件系统和高性能的 `samqfs2ma` 文件系统：

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment  Equipment  Family    Device    Additional
# Identifier     Ordinal   Type       Set       State     Parameters
#-----
samqfsms        100       ms         samqfsms  on
/dev/dsk/c1t3d0s3 101       md         samqfsms  on
/dev/dsk/c1t4d0s5 102       md         samqfsms  on
samqfs2ma       200       ma         samqfs2ma on
/dev/dsk/c1t3d0s3 201       mm         samqfs2ma on
/dev/dsk/c1t3d0s5 202       md         samqfs2ma on
/dev/dsk/c1t4d0s5 203       md         samqfs2ma on
```

- 如果要向通用的 `ms` 文件系统添加设备，请在 `mcf` 文件中向文件系统定义的末尾添加额外的数据/元数据设备。然后保存文件并关闭编辑器。

最多可以添加 252 个逻辑设备。在示例中，将两个设备（103 和 104）添加到 `samqfsms` 文件系统：

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment  Equipment  Family    Device    Additional
# Identifier     Ordinal   Type       Set       State     Parameters
#-----
samqfsms        100       ms         samqfsms  on
/dev/dsk/c1t3d0s3 101       md         samqfsms  on
/dev/dsk/c1t4d0s5 102       md         samqfsms  on
/dev/dsk/c1t3d0s7 103       md         samqfsms  on
/dev/dsk/c1t4d0s7 104       md         samqfsms  on
:wq
[server1]root@solaris:~#
```

- 如果要向高性能 `ma` 文件系统添加设备，请在 `mcf` 文件中向文件系统定义的末尾添加数据设备以及一个或多个 `mm` 磁盘设备。然后保存文件并关闭编辑器。

请始终将新设备添加在现有设备列表的末尾。最多可以添加 252 个设备，在添加数据设备时按比例添加元数据设备。在示例中，将一个 `mm` 元数据设备（204）和两个 `md` 数据设备（205 和 206）添加到 `samqfs2ma` 文件系统：

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
```

```

# Equipment      Equipment  Equipment  Family    Device  Additional
# Identifier      Ordinal    Type        Set       State   Parameters
#-----
...
samqfs2ma        200        ma          samqfs2ma on
/dev/dsk/c1t3d0s3 201        mm          samqfs2ma on
/dev/dsk/c1t3d0s5 202        md          samqfs2ma on
/dev/dsk/c1t4d0s5 203        md          samqfs2ma on
/dev/dsk/c1t5d0s6 204        mm          samqfs2ma on
/dev/dsk/c1t3d0s7 205        md          samqfs2ma on
/dev/dsk/c1t4d0s7 206        md          samqfs2ma on
:wq
[server1]root@solaris:~#

```

5. 运行 `sam-fsd` 命令检查 `mcf` 文件中是否有错误并纠正检测到的任何错误。

`sam-fsd` 是一个读取 Oracle HSM 配置文件的初始化命令。该命令会在遇到以下错误时停止：

```
[server1]root@solaris:~# sam-fsd
```

6. 如果 `sam-fsd` 命令在 `mcf` 文件中找到错误，请编辑该文件以更正错误，并按照前一步骤中的描述重新检查。

在下面的示例中，`sam-fsd` 报告设备中出现的未指定问题：

```
[server1]root@solaris:~# sam-fsd
Problem in mcf file /etc/opt/SUNWsamfs/mcf for filesystem samqfsms
sam-fsd: Problem with file system devices.
```

通常，这样的错误是不经意的键入错误导致的。此处，在编辑器中打开 `mcf` 文件后，发现在设备 104（第二个新的 `md` 设备）的设备名称中键入了一个字符 `o` 而非 `0`：

```

samqfsms          100        ms          samqfsms  on
/dev/dsk/c1t3d0s3 101        md          samqfsms  on
/dev/dsk/c1t4d0s5 102        md          samqfsms  on
/dev/dsk/c1t3d0s7 103        md          samqfsms  on
/dev/dsk/c1t4dos7 104        md          samqfsms  on

```

7. 如果 `sam-fsd` 命令运行时未出错，则表明 `mcf` 文件正确。继续执行下一步。

以下示例是无误输出的部分列表：

```
[server1]root@solaris:~# sam-fsd
```

```
Trace file controls:
sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
              cust err fatal ipc misc proc date
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
[server1]root@solaris:~#
```

8. 指示 Oracle HSM 软件重新读取 *mcf* 文件并相应地重新配置自身。使用命令 *samd config*。

```
[server1]root@solaris:~# samd config
Configuring SAM-FS
[server1]root@solaris:~#
```

9. 确保 *samd config* 已更新了 Oracle HSM 文件系统配置以包括新设备。使用命令 *samcmd f*。

设备应当处于 *off* 状态。在示例中，*samcmd f* 显示了新设备 103 和 104，这两个设备都处于 *off* 状态：

```
[server1]root@solaris:~# samcmd f
File systems samcmd      5.4 16:57:35 Feb 27 2014
samcmd on server1
ty      eq  state device_name      status      high low mountpoint server
ms      100  on   samqfsms           m----2----- 80% 70% /samqfsms
md      101  on   /dev/dsk/c1t3d0s3
md      102  on   /dev/dsk/c1t4d0s5
md      103  off  /dev/dsk/c1t3d0s7
md      104  off  /dev/dsk/c1t4d0s7
[server1]root@solaris:~#
```

10. 启用新添加的设备。针对每个设备，使用命令 *samcmd add equipment-number*，其中 *equipment-number* 是在 */etc/opt/SUNWsamfs/mcf* 文件中分配给设备的设备序号。

在示例中，启用新设备 103 和 104：

```
[server1]root@solaris:~# samcmd add 103
[server1]root@solaris:~# samcmd add 104
```

11. 如果要向共享文件系统添加设备，请转到[“完成对添加到共享文件系统的新设备的配置”](#)。

12. 如果要向非共享的独立文件系统添加设备，请确保设备已添加并且已准备就绪可供文件系统使用。使用命令 `samcmd m`，然后检查结果。

当设备处于 `on` 状态时，说明它已成功添加并准备就绪可供使用。在示例中，成功添加了设备 `103` 和 `104`：

```
[server1]root@solaris:~# samcmd f
Mass storage status samcmd      5.4 17:17:08 Feb 27 2014
samcmd on server1
ty  eq  status      use state ord  capacity      free  ra  part high low
ms  100 m----2----- 13% on          3.840G      3.588G 1M   16  80% 70%
   md  101          31% on          0  959.938M  834.250M
   md  102          13% on          1  959.938M  834.250M
   md  103          0% on          2  959.938M  959.938M
   md  104          0% on          3  959.938M  959.938M
[server1]root@solaris:~#
```

13. 在此处停止。

完成对添加到共享文件系统的新设备的配置

当向共享文件系统添加设备时，在所有文件系统主机上配置设备之前，必须执行几个额外的步骤。执行如下操作：

1. 以 `root` 用户身份登录到文件系统元数据主机。

在示例中，元数据服务器主机名为 `metadata-server`：

```
[metadata-server]root@solaris:~#
```

2. 确保新设备已添加到元数据服务器。使用命令 `samcmd m`。

当设备处于 `unavail` 状态时，说明它已成功添加但尚未准备就绪可供使用。在示例中，成功添加了设备 `103` 和 `104`：

```
[metadata-server]root@solaris:~# samcmd f
Mass storage status samcmd      5.4 17:17:08 Feb 27 2014
samcmd on metadata-server
ty  eq  status      use state ord  capacity      free  ra  part high low
ms  100 m----2----- 13% on          3.840G      3.588G 1M   16  80% 70%
   md  101          31% on          0  959.938M  834.250M
   md  102          13% on          1  959.938M  834.250M
   md  103          0% unavail  2  959.938M  959.938M
   md  104          0% unavail  3  959.938M  959.938M
[metadata-server]root@solaris:~#
```

- 以 *root* 用户身份登录到每个文件系统客户机主机。

注意要包括潜在的元数据服务器，因为它们也是客户机。在示例中，需要登录到名为 *potential-metadata-server* 的潜在元数据服务器和两个分别名为 *client1* 和 *client2Linux* 的客户机。因此，打开三个终端窗口并使用安全 shell (*ssh*):

```
[metadata-server]root@solaris:~# ssh root@potential-metadata-server
Password:
[potential-metadata-server]root@solaris:~#
[metadata-server]root@solaris:~# ssh root@client1
Password:
[client1]root@solaris:~#
[metadata-server]root@solaris:~# ssh root@client2Linux
Password:
[client2Linux]:[root@linux ~]#
```

- 如果客户机是 Linux 客户机，请卸载共享文件系统。

```
[client2Linux]:[root@linux ~]# umount /samqfsms
```

- 在每台客户机上，在文本编辑器中打开 */etc/opt/SUNWsamfs/mcf* 文件，并在文件系统定义的末尾添加新设备，正如在服务器上所做的一样。

在示例中，将设备 *103* 和 *104* 添加到 *client1* 上的 *mcf* 文件中：

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment  Equipment  Family    Device    Additional
# Identifier      Ordinal   Type       Set       State     Parameters
#-----
samqfsms         100      ms        samqfsms  on        shared
/dev/dsk/c1t3d0s3 101      md        samqfsms  on
/dev/dsk/c1t4d0s5 102      md        samqfsms  on
/dev/dsk/c1t3d0s7 103      md        samqfsms  on
/dev/dsk/c1t4d0s7 104      md        samqfsms  on
:wq
[metadata-server]root@solaris:~#
```

- 在每台客户机上，运行 *sam-fsd* 命令检查 *mcf* 文件中是否有错误并纠正检测到的任何错误。

```
[metadata-server]root@solaris:~# sam-fsd
```

- 在每台客户机上，指示 Oracle HSM 软件重新读取 *mcf* 文件并相应地配置自身：

```
[metadata-server]root@solaris:~# samd config
```


8. 如果客户机是 Linux 客户机，请挂载共享文件系统。

```
[client2Linux]:[root@linux ~]# mount /samqfsms
```

9. 在配置所有客户机后，返回到元数据服务器，并在新设备上启用存储分配。针对每个设备，使用命令 `samcmd alloc equipment-number`，其中 `equipment-number` 是在 `/etc/opt/SUNWsamfs/mcf` 文件中分配给设备的设备序号。

在示例中，在设备 103 和 104 上启用存储分配：

```
[metadata-server]root@solaris:~# samcmd alloc 103
[metadata-server]root@solaris:~# samcmd alloc 104
```

10. 最后，确保设备已准备就绪可供文件系统使用。使用命令 `samcmd m`，然后检查结果。

当设备处于 `on` 状态时，说明它已成功添加并准备就绪可供使用。在示例中，成功添加了设备 103 和 104：

```
[metadata-server]root@solaris:~# samcmd f
Mass storage status samcmd      5.4 17:17:08 Feb 27 2014
samcmd on metadata-server

```

ty	eq	status	use	state	ord	capacity	free	ra	part	high	low
ms	100	m----2-----	13%	on		3.840G	3.588G	1M	16	80%	70%
md	101		31%	on	0	959.938M	834.250M				
md	102		13%	on	1	959.938M	834.250M				
md	103		0%	on	2	959.938M	959.938M				
md	104		0%	on	3	959.938M	959.938M				

```
[metadata-server]root@solaris:~#
```

11. 在此处停止。

向未挂载的文件系统添加设备

执行如下操作：

1. 以 `root` 用户身份登录到文件系统服务器主机。

在示例中，元数据服务器主机名为 `server1`：

```
[server1]root@solaris:~#
```

2. 在卸载归档文件系统之前，必须执行“[使归档和回写进程闲置](#)”过程。
3. 卸载文件系统。

卸载文件系统之前，不要继续操作。在示例中，卸载文件系统 *hsmfs1*：

```
[server1]root@solaris:~# umount hsmfs1
```

4. 在文本编辑器中打开 */etc/opt/SUNWsamfs/mcf* 文件并找到需要扩展的文件系统。

在示例中，使用 *vi* 编辑器。需要扩展 *hsmfs1* 文件系统：

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment          Equipment Equipment Family   Device  Additional
# Identifier         Ordinal   Type     Set     State   Parameters
#-----
hsmfs1              100      ms       hsmfs1  on
/dev/dsk/c1t3d0s3  101      md       hsmfs1  on
/dev/dsk/c1t4d0s5  102      md       hsmfs1  on
```

5. 如果向高性能 *ma* 文件系统添加设备，必须随数据存储一起添加元数据存储。添加足够的附加 *mm* 磁盘设备，以便存储您添加的数据设备的元数据。然后保存文件并关闭编辑器。

最多可以添加 252 个逻辑设备。在示例中，将一个 *mm* 元数据设备添加到 *samqfs2ma* 文件系统，将两个数据设备添加到 *samqfs2ma* 文件系统：

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment          Equipment Equipment Family   Device  Additional
# Identifier         Ordinal   Type     Set     State   Parameters
#-----
samqfs2ma           200      ma       samqfs2ma on
/dev/dsk/c1t3d0s3  201      mm       samqfs2ma on
/dev/dsk/c1t5d0s6  204      mm       samqfs2ma on
/dev/dsk/c1t3d0s5  202      md       samqfs2ma on
/dev/dsk/c1t4d0s5  203      md       samqfs2ma on
/dev/dsk/c1t3d0s7  205      md       samqfs2ma on
/dev/dsk/c1t4dos7  206      md       samqfs2ma on
:wq
[server1]root@solaris:~#
```

6. 如果向通用的 *ms* 文件系统添加设备，请在 *mcf* 文件中向文件系统定义添加额外的数据/元数据设备。然后保存文件并关闭编辑器。

最多可以添加 252 个逻辑设备。在示例中，将两个设备添加到 *hsmfs1* 文件系统中：

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
```

```

# Equipment      Equipment  Equipment  Family    Device    Additional
# Identifier     Ordinal   Type       Set       State     Parameters
#-----
hsmfs1           100       ms         hsmfs1    on
/dev/dsk/c1t3d0s3  101       md         hsmfs1    on
/dev/dsk/c1t4d0s5  102       md         hsmfs1    on
/dev/dsk/c1t3d0s7  103       md         hsmfs1    on
/dev/dsk/c1t4d0s7  104       md         hsmfs1    on
:wq
[server1]root@solaris:~#

```

7. 运行 `sam-fsd` 命令检查 `mcf` 文件中是否有错误并纠正检测到的任何错误。

`sam-fsd` 是一个读取 Oracle HSM 配置文件的初始化命令。该命令会在遇到以下错误时停止：

```
[server1]root@solaris:~# sam-fsd
```

8. 指示 Oracle HSM 软件重新读取 `mcf` 文件并相应地重新配置自身：

```
root@solaris:~# samd config
```

9. 将新设备纳入文件系统中。使用命令 `samgrowfs family-set-name`，其中 `family-set-name` 是 `/etc/opt/SUNWsamfs/mcf` 文件中为文件系统指定的系列集名称。

在示例中，增大 `hsmfs1` 文件系统：

```
[server1]root@solaris:~# samgrowfs hsmfs1
```

10. 重新挂载文件系统。

```
[server1]root@solaris:~# mount /hsm/hsmfs1
```

11. 如果您向归档文件系统添加了设备，请重新启动 Oracle HSM 磁带库管理守护进程。使用命令 `samd start`。

```
[server1]root@solaris:~# samd start
```

12. 如果您在进行更改之前忘了选择卸载文件系统并因此无法挂载文件系统，请通过删除对所添加设备的引用来恢复原始 `mcf` 文件。然后运行 `samd config` 来恢复配置，卸载文件系统并从头开始。

13. 在此处停止。

从文件系统中删除数据设备

需要时，可以从挂载的 Oracle HSM 文件系统中删除数据设备。通常，当需要更换发生故障的单元或需要释放未充分利用的设备以另作它用时，这很有必要。不过，这存在一些限制。

您只能删除数据设备。无法删除用来存储元数据的任何设备，因为元数据定义了文件系统自身的组织。这意味着您只能从高性能 *ma* 文件系统中删除 *md*、*mr* 和分散读写组设备。无法从 *ma* 文件系统中删除 *mm* 元数据设备。由于 *md* 设备既存储数据也存储元数据，因此您也无法从通用的 *ms* 文件系统中移除这些设备。

要删除设备，还必须有地方来移动位于目标设备上的任何有效的数据文件。这意味着您无法删除所有设备。文件系统中必须有一个设备始终保持可用，并且该设备必须有足够的空闲容量来存储位于所删除设备上的所有文件。因此，如果您需要删除一个分散读写组，则必须具有配置了相同数目的成员设备的另一个分散读写组。

要删除设备，请执行如下操作：

- 确保对文件系统元数据和数据进行备份
- 从已挂载的高性能文件系统中删除设备。

确保对文件系统元数据和数据进行备份

执行以下任务：

- 运行 **samexplorer**
- 为文件系统创建恢复点文件。

运行 samexplorer

1. 以 *root* 用户身份登录到文件系统服务器主机。

在示例中，元数据服务器主机名为 *server1*：

```
[server1]root@solaris:~#
```

2. 创建一个 *samexplorer* 报告。使用命令 *samexplorer path/hostname.YYYYMMDD.hmmz.tar.gz*，其中：
 - *path* 是指向选定目录的路径。
 - *hostname* 是 Oracle HSM 文件系统主机的名称。
 - *YYYYMMDD.hmmz* 是日期和时间戳。

默认情况下，该文件名为 */tmp/*

SAMreport.hostname.YYYYMMDD.hmmz.tar.gz。在示例中，使用目录 */zfs1/tmp/*，其中 */zfs1* 是与 Oracle HSM 文件系统没有共用组件的文件系统（请注意，下面的命令是作为单行输入的一使用反斜杠字符对换行符进行转义）：

```
[server1]root@solaris:~# samexplorer /
/zfs1/sam_config/explorer/samhost1.20140130.1659MST.tar.gz

Report name:      /zfs1/sam_config/explorer/samhost1.20140130.1659MST.tar.gz
Lines per file:  1000
Output format:   tar.gz (default) Use -u for unarchived/uncompressed.

Please wait.....
Please wait.....
Please wait.....

The following files should now be ftp'ed to your support provider
as ftp type binary.

/zfs1/sam_config/explorer/samhost1.20140130.1659MST.tar.gz
```

3. 为文件系统创建恢复点文件。

为文件系统创建恢复点文件

1. 以 *root* 用户身份登录到文件系统服务器主机。

在示例中，元数据服务器主机名为 *server1*：

```
[server1]root@solaris:~#
```

2. 选择将用来存储恢复点文件的位置。所选择的位置不得与您要备份的文件系统共享任何设备，并且必须有空间来存储非常大的文件。

打算删除的设备可能包含尚未归档的文件。因为这样的文件只作为单个副本存在，所以必须创建一个至少存储某些数据和元数据的恢复点文件。这会大大增加恢复点文件的大小。

在示例中，在与 Oracle HSM 文件系统没有共用组件的文件系统 */zfs1* 中创建一个子目录 *tmp/*：

```
[server1]root@solaris:~# mkdir /zfs1/tmp/
[server1]root@solaris:~#
```

3. 转到文件系统的根目录。

在示例中，转到挂载点目录 */hsm/hsmfs1*：

```
[server1]root@solaris:~# cd /hsm/hsmfs1
```

```
[server1]root@solaris:~#
```

4. 备份文件系统元数据和任何未归档的数据。使用命令 `samfsdump -f -u recovery-point`，其中 `recovery-point` 是已完成的恢复点文件的路径和文件名。

请注意，`-u` 选项将未归档文件的数据部分添加到恢复点。这会大大增加该文件的大小。

在示例中，为目录 `/zfs1/tmp/` 中名为 `hsmfs1-20140313.025215` 的 `hsmfs1` 文件系统创建了恢复点文件。我们使用命令 `ls -l` 检查结果（请注意，下面的第二条命令是作为单行输入的一使用反斜杠字符对换行符进行转义）：

```
[server1]root@solaris:~# cd /hsm/hsmfs1
[server1]root@solaris:~# samfsdump -f /
/zfs1/tmp/hsm/hsmfs1-`date +%Y%m%d.%H%M%S` -T /hsm/hsmfs1
samfsdump statistics:
  Files:                10010
  Directories:          2
  Symbolic links:       0
  Resource files:       0
  Files as members of hard links :    0
  Files as first hard link :    0
  File segments:        0
  File archives:        10010
  Damaged files:        0
  Files with data:      0
  File warnings:        0
  Errors:                0
  Unprocessed dirs:    0
  File data bytes:      0
[server1]root@solaris:~# ls -l /zfs1/tmp/hsmfs1*
-rw-r--r-- 1 root other 5376517 Mar 13 02:52 /zfs1/tmp/hsm/hsmfs1-20140313.025215
[server1]root@solaris:~#
```

5. 现在，从已挂载的高性能文件系统中删除设备。

从已挂载的高性能文件系统中删除设备

一次只能删除一个设备。针对每个设备，执行如下操作：

1. 以 `root` 用户身份登录到文件系统服务器主机。

在示例中，元数据服务器主机名为 `server1`：

```
[server1]root@solaris:~#
```

2. 打开 `/etc/opt/SUNWsamfs/mcf` 文件，并记录需要移除的设备的设备序号。

在示例中，使用 `vi` 编辑器。需要从 `hsmfs1` 文件系统的设备列表中删除设备 `/dev/dsk/c1t4d0s7`。设备序号为 `104`：

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment  Equipment  Family    Device    Additional
# Identifier      Ordinal    Type        Set        State     Parameters
#-----
hsmfs1           100        ms          hsmfs1    on
/dev/dsk/c1t3d0s3 101        md          hsmfs1    on
/dev/dsk/c1t4d0s5 102        md          hsmfs1    on
/dev/dsk/c1t3d0s7 103        md          hsmfs1    on
/dev/dsk/c1t4d0s7 104        md          hsmfs1    on
:q
[server1]root@solaris:~#
```

3. 在尝试删除某个设备之前，请确保文件系统剩余设备可以接受必须从您要删除的设备中移走的文件。
 - 确保剩余设备具有足够的容量。
 - 如果设备是一个分散读写组，请确保文件系统包含具有等效配置的另一个分散读写组。

例如，如果您打算删除的分散读写组具有四个设备成员，则您必须具有处于 ON 状态并且有四个设备成员的另一个分散读写组。

4. 确保您打算修改的文件系统具有一个 2A 版超级块。使用命令 `samfsinfo filesystem-name`，其中 `filesystem-name` 是文件系统的名称。

在示例中，文件系统 `hsmfs1` 使用一个 2A 版超级块：

```
[server1]root@solaris:~# /opt/SUNWsamfs/sbin/samfsinfo hsmfs1
samfsinfo: filesystem hsmfs1 is mounted.
name:      hsmfs1      version:      2A
time:      Tuesday, June 28, 2011 6:07:36 AM MDT
feature:   Aligned Maps
count:     4
...
[server1]root@solaris:~#
```

5. 如果文件系统没有 2A 版超级块，请在此处停止。当此文件系统处于已挂载状态时，无法删除设备。
6. 如果要从 Oracle HSM 归档文件系统中删除设备，请从要删除的磁盘设备中释放所有已归档的文件。使用命令 `samcmd release equipment-number`，其中 `equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中标识该设备的设备序号。

如果设备是一个分散读写组，请提供该组中任何设备的设备编号。

Oracle HSM 软件将指定设备的状态更改为 *noalloc*（不分配），以便不在其上存储新文件，并开始释放以前归档的文件。在设备不包含未归档的文件之后，软件将从文件系统配置中删除设备并将其状态更改为 *off*。

在示例中，释放归档文件系统 *hsmfs1* 中设备 *104* 中的文件：

```
[server1]root@solaris:~# samcmd release 104
```

7. 如果要从 Oracle HSM 非归档文件系统中删除设备，请将所有剩余的有效文件从您要删除的磁盘设备中移走。使用命令 *samcmd remove equipment-number*，其中 *equipment-number* 是 */etc/opt/SUNWsamfs/mcf* 文件中标识该设备的设备序号。

Oracle HSM 软件将指定设备的状态更改为 *noalloc*（不分配），以便不在其上存储新文件，并开始将包含有效数据的文件移到文件系统剩余的设备。在移走所有文件后，软件将从文件系统配置中删除设备并将其状态更改为 *off*。

在示例中，将文件从设备 *104* 移走：

```
[server1]root@solaris:~# samcmd remove 104
```

8. 监视选定的进程 *samcmd remove* 或 *samcmd release* 的进度。使用命令 *samcmd m* 并/或查看日志文件和 */var/opt/SUNWsamfs/trace/sam-shrink* 文件。

如果所有文件都已归档，则 *release* 进程可以相当快地完成，因为它只是释放与已复制到归档介质的文件相关的空间。根据数据量和文件数目，*remove* 进程可能会花费相当长的时间，因为它必须在磁盘设备之间移动文件。

```
[server1]root@solaris:~# samcmd m
ty   eq  status      use state ord  capacity    free   ra  part high low
ms   100 m----2----- 27% on          3.691G   2.628G   1M   16  80% 70%
md   101          27% on          0 959.938M  703.188M
md   102          28% on          1 899.938M  646.625M
md   103          13% on          2 959.938M  834.250M
md   104          0% noalloc    3 959.938M  959.938M
[server1]root@solaris:~#
```

9. 如果您使用 *samcmd release* 并且目标设备没有进入 *off* 状态，则表明设备上有未归档的文件。请等待归档程序运行并完成归档。然后再次使用命令 *samcmd release*。可以使用命令 *samcmd a* 检查归档操作的进度。

在未归档的文件归档之前，*release* 进程无法释放磁盘空间。


```
[server1]root@solaris:~# samcmd a
Archiver status samcmd      5.4 14:12:14 Mar  1 2014
samcmd on server1
sam-archiverd:  Waiting for resources
sam-arfind:  hsmfs1  mounted at /hsm/hsmfs1
Files waiting to start      4  schedule      2  archiving      2
[server1]root@solaris:~#
```

10. 如果 `samcmd release` 因为一个或多个未归档的文件无法归档而失败，请将未归档的文件移到另一设备。使用命令 `samcmd remove equipment-number`，正如从非归档独立文件系统删除设备时一样。

在示例中，将文件从设备 104 移走：

```
[server1]root@solaris:~# samcmd remove 104
```

11. 在设备状态变为 `off` 之后，在文本编辑器中打开 `/etc/opt/SUNWsamfs/mcf` 文件，找到该文件系统，并更新设备列表以反映更改。保存文件并关闭编辑器。

在本示例中，`samcmd m` 显示 104 处于 `off` 状态。因此，使用 `vi` 编辑器打开 `mcf` 文件。随后从 `hsmfs1` 文件系统的设备列表中删除设备 104 的条目并保存更改：

```
[server1]root@solaris:~# samcmd m
ty      eq status      use state ord  capacity      free      ra  part high low
ms      100 m----2----- 27% on          3.691G    2.628G    1M   16  80% 70%
md      101          27% on          0  959.938M    703.188M
md      102          28% on          1  899.938M    646.625M
md      103          13% on          2  959.938M    834.250M
md      104          0% off         3  959.938M    959.938M
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment  Equipment  Family      Device      Additional
# Identifier      Ordinal    Type        Set          State        Parameters
#-----
hsmfs1          100      ms          hsmfs1      on
/dev/dsk/c1t3d0s3  101      md          hsmfs1      on
/dev/dsk/c1t4d0s5  102      md          hsmfs1      on
/dev/dsk/c1t3d0s7  103      md          hsmfs1      on
:wq
[server1]root@solaris:~#
```

12. 运行 `sam-fsd` 命令检查修改后的 `mcf` 文件中是否有错误并纠正检测到的任何错误。

如果遇到错误，`sam-fsd` 命令将停止：

```
[server1]root@solaris:~# sam-fsd
```

13. 指示 Oracle HSM 软件重新读取 *mcf* 文件并相应地重新配置自身：

```
[server1]root@solaris:~# samd config
```

14. 在此处停止。

管理 Oracle HSM 共享文件系统

本节概述以下任务：

- [挂载和卸载共享文件系统](#)
- [更改共享文件系统的主机配置](#)
- [从活动元数据服务器切换到潜在元数据服务器](#)
- [将非共享文件系统转换为共享文件系统](#)
- [将共享文件系统转换为非共享文件系统。](#)

挂载和卸载共享文件系统

当您挂载或卸载共享文件系统时，元数据服务器和客户机的安装或卸载顺序非常重要。

为了便于故障转移，元数据服务器和所有潜在元数据服务器上的挂载选项应相同。例如，您可以创建包含挂载选项的 *samfs.cmd* 文件，并将该文件复制到所有主机。

有关挂载共享文件的更多信息，请参见 *mount_samfs* 手册页。

挂载共享文件系统

1. 以 *root* 用户身份登录到 Oracle HSM 元数据服务器和客户机主机。

在示例中，登录到 *sharefs* 文件系统的元数据服务器主机 *sharefs-mds*。然后我们针对每个客户机 (*sharefs-client1* 和 *sharefs-client2*) 打开一个终端窗口。我们使用 *ssh* (Secure Shell, 安全 Shell) 进行登录：

```
[sharefs-mds]root@solaris:~# ssh root@sharefs-client1
```

```
Password:
```

```
[sharefs-client1]root@solaris:~#
```

```
[sharefs-mds]root@solaris:~# ssh root@sharefs-client2
```

```
Password:
```

```
[sharefs-client2]root@solaris:~#
```

2. 如果文件系统在 Solaris `/etc/vfstab` 文件中有一个条目，请使用命令 `mount mountpoint` 在元数据服务器主机上挂载共享文件系统，其中 `mountpoint` 是主机的根文件系统上的挂载点目录。

请始终先在元数据服务器主机上挂载文件系统，然后再在客户机上挂载文件系统。

在示例中，`sharefs` 文件系统在 `/etc/vfstab` 文件中具有以下条目：

```
sharefs - /sharefs samfs - no shared
```

因此，可以通过仅提供挂载点参数来挂载文件系统：

```
[sharefs-mds]root@solaris:~# mount /sharefs
[sharefs-mds]root@solaris:~#
```

3. 如果文件系统在 Solaris `/etc/vfstab` 文件中没有条目，请使用命令 `mount -F samfs -o shared mountpoint` 在元数据服务器主机上挂载共享文件系统，其中 `mountpoint` 是主机的根文件系统上的挂载点目录。

请始终先在元数据服务器主机上挂载文件系统，然后再在客户机上挂载文件系统。

在示例中，`sharefs` 文件系统在 `/etc/vfstab` 文件中没有条目：

```
[sharefs-mds]root@solaris:~# mount -F samfs -o shared /sharefs
[sharefs-mds]root@solaris:~#
```

4. 如果文件系统在 Solaris `/etc/vfstab` 文件中有一个条目，请使用命令 `mount mountpoint` 在每台客户机主机上挂载共享文件系统，其中 `mountpoint` 是主机的根文件系统上的挂载点目录。

您可以按照任意顺序在不同客户机主机上挂载文件系统。

```
[sharefs-client1]root@solaris:~# mount /sharefs
[sharefs-client1]root@solaris:~#
```

```
[sharefs-client2]root@solaris:~# mount /sharefs
[sharefs-client2]root@solaris:~#
```

5. 如果文件系统在 Solaris `/etc/vfstab` 文件中没有条目，请使用命令 `mount -F samfs -o shared mountpoint` 在每个客户机主机上挂载共享文件系统，其中 `mountpoint` 是主机的根文件系统上的挂载点目录。

您可以按照任意顺序在不同客户机主机上挂载文件系统。

```
[sharefs-client1]root@solaris:~# mount -F samfs -o shared /sharefs
```

```
[sharefs-client1]root@solaris:~#  
  
[sharefs-client2]root@solaris:~# mount -F samfs -o shared /sharefs  
[sharefs-client2]root@solaris:~#
```

6. 在此处停止。

卸载共享文件系统

1. 以 *root* 用户身份登录到 Oracle HSM 元数据服务器和客户机主机。

在示例中，登录到 *sharefs* 文件系统的元数据服务器主机 *sharefs-mds*。然后打开每台客户机 (*sharefs-client1* 和 *sharefs-client2*) 的终端窗口，并使用 *ssh* (安全 Shell) 登录：

```
[sharefs-mds]root@solaris:~# ssh root@sharefs-client1  
Password:  
[sharefs-client1]root@solaris:~#  
  
[sharefs-mds]root@solaris:~# ssh root@sharefs-client2  
Password:  
[sharefs-client2]root@solaris:~#
```

2. 如果通过 NFS 或 SAMBA 共享文件系统，请先取消文件系统共享，然后再卸载文件系统。在元数据服务器上，使用命令 *unshare mount-point*，其中 *mount-point* 是 Oracle HSM 文件系统的挂载点目录。

```
[sharefs-mds]root@solaris:~# unshare /sharefs  
[sharefs-mds]root@solaris:~#
```

3. 从每台客户机卸载 Oracle HSM 共享文件系统。使用命令 *umount mount-point*，其中 *mount-point* 是 Oracle HSM 文件系统的挂载点目录。

有关进一步的详细信息，请参见 *umount_samfs* 手册页。在示例中，从两台客户机 (*sharefs-client1* 和 *sharefs-client2*) 中卸载 */sharedqfs1*：

```
[sharefs-client1]root@solaris:~# umount /sharefs  
[sharefs-client1]root@solaris:~# exit  
[sharefs-mds]root@solaris:~#  
  
[sharefs-client2]root@solaris:~# umount /sharefs  
[sharefs-client1]root@solaris:~# exit  
[sharefs-mds]root@solaris:~#
```

4. 从元数据服务器中卸载 Oracle HSM 共享文件系统。使用命令 `umount -o await_clients=interval mount-point`，其中 `mount-point` 是 Oracle HSM 文件系统的挂载点目录，`interval` 是 `-o await_clients` 选项延迟执行的秒数。

在 Oracle HSM 共享文件系统的元数据服务器上发出 `umount` 命令时，`-o await_clients` 选项使 `umount` 等待指定的秒数，以便客户机有时间卸载共享资源。如果您卸载非共享文件系统或者在 Oracle HSM 客户机上发出该命令，则该选项不起作用。有关进一步的详细信息，请参见 `umount_samfs` 手册页。

在示例中，从服务器卸载 `/sharefs` 文件系统，从而允许客户机在 60 秒内卸载：

```
[sharefs-mds]root@solaris:~# umount -o await_clients=60 /sharefs
[sharefs-mds]root@solaris:~#
```

5. 在此处停止。

更改共享文件系统的主机配置

本节提供有关将额外的主机配置为共享文件的客户机和有关取消配置现有客户机的说明。其中包括以下几节：

- [配置额外的文件系统客户机](#)
- [从共享文件系统配置中删除主机](#)
- [为分布式磁带 I/O 配置数据移动器客户机](#)
- [使用永久绑定连接磁带机。](#)

配置额外的文件系统客户机

向共享文件系统添加客户机主机的过程包括三个部分：

- 首先，在共享文件系统配置中添加主机信息。
- 然后，在主机上，使用特定于主机操作系统 Solaris 或 Linux 的过程配置共享文件系统。
- 最后，在主机上使用特定于主机操作系统 Solaris 或 Linux 的过程挂载共享文件系统。

在共享文件系统配置中添加主机信息

1. 以 `root` 用户身份登录到 Oracle HSM 元数据服务器。

在示例中，Oracle HSM 共享文件系统是 `sharefs`，元数据服务器主机是 `sharefs-mds`：

```
[sharefs-mds]root@solaris:~#
```

2. 备份文件 `/etc/opt/SUNWsamfs/hosts.filesystem`，其中 `filesystem` 是要在其中添加客户机主机的文件系统的名称。

请注意，下面的命令是作为单行输入的一使用反斜杠字符对换行符进行转义：

```
[sharefs-mds]root@solaris:~# cp /etc/opt/SUNWsamfs/hosts.sharefs /etc/opt/SUNWsamfs/hosts.sharefs.bak
```

3. 如果共享文件系统已挂载，请从活动元数据服务器运行命令 `samsharefs filesystem`，并将输出重定向到文件 `/etc/opt/SUNWsamfs/hosts.filesystem`，其中 `filesystem` 是您要向其添加客户机主机的文件系统的名称。

`samsharefs` 命令显示 Oracle HSM 共享文件系统的主机配置。将输出重定向到文件会创建一个新的 `hosts` 文件（请注意，下面的命令是作为单行输入的一使用反斜杠字符对换行符进行转义）：

```
[sharefs-mds]root@solaris:~# samsharefs sharedqfs1 > /etc/opt/SUNWsamfs/hosts.sharedqfs1
```

4. 如果共享文件系统未挂载，请从活动或潜在的元数据服务器运行命令 `samsharefs -R filesystem`，并将输出重定向到文件 `/etc/opt/SUNWsamfs/hosts.filesystem`，其中 `filesystem` 是您要向其添加客户机主机的文件系统的名称。

只能从活动元数据服务器或潜在元数据服务器上运行 `samsharefs -R` 命令（有关更多详细信息，请参见 `samsharefs` 手册页）。`samsharefs` 命令显示 Oracle HSM 共享文件系统的主机配置。将输出重定向到文件会创建一个新的 `hosts` 文件。在本示例中，我们从元数据服务器 `sharefs-mds` 运行命令（请注意，下面的命令是作为单行输入的一使用反斜杠字符对换行符进行转义）：

```
[sharefs-mds]root@solaris:~# samsharefs -R sharedqfs1 /> /etc/opt/SUNWsamfs/hosts.sharedqfs1
```

5. 在文本编辑器中打开新创建的 `hosts` 文件。

在示例中，使用 `vi` 编辑器。主机配置包括活动元数据服务器 `sharefs-mds`、一台同时也是潜在元数据服务器的客户机 `sharefs-mds_alt` 以及其他两个客户机 `sharefs-client1` 和 `sharefs-client2`：

```
[sharefs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sharefs
#
#Host Name          Network Interface    Server Ordinal  On/ Off  Additional Parameters
#-----
sharefs-mds         10.79.213.117        1      0      0      server
sharefs-mds_alt     10.79.213.217        2      0
```

```
sharefs-client1    10.79.213.133      0      0
sharefs-client2    10.79.213.47      0      0
```

- 在 `hosts` 文件中，为新的客户机主机添加一行，保存文件，然后关闭编辑器。

在示例中，为主机 `sharefs-client3` 添加一个条目：

```
[sharefs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sharefs
#
#Host Name          Network Interface    Server  On/  Additional
#                   -----            Ordinal Off  Parameters
#-----
sharefs-mds         10.79.213.117        1      0    server
sharefs-mds_alt    10.79.213.217        2      0
sharefs-client1    10.79.213.133        0      0
sharefs-client2    10.79.213.47         0      0
sharefs-client3    10.79.213.49         0      0
:wq
[sharefs-mds]root@solaris:~#
```

- 如果已挂载文件系统，请从活动元数据服务器更新文件系统。使用命令 `samsharefs -u filesystem`，其中 `filesystem` 是要在其中添加客户机主机的文件系统的名称。

`samsharefs` 命令将重新读取修改后的 `hosts` 文件并更新配置：

```
[sharefs-mds]root@solaris:~# samsharefs -u sharefs1
```

- 如果未挂载文件系统，请从活动或潜在元数据服务器更新文件系统。使用命令 `samsharefs -R -u filesystem`，其中 `filesystem` 是要在其中添加客户机主机的文件系统的名称。

`samsharefs` 命令将重新读取修改后的 `hosts` 文件并更新配置：

```
[sharefs-mds]root@solaris:~# samsharefs -R -u sharefs1
```

- 如果要将 Solaris 主机添加为客户机，请转到[“在 Solaris 客户机上配置共享文件系统”](#)。
- 如果要将 Linux 主机添加为客户机，请转到[“在 Linux 客户机主机上配置共享文件系统”](#)。

在 Solaris 客户机上配置共享文件系统

- 在共享文件系统客户机上，以 `root` 用户身份登录。

在示例中，Oracle HSM 共享文件系统是 *sharefs*，客户机主机是 *sharefs-client1*：

```
[sharefs-client1]root@solaris:~#
```

2. 在终端窗口中，检索共享文件的配置信息。使用命令 *samfsconfig device-path*，其中 *device-path* 是命令应当从其开始搜索文件系统磁盘设备的位置（例如 */dev/dsk/** 或 */dev/zvol/dsk/rpool/**）。

```
[sharefs-client1]root@solaris:~# samfsconfig /dev/dsk/*
```

3. 如果主机对文件系统的元数据设备具有访问权限并因此适合用作潜在的元数据服务器，则 *samfsconfig* 输出非常类似于您在文件系统元数据服务器上创建的 *mcf* 文件。

在我们的示例中，主机 *sharefs-client1* 对元数据设备（设备类型为 *mm*）具有访问权限，因此，命令输出将显示服务器 *sharefs-mds* 上的 *mcf* 文件中列出的相同设备。只有主机分配的设备控制器编号不同：

```
[sharefs-client1]root@solaris:~# samfsconfig /dev/dsk/*
# Family Set 'sharefs' Created Thu Feb 21 07:17:00 2013
# Generation 0 Eq count 4 Eq meta count 1
sharefs          300          ma          sharefs  -
/dev/dsk/c1t0d0s0 301          mm          sharefs  -
/dev/dsk/c1t3d0s0 302          mr          sharefs  -
/dev/dsk/c1t3d0s1 303          mr          sharefs  -
```

4. 如果主机对文件系统的元数据设备没有访问权限，则 *samfsconfig* 命令无法找到元数据设备，因此无法将它发现的 Oracle HSM 设备纳入文件系统配置中。命令输出在 *Missing Slices* 下列出了 *Ordinal 0*（元数据设备），未包括标识文件系统系列集的行，并且注释掉了数据设备的列表。

在示例中，主机 *sharefs-client2* 仅对数据设备具有访问权限。因此，*samfsconfig* 输出类似于以下内容：

```
[sharefs-client2]root@solaris:~# samfsconfig /dev/dsk/*
# Family Set 'sharefs' Created Thu Feb 21 07:17:00 2013
# Missing slices
# Ordinal 0
# /dev/dsk/c4t3d0s0 302          mr          sharefs  -
# /dev/dsk/c4t3d0s1 303          mr          sharefs  -
```

5. 从 *samfsconfig* 输出复制共享文件系统的条目。然后，在另一个窗口中，在文本编辑器中打开 */etc/opt/SUNWsamfs/mcf* 文件，并将复制的条目粘贴到文件中。

在第一个示例中，主机 *sharefs-client1* 对文件系统的元数据设备具有访问权限，因此，*mcf* 文件起初类似于以下内容：

```
[sharefs-client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment  Equipment  Family    Device    Additional
# Identifier      Ordinal    Type        Set        State     Parameters
#-----
sharefs          300        ma          sharefs   -         -
/dev/dsk/c1t0d0s0 301        mm          sharefs   -         -
/dev/dsk/c1t3d0s0 302        mr          sharefs   -         -
/dev/dsk/c1t3d0s1 303        mr          sharefs   -         -
```

在第二个示例中，主机 *sharefs-client2* 对文件系统的元数据设备没有访问权限，因此，*mcf* 文件起初类似于以下内容：

```
[sharefs-client2]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment  Equipment  Family    Device    Additional
# Identifier      Ordinal    Type        Set        State     Parameters
#-----
# /dev/dsk/c4t3d0s0 302        mr          sharefs   -         -
# /dev/dsk/c4t3d0s1 303        mr          sharefs   -         -
```

6. 如果主机对文件系统的元数据设备具有访问权限，请将 *shared* 参数添加到共享文件系统条目的 *Additional Parameters* 字段中。

在第一个示例中，主机 *sharefs-client1* 对元数据具有访问权限：

```
[sharefs-client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment  Equipment  Family    Device    Additional
# Identifier      Ordinal    Type        Set        State     Parameters
#-----
sharefs          300        ma          sharefs   -         shared
/dev/dsk/c1t0d0s0 301        mm          sharefs   -         -
/dev/dsk/c1t3d0s0 302        mr          sharefs   -         -
/dev/dsk/c1t3d0s1 303        mr          sharefs   -         -
```

7. 如果主机对文件系统的元数据设备没有访问权限，请为共享文件系统添加一行并包括 *shared* 参数

```
[sharefs-client2]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment  Equipment  Family    Device    Additional
# Identifier      Ordinal    Type        Set        State     Parameters
#-----
sharefs          300        ma          sharefs   -         shared
```

```
# /dev/dsk/c4t3d0s0 302 mr sharefs -
# /dev/dsk/c4t3d0s1 303 mr sharefs -
```

8. 如果主机对文件系统的元数据设备没有访问权限，请为元数据设备添加一行。将 *Equipment Identifier* 字段设置为 *nodev*（无设备），将其余字段设置为它们在元数据服务器上的同一值：

```
[sharefs-client2]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment Equipment Family   Device  Additional
# Identifier      Ordinal   Type     Set     State   Parameters
#-----
sharefs          300      ma      sharefs  on      shared
nodev          301      mm      sharefs  on
# /dev/dsk/c4t3d0s0 302      mr      sharefs  -
# /dev/dsk/c4t3d0s1 303      mr      sharefs  -
```

9. 如果主机对文件系统的元数据设备没有访问权限，请取消注释数据设备的条目。

```
[sharefs-client2]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment Equipment Family   Device  Additional
# Identifier      Ordinal   Type     Set     State   Parameters
#-----
sharefs          300      ma      sharefs  on      shared
nodev           301      mm      sharefs  on
/dev/dsk/c4t3d0s0 302      mr      sharefs  -
/dev/dsk/c4t3d0s1 303      mr      sharefs  -
```

10. 确保所有设备的 *Device State* 字段设置为 *on*，保存 *mcf* 文件并关闭编辑器。

在第一个示例中，主机 *sharefs-client1* 对文件系统的元数据设备具有访问权限，因此，*mcf* 文件最终类似于以下内容：

```
[sharefs-client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment Equipment Family   Device  Additional
# Identifier      Ordinal   Type     Set     State   Parameters
#-----
sharefs          300      ma      sharefs  on      shared
/dev/dsk/c1t0d0s0 301      mm      sharefs  on
/dev/dsk/c1t3d0s0 302      mr      sharefs  on
/dev/dsk/c1t3d0s1 303      mr      sharefs  on
:wq
[sharefs-client1]root@solaris:~#
```

在第二个示例中，主机 *sharefs-client2* 对文件系统的元数据设备没有访问权限，因此，*mcf* 文件最终类似于以下内容：

```
[sharefs-client2]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment  Equipment  Family    Device    Additional
# Identifier      Ordinal    Type        Set        State     Parameters
#-----
sharefs           300        ma          sharefs   on        shared
nodev             301        mm          sharefs   on
/dev/dsk/c4t3d0s0 302        mr          sharefs   on
/dev/dsk/c4t3d0s1 303        mr          sharefs   on
:wq
[sharefs-client2]root@solaris:~#
```

11. 运行 *sam-fsd* 命令检查 *mcf* 文件中是否有错误并更正发现的任何错误。

sam-fsd 是一个读取 Oracle HSM 配置文件的初始化命令。如果遇到错误，该命令将停止。在示例中，检查 *sharefs-client1* 上的 *mcf* 文件：

```
[sharefs-client1]root@solaris:~# sam-fsd
```

12. 接着，在 Solaris 主机上挂载共享文件系统。

在 Solaris 主机上挂载共享文件系统

1. 在共享文件系统主机上，以 *root* 用户身份登录。

在示例中，Oracle HSM 共享文件系统是 *sharefs*，主机是名为 *sharefs-client1* 的客户机：

```
[sharefs-client1]root@solaris:~#
```

2. 备份操作系统的 */etc/vfstab* 文件。

```
[sharefs-client1]root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
```

3. 在文本编辑器中打开 */etc/vfstab* 文件，并为共享文件系统添加一行。

在示例中，在 *vi* 文本编辑器中打开文件并为 *sharefs* 系列集设备添加一行：

```
[sharefs-client1]root@solaris:~# vi /etc/vfstab
#File
#Device  Device  Mount  System  fsck  Mount  Mount
#to Mount to fsck Point  Type   Pass  at Boot Options
```

```
#-----
/devices - /devices devfs - no -
/proc - /proc proc - no -
...
sharefs - /sharefs samfs - no
```

4. 要将文件系统作为共享文件系统挂载到客户机上，请在共享文件系统 *vfstab* 条目的 *Mount Options* 列中输入 *shared* 选项。

如果希望当前客户机将共享文件系统 *sharefs* 挂载为只读的，则按下例中所示编辑 *vfstab* 条目：

```
#File
#Device Device Mount System fsck Mount Mount
#to Mount to fsck Point Type Pass at Boot Options
#-----
/devices - /devices devfs - no -
/proc - /proc proc - no -
...
sharefs - /sharefs samfs - no shared
```

5. 使用逗号作为分隔符添加所需的任何其他挂载选项，并对 */etc/vfstab* 文件进行所需的任何其他更改。然后，保存 */etc/vfstab* 文件。

在示例中，未添加任何额外的挂载选项：

```
#File
#Device Device Mount System fsck Mount Mount
#to Mount to fsck Point Type Pass at Boot Options
#-----
/devices - /devices devfs - no -
/proc - /proc proc - no -
...
sharefs - /sharefs samfs - no shared
:wq
[sharefs-client1]root@solaris:~#
```

6. 创建在 */etc/vfstab* 文件中指定的挂载点，并为挂载点设置访问权限。

挂载点权限必须与元数据服务器上的权限相同，并与所有其他客户机保持一致。用户必须具有执行 (x) 权限才能转到挂载点目录并访问挂载的文件系统中的文件。在示例中，创建 */sharefs* 挂载点目录并将权限设置为 755 (-rwxr-xr-x)：

```
[sharefs-client1]root@solaris:~# mkdir /sharefs
[sharefs-client1]root@solaris:~# chmod 755 /sharefs
```

```
[sharefs-client1]root@solaris:~#
```

7. 挂载共享文件系统：

```
[sharefs-client1]root@solaris:~# mount /sharefs
[sharefs-client1]root@solaris:~#
```

8. 如果要将潜在的元数据服务器主机添加为分布式磁带 I/O 数据移动器，请转到[“为分布式磁带 I/O 配置数据移动器客户机”](#)。
9. 在此处停止。

在 Linux 客户机主机上配置共享文件系统

1. 在 Linux 客户机上，以 *root* 用户身份登录。

在示例中，Oracle HSM 共享文件系统是 *sharefs*，主机是名为 *sharefs-clientL* 的 Linux 客户机：

```
[sharefs-clientL][root@linux ~]#
```

2. 在终端窗口中，使用 *samfsconfig device-path* 命令检索共享文件的配置信息，其中 *device-path* 是命令应当从其开始搜索文件系统磁盘设备的位置（例如 */dev/**）。

因为 Linux 主机对文件系统的元数据设备没有访问权限，*samfsconfig* 命令无法找到元数据设备，因此无法将其发现的 Oracle HSM 设备纳入文件系统配置中。命令输出在 *Missing Slices* 下列出了 *Ordinal 0*（元数据设备），未包括标识文件系统系列集的行，并且注释掉了数据设备的列表。

在示例中，Linux 主机 *sharefs-clientL* 的 *samfsconfig* 输出类似于以下内容：

```
[sharefs-clientL][root@linux ~]# samfsconfig /dev/*
# Family Set 'sharefs' Created Thu Feb 21 07:17:00 2013
#
# Missing slices
# Ordinal 0
# /dev/sda4          302      mr      sharefs  -
# /dev/sda5          303      mr      sharefs  -
```

3. 从 *samfsconfig* 输出复制共享文件的条目。然后，在另一个窗口中，在文本编辑器中打开 */etc/opt/SUNWsamfs/mcf* 文件，并将复制的条目粘贴到文件中。

在示例中，Linux 主机 *sharefs-clientL* 上的 *mcf* 文件起初类似于以下内容：

```
[sharefs-clientL][root@linux ~]# vi /etc/opt/SUNWsamfs/mcf
```

```

# Equipment      Equipment  Equipment  Family    Device    Additional
# Identifier     Ordinal   Type       Set       State     Parameters
#-----
# /dev/sda4      302      mr        sharefs   -         -
# /dev/sda5      303      mr        sharefs   -         -

```

4. 在 *mcf* 文件中，为共享文件系统插入一行并包括 *shared* 参数。

```

# Equipment      Equipment  Equipment  Family    Device    Additional
# Identifier     Ordinal   Type       Set       State     Parameters
#-----
sharefs        300      ma       sharefs   -         shared
# /dev/sda4      302      mr        sharefs   -         -
# /dev/sda5      303      mr        sharefs   -         -

```

5. 在 *mcf* 文件中，为文件系统的元数据设备插入行。因为 Linux 主机对元数据设备没有访问权限，因此将 *Equipment Identifier* 字段设置为 *nodev*（无设备），将其余字段设置为它们在元数据服务器上的同一值：

```

# Equipment      Equipment  Equipment  Family    Device    Additional
# Identifier     Ordinal   Type       Set       State     Parameters
#-----
sharefs          300      ma        sharefs   on        shared
nodev         301      mm       sharefs   on
# /dev/sda4      302      mr        sharefs   -         -
# /dev/sda5      303      mr        sharefs   -         -

```

6. 在 *mcf* 文件中，取消注释 Linux 数据设备的条目。

```

# Equipment      Equipment  Equipment  Family    Device    Additional
# Identifier     Ordinal   Type       Set       State     Parameters
#-----
sharefs          300      ma        sharefs   on        shared
nodev            301      mm        sharefs   on
/dev/sda4      302      mr       sharefs   -         -
/dev/sda5      303      mr       sharefs   -         -

```

7. 确保所有设备的 *Device State* 字段设置为 *on*，保存 *mcf* 文件。

```

# Equipment      Equipment  Equipment  Family    Device    Additional
# Identifier     Ordinal   Type       Set       State     Parameters
#-----
sharefs          300      ma        sharefs   on       shared
nodev            301      mm        sharefs   on
/dev/sda4      302      mr        sharefs   on

```

```

/dev/sda5          303          mr          sharefs      on
:wq
[sharefs-clientL][root@linux ~]#

```

- 运行 `sam-fsd` 命令检查 `mcf` 文件中是否有错误并更正发现的任何错误。

`sam-fsd` 是一个读取 Oracle HSM 配置文件的初始化命令。如果遇到错误，该命令将停止。在示例中，检查 Linux 客户机 `sharefs-clientL` 上的 `mcf` 文件：

```
[sharefs-clientL][root@linux ~]# sam-fsd
```

- 现在，在 Linux 主机上挂载共享文件系统。

在 Linux 客户机主机上挂载共享文件系统

- 在 Linux 客户机上，以 `root` 用户身份登录。

在示例中，Oracle HSM 共享文件系统是 `sharefs`，主机是名为 `sharefs-clientL` 的 Linux 客户机：

```
[sharefs-clientL][root@linux ~]#
```

- 备份操作系统的 `/etc/fstab` 文件。

```
[sharefs-clientL][root@linux ~]# cp /etc/fstab /etc/fstab.backup
```

- 在文本编辑器中打开 `/etc/fstab` 文件，并为共享文件系统添加一行。

在示例中，使用 `vi` 文本编辑器为 `sharefs` 系列集设备添加一行：

```

[sharefs-clientL][root@linux ~]# vi /etc/fstab
#File
#Device    Mount      System    Mount          Dump      Pass
#to Mount  Point     Type      Options        Frequency Number
#-----  -
...
/proc      /proc      proc      defaults
sharefs    /sharefs   samfs

```

- 在文件的第四列中，添加必需的 `shared` 挂载选项。

```

[sharefs-clientL][root@linux ~]# vi /etc/fstab
#File
#Device    Mount      System    Mount          Dump      Pass
#to Mount  Point     Type      Options        Frequency Number

```

```
#-----
...
/proc      /proc      proc      defaults
sharefs    /sharefs   samfs     shared
```

5. 在文件的第四列中，使用逗号作为分隔符添加所需的任何其他挂载选项。

Linux 客户机支持以下其他挂载选项：

- *rw*、*ro*
- *retry*
- *meta_timeo*
- *rdlease*、*wrlease*、*aplease*
- *minallocsz*、*maxallocsz*
- *noauto*、*auto*

在示例中，添加选项 *noauto*：

```
#File
#Device    Mount      System    Mount                               Dump      Pass
#to Mount  Point      Type      Options                             Frequency Number
#-----
...
/proc      /proc      proc      defaults
sharefs    /sharefs   samfs     shared, noauto
```

6. 在文件的其余两列中输入零 (0)。然后，保存 */etc/fstab* 文件。

```
#File
#Device    Mount      System    Mount                               Dump      Pass
#to Mount  Point      Type      Options                             Frequency Number
#-----
...
/proc      /proc      proc      defaults
sharefs    /sharefs   samfs     shared, noauto                       0          0
:wq
[sharefs-clientL][root@linux ~]#
```

7. 创建 */etc/fstab* 文件中指定的挂载点，并为挂载点设置访问权限。

挂载点权限必须与元数据服务器上的权限相同，并与所有其他客户机保持一致。用户必须具有执行 (x) 权限才能转到挂载点目录并访问挂载的文件系统中的文件。在示例中，创建 */sharefs* 挂载点目录并将权限设置为 755 (*-rwxr-xr-x*)：

```
[sharefs-clientL][root@linux ~]# mkdir /sharefs
```



```
[sharefs-clientL][root@linux ~]# chmod 755 /sharefs
```

- 挂载共享文件系统。使用命令 `mount mountpoint`，其中 `mountpoint` 是在 `/etc/fstab` 文件中指定的挂载点目录。

如示例所示，`mount` 命令生成一个警告。这是正常的并且可以忽略：

```
[sharefs-clientL][root@linux ~]# mount /sharefs
Warning: loading SUNWqfs will taint the kernel: SMI license
See http://www.tux.org/lkml/#export-tainted for information
about tainted modules. Module SUNWqfs loaded with warnings
```

- 在此处停止。

从共享文件系统配置中删除主机

要从共享文件中删除主机，只需要将其从服务器配置中删除，如下文所述（要完全取消配置主机，请卸载软件和配置文件）：

从文件系统 `Hosts` 文件中删除主机

- 以 `root` 用户身份登录到 Oracle HSM 元数据服务器。

在示例中，Oracle HSM 共享文件系统是 `sharefs`，元数据服务器主机是 `sharefs-mds`：

```
[sharefs-mds]root@solaris:~#
```

- 以 `root` 用户身份登录到每台客户机，并卸载共享文件系统。

请注意，潜在的元数据服务器本身也是客户机。在示例中，有三台客户机：`sharefs-client1`、`sharefs-client2` 和潜在元数据服务器 `sharefs-mds_alt`。对于每台客户机，使用 `ssh` 登录，卸载文件系统 `sharefs`，然后关闭 `ssh` 会话：

```
[sharefs-mds]root@solaris:~# ssh root@sharefs-client1
Password:
[sharefs-client1]root@solaris:~# umount sharefs
[sharefs-client1]root@solaris:~# exit
[sharefs-mds]root@solaris:~# ssh root@sharefs-client2
Password:
[sharefs-client2]root@solaris:~# umount sharefs
[sharefs-client2]root@solaris:~# exit
[sharefs-mds]root@solaris:~# ssh root@sharefs-mds_alt
Password:
```

```
[sharefs-mds_alt]root@solaris:~# umount sharefs
root@solaris:~# exit
[sharefs-mds]root@solaris:~#
```

3. 在元数据服务器上，卸载共享文件系统。

```
[sharefs-mds]root@solaris:~# umount sharefs
```

4. 在元数据服务器上，将文件 `/etc/opt/SUNWsamfs/hosts.filesystem` 重命名为 `/etc/opt/SUNWsamfs/hosts.filesystem.bak`，其中 `filesystem` 是您要从中删除客户机主机的文件系统的名称。

请注意，下面的命令是作为单行输入的一使用反斜杠字符对换行符进行转义：

```
[sharefs-mds]root@solaris:~# mv /etc/opt/SUNWsamfs/hosts.sharefs /
/etc/opt/SUNWsamfs/hosts.sharefs.bak
```

5. 将当前的共享文件系统主机配置捕获到一个文件。在元数据服务器上，运行命令 `samsharefs -R filesystem`，将输出重定向到文件 `/etc/opt/SUNWsamfs/hosts.filesystem`，其中 `filesystem` 是您要向其添加客户机主机的文件系统的名称。

`samsharefs` 命令显示指定的 Oracle HSM 共享文件系统的主机配置。将输出重定向到文件会创建一个新的 `hosts` 文件。在示例中，从元数据服务器 `sharefs-mds` 运行命令：

```
[sharefs-mds]root@solaris:~# samsharefs -R sharedqfs1 > /
/etc/opt/SUNWsamfs/hosts.sharedqfs1
```

6. 在文本编辑器中打开新创建的 `hosts` 文件。

在示例中，使用 `vi` 编辑器。此处需要删除客户机 `sharefs-client3`：

```
[sharefs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sharefs
#
#Host Name          Network Interface      Server  On/  Additional
#Ordinal Off  Parameters
#-----
sharefs-mds         10.79.213.117          1      0    server
sharefs-mds_alt     10.79.213.217          2      0
sharefs-client1     10.79.213.133          0      0
sharefs-client2     10.79.213.47           0      0
sharefs-client3     10.79.213.49           0      0
```

7. 在 `hosts` 文件中，删除与您需要删除的客户机主机对应的行。然后保存文件并关闭编辑器。

在示例中，删除主机 *sharefs-client3* 的条目：

```
[sharefs-mds]root@solaris:~# vi /etc/opt/SUNwsamfs/hosts.sharefs
#
#Host Name          Network Interface      Server  On/  Additional
#Ordinal            Off  Parameters
#-----
sharefs-mds         10.79.213.117          1      0    server
sharefs-mds_alt     10.79.213.217          2      0
sharefs-client1     10.79.213.133          0      0
sharefs-client2     10.79.213.47           0      0
:wq
[sharefs-mds]root@solaris:~#
```

8. 使用修改后的 *hosts* 文件更新文件系统。在元数据服务器中，使用命令 *samsharefs -R -u filesystem*，其中 *filesystem* 是要从其中移除客户机主机的文件系统的名称。

```
[sharefs-mds]root@solaris:~# samsharefs -u sharefs
```

9. 在元数据服务器主机上，挂载共享文件系统。

在示例中，*/etc/vfstab* 文件包含 *sharefs* 文件系统的条目，因此使用简单的挂载语法（有关完整信息，请参见 *mount_samfs* 手册页）：

```
[sharefs-mds]root@solaris:~# mount sharefs
```

10. 在每台客户机主机上，挂载共享文件系统。

请注意，潜在的元数据服务器本身也是客户机。在示例中，有三台客户机：*sharefs-client1*、*sharefs-client2* 和潜在元数据服务器 *sharefs-mds_alt*。对于每台客户机，使用 *ssh* 登录，卸载文件系统 *sharefs*，然后关闭 *ssh* 会话：

```
[sharefs-mds]root@solaris:~# ssh root@sharefs-mds_alt
Password:
[sharefs-mds_alt]root@solaris:~# mount sharefs
sharefs-mds_alt]root@solaris:~# exit
[sharefs-mds]root@solaris:~# ssh root@sharefs-client1
Password:
[sharefs-client1]root@solaris:~# mount sharefs
sharefs-client1]root@solaris:~# exit
[sharefs-mds]root@solaris:~# ssh root@sharefs-client2
Password:
[sharefs-client2]root@solaris:~# mount sharefs
```

```
sharefs-client2]root@solaris:~# exit  
[sharefs-mds]root@solaris:~#
```

11. 在此处停止。

为分布式磁带 I/O 配置数据移动器客户机

从 Oracle HSM 发行版 6.1 开始，在 Solaris 11 或更高版本上运行的共享归档文件系统的任何客户机都可以连接磁带机并代表文件系统执行磁带 I/O。将磁带 I/O 分布在这些数据移动器主机上可以大大降低服务器开销，提高文件系统性能，并且在扩展 Oracle HSM 实施时可以更灵活。当归档需求增加时，现在可以选择用更强大的系统替换 Oracle HSM 元数据服务器（垂直扩展）或者将负载分布在更多的客户机上（水平扩展）。

配置数据移动器客户机

要为客户机配置分布式磁带 I/O，请执行如下操作：

1. 将要用于分布式 I/O 的所有设备连接到客户机。
2. 如果您尚未“[使用永久绑定连接磁带机](#)”，请执行该过程。然后返回此处。
3. 以 *root* 用户身份登录到共享的归档文件系统的元数据服务器。

在示例中，主机名是 *samsharefs-mds*：

```
[samsharefs-mds]root@solaris:~#
```

4. 确保元数据服务器运行的是 Oracle HSM Solaris 11 或更高版本。

```
[samsharefs-mds]root@solaris:~# uname -r  
5.11  
[samsharefs-mds]root@solaris:~#
```

5. 确保用作数据移动器的所有客户机运行的是 Oracle HSM Solaris 11 或更高版本。

在示例中，打开每台客户机主机 *samsharefs-client1* 和 *samsharefs-client2* 的终端窗口，然后使用 *ssh* 远程登录。登录标题将显示 Solaris 版本：

```
[samsharefs-mds]root@solaris:~# ssh root@samsharefs-client1  
...  
Oracle Corporation      SunOS 5.11      11.1      September 2013  
[samsharefs-client1]root@solaris:~#
```

```
[samsharefs-mds]root@solaris:~# ssh root@samsharefs-client2  
...  
Oracle Corporation      SunOS 5.11      11.1      September 2013
```

```
[samsharefs-client2]root@solaris:~#
```

- 在元数据服务器上，在文本编辑器中打开文件 `/etc/opt/SUNWsamfs/defaults.conf`，通过取消注释行 `distio =` 并将值设置为 `on` 来启用分布式 I/O。

默认情况下，`distio` 处于 `off`（禁用）状态。

在示例中，在 `vi` 编辑器中打开文件并添加行：

```
[samsharefs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults. To change the default behavior, uncomment the
# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
...
distio = on
```

- 接下来，标识应当参与分布式 I/O 的设备类型。要将设备类型 `dev` 用于分布式 I/O，请将 `dev_distio = on` 行添加到 `defaults.conf` 文件。要将设备类型 `dev` 从分布式 I/O 中排除，请添加 `dev_distio = off` 行。保存文件并关闭编辑器。

默认情况下，允许 Oracle HSM T10000 磁带机和 LTO 磁带机参与分布式 I/O (`ti_distio = on` 且 `li_distio = on`)，并排除所有其他类型。在示例中，排除 LTO 驱动器：

```
[samsharefs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults. To change the default behavior, uncomment the
# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
...
distio = on
li_distio = off
:wq
[samsharefs-mds]root@solaris:~#
```

- 在用作数据移动器的每台客户机上，编辑 `defaults.conf` 文件，使其与服务器上的该文件匹配。

在示例中，使用 `vi` 编辑客户机 `samsharefs-client1` 上的 `defaults.conf` 文件，保存该文件，然后关闭编辑器：

```
[samsharefs-mds]root@solaris:~# ssh root@samsharefs-client1
Password:
[samsharefs-client1]root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults. To change the default behavior, uncomment the
```

```

# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
...
distio = on
li_distio = off
:wq
[samsharefs-client1]root@solaris:~#
[samsharefs-mds]root@solaris:~#

```

9. 在用作数据移动器的每台客户机上，在文本编辑器中打开 `/etc/opt/SUNWsamfs/mcf` 文件。添加元数据服务器用于分布式磁带 I/O 的所有磁带设备。确保设备顺序和设备编号与其在元数据服务器上的 `mcf` 文件中的对应项完全相同。

在示例中，使用 `vi` 编辑客户机 `samsharefs-client1` 上的 `mcf` 文件：

```

[samsharefs-client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment          Equipment Equipment Family      Device Additional
# Identifier          Ordinal   Type      Set        State  Parameters
#-----
samsharefs           800      ms        samsharefs on
...
# Archival storage for copies:
/dev/rmt/60cbn       901      ti        on
/dev/rmt/61cbn       902      ti        on
/dev/rmt/62cbn       903      ti        on
/dev/rmt/63cbn       904      ti        on

```

10. 如果在用作数据移动器的客户机上配置了元数据服务器上的 `/etc/opt/SUNWsamfs/mcf` 文件中列出的磁带库，请将磁带库系列集指定为用于分布式磁带 I/O 的磁带设备的系列集名称。保存文件。

在示例中，在主机上配置了磁带库，因此为磁带设备使用系列集名称 `library1`：

```

[samsharefs-client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment          Equipment Equipment Family      Device Additional
# Identifier          Ordinal   Type      Set        State  Parameters
#-----
samsharefs           800      ms        samsharefs on
...
# Archival storage for copies:
/dev/scsi/changer/c1t0d5 900      rb        library1   on
/dev/rmt/60cbn       901      ti        library1   on
/dev/rmt/61cbn       902      ti        library1   on
/dev/rmt/62cbn       903      ti        library1   on
/dev/rmt/63cbn       904      ti        library1   on

```

```
:wq
[samsharefs-client1]root@solaris:~#
```

11. 如果在用作数据移动器的客户机上未配置元数据服务器上的 `/etc/opt/SUNWsamfs/mcf` 文件中列出的磁带库，请使用连字符 (-) 作为用于分布式磁带 I/O 的磁带设备的系列集名称。

在示例中，未在主机上配置磁带库：

```
[samsharefs-client2]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment          Equipment Equipment Family      Device Additional
# Identifier          Ordinal   Type      Set          State  Parameters
#-----
samsharefs           800      ms        samsharefs  on
...
# Archival storage for copies:
/dev/rmt/60cbn       901      ti        -            on
/dev/rmt/61cbn       902      ti        -            on
/dev/rmt/62cbn       903      ti        -            on
/dev/rmt/63cbn       904      ti        -            on
:wq
[samsharefs-client2]root@solaris:~#
```

12. 如果您需要为特定的归档集副本启用或禁用分布式磁带 I/O，请在文本编辑器中打开服务器的 `/etc/opt/SUNWsamfs/archiver.cmd` 文件并向 `copy` 指令添加 `-distio` 参数。设置 `-distio on` 以启用或设置 `-distio off` 以禁用分布式 I/O。保存文件并关闭编辑器。

在示例中，使用 `vi` 编辑器为副本 1 关闭分布式 I/O，为副本 2 打开分布式 I/O：

```
[samsharefs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
# archiver.cmd
# Generated by config api Mon Nov 22 14:31:39 2013
...
#
# Copy Parameters Directives
params
allsets -sort path -offline_copy stageahead
allsets.1 -startage 10m -startsize 500M -startcount 500000 -distio off
allsets.2 -startage 24h -startsize 20G -startcount 500000 -distio on
:wq
[samsharefs-mds]root@solaris:~#
```

13. 在各主机上，运行 `sam-fsd` 命令检查 `mcf` 文件中是否有错误并纠正发现的任何错误。

`sam-fsd` 是一个读取 Oracle HSM 配置文件的初始化命令。如果遇到错误，该命令将停止。在示例中，检查 Linux 客户机 `sharefs-clientL` 上的 `mcf` 文件：

```
[sharefs-clientL][root@linux ~]# sam-fsd
```

14. 在服务器上，指示 Oracle HSM 软件读取修改后的配置文件，并相应地配置自身。使用命令 `samd config` 并纠正发现的任何错误。

在示例中，在服务器 `sharefs-mds` 上运行 `samd config` 命令：

```
[samsharefs-mds]root@solaris:~# samd config
```

15. 在此处停止。

使用永久绑定连接磁带机

如果添加的主机要用作潜在元数据服务器或分布式 I/O 数据移动器客户机，您必须使用永久绑定配置可移除介质设备。Solaris 操作系统按照启动时发现设备的顺序将驱动器连接到系统设备。该顺序可能会也可能不会反映其他文件系统主机发现设备的顺序或设备在磁带库中的物理安装顺序。因此，您需要使用与将设备绑定到其他主机相同的方式将其绑定到新主机，顺序与其在可移除介质库中的安装顺序相同。

下面的过程概述了必需的步骤（有关完整信息，请参见 `devfsadm` 和 `devlinks` 手册页以及适用于您的 Solaris 操作系统版本的管理文档）：

- 如果您在磁带库中移动、添加或删除了驱动器，或者更换或重新配置了与归档 Oracle HSM 共享文件系统关联的磁带库，请更新永久绑定以反映所做的更改。
- 如果要将新的元数据服务器或数据移动器客户机添加到归档 Oracle HSM 共享文件系统，请将新的文件系统主机永久绑定到可移除介质设备。

更新永久绑定以反映对硬件配置的更改

1. 以 `root` 用户身份登录到活动元数据服务器主机。

```
[sharefs-mds]root@solaris:~#
```

2. 按“[确定驱动器在库中的安装顺序](#)”中所述创建一个新的驱动器映射文件。

在示例中，`device-mappings.txt` 文件类似于以下内容：

```
[sharefs-mds]root@solaris:~# vi /root/device-mappings.txt
LIBRARY SOLARIS          SOLARIS
DEVICE  LOGICAL            PHYSICAL
NUMBER  DEVICE                DEVICE
```



```

-----
2 /dev/rmt/0cbn -> ../../devices/pci@8.../st@w500104f00093c438,0:cbn
1 /dev/rmt/1cbn -> ../../devices/pci@8.../st@w500104f0008120fe,0:cbn
3 /dev/rmt/2cbn -> ../../devices/pci@8.../st@w500104f000c086e1,0:cbn
4 /dev/rmt/3cbn -> ../../devices/pci@8.../st@w500104f000b6d98d,0:cbn

```

3. 在文本编辑器中打开 `/etc/devlink.tab` 文件。

在示例中，使用 `vi` 编辑器：

```

[sharefs-mds]root@solaris:~# vi /etc/devlink.tab
# Copyright (c) 1993, 2011, Oracle and/or its affiliates. All rights reserved.
# This is the table used by devlinks
# Each entry should have 2 fields; but may have 3. Fields are separated
# by single tab ('t') characters.
...

```

4. 将 `device-mappings.txt` 文件用作指南，将 Solaris 磁带设备树中的起始节点重新映射到磁带库中的第一个驱动器。在 `/etc/devlink.tab` 文件中，添加 `type=ddi_byte:tape; addr=device_address,0; rmt/node-number/M0` 格式的行，其中 `device_address` 是设备的物理地址，`node-number` 是 Solaris 设备树中的一个位置，该位置高到足以避免与 Solaris 自动配置（Solaris 从 0 节点启动）的任何设备冲突。

在示例中，记下磁带库中的第一个设备 `1` 的设备地址 `w500104f0008120fe`，可以看到该设备当前连接到位于 `rmt/1` 的主机：

```

[sharefs-mds] vi /root/device-mappings.txt
LIBRARY SOLARIS          SOLARIS
DEVICE  LOGICAL           PHYSICAL
NUMBER  DEVICE              DEVICE
-----
2 /dev/rmt/0cbn -> ../../devices/pci@8.../st@w500104f00093c438,0:cbn
1 /dev/rmt/1cbn -> ../../devices/pci@8.../st@w500104f0008120fe,0:cbn
3 /dev/rmt/2cbn -> ../../devices/pci@8.../st@w500104f000c086e1,0:cbn
4 /dev/rmt/3cbn -> ../../devices/pci@8.../st@w500104f000b6d98d,0:cbn

```

因此，在 `/etc/devlink.tab` 中创建一行来将 `rmt/60` 重新映射到磁带库中编号为 `1` 的驱动器 `w500104f0008120fe`：

```

[sharefs-mds]root@solaris:~# vi /etc/devlink.tab
# Copyright (c) 1993, 2011, Oracle and/or its affiliates. All rights reserved.
...
type=ddi_byte:tape;addr=w500104f0008120fe,0;      rmt/60/M0

```

```
:w
```

5. 继续在 `/etc/devlink.tab` 文件中为分配用于 Oracle HSM 归档的每个磁带设备添加行，使元数据服务器上的设备树中的驱动器顺序与磁带库中的安装顺序匹配。保存文件并关闭编辑器。

在示例中，记下剩余的三个设备（位于 `w500104f00093c438` 的磁带库驱动器 2、位于 `w500104f000c086e1` 的磁带库驱动器 3 和位于 `w500104f000c086e1` 的磁带库驱动器 4）的顺序和地址：

```
[sharefs-mds]root@solaris:~# vi /root/device-mappings.txt
...
 2  /dev/rmt/0cbn -> ../../devices/pci@8/.../st@w500104f00093c438,0:cbn
 1  /dev/rmt/1cbn -> ../../devices/pci@8/.../st@w500104f0008120fe,0:cbn
 3  /dev/rmt/2cbn -> ../../devices/pci@8/.../st@w500104f000c086e1,0:cbn
 4  /dev/rmt/3cbn -> ../../devices/pci@8/.../st@w500104f000b6d98d,0:cbn
```

然后，将设备地址映射到接下来的三个 Solaris 设备节点，保持与磁带库中相同的顺序：

```
[sharefs-mds]root@solaris:~# vi /etc/devlink.tab
...
type=ddi_byte:tape;addr=w500104f0008120fe,0;    rmt/60/M0
type=ddi_byte:tape;addr=w500104f00093c438,0;    rmt/61/M0
type=ddi_byte:tape;addr=w500104f000c086e1,0;    rmt/62/M0
type=ddi_byte:tape;addr=w500104f000b6d98d,0;    rmt/63/M0
:wq
[sharefs-mds]root@solaris:~#
```

6. 删除 `/dev/rmt` 中指向磁带设备的所有现有链接。

```
[sharefs-mds]root@solaris:~# rm /dev/rmt/*
```

7. 根据 `/etc/devlink.tab` 文件中的条目创建新的永久磁带设备链接。使用命令 `devfsadm -c tape`。

`devfsadm` 命令每次运行时，都会使用 `/etc/devlink.tab` 文件指定的配置为该文件中指定的设备创建新的磁带设备链接。`-c tape` 选项将命令限制为仅为磁带类设备创建新链接：

```
[sharefs-mds]root@solaris:~# devfsadm -c tape
```

- 对共享文件系统配置中的每个潜在元数据服务器和数据移动器重复该操作。在每种情况下，将相同的行添加到 `/etc/devlink.tab` 文件中，删除 `/dev/rmt` 中的链接，然后运行 `devfsadm -c tape`。

在示例中，使用 `ssh` 依次登录到每个主机，然后配置相同的四个逻辑设备 `rmt/60/M0`、`rmt/61/M0`、`rmt/62/M0` 和 `rmt/63/M0`：

```
[sharefs-mds]root@solaris:~# ssh root@sharefs-mds_alt
Password:
[sharefs-mds_alt]root@solaris:~# vi /etc/devlink.tab
...
type=ddi_byte:tape;addr=w500104f0008120fe,0;    rmt/60/M0
type=ddi_byte:tape;addr=w500104f00093c438,0;    rmt/61/M0
type=ddi_byte:tape;addr=w500104f000c086e1,0;    rmt/62/M0
type=ddi_byte:tape;addr=w500104f000b6d98d,0;    rmt/63/M0
:wq
[sharefs-mds_alt]root@solaris:~# rm /dev/rmt/*
[sharefs-mds_alt]root@solaris:~# devfsadm -c tape
[sharefs-mds_alt]root@solaris:~# exit
sharefs-mds]root@solaris:~# ssh root@sharefs-client1
Password:
[sharefs-client1]root@solaris:~# vi /etc/devlink.tab
...
type=ddi_byte:tape;addr=w500104f0008120fe,0;    rmt/60/M0
type=ddi_byte:tape;addr=w500104f00093c438,0;    rmt/61/M0
type=ddi_byte:tape;addr=w500104f000c086e1,0;    rmt/62/M0
type=ddi_byte:tape;addr=w500104f000b6d98d,0;    rmt/63/M0
:wq
[sharefs-client1]root@solaris:~# rm /dev/rmt/*
[sharefs-client1]root@solaris:~# devfsadm -c tape
[sharefs-client1]root@solaris:~# exit
[sharefs-mds]root@solaris:~#
```

- 返回到“为分布式磁带 I/O 配置数据移动器客户机”或“配置额外的文件系统客户机”。

将新的文件系统主机永久绑定到可移除介质设备

- 以 `root` 用户身份登录主机。

```
[sharefs-mds]root@solaris:~#
```

- 如果介质库中驱动器的物理顺序在配置现有的文件系统主机后发生了更改，请按“确定驱动器在库中的安装顺序”中所述创建新的映射文件。

在示例中，*device-mappings.txt* 文件类似于以下内容：

```
[sharefs-mds]root@solaris:~# vi /root/device-mappings.txt
LIBRARY SOLARIS          SOLARIS
DEVICE  LOGICAL          PHYSICAL
NUMBER  DEVICE            DEVICE
-----
2    /dev/rmt/0cbn -> ../../devices/pci@8.../st@w500104f00093c438,0:cbn
1    /dev/rmt/1cbn -> ../../devices/pci@8.../st@w500104f0008120fe,0:cbn
3    /dev/rmt/2cbn -> ../../devices/pci@8.../st@w500104f000c086e1,0:cbn
4    /dev/rmt/3cbn -> ../../devices/pci@8.../st@w500104f000b6d98d,0:cbn
```

3. 在文本编辑器中打开 */etc/devlink.tab* 文件。

在示例中，使用 *vi* 编辑器：

```
[sharefs-mds]root@solaris:~# vi /etc/devlink.tab
# Copyright (c) 1993, 2011, Oracle and/or its affiliates. All rights reserved.
# This is the table used by devlinks
# Each entry should have 2 fields; but may have 3.  Fields are separated
# by single tab ('/t') characters.
...
```

4. 使用 *device-mappings.txt* 文件作为指南，将 Solaris 磁带设备树中的起始节点 *rmt/node-number* 重新映射到磁带库中的第一个驱动器。向 */etc/devlink.tab* 文件添加一行，格式为 *type=ddi_byte:tape; addr=device_address,0; rmt/node-number/M0*，其中：*device_address* 是设备的物理地址，*node-number* 是设备在 Solaris 设备树中的位置。选择一个足够大的节点编号以避免与 Solaris 自动配置的任何设备冲突（Solaris 从节点 0 开始）。

在示例中，记下磁带库中的第一个设备 1 的设备地址 *w500104f0008120fe*，可以看到该设备当前连接到位于 *rmt/1* 的主机：

```
[sharefs-mds] vi /root/device-mappings.txt
LIBRARY SOLARIS          SOLARIS
DEVICE  LOGICAL          PHYSICAL
NUMBER  DEVICE            DEVICE
-----
2    /dev/rmt/0cbn -> ../../devices/pci@8.../st@w500104f00093c438,0:cbn
1    /dev/rmt/1cbn -> ../../devices/pci@8.../st@w500104f0008120fe,0:cbn
3    /dev/rmt/2cbn -> ../../devices/pci@8.../st@w500104f000c086e1,0:cbn
4    /dev/rmt/3cbn -> ../../devices/pci@8.../st@w500104f000b6d98d,0:cbn
```

因此，在 `/etc/devlink.tab` 中创建一行来将 `rmt/60` 重新映射到磁带库中编号为 1 的驱动器 `w500104f0008120fe`：

```
[sharefs-mds]root@solaris:~# vi /etc/devlink.tab
# Copyright (c) 1993, 2011, Oracle and/or its affiliates. All rights reserved.
...
type=ddi_byte:tape;addr=w500104f0008120fe,0;    rmt/60/M0
:w
```

5. 继续在 `/etc/devlink.tab` 文件中为分配用于 Oracle HSM 归档的每个磁带设备添加行，使元数据服务器上的设备树中的驱动器顺序与磁带库中的安装顺序匹配。保存文件。

在示例中，记下剩余的三个设备（位于 `w500104f00093c438` 的磁带库驱动器 2、位于 `w500104f000c086e1` 的磁带库驱动器 3 和位于 `w500104f000c086e1` 的磁带库驱动器 4）的顺序和地址：

```
[sharefs-mds]root@solaris:~# vi /root/device-mappings.txt
...
2    /dev/rmt/0cbn -> ../../devices/pci@8.../st@w500104f00093c438,0:cbn
1    /dev/rmt/1cbn -> ../../devices/pci@8.../st@w500104f0008120fe,0:cbn
3    /dev/rmt/2cbn -> ../../devices/pci@8.../st@w500104f000c086e1,0:cbn
4    /dev/rmt/3cbn -> ../../devices/pci@8.../st@w500104f000b6d98d,0:cbn
```

然后，将设备地址映射到接下来的三个 Solaris 设备节点，保持与磁带库中相同的顺序：

```
[sharefs-mds]root@solaris:~# vi /etc/devlink.tab
...
type=ddi_byte:tape;addr=w500104f0008120fe,0;    rmt/60/M0
type=ddi_byte:tape;addr=w500104f00093c438,0;    rmt/61/M0
type=ddi_byte:tape;addr=w500104f000c086e1,0;    rmt/62/M0
type=ddi_byte:tape;addr=w500104f000b6d98d,0;    rmt/63/M0
:wq
[sharefs-mds]root@solaris:~#
```

6. 删除 `/dev/rmt` 中指向磁带设备的所有现有链接。

```
[sharefs-mds]root@solaris:~# rm /dev/rmt/*
```

7. 根据 `/etc/devlink.tab` 文件中的条目创建新的永久磁带设备链接。使用命令 `devfsadm -c tape`。

`devfsadm` 命令每次运行时，都会使用 `/etc/devlink.tab` 文件指定的配置为该文件中指定的设备创建新的磁带设备链接。`-c tape` 选项将命令限制为仅为磁带类设备创建新链接：

```
[sharefs-mds]root@solaris:~# devfsadm -c tape
```

8. 在共享文件系统配置中的每台潜在元数据服务器和数据移动器上，向 `/etc/devlink.tab` 文件中添加相同的行，删除 `/dev/rmt` 中的链接，然后运行 `devfsadm -c tape`。

在示例中，使用 `ssh` 登录到潜在元数据服务器主机 `sharefs-mds_alt` 和客户机主机 `sharefs-client1`。然后配置相同的四个逻辑设备 `rmt/60/M0`、`rmt/61/M0`、`rmt/62/M0` 和 `rmt/63/M0`，在每台客户机上：

```
[sharefs-mds]root@solaris:~# ssh root@sharefs-mds_alt
Password:
[sharefs-mds_alt]root@solaris:~# vi /etc/devlink.tab
...
type=ddi_byte:tape;addr=w500104f0008120fe,0;      rmt/60/M0
type=ddi_byte:tape;addr=w500104f00093c438,0;      rmt/61/M0
type=ddi_byte:tape;addr=w500104f000c086e1,0;      rmt/62/M0
type=ddi_byte:tape;addr=w500104f000b6d98d,0;      rmt/63/M0
:wq
[sharefs-mds_alt]root@solaris:~# rm /dev/rmt/*
[sharefs-mds_alt]root@solaris:~# devfsadm -c tape
[sharefs-mds_alt]root@solaris:~# exit
[sharefs-mds]root@solaris:~# ssh root@sharefs-client1
Password:
[sharefs-client1]root@solaris:~# vi /etc/devlink.tab
...
type=ddi_byte:tape;addr=w500104f0008120fe,0;      rmt/60/M0
type=ddi_byte:tape;addr=w500104f00093c438,0;      rmt/61/M0
type=ddi_byte:tape;addr=w500104f000c086e1,0;      rmt/62/M0
type=ddi_byte:tape;addr=w500104f000b6d98d,0;      rmt/63/M0
:wq
[sharefs-client1]root@solaris:~# rm /dev/rmt/*
[sharefs-client1]root@solaris:~# devfsadm -c tape
[sharefs-client1]root@solaris:~# exit
[sharefs-mds]root@solaris:~#
```

9. 返回到“为分布式磁带 I/O 配置数据移动器客户机”或“配置额外的文件系统客户机”。

从活动元数据服务器切换到潜在元数据服务器

本节中的过程将文件系统的元数据服务从当前主机（活动元数据服务器）移动到备用主机（潜在元数据服务器）。使用哪个过程取决于您要替换的服务器主机的运行状况：

- [激活潜在元数据服务器以替换有故障的活动元数据服务器](#)
- [激活潜在元数据服务器以替换运行正常的活动元数据服务器](#)

激活潜在元数据服务器以替换有故障的活动元数据服务器

此过程允许您将元数据服务从已停止工作的活动元数据服务器主机移走。并激活潜在元数据服务器，即使文件系统仍处于已挂载状态。执行如下操作：

注意：

在已停止、已禁用有故障的元数据服务器或已断开其连接之前，不要激活潜在元数据服务器！

要在文件系统处于已挂载状态且活动元数据服务器停机时激活潜在服务器，需要调用带 `-R` 选项的 `samsharefs` 命令，这将对原始设备而不是文件系统接口进行操作。因此，如果您在有故障的服务器仍然连接到设备时激活潜在元数据服务器，则有故障的服务器可能会损坏文件系统。

1. 如果活动元数据服务器有故障，请确保在您执行任何其他操作之前它无法访问元数据设备。关闭受影响主机的电源，或者将有故障的主机断开与元数据设备的连接。
2. 至少等到最大租约到期，以便所有客户机的读取、写入和附加租约都到期。
3. 以 `root` 用户身份登录到潜在元数据服务器。

在示例中，登录到潜在元数据服务器 `sharefs-mds_alt`：

```
[sharefs-mds_alt]root@solaris:~#
```

4. 激活潜在元数据服务器。从潜在元数据服务器上发出命令 `samsharefs -R -s server file-system`，其中 `server` 是潜在元数据服务器的主机名，`file-system` 是 Oracle HSM 共享文件系统的名称。

在示例中，潜在元数据服务器是 `sharefs-mds_alt`，文件系统名称是 `sharefs`：

```
[sharefs-mds_alt]root@solaris:~# samsharefs -R -s sharefs-mds_alt sharefs
```

5. 如果需要检查文件系统的完整性并修复可能的问题，现在请使用[“卸载共享文件系统”](#)过程卸载文件系统。
6. 如果已卸载了文件系统，请执行文件系统检查。使用命令 `samfsck -F file-system`，其中 `-F` 指定修复错误，`file-system` 是文件系统的名称。

在示例中，检查并修复名为 `sharefs` 的文件系统：

```
[sharefs-mds_alt]root@solaris:~# samfsck -F sharefs
```

7. 在此处停止。

激活潜在元数据服务器以替换运行正常的活动元数据服务器

必要时，您可以将元数据服务从运行正常的活动元数据服务器主机移走，并将其移到新激活的潜在元数据服务器上。例如，在升级或更换原始服务器主机或其某些组件时，您可以将元数据服务转移到备用主机以使文件系统保持可用。执行如下操作：

1. 以 *root* 用户身份登录到活动元数据服务器和潜在元数据服务器。

在示例中，登录到活动元数据服务器 *sharefs-mds*。然后，在另一个终端窗口中，使用安全 shell (secure shell, *ssh*) 登录到潜在元数据服务器 *sharefs-mds_alt*：

```
[sharefs-mds]root@solaris:~#
```

```
[sharefs-mds]root@solaris:~# ssh root@sharefs-mds_alt
```

```
Password:
```

```
[sharefs-mds-alt]root@solaris:~#
```

2. 如果活动元数据服务器挂载了 Oracle HSM 归档文件系统，请完成处于活动状态的归档和回写作业并停止任何新活动，然后再继续操作。请参见“[使归档和回写进程闲置](#)”。
3. 如果活动元数据服务器挂载了 Oracle HSM 归档文件系统，请使可移除介质驱动器闲置并停止磁带库控制守护进程。请参见“[停止归档和回写进程](#)”。
4. 如果您使用 *crontab* 条目来运行回收程序进程，请删除该条目并确保回收程序当前未在运行。
5. 激活潜在元数据服务器。从潜在元数据服务器上发出命令 *samsharefs -s server file-system*，其中 *server* 是潜在元数据服务器的主机名，*file-system* 是 Oracle HSM 共享文件系统的名称。

在示例中，潜在元数据服务器是 *sharefs-mds_alt*，文件系统名称是 *sharefs*：

```
[sharefs-mds_alt]root@solaris:~# samsharefs -s sharefs-mds_alt sharefs
```

6. 在潜在元数据服务器上加载配置文件并启动 Oracle HSM 进程。使用命令 *samd config*。

对于归档共享文件系统，*samd config* 命令会重新启动归档进程和磁带库控制守护进程。但是，对于正在等待文件从磁带回写到主磁盘高速缓存的共享文件系统客户机，必须重新发出回写请求。

7. 如果您仍然需要使用 *crontab* 条目来运行回收程序进程，请恢复该条目。
8. 在此处停止。

将非共享文件系统转换为共享文件系统

要将非共享文件系统转换为共享文件系统，请执行以下任务：

- 在活动元数据服务器和潜在元数据服务器上创建 [Hosts 文件](#)
- [共享非共享的文件系统并配置客户机](#)

在活动元数据服务器和潜在元数据服务器上创建 Hosts 文件

在每台元数据服务器上，您必须创建一个 hosts 文件，在其中列出共享文件系统的服务器和客户机的网络地址信息。hosts 文件与 *mcf* 文件一起存储在 */etc/opt/SUNwsamfs/* 目录中。在共享文件系统的初始创建过程中，*sammkfs -S* 命令使用该文件中存储的设置配置共享。因此，现在请使用下面的过程创建该文件。

1. 收集将作为客户机共享文件系统的主机的网络主机名和 IP 地址。

在下面的示例中，客户机 *hsmfs1-mds_alt*（潜在元数据服务器）、*hsmfs1-client1* 和 *hsmfs1-client2* 将共享 *hsmfs1* 文件系统。

2. 以 *root* 用户身份登录到元数据服务器。

在示例中，登录到主机 *hsmfs1-mds*：

```
[hsmfs1-mds]root@solaris:~#
```

3. 使用文本编辑器在元数据服务器上创建文件 */etc/opt/SUNwsamfs/hosts.family-set-name*，将 *family-set-name* 替换为您打算共享的文件系统的系列集名称。

在示例中，使用 *vi* 文本编辑器创建文件 *hosts.hsmfs1*。此处添加了一些可选的标题，使每个行以井号 (#) 开头，表示是注释：

```
[hsmfs1-mds]root@solaris:~# vi /etc/opt/SUNwsamfs/hosts.hsmfs1
# /etc/opt/SUNwsamfs/hosts.hsmfs1
#
# Host Name                Network Interface      Server  On/  Additional
#                          Ordinal  Off  Parameters
#-----
```

4. 在第一列中输入元数据服务器的主机名，在第二列中输入相应的 IP 地址或域名。用空格字符分隔各列。

在示例中，输入元数据服务器的主机名和 IP 地址，分别为 *hsmfs1-mds* 和 *10.79.213.117*：

```
[hsmfs1-mds]root@solaris:~# vi /etc/opt/SUNwsamfs/hosts.hsmfs1
# /etc/opt/SUNwsamfs/hosts.hsmfs1
```

```

#
#Host Name          Network Interface      Server  On/  Additional
#-----
#-----
#-----
#-----
#-----
hsmfs1-mds          10.79.213.117

```

- 添加第三列，用空格字符将其与网络地址隔开。在此列中，输入服务器的序号（1 表示活动元数据服务器、2 表示第一台潜在元数据服务器，依此类推）。

在此示例中，只有一台元数据服务器，因此输入 1：

```

[hsmfs1-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.hsmfs1
# /etc/opt/SUNWsamfs/hosts.hsmfs1
#
#Host Name          Network Interface      Server  On/  Additional
#-----
#-----
#-----
#-----
#-----
hsmfs1-mds          10.79.213.117          1

```

- 添加第四列，用空格字符将其与服务器序号隔开。在此列中，输入 0（零）。

第四列中的值为 0、-（连字符）或空表示该主机处于 *on* 状态—配置为对共享文件系统有访问权限。1（数字一）表示该主机处于 *off* 状态—配置为对文件系统没有访问权限（有关在管理共享文件系统时使用这些值的信息，请参见 *samhsmfs1* 手册页）。

```

[hsmfs1-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.hsmfs1
# /etc/opt/SUNWsamfs/hosts.hsmfs1
#
#Host Name          Network Interface      Server  On/  Additional
#-----
#-----
#-----
#-----
#-----
hsmfs1-mds          10.79.213.117          1      0

```

- 添加第五列，用空格字符将其与开启/关闭状态列隔开。在此列中，输入关键字 *server* 以指示该主机是当前处于活动状态的元数据服务器：

```

[hsmfs1-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.hsmfs1
# /etc/opt/SUNWsamfs/hosts.hsmfs1
#
#Host Name          Network Interface      Server  On/  Additional
#-----
#-----
#-----
#-----
#-----
hsmfs1-mds          10.79.213.117          1      0      server

```

- 如果您计划包括一台或多台主机作为潜在元数据服务器，请为每台主机创建一个条目。每次递增服务器序号。但不要包括 *server* 关键字（每个文件系统上只能有一个活动元数据服务器）。

在示例中，主机 *hsmfs1-mds_alt* 是服务器序号为 2 的一台潜在元数据服务器。除非将其激活为元数据服务器，否则它将作为客户机：

```
[hsmfs1-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.hsmfs1
# /etc/opt/SUNWsamfs/hosts.hsmfs1
#
#Host Name          Network Interface      Server  On/  Additional
#                   |                      Ordinal Off  Parameters
#-----|-----|-----|---|-----
hsmfs1-mds          10.79.213.117          1      0    server
hsmfs1-mds_alt      10.79.213.217          2      0
```

9. 为每台客户机主机添加一行，每行中的服务器序号值都为 0。

服务器序号为 0 表示该主机是一台客户机。在示例中，添加两台客户机 *hsmfs1-client1* 和 *hsmfs1-client2*。

```
[hsmfs1-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.hsmfs1
# /etc/opt/SUNWsamfs/hosts.hsmfs1
#
#Host Name          Network Interface      Server  On/  Additional
#                   |                      Ordinal Off  Parameters
#-----|-----|-----|---|-----
hsmfs1-mds          10.79.213.17           1      0    server
hsmfs1-mds_alt      10.79.213.7            2      0
hsmfs1-client1      10.79.213.33           0      0
hsmfs1-client2      10.79.213.47           0      0
```

10. 保存 */etc/opt/SUNWsamfs/hosts.family-set-name* 文件并退出编辑器。

在示例中，保存对 */etc/opt/SUNWsamfs/hosts.hsmfs1* 的更改并退出 *vi* 编辑器：

```
[hsmfs1-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.hsmfs1
# /etc/opt/SUNWsamfs/hosts.hsmfs1
#
#Host Name          Network Interface      Server  On/  Additional
#                   |                      Ordinal Off  Parameters
#-----|-----|-----|---|-----
hsmfs1-mds          10.79.213.117          1      0    server
hsmfs1-mds          10.79.213.117          1      0    server
hsmfs1-mds_alt      10.79.213.217          2      0
hsmfs1-client1      10.79.213.133          0      0
hsmfs1-client2      10.79.213.147          0      0
:wq
[hsmfs1-mds]root@solaris:~#
```

11. 在共享文件系统配置中包括的所有潜在元数据服务器上放置 `/etc/opt/SUNWsamfs/hosts.family-set-name` 新文件的一个副本。

在示例中，在主机 `hsmfs1-mds_alt` 上放置一个副本：

```
[hsmfs1-mds]root@solaris:~# sftp root@hsmfs1-mds_alt
Password:
sftp> cd /etc/opt/SUNWsamfs/
sftp> put /etc/opt/SUNWsamfs/hosts.hsmfs1
sftp> bye
[hsmfs1-mds]root@solaris:~#
```

12. 现在，[共享非共享的文件系统并配置客户机](#)。

共享非共享的文件系统并配置客户机

1. 以 `root` 用户身份登录到元数据服务器。

在示例中，登录到主机 `hsmfs1-mds`：

```
[hsmfs1-mds]root@solaris:~#
```

2. 如果您没有系统文件和配置文件的当前备份副本，现在请创建副本。请参见[“备份 Oracle HSM 配置”](#)。
3. 如果您没有当前的文件系统恢复点文件和归档日志的最新副本，现在请创建它们。请参见[“备份文件系统”](#)。

如果您在初始配置过程中为文件系统设置了自动化备份流程，可能不需要额外的备份。

4. 如果您在转换归档文件系统，请完成处于活动状态的归档和回写作业并停止任何新活动，然后再继续操作。请参见[“使归档和回写进程闲置”](#)和[“停止归档和回写进程”](#)。
5. 卸载文件系统。使用命令 `umount family-set-name`，其中 `family-set-name` 是您打算共享的文件系统的系列集名称。

有关挂载和卸载 Oracle HSM 文件系统的更多信息，请参见 `mount_samfs` 手册页。在示例中，卸载 `hsmfs1` 文件系统：

```
[hsmfs1-mds]root@solaris:~# umount hsmfs1
[hsmfs1-mds]root@solaris:~#
```

6. 将文件系统转换为 Oracle HSM 共享文件系统。使用命令 `samfsck -S -F file-system-name`，其中 `file-system-name` 是文件系统的系列集名称。

在示例中，转换名为 `hsmfs1` 的文件系统：

```
[hsmfs1-mds]root@solaris:~# samfsck -S -F hsmfs1
```

7. 在文本编辑器中打开 `/etc/opt/SUNWsamfs/mcf` 文件，并找到该文件系统对应的行。

在示例中，使用 `vi` 编辑器：

```
[hsmfs1-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment  Equipment  Family   Device   Additional
# Identifier      Ordinal    Type        Set      State    Parameters
#-----
hsmfs1           200        ma          hsmfs1  on
/dev/dsk/c0t0d0s0 201        mm          hsmfs1  on
/dev/dsk/c0t3d0s0 202        md          hsmfs1  on
/dev/dsk/c0t3d0s1 203        md          hsmfs1  on
```

8. 在 `mcf` 文件中，向文件系统条目的最后一列中的 "Additional Parameters" 字段中添加 `shared` 参数。然后保存文件并关闭编辑器。

```
[hsmfs1-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment  Equipment  Family   Device   Additional
# Identifier      Ordinal    Type        Set      State    Parameters
#-----
hsmfs1           200        ma          hsmfs1  on       shared
/dev/dsk/c0t0d0s0 201        mm          hsmfs1  on
/dev/dsk/c0t3d0s0 202        md          hsmfs1  on
/dev/dsk/c0t3d0s1 203        md          hsmfs1  on
:wq
[hsmfs1-mds]root@solaris:~#
```

9. 在文本编辑器中打开 `/etc/vfstab` 文件，并找到该文件系统对应的行。

在示例中，使用 `vi` 编辑器：

```
[hsmfs1-mds]root@solaris:~# vi /etc/vfstab
#File
#Device  Device  Mount  System  fsck  Mount  Mount
#to Mount to fsck Point  Type  Pass  at Boot Options
#-----
/devices -      /devices  devfs  -      no    -
/proc   -      /proc     proc   -      no    -
...
hsmfs1  -      /hsm/hsmfs1  samfs  -      yes
```

10. 在 `/etc/vfstab` 文件中，向文件系统条目的最后一列中的 "Mount Options" 字段中添加 `shared` 挂载参数。然后保存文件并关闭编辑器。

```
[hsmfs1-mds]root@solaris:~# vi /etc/vfstab
#File
#Device      Device      Mount      System  fsck  Mount      Mount
#to Mount    to fsck    Point      Type    Pass  at Boot    Options
#-----
/devices     -          /devices  devfs   -     no         -
/proc        -          /proc     proc    -     no         -
...
hsmfs1      -          /hsm/hsmfs1  samfs   -     yes        shared
:wq
[hsmfs1-mds]root@solaris:~#
```

11. 初始化共享文件和主机配置。使用命令 `samsharefs -u -R family-set-name`，其中 `family-set-name` 是文件系统的系列集名称。

```
[hsmfs1-mds]root@solaris:~# samsharefs -u -R hsmfs1
```

12. 指示 Oracle HSM 软件重新读取 `mcf` 文件并相应地重新配置自身：

```
[hsmfs1-mds]root@solaris:~# samd config
```

13. 在元数据服务器上挂载共享文件系统。

```
[hsmfs1-mds]root@solaris:~# mount /hsm/hsmfs1
```

14. 如果您的主机配置了多个网络接口，请参见["使用本地 Hosts 文件来路由网络通信"](#)。
15. 使用["配置额外的文件系统客户机"](#)中概述的过程向刚刚共享的文件系统添加必需的客户机。

使用本地 Hosts 文件来路由网络通信

单个主机不需要本地 `hosts` 文件。元数据服务器上文件系统的全局文件标识了所有文件系统主机的活动元数据服务器以及活动元数据服务器和潜在元数据服务器的网络接口（请参见["在活动元数据服务器和潜在元数据服务器上创建 Hosts 文件"](#)）。但是，当您需要对具有多个网络接口的文件系统主机之间的网络通信路由进行选择时，本地 `hosts` 文件会很有用。

每个文件系统主机都会识别其他主机的网络接口，方法是先检查元数据服务器上的 `/etc/opt/SUNwsamfs/hosts.family-set-name` 文件，其中 `family-set-name` 是 `/etc/opt/SUNwsamfs/mcf` 文件中指定的文件系统系列的名称。然后，主机检查自己的特定 `/etc/opt/SUNwsamfs/hosts.family-set-name.local` 文件。如果没有本地

hosts 文件，则主机将按照全局 hosts 文件中指定的顺序使用该全局文件中指定的接口地址。但是，如果存在本地 hosts 文件，则主机会将其与全局文件进行比较并按照本地文件中指定的顺序仅使用两个文件中同时列出的那些接口。通过在每个文件中使用不同的地址，您可以控制不同主机使用的接口。

要配置本地 hosts 文件，请使用下面概述的过程：

1. 在元数据服务器主机和每台潜在元数据服务器主机上，创建全局 hosts 文件 `/etc/opt/SUNwsamfs/hosts.family-set-name` 的一个副本，如“在活动元数据服务器和潜在元数据服务器上创建 Hosts 文件”中所述。

对于本节中的示例，共享文件系统 `sharefs2` 包括一台活动元数据服务器 `sharefs2-mds` 和一台潜在元数据服务器 `sharefs2-mds_alt`，每台都具有两个网络接口。还有两台客户机 `sharefs2-client1` 和 `sharefs2-client2`。

此处需要活动元数据服务器和潜在元数据服务器彼此之间通过专用网络地址进行通信，与客户机的通信通过域名服务 (Domain Name Service, DNS) 可以解析为公共局域网 (local area network, LAN) 地址的主机名进行。因此，文件系统的全局 hosts 文件 `/etc/opt/SUNwsamfs/hosts.sharefs2` 在活动服务器和潜在服务器条目的 `Network Interface` 字段中指定了一个专用网络地址，并为每台客户机的接口地址指定了一个主机名：该文件如下所示：

```
# /etc/opt/SUNwsamfs/hosts.sharefs2
#
#Host Name          Network Interface      Server  On/  Additional
#-----          -----
#                   -----
sharefs2-mds        172.16.0.129           1      0    server
sharefs2-mds_alt    172.16.0.130           2      0
sharefs2-client1    sharefs2-client1       0      0
sharefs2-client2    sharefs2-client2       0      0
```

2. 使用路径和文件名 `/etc/opt/SUNwsamfs/hosts.family-set-name.local` 在每台活动元数据服务器和潜在元数据服务器上创建一个本地 hosts 文件，其中 `family-set-name` 是为 `/etc/opt/SUNwsamfs/mcf` 文件中的共享文件系统指定的名称。请仅包括您希望活动服务器和潜在服务器使用的网络的接口。

在示例中，需要活动元数据服务器和潜在元数据服务器彼此之间通过专用网络进行通信，因此，每台服务器的本地 hosts 文件 `hosts.sharefs2.local` 针对活动服务器和潜在服务器仅列出了专用地址：

```
[sharefs2-mds]root@solaris:~# vi /etc/opt/SUNwsamfs/hosts.sharefs2.local
# /etc/opt/SUNwsamfs/hosts.sharefs2
#
#Host Name          Network Interface      Server  On/  Additional
#-----          -----
#                   -----
sharefs2-mds        172.16.0.129           1      0    server
```

```

sharefs2-mds_alt      172.16.0.130          2          0
:wq
[sharefs2-mds]root@solaris:~# ssh root@sharefs2-mds_alt
Password:

[sharefs2-mds_alt]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sharefs2.local
# /etc/opt/SUNWsamfs/hosts.sharefs2
#
#Host Name           Network Interface      Server  On/  Additional
#Ordinal Off  Parameters
#-----
sharefs2-mds         172.16.0.129           1       0    server
sharefs2-mds_alt     172.16.0.130           2       0
:wq
[sharefs2-mds_alt]root@solaris:~# exit
[sharefs2-mds]root@solaris:~#

```

- 使用路径和文件名 `/etc/opt/SUNWsamfs/hosts.family-set-name.local` 在每个客户机上创建一个本地 hosts 文件，其中 `family-set-name` 是为 `/etc/opt/SUNWsamfs/mcf` 文件中的共享文件系统指定的名称。请仅包括您希望客户机使用的网络的接口。

在示例中，需要客户机仅通过公共网络与服务器进行通信。因此，该文件仅包括活动元数据服务器和潜在元数据服务器的主机名：

```

[sharefs2-mds]root@solaris:~# ssh root@sharefs2-client1
Password:
[sharefs2-client1]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sharefs2.local
# /etc/opt/SUNWsamfs/hosts.sharefs2
#
#Host Name           Network Interface      Server  On/  Additional
#Ordinal Off  Parameters
#-----
sharefs2-mds         sharefs2-mds           1       0    server
sharefs2-mds_alt     sharefs2-mds_alt       2       0
:wq
[sharefs2-client1]root@solaris:~# exit
[sharefs2-mds]root@solaris:~# ssh root@sharefs2-client2
Password:

[sharefs2-client2]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sharefs2.local
# /etc/opt/SUNWsamfs/hosts.sharefs2
#
#Host Name           Network Interface      Server  On/  Additional
#Ordinal Off  Parameters
#-----
sharefs2-mds         sharefs2-mds           1       0    server
sharefs2-mds_alt     sharefs2-mds_alt       2       0

```



```
:wq
[sharefs2-client2]root@solaris:~# exit
[sharefs2-mds]root@solaris:~#
```

4. 如果您在完成服务器配置的同时启动了此过程，请添加客户机。转至[“配置额外的文件系统客户机”](#)。

将共享文件系统转换为非共享文件系统

当需要取消文件系统共享时，请执行如下操作：

将共享元数据服务器转换为非共享系统

1. 以 *root* 用户身份登录到元数据服务器。

在示例中，登录到主机 *hsmfs1-mds*：

```
[hsmfs1-mds]root@solaris:~#
```

2. 使用[“从文件系统 Hosts 文件中删除主机”](#)过程从元数据服务器配置中删除客户机。
3. 如果您没有系统文件和配置文件的当前备份副本，现在请创建副本。请参见[“备份 Oracle HSM 配置”](#)。
4. 如果您没有当前的文件系统恢复点文件和归档日志的最新副本，现在请创建它们。请参见[“备份文件系统”](#)。

如果您在初始配置过程中为文件系统设置了自动化备份流程，可能不需要额外的备份。

5. 如果您在转换归档文件系统，请完成处于活动状态的归档和回写作业并停止任何新活动，然后再继续操作。请参见[“使归档和回写进程闲置”](#)和[“停止归档和回写进程”](#)。
6. 卸载文件系统。使用命令 `umount family-set-name`，其中 *family-set-name* 是 `/etc/opt/SUNWsamfs/mcf` 文件中共享文件系统的系列集名称。

有关挂载和卸载 Oracle HSM 文件系统的更多信息，请参见 *mount_samfs* 手册页。在示例中，卸载 *hsmfs1* 文件系统：

```
[hsmfs1-mds]root@solaris:~# umount hsmfs1
```

7. 将 Oracle HSM 共享文件系统转换为非共享文件系统。使用命令 `samfsck -F -U file-system-name`，其中 *file-system-name* 是为 `/etc/opt/SUNWsamfs/mcf` 文件中共享文件系统指定的名称。

在示例中，转换名为 *hsmfs1* 的文件系统：

```
[hsmfs1-mds]root@solaris:~# samfsck -F -U hsmfs1
```

8. 在文本编辑器中打开 `/etc/opt/SUNWsamfs/mcf` 文件，并找到该文件系统对应的行。

在示例中，使用 `vi` 编辑器：

```
[hsmfs1-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment  Equipment  Family  Device  Additional
# Identifier      Ordinal   Type       Set     State   Parameters
#-----
hsmfs1          200      ma        hsmfs1 on     shared
/dev/dsk/c0t0d0s0  201      mm         hsmfs1  on
/dev/dsk/c0t3d0s0  202      md         hsmfs1  on
/dev/dsk/c0t3d0s1  203      md         hsmfs1  on
```

9. 在 `mcf` 文件中，从文件系统条目的最后一列的 "Additional Parameters" 字段中删除 `shared` 参数。然后保存文件并关闭编辑器。

```
[hsmfs1-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment  Equipment  Family  Device  Additional
# Identifier      Ordinal   Type       Set     State   Parameters
#-----
hsmfs1          200      ma        hsmfs1 on
/dev/dsk/c0t0d0s0  201      mm         hsmfs1  on
/dev/dsk/c0t3d0s0  202      md         hsmfs1  on
/dev/dsk/c0t3d0s1  203      md         hsmfs1  on
:wq
[hsmfs1-mds]root@solaris:~#
```

10. 在文本编辑器中打开 `/etc/vfstab` 文件，并找到该文件系统对应的行。

在示例中，使用 `vi` 编辑器：

```
[hsmfs1-mds]root@solaris:~# vi /etc/vfstab
#File
#Device  Device  Mount  System  fsck  Mount  Mount
#to Mount to fsck Point  Type  Pass  at Boot Options
#-----
/devices -      /devices  devfs  -    no    -
/proc   -      /proc     proc   -    no    -
...
hsmfs1 -      /hsm/hsmfs1  samfs  -    yes   shared
```

11. 在 `/etc/vfstab` 文件中，从文件系统条目的最后一列中的 "Mount Options" 字段中删除 `shared` 挂载选项。然后保存文件并关闭编辑器。

在示例中，使用 `vi` 编辑器：

```
[hsmfs1-mds]root@solaris:~# vi /etc/vfstab
#File
#Device   Device   Mount   System  fsck  Mount   Mount
#to Mount  to fsck  Point   Type    Pass  at Boot  Options
#-----  -----  -----  -----  ----  -----  -----
/devices  -        /devices devfs   -     no      -
/proc     -        /proc   proc    -     no      -
...
hsmfs1   -        /hsm/hsmfs1 samfs   -     yes
:wq
[hsmfs1-mds]root@solaris:~#
```

12. 删除文件 `/etc/opt/SUNWsamfs/hosts.file-system-name`。
13. 指示 Oracle HSM 软件重新读取 `mcf` 文件并相应地重新配置自身：

```
[hsmfs1-mds]root@solaris:~# samd config
```

14. 挂载文件系统。

```
[hsmfs1]root@solaris:~# mount /hsm/hsmfs1
```

15. 在此处停止。

第 4 章 管理文件和目录

本章介绍以下主题：

- [设置 Oracle HSM 文件属性](#)
- [使用扩展文件属性](#)
- [适应大型文件](#)
- [使用 Linear Tape File System \(LTFS\) 卷](#)
- [管理 SMB/CIFS 共享中的目录和文件](#)
- [管理访问控制列表。](#)

设置 Oracle HSM 文件属性

用户可以通过熟悉的“标准 UNIX 文件系统”界面进行交互是 Oracle Hierarchical Storage Manager and StorageTek QFS Software 的一项重要优势。大多数用户甚至不需要知道差异。但是，在必要时，Oracle HSM 文件系统可以向高级用户提供更强大的功能。Oracle HSM 文件属性允许用户优化文件系统处理单个文件和目录的行为。如果用户了解自己的工作负荷以及数据特性，则可以极大地提高各个文件的性能。例如，用户可以根据给定文件或目录中数据的特性指定直接 I/O 或缓冲 I/O。用户可以预分配文件系统空间，以便大型文件能以更加连续的方式写入，并可在写入特定文件或目录时指定分散读写宽度。

`setfa` 命令可对新的和现有的文件和目录设置这些文件属性。该命令可以创建不存在的指定文件或目录。当应用于目录时，该命令可对该目录中的所有文件和子目录设置指定属性。后续创建的文件和目录会继承这些属性。

下面概述了一些基本任务（有关其他信息，请参见 `setfa` 手册页）。

- [恢复默认文件属性值](#)
- [预分配文件系统空间](#)
- [为文件指定循环分配或分散读写分配](#)
- [在指定的分散读写组设备上分配文件存储。](#)

恢复默认文件属性值

1. 登录到文件系统主机。

```
user@solaris:~#
```

2. 要将文件重置为默认属性值，请使用命令 `setfa -d file`，其中 `file` 是文件的路径和名称。

在该示例中，将文件 `/samfs1/data/2014/03/series3.15` 重置为默认值：

```
user@solaris:~# setfa -d /samfs1/data/2014/03/series3.15
```

3. 要以递归方式将某个目录及其所有内容重置为默认值，请使用命令 `setfa -r directory`，其中 `directory` 是目录的路径和名称。

在该示例中，将子目录 `/samfs1/data/2014/02` 重置为默认值：

```
user@solaris:~# setfa -r /samfs1/data/2014/02/
```

4. 在此处停止。

预分配文件系统空间

为文件预分配空间可保证在写入文件时有足够的空间按顺序写出整个文件。以连续块形式读写大型文件可以减少与寻找块以及缓冲较小、较分散的数据块有关的开销，从而提高效率和整体性能。因此，预分配最适合写入数量可预测的较大数据块。文件关闭时，预分配但未使用的空间仍属于文件，在删除整个文件之前，不会释放此空间用于其他用途。

1. 登录到文件系统主机。

```
user@solaris:~#
```

2. 如果需要预分配空间来写入已包含数据的现有文件，请使用命令 `setfa -L number-bytes file`，其中 `k` 是一个整数或整数加 `k`（表示千字节）、`m`（表示兆字节）或 `g`（表示千兆字节），而 `file` 则是文件的名称。

命令 `setfa -L` 使用标准分配。该命令支持分散读写。预分配的文件可以增长至超过其预分配大小。在该示例中，将预分配 121 MB 给现有的文件 `tests/series119b`：

```
user@solaris:~# setfa -L 121m tests/series119b
```

3. 如果需要预分配空间以写入未分配存储块的新文件，请使用命令 `setfa -l number-bytes file`，其中：
 - `l` 是小写字母 "l"。
 - `number-bytes` 是整数或整数加 `k`（表示千字节）、`m`（表示兆字节）或 `g`（表示千兆字节）。
 - `file` 是文件的名称。

命令 `setfa -l` 可预分配指定的字节数。结果文件的大小固定在为其预分配的大小，不能高出也不能小于其预分配大小。在该示例中，将创建文件 `data/2014/a3168445`，并为其内容预分配 2 GB 的空间：

```
user@solaris:~# setfa -l 2g data/2014/a3168445
```

4. 在此处停止。

为文件指定循环分配或分散读写分配

默认情况下，Oracle HSM 文件系统使用在挂载时为文件系统指定的分配方法。但是，用户可以为指定的目录或文件指定一个首选分配方法：循环分配或者具有指定分散读写宽度的分散读写分配。

1. 登录到文件系统主机。

```
user@solaris:~#
```

2. 要指定循环分配，请将分散读写宽度指定为 0（零）。使用命令 `setfa -s 0 directory-or-file`，其中 `directory-or-file` 是要使用指定的分配方法写入的目录或文件的名称。

分散读写宽度为 0（零）时，指定无分散读写的循环分配。文件系统在下一个可用设备上开始写入文件。文件系统在同一个设备上为文件写入连续的磁盘分配单元 (disk allocation unit, DAU)，直到文件完成或者设备空间耗尽。如果设备空间耗尽，文件系统将移到下一个可用设备，然后继续写入磁盘分配单元。该过程会重复到文件完成。在该示例中，为写入到 `data/field-reports` 目录的所有文件指定循环分配：

```
user@solaris:~# setfa -s 0 data/field-reports
```

3. 要指定分散读写分配，请指定分散读写宽度。使用命令 `setfa -s stripe-width directory-or-file`，其中 `stripe-width` 为 [1-255] 范围内的一个整数，`directory-or-file` 是要使用指定分配方法写入的目录或文件的名称。

如果分散读写宽度在 [1-255] 范围内，即指定分散读写分配。文件系统以并行方式向多个设备写入分散读写宽度中指定的磁盘分配单元 (DAU) 数量，直到文件完成。在该示例中，为写入到相应目录中的所有文件指定分散读写宽度为 1 的分散读写分配（即写入到 `data/field-reports` 目录 `data/2014/` 的所有文件的文件分配），这样文件系统将向每个可用设备写入一个磁盘分配单元，直到文件完成：

```
user@solaris:~# setfa -s 1 data/2014/
```

4. 在此处停止。

在指定的分散读写组设备上分配文件存储

用户可以指定循环分配或分散读写分配应从哪个分散读写组设备开始。Oracle HSM 分散读写组是在多个物理卷上分散读写数据的逻辑卷。使用循环文件分配时，会在指定的分散读写组上写入整个文件。使用分散读写分配时，会在指定的分散读写组上进行第一次分配。

1. 登录到文件系统主机。

```
user@solaris:~#
```

2. 要将整个文件写入特定的分散读写组，请使用循环分配。使用命令 `setfa -s 0 -g stripe-group-number`，其中 `stripe-group-number` 为 `[0-127]` 范围内的一个整数，用于标识指定的分散读写组。

在本示例中，我们在写入文件 `reports/site51` 时指定从分散读写组 `0` 开始的循环分配：

```
user@solaris:~# setfa -s 0 -g0 reports/site51
```

3. 要在从指定分散读写组开始的多个分散读写组上分散读写文件，请使用分散读写分配。使用命令 `setfa -s stripe-width -g stripe-group-number`，其中 `stripe-width` 为 `[1-255]` 范围内的一个整数，用于指定一定数量的磁盘分配单元；`stripe-group-number` 是在 `[0-127]` 范围内的整数，用于标识指定的分散读写组。

在本示例中，我们为文件 `assessments/site52` 指定分散读写分配。我们为每个组指定三个磁盘分配单元（从分散读写组 `21` 开始）：

```
user@solaris:~# setfa -s 3 -g21 assessments/site52
```

4. 在此处停止。

使用扩展文件属性

与其他 Solaris 和 Linux 文件系统一样，Oracle HSM 文件系统支持扩展文件属性。扩展属性存放由用户或应用程序（而非文件系统本身）与某个文件关联的任意元数据。扩展属性已用来存放文件摘要、作者姓名、源应用程序的名称以及文本文件使用的字符编码。

从发行版 6.1 开始，Oracle HSM 存储小型扩展属性文件，这些文件在元数据分区内的扩展 inode 中（而不是数据分区中的块）包含 464 个或更少字符。当使用扩展属性且文件系统元数据存储于较快的设备（如闪存）上时，新方法会显著提高文件系统性能。

每当您创建新文件系统或者从恢复点 (*samfsdump*) 文件恢复旧文件系统时，扩展文件属性会自动启用。有关使用扩展属性的更多信息，请参见 Solaris *fsattr(5)* 和 Linux *xattr(7)* 手册页。

适应大型文件

Oracle HSM 文件系统特别适合处理超大文件。本节介绍以下主题：

- [管理特大型文件的磁盘高速缓存](#)
- [文件分段](#)
- [将可移除介质文件用于大型数据集](#)

管理特大型文件的磁盘高速缓存

在处理特大型文件时，请特别注意可用的磁盘高速缓存大小。如果尝试写入的文件比磁盘高速缓存大，则非归档文件系统将返回 *ENOSPC* 错误，而归档文件系统则会等待可能永远也不会变得可用的空间，从而导致应用程序受阻。

Oracle HSM 提供了两种可能的备选方法来增加磁盘高速缓存的大小：

- 文件分段，使用户在任何给定时间仅将大型文件的一部分回写到磁盘。
- 将可移除介质文件用于大型数据集，从而使用户绝不会将数据回写到磁盘。

文件分段

为文件设置 Oracle HSM 分段属性时，文件系统会将文件分成指定大小的段，并管理访问请求，从而使得在任何给定时间只有当前需要的段位于磁盘上。文件的剩余部分位于可移除介质上。

对大型文件分段有多项优势：

- 用户可以创建和访问大于可用磁盘高速缓存大小的文件。
由于在任何给定时间只有一些段位于高速缓存中，因此只需要选择适合磁盘高速缓存的段大小。整个文件可以增长到介质能容纳的任何大小。
- 用户可以更快地访问从磁盘高速缓存中释放的大型文件。将大型文件的一部分回写到磁盘比等待整个文件回写要快得多。
- 文件分段后，归档的速度和效率都会提升，因为仅重新归档每个文件发生更改的部分。
- 可以在挂载到多个驱动器的可移除介质卷上分散读写文件。随后，可以继续以并行方式进行归档和回写操作，从而进一步提高性能。

存在两项限制：

- 不能在共享文件系统中进行文件分段。
- 不能将二进制可执行文件分段，因为 Solaris 内存映射函数 *mmap()* 无法将分段文件中的字节映射到进程的地址空间。

要创建分段文件，请执行如下操作：

将文件分段

1. 登录到文件系统主机。

```
user@solaris:~#
```

2. 选择或者在必要时创建需要分段的文件。
3. 要将单个文件分段，请使用命令 `segment [-s stage_ahead] -l segment_size file-path-name`，其中：

- `stage_ahead` (可选) 是一个整数，用于指定在访问给定段时要读取的连续额外段的数量。恰当地选择值可以提高系统页面高速缓存的利用率，从而提高 I/O 性能。默认值为 0 (禁用)。
- `segment_size` 是一个整数及单位，放在一起指定每个分段的大小。支持的单位包括 *k* (千字节)、*m* (兆字节) 以及 *g* (千兆字节)。最小大小为一兆字节 (`1m` 或 `1024k`)。
- `file-path-name` 是文件的路径和文件名。

有关完整的详细信息，请参见 `segment` 手册页。在示例中，使用 1.5 兆字节 (`1536k`) 段大小对文件 `201401.dat` 分段：

```
user@solaris:~# segment -l 1536k 201401.dat
```

4. 要以递归方式对某个目录及其所有子目录中的文件分段，请使用命令 `segment [-s stage_ahead] -l segment_size -r directory-path-name`，其中 `directory-path-name` 是起始目录的路径和名称。

在示例中，使用 1 兆字节 (`1m`) 段大小对 `/hsm/hsmfs1/data` 目录及其子目录中的所有文件分段：

```
user@solaris:~# segment -l 1m -r /hsm/hsmfs1/data
```

5. 在此处停止。

在多个卷上分散读写分段文件

通过将分段文件分配给指定了多个驱动器的归档集，可以为分段文件配置分散读写 I/O。执行如下操作：

1. 以 `root` 用户身份登录主机。

```
root@solaris:~#
```

2. 在文本编辑器中打开文件 `/etc/opt/SUNWsamfs/archiver.cmd`。

在示例中，使用 `vi` 编辑器打开该文件：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
# Configuration file for Oracle HSM archiving file systems ...
```

3. 要在多个驱动器上分散读写分段文件，请为包含分段文件的每个归档集的每个副本指定至少使用两个驱动器。在 `archiver.cmd` 文件中，找到 `params` 部分。确保每个副本的参数包含 `-drives number` 参数，其中 `number` 是二 (2) 或更多。进行所需的更改，保存文件，然后关闭编辑器。

在该示例中，`archiver.cmd` 文件为所有配置的归档集的全部三个副本指定了两个驱动器：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
# Configuration file for Oracle HSM archiving file systems ...
...
#-----
# Copy Parameters
params
allsets -sort path -offline_copy stageahead -reserve set
allsets.1 -startage 10m -drives 2
allsets.2 -startage 24h -drives 2
allsets.3 -startage 48h -drives 2
endparams
...
:wq
root@solaris:~#
```

4. 检查 `archiver.cmd` 文件中的错误。使用命令 `archiver -lv`。

`archiver -lv` 命令将 `archiver.cmd` 文件输出到屏幕，如果未发现错误将生成一个配置报告。否则，将记录所有错误并停止。

```
root@solaris:~# archiver -lv
Reading '/etc/opt/SUNWsamfs/archiver.cmd'.
...
Total space available: 300T
root@solaris:~#
```

5. 指示 Oracle HSM 软件重新读取 `archiver.cmd` 文件并相应地重新配置自身。使用 `/opt/SUNWsamfs/sbin/samd config` 命令。

```
root@solaris:~# samd config
```

6. 在此处停止。

将可移除介质文件用于大型数据集

Oracle HSM 可移除介质文件完全位于可移除介质上，因此从来不会占用文件系统磁盘高速缓存中的空间。文件系统将可移除介质文件直接读取到内存中。因此，存储介质根本不会限制文件的大小。超过单个介质磁带容量的可移除文件可能会成为多磁带卷溢出文件。文件系统按顺序读取数据并将其写入介质。

在大多数方面，可移除介质文件就像是典型的 UNIX 文件。其具有权限、用户名、组名和文件大小。用户或应用程序请求可移除介质文件时，系统会自动挂载对应的卷，用户从内存中访问数据，就像数据在磁盘上一样。但是，可移除介质文件不同于其他 Oracle HSM 文件的地方主要在两个方面：Oracle Hierarchical Storage Manager 软件从不对其归档，并且在 NFS 中不支持这类文件。

Oracle Hierarchical Storage Manager 软件不管理可移除介质文件。永远不会归档或释放这类文件，也永远不会回收包含它们的介质。当需要将可移除介质用于归档以外的目的时，可移除介质文件很有用。这类文件非常适合创建可移除的灾难恢复卷，用于备份 Oracle HSM 配置文件和元数据转储文件。还可以从外部卷（由其他应用程序创建的卷）读取数据，即以只读方式装入卷并将文件作为可移除介质文件读取到内存中。

由于无法释放可移除介质文件，并且不能回收关联的卷，因此，通常应将可移除介质文件单独放在专用卷上，而不应将它们与归档副本混合存放。

创建可移除介质或卷溢出文件

1. 登录到文件系统主机。

```
user@solaris:~#
```

2. 为可移除介质文件选择 Oracle HSM 文件系统、路径和文件名。

创建了可移除介质文件之后，文件系统将使用来自可移除介质的数据响应对此路径和文件名的请求。

3. 创建可移除介质文件。使用命令 `request -m media-type -v volume-specifier data-file`，其中 `mediatype` 是附录 A, [设备类型词汇表](#) 中列出的两字符介质类型代码之一，`data-file` 是为可移除介质文件选择的路径和名称，`volume-specifier` 是以下对象之一：

- 卷序列号或以斜杠分隔的卷序列号列表

在第一个示例中，在 LTO (*li*) 卷 *VOL080* 上创建了 *file1*：

```
user@solaris:~# request -m li -v VOL080 /hsm/hsmfs1/data/file1
```

在第二个示例中，在 LTO (*li*) 卷 *VOL080*、*VOL082* 和 *VOL098* 上创建了 *file2*：

```
user@solaris:~# request -m li -v VOL081/VOL082/VOL098 /hsm/hsmfs1/data/file2
```

- *-l volume-list-file*，其中 *volume-list-file* 是文件的路径和名称；在每一行上，列出一个卷序列号以及可选的空格和十进制或十六进制数值，该数值指定在指定卷上的开始位置（十六进制数值带 *0x* 前缀）。

在示例中，使用 *vi* 编辑器在文件 *vsnsfile3* 中列出的 LTO (*li*) 卷上创建 *file3*：

```
user@solaris:~# vi vsnsfile3
VOL180
VOL181
VOL182
:wq
user@solaris:~# request -m li -v -l vsnsfile3 /hsm/hsmfs1/data/file3
```

4. 在此处停止。

读取外部磁带卷作为可移除介质文件

1. 登录到文件系统主机。

```
user@solaris:~#
```

2. 确保外部磁带有条码、启用写保护、以只读方式打开并且定位到 *0*。
3. 为可移除介质文件选择 Oracle HSM 文件系统、路径和文件名。

创建了可移除介质文件之后，文件系统将使用来自外部磁带的响应数据对此路径和文件名的请求。

4. 使用 *-N*（外部介质）选项创建可移除介质文件。使用命令 *request -m mediatype -N -v volume-serial-number data-file*，其中：
 - *mediatype* 为附录 A, [设备类型词汇表](#)中列出的由两个字符表示的介质类型代码之一。
 - *volume-serial-number* 是外部磁带的卷序列号。
 - *data-file* 是可移除介质文件的路径和名称。

在该示例中，为外部 LTO (*li*) 卷 *FOR991* 创建了一个可移除介质文件：

```
user@solaris:~# request -m li -N -v FOR991 /hsm/hsmfs1/foreignfile
```

5. 在此处停止。

使用 Linear Tape File System (LTFS) 卷

Linear Tape File System 是自描述磁带格式，可将顺序访问磁带介质上的数据组织到文件系统中，以便可以像文件位于随机访问磁盘上一样对其进行访问。Oracle HSM 提供对 LTFS 的广泛支持。该软件支持在 Oracle HSM 文件系统中使用 LTFS 文件，并提供了创建、访问和管理 LTFS 介质的工具。

本节介绍以下主题：

- [将 LTFS 介质导入到磁带库](#)
- [将 LTFS 目录和文件连接到 Oracle HSM 文件系统](#)
- [使用 Oracle HSM 软件访问 LTFS 介质](#)
- [使用 Oracle HSM 软件管理 LTFS 介质](#)

将 LTFS 介质导入到磁带库

Oracle HSM 软件会自动识别 LTFS 介质。因此，可以使用 *samimport* 命令导入 LTFS 卷，就像导入其他任何介质一样。有关其他信息，请参见[“导入和导出可移除介质”](#)和 *samimport* 手册页。

将 LTFS 目录和文件连接到 Oracle HSM 文件系统

Oracle HSM 软件可以将 Linear Tape File System (LTFS) 目录和文件连接到 Oracle HSM 文件系统，使这些目录和文件可以当作 Oracle HSM 文件进行访问和管理。该软件将 LTFS 元数据从 LTFS 卷复制到 Oracle HSM 文件系统中的空目录。使用此元数据，Oracle HSM 就像管理归档的 Oracle HSM 文件一样管理 LTFS 介质和文件。用户依次访问 LTFS 文件或者同时访问这些 LTFS 文件时，只要 LTFS 元数据就位，就将 LTFS 文件从 LTFS 介质回写到 Oracle HSM 磁盘高速缓存供使用。将应用 Oracle HSM 文件系统的归档和空间管理策略，就像对待任何 Oracle HSM 文件一样。

本节介绍以下任务：

- [使 LTFS 文件可按需访问](#)
- [使 LTFS 文件在磁盘高速缓存中可立即访问](#)

使 LTFS 文件可按需访问

将 LTFS 文件连接到 Oracle HSM 文件系统时，Oracle HSM 软件会将文件系统元数据从 LTFS 卷复制到 Oracle HSM 文件系统中的指定目录。然后，在用户访问这些文件时，将其回写到磁盘高速缓存。要连接 LTFS 文件，请执行如下操作：

1. 登录到文件系统主机。

```
user@solaris:~#
```

2. 在承载 LTFS 文件的 Oracle HSM 文件系统中，创建用于存放 LTFS 元数据的目录。

在示例中，在文件系统挂载点 `/hsm/hsmfs1` 下创建了目录 `ltfs1/`：

```
user@solaris:~# mkdir /hsm/hsmfs1/ltfs1
user@solaris:~#
```

3. 将 LTFS 文件连接到 Oracle HSM 文件系统。使用命令 `samltfs attach LTFS-media-type.LTFS-volume-serial-number SAMQFS-directory`，其中：
 - `LTFS-media-type` 是存放 LTFS 数据的介质类型的两字符介质类型代码（请参见附录 A, 设备类型词汇表）。
 - `LTFS-volume-serial-number` 是 LTFS 卷的六字符卷序列号（使用字母数字）。
 - 指定的介质类型和卷序列号用于标识目录列为 LTFS 卷的卷。

在 Oracle HSM 目录中，LTFS 介质是无标签的并标记为 `non-SAM` 和 `tfs`。

- `SAMQFS-directory` 是将存放 LTFS 元数据的目录的路径和名称。

在该示例中，连接了 LTO (*li*) 卷 `TFS233`：

```
user@solaris:~# samltfs attach li.TFS233 /hsm/hsmfs1/ltfs1
user@solaris:~#
```

4. 在此处停止。

使 LTFS 文件在磁盘高速缓存中可立即访问

将 LTFS 文件获取到 Oracle HSM 文件系统中时，Oracle HSM 软件会将文件系统元数据从 LTFS 卷复制到 Oracle HSM 文件系统中的指定目录，并立即将所有文件回写到磁盘高速缓存。要获取 LTFS 文件，请执行如下操作：

1. 登录到文件系统主机。

```
user@solaris:~#
```

2. 在承载 LTFS 文件的 Oracle HSM 文件系统中，创建用于存放 LTFS 元数据的目录。

在示例中，在文件系统挂载点 `/hsm/hsmfs1` 下创建了目录 `ltfs2/`：

```
user@solaris:~# mkdir /hsm/hsmfs1/ltfs2
user@solaris:~#
```

3. 将 LTFS 文件获取到 Oracle HSM 文件系统中。使用命令 `samltps ingest LTFS-media-type.LTFS-volume-serial-number SAMQFS-directory`，其中：
 - `LTFS-media-type` 是存放 LTFS 数据的介质类型的两字符介质类型代码（请参见附录 A, 设备类型词汇表）。
 - `LTFS-volume-serial-number` 是 LTFS 卷的六字符卷序列号（使用字母数字）。
 - 指定的介质类型和卷序列号用于标识目录列为 LTFS 卷的卷。

在 Oracle HSM 目录中，LTFS 介质是无标签的并标记为 `non-SAM` 和 `tfs`。

- `SAMQFS-directory` 是存放 LTFS 元数据的目录的路径和名称。

在该示例中，获取了 LTO (*Li*) 卷 `TFS234`：

```
user@solaris:~# samltfs ingest li.TFS234 /hsm/hsmfs1/ltfs2
user@solaris:~#
```

4. 在此处停止。

使用 Oracle HSM 软件访问 LTFS 介质

Oracle HSM 软件还可以在主机上装入和卸载 LTFS 介质，以及使用 Oracle HSM `defaults.conf` 文件中指定的 LTFS 挂载点挂载或卸载 LTFS 文件系统。

- 将 LTFS 卷装入磁带机并挂载 LTFS 文件系统
- 卸载 LTFS 文件系统，并从磁带机中卸载卷
- 显示 LTFS 配置和状态信息。

将 LTFS 卷装入磁带机并挂载 LTFS 文件系统

1. 登录到文件系统主机。

```
user@solaris:~#
```

2. 将 LTFS 卷装入磁带机并在 `defaults.conf` 文件中指定的挂载点上挂载文件系统。使用命令 `samltps load LTFS-media-type.LTFS-volume-serial-number`，其中：
 - `LTFS-media-type` 是存放 LTFS 数据的介质类型的两字符介质类型代码（请参见附录 A, 设备类型词汇表）。
 - `LTFS-volume-serial-number` 是 LTFS 卷的六字符卷序列号（使用字母数字）。
 - 指定的介质类型和卷序列号用于标识目录列为 LTFS 卷的卷。

在 Oracle HSM 目录中，LTFS 介质是无标签的并标记为 `non-SAM` 和 `tfs`。

在该示例中，装入 LTO (*li*) 卷 *TFS434* 并将其挂载到 *defaults.conf* 文件中指定的目录 */mnt/ltfs* 下：

```
user@solaris:~# samltfs load li.TFS234
```

3. 在此处停止。

卸载 LTFS 文件系统，并从磁带机中卸载卷

1. 登录到文件系统主机。

```
user@solaris:~#
```

2. 卸载 LTFS 文件系统，并从磁带机中卸载对应的卷。使用命令 *samltfs unload LTFS-media-type.LTFS-volume-serial-number*，其中：

- *LTFS-media-type* 是存放 LTFS 数据的介质类型的两字符介质类型代码（请参见附录 A, [设备类型词汇表](#)）。
- *LTFS-volume-serial-number* 是 LTFS 卷的六字符卷序列号（使用字母数字）。
- 指定的介质类型和卷序列号用于标识目录列为 LTFS 卷的 LTFS 卷。

在 Oracle HSM 目录中，LTFS 介质是无标签的并标记为 *non-SAM* 和 *tfs*。

在该示例中，卸载 LTFS 文件系统并卸载 LTO (*li*) 卷 *TFS435*：

```
user@solaris:~# samltfs unload li.TFS435
```

3. 在此处停止。

使用 Oracle HSM 软件管理 LTFS 介质

Oracle HSM 软件提供了创建、清除和验证 LTFS 介质所需的基本工具：

- [将卷格式化为 LTFS 文件系统](#)
- [清除 LTFS 数据，并从卷中删除格式和分区](#)
- [检查 LTFS 文件系统的完整性](#)

将卷格式化为 LTFS 文件系统

1. 登录到文件系统主机。

```
user@solaris:~#
```

2. 对 LTFS 文件系统的可移除介质卷进行分区和格式化。使用命令 *samltfs mkltfs media-type.volume-serial-number*，其中：

- *media-type* 是 LTFS 兼容介质类型的两字符介质类型代码（请参见[附录 A, 设备类型词汇表](#)）。
- *volume-serial-number* 是使用字母数字的六字符卷序列号。

在该示例中，对 LTO (*li*) 卷 *VOL234* 分区并将其格式化为 LTFS 卷：

```
user@solaris:~# samltfs mkltfs li.VOL234
```

3. 在此处停止。

清除 LTFS 数据，并从卷中删除格式和分区

1. 登录到文件系统主机。

```
user@solaris:~#
```

2. 清除 LTFS 卷并将其恢复为一般用途。使用命令 *samltfs unltfs media-type.volume-serial-number*，其中：

- *media-type* 是 LTFS 兼容介质类型的两字符介质类型代码（请参见[附录 A, 设备类型词汇表](#)）。
- *volume-serial-number* 是使用字母数字的六字符卷序列号。

在该示例中，清除 LTFS 文件系统数据和元数据，并删除 LTO (*li*) 卷 *VOL234* 上的分区：

```
user@solaris:~# samltfs unltfs li.VOL234
```

3. 在此处停止。

检查 LTFS 文件系统的完整性

1. 登录到文件系统主机。

```
user@solaris:~#
```

2. 检查 LTFS 文件系统的完整性。使用命令 *samltfs ltfscck LTFS-media-type.LTFS-volume-serial-number*，其中：

- *LTFS-media-type* 是存放 LTFS 数据的介质类型的两字符介质类型代码（请参见[附录 A, 设备类型词汇表](#)）。
- *LTFS-volume-serial-number* 是 LTFS 卷的六字符卷序列号（使用字母数字）。
- 指定的介质类型和卷序列号用于标识目录列为 LTFS 卷的 LTFS 卷。

在 Oracle HSM 目录中，LTFS 介质是无标签的并标记为 *non-SAM* 和 *tfso*。

在该示例中，检查 LTO (*li*) 卷 *VOL234* 上的 LTFS 文件系统：

```
user@solaris:~# samltfs ltfck li.VOL234
```

3. 在此处停止。

显示 LTFS 配置和状态信息

要显示 LTFS 的配置和状态，请使用命令 `samltfs status`。

```
user@solaris:~# samltfs status
```

管理 SMB/CIFS 共享中的目录和文件

本节介绍以下主题：

- [管理 SMB/CIFS 共享中的系统属性](#)
- [管理访问控制列表](#)。

管理 SMB/CIFS 共享中的系统属性

系统属性通过将 Oracle HSM 文件与可使用 Microsoft Windows 文件系统解释的非 UNIX 元数据关联来支持 SMB/CIFS 文件共享。本节首先简要介绍 Oracle HSM 支持的系统属性。然后，提供以下任务的基本说明：

- [显示系统属性](#)
- [修改系统属性](#)。

Oracle HSM 支持的系统属性

系统属性是布尔 (`true` 或 `false`) 值，它由值为 `true` 的属性 `name` 或者值为 `false` 的名称求反 `noname` 表示。Oracle HSM 为了支持 SMB/CIFS 文件共享提供了以下系统属性：

- `appendonly` 表示用户只能将数据附加到文件。`noappendonly` 表示此限制无效。
- `archive` 表示文件自上次复制或备份以来已发生更改。`noarchive` 表示文件自上次复制或备份以来未发生更改。Oracle HSM 当前不使用此属性。
- `hidden` 表示默认情况下文件不在文件列表中显示。`nohidden` 表示默认情况下显示文件。
- `immutable` 表示不能更改或删除目录或文件及其内容。`noimmutable` 表示可以更改或删除目录或文件。
- `nodump` 表示不能备份文件。`nonodump` 表示可以备份文件。Oracle Solaris 不使用此属性。

- *nounlink* 表示不能删除或重命名文件或目录及其内容。*nonounlink* 表示可以删除或重命名文件或目录及其内容。
- *offline* 表示已从 Oracle HSM 文件系统释放文件。Microsoft Windows 系统不预览文件。*nooffline* 表示文件处于联机状态，并且尚未从 Oracle HSM 文件系统释放。
- *readonly* 表示不能删除或修改文件。*noreadonly* 表示可以删除或修改文件。在应用于目录时，将忽略该属性。
- *sparse* 表示存储的文件仅包含非零数据，并且零已缩减到在不支持稀疏文件的文件系统中访问此文件或者将此文件复制到此类系统时，文件系统可以恢复零的范围中。*nosparse* 表示文件不是稀疏文件。
- *system* 表示文件对 Microsoft Windows 操作系统很重要，不得更改或删除，并且默认情况下不应在列表中显示。*nosystem* 表示文件不是系统文件。

显示系统属性

要查看 Oracle HSM 文件的系统属性，请使用 Solaris 命令 `ls -/v file`，其中 *file* 是文件的路径和名称。

在示例中，列出了文件 `/hsm/hsmfs1/documents/master-plan.odt` 的系统属性：

```
user@solaris:~# ls -/v /hsm/hsmfs1/documents/master-plan.odt
-rw-r--r--  1 root root  40560 Mar 4 15:52 /hsm/hsmfs1/documents/master-plan.odt
{archive,nohidden,noreadonly,nosystem,noappendonly,nonodump,noimmutable,nonounlink,
 nooffline,nosparse}
user@solaris:~#
```

修改系统属性

要将文件的系统属性值更改为指定的值，请使用 Solaris 命令 `chmod S +v{attributes}`，其中 *attributes* 是 Oracle HSM 支持的系统属性的逗号分隔列表。

有关语法和可用选项的全面说明，请参见 `chmod` 手册页。在该示例中，将归档属性从 *noarchive* (false) 更改为 *archive* (true)：

```
root@solaris:~# ls -/v /hsm/hsmfs1/documents/master-plan.odt
-r-xr-xr-x 1 root root 40561 Mar 4 15:52 /hsm/hsmfs1/documents/master-plan.odt
{noarchive,nohidden,readonly,nosystem,noappendonly,nonodump,noimmutable,
 nonounlink,offline,nosparse}
root@solaris:~# chmod S+v{archive} /hsm/hsmfs1/documents/master-plan.odt
root@solaris:~# ls -/v /hsm/hsmfs1/documents/master-plan.odt
-r-xr-xr-x 1 root root 40561 Mar 4 15:52 /hsm/hsmfs1/documents/master-plan.odt
{archive,nohidden,readonly,nosystem,noappendonly,nonodump,noimmutable,
 nonounlink,offline,nosparse}
```

管理访问控制列表

访问控制列表 (Access Control List, ACL) 是定义文件或目录访问权限的表。该表中的每个记录或访问控制条目 (Access Control Entry, ACE) 定义特定用户、组或者一类用

户或组的访问权限。默认情况下，使用 Oracle HSM 发行版 6.1 创建的新文件系统使用在网络文件系统 (Network File System, NFS) 版本 4 和 Solaris 11 中引入的访问控制列表 (Access Control List, ACL) 实施。

Solaris ACL 管理、语法和用法的全面介绍不在本文档的范围内。有关完整信息，请参见 *Oracle Solaris 11.1* 信息库（位于 *docs.oracle.com* 中）中提供的《Oracle Solaris 11.1 管理：ZFS 文件系统》一书中的“使用 ACL 和属性保护 Oracle Solaris ZFS 文件”一章。另请参见 Solaris *ls* 和 *chmod* 手册页。

第 5 章 管理库、介质和驱动器

本章介绍以下主题：

- [管理自动化介质库](#)
- [管理驱动器](#)
- [管理可移除介质。](#)

管理自动化介质库

本节介绍与库维护和管理关联的基本任务：

- [使库联机和脱机](#)
- [导入和导出可移除介质](#)
- [维护库目录](#)
- [确定驱动器在库中的安装顺序。](#)

使库联机和脱机

- [使库脱机](#)
- [使库联机。](#)

使库脱机

如果仅需要在库上停止 Oracle HSM 运行或需要关闭库的电源，请先按如下所述使库脱机：

1. 以 *root* 用户身份登录到文件系统主机。

```
root@solaris:~#
```

2. 完成处于活动状态的归档和回写作业并阻止任何新作业启动。请参见[“使归档和回写进程闲置”](#)。
3. 停止驱动器和库活动。请参见[“停止归档和回写进程”](#)。
4. 使库脱机。使用命令 `samcmd off library-equipment-number`，其中 `library-equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给库的设备序号。

将库置于 *off* 状态可以停止 I/O 操作，并使该库不再受 Oracle HSM 软件控制。任何尚未关闭电源的驱动器会保持 *on* 状态。在示例中，使库 800 脱机并使用 *samcmd c* 检查结果：

```
root@solaris:~# samcmd off 800
root@solaris:~# samcmd c
Device configuration samcmd      5.4 14:34:04 Mar  7 2014
samcmd on hsmfs1host
Device configuration:
ty  eq state  device_name                fs  family_set
sn  800 off    /dev/scsi/changer/c1t2d0   800 lib800
li  801 on     /dev/rmt/0cbn              800 lib800
li  802 on     /dev/rmt/1cbn              800 lib800
li  803 on     /dev/rmt/2cbn              800 lib800
li  804 on     /dev/rmt/3cbn              800 lib800
hy  900 on     historian                    900
root@solaris:~#
```

5. 准备好后，使磁带库联机。

使库联机

1. 以 *root* 用户身份登录到文件系统主机。

```
root@solaris:~#
```

2. 使库联机。使用命令 *samcmd on library-equipment-number*，其中 *library-equipment-number* 是 */etc/opt/SUNWsamfs/mcf* 文件中分配给库的设备序号。

库将联机。Oracle HSM 软件查询设备状态并根据需要更新目录。在示例中，使库 800 联机并使用 *samcmd c* 检查结果：

```
root@solaris:~# samcmd on 800
root@solaris:~# samcmd c
Device configuration samcmd      5.4 15:04:14 Mar  7 2014
samcmd on hsmfs1host
Device configuration:
ty  eq state  device_name                fs  family_set
sn  800 on     /dev/scsi/changer/c1t2d0   800 lib800
li  801 on     /dev/rmt/0cbn              800 lib800
li  802 on     /dev/rmt/1cbn              800 lib800
li  803 on     /dev/rmt/2cbn              800 lib800
li  804 on     /dev/rmt/3cbn              800 lib800
hy  900 on     historian                    900
root@solaris:~#
```


3. 在此处停止。

导入和导出可移除介质

许多自动化库都包括一个装卸托架，以便可以在身体不进入库的情况下添加或移除介质磁带。该托架可能称为邮箱、邮槽、介质存取口 (media access port, MAP) 或磁带存取口 (cartridge access port, CAP)，具体取决于供应商。对于该类型的库，可以使用 Oracle HSM 命令执行以下任务：

- 导入可移除介质磁带
- 导出可移除介质磁带。

如果您的库不包括邮箱，请查阅库供应商提供的文档以及您的本地站点策略，以获取有关添加和移除库介质的说明。当库在更改后重新初始化以及审计其内容时，Oracle HSM 软件将自动更新其库和历史记录目录。

导入可移除介质磁带

如果在 Oracle HSM 软件启动时库邮箱中包含介质磁带，该软件会自动将其装入库中。在软件运行后，您随时可以使用以下过程从邮箱导入介质：

1. 根据库供应商的说明将介质磁带置于邮箱中。
2. 以 *root* 用户身份登录到文件系统主机。

```
root@solaris:~#
```

3. 将磁带导入自动化库中。使用命令 *samimport library-equipment-number*，其中 *library-equipment-number* 是 */etc/opt/SUNWsamfs/mcf* 文件中为库指定的设备序号。

Oracle HSM 软件会将介质分配给存储插槽并记录其位置。在示例中，将介质导入库 *800* 中。

```
root@solaris:~# samimport 800
```

4. 在此处停止。

导出可移除介质磁带

1. 以 *root* 用户身份登录到文件系统主机。

```
root@solaris:~#
```

2. 如果需要，在导出磁带之前，向磁带的目录记录中添加一条信息性备注。使用命令 *chmed -I "note" identifier*，其中 *note* 是最多包含 128 个字符的字符串，*identifier* 是以下项之一：

- *mediatype.volume-serial-number*，其中 *mediatype* 是附录 A, 设备类型词汇表中列出的两字符介质类型代码之一，*volume-serial-number* 是在磁带库中唯一地标识卷的六字符字母数字字符串。
- *library-equipment-number:slot*，其中 *library-equipment-number* 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给自动化磁带库的设备序号，*slot* 是磁带在磁带库内所处的插槽地址。

在导出卷后，备注将保留在历史记录目录中。在示例中，向 LTO (*li*) 磁带 *VOL054* 的目录条目添加了一条备注：

```
root@solaris:~# chmed -I "To vault 20150411" li.VOL054
```

3. 要将磁带从指定的存储插槽移至邮箱，请使用命令 `samexport library-equipment-number:slot`，其中 *library-equipment-number* 是为 `/etc/opt/SUNWsamfs/mcf` 文件中的自动化磁带库指定的设备序号，*slot* 是磁带在磁带库中所处的插槽地址。

在示例中，导出了位于磁带库 *800* 的插槽 *11* 中的磁带：

```
root@solaris:~# samexport 800:11
```

4. 要将指定的磁带移动到邮箱，请使用命令 `samexport mediatype.volume-serial-number`，其中 *mediatype* 是附录 A, 设备类型词汇表中列出的两字符介质类型代码之一，*volume-serial-number* 是在库中唯一地标识卷的字母数字字符串。

Oracle HSM 软件会将磁带添加到由 Oracle HSM **historian** (历史记录) 维护的目录中。在示例中，导出了 LTO (*li*) 磁带 *VOL109*：

```
root@solaris:~# samexport li.VOL109
```

5. 根据库供应商的说明从邮箱中移除介质磁带。
6. 在此处停止。

维护库目录

Oracle Hierarchical Storage Manager 库目录是自动化库及其内容在软件中的内部表示形式。如果自动化库采用直接连接方式，则 Oracle HSM 软件对库及其内容具有完全控制权限。相应地，库目录的条目是物理库中插槽的一对一表示。如果自动化库是网络连接的，则 Oracle HSM 仅访问库软件以虚拟库或库分区形式提供的那部分库。因此，Oracle HSM 库目录的条目仅反映部分库的内容。

本节介绍以下任务：

- [查看库目录](#)
- [审计库插槽的内容](#)

- 审计整个直接连接的自动化库
- 从目录中清除介质错误。

查看库目录

1. 要查看最常用的库目录信息，请使用命令 `samcmd v library-equipment-number`，其中 `library-equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给自动化磁带库的设备序号。

在示例中，显示库 800 的目录

```
root@solaris:~# samcmd v 800
Robot catalog samcmd      5.4      16:45:25 Mar 14 2014
samcmd on samqfshost          count 32
Robot VSN catalog by slot    : eq 800
slot      access time count use  flags          ty vsn
   0      2014/03/14 11:23  875  0%  -il-o-b----- li VOL001
   1      2014/03/13 17:54  866  0%  -il-o-b----- li VOL002
   2      2014/03/14 11:26   3  0%  -il-o-b----- li VOL003
   3      2014/03/14 10:33   3  0%  -il-o-b----- li VOL004
   4      2014/03/14 11:34   5  0%  -il-o-b----- li VOL005
   5      2014/03/14 11:32   2  0%  -ilEo-b----f  li VOL006 MEDIA ERROR
   6      2014/03/13 18:07   2  0%  -il-o-b----- li VOL007
   7      2014/03/13 18:07   1  0%  -il-o-b----- li VOL008
   8      2014/03/13 18:07   1  0%  -il-o-b----- li VOL009
   ...
  18      2014/03/13 18:16   1  0%  -il-o-b----- li VOL019
  19      none                50  0%  -il-oCb----- li CLN020
```

2. 要使用 `samcmd v` 显示结果来确定卷的状态，请检查 `flags` 列中的条目并参考以下列表：
 - *A* 表示插槽需要审计。
 - *C* 表示插槽中有清洗磁带。
 - *E* 表示卷损坏或清洗介质已过期。
 - *L* 表示卷是 Linear Tape File System (LTFS) 卷。
 - *N* 表示卷是外部介质（而非 Oracle HSM 格式）。
 - *R* 表示卷只读（软件标志）。
 - *U* 表示卷不可用。
 - *W* 表示卷处于物理写保护状态。
 - *X* 表示插槽是导出插槽。
 - *b* 表示卷有条形码。
 - *c* 表示卷可回收。
 - *f* 表示归档程序发现卷已满或已损坏。

- *d* 表示卷有重复的卷序列号 (volume serial number, VSN)。
 - *l* 表示卷有标签。
 - *o* 表示插槽已占用。
 - *p* 表示卷是高优先级卷。
 - - 表示对应的标志未设置。
3. 要使用 `samcmd v` 显示结果来识别用于卷的介质类型, 请参考 *ty* 列并查找[附录 A, 设备类型词汇表](#)或 *mcf* 手册页中所示的代码。
 4. 要列出目录中的所有信息, 请使用命令 `dump_cat catalog-path-name`, 其中 *catalog-path-name* 是目录文件的路径和文件名, 如 `/etc/opt/SUNWsamfs/mcf` 文件中所指定。

在示例中, 转储目录文件 `catalog/800_cat`。

```
root@solaris:~# dump_cat catalog/800_cat
# audit_time Wed Dec 31 17:00:00 1969
# version 530 count 32 mediatype
#Index VSN Barcode Type PTOC Access Capacity ... LVTime LVPos
#
  0 S00001 S00001L4 li 0x747 875 512000 ... 0 0x3
  1 S00002 S00002L4 li 0x5db 866 512000 ... 0 0x3
13 S00014 S00014L4 li 0 4 512000 ... 0 0
17 S00018 S00018L4 li 0 1 512000 ... 0 0
18 S00003 S00003L4 li 0 3 512000 ... 0 0
```

5. 在此处停止。

审计库插槽的内容

要使用所报告的可移除介质卷上剩余的空间更新库目录, 请审计库插槽。使用命令 `auditslot`。

1. 以 `root` 用户身份登录到文件系统主机。

```
root@solaris:~#
```

2. 要审计指定的磁带卷, 请跳到 EOD (*end of data*, 数据末尾), 并使用命令 `auditslot -e library-equipment-number:slot` 更新可用空间, 其中 *library-equipment-number* 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给磁带库的设备序号, *slot* 是磁带在磁带库内的位置。

`auditslot` 命令用于装入包含卷的磁带、读取标签并更新插槽的库目录条目。请注意, 一旦开始跳到 EOD 就无法中断此操作, 并且在某些情况下可能要花费数小时才能完成。在示例中, 审计磁带库 `800` 中的插槽 `11`:

```
root@solaris:~# auditslot -e 800:11
```

```
root@solaris:~#
```

- 要审计指定的光盘卷，请使用命令 `auditslot library-equipment-number:slot[:side]`，其中 `library-equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给库的设备序号，`slot` 是磁带在库内的位置，`side`（可选）是双面光盘的指定面。

在示例中，审计光盘库 700 的插槽 21 中卷的第 1 面：

```
root@solaris:~# auditslot 800:21:1
root@solaris:~#
```

- 在此处停止。

审计整个直接连接的自动化库

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

- 在未使用 Oracle HSM 命令的情况下移动了自动化库中的磁带后
- 库目录可能不可靠时（例如在电源断电后）
- 在没有配备邮箱的自动化库中添加、移除或移动了磁带时。

要执行完全审计，请使用命令 `samcmd audit library-equipment-number`，其中 `library-equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给库的设备序号。

请注意，完整的审计将花费较长时间，具体取决于包含介质的插槽的数量。

在示例中，审计了磁带库 800：

```
root@solaris:~# audit 800
root@solaris:~#
```

从目录中清除介质错误

如果 Oracle HSM 在使用可移除介质磁带时遇到问题，它会在对应的目录条目中设置一个错误标志。介质可能已磨损、损坏或过期（对于清洗介质）。在这些情况下，不应重用介质。但是在访问介质时发生的问题还可能由驱动器故障引起的，在这种情况下，可以毫无困难地重新使用介质。在后一种情况下，您需要清除磁带的错误标志。

在清除错误标志之前，确保您知道问题的性质。错误标志对于 Oracle HSM 运行和数据安全至关重要。如果磁带确实有故障，则不需要清除此标志。

如果您有把握，则可以清除错误并尝试使用磁带。执行如下操作：

- 以 `root` 用户身份登录到文件系统主机。

```
root@solaris:~#
```

2. 检查可移除介质卷的状态。使用命令 `samcmd r`。

在示例中，`samcmd r` 命令显示驱动器 801 在 LTO (1i) 卷 VOL004 上设置了错误标志。

```
root@solaris:~# samcmd r
Removable media status: all          samcmd 5.4          17:40:11 Mar 13 2014
ty  eq  status      act  use  state  vsn
li  801 -E-----r    0   0%  notrdy VOL004 MEDIA ERROR
      MEDIA ERROR
li  802 -----p    0   0%  notrdy
      empty
li  803 -----p    0   0%  notrdy
      empty
li  804 -----p    0   0%  notrdy
      empty
root@solaris:~#
```

3. 如果对设置错误标志的驱动器存有疑问，请卸载磁带并清除错误标志。使用命令 `samcmd unload drive-number`，其中 `drive-number` 是为 `/etc/opt/SUNWsamfs/mcf` 文件中的驱动器指定的设备序号。

在示例中，卸载驱动器 801：

```
root@solaris:~# samcmd unload 801
```

4. 要清除指定卷序列号和介质类型的介质错误标志，请使用命令 `chmed -E mediatype.volume-serial-number`，其中 `mediatype` 是附录 A, 设备类型词汇表中列出的两字符介质类型代码之一，`volume-serial-number` 是在磁带库中唯一地标识卷的字母数字字符串。

在示例中，清除了 LTO (1i) 卷 VOL004 上的错误标志：

```
root@solaris:~# chmed -E li.VOL004
  3:0 li VOL004   Ail---b-----  2.3T  2.3T  0          0 800 4  0 //
root@solaris:~#
```

5. 要为指定的磁带库插槽中的磁带清除介质错误标志，请使用命令 `chmed -E library-equipment-number:slot[:disk-side]`，其中 `library-equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给自动化磁带库的设备序号，`slot` 是目标卷在磁带库内所处的插槽地址，可选的 `disk-side` 值 (1 或 2) 指定双面磁光盘的其中一面。

在示例中，清除了库 800 插槽 31 中的磁带上的错误标志：

```
root@solaris:~# chmed -E 800:31
```

- 更新库目录以反映更改。使用命令 `auditslot -e library-equipment-number:slot[:disk-side]`。

在示例中，通过审计库 800 的插槽 31 来更新目录：

```
root@solaris:~# auditslot -e 800:31
root@solaris:~#
```

- 将磁带装入另一个驱动器中，并查看是否还发生错误。使用命令 `samcmd load media-type.volume-serial-number`，其中 `mediatype` 是附录 A, 设备类型词汇表中列出的两字符介质类型代码之一，`volume-serial-number` 是在磁带库中唯一地标识卷的字母数字字符串。

```
root@solaris:~# samcmd load li.VOL004
root@solaris:~#
```

- 重新检查可移除介质卷的状态。使用命令 `samcmd r`。

```
root@solaris:~# samcmd r
Removable media status: all          samcmd 5.4          17:42:10 Mar 13 2014
ty  eq status      act use state vsn
li  801 -----p   0  0% notrdy
      empty
li  802 --l-----r   0  0% ready  VOL004
      idle
li  803 -----p   0  0% notrdy
      empty
li  804 -----p   0  0% notrdy
      empty
root@solaris:~#
```

- 如果在新驱动器上没有再发生错误，则表明磁带没有问题。
- 如果错误再次发生，则考虑停用可移除介质卷。
- 在此处停止。

管理历史记录目录

Oracle Hierarchical Storage Manager 历史记录是一个伪库，它有一个目录但没有设备。历史记录将不再受 Oracle HSM 直接控制的卷编入目录中。因此，它维护着已从库中导出并发送到异地进行存储的卷的记录，以及手动装入到独立驱动器中的卷的记

录。当您从库中导出卷时，Oracle HSM 会自动更新历史记录目录。但是，您还可以使用历史记录通过添加和/或删除记录并附加备注来手动维护记录。通常，与历史记录的交互就像与物理介质库交互一样：

本节概述以下任务：

- [查看历史记录目录](#)
- [在历史记录目录中添加条目](#)
- [从历史记录目录中删除条目。](#)

查看历史记录目录

您可以与物理库一样的方式查看历史记录目录。使用命令 `samcmd v historian-equipment-number`，其中 `historian-equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给历史记录的设备序号。

在示例中，显示设备序号为 `900` 的历史记录的目录：

```
root@solaris:~# samcmd v 900
Robot catalog samcmd      5.4      16:45:25 Mar 14 2014
samcmd on samqfshost      count 32
Robot VSN catalog by slot : eq 900
slot      access time count use  flags      ty vsn
  0      2014/03/14 11:23 875  0%  -il-o-b-----  li EXT001
  1      2014/03/13 17:54 866  0%  -il-o-b-----  li EXT002
```

在历史记录目录中添加条目

要将条目添加到历史记录目录中，请执行如下操作：

1. 要在历史记录目录中为指定的卷序列号添加条目，请使用命令 `samimport -v volume-serial-number -m mediatype historian-equipment-number`，其中：
 - `volume-serial-number` 是在目录中唯一地标识卷的字母数字字符串。
 - `mediatype` 为[附录 A, 设备类型词汇表](#)中列出的由两个字符表示的介质类型代码之一。
 - `historian-equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给历史记录的设备序号。

在示例中，在历史记录 `900` 的目录中为 LTO (`li`) 卷 `EXT003` 添加一条记录：

```
root@solaris:~# samimport -v EXT003 -m li 900
]
```

2. 要在历史记录目录中为指定条码添加条目，请使用命令 `samimport -b barcode -m mediatype historian-equipment-number`，其中 `barcode` 是粘贴在对应的物理磁带上的条码。

在示例中，在历史记录 `900` 的目录中为条码为 `EXT003L4` 的 LTO (`li`) 卷添加一条记录：


```
root@solaris:~# samimport -b EXT003L4 -m li 900
```

3. 在此处停止。

从历史记录目录中删除条目

要从历史记录目录中删除条目，请使用命令 `samexport historian-equipment-number:slot`，其中 `historian-equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给历史记录的设备序号，`slot` 是记录对应的历史记录插槽地址。

在示例中，从历史库 `900` 的目录中删除插槽 `1` 中的卷 `EXT002` 的记录：

```
root@solaris:~# samcmd v 900
Robot catalog samcmd      5.4    16:45:25 Mar 14 2014
samcmd on samqfshost      count 32
Robot VSN catalog by slot : eq 900
slot      access time count use flags          ty vsn
  0      2014/03/14 11:23 875 0% -il-o-b----- li EXT001
  1      2014/03/13 17:54 866 0% -il-o-b----- li EXT002
  2      2014/03/13 17:57 866 0% -il-o-b----- li EXT003
root@solaris:~# samexport 900:1
```

更新历史记录信息

您可以更新历史记录目录条目中的信息字段以指明导出卷的处置或状态发生更改。使用命令 `chmed -I "note" identifier`，其中 `note` 是最多包含 128 个字符的字符串，`identifier` 是以下项之一：

- `mediatype.volume-serial-number`，其中 `mediatype` 是[附录 A, 设备类型词汇表](#)中列出的两字符介质类型代码之一，`volume-serial-number` 是在磁带库中唯一地标识卷的字母数字字符串。或使用命令
- `library-equipment-number:slot`，其中 `library-equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给自动化磁带库的设备序号，`slot` 是磁带在磁带库内所处的插槽地址。

在示例中，指明 LTO (`li`) 磁带 `VOL06E` 已从保管库调回、成功验证并返回了保管库：

```
root@solaris:~# chmed -I "validated and revaulted 20150310" li.VOL06A
```

确定驱动器在库中的安装顺序

如果自动化库包含多个驱动器，则这些驱动器在 `mcf` 文件中的顺序必须与库控制器看到它们的顺序相同。此顺序可以不同于在主机上看到这些设备的顺序和在主机的 `/var/adm/messages` 文件中报告这些设备的顺序。因此，只要配置了 Oracle Hierarchical Storage Manager 元数据服务器和数据移动器主机，更改了库或更改了库配置时，就应当执行下面列出的任务来检查驱动器顺序：

- [为库和 Solaris 主机收集驱动器信息](#)

- 将直接连接的库中的驱动器映射到 Solaris 设备名称或将 ACSLS 连接的库中的驱动器映射到 Solaris 设备名称（取决于要使用的设备）。

为库和 Solaris 主机收集驱动器信息

1. 查阅库文档。记下如何识别驱动器和目标。如果有本地操作面板，请了解可以如何使用它来确定驱动器顺序。
2. 如果库上装有一个本地操作面板，请使用它来确定驱动器连接到控制器的顺序。确定每个驱动器的 SCSI 目标标识符或全局名称。
3. 以 `root` 用户身份登录到 Solaris 主机。

```
root@solaris:~#
```

4. 列出 `/dev/rmt/` 中的 Solaris 逻辑设备名称，将输出重定向到一个文本文件。

在示例中，将 `/dev/rmt/` 中的列表重定向到 `root` 用户的主目录中的 `device-mappings.txt` 文件：

```
root@solaris:~# ls -l /dev/rmt/ > /root/device-mappings.txt
```

5. 现在，使用特定于您设备的过程将驱动器映射到 Solaris 设备名称：直接连接的磁带库或 ACSLS 连接的库。

将直接连接的库中的驱动器映射到 Solaris 设备名称

针对 `/dev/rmt/` 中列出的每个 Solaris 逻辑设备名称和库分配给 Oracle HSM 服务器主机的每个驱动器，执行以下过程：

1. 如果您尚未登录到 Oracle HSM Solaris 主机，请以 `root` 用户身份登录。

```
root@solaris:~#
```

2. 停止所有正在运行的归档过程，以便驱动器未在使用中。请参见“[使归档和回写进程闲置](#)”和“[停止归档和回写进程](#)”。
3. 在文本编辑器中，打开在“[为库和 Solaris 主机收集驱动器信息](#)”过程中创建的设备映射文件。将文件组织成简单的表并保存更改。

在后续步骤中，您需要参考此信息。在示例中，使用 `vi` 编辑器从 `/dev/rmt/` 列表中删除权限、所有权和日期属性，同时为库设备信息添加标题并预留空间：

```
root@solaris:~# vi /root/device-mappings.txt
LIBRARY SOLARIS          SOLARIS
DEVICE  LOGICAL          PHYSICAL
NUMBER  DEVICE            DEVICE
-----
/dev/rmt/0cbl -> ../../devices/pci@8.../st@w500104f00093c438,0:cbl
```

```

/dev/rmt/1cbn -> ../../devices/pci@8.../st@w500104f0008120fe,0:cbn
/dev/rmt/2cbn -> ../../devices/pci@8.../st@w500104f000c086e1,0:cbn
/dev/rmt/3cbn -> ../../devices/pci@8.../st@w500104f000b6d98d,0:cbn
:w

```

4. 在库中，确保所有驱动器都是空的。
5. 将一个磁带装入到磁带库中您尚未映射到 Solaris 逻辑设备名称的第一个驱动器中。

为了下面示例的需要，将一个 LTO4 磁带装入到 HP Ultrium LTO4 磁带机中。

6. 如果要映射磁带库中的驱动器，请确定与装入了磁带的驱动器对应的 Solaris `/dev/rmt/` 条目。在确定驱动器之前，请运行命令 `mt -f /dev/rmt/number status`，其中 `number` 标识 `/dev/rmt/` 中的驱动器。

在示例中，位于 `/dev/rmt/0` 的驱动器为空，但位于 `/dev/rmt/1` 的驱动器装有磁带。因此，磁带库标识为驱动器 1 的驱动器对应于 Solaris `/dev/rmt/1`：

```

root@solaris:~# mt -f /dev/rmt/0 status
/dev/rmt/0: no tape loaded or drive offline
root@solaris:~# mt -f /dev/rmt/1 status
HP Ultrium LTO 4 tape drive:
   sense key(0x0)= No Additional Sense   residual= 0   retries= 0
   file no= 0   block no= 3

```

7. 在您在前面的过程中创建的设备映射文件中，找到与装有磁带的 Solaris 设备对应的条目，并在所提供的空间中输入磁带库的设备标识符。然后保存文件。

在示例中，在 `/dev/rmt/1` 对应的行的 `LIBRARY DEVICE NUMBER` 字段中输入 1：

```

root@solaris:~# vi /root/device-mappings.txt
LIBRARY SOLARIS          SOLARIS
DEVICE LOGICAL          PHYSICAL
NUMBER DEVICE          DEVICE
-----
   /dev/rmt/0cbn -> ../../devices/pci@8.../st@w500104f00093c438,0:cbn
  1 /dev/rmt/1cbn -> ../../devices/pci@8.../st@w500104f0008120fe,0:cbn
   /dev/rmt/2cbn -> ../../devices/pci@8.../st@w500104f000c086e1,0:cbn
   /dev/rmt/3cbn -> ../../devices/pci@8.../st@w500104f000b6d98d,0:cbn
:w

```

8. 卸载磁带。
9. 重复该过程，直到设备映射文件中包含了将所有设备映射到 Solaris 逻辑设备名称的条目。然后保存文件并关闭编辑器。

```

root@solaris:~# vi /root/device-mappings.txt

```

```

LIBRARY SOLARIS          SOLARIS
DEVICE  LOGICAL          PHYSICAL
NUMBER  DEVICE           DEVICE
-----
  2    /dev/rmt/0cbn -> ../../devices/pci@8.../st@w500104f00093c438,0:cbn
  1    /dev/rmt/1cbn -> ../../devices/pci@8.../st@w500104f0008120fe,0:cbn
  3    /dev/rmt/2cbn -> ../../devices/pci@8.../st@w500104f000c086e1,0:cbn
  4    /dev/rmt/3cbn -> ../../devices/pci@8.../st@w500104f000b6d98d,0:cbn
:wq
root@solaris:~#

```

10. 在此处停止。保留映射文件供以后使用。

将 ACSLS 连接的库中的驱动器映射到 Solaris 设备名称

1. 如果您尚未登录到 Oracle HSM Solaris 主机，请以 *root* 用户身份登录。

```
root@solaris:~#
```

2. 停止所有正在运行的归档过程，以便驱动器未在使用中。请参见["使归档和回写进程闲置"](#)和["停止归档和回写进程"](#)。
3. 在文本编辑器中，打开在["为库和 Solaris 主机收集驱动器信息"](#)过程中创建的设备映射文件。将文件组织成简单的表。

在后续步骤中，您需要参考此信息。在示例中，使用 *vi* 编辑器从 */dev/rmt/* 列表中删除权限、所有权和日期属性，同时为库设备信息添加标题并预留空间：

```

root@solaris:~# vi /root/device-mappings.txt
SOLARIS LOGICAL DEVICE  DEVICE SERIAL NUMBER  ACSLS DEVICE ADDRESS
-----
/dev/rmt/0
/dev/rmt/1
/dev/rmt/2
/dev/rmt/3

```

4. 针对 */dev/rmt/* 中列出的每个逻辑设备名称，使用命令 *luxadm display /dev/rmt/number* 显示序列号，其中 *number* 标识 */dev/rmt/* 中的驱动器。

在示例中，获取了设备 */dev/rmt/0* 的序列号 *HU92K00200*：

```

root@solaris:~# luxadm display /dev/rmt/0
DEVICE PROPERTIES for tape: /dev/rmt/0
Vendor: HP
Product ID: Ultrium 4-SCSI
Revision: G25W

```

```
Serial Num: HU92K00200
...
Path status: Ready
root@solaris:~#
```

5. 然后，使用文本编辑器在 *device-mappings.txt* 文件的相应行中输入每个设备的序列号。

在示例中，使用 *vi* 编辑器在 *device-mappings.txt* 文件中记录设备 */dev/rmt/0* 的序列号：

```
root@solaris:~# vi /root/device-mappings.txt
SOLARIS LOGICAL DEVICE  DEVICE SERIAL NUMBER  ACSLS DEVICE ADDRESS
-----
/dev/rmt/0              HU92K00200
/dev/rmt/1
/dev/rmt/2
/dev/rmt/3
```

6. 对于映射到 */dev/rmt/* 的每个设备序列号，获取对应的 ACSLS 驱动器地址。使用 ACSLS 命令 *display drive * -f serial_num*。

在示例中，获取了设备 *HU92K00200* (*/dev/rmt/0*)、*HU92K00208* (*/dev/rmt/1*)、*HU92K00339* (*/dev/rmt/2*)、*HU92K00289* (*/dev/rmt/3*) 的 ACSLS 地址：

```
ACSSA> display drive * -f serial_num
2014-03-29 10:49:12 Display Drive
Acs  Lsm  Panel  Drive  Serial_num
0    2    10    16    331002031352
0    2    10    17    HU92K00200
0    2    10    18    HU92K00208
0    3    10    10    HU92K00339
0    3    10    11    HU92K00189
0    3    10    12    HU92K00289
root@solaris:~#
```

7. 使用文本编辑器在 *device-mappings.txt* 文件的相应行中输入每个序列号的 ACSLS 地址。保存文件并关闭编辑器。

在示例中，使用 *vi* 编辑器在 *device-mappings.txt* 文件中记录信息：

```
root@solaris:~# vi /root/device-mappings.txt
SOLARIS LOGICAL DEVICE  DEVICE SERIAL NUMBER  ACSLS DEVICE ADDRESS
-----
```

```

/dev/rmt/0          HU92K00200          (acs=0, lsm=2, panel=10, drive=17)
/dev/rmt/1          HU92K00208          (acs=0, lsm=2, panel=10, drive=18)
/dev/rmt/2          HU92K00339          (acs=0, lsm=2, panel=10, drive=10)
/dev/rmt/3          HU92K00289          (acs=0, lsm=2, panel=10, drive=12)
:wq
root@solaris:~#

```

- 在此处停止。保留映射文件供以后使用。

管理驱动器

您可以从 Oracle HSM 界面执行各种驱动器管理任务，包括以下任务：

- [装入和卸载驱动器](#)
- [清洗磁带机](#)
- [使用具备加密功能的驱动器](#)
- [处理驱动器问题。](#)

装入和卸载驱动器

当可移除介质存储在自动化库中时，文件系统归档和回写进程会根据需要自动将磁带装入驱动器中。但是，您还可以在管理可移除介质文件、备份 Oracle HSM 配置或恢复文件系统时按需装入磁带。本节介绍以下主题：

- [装入和卸载安装在自动化库中的驱动器](#)
- [手动装入和卸载独立驱动器](#)
- [必须手动装入卷时通知操作员。](#)

装入和卸载安装在自动化库中的驱动器

- [从指定的库位置装入驱动器](#)
- [使用指定的介质类型和卷序列号装入驱动器](#)
- [卸载库中的指定驱动器。](#)

从指定的库位置装入驱动器

使用命令 `samcmd load library-equipment-number:slot[:disk-side]`，其中 `library-equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给自动化磁带库的设备序号，`slot` 是目标卷在磁带库内所处的插槽地址，可选的 `disk-side` 值（1 或 2）指定双面磁光盘的其中一面。

磁带将装入到库中下一个可用的驱动器中。在示例中，将盒式磁带装入到磁带库 800 的插槽 11 中：

```
root@solaris:~# samcmd load 800:11
```

使用指定的介质类型和卷序列号装入驱动器

使用命令 `samcmd load mediatype.volume-serial-number`，其中 `mediatype` 是附录 A, 设备类型词汇表中列出的两字符介质类型代码之一，`volume-serial-number` 是在磁带库中唯一地标识卷的字母数字字符串。

磁带将装入到库中下一个可用的驱动器中。在示例中，装入 LTO (Li) 磁带 `VOL109`：

```
root@solaris:~# samcmd load li.VOL109
```

卸载库中的指定驱动器

使用命令 `samcmd unload drive-equipment-number`，其中 `drive-equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给驱动器的设备序号。

磁带将被卸载，即使驱动器处于 `unavail` 状态。在示例中，卸载驱动器 `801`：

```
root@solaris:~# samcmd unload 801]
```

手动装入和卸载独立驱动器

Oracle HSM 软件将独立的可移除介质驱动器视为具有自己的目录的小型单插槽库。

向独立驱动器中装入磁带

要装入独立驱动器，请按生产商提供的说明将磁带放到驱动器中。Oracle HSM 系统将识别装入的磁带，读取标签并为驱动器更新目录。

从独立驱动器中卸载磁带

要卸载独立驱动器，请执行如下操作：

1. 使驱动器闲置。使用命令 `samcmd idle drive-equipment-number`，其中 `drive-equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给驱动器的设备序号。

使驱动器闲置时，Oracle HSM 软件将完成在使用驱动器的任何当前归档进程且不启动任何新进程。

```
root@solaris:~# samcmd idle 801
```

2. 等待 Oracle HSM 完成并将驱动器转换为 `off` 状态。

可以使用命令 `samcmd r` 检查驱动器的状态。

3. 根据供应商提供的说明移除磁带。

4. 在此处停止。

必须手动装入卷时通知操作员

如果您在使用独立驱动器，或者将必需的磁带存储在保管库中或存储在库外的某个其他位置，则当 Oracle HSM 软件需要操作员装入非驻留磁带时，它可以向指定的地址发送电子邮件。要启用此功能，请遵循以下过程：

启用装入通知

1. 以 *root* 用户身份登录到文件系统主机。

```
root@solaris:~#
```

2. 将文件 *load_notify.sh* 从目录 */opt/SUNWsamfs/examples/* 复制到目录 */etc/opt/SUNWsamfs/scripts/*。

请注意，下面的命令是作为单行输入的一使用反斜杠字符对换行符进行转义：

```
root@solaris:~# cp /opt/SUNWsamfs/examples/load_notify.sh /  
/etc/opt/SUNWsamfs/scripts/  
root@solaris:~#
```

3. 在文本编辑器中打开文件 */etc/opt/SUNWsamfs/defaults.conf*。搜索指令 *exported_media*。根据需要，取消注释该行或添加该行，并将其值设置为 *exported_media=available*。

在示例中，使用 *vi* 编辑器：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf  
# These are the defaults.  
...  
exported_media=available
```

4. 在文件 */etc/opt/SUNWsamfs/defaults.conf* 中，搜索指令 *attended*。如有必要，取消注释该行或添加该行。将其值设置为 *attended=yes*。保存文件并关闭编辑器。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf  
# These are the defaults. ...  
# These are the defaults. ...  
exported_media=available  
attended=yes  
:wq  
root@solaris:~#
```


5. 在文本编辑器中打开文件 `/etc/opt/SUNWsamfs/scripts/load_notify.sh`。找到通知电子邮件的默认收件人 `root`。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/scripts/load_notify.sh
#       Notify operator to load volume.
...
# Change the email address on the following line to send email to
# the appropriate recipient.
/bin/ppriv -s I=basic -e /usr/bin/mailx -s "SAM-FS needs VSN $5" root <<EOF
...
```

6. 在文件 `/etc/opt/SUNWsamfs/scripts/load_notify.sh` 中，将通知电子邮件的收件人从默认收件人 `root` 更改为负责非驻留卷的操作员的电子邮件地址。保存文件并关闭编辑器。

在示例中，将收件人更改为 `tapetech`：

```
#       Notify operator to load volume.
...
/bin/ppriv -s I=basic -e /usr/bin/mailx -s "SAM-FS needs VSN $5" tapetech <<EOF
...
:wq
root@solaris:~#
```

7. 重新初始化 Oracle HSM 软件。使用 `sam-fsd` 命令。

`sam-fsd` 是一个读取 Oracle HSM 配置文件的初始化命令。该命令会在遇到以下错误时停止：

```
root@solaris:~# sam-fsd
```

8. 指示 Oracle HSM 软件重新读取 `mcf` 文件并相应地重新配置文件系统和硬件。使用命令 `samd config`：

```
root@solaris:~# samd config
```

9. 在此处停止。

清洗磁带机

现代的 Oracle StorageTek T10000D 和线性磁带开放格式 (Linear Tape Open, LTO) 磁带机可以自我监视并根据需要请求清洗。Oracle Hierarchical Storage Manager 软件接受这些请求并在需要时自动装入清洗磁带。因此，在大多数情况下，您只需确保磁带库中包含足够的、Oracle HSM 能够找到的清洗磁带。

当驱动器请求的清洗不可行时，可以手动启动清洗。但是，请注意：大多数生产商强烈反对在驱动器未发出请求的情况下进行例行清洗。清洗磁带有磨损。过度使用可能会损坏驱动器和介质。因此，请谨慎行事并遵循生产商的建议。

本节的其余部分提供以下任务的说明：

- [提供足够的清洗磁带](#)
- [启用自动化磁带机清洗（建议配置）](#)
- [手动清洗磁带机。](#)

提供足够的清洗磁带

1. 以 `root` 用户身份登录到文件系统主机。

```
root@solaris:~#
```

2. 如果您打算配置自动清洗（建议配置）并且您的磁带库具有两个以上驱动器，请确保为列出了磁带库中磁带的每个文件系统目录至少提供两个清洗磁带。

如果当驱动器需要清洗时没有清洗磁带可用，则 Oracle HSM 软件会将驱动器状态设置为 `down`，直到清洗完成。

3. 将清洗磁带置于库中转槽（也称为磁带存取口）中。
4. 将清洗磁带导入自动化磁带库中。使用命令 `samimport library-equipment-number`，其中 `library-equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给库的设备序号。

在本示例中，我们将清洗磁带置于库 `800` 的中转槽中并将其导入库：

```
root@solaris:~# samimport 800
```

5. 如果清洗磁带标签为 `CLEAN` 或者以字母 `CLN` 开头，请在此处停止。

Oracle HSM 软件将识别清洗磁带并将其从邮箱移动到存储插槽中。Oracle HSM 将更新磁带库目录，设置清洗介质标志，并将访问计数设置为介质类型建议的最大清洗次数（每使用磁带清洗一次驱动器，此计数便减少一次）。

6. 如果磁带没有标签，请将其标记为清洗介质。使用命令 `chmed +C library-equipment-number:slot`，其中 `library-equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给库的设备序号，`slot` 是清洗磁带在库内的位置。

在示例中，在磁带库 `800` 的插槽 `31` 中的磁带上设置 `C`（清洗介质）标志。

```
root@solaris:~# chmed +C 800:31
```

7. 将访问计数设置为介质类型建议的最大清洗次数。使用命令 `chmed -count cleanings library-equipment-number:slot`，其中：

- *cleanings* 是制造商建议的每个磁带的最大清洗次数。
- *library-equipment-number* 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给库的设备序号。
- *slot* 是清洗磁带在磁带库内的位置。

每使用磁带清洗一次驱动器，清洗计数便减少一次。在示例中，将计数设置为最大清洗次数 50，这是针对磁带库 800 中使用的 LTO (*Li* 类型) 清洗磁带建议的最大次数：

```
root@solaris:~# chmed -count 50 800:31
```

8. 接着，启用自动化磁带机清洗（建议配置）或在此处停止。

启用自动化磁带机清洗（建议配置）

1. 以 *root* 用户身份登录到文件系统主机。

```
root@solaris:~#
```

2. 如果磁带库中包含您希望使用的自动清洗功能，请根据磁带库制造商的建议配置该功能。在此处停止。

现在，当驱动器请求清洗时，磁带库将自动提供所需的清洗介质。

3. 如果磁带库中包含您不希望使用的自动清洗功能，请根据制造商的建议禁用该功能。
4. 在文本编辑器中打开文件 `/etc/opt/SUNWsamfs/defaults.conf`，然后启用 Oracle HSM 自动清洗。添加行 `tapeclean = all autoclean on logsense on`。然后保存文件并关闭编辑器。

在示例中，使用 *vi* 编辑器：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults. ...
#sef = all on once
...
tapeclean = all autoclean on logsense on
:wq
root@solaris:~#
```

5. 重新初始化 Oracle HSM 软件。使用 `sam-fsd` 命令。

`sam-fsd` 是一个读取 Oracle HSM 配置文件的初始化命令。该命令会在遇到以下错误时停止：

```
root@solaris:~# sam-fsd
```

6. 指示 Oracle HSM 软件重新读取 *mcf* 文件并相应地重新配置文件系统和硬件。使用命令 *samd config*:

```
root@solaris:~# samd config
```

7. 在此处停止。

手动清洗磁带机

1. 在继续操作之前，请查看驱动器生产商提供的手动清洗指南。

请谨慎行事。过于频繁的清洗是驱动器损坏的常见原因。许多制造商现在强烈反对进行例行或调度的清洗。因此，请确保您了解驱动器何时需要清洗。

2. 监视设备日志以查找表明驱动器需要清洗的征兆。在 */var/opt/SUNwsamfs/devlog/* 目录中针对每个 *drive-equipment-number* 都有一个日志，其中 *drive-equipment-number* 是 */etc/opt/SUNwsamfs/mcf* 文件中分配给磁带库的设备序号。
3. 监视系统日志文件 */var/adm/messages* 以了解设备错误。
4. 清洗磁带机。使用命令 *cleandrive drive-equipment-number*。

在示例中，清洗驱动器 802:

```
root@solaris:~# cleandrive 802
```

5. 在此处停止。

使用具备加密功能的驱动器

如果要将文件归档到具有加密功能的驱动器，请在规划归档操作时考虑以下几点:

- 请勿在库中混用具备加密功能的驱动器和不具备加密功能的驱动器。
- 驱动器启用了加密之后，便无法禁用加密。
- 请勿在磁带卷上混用加密和非加密文件。
- 具备加密功能的驱动器无法将加密的文件附加到包含非加密数据的磁带卷。
- 具有加密功能的驱动器可以读取非加密数据。

有关详细信息，请查阅驱动器和加密密钥管理系统的相关文档。

处理驱动器问题

通常，您需要根据供应商的建议来处理驱动器问题。但可能需要首先执行以下一个或两个任务，然后才可以开始驱动器维护、故障排除或修复:

- [使驱动器脱机以进行维护或修复](#)

- 在解决驱动器问题后将介质返回到库中

使驱动器脱机以进行维护或修复

1. 以 `root` 用户身份登录到文件系统主机。

```
root@solaris:~#
```

2. 停止归档和回写进程，如“[使归档和回写进程闲置](#)”中所述。
3. 停止归档和回写进程，然后使驱动器脱机。使用“[停止归档和回写进程](#)”过程。
4. 执行供应商指定的维护、诊断或修复过程。

例如，在尝试移除卡住的磁带时，请务必查看供应商的建议。采用不当的方法移除卡住的磁带可能会损坏磁带和驱动器。

5. 驱动器重新运行后，使磁带库和驱动器联机，然后重新启动归档和回写进程。使用“[重新启动归档和回写进程](#)”过程。
6. 在此处停止。

在解决驱动器问题后将介质返回到库中

如果驱动器因为装入到其中的介质而发生问题，则在修复过程中您可能需要手动移除介质。这可能会导致目录不一致。因此，请遵循以下相应过程：

将介质返回尚未执行自动审计的磁带库

在进行修复后重新将磁带库和驱动器联机时，要将介质返回未执行自动审计的磁带库，请执行如下操作：

1. 手动将磁带返回其存储插槽。

在这种情况下，目录尚未更新，将继续列出磁带库内容中的磁带。因此您通过将磁带重新放入其以前占用的同一插槽来更正差异。

2. 更新 Oracle HSM 目录以显示该插槽已被重新占用。使用命令 `chmed library-equipment-number:slot`，其中 `slot` 是插槽在库内的地址。

在示例中，更新库 `800` 的插槽 `42` 的状态：

```
root@solaris:~# chmed +o 800:42
root@solaris:~#
```

3. 在此处停止。

自动审计后将介质返回磁带库

如果当在修复操作后重新将库和驱动器联机时库执行自动审计，请执行如下操作：

1. 将磁带放置在库中转槽中。
2. 将磁带导入库中。使用命令 `samimport library-equipment-number`。

在这种情况下，审计已对目录进行了协调，因此目录不再将该磁带列在库中。因此，导入磁带会同时将其添加到库和 Oracle HSM 目录中。在示例中，将磁带置于库 800 的邮槽中并将其导入库中。

```
root@solaris:~# samimport 800
```

3. 在此处停止。

管理可移除介质

本节介绍以下主题：

- [为可移除介质设置标签](#)
- [维护数据完整性](#)

为可移除介质设置标签

注意：

为磁带设置标签或重新设置标签会导致磁带中的数据永久不可访问。只有当您确定不再需要磁带中存储的数据时，才应为磁带重新设置标签。

设置标签进程会在记录介质上写入标识信息，并进行初始化以供使用（有关完整信息，请参见 ANSI X3.27-1987 *File Structure and Labeling of Magnetic Tapes for Information Interchange*（用于信息交换的磁带的文件结构和标签设置））。

需要为介质设置标签时，请选择以下相应过程：

- [基于条码生成标签](#)
- [为新磁带设置标签或者为现有磁带重新设置标签](#)
- [为新光盘设置标签或为现有光盘重新设置标签。](#)

基于条码生成标签

要使用磁带上的条形码所派生的卷序列号 (volume serial number, VSN) 自动为已启用写入的无标签磁带设置标签，请执行如下操作。

1. 确保所有条形码均可读。
2. 以 `root` 用户身份登录到文件系统主机。

```
root@solaris:~#
```

3. 在文本编辑器中打开 `/etc/opt/SUNWsamfs/defaults.conf` 文件。

在示例中，使用 *vi* 编辑器查看该文件：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.
...
```

4. 如果需要通过相应条形码中的前六个字符生成卷序列号 (volume serial number, VSN)，请首先查看是否已将 Oracle HSM 设置为默认值 *barcodes*。在 *defaults.conf* 文件中，找到 *labels* 指令行（如果存在）。如果 *labels* 指令设置为 *barcodes*、已注释掉或在文件中不存在，则将 Oracle HSM 设置为默认值 *barcodes*。

在本示例中，*defaults.conf* 文件包含行 *#labels = barcodes*：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.
...
#labels = barcodes
root@solaris:~#
```

5. 如果需要通过相应条形码中的前六个字符生成卷序列号 (volume serial number, VSN) 并且已将 Oracle HSM 设置为默认值，则关闭 *defaults.conf* 文件，而不进行任何更改。在此处停止。

将 *labels* 指令设置为 *barcodes* 时，软件会通过相应条形码中的前六个字符自动生成所需的卷序列号 (volume serial number, VSN)。在本示例中，Oracle HSM 使用默认设置。因此我们关闭 *vi* 编辑器，而不保存文件：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.
...
#labels = barcodes
:q
root@solaris:~#
```

6. 否则，如果需要通过相应条形码中的前六个字符生成卷序列号 (volume serial number, VSN)，则输入 *labels = barcodes*，注释掉非默认指令，或删除非默认指令即可。然后保存文件并关闭编辑器。

在本示例中，已将指令设置为非默认值 *barcodes_low*。因此我们注释掉非默认行。我们插入行 *labels = barcodes*。保存修改后的文件并关闭编辑器：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.
...
```

```
#labels = barcodes_low
labels = barcodes
:q
root@solaris:~#
```

7. 如果需要通过磁带条形码的后六个字符生成卷序列号 (volume serial number, VSN), 请将 *labels* 参数的值设置为 *barcodes_low*。保存文件并关闭编辑器。

在本示例中, 我们插入行 *labels = barcodes_low*, 保存文件, 然后关闭编辑器:

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.
...
labels = barcodes_low
:wq
root@solaris:~#
```

8. 如果您编辑了 *defaults.conf* 文件, 请运行 *sam-fsd* 命令。

sam-fsd 是一个读取 Oracle HSM 配置文件的初始化命令。该命令会在遇到以下错误时停止:

```
root@solaris:~# sam-fsd
```

9. 如果您编辑了 *defaults.conf* 文件, 请指示 Oracle HSM 软件重新读取 *mcf* 文件并相应地重新配置自身。使用命令 *samd config*。

```
[metadata-server]root@solaris:~# samd config
```

10. 在此处停止。

为新磁带设置标签或者为现有磁带重新设置标签

注意:

为磁带设置标签或重新设置标签会导致磁带中的数据永久不可访问。只有当您确定不再需要磁带中存储的数据时, 才应为磁带重新设置标签。

1. 以 *root* 用户身份登录到文件系统主机。

```
root@solaris:~#
```

2. 要为已装入到驱动器中的新磁带设置标签, 请使用命令 *tplabel -new volume-serial-number drive-equipment-number*, 其中:
 - *volume-serial-number* 是所需的卷序列号。

- *drive-equipment-number* 是 */etc/opt/SUNWsamfs/mcf* 文件中分配给驱动器的设备序号。

在示例中，将卷序列号 *VOL600* 分配给驱动器 *803* 中的新磁带：

```
root@solaris:~# tplabel -new -vsn VOL600 803
root@solaris:~#
```

3. 要为位于自动化介质库中的新磁带设置标签，请使用命令 *tplabel -new volume-serial-number library-equipment-number:slot*，其中：
 - *volume-serial-number* 是所需的卷序列号。
 - *library-equipment-number* 是 */etc/opt/SUNWsamfs/mcf* 文件中分配给驱动器的设备序号。
 - *slot* 是磁带在磁带库内的位置。

在示例中，将卷序列号 *VOL601* 分配给磁带库 *800* 的插槽 *19* 中的新磁带：

```
root@solaris:~# tplabel -new -vsn VOL601 800:19
root@solaris:~#
```

4. 要为装入到驱动器中的磁带重新设置标签，请使用命令 *tplabel -old old-volume-serial-number -new new-volume-serial-number drive-equipment-number*，其中：
 - *volume-serial-number* 是所需的卷序列号。
 - *drive-equipment-number* 是 */etc/opt/SUNWsamfs/mcf* 文件中分配给驱动器的设备序号。

在本示例中，我们重新初始化驱动器 *804* 中的盒式磁带，将旧卷序列号 *AZ0001* 替换为新卷序列号 *VOL120*：

```
root@solaris:~# tplabel -old AZ0001 -vsn VOL120 804
root@solaris:~#
```

5. 要为位于磁带库中的磁带重新设置标签，请使用命令 *tplabel -old old-volume-serial-number -new new-volume-serial-number library-equipment-number:slot*，其中：
 - *volume-serial-number* 是所需的卷序列号。
 - *library-equipment-number* 是 */etc/opt/SUNWsamfs/mcf* 文件中分配给驱动器的设备序号。
 - *slot* 是磁带在磁带库内的位置。

如果需要，您可以重用现有的卷序列号。在本示例中，我们重新初始化库 *800* 的插槽 *23* 中的盒式磁带（通过将其重新标记为其现有卷序列号 *VOL121*）：

```
root@solaris:~# tplabel -old VOL601 -vsn VOL601 800:23
root@solaris:~#
```

6. 在此处停止。

为新光盘设置标签或为现有光盘重新设置标签

1. 以 *root* 用户身份登录到文件系统主机。

```
root@solaris:~#
```

2. 要为装入到驱动器中的新光学磁带设置标签，请使用命令 `odlabel -new volume-serial-number drive-equipment-number[:side]`，其中：

- *volume-serial-number* 是所需的卷序列号。
- *drive-equipment-number* 是 `/etc/opt/SUNwsamfs/mcf` 文件中分配给驱动器的设备序号。
- *side* (可选) 是双面磁盘的指定面。

在示例中，将卷序列号 *OD1700* 分配给驱动器 *701* 中的新单面光学磁带：

```
root@solaris:~# odlabel -new -vsn OD1700 701
root@solaris:~#
```

3. 要为位于自动化介质库中的新光学磁带设置标签，请使用命令 `odlabel -new volume-serial-number library-equipment-number:slot[:side]`，其中：

- *volume-serial-number* 是所需的卷序列号。
- *library-equipment-number* 是 `/etc/opt/SUNwsamfs/mcf` 文件中分配给驱动器的设备序号。
- *slot* 是磁带在磁带库内的位置，*side* (可选) 是双面磁盘的指定面。

在示例中，将卷序列号 *OD1701* 分配给库 *700* 的插槽 *42* 中的新双面光学磁带的第 *2* 面。

```
root@solaris:~# odlabel -new -vsn OD1701 700:42:2
root@solaris:~#
```

4. 要为装入到驱动器中的光学磁带重新设置标签，请使用命令 `odlabel -old old-volume-serial-number -new new-volume-serial-number drive-equipment-number[:side]`，其中：

- *volume-serial-number* 是所需的卷序列号。
- *drive-equipment-number* 是 `/etc/opt/SUNwsamfs/mcf` 文件中分配给驱动器的设备序号。
- *side* (可选) 是双面磁盘的指定面。

在本示例中，我们重新初始化驱动器 702 中的光学磁带，将旧卷序列号 0D1120 替换为新卷序列号 0D1120：

```
root@solaris:~# odlabel -old OD0001 -vsn OD1120 702
root@solaris:~#
```

5. 要为位于自动化介质库中的现有光学磁带重新设置标签，请使用命令 `odlabel -old volume-serial-number library-equipment-number:slot[:side]`，其中：
 - `volume-serial-number` 是所需的卷序列号。
 - `library-equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给库的设备序号。
 - `side` (可选) 是双面磁盘的指定面。

如果需要，您可以重用现有的卷序列号。在本示例中，我们重新初始化库 700 的插槽 23 中的光学磁带（通过将其重新标记为其现有卷序列号 0D1121）：

```
root@solaris:~# odlabel -old OD1121 -vsn OD1121 800:23
root@solaris:~#
```

6. 在此处停止。

维护数据完整性

为了维护可移除的磁带介质上所存储数据文件的完整性，Oracle Hierarchical Storage Manager 软件既提供了按需工具又提供了自动化工具。本节介绍以下主题：

- [显示数据完整性验证 \(Data Integrity Validation, DIV\) 设置和状态](#)
- [检查给定磁带卷的完整性](#)
- [监视自动化完整性验证](#)

显示数据完整性验证 (Data Integrity Validation, DIV) 设置和状态

本节介绍以下任务：

- [显示 DIV 设置](#)
- [监视归档文件的“写入后验证”状态](#)
- [监视设备的“写入后验证”状态](#)

显示 DIV 设置

要显示数据完整性验证 (Data Integrity Validation, DIV) 设置，请使用命令 `samcmd L` 并将输出通过管道发送到 Solaris `grep` 命令和正则表达式 `div`。

在示例中，DIV 处于 *OFF* 状态：

```
root@solaris:~# samcmd L | grep div
div                OFF
root@solaris:~#
```

监视归档文件的“写入后验证”状态

要监视归档文件在归档期间的验证状态，请使用 *samu* 接口。使用命令 *samu -d a*。

```
root@solaris:~# samu -d a
Archiver status                samu 5.4                22:22:31 Mar 4 2014
sam-archiverd: Archiving files
sam-arfind: hsmfs1 mounted at /hsm/hsmfs1
Files waiting to start      12,576 schedule      26,695 archiving      13,120
...
sam-arcopy: qfs.arset1.2.344 ti.TKC960
Verifying archive file at position 1175
```

监视设备的“写入后验证”状态

要监视设备在归档期间的验证状态，请使用 *samu* 接口。使用命令 *samu -d s*：

```
root@solaris:~# samu -d s
Device status                samu 5.4                22:27:53 Mar 4 2014
ty      eq state  device_name                fs  status
sn      800 on    /dev/scsi/changer/c1t2d0    800 n-----r
ti      801 on    /dev/rmt/0cbn              800 -----p
...
hy      805 on    historian                    805 -----
ti      91 on    /dev/rmt/4cbn              90  -l----oVr
Verify averaging 240.9 MB/s
```

检查给定磁带卷的完整性

当需要按需验证特定磁带卷的数据完整性时，请使用 Oracle HSM *tpverify* 命令。*tpverify* 命令支持 Oracle T10000C/D、LTO 和其他常用的介质。T10000C/D 介质使用 Oracle 数据完整性验证功能进行验证。其他格式使用广泛支持的 SCSI *verify(6)* 命令进行检查。

下面各节概述了 *tpverify* 的一些使用方式。有关语法和选项的完整详细信息，请参见 *tpverify* 手册页：

- [验证通过磁带库位置指定的磁带上的数据](#)
- [验证通过介质类型和卷序列号指定的磁带上的数据](#)
- [验证使用指定驱动器的磁带上的数据](#)
- [从磁带开头处重新开始数据验证](#)
- [验证 T10000C/D 磁带上所有块的 ECC](#)
- [验证 T10000C/D 磁带上所有块的 ECC 和 DIV 校验和](#)

- [重新构建 T1000C/D 磁带的介质信息区域 \(Media Information Region, MIR\)](#)
- [取消指定磁带的验证](#)
- [显示磁带的 DIV 状态和验证进度](#)

验证通过磁带库位置指定的磁带上的数据

使用命令 `tpverify library-equipment-number:slot`，其中 `library-equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给自动化磁带库的设备序号，`slot` 是目标卷在磁带库内所处的插槽地址。

`tpverify` 命令通过检查磁带库介质目录来定位之前验证的最后磁带位置。然后，它将磁带装入第一个可用的驱动器并从它最后停止的点开始验证，并使用默认方法检查：对于 T1000C/D 介质，使用 `tpverify` 标准方法；对于其他介质，使用 `SCSI verify(6)`。标准方法在速度方面进行了优化，并对 Oracle HSM 介质的边缘、开头、末尾和前 1,000 个块进行验证。

在示例中，使用标准方法验证存储在磁带库 800 的插槽 9 中的 T1000D 磁带：

```
root@solaris:~# tpverify 800:9
```

验证通过介质类型和卷序列号指定的磁带上的数据

使用命令 `tpverify mediatype.volume-serial-number`，其中 `mediatype` 是[附录 A, 设备类型词汇表](#)中列出的两字符介质类型代码之一，`volume-serial-number` 是在磁带库中唯一地标识卷的字母数字字符串。

`tpverify` 命令通过检查磁带库介质目录来定位之前验证的最后磁带位置。然后，它将磁带装入第一个可用的驱动器并从它最后停止的点开始验证，并使用默认方法检查：对于 T1000C/D 介质，使用 `tpverify` 标准方法；对于其他介质，使用 `SCSI verify(6)`。

在示例中，使用 `SCSI verify(6)` 命令验证 LTO (1i) 卷 `VOL006`：

```
root@solaris:~# tpverify li.VOL006
```

验证使用指定驱动器的磁带上的数据

使用命令 `tpverify library-equipment-number:slot device-equipment-number`，其中：

- `library-equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给自动化磁带库的设备序号。
- `slot` 是目标卷在磁带库内所处的插槽地址。

- *device-equipment-number* 是 */etc/opt/SUNWsamfs/mcf* 文件中分配给驱动器的设备序号。

在示例中，验证存储在磁带库 800 的插槽 17 中使用驱动器 803 的 T10000D 磁带：

```
root@solaris:~# tpverify 800:17 803
```

从磁带开头处重新开始数据验证

使用命令 *tpverify -a library-equipment-number:slot* 或 *tpverify -a mediatype.volume-serial-number*，其中：

- *library-equipment-number* 是 */etc/opt/SUNWsamfs/mcf* 文件中分配给自动化磁带库的设备序号。
- *slot* 是目标卷在磁带库内所处的插槽地址。
- *mediatype* 为附录 A, [设备类型词汇表](#) 中列出的由两个字符表示的介质类型代码之一。
- *volume-serial-number* 是在库中唯一地标识卷的字母数字字符串。

-a 选项覆盖默认行为并忽略介质目录中记录的位置，从介质开头处开始验证。

在示例中，从磁带开头处验证 LTO (*li*) 卷 *VOL016*：

```
root@solaris:~# tpverify -a li.VOL016
```

验证 T10000C/D 磁带上所有块的 ECC

使用命令 *tpverify -C library-equipment-number:slot* 或 *tpverify -C mediatype.volume-serial-number*，其中：

- *library-equipment-number* 是 */etc/opt/SUNWsamfs/mcf* 文件中分配给自动化磁带库的设备序号。
- *slot* 是目标卷在磁带库内所处的插槽地址。
- *mediatype* 为附录 A, [设备类型词汇表](#) 中列出的由两个字符表示的介质类型代码之一。
- *volume-serial-number* 是在库中唯一地标识卷的字母数字字符串。

tpverify 命令通过检查磁带库介质目录来定位之前验证的最后磁带位置。然后，使用由 *-c* 选项指定的“完整”(Complete) 方法从最后停止的点开始验证。“完整”方法比标准方法更为全面，但同时也明显更慢。该方法将检查介质上所有块的错误校正代码 (Error Correction Code, ECC)。

在示例中，使用“完整”方法验证 T10000D (*ti*) 卷 *VOL516*：

```
root@solaris:~# tpverify -C ti.VOL516
```

验证 T1000C/D 磁带上所有块的 ECC 和 DIV 校验和

使用命令 `tpverify -P library-equipment-number:slot` 或 `tpverify -P mediatype.volume-serial-number`，其中：

- `library-equipment-number` 是 `/etc/opt/SUNwsamfs/mcf` 文件中分配给自动化磁带库的设备序号。
- `slot` 是目标卷在磁带库内所处的插槽地址。
- `mediatype` 为附录 A, [设备类型词汇表](#) 中列出的由两个字符表示的介质类型代码之一。
- `volume-serial-number` 是在库中唯一地标识卷的字母数字字符串。

`tpverify` 命令通过检查磁带库介质目录来定位之前验证的最后磁带位置。然后，使用由 `-P` 选项指定的“完整加强版”(Complete Plus) 方法从最后停止的点开始验证。“完整加强版”方法十分全面，但也比其他方法更慢。该方法将检查介质上所有块的错误校正代码 (Error Correction Code, ECC) 和数据完整性验证校验和。

在示例中，使用“完整加强版”方法验证 T1000D (*ti*) 卷 VOL521:

```
root@solaris:~# tpverify -P ti.VOL521
```

重新构建 T1000C/D 磁带的介质信息区域 (Media Information Region, MIR)

使用命令 `tpverify -M library-equipment-number:slot` 或 `tpverify -M mediatype.volume-serial-number`，其中：

- `library-equipment-number` 是 `/etc/opt/SUNwsamfs/mcf` 文件中分配给自动化磁带库的设备序号。
- `slot` 是目标卷在磁带库内所处的插槽地址。
- `mediatype` 为附录 A, [设备类型词汇表](#) 中列出的由两个字符表示的介质类型代码之一。
- `volume-serial-number` 是在库中唯一地标识卷的字母数字字符串。

`tpverify` 命令在 Oracle StorageTek 盒式磁带上重新构建缺少或已损坏的介质信息区域 (media information region, MIR)，即使该磁带在介质目录中标记为错误时也是如此。检测到 MIR 损坏时将自动指定重新构建操作。

在示例中，使用“MIR 重新构建”方法验证 T1000D (*ti*) 卷 VOL523:

```
root@solaris:~# tpverify -M ti.VOL523
```

取消指定磁带的的数据验证

使用命令 `tpverify -c library-equipment-number:slot` 或 `tpverify -c mediatype.volume-serial-number`，其中：

- `library-equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给自动化磁带库的设备序号。
- `slot` 是目标卷在磁带库内所处的插槽地址。
- `mediatype` 为附录 A, [设备类型词汇表](#) 中列出的由两个字符表示的介质类型代码之一。
- `volume-serial-number` 是在库中唯一地标识卷的字母数字字符串。

`tpverify -c` 命令取消当前验证操作并将磁带上最后验证的位置记录在介质目录中。因此，您可以停止某个验证作业以释放驱动器或卷来进行归档或回写，稍后再从同一个点继续验证。

在示例中，取消 T10000D (*ti*) 卷 VOL533 的验证：

```
root@solaris:~# tpverify -c ti.VOL523
```

显示磁带的 DIV 状态和验证进度

使用命令 `itemize -2 library-equipment-number`，其中 `library-equipment-number` 是 `/etc/opt/SUNWsamfs/mcf` 文件中分配给自动化磁带库的设备序号。

`itemize -2` 命令将指定磁带库中的介质编入目录并列每个卷的 DIV 状态和验证进度。

在示例中，显示设备序号为 800 的库中的卷的验证状态。`lvtime`（最后验证时间）字段显示 `tpverify` 最后完成磁带的完整验证的时间。`status` 字段值为 `div` 表示磁带支持 DIV，值为 `none` 表示磁带不支持 DIV。`lvpos`（最后验证位置）字段显示 `tpverify` 最后在何处取消以及再次运行时将从何处开始。

```
root@solaris:~# itemize -2 800
Robot VSN catalog: eq: 800          count: 60
slot  access_time  count  use  ty  vsn
      lvtime      status
  0   Apr 2 16:34    6   0%  ti  VOL519
      Apr 2 09:23   div
  1   Apr 2 16:17   28  29%  ti  VOL510
      Apr 2 16:17   div
      0x9bb9
  2   none          0   0%  ti  VOL511
      none          none
      0
...
root@solaris:~#
```

监视自动化完整性验证

定期介质验证是 `tpverify` 命令的自动化形式。本节提供有关可能偶尔需要执行的维护任务的说明。这些任务包括：

- 查看和验证 `verifyd.cmd` 配置文件
- 重新加载 `verifyd.cmd` 配置文件
- 显示所有定期介质验证磁带缺陷数据库中列出的缺陷
- 显示针对特定卷列出的缺陷
- 清除定期介质验证磁带缺陷数据库中列出的缺陷。

有关配置定期介质验证的说明，请参见客户文档库 (<http://docs.oracle.com/en/storage/#sw>) 中的《Oracle Hierarchical Storage Manager and StorageTek QFS Software 安装和配置指南》。

查看和验证 `verifyd.cmd` 配置文件

要随时查看 `verifyd.cmd` 文件或者在编辑后验证该文件，请使用命令 `tpverify -x`。

`tpverify -x` 命令检查 `/etc/opt/SUNWsamfs/verifyd.cmd` 文件并指出错误或显示文件的内容。

```
root@solaris:~# tpverify -x
Reading '/etc/opt/SUNWsamfs/verifyd.cmd'.
PMV: off
    Run-time:
    Start Time: 2200
End Time: 0500
PMV Scan: all
PMV Method: Standard
STA Scan: off
Action: none
PMV VSNS: all
PMV Policy:
    Last Verified Age: 6m
root@solaris:~#
```

重新加载 `verifyd.cmd` 配置文件

要在不停止验证过程的情况下重新加载 `verifyd.cmd` 文件，请使用命令 `tpverify -r`。

```
root@solaris:~# tpverify -r
root@solaris:~#
```

显示所有定期介质验证磁带缺陷数据库中列出的缺陷

要列出定期介质验证已查明并存储在磁带缺陷数据库中的所有缺陷，请使用命令 `tpverify -l`。

在示例中，数据库中没有缺陷：

```
root@solaris:~# tpverify -l
```

```
No defects found.  
root@solaris:~#
```

显示针对特定卷列出的缺陷

要列出已在特定卷上确定的所有缺陷，请使用命令 `tpverify -l mediatype.volume-serial-number`，其中：

- `mediatype`（可选）是附录 A, 设备类型词汇表中列出的两字符介质类型代码之一。
- `volume-serial-number` 是在库中唯一地标识卷的字母数字字符串。

在示例中，数据库中没有针对 LTO (*ti*) 卷 `VOL514` 列出的缺陷：

```
root@solaris:~# tpverify -l ti.VOL514  
No defects found.  
root@solaris:~#
```

清除定期介质验证磁带缺陷数据库中列出的缺陷

要从磁带缺陷数据库中删除定期介质验证已查明的所有缺陷，请使用命令 `tpverify -d`。

要删除针对特定卷列出的所有缺陷，请使用命令 `tpverify -d mediatype.volume-serial-number`，其中：

- `mediatype`（可选）是附录 A, 设备类型词汇表中列出的两字符介质类型代码之一。
- `volume-serial-number` 是在库中唯一地标识卷的字母数字字符串。

```
root@solaris:~# tpverify -d  
root@solaris:~# tpverify -d ti.VOL514  
root@solaris:~#
```

第 6 章 为数字保留管理归档

到目前为止，本文档讨论了将 Oracle Hierarchical Storage Manager and StorageTek QFS Software 解决方案作为普通的 UNIX 文件系统进行管理，在这些文件系统中，用户和应用程序会定期创建、修改和删除文件。重点关注磁盘高速缓存，归档主要充当高度集成的备份服务。在本章中，我们将关注重点重新置于用于长期数据保留的作为系统信息库的归档和管理解决方案。以前涵盖的管理原则和技术仍适用。但是，现在，磁盘高速缓存主要用于将文件获取到在获取之后不允许删除或修改的归档中。

确切的要求会有所不同。保留法律规定期间的业务或医疗记录的系统信息库可能需要定期放弃记录。但是，存储科学数据、历史记录、家谱记录、数字音乐、电影或电视节目的归档实际上可能需要永久存储内容。因此，Oracle HSM 通过多种方式支持数字保留：

- 使用消息摘要（校验和）可以检测文件的损坏、数据破坏和未经授权的修改，以便您可以更正硬件问题并将不可靠的文件替换为归档中其他位置存储的可靠副本。
- 文件固定性属性与消息摘要一起可确保只有超级用户可以更改已经修复的文件。每当 Oracle HSM 回写或归档固定文件时，它会重新验证与固定性属性一起存储的校验和以证明文件保持未变。
- Oracle HSM 一次写入多次读取 (Write Once Read Many, WORM) 文件系统可以使文件变为只读并在指定时间段内实施保留。这些文件系统可以配置为超级用户无法更改文件或文件属性（如上面讨论的固定性属性）。

本章首先简要回顾了基本的 Oracle HSM 数据保护措施，这些措施构成了任何长期存储解决方案（[为保留配置文件系统](#)）的基础。本章随后介绍了具体解决数据保留的任务：

- [使用消息摘要（校验和）](#)
- [将文件设置为不可变](#)
- [使用 WORM 文件系统](#)

为保留配置文件系统

每个保留解决方案都以一个可靠的高度冗余文件系统开始。因此请回顾配套《Oracle Hierarchical Storage Manager and StorageTek QFS Software 安装和配置指南》的实现章节（如果尚未这样做）。通过提供冗余服务器、网络连接和存储设备保护对归档的访问权限。通过为每个文件至少配置两个额外的副本（每个副本存储在独立介质上）来保护文件数据。在多数情况下，最好将一个副本归档到磁盘或固态存储设备

上，将两个副本归档到磁带介质上。如有可能，请确保通过实现 Oracle HSM 数据完整性验证功能来确保正确写入和读取了磁带块。通过定期生成转储文件和定期备份归档日志来保护文件系统元数据。

使用消息摘要（校验和）

消息摘要（校验和）允许保留人员检验归档文件是否存在可能会指示逐渐恶化、硬件错误、操作员错误的更改，或者是否存在对内容故意进行的未经授权的更改。消息摘要只是文件内容的、已由单向加密散列函数生成的数学汇总。加密散列函数对其输入数据中的更改极其敏感。即使在输入中执行微小的更改也会在输出中生成较大的更改。因此，消息摘要最适合用于检测文件损坏和未经授权的更改。重新计算文件的摘要并将结果值与存储的摘要值进行比较可显示文件是否已更改。

Oracle Hierarchical Storage Manager 文件系统可以使用下面的任何加密散列函数获取、创建、存储和验证消息摘要：

- SHA1，加密函数的安全散列算法系列的 160 位成员

安全散列算法在美国联邦信息处理标准 (*Federal Information Processing Standard, FIPS*) 出版物 180-4 国家标准和技术协会 (2012) 中定义。Oracle HSM 在默认情况下使用 SHA1。

- SHA256，安全散列算法系列的 256 位成员
- SHA384，安全散列算法系列的 384 位成员
- SHA512，安全散列算法系列的 512 位成员
- MD5，由 Internet 工程任务组 (Internet Engineering Task Force, IETF) 在互联网信息文档和标准 (request for comments, RFC) 1321 中定义的 128 位消息摘要函数
- 专有的 128 位 Oracle HSM 函数，现在主要用于与早期的 Storage Archive Manager 实现向后兼容。

当文件获取到系统信息库中时，用户可以提供现有的摘要值；或者用户可以让文件系统立即或在首次归档文件时计算摘要值。Oracle HSM 文件系统使用特殊的文件属性将摘要值和文件系统元数据一起存储。在设置属性之后，文件系统会重新计算摘要，并在重新归档相应的文件和（可选）将文件从归档介质回写到磁盘高速缓存时对照存储的值验证摘要。

但是，请注意，Oracle HSM 介质迁移功能将文件复制到新介质，而不重新计算校验和（有关介质迁移的信息，请参见第 8 章 [迁移到新存储介质](#)）。如果文件未正确复制，则说明在对文件重新回写和验证之前，将存在检测不到损坏的小风险。使用数据完整性验证 (Data Integrity Validation, DIV) 可将此风险降到最低（请参见《Oracle Hierarchical Storage Manager and StorageTek QFS Software 安装和配置指南》了解详细信息）。

在开始使用消息摘要之前，应当首先[确保文件系统主机性能将足够高](#)。有关提供、生成和验证摘要的说明，请参阅以下各部分：

- [为文件提供消息摘要并启用验证](#)

- 为文件生成消息摘要并启用验证
- 为目录中的每个文件生成消息摘要并启用验证
- 在回写期间验证文件的消息摘要
- 在对文件归档之前更改消息摘要和验证属性

确保文件系统主机性能将足够高

如果您打算充分利用消息摘要，请确保文件系统主机的计算资源足以实现高性能。大多数现代平台都合并了专用的加密硬件，加密硬件可以高效执行专业计算而不占用中央处理器周期。如果可用，请确保充分利用这些功能。

要检查潜在文件系统主机的功能，请执行如下操作：

1. 以 *root* 用户身份登录到文件系统主机：

```
root@solaris:~#
```

2. 确保主机操作系统运行 Solaris 11.1 或更高版本。使用命令 *uname -v*。

早期版本的操作系统不支持散列函数的硬件加速。在示例中，主机操作系统是 Solaris 11.2：

```
root@solaris:~# uname -v
11.2
root@solaris:~#
```

3. 显示指令集体系结构。在命令提示符处，输入命令 *isainfo -v*：

```
root@solaris:~# isainfo -v
```

4. 如果 Solaris 11 主机是 Oracle Sun SPARC T3 或更高版本的系统，则 *isainfo -v* 命令的输出应当列出支持 *sha512*、*sha256*、*sha1* 和 *md5* 加密算法的指令集。

在示例中，Sun SPARC T4-2 主机为 SHA1、SHA2 和 MD5 算法系列提供硬件加速：

```
[Sun_SPARC_T4-2]root@solaris:~# isainfo -v
64-bit sparcv9 applications
      crc32c cbcond pause mont mpmul sha512 sha256 sha1 md5 camellia kasumi
      des aes ima hpc vis3 fmaf asi_blk_init vis2 vis popc
root@solaris:~#
```

5. 如果 Solaris 主机是 x86/64 系统，它将支持 SHA-1 硬件加速，但前提是 *isainfo -v* 命令的输出中包括 *ssse3*（补充流 SIMD 扩展 3）指令集。

在示例中，Sun X3-2 主机支持 SHA-1 摘要的硬件加速：

```
[Sun_X3-2]root@solaris:~# isainfo -v
64-bit amd64 applications
    avx xsave pclmulqdq aes sse4.2 sse4.1 ssse3 popcnt tscp ahf cx16 sse3
    sse2 sse fxsr mmx cmov amd_sysc cx8 tsc fpu
root@solaris:~#
```

为文件提供消息摘要并启用验证

在对已经具有关联消息摘要的文件进行归档时，请执行如下操作。

1. 以 *root* 用户身份登录到文件系统主机：

```
root@solaris:~#
```

2. 在命令提示符处，输入命令 `ssum -a algorithm -h digest -G [-u]filename`，其中：
 - `-a algorithm` 标识在对照所提供的消息摘要验证文件时，文件系统应当使用的加密散列函数。
 - `-h digest` 标识文件系统应当用来验证文件的消息摘要。
 - `-G` 指定立即验证。文件系统将 `hash` 文件属性设置为所提供的消息摘要的值，独立计算该文件的消息摘要，并将结果与所存储的值进行比较。如果所提供的摘要和计算得出的摘要相匹配，则文件系统会为文件设置 `validated` 属性。它随后会设置 `generate` 属性，以便在对文件重新归档时重新检查其有效性。
 - `-u` 设置 `use` 文件属性（可选）。在对文件回写时，文件系统会重新计算摘要并对照 `hash` 属性中存储的值验证结果。
 - `filename` 是文件的路径和名称。

在示例中，提供了一个 SHA256 摘要，并让文件系统立即为 `data10` 文件重新计算摘要值并对照所提供的值验证计算结果。在使用命令 `sls -D -h data10` 检查文件属性时，会发现 `generate` 和 `validated` 文件属性已经设置，`algorithm` 属性已经设置为 `SHA-256`，摘要值已经计算而且已经存储在 `hash` 属性中：

```
root@solaris:~# ssum -h f03ce01b3828...f7459503007e -a sha256 -g data10
root@solaris:~# sls -D -h data10
data10:
mode: -rw-r--r--  links: 1  owner: root      group: root
length: 14975  admin id: 0  inode: 90217.1
project: user.root(1)
access: Jul 16 16:14  modification: Jul 16 16:14
changed: Jul 16 16:15  attributes: Jul 16 16:14
creation: Jul 16 16:14  residence: Jul 16 16:14
checksum: generate validated  algorithm: SHA-256
hash: f03ce01b3828...f7459503007e
root@solaris:~#
```

3. 必要时，按照通常的方式编辑该文件。

在示例中，自上次对名为 *data10m* 的文件进行归档以来，对该文件进行过修改。*sls -D -h* 命令显示已在两个副本上设置 *s*（过时）标志，因为这两个副本都未反映最新的更改。使用 Solaris *digest* 命令检查已修改文件的 SHA-256 摘要时，我们发现该文件的 *hash* 属性还存储过时摘要值：

```
root@solaris:~# sls -D -h data10m
data10m:
  mode: -rw-r--r--   links:   1  owner: root      group: root
  length:   14983  admin id:    0  inode:   90307.1
  project: user.root(1)
  copy 1: S-----  Jul 17 16:47      dd.1    dk diskarchive f221
  copy 2: S-----  Jul 20 11:31      a8d.1   li VOL002
  access:      Jul 20 11:32  modification: Jul 20 11:31
  changed:     Jul 17 16:37  attributes:   Jul 17 16:36
  creation:    Jul 17 16:36  residence:    Jul 17 16:36
  checksum: generate algorithm: SHA-256
  hash: f03ce01b3828...f7459503007e
root@solaris:~# digest -a sha256 data10m
56c55bb421cc...71ac2ac0b7b0
root@solaris:~#
```

4. 必要时，可以在重新归档之前更改已修改文件的摘要属性。

在示例中，将摘要算法从 SHA256 更改为 SHA1，便可立即显示其效果：

```
root@solaris:~# ssum -a sha1 -G data10m
root@solaris:~# sls -D -h data10m
data10m:
  mode: -rw-r--r--   links:   1  owner: root      group: root
  length:   14983  admin id:    0  inode:   90307.1
  project: user.root(1)
  release -a;
  copy 1: S-----  Jul 20 13:00      e0.1    dk diskarchive f224
  copy 2: S-----  Jul 20 13:05      a93.1   li VOL002
  access:      Jul 20 16:39  modification: Jul 20 16:39
  changed:     Jul 17 16:37  attributes:   Jul 17 16:36
  creation:    Jul 17 16:36  residence:    Jul 20 16:29
  checksum: generate validated algorithm: SHA-1
  hash: 92003525f0f8...53e29d0718c8
root@solaris:~#
```

5. 否则，等待文件系统归档已修改的文件并自动更新与摘要相关的属性。

在对已修改的文件归档时，文件系统会重新计算摘要值，将新值存储到 *hash* 属性，并在旧版本文件的所有归档副本上设置 *s*（过时）标志。在示例中，对 *data10m* 文件进行了编辑但未更改摘要属性。归档程序已经按照计划在磁盘上创建了一个新的 *copy 1*，并更新了 *hash* 属性。未经修改的文件的副本仍保留在磁带上并标记为 *s*（过时），直到归档程序可以创建 *copy 2* 为止：

```
root@solaris:~# sls -D -h data10m
data10m:
mode: -rw-r--r--   links:   1  owner: root      group: root
length:      14983  admin id:    0  inode:   90307.1
project: user.root(1)
copy 1: ----- Jul 17 16:47      dd.1    dk diskarchive f221
copy 2: s----- Jul 20 11:31      a8d.1   li VOL002
access:      Jul 20 11:32  modification: Jul 20 11:31
changed:     Jul 17 16:37  attributes:   Jul 17 16:36
creation:    Jul 17 16:36  residence:    Jul 17 16:36
checksum: generate algorithm: SHA-256
hash: 56c55bb421cc...71ac2ac0b7b0
```

为文件生成消息摘要并启用验证

要为文件生成摘要并启用验证，请执行如下操作：

1. 以 *root* 用户身份登录到文件系统主机：

```
root@solaris:~#
```

2. 在命令提示符处，输入命令 `ssum -a algorithm -g|G [-u] filename`，其中：
 - `-a algorithm` 指定在为文件生成消息摘要时，文件系统将使用的加密散列函数。
 - `-g` 为文件设置 *generate* 文件属性。首次对文件归档时，文件系统会计算消息摘要。在对文件重新归档时，文件系统会重新计算摘要并对照所存储的值验证结果。
 - `-G` 为文件设置 *generate* 和 *validate* 文件属性。文件系统会立即计算消息摘要并将结果存储在 *hash* 属性中。在对文件归档时，文件系统会重新计算摘要并对照所存储的值验证结果。
 - `-u` 设置 *use* 文件属性（可选）。在对文件回写时，文件系统会重新计算摘要并对照 *hash* 属性中存储的值验证结果。
 - *filename* 是文件的路径和名称。

在示例中，在归档之前，让文件系统使用 SHA256 算法为 *data11* 文件计算摘要。在使用命令 `sls -D -h data10` 检查文件属性时，我们发现，对于每个文件，*generate* 文件属性已经设置，并且 *algorithm* 属性已经设置为 *SHA-256*。文件尚未进行归档，因此摘要值尚未计算而且未存储在 *hash* 属性中：


```

root@solaris:~# ssum -a sha256 -g data11
root@solaris:~# sls -D -h data11
data11:
  mode: -rw-r--r--   links:  1  owner: root      group: root
  length:  14975  admin id:  0  inode:  90218.1
  project: user.root(1)
  access:      Jul 16 16:14  modification: Jul 16 16:14
  changed:     Jul 16 16:22  attributes:   Jul 16 16:14
  creation:    Jul 16 16:14  residence:    Jul 16 16:14
  checksum: generate  algorithm: SHA-256
  hash:
root@solaris:~#

```

3. 必要时，按照通常的方式编辑该文件。

在示例中，自上次对名为 *data11m* 的文件进行归档以来，对该文件进行过修改。*sls -D -h* 命令显示已在两个副本上设置 *s*（过时）标志，因为这两个副本都未反映最新的更改。使用 Solaris *digest* 命令检查已修改文件的 SHA-256 摘要时，我们发现该文件的 *hash* 属性还存储过时摘要值：

```

root@solaris:~# sls -D -h data11m
data11m:
  mode: -rw-r--r--   links:  1  owner: root      group: root
  length:  14983  admin id:  0  inode:  90307.1
  project: user.root(1)
  copy 1: s----- Jul 17 16:47      dd.1    dk diskarchive f221
  copy 2: s----- Jul 20 11:31      a8d.1   li VOL002
  access:      Jul 20 11:32  modification: Jul 20 11:31
  changed:     Jul 17 16:37  attributes:   Jul 17 16:36
  creation:    Jul 17 16:36  residence:    Jul 17 16:36
  checksum: generate  algorithm: SHA-256
  hash: f03ce01b3828...f7459503007e
root@solaris:~# digest -a sha256 data11m
56c55bb421cc...71ac2ac0b7b0
root@solaris:~#

```

4. 必要时，可以在重新归档之前更改已修改文件的摘要属性。

在示例中，将摘要算法从 SHA256 更改为 SHA1，便可立即显示其效果：

```

root@solaris:~# ssum -a sha1 -G data11m
root@solaris:~# sls -D -h data11m
data11m:
  mode: -rw-r--r--   links:  1  owner: root      group: root

```

```

length:      14983  admin id:      0  inode:      90307.1
project: user.root(1)
release -a;
copy 1: S----- Jul 20 13:00      e0.1    dk diskarchive f224
copy 2: S----- Jul 20 13:05      a93.1   li VOL002
access:      Jul 20 16:39  modification: Jul 20 16:39
changed:     Jul 17 16:37  attributes:   Jul 17 16:36
creation:    Jul 17 16:36  residence:    Jul 20 16:29
checksum: generate validated  algorithm: SHA-1
hash: 92003525f0f8...53e29d0718c8
root@solaris:~#

```

5. 否则，等待文件系统归档已修改的文件并自动更新与摘要相关的属性。

在对已修改的文件归档时，文件系统会重新计算摘要值，将新值存储到 *hash* 属性，并在旧版本文件的所有归档副本上设置 *s*（过时）标志。

在示例中，对 *data11m* 文件进行了编辑但未更改摘要属性。归档程序已经按照计划在磁盘上创建了一个新的 *copy 1*，并更新了 *hash* 属性。未经修改的文件的副本仍保留在磁带上并标记为 *s*（过时），直到归档程序可以创建 *copy 2* 为止：

```

root@solaris:~# sls -D -h data11m
mdata11:
mode: -rw-r--r--  links:  1  owner: root      group: root
length:      14983  admin id:      0  inode:      90307.1
project: user.root(1)
copy 1: ----- Jul 17 16:47      dd.1    dk diskarchive f221
copy 2: s----- Jul 20 11:31      a8d.1   li VOL002
access:      Jul 20 11:32  modification: Jul 20 11:31
changed:     Jul 17 16:37  attributes:   Jul 17 16:36
creation:    Jul 17 16:36  residence:    Jul 17 16:36
checksum: generate  algorithm: SHA-256
hash: 56c55bb421cc...71ac2ac0b7b0

```

为目录中的每个文件生成消息摘要并启用验证

要为目录中的每个文件以递归方式生成摘要并设置验证属性，请执行如下操作：

1. 以 *root* 用户身份登录到文件系统主机：

```
root@solaris:~#
```

2. 在命令提示符处，输入命令 `ssum -a algorithm -g|G [-u] -r directoryname`，其中：

- `-a algorithm` 指定在生成消息摘要时，文件系统将使用的加密散列函数。

- `-g` 为每个文件设置 `generate` 文件属性。首次对文件归档时，文件系统会为该文件计算消息摘要。在对文件重新归档时，文件系统会重新计算摘要并对照所存储的值验证结果。
- `-G` 为每个文件设置 `generate` 和 `validate` 文件属性。文件系统会立即计算消息摘要并将结果存储在 `hash` 属性中。在对文件归档时，文件系统会重新计算摘要并对照所存储的值验证结果。
- `-u` 设置 `use` 文件属性（可选）。在对文件回写时，文件系统会重新计算摘要并对照所存储的值验证结果。
- `-r` 以递归方式将该命令应用于指定目录中的所有文件。
- `directoryname` 是目录的路径和名称。

在第一个示例中，在归档之前，我们让文件系统使用 SHA256 算法为 `datasetA` 目录中的文件计算摘要。在使用命令 `sls -D -h datasetA` 检查文件属性时，我们发现，对于每个文件，`generate` 文件属性已经设置，并且 `algorithm` 属性已经设置为 `SHA-256`。这些文件尚未进行归档，因此摘要值尚未计算而且未存储在 `hash` 属性中：

```
root@solaris:~# ssum -a sha256 -g -r datasetA
root@solaris:~# sls -D -h datasetA
datasetA/pdata0:
mode: -rw-r--r--   links: 1 owner: root      group: root
length: 14983 admin id: 0 inode: 90232.1
project: user.root(1)
access: Jul 16 16:47 modification: Jul 16 16:47
changed: Jul 16 16:47 attributes: Jul 16 16:47
creation: Jul 16 16:47 residence: Jul 16 16:47
checksum: generate algorithm: SHA-256
hash:
...
datasetA/pdata20:
mode: -rw-r--r--   links: 1 owner: root      group: root
length: 14983 admin id: 0 inode: 90234.1
project: user.root(1)
access: Jul 16 16:47 modification: Jul 16 16:47
changed: Jul 16 16:47 attributes: Jul 16 16:47
creation: Jul 16 16:47 residence: Jul 16 16:47
checksum: generate algorithm: SHA-256
hash:
...
root@solaris:~#
```

在第二个示例中，在归档之前，我们让文件系统使用 SHA256 算法为 `datasetB` 目录中的文件计算摘要。在使用命令 `sls -D -h datasetB` 检查文件属性时，我

们发现 *generate* 和 *validated* 文件属性已经设置，*algorithm* 属性已经设置为 *SHA-256*，摘要值已经计算而且已经存储在 *hash* 属性中：

```
root@solaris:~# ssum -a sha256 -G -r datasetB
root@solaris:~# sls -D -h datasetB
datasetB/qdata0:
  mode: -rw-r--r--   links: 1 owner: root      group: root
  length: 14983 admin id: 0 inode: 90232.1
  project: user.root(1)
  access: Jul 16 16:47 modification: Jul 16 16:47
  changed: Jul 16 16:47 attributes: Jul 16 16:47
  creation: Jul 16 16:47 residence: Jul 16 16:47
  checksum: generate validated algorithm: SHA-256
  hash: 4d2800eb82b3...520341edde95
...
datasetB/qdata12:
  mode: -rw-r--r--   links: 1 owner: root      group: root
  length: 14983 admin id: 0 inode: 90234.1
  project: user.root(1)
  access: Jul 16 16:47 modification: Jul 16 16:47
  changed: Jul 16 16:47 attributes: Jul 16 16:47
  creation: Jul 16 16:47 residence: Jul 16 16:47
  checksum: generate validated algorithm: SHA-256
  hash: 5b057f1b7b48...88c590d47dec
...
root@solaris:~#
```

在回写期间验证文件的消息摘要

需要时，可以在将文件回写到磁盘高速缓存以供使用之前验证该文件。执行如下操作：

1. 以 *root* 用户身份登录到文件系统主机：

```
root@solaris:~#
```

2. 在命令提示符处，输入命令 `ssum -u [-a algorithm [-h digest] -g|G] filename`，其中：
 - *-u* 通过在回写之前设置 *use* 文件属性来指定验证。在为某个文件设置 *use* 属性时，文件系统不会将该文件从归档介质复制到磁盘高速缓存，直到它生成了消息摘要并对照该文件的 *hash* 属性中存储的值成功验证了结果。
 - *-a algorithm*、*-h digest* 和 *-g|G* 是可选参数，这些参数在文件上设置必需的 *algorithm*、*hash* 和 *generate* 属性（如果以前未设置）。
 - *filename* 是文件的路径和名称。

在示例中，已经为 `data102` 文件启用了验证。正如命令 `sls -D -h data102` 所示，`generate` 和 `validated` 文件属性已经设置，`algorithm` 属性已经设置为 `SHA-256`，摘要值已经计算而且已经存储在 `hash` 属性中：

```
root@solaris:~# ssum -a sha256 -F data102
root@solaris:~# sls -D -h data102
data102:
  mode: -rw-r--r--   links: 1  owner: root      group: root
  length: 14979  admin id: 0  inode: 90264.1
  project: user.root(1)
  access: Jul 16 17:34  modification: Jul 16 17:34
  changed: Jul 16 17:34  attributes: Jul 16 17:34
  creation: Jul 16 17:34  residence: Jul 16 17:34
  checksum: generate validated  algorithm: SHA-256
  hash: baae932ce1cf...93166a2e36b5
root@solaris:~#
```

因此，可以设置 `use` 属性以确保文件系统在回写该文件之前对其进行验证。命令 `sls -D -h data102` 显示现在设置了 `use` 属性：

```
root@solaris:~# ssum -u data102
root@solaris:~# sls -D -h data102
data102:
  mode: -rw-r--r--   links: 1  owner: root      group: root
  length: 14979  admin id: 0  inode: 90264.1
  project: user.root(1)
  access: Jul 16 17:34  modification: Jul 16 17:34
  changed: Jul 16 17:34  attributes: Jul 16 17:34
  creation: Jul 16 17:34  residence: Jul 16 17:34
  checksum: generate use validated  algorithm: SHA-256
  hash: baae932ce1cf...93166a2e36b5
root@solaris:~#
```

在对文件归档之前更改消息摘要和验证属性

如果某个文件尚未设置为不可变而且尚未归档，则可以使用下面的过程更改消息摘要和验证属性。

1. 以 `root` 用户身份登录到文件系统主机：

```
root@solaris:~#
```

2. 必要时，更改摘要算法。在命令提示符处，输入命令 `ssum -a newalgorithm filename`，其中：

- `-a newalgorithm` 指定用来替换以前指定的摘要算法的加密散列函数。
- `filename` 是文件的路径和名称。

在示例中，我们的保留策略需要使用高度抗冲突的 SHA256 函数。但是，正如命令 `sls -D -h` 所示，我们在设置 `data319` 文件的摘要属性时无意中指定了 SHA1 算法。由于尚未对该文件归档，因此我们可以成功地将算法更改为 SHA256：

```
root@solaris:~# sls -D -h data319
data319:
  mode: -rw-r--r--   links: 1  owner: root      group: root
  length: 14983  admin id: 0  inode: 90301.1
  project: user.root(1)
  access: Jul 17 15:27  modification: Jul 17 15:27
  changed: Jul 17 15:28  attributes: Jul 17 15:27
  creation: Jul 17 15:27  residence: Jul 17 15:27
  checksum: generate algorithm: SHA-1
  hash:
root@solaris:~# ssum -a sha256 data319
root@solaris:~# sls -D -h data319
data319:
  mode: -rw-r--r--   links: 1  owner: root      group: root
  length: 14983  admin id: 0  inode: 90301.1
  project: user.root(1)
  access: Jul 17 15:27  modification: Jul 17 15:27
  changed: Jul 17 15:28  attributes: Jul 17 15:27
  creation: Jul 17 15:27  residence: Jul 17 15:27
  checksum: generate algorithm: SHA-256
  hash:
root@solaris:~#
```

3. 必要时，清除摘要属性并恢复默认文件设置。在命令提示符处，输入命令 `ssum -d filename`，其中：
 - `-d` 将文件摘要属性重置为其默认值。
 - `filename` 是文件的路径和名称。

在示例中，我们并不打算为 `data44` 文件配置消息摘要和验证。但是，正如命令 `sls -D -h` 所示，我们无意中进行了配置。由于该文件尚未进行归档，因此我们可以成功地清除 `generate` 和 `use`，这两个属性在归档和回写期间控制消息验证。`validated`、`algorithm` 和 `hash` 属性中的数据仍保留但不会影响文件系统的行为：

```
root@solaris:~# sls -D -h data44
data44:
  mode: -rw-r--r--   links: 1  owner: root      group: root
```

```

length:      14983  admin id:      0  inode:      90292.1
project: user.root(1)
access:      Jul 17 14:58  modification: Jul 17 14:57
changed:     Jul 17 14:58  attributes:   Jul 17 14:57
creation:    Jul 17 14:57  residence:    Jul 17 14:57
checksum: generate use validated algorithm: SHA-256
hash: 3b4b15f8f69c...bae62c7e7568
root@solaris:~# ssum -d data44
root@solaris:~# sls -D -h data44
data44:
mode: -rw-r--r--  links: 1  owner: root      group: root
length: 14983  admin id: 0  inode: 90292.1
project: user.root(1)
access: Jul 17 14:58  modification: Jul 17 14:57
changed: Jul 17 14:58  attributes: Jul 17 14:57
creation: Jul 17 14:57  residence: Jul 17 14:57
checksum: validated algorithm: SHA-256
hash: 3b4b15f8f69c...bae62c7e7568
root@solaris:~#

```

4. 必要时，在对文件归档之前重置任何必需的消息摘要和验证属性。在命令提示符处，输入带有相应选项和文件名的命令 `ssum`。

在示例中，我们决定在 `qndat44` 文件上重新启用消息摘要并在归档之前验证摘要。但是，我们不需要在回写之前进行验证。因此，我们恢复 `generate` 属性但不恢复 `use` 属性：

```

root@solaris:~# ssum -g data44
root@solaris:~# sls -D -h data44
data44:
mode: -rw-r--r--  links: 1  owner: root      group: root
length: 14983  admin id: 0  inode: 90292.1
project: user.root(1)
access: Jul 17 14:58  modification: Jul 17 14:57
changed: Jul 17 14:58  attributes: Jul 17 14:57
creation: Jul 17 14:57  residence: Jul 17 14:57
checksum: generate validated algorithm: SHA-256
hash: 3b4b15f8f69c...bae62c7e7568
root@solaris:~#

```

将文件设置为不可变

保留要求通常需要使用可确保文件固定性的机制。归档必须既防止发生更改又证明此类更改尚未发生。为了提供固定性，Oracle HSM 归档文件系统将上面讨论的消息摘要和与摘要相关的文件属性与将文件呈现为不可变的额外属性结合使用。在将文件设

置为不可变之后，只有具有超级用户授权的用户才能更改文件状态。将不可变性与严格的一次写入多次读取 (Write Once Read Many, WORM) 文件系统结合使用时，即使超级用户也将无法进行更改（有关详细信息，请参见[“了解 WORM 文件系统”](#)）。

您可以按照下面的任一方式将文件设置为不可变：

- [提供消息摘要并将文件设置为不可变](#)
- [生成消息摘要并将文件设置为不可变](#)

提供消息摘要并将文件设置为不可变

如果需要确保文件在获取到归档中之后仍保持不变，请执行如下操作。

1. 以 *root* 用户身份登录到文件系统主机：

```
root@solaris:~#
```

2. 在命令提示符处，输入命令 `ssum -a algorithm [-h digest] -F filename`，其中：
 - `-a algorithm` 标识在对照所提供的消息摘要验证文件时，文件系统应当使用的加密散列函数。
 - `-h digest` 标识文件系统应当用来验证文件的消息摘要。
 - `-F` 指定立即验证和不可变性，并设置 `fixity`、`generate`、`validated` 和 `use` 文件属性。文件系统立即计算并验证消息摘要。在对文件回写或归档之后，文件系统会重新计算并重新验证消息摘要。
 - `filename` 是文件的路径和名称。

在示例中，提供了一个 SHA256 摘要并通知文件系统为 `data20` 文件重新计算摘要、验证摘要值并将该文件设置为不可变。在使用命令 `sls -D -h data10` 检查文件属性时，我们会发现，对于每个文件，`fixity`、`generate`、`use` 和 `validated` 文件属性已经设置，`algorithm` 属性已经设置为 `SHA-256`，摘要值已经计算而且已经存储在 `hash` 属性中：

```
root@solaris:~# ssum -h bfaefde932cf...d450892eda63 -a sha256 -F data20
root@solaris:~# sls -D -h data20
data20:
mode: -rw-r--r--   links: 1  owner: root      group: root
length: 14979  admin id: 0  inode: 90264.1
project: user.root(1)
access: Jul 16 17:34  modification: Jul 16 17:34
changed: Jul 16 17:34  attributes: Jul 16 17:34
creation: Jul 16 17:34  residence: Jul 16 17:34
checksum: fixity generate use validated  algorithm: SHA-256
hash: bfaefde932cf...d450892eda63
root@solaris:~#
```


生成消息摘要并将文件设置为不可变

如果对已经具有关联消息摘要的文件进行归档并且需要确保文件在获取到归档中之后仍保持不变，请执行如下操作。

1. 以 *root* 用户身份登录到文件系统主机：

```
root@solaris:~#
```

2. 在命令提示符处，输入命令 `ssum -a algorithm [-h digest] -F filename`，其中：
 - `-a algorithm` 标识用来生成 `-h digest` 参数中所指定的摘要的加密散列函数。
 - `-F` 设置 `fixity`、`generate`、`validated` 和 `use` 文件属性。文件系统立即计算并验证消息摘要。在对文件回写或归档之后，文件系统会重新计算并重新验证消息摘要。
 - `filename` 是文件的路径和名称。

在示例中，我们通知文件系统为 `data200` 文件计算 SHA256 摘要、验证摘要值并将该文件设置为不可变。在使用命令 `sls -D -h data10` 检查文件属性时，我们会发现，对于每个文件，`fixity`、`generate`、`validated` 和 `use` 文件属性已经设置，`algorithm` 属性已经设置为 `SHA-256`，摘要值已经计算而且已经存储在 `hash` 属性中：

```
root@solaris:~# ssum -a sha256 -F data200
root@solaris:~# sls -D -h data200
data200:
mode: -rw-r--r--   links:   1  owner: root       group: root
length:   14979  admin id:   0  inode:   90264.1
project: user.root(1)
access:      Jul 16 17:34  modification: Jul 16 17:34
changed:     Jul 16 17:34  attributes:   Jul 16 17:34
creation:    Jul 16 17:34  residence:    Jul 16 17:34
checksum: fixity generate use validated  algorithm: SHA-256
hash: efde93cc12cf...d496602e36dd
root@solaris:~#
```

检查文件摘要和固定性属性

要查看一个或多个文件的消息摘要和固定性属性，请使用 Oracle HSM 目录列出命令 `sls`。请执行如下操作。

列出消息摘要和验证属性

1. 以 *root* 用户身份登录到文件系统主机：

```
root@solaris:~#
```

2. 在命令提示符处，输入命令 `sls -D -h filename`，其中：

- `-D` 指定文件属性的详细显示内容。
- `-h` 在显示内容中包含散列（摘要）值。
- `filename` 按路径和名称标识一个或多个文件。

在示例中，可以查看所显示内容的 `checksum` 和 `hash` 字段中列出的 `data02` 文件的文件摘要属性：

```
root@solaris:~# sls -D -h data02
data02:
mode: -rw-r--r--   links: 1 owner: root      group: root
length: 14975  admin id: 0 inode: 90217.1
project: user.root(1)
access: Jul 16 16:14  modification: Jul 16 16:14
changed: Jul 16 16:15  attributes: Jul 16 16:14
creation: Jul 16 16:14  residence: Jul 16 16:14
checksum: generate use validated algorithm: SHA-256
hash: f03ce01b3828...f7459503007e
root@solaris:~#
```

- `hash` 属性存储 `f03ce01b3828...f7459503007e` 文件的消息摘要。
- `algorithm` 属性显示 `SHA-256` 加密散列函数生成了所存储的消息摘要。
- `generate` 属性显示文件系统会在对文件归档时，独立地重新计算消息摘要并对照所存储的值验证摘要值。
- `use` 属性显示文件系统会在对文件回写时，独立地重新计算消息摘要并对照所存储的值验证摘要值。
- `validated` 属性显示在上次检查时，独立计算的消息摘要与 `hash` 属性中存储的值相匹配。
- `fixity` 属性会在文件设置为不可变时显示。

使用 WORM 文件系统

当法律或归档注意事项要求使用一次写入多次读取 (write-once read-many, WORM) 文件系统时，可以在已经配置为支持 WORM 的任何 Oracle HSM 文件系统中创建 WORM 目录。本节的重点是了解 WORM 文件系统以及在处理 WORM 文件和目录时需要执行的特定任务，其中包括：

- [为目录启用 WORM](#)
- [为文件激活 WORM 保护](#)
- [查找并列 WORM 文件](#)

有关为文件系统启用 WORM 支持的信息，请参见《Oracle Hierarchical Storage Manager and StorageTek QFS Software 安装和配置指南》。

了解 WORM 文件系统

一次写入多次读取 (Write Once Read Many, WORM) 文件系统通过让用户将文件设置为在指定保持期内只读来保护数据。启用 WORM 的 Oracle HSM 文件系统支持默认的和可定制的文件保持期、数据和路径的不可更改性以及 WORM 设置的子目录继承性。

根据文件系统的配置情况，可以使用两种 Oracle HSM WORM 模式中的一种：

- 标准符合性模式（默认）
- 仿真模式

在标准 WORM 模式下挂载的文件系统中，用户通过执行命令 `chmod 4000 path_name` 针对目录启用 WORM 并为文件启动只读保持期，其中 `path_name` 是文件或目录的路径和名称。这会设置 UNIX `setuid`（执行时设置用户 ID）权限。针对同时具有 `execute` 权限的文件设置 `setuid` 权限会面临安全风险，因此，在标准 WORM 模式下，只有非可执行文件才能设置为只读。

在 WORM 仿真模式下挂载的文件系统中，用户通过执行命令 `chmod 555 path_name` 针对目录启用 WORM 并为文件启动只读保持期，其中 `path_name` 是可写文件或目录的路径和名称。由于仿真模式不需要 `setuid` 权限，因此可以将可执行文件设置为只读并为其分配保持期。

标准模式和仿真模式具有严格的 WORM 实施和不太严格的轻量级实施。在触发了对文件或目录的保留之后，严格实施和轻量级实施都不允许更改数据或路径。这两个模式均将默认保持期设置为 43,200 分钟（30 天）。但是，轻量级实施会放松对 `root` 用户的一些限制。

严格实施不允许任何人在保持期结束之前缩短指定的保持期或者删除文件或目录。严格实施也不允许使用 `sammkfs` 来删除存放当前保留的文件和目录的卷。因此，严格实施适用于满足最严苛的法律、法规遵从性和保留要求。

轻量级实施允许 `root` 用户使用 `sammkfs` 命令缩短保持期、删除文件和目录。这会为防止出现数据意外丢失而提供高级别的防护，并在管理文件系统和存储资源时提供更大的灵活性。但是，允许超级用户进行这种程度的控制的文件系统可能无法满足某些法律和法规遵从性要求。

可以创建到 WORM 文件的硬链接和软链接。只能对启用了 WORM 的目录中的文件创建硬链接。创建硬链接后，该链接与原始文件具有同样的 WORM 特征。也可以创建软链接，但软链接无法使用 WORM 功能。可以对 Oracle HSM 文件系统的任何目录中的 WORM 文件创建软链接。

有关创建和配置 WORM 文件系统的完整信息，请参见客户文档库中的《Oracle Hierarchical Storage Manager and StorageTek QFS Software 安装和配置指南》。

为目录启用 WORM

在为目录启用 WORM 时，会添加针对 WORM 文件的支持，但不会以其他方式更改目录的特征。用户可以继续创建和编辑启用了 WORM 的目录中的文件，并且可以删除启用了 WORM 但不包含 WORM 文件的目录。要为目录启用 WORM，请执行如下操作：

1. 登录文件系统服务器。

```
user@solaris:~#
```

2. 查看目录是否启用了 WORM。使用命令 `sls -Dd directory`，其中 `directory` 是目录的路径和名称。在该命令的输出中查找属性 `worm-capable`。

通常，目录会启用 WORM，因为当用户为目录启用 WORM 时，所有当前的和将来的子目录都将继承 WORM 功能（有关该命令的完整信息，请参见 `sls` 手册页）。在第一个示例中，我们发现，目标目录 `/hsm/hsmfs1/records` 已启用 WORM：

```
user@solaris:~# sls -Dd /hsm/hsmfs1/records/2013/
/hsm/hsmfs1/records/2013:
mode: drwxr-xr-x   links: 2   owner: root       group: root
length: 4096   admin id: 0   inode: 1048.1
project: user.root(1)
access: Mar 3 12:15   modification: Mar 3 12:15
changed: Mar 3 12:15   attributes: Mar 3 12:15
creation: Mar 3 12:15   residence: Mar 3 12:15
worm-capable      retention-period: 0y, 30d, 0h, 0m
```

但在第二个示例中，目标目录 `/hsm/hsmfs1/documents` 未启用 WORM：

```
user@solaris:~# sls -Dd /hsm/hsmfs1/documents
/hsm/hsmfs1/documents
mode: drwxr-xr-x   links: 2   owner: root       group: root
length: 4096   admin id: 0   inode: 1049.1
project: user.root(1)
access: Mar 3 12:28   modification: Mar 3 12:28
changed: Mar 3 12:28   attributes: Mar 3 12:28
creation: Mar 3 12:28   residence: Mar 3 12:28
```

3. 如果目录未启用 WORM，但挂载文件系统时使用了 `worm_capable` 或 `worm_lite` 挂载选项，则可使用 Solaris 命令 `chmod 4000 directory-name` 启用 WORM 支持，其中，`directory-name` 是将存放 WORM 文件的目录的路径和名称。

命令 `chmod 4000` 为目录设置 `setuid` (执行时设置用户 ID) 属性并启用标准 WORM 支持。在示例中, 为目录 `/hsm/hsmfs1/documents` 启用 WORM 并使用 `sls -Dd` 检查结果。该操作将成功, 而且将针对该目录启用 WORM:

```
user@solaris:~# chmod 4000 /hsm/hsmfs1/documents
user@solaris:~# sls -Dd /hsm/hsmfs1/documents
/hsm/hsmfs1/documents
mode: drwxr-xr-x  links: 2  owner: root      group: root
length: 4096  admin id: 0  inode: 1049.1
project: user.root(1)
access: Mar 3 12:28  modification: Mar 3 12:28
changed: Mar 3 12:28  attributes: Mar 3 12:28
creation: Mar 3 12:28  residence: Mar 3 12:28
worm-capable      retention-period: 0y, 30d, 0h, 0m
```

4. 如果目录未启用 WORM, 但挂载文件系统时使用了 `worm_emul` 或 `emul_lite` 挂载选项, 则可使用 Solaris 命令 `chmod 555 directory-name` 启用 WORM 支持, 其中, `directory-name` 是将存放 WORM 文件的目录的路径和名称。

命令 `chmod 555` 会删除目录的写入权限并启用 WORM 仿真支持。在示例中, 为目录 `/hsm/hsmfs1/documents` 启用 WORM 并使用命令 `sls -Dd` 检查结果。该操作将成功, 而且将针对该目录启用 WORM:

```
user@solaris:~# chmod 555 /hsm/hsmfs1/documents
user@solaris:~# sls -Dd /hsm/hsmfs1/documents
/hsm/hsmfs1/documents
mode: drwxr-xr-x  links: 2  owner: root      group: root
length: 4096  admin id: 0  inode: 1049.1
project: user.root(1)
access: Mar 3 12:28  modification: Mar 3 12:28
changed: Mar 3 12:28  attributes: Mar 3 12:28
creation: Mar 3 12:28  residence: Mar 3 12:28
worm-capable      retention-period: 0y, 30d, 0h, 0m
```

为文件激活 WORM 保护

如果针对启用了 WORM 的目录中的文件激活 WORM 保护, 则在保持期内, 文件系统不再允许对文件数据或数据路径进行修改。因此, 您必须谨慎使用。要激活 WORM 保护, 请执行如下操作:

1. 登录文件系统服务器。

```
user@solaris:~#
```

- 如果文件需要保留的期限与文件系统的默认设置不同，可通过更改文件的访问时间指定所需的保留时间。使用 Solaris 命令 `touch -a -texpiration-date path-name`，其中：
 - `expiration-date` 是一个数字字符串，其中包括四位年、两位月、两位月中日期、两位小时、两位分钟和（可选）两位秒钟。
 - `path-name` 是文件的路径和名称。

请注意，Oracle Solaris UNIX 实用程序（如 `touch`）不能将保持期延长到 2038 年 1 月 18 日下午 10:14 之后。这些实用程序使用带符号的 32 位数字表示自 1970 年 1 月 1 日以来的时间（以秒为单位）。因此，如果需要将文件保留到此截止日期之后，请使用默认保持期。

在示例中，将文件的保持期设置为在 2019 年 10 月 4 日上午 11:59 之后到期：

```
user@solaris:~# touch -a -t201910141159 /hsm/hsmfs1/plans/master.odt
```

- 如果挂载文件系统时使用了 `worm_capable` 或 `worm_lite` 挂载选项，则可使用 Solaris 命令 `chmod 4000 path-name` 激活 WORM 保护，其中 `path-name` 是文件的路径和名称。

`chmod 4000` 命令为指定的文件设置 `setuid`（执行时设置用户 ID）属性。为可执行文件设置此属性是不安全的。因此，如果挂载文件系统时使用了 `worm_capable` 或 `worm_lite` 挂载选项，将无法为具有 UNIX `execute` 权限的文件设置 WORM 保护。

在示例中，为文件 `master.odt` 激活 WORM 保护。我们使用 `sls -D` 检查结果，并注意到 `retention` 属性现在设置为 `active`，`retention-period` 设置为四年：

```
user@solaris:~# chmod 4000 /hsm/hsmfs1/plans/master.odt
user@solaris:~# sls -Dd /hsm/hsmfs1/plans/master.odt
/hsm/hsmfs1/plans/master.odt:
mode: -r-xr-xr-x  links: 1  owner: root      group: root
length:      104  admin id: 0  inode: 1051.1
project: user.root(1)
access:      Mar  4 2018  modification: Mar  3 13:14
changed:     Mar  3 13:16  retention-end: Apr  2 14:16 2014
creation:    Mar  3 13:16  residence:     Mar  3 13:16
retention:   active      retention-period: 4y, 0d, 0h, 0m
```

- 如果挂载文件系统时使用了 `worm_emul` 或 `worm_lite` 挂载选项，则可使用 Solaris 命令 `chmod 555 path-name` 激活 WORM 保护，其中 `path-name` 是文件的路径和名称。

命令 `chmod 555` 会删除对目录的写入权限。因此，您可以在需要时通过 WORM 保护可执行文件。在该示例中，为文件 `master-plan.odt` 激活 WORM

保留。我们使用 `sls -D` 检查结果，并注意到 `retention` 属性现在设置为 `active`，`retention-period` 设置为四年：

```
user@solaris:~# chmod 555 /hsm/hsmfs1/plans/master.odt
user@solaris:~# sls -Dd /hsm/hsmfs1/plans/master.odt
/hsm/hsmfs1/plans/master.odt:
  mode: -r-xr-xr-x   links:   1  owner: root       group: root
  length:          104  admin id:   0  inode:   1051.1
  project: user.root(1)
  access:         Mar  4 2018  modification: Mar  3 13:14
  changed:        Mar  3 13:16  retention-end: Apr  2 14:16 2014
  creation:       Mar  3 13:16  residence:    Mar  3 13:16
  retention:      active      retention-period: 4y, 0d, 0h, 0m
```

查找并列出 WORM 文件

要查找并列出符合指定搜索条件的 WORM 文件，请使用 `sfind` 命令。执行如下操作：

1. 登录文件系统服务器。

```
user@solaris:~#
```

2. 要列出受 WORM 保护并在保持期内的文件，请使用命令 `sfind starting-directory -ractive`，其中 `starting-directory` 是需要列表进程开始列举的目录的路径和名称。

```
user@solaris:~# sfind /hsm/hsmfs1/ -ractive
/hsm/hsmfs1/documents/2013/master-plan.odt
/hsm/hsmfs1/documents/2013/schedule.ods
/samma1/records/2013/progress/report01.odt
/samma1/records/2013/progress/report02.odt
/samma1/records/2013/progress/report03.odt ...
user@solaris:~#
```

3. 要列出保持期已到期的受 WORM 保护的文件，请使用命令 `sfind starting-directory -rover`，其中 `starting-directory` 是需要列表进程开始列举的目录的路径和名称。

```
user@solaris:~# sfind /hsm/hsmfs1/ -rover
/samma1/documents/2007/master-plan.odt
/samma1/documents/2007/schedule.ods
user@solaris:~#
```

4. 要列出其保持期在指定日期和时间之后将过期的、受 WORM 保护的文件，使用命令 `sfind starting-directory -rafter expiration-date`，其中：

- *starting-directory* 是您希望启动列表进程的目录的路径和名称
- *expiration-date* 是一个数字字符串，其中包括四位年、两位月、两位月中日期、两位小时、两位分钟和（可选）两位秒钟。

在示例中，列出保持期将在 2015 年 1 月 1 日午夜后一分钟到期的所有文件：

```
user@solaris:~# sfind /hsm/hsmfs1/ -rafter 201501010001
/hsm/hsmfs1/documents/2013/master-plan.odt
user@solaris:~#
```

5. 要列出必须在文件系统中至少保留指定时间的受 WORM 保护的文件，请使用命令 *sfind starting-directory -rremain time-remaining*，其中：

- *starting-directory* 是在目录树中开始搜索的位置。
- *time-remaining* 是由非负整数与以下时间单位组成的字符串：*y*（表示年数）、*d*（表示天数）、*h*（表示小时数）、*m*（表示分钟数）。

在示例中，在目录 */hsm/hsmfs1/* 下查找至少还要保留三年的所有文件：

```
user@solaris:~# sfind /hsm/hsmfs1/ -rremain 3y
/hsm/hsmfs1/documents/2013/master-plan.odt
user@solaris:~#
```

6. 要列出必须在文件系统中保留超过指定时间的受 WORM 保护的文件，请使用命令 *sfind starting-directory -rlonger time*，其中：

- *starting-directory* 是在目录树中开始搜索的位置。
- *time-remaining* 是由非负整数与以下时间单位组成的字符串：*y*（表示年数）、*d*（表示天数）、*h*（表示小时数）、*m*（表示分钟数）。

在示例中，在目录 */hsm/hsmfs1/* 下查找将保留三年零九十天以上的所有文件：

```
user@solaris:~# sfind /hsm/hsmfs1/ -rremain 3y90d
/hsm/hsmfs1/documents/2013/master-plan.odt
user@solaris:~#
```

7. 要列出必须在文件系统中永久保留的受 WORM 保护的文件，可使用命令 *sfind starting-directory -rpermanent*。

在示例中，我们发现，在目录 */hsm/hsmfs1/* 下没有要永久保留的文件：

```
user@solaris:~# sfind /hsm/hsmfs1/ -rpermanent
user@solaris:~#
```


第 7 章 备份配置和文件系统

安装和配置 Oracle Hierarchical Storage Manager and StorageTek QFS Software 时，创建用于存储恢复点文件和归档程序日志副本的安全位置。还配置了创建恢复点、备份日志以及保护系统配置的自动化过程。这些步骤为您的文件系统提供核心保护。但您可能还需要不时地采取非预定保护措施：

- 在预计可能发生的中断性事件（例如对物理基础结构或数据中心设施的较大更改）发生之前备份 Oracle HSM 配置和文件系统。
- 升级或重新配置软件、操作系统或主机平台之后备份 Oracle HSM 配置和文件系统，以保护当前配置。
- 在进行 Oracle HSM 支持服务之前，收集所需的配置和状态信息。

本章概括介绍根据需要收集、创建及存储配置和文件系统恢复文件的过程。其中包含三个主要部分：

- [备份文件系统](#)
- [备份 Oracle HSM 配置](#)
- [使用 samexplorer 收集配置和诊断信息](#)。

请注意，本章中所有任务都是通过命令行界面执行。如果您希望使用 Oracle HSM Manager 图形用户界面，请参阅联机帮助了解详细过程。

备份文件系统

本节首先简要介绍 Oracle HSM 文件系统保护（[“了解恢复点和归档日志”](#)）。然后介绍执行以下任务的过程：

- [根据需要创建恢复点](#)
- [备份归档程序日志](#)
- [使用 samexplorer 收集配置和诊断信息](#)
- [手动备份 Oracle HSM 配置](#)。

了解恢复点和归档日志

为了保护文件系统，需要做两件事：

- 必须保护存储数据的文件。

- 必须保护文件系统本身，以便能够使用、组织、查找、访问及管理数据。

在 Oracle HSM 归档文件系统中，由归档程序自动保护文件数据：修改的文件会自动复制到归档存储介质（例如磁带）上。但如果您只备份了文件，之后磁盘设备或 RAID 组遇到不可恢复的故障，您虽有数据但很难使用它。您必须创建替代文件系统、标识每个文件、确定其在新文件系统中的适当位置、获取新文件系统，并重新创建其与用户、应用程序及其他文件之间丢失的关系。这种恢复即使是在最好的情况下也是一个艰巨且冗长的过程。

因此，为了快速、高效恢复，必须主动保护能够使文件和归档副本可用的文件系统元数据。必须备份归档在可移除介质上的副本的目录路径、inode、访问控制、符号链接及指针。

可通过安排恢复点和保存归档日志保护 Oracle HSM 文件系统元数据。恢复点是一个压缩文件，存储 Oracle HSM 文件系统的元数据时间点备份副本。发生数据丢失时（从用户文件的意外删除到整个文件系统的灾难性丢失），都可以通过找到文件或文件系统保持完好的最后恢复点，将文件或文件系统几乎立刻恢复到最后已知正常状态，然后，可恢复在该时间点记录的元数据，并将元数据中指示的文件从归档介质回写到磁盘高速缓存中，或者（更适合）让文件系统按需回写文件，以供用户和应用程序访问这些文件。

与任何时间点备份副本一样，恢复点很少是出现故障时文件系统状态的完整记录。在完成一个恢复点之后和创建下一个恢复点之前，至少会创建和更改一些文件，这是不可避免的。您可以且应该通过恰当安排来最大限度消除这种问题，即频繁创建恢复点并在文件系统不使用时创建恢复点。但实际上这种安排只能是一种折中策略，因为文件系统总会处于使用状态。

因此，还必须保存归档程序日志文件的时间点副本。归档每个数据文件时，日志文件将记录归档介质的卷序列号、归档集和副本份数、归档 (*tar*) 文件在介质上的位置以及 *tar* 文件内数据文件的路径和名称。有了这些信息，可以使用 Solaris 或 Oracle HSM *tar* 实用程序恢复在恢复点中缺失的文件。但这些信息具有易失性。与多数系统日志一样，归档程序日志会快速增长，因此必定会经常被覆盖。如果不定期生成副本来补充恢复点信息，在需要时就没有日志信息可用。

本节的其余部分提供有关按需创建恢复点和日志副本的说明。包含以下小节：

- [根据需要创建恢复点](#)
- [备份归档程序日志](#)。

根据需要创建恢复点

有时可能需要在常规计划之外的时间点从归档文件系统获取元数据。例如，每当您参与可能中断系统或设施的维护时，您可以在恢复点之前或之后创建以确保文件系统受到保护。

要根据需要启动非预定的按需恢复点创建，请执行如下操作：

1. 以 *root* 用户身份登录到 Oracle HSM 服务器主机。

```
root@solaris:~#
```

2. 选择将存储恢复点的独立位置。

在示例中，在初始配置文件系统时为恢复点创建的目录下创建一个子目录 *unscheduled/*。*/zfs1* 文件系统位于远程位置，且与 Oracle HSM 文件系统没有共用组件。

```
root@solaris:~# mkdir /zfs1/samqfs_recovery/unscheduled
root@solaris:~#
```

3. 转到文件系统的根目录。

在示例中，转到挂载点目录 */samqfs*：

```
root@solaris:~# cd /samqfs
root@solaris:~#
```

4. 如果要备份其数据复制到可移除介质的归档文件系统，仅备份元数据。使用命令 *samfsdump -f recovery-point*，其中 *recovery-point* 是完成的恢复点文件的路径和文件名。

有关其他详细信息，请参见 *samfsdump* 手册页。在示例中，在预定的维护相关断电之前创建 *samqfs* 文件系统的非预定恢复点。我们在目录 */zfs1/samqfs_recovery/unscheduled/* 中创建恢复点文件 *20150315pre-outage*（请注意，下面的第二条命令是作为单行输入的一使用反斜杠字符对换行符进行转义）：

```
root@solaris:~# cd /samqfs
root@solaris:~# samfsdump -f /
/zfs1/samqfs_recovery/unscheduled/20150315pre-outage
root@solaris:~#
```

5. 如果要备份数据未复制到可移除介质的独立文件系统，需要备份元数据和数据。使用命令 *samfsdump -U -f recovery-point*，其中 *recovery-point* 是完成的恢复点文件的路径和文件名。

请注意，包含数据和元数据的恢复点文件会非常大。有关其他详细信息，请参见 *samfsdump* 手册页。在示例中，创建 *samqfs* 文件系统的非预定恢复点。我们在远程目录 */zfs1/samqfs_recovery/unscheduled/* 中创建恢复点文件 *20150315pre-outage*（请注意，下面的第二条命令是作为单行输入的一使用反斜杠字符对换行符进行转义）：

```
root@solaris:~# cd /samqfs
root@solaris:~# samfsdump -f -U /
```

```
/zfs1/samqfs_recovery/unscheduled/20150315pre-outage
root@solaris:~#
```

6. 如果要备份归档文件系统，则备份归档程序日志。
7. 否则，根据具体情况，可能还需要运行 *samexplorer* 并手动备份 Oracle HSM 配置。

备份归档程序日志

虽然恢复点文件几乎包含恢复文件系统时所需的所有信息，但并不包含创建恢复点之后创建或修改的文件的元数据。由于归档程序日志列出了已归档的所有文件及其在磁带上的位置，可以使用归档程序日志来恢复创建恢复点之后归档的任何文件。因此，如果可能，每次创建非预定的恢复点后，请为归档程序日志文件创建非预定的副本。请执行如下操作。

1. 以 *root* 用户身份登录到 Oracle HSM 服务器主机。

```
root@solaris:~#
```

2. 选择将要存储备份归档程序日志的独立位置。

在示例中，决定将日志存储在上面创建的对应非预定恢复点的同一目录中。/*zfs1* 文件系统位于远程位置，且与 Oracle HSM 文件系统没有共用组件。

```
root@solaris:~# ls /zfs1/samqfs_recovery/unscheduled
20150315pre-outage
root@solaris:~#
```

3. 将当前归档程序日志复制到所选位置，并为其指定唯一的名称。使用命令 *cp /var/adm/samqfs.archive.log path/"date +%y%m%d"*；其中 *path* 是所选位置的路径。

请注意，下面的命令是作为单行输入的一使用反斜杠字符对换行符进行转义：

```
root@solaris:~# cp /var/adm/samqfs.archive.log /
/zfs1/samqfs_recovery/unscheduled/20150315pre-outage/"date +%y%m%d".archive.log
root@solaris:~#
```

4. 根据具体情况，可能还需要运行 *samexplorer* 并手动备份 Oracle HSM 配置。

备份 Oracle HSM 配置

只要更改了 Oracle HSM 配置，就应备份所有修改的配置文件和相关信息以保护您的投资。执行以下任务：

- [手动备份 Oracle HSM 配置](#)

- [运行 samexplorer](#).

手动备份 Oracle HSM 配置

为了实现完全冗余，只要对软件、操作系统或主机做了重大更改，就应创建配置文件的本地副本。执行如下操作：

1. 以 `root` 用户身份登录到文件系统主机。

```
root@solaris:~#
```

2. 在保存备份配置信息的子目录中，为 Oracle HSM 配置的手动备份副本创建子目录。使用命令 `mkdir mount-point/path`，其中 `mount-point` 是所选独立文件系统的挂载点目录，`path` 是所选目录的路径和名称。

在示例中，要为归档文件系统 `/samqfs` 配置恢复点。因此创建了目录 `/zfs1/sam_config/samconfig`：

```
root@solaris:~# mkdir /zfs1/sam_config/samconfig
```

3. 在保存 Oracle HSM 配置的手动备份副本的子目录中，为当前 Oracle HSM 配置创建子目录，使用命令 `mkdir mount-point/path/subdirectory`，其中 `mount-point` 是所选独立文件系统的挂载点，`path/subdirectory` 是所选子目录的路径和名称。

在示例中，在最初配置时为此目的创建的目录 `/zfs1/sam_config/samconfig` 中创建子目录。将日期用于命名子目录：

```
root@solaris:~# mkdir /zfs1/sam_config/samconfig/20150315
```

4. 将配置文件复制到其他文件系统。

```
/etc/opt/SUNWsamfs/
  mcf
  archiver.cmd
  defaults.conf
  diskvols.conf
  hosts.family-set-name
  hosts.family-set-name.local
  preview.cmd
  recycler.cmd
  releaser.cmd
  rft.cmd
  samfs.cmd
  stager.cmd
  inquiry.conf
```

```

samremote          # SAM-Remote server configuration file
family-set-name    # SAM-Remote client configuration file
network-attached-library # Parameters file
scripts/*          # Back up all locally modified files
/var/opt/SUNWsamfs/

```

5. 备份所有磁带库目录数据，包括历史记录所维护的数据。对于每个目录，使用命令 `/opt/SUNWsamfs/sbin/dump_cat -V catalog-file`，其中 *catalog-file* 是目录文件的路径和名称。将输出重定向到新位置中的 *dump-file*。

在本示例中，我们将 *library1* 的目录数据转储到已挂载 NFS 的独立文件系统 *zfs1* 上的目录中的文件 *library1cat.dump*（请注意，下面的命令是作为单行输入的一使用反斜杠字符对换行符进行转义）：

```

root@solaris:~# dump_cat -V /var/opt/SUNWsamfs/catalog/library1cat > /zfs1/sam
_config/20150315/catalogs/library1cat.dump

```

6. 请复制 Oracle HSM 安装和配置过程中所修改的系统配置文件。其中可能包括：

```

/etc/
  syslog.conf
  system
  vfstab
/kernel/drv/
  sgen.conf
  samst.conf
  samrd.conf
  sd.conf
  ssd.conf
  st.conf
/usr/kernel/drv/dst.conf

```

7. 将作为 Oracle HSM 配置的一部分而创建的任何定制 shell 脚本和 *crontab* 条目复制到选定的子目录中。

例如，如果创建了 *crontab* 条目来管理恢复点创建，现在就保存其副本。

8. 将作为 Oracle HSM 配置的一部分而创建的任何定制 shell 脚本和 *crontab* 条目复制到选定的子目录中。

例如，如果创建了 *crontab* 条目来管理恢复点创建，现在就保存其副本。

9. 记录当前安装的软件（包括 Oracle HSM、Solaris 和 Solaris Cluster（如果适用））的修订级别，并将信息副本保存在所选子目录的 *readme* 文件中。
10. 在所选子目录中，保存下载的 Oracle HSM Oracle HSM、Solaris 和 Solaris Cluster 软件包的副本，以便在需要时可以快速恢复软件。

11. 接着，运行 *samexplorer*。

使用 samexplorer 收集配置和诊断信息

samexplorer 是一个诊断工具，用于捕获和报告 Oracle HSM 软件和文件系统配置和状态的全面信息。只要更改了 Oracle HSM 配置，就应该运行 *samexplorer* 并存储与配置文件的备份副本相关的结果报告。此外，在排除问题时以及 Oracle HSM 支持服务人员要求这么做时，也应该运行 *samexplorer*。执行如下操作：

运行 samexplorer

1. 以 *root* 用户身份登录到文件系统主机。
2. 在保存备份配置信息的目录中，为 *samexplorer* 报告创建子目录。使用命令 `mkdir mount-point/path`，其中 *mount-point* 是所选独立文件系统的挂载点目录，*path* 是所选目录的路径和名称。

在示例中，在最初配置时为此目的创建的目录 `/zfs1/sam_config/` 中创建新子目录。将新子目录命名为 *explorer/*：

```
root@solaris:~# mkdir /zfs1/sam_config/explorer
```

3. 在所选目录中创建 *samexplorer* 报告。使用命令 `samexplorer path/hostname.YYYYMMDD.hhmmz.tar.gz`，其中 *path* 是所选目录的路径，*hostname* 是 Oracle HSM 文件系统主机的名称，*YYYYMMDD.hhmmz* 是日期和时间戳。

默认文件名是 `/tmp/SAMreport.hostname.YYYYMMDD.hhmmz.tar.gz`。在本示例中，我们针对主机 *samhost1* 创建一个报告，日期为 2015 年 3 月 15 日，时间为山地时间下午 4:59（请注意，下面的命令是作为单行输入的一使用反斜杠字符对换行符进行转义）：

```
root@solaris:~# samexplorer /
/zfs1/sam_config/explorer/samhost1.20150315.1659MST.tar.gz

Report name:      /zfs1/sam_config/explorer/samhost1.20150315.1659MST.tar.gz
Lines per file:  1000
Output format:   tar.gz (default) Use -u for unarchived/uncompressed.

Please wait.....
Please wait.....
Please wait.....

The following files should now be ftp'ed to your support provider
as ftp type binary.
```

```
/zfs1/sam_config/explorer/samhost1.20150315.1659MST.tar.gz
```

4. 只要对文件系统配置做了重大更改，就应重复该过程。
5. 在此处停止。Oracle HSM 配置现已备份。

第 8 章 迁移到新存储介质

磁带是高度可靠且稳定的介质，适用于长期存储和保留数据。但它的寿命有限。正常的机械过程（挂载、拉伸和读取/写入）中所产生的磨损和损伤会随着时间累积。当具有更高性能的新设备可用时，较旧的驱动器便会更难支持，并且兼容的介质变得更昂贵而且更难找到。因此，在某个点，您需要将归档传输到新介质。

在 Oracle HSM 分层文件系统中，将旧介质替换为新介质是复杂的过程。磁带介质是文件系统不可或缺的部分。文件系统元数据记录每个文件的数据的多个副本的位置，一些副本位于磁盘上，另一些位于磁带副本上。因此文件系统 inode 必须进行更新，以便在复制磁带后反映已迁移文件副本的新位置。同时，必须对介质和驱动器进行管理，以便在创建副本时不会干扰正常的文件系统操作（如归档和回写）。

Oracle Hierarchical Storage Manager 提供两种方法来管理复杂的介质迁移。每种方法都有其优点，具体取决于您的要求。

Oracle HSM 6.1 中引入的介质迁移功能将整个卷从一个磁带库驱动器上挂载的介质迁移到另一个磁带库驱动器上挂载的介质，并在该过程中更新文件系统元数据。这样便能够最大程度地降低系统开销和管理员工作负荷。当不需要对驱动器进行归档或回写时，会在后台复制卷。您可以指定所用驱动器的数量以及可能发生迁移的时间。或者您可以让 Oracle HSM 在驱动器空闲时迁移卷。如果归档或回写作业需要某个驱动器或卷，则介质迁移过程会服从优先级更高的操作。如果已正确配置，而且存在 StorageTek T1000D（或更高版本）目标驱动器，则可以使用 T10000 扩展复制功能和 Oracle HSM *xcopy* 选项进行迁移。在发出请求之后，驱动器会自行处理复制，而不使用服务器资源。否则，您仍可以使用 Oracle HSM 服务器复制选项来最小化服务器负荷。文件系统服务器随后将通过一个可配置的 I/O 缓冲区在驱动器之间复制卷。

常规的归档过程还可以迁移数据，一次复制一个归档文件。可以将系统配置为将文件从旧介质回写到磁盘高速缓存，然后再将它们重新归档到新介质。该逐个文件方法可使您更大程度地对文件的分组和分发方式进行控制。但它需要更多的管理。您可以亲自分配磁盘高速缓存和驱动器资源，因此，如果您需要最大程度地降低对常规文件系统操作的干扰，则必须仔细规划。

本文档的其余部分会引导您完成该过程：

- [准备迁移](#)
- [选择迁移方法](#)
- 通过[迁移完整卷](#)或通过[回写文件并将其重新归档到替换介质中](#)迁移数据。

准备迁移

在进一步继续操作之前，执行以下步骤：

- [确保文件系统已备份](#)
- [提供所需的介质](#)

确保文件系统已备份

在启动介质迁移之前，确保通常保护 Oracle HSM 归档数据的恢复机制在转换期间和转换之后仍保持有效。在迁移操作过程中，可能会出现或多或少的加密硬件故障和用户错误。因此，您需要始终确保可以从现存的 *samfsdump* 恢复点文件恢复文件和/或完成文件系统。

在迁移期间和在迁移之后的某个时间，恢复将取决于引用源磁带卷的恢复点文件，而不是新的目标卷。如果严重硬件故障禁用文件系统，而且这些旧磁带不可用，则将无法进行恢复。

因此，至少对保留旧磁带方面进行计划，直到您已创建足够的新恢复点，以便从新介质恢复当前的文件系统。如果您需要确保将文件恢复到特定的时间点，则可能需要将旧介质保留更长时间（如果无法无限期保留）。最好将旧卷保留在便于访问它们的磁带库中。

提供所需的介质

确保目标磁带库包含足够的、可存放已迁移文件的介质。确保所有的卷带有正确的标签，如“[为新磁带设置标签或者为现有磁带重新设置标签](#)”中所述。如果卷不带标签，迁移将失败。

选择迁移方法

所选迁移方法取决于归档状态以及用户和应用程序要求。使用下面的过程做出决定。

选择最符合您需求的迁移方法

1. 确定归档在迁移过程中是否仍保持运行。

停止归档并使所有的资源专门用于迁移可能会简化任务并加速完成。但是，如果归档正在使用中，则它的实用性不大。

2. 如果您需要选择性地迁移多组归档文件（而不是迁移整个卷），或者需要保持多组归档文件之间的指定关系，请使用回写和重新归档方法。转至“[回写文件并将其重新归档到替换介质中](#)”。
3. 如果您只需将旧卷复制到新介质和/或需要最小化迁移对文件系统操作的影响，请使用卷迁移方法。

4. 如果您没有可用作目标驱动器的光纤通道 Oracle StorageTek T10000D（或更高版本）驱动器，或者源或目标磁带不共享公用块大小，请使用服务器复制方法。

在该模式下，Oracle HSM 软件仅将有效的归档文件从源驱动器复制到文件系统服务器上可配置的 I/O 缓冲区。如果源块大小不同于目标块大小，则软件会自动进行调整，但前提是目标块大于源块。软件随后会将磁带块从缓冲区发送到目标驱动器。

5. 如果您有光纤驱动器 StorageTek T10000D（或更高版本）目标驱动器，这些驱动器均运行最新的固件，源磁带和目标磁带共享相同的块大小，而且源驱动器和目标驱动器通过同一个存储区域网络 (storage area network, SAN) 开关建立连接，则使用 Oracle HSM *xcopy* 选项。

在指定 *xcopy* 时，文件系统服务器向驱动器发送 SCSI 复制请求，T10000D 驱动器将从第一个有效的归档文件开始，将源逐块复制到目标磁带。如果 *xcopy* 操作由于某种原因失败，则迁移软件会自动切换到服务器复制方法。*xcopy* 方法会最大限度地提高性能并最大限度地降低服务器开销。

有关驱动器和固件要求的更多信息，请参见发行说明 *README.txt*，位于下载 ZIP 文件中，或文件系统服务器上的 */opt/SUNWsamfs/doc/README.txt* 中

6. 如果源卷包含相对较少的过期文件，请在 *eod*（数据末尾）模式下使用 *xcopy* 选项。

在该模式下，T10000 磁带机会复制在第一个有效文件和磁带上的数据末尾 (end-of-data, EOD) 标志之间找到的所有归档文件。如果其中的某些文件已过时，则它们会复制到包含有效文件的目标卷。

7. 如果源卷包含许多过期文件，请在 *repack* 模式下使用 *xcopy* 选项。

在该模式下，T10000 磁带机仅将未过期的归档文件复制到目标卷。

8. 转至[回写文件并将其重新归档到替换介质中](#)。

迁移完整卷

选择服务器复制或直接复制方法并通过创建 *migrationd.cmd* 文件来配置迁移。执行以下任务：

- [创建 migrationd.cmd 文件](#)
- [检查处于活动状态的迁移作业](#)
- [迁移卷](#)

创建 migrationd.cmd 文件

1. 以 *root* 用户身份登录到 Oracle HSM 元数据服务器。

```
root@solaris:~#
```

- 在文本编辑器中打开 `/etc/opt/SUNWsamfs/migrationd.cmd` 文件。

在示例中，在 `vi` 编辑器中打开新文件并添加一个初始注释：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
# /etc/opt/SUNWsamfs/migrationd.cmd
# A configuration file for migrating data from old tape volumes to replacements
```

- 如果只需迁移少量卷，请指定每个源卷、目标卷和迁移方向。对于每个源卷，输入 `migrate = from source to destination` 格式的行，其中：
 - `media_type` 是用来标识存放源的介质种类的两字母代码（有关详细信息，请参见附录 A, 设备类型词汇表）。
 - `VSN` 是用来标识磁带库中磁带卷的唯一卷序列号。

在示例中，要将数据从旧的 LTO (`li`) 磁带 `VOL305` 迁移到新的 Oracle StorageTek T10000 (`ti`) 磁带盒 `VOL820`：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
# /etc/opt/SUNWsamfs/migrationd.cmd
# A configuration file for migrating data from old tape volumes to replacements
# Migrate a single volume.
migrate = from li VOL305 to ti VOL820
```

- 如果您需要迁移大量卷，则需要为源卷和目标卷定义介质池。通过输入 `vsnpool = poolname library equipment_number media_type VSNlist` 格式的行定义每个池，其中：
 - `name` 唯一地标识池。
 - `equipment_number` 是 `mcf` 为存放源卷的磁带库分配的序号。
 - `media_type` 是用来标识存放源的介质种类的两字母代码（有关详细信息，请参见附录 A, 设备类型词汇表）。
 - `VSNlist` 是用来标识 VSN 组和范围的用空格分隔的文本 VSN 和/或正则表达式。

在示例中，将数据从旧的 LTO4 (`li`) 磁带卷迁移到新的 LTO6 (`ti`) 磁带盒。可以为源池 `pool1` 添加一行，以表示磁带库 `20` 中将迁移的 LTO4 卷。这些包括 VSN 位于 `VOL000` 到 `VOL299` 范围内的卷，以及两个单独的卷 `VOL300` 和 `VOL304`。然后，为目标池 `pool2` 添加一行，以表示磁带库 `30` 中 LTO6 卷的范围。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
# /etc/opt/SUNWsamfs/migrationd.cmd
# A configuration file for migrating data from old tape volumes to replacements
# pool1 contains the source volumes
vsnpool = pool1 library 20 li ^VOL[0-2][0-9][0-9] VOL300 VOL304
# pool2 contains the destination volumes
```

```
vsnpool = pool2 library 30 li ^VOL50[0-9]
```

5. 如果已经定义了源和目标介质池，则指定迁移方向。输入 `migrate = from sourcepool to destinationpool` 格式的行，其中：
 - `sourcepool` 是包含将迁移的数据的介质池。
 - `destinationpool` 是将接收迁移数据的介质池。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
vsnpool = pool1 library 20 li ^VOL[0-2][0-9][0-9] VOL300 VOL304
# pool2 contains the destination volumes
vsnpool = pool2 library 30 ti ^VOL50[0-9]
# Migrate data from tapes in pool1 to tapes in pool2.
migrate = from pool1 to pool2
```

6. 如果您打算以独占的方式使用服务器复制方法，则禁用 `xcopy` 功能。输入 `xcopy = off` 格式的行。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
# Disable xcopy and the StorageTek T10000 Extended Copy feature.
xcopy = off
```

7. 如果您打算以独占的方式使用 StorageTek T10000 扩展复制功能，而且不打算在支持此功能的驱动器不可用时迁移数据，则仅启用 `xcopy` 迁移。输入 `xcopy = only` 格式的行。

在示例中，仅启用 `xcopy`。如果源驱动器或目标驱动器不支持扩展复制功能，则迁移软件将自动取消迁移：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
# Enable xcopy, StorageTek T10000D Extended Copy feature.
# If the source or destination is not xcopy capable, cancel migration.
xcopy = only
```

8. 如果您打算尽可能利用 StorageTek T10000 扩展复制功能，则启用 `xcopy` 迁移方法。输入 `xcopy = on` 格式的行。

在示例中，启用 `xcopy`，即便在迁移期间，兼容的驱动器可能并非始终可用。如果源驱动器或目标驱动器不支持扩展复制功能，则迁移软件将自动切换到服务器复制模式：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
```

```
# Enable xcopy, StorageTek T10000D Extended Copy feature.
# If the source or destination is not xcopy capable, automatically switch
# to the server buffer copy.
xcopy = on
```

9. 如果您打算使用 *xcopy* 方法迁移包含几个过期文件的磁带卷，则将 *xcopy* 设置为在数据末尾 (*eod*) 模式下运行。输入 *xcopy_eod = on* 格式的行。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
xcopy = on
xcopy_eod = on
```

10. 如果您打算使用 *xcopy* 方法迁移包含大量过期文件的磁带卷，则将 *xcopy* 设置为在重新打包模式下运行。输入 *xcopy_eod = off* 格式的行。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
xcopy = on
xcopy_eod = off
```

11. 指定在 *xcopy* 会受到优先级较高的归档或回写任务的干扰之前必须复制的最小数据量。输入 *xcopy_minsize = amountunits* 格式的行，其中：

- *amount* 是整数。
- *units* 为 *k* (表示千字节)、*M* (表示兆字节)、*G* (表示千兆字节)、*T* (表示兆兆字节)、*P* (表示千兆兆字节) 或 *E* (表示艾字节)。

此值为更多其他任务定义 T10000 磁带机的高效利用率和驱动器的可用性之间的折衷。值越大，向驱动器中写入数据时越高效。值越小，用于归档和回写的驱动器的可用性越高。在示例中，将最小副本大小设置为 30 千兆字节：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
xcopy_eod = on
# xcopy can be interrupted after 30GB copied.
xcopy_minsize = 30G
```

12. 定义允许迁移作业运行的每日时间段。输入 *runtime = window* 格式的行，其中 *window* 是以下值之一：

- *always* 使迁移守护进程在不需要归档或回写驱动器和介质时迁移数据。如果迁移守护进程正在使用需要归档或回写的驱动器或介质，则迁移守护进程会移交它们。
- *start_time end_time*，其中 *start_time* 和 *end_time* 分别是允许的时间段开始和结束的时间，用 24 小时制 (*HHMM*) 表示。

在任何时间都可以通过发出命令 `samcmd`、`migstart`、`migidle` 或 `migstop` 覆盖此指令。

当回写程序或归档程序需要卷和驱动器时，迁移服务会放弃这些卷和驱动器。因此，除非遇到归档或回写问题（例如，在高峰时间），否则应当接受默认值 `always`：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
xcopy_minsize = 30G
# Run all of the time. Migration daemon will yield VSNs and drives when
# resources are wanted by the SAM-QFS archiver and stager.
run_time = always
```

13. 通过指定日志目录启用日志记录。输入 `logdir = path` 格式的行，其中 `path` 是目录的路径和名称。

在定义了目录之后，迁移守护进程记录从每个源卷迁移的每个归档文件的目标。每个卷都有一个名为 `media_type.VSN` 的日志文件，其中：

- `media_type` 是用来标识源介质种类的两字母代码（有关详细信息，请参见附录 A, 设备类型词汇表）。
- `VSN` 是用来标识源卷的唯一卷序列号。

因此，例如，具有 VSN `VOL300` 的源卷的日志文件将命名为 `li.VOL300`。

与归档程序日志一样，在灾难恢复期间，这些迁移日志可能会非常宝贵（有关详细信息，请参见“了解恢复点和归档日志”和《Oracle Hierarchical Storage Manager and StorageTek QFS Software 文件系统恢复指南》）。因此，如果可以，请始终指定日志目录。选择一个不会受 Oracle HSM 软件失败或硬件故障影响的位置，如 `/var/adm/`。在示例中，指定 `/var/adm/hsm_migration_logs` 目录：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
run_time = always
# Log directory for the migration logs.
logdir = /var/adm/hsm_migration_logs
```

14. 为迁移 inode 数据库指定主目录。输入 `dbdir = path` 格式的行，其中，`path` 是目录的绝对路径名称。

会为每个源卷创建一个 inode 数据库并在迁移期间维护该数据库。会为在源卷上找到的每个归档副本创建一个 224 字节的数据库记录。因此，您选择的位置必须具有足够的磁盘空间，以便能够容纳可以放在源介质上的最大数量的副本。例如，每个 Oracle StorageTek T10000D 卷最多可以存放 8,200,104,892 个归档副本。因

此，对于将在任何给定时间迁移的每个 T10000D 卷，您将大约需要 1.67 TB 数据库空间（有关详细信息，请参见 *migration.cmd (1m)* 手册页）。

默认数据库位置是 `/var/opt/SUNWsamfs/sammig/db`。在示例中，指定默认目录：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
logdir = /var/adm/hsm_migration_logs
# database home directory
dbdir = /var/opt/SUNWsamfs/sammig/db
```

15. 为目标设备设置迁移缓冲区大小。输入 `bufsize = media_type blocks` 格式的行，其中：

- `media_type` 是用来标识存放源的介质种类的两字母代码（有关详细信息，请参见附录 A, [设备类型词汇表](#)）。
- `size` 是 `[2-8192]` 范围中的整数，其中，整数值指定缓冲区应当能够存放的磁带块数。默认值为 64。

在示例中，分配了足以存放默认数量的 Oracle StorageTek T10000 磁带块的空间。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
# database home directory
dbdir = /var/opt/SUNWsamfs/sammig/db
# allocate buffer space for 64 T10000D tape blocks
bufsize = ti 64
```

16. 指定每个磁带库可用于迁移的最大驱动器数。输入 `max_drives = library-list` 格式的行，其中：

- `library-list` 是用空格分隔的磁带库条目列表，每个条目都采用 `library equipment-number device-count` 格式。
- `equipment-number` 是为 `mcf` 文件中的磁带库分配的设备序号。
- `device-count` 是可以用在指定磁带库中的驱动器数。默认情况下，`device-count` 设置为磁带库中的驱动器数。

当回写程序或归档程序需要卷和驱动器时，迁移服务会放弃这些卷和驱动器。因此，除非遇到归档或回写问题，否则应当接受默认设置并允许迁移功能使用任何可用驱动器。在示例中，发现了的确需要限制驱动器的用法。因此，我们在磁带库 20 中分配八个要迁移的驱动器，在磁带库 30 中分配六个要迁移的驱动器，在磁带库 40 中分配两个要迁移的驱动器：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
```



```

...
dbdir = /var/opt/SUNWsamfs/sammig/db
# allocate buffer space for 64 T10000D tape blocks
bufsize = ti 64
# For migration, use 8 drives in library 20, 6 in 30, and 2 in 40
max_drives = library 20 8 library 30 6 library 40 2

```

17. 指定同时运行的与迁移相关的复制操作的最大数量。输入 `max_copy = processes` 格式的行，其中，`processes` 是整数。

默认值是最大值，等于 `mcf` 文件中列出的所有磁带库中所配置的驱动器数除以 2。在示例中，最多允许八个同时进行的复制进程：

```

root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
bufsize = ti 64
# For migration, use 8 drives in library 20, 6 in 30, and 2 in 40
max_drives = library 20 8 library 30 6 library 40 2
# Up to 8 sam-migcopy process can be run simultaneously.
max_copy = 8

```

18. 指定同时运行的与迁移相关的磁带扫描操作的最大数量。输入 `max_scan = processes` 格式的行，其中，`processes` 是整数。

要标识迁移源 VSN 上的归档副本，`sam-migrationd` 守护进程扫描 `mcf` 中配置的所有文件系统，从磁盘高速缓存中读取所有的 inode，并将每个 inode 中的 `vsni` 字段与迁移源卷的卷序列号 (Volume Serial Number, VSN) 进行比较。该进程会增加文件系统中的元数据活动，并可能会因此对文件系统性能产生负面影响。

选择一个可以将可接受的文件系统性能与迁移速度最好地进行平衡的值，或者针对大多数用法接受默认值 4。如果打算停止文件系统以实现最快的迁移，请将 `max_scan` 设置为 0，以便所有的源卷仅扫描一次。在示例中，根据经验了解，最多可以允许同时有八个扫描进程，而不会影响正常的文件系统操作：

```

root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
bufsize = ti 64
# For migration, use 8 drives in library 20, 6 in 30, and 2 in 40
max_drives = library 20 8 library 30 6 library 40 2
# Run up to 8 sam-migcopy processes simultaneously.
max_copy = 8
# Scan up to 8 VSNs simultaneously.
max_scan = 8

```

19. 保存文件并关闭编辑器。

```

root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
max_copy = 8
# Scan up to 8 VSNS simultaneously.
max_scan = 8
:wq
root@solaris:~#

```

检查处于活动状态的迁移作业

本节中的说明描述如何使用 `samcmd` 命令从 shell 命令提示符输入命令。但是，请注意，所有的命令还可以从 `samu` 界面以 `:command` 格式输入，其中 `command` 是命令的名称。

1. 如果尚未登录到 Oracle HSM 元数据服务器，则以 `root` 用户身份登录。

```
root@solaris:~#
```

2. 确保上一个迁移当前处于非活动状态或者已完成。首先，检查当前的迁移状态。使用 `samcmd x` 命令。

如果另一个迁移复制作业正在进行，则以下命令会按介质类型和 VSN、复制模式、完成百分比和当前的复制状态列出源和目标卷：

```

root@solaris:~# samcmd x
Migration status  samcmd  version HH:MM:SS month day year
samcmd on hsm61sol
Status: Stop: Waiting for :migstart
source  dest    cmod perc status
li VOL004 li VOL042 - 60% Copy idled

```

否则，如果没有其他迁移复制作业正在运行，则以下命令将不列出任何作业：

```

root@solaris:~# samcmd x
Migration status      samu      ver  time date
Source Vsns - wait:  0 fsscan: 0 copy: 0 update ino: 0 log: 0 done:  0
Status: Idle: Waiting for :migstart
source dest  cmod  perc  status

```

3. 接着，检查当前任何源 (S) 和/或目标 (D) 卷的状态。使用命令 `samcmd y`。

在第一个示例中，所列出的唯一源卷和目标卷的作业 `end time` 为 `10/16 12:14`。源卷的复制已完成。因此当前没有运行任何作业：

```
root@solaris:~# samcmd y
```

```

Migration vsn list  samcmd  version HH:MM:SS month day year
Status:  Run  Vsns:2 src:1 dest:1 maxcopy:2
ord m ty vsn      start time  end time   status  Inodes done/tot   bytes
  0 s li VOLa01  10/16 12:12 10/16 12:14 complete 35023/35023 550.00M
  0 D li VOLa80  10/16 12:12 10/16 12:14 avail          550.00M

```

在第二个示例中，源卷和目标卷的作业 *end time* 为 *none*。源卷仍复制到目标卷。因此，迁移作业仍在运行：

```

root@solaris:~# samcmd y
Migration vsn list  samcmd  version HH:MM:SS month day year
Status:  Run  Vsns:2 src:1 dest:1 maxcopy:2
ord m ty vsn      start time  end time   status  Inodes done/tot   bytes
  0 s li VOLa02  10/16 12:12 none      copy          0/35023 164.50M
  0 D li VOLa81  10/16 12:12 none      copy          148.75M

```

4. 最后，检查磁带库目录中的卷列表。使用命令 `samcmd v`。在输出中查找以下标志：
 - *R* 表示卷是只读的。当迁移启动时，源卷标记为只读。
 - *s* (源) 表示仍从该卷复制数据。
 - *D* (目标) 表示仍将数据复制到该卷。
 - *m* 表示源卷已完成迁移。
 - *e* 表示源卷由于错误无法迁移。

在示例中，卷 `VOLa01` 成功迁移到 `VOLa80`。卷 `VOLa02` 仍然正在迁移到 `VOLa81`，但无法迁移。

```

root@solaris:~# samcmd v
Robot catalog  samcmd  version HH:MM:SS month day year
Robot VSN catalog by slot      : eq 800
slot          access time count use flags          ty vsn
count 64
  0    2015/06/29 17:00    1 95% -il---b--Rm-  li VOLa01
  1    2015/07/02 17:43    2 89% -il-o-b--RS-  li VOLa02
  2    2015/07/02 18:31    2 89% -il-o-b--Re-  li VOLa03
  ...
 51    2015/10/16 15:18    2 82% -il-o-b----- li VOLa80
 52    2015/10/16 15:25    2 84% -il-o-b---D-  li VOLa81

```

5. 如果作业正在运行，请等待它们完成。
6. 否则，在确保当前没有运行任何迁移之后，[迁移卷](#)。

迁移卷

本节中的说明描述如何使用 `samcmd` 命令从 shell 命令提示符输入命令。但是，请注意，所有的命令还可以从 `samu` 界面以 `:command` 格式输入，其中 `command` 是命令的名称。

1. 如果尚未登录到 Oracle HSM 元数据服务器，则以 `root` 用户身份登录。

```
root@solaris:~#
```

2. 确保源文件系统已挂载。
3. 激活 `migrationd.cmd` 文件。使用命令 `samcmd migconfig`。

如果配置成功，则命令显示 `Configuring migration` 消息并指引您查看日志文件以了解详细信息：

```
root@solaris:~# samcmd migconfig
samcmd: migconfig: Configuring migration (see /var/opt/SUNWsamfs/sammig/logfile)
root@solaris:~#
```

否则，该命令会停止并出现错误。要么是在发出配置命令之前已经无法停止迁移过程，要么是在磁带机仍等待迁移时已停止迁移：

```
root@solaris:~# samcmd migconfig
samcmd: migconfig: Can't configure migration, migration status is not stop, or
migration job is pending
root@solaris:~#
```

4. 显示迁移配置。使用命令 `samcmd y`。

如果配置成功，则列出指定的卷，源卷的状态为 `sched_wait`（已计划，正在等待），目标卷的状态为 `avail`（可用）。在示例中，配置已成功：

```
root@solaris:~# samcmd y
Migration vsn list  samcmd  version HH:MM:SS month day year
samcmd on hsm61sol
Status: Stop: Waiting for :migstart  Vsns:2 src:1 dest:1 maxcopy:2
ord m ty vsn  start time  end time  status  Inodes done/tot  bytes
  0 S li VOL001 none      none      sched_wait      0/0      0
  0 D li VOL501 none      none      avail           0/0      0
```

5. 如果配置成功，则启动迁移。使用命令 `samcmd migstart`。

```
root@solaris:~# samcmd migstart
samcmd: migstart: State changed to start
```

```
root@solaris:~#
```

6. 检查迁移状态。使用命令 `samcmd x` 和 `samcmd y`。

在示例中，迁移刚刚启动。`Migration status` 屏幕显示作业状态现在为 `Run`，1 个副本正在使用 `s`（服务器）复制模式 (`cmode`)，复制完成的进度为 `0%`，`0` 个 inode 已更新，源卷状态仍为 `Loading`：

```
root@solaris:~# samcmd x
Migration status   samcmd  version HH:MM:SS month day year
Source Vsns - wait: 0 fsscan: 0 copy: 1 update ino: 0 log: 0 done: 0
Status: Run
source    dest      cmode perc status
li VOL001 li VOL501 s      0%   Loading li.VOL001
```

`Migration vsn list` 屏幕显示 2 个卷当前正在处理，1 个源和 1 个目标。这两个卷的状态现在是 `copy`，这表示源正复制到目标。此时，已将 `0` 字节从源复制到目标，`35023` 个 inode 中的所有 inode 都尚未更新：

```
root@solaris:~# samcmd y
Migration vsn list   samcmd  version HH:MM:SS month day year
Status: Run Vsns:2 src:1 dest:1 maxcopy:2
ord m ty vsn      start time  end time  status  Inodes done/tot  bytes
 0 S li VOL001  10/16 12:12 none    copy    0/35023    0
 0 D li VOL501  10/16 12:12 none    copy    0/35023    0
```

7. 再次使用命令 `samcmd x` 和 `samcmd y` 定期重新检查迁移状态。

在示例中，`Migration status` 屏幕显示复制现在完成了 `23%`，已经从源读取了 `560 (0x00000230)` 个磁带块：

```
root@solaris:~# samcmd x
Migration status   samcmd  version HH:MM:SS month day year
Source Vsns - wait: 0 fsscan: 0 copy: 1 update ino: 0 log: 0 done: 0
Status: Run
source    dest      cmode perc status
li VOL001 li VOL501 s      24% 0x00000230 blocks read
```

`Migration vsn list` 屏幕显示已从源卷读取 `164.50` 兆字节，已向目标卷中写入 `148.75` 兆字节：

```
root@solaris:~# samcmd y
Migration vsn list   samcmd  version HH:MM:SS month day year
Status: Run Vsns:2 src:1 dest:1 maxcopy:2
```

```

ord m ty vsn      start time  end time  status  Inodes done/tot    bytes
  0 S li VOL001  10/16 12:12 none    copy           0/35023  164.50M
  0 D li VOL501  10/16 12:12 none    copy           0/35023  148.75M

```

8. 当迁移完成时，检查最终状态。使用命令 `samcmd x` 和 `samcmd y` 并检查迁移日志文件：

在示例中，屏幕上不再列出源卷和目标卷 *Migration status*，现在显示 1 个复制现已完成。请注意，迁移状态仍为 *Run*，它将一直保持此状态，直到我们输入 *migidle* 或 *migstop* 命令：

```

root@solaris:~# samcmd x
Migration status    samcmd  version HH:MM:SS month day year
Source Vsns - wait: 0 fsscan: 0 copy: 0 update ino: 0 log: 0 done: 1
Status: Run
source  dest   cmod perc status

```

Migration vsn list 屏幕显示已从源卷读取 550.00 兆字节，已向目标卷中写入 550.50 兆字节：35023 个 inode 都已经进行更新，以反映已迁移的归档副本的新位置：

```

root@solaris:~# samcmd y
Migration vsn list  samcmd  version HH:MM:SS month day year
Status: Run Vsns:2 src:1 dest:1 maxcopy:2
ord m ty vsn      start time  end time  status  Inodes done/tot    bytes
  0 S li VOL001  10/16 12:12 10/16 12:14 complete 35023/35023 550.00M
  0 D li VOL012  10/16 12:12 10/16 12:14 avail           0/0         550.00M

```

迁移守护进程的日志文件会列出迁移的每个阶段并最终给出摘要。在示例中，使用 Solaris *tail* 命令查看最新的条目：

```

root@solaris:~# tail /var/opt/SUNwsamfs/sammig/logfile
date time Info: Schedule: Create VsnList file.
date time Info: Schedule: VsnList file created, source: 1, destination: 1.
date time Info: Schedule: Migration status changed to Start.
date time Info: 'li.VOL001' Filesystem scan: Started
date time Info: 'li.VOL001' Filesystem scan: Completed, total copy bytes: 517.2M,
inodes: 35023, multi vsn copy: 0, removable-media file: 0, obsolete copy: 0
date time Info: 'li.VOL001' Copy: Started, pid: 2459 destination 'li.VOL012'
date time Info: 'li.VOL001' Copy: Mode - server copy
date time Info: 'li.VOL001' Copy: Server copy started from position 0x4.
date time Info: 'li.VOL001' Copy: Tar header check started from position 0x4.
date time Info: 'li.VOL001' Copy: Tar header check succeeded, 5 inodes checked, 0
tar header error found.

```

```

date time Info: 'li.VOL001' Copy: Completed, pid: 2459, exit status: 12, signal: 0
date time Info: 'li.VOL001' Update inode: Started, source position: 0
date time Info: 'li.VOL001' Update inode: Completed.
date time Info: 'li.VOL001' Log: Started, source position: 0
date time Info: 'li.VOL001' Log: Completed.
date time Summ: 'li.VOL001'
date time Summ: 'li.VOL001' ===== Summary =====
date time Summ: 'li.VOL001' Status:      Complete
date time Summ: 'li.VOL001' Copy mode:  Server copy
date time Summ: 'li.VOL001' Start at:   date time
date time Summ: 'li.VOL001' End at:     date time
date time Summ: 'li.VOL001' Bytes:      550.00M
date time Summ: 'li.VOL001' Archive copies:                35023
date time Summ: 'li.VOL001' Read error copies:              0
date time Summ: 'li.VOL001' Multi vsn copies:               0
date time Summ: 'li.VOL001' Removable-Media file:          0
date time Summ: 'li.VOL001' ---Dest---   ---Bytes---   ---Copies---
date time Summ: 'li.VOL001' li VOL501      550.00M      35023
root@solaris:~#

```

9. 最后，确保将卷迁移日志复制到安全位置。

这些日志记录从每个源卷迁出的每个归档文件副本的目标卷和开始位置。当您需要恢复文件或文件系统时，此信息至关重要。因此，Oracle 强烈建议您将这些文件的备份副本与恢复点和归档程序日志文件保留在一起，如[第 7 章 备份配置和文件系统](#)中和《Oracle Hierarchical Storage Manager and StorageTek QFS Software 安装和配置指南》中的相应章节中所述。

迁移守护进程会在 *migrationd.cmd* 文件所指定的目录中创建迁移日志文件。对于所迁移的每个卷，它会创建一个名为 *media_type.VSN* 的文件，其中：

- *media_type* 是用来标识源介质类型的两字母代码（有关详细信息，请参见[附录 A, 设备类型词汇表](#)）。
- *VSN* 是用来标识源卷的唯一卷序列号。

在示例中，将卷日志从指定的日志目录 */var/adm/hsm_migration_logs/* 复制到 NFS 挂载的远程文件系统上用来保留文件系统恢复资源的目录：

```

root@solaris:~# ls /var/adm/hsm_migration_logs/
li.VOL001 li.VOL002 li.VOL003 li.VOL004 li.VOL005 li.VOL006 ... ti.801 ...
root@solaris:~# cp /var/adm/hsm_migration_logs/*.* /zfs/recover/hsmfs1/2015mig/

```

10. 在重新归档所有文件之后，会根据您的要求处置磁带（请参见[“迁移之后处置旧介质”](#)）。
11. 在此处停止。迁移已完成。

回写文件并将其重新归档到替换介质中

要使用回写和重新归档方法将归档文件从旧介质迁移到新介质，需要确定要迁移的文件，将其回写到磁盘高速缓存中，然后将其写入新介质，此过程不会干扰正常的文件系统操作。本章介绍该过程中的以下阶段：

- [估算可用资源](#)
- [将归档过程配置为使用新介质](#)
- [将数据迁移到替换介质](#)

估算可用资源

回写和重新归档过程的细节很大程度上取决于两个因素：可用的磁盘存储量和可用的可移除介质驱动器数量。在介质迁移过程中，Oracle HSM 回写程序将旧可移除卷装入可以读取旧介质格式的驱动器并将归档的文件恢复到磁盘高速缓存中。然后 Oracle HSM 归档程序使用可以写入新介质格式的驱动器将文件重新归档至新的可移除卷。因此，理想情况下，一次性将任何给定磁带卷上的所有文件回写到磁盘上，然后直接将其归档到新介质。

为此，必须在迁移期间准备充足的专用资源：

- 与整个磁带容量相当的磁盘空间
- 一个专用的读取旧磁带格式的驱动器
- 一个专用的写入新格式的驱动器。

如果您可以停止文件系统直到迁移完成，则上述各项不是问题。但在生产设置下迁移数据，而不过多干扰正在进行的文件系统和归档操作，还需要考虑一些细节。如果可用磁盘空间或磁带机不足，需要确定可以为迁移准备的合理备用资源，然后调整迁移过程。因此执行如下操作：

1. 估算可以用于迁移但不会影响正常的文件系统操作的磁盘高速缓存量。
2. 估算可以专供迁移的磁带机数量。

如果可用磁带机数量有限，需要计划减少回写和归档进程，以免迁移过程影响正常的操作。

3. 根据上面的估算结果，确定回写和归档参数。确定可用磁盘空间在任何时刻可以容纳的最大迁移文件数，以及允许文件从高速缓存移出并移入新介质的最大速率。
4. 在估算了资源之后，计划旧介质的迁移后处置。

将归档过程配置为使用新介质

将新介质添加到 `archiver.cmd` 文件中并修改归档复制指令，以便始终使用新介质创建一个副本。

1. 在文本编辑器中打开 `/etc/opt/SUNWsamfs/archiver.cmd` 文件。

归档策略指定两个副本，两者都写入要替换的介质类型。在示例中，用 `vi` 编辑器打开文件。此处要替换 DLT 磁带（键入 `lt`）：

```
root@solaris: vi /etc/opt/SUNWsamfs/archiver.cmd
# =====
# /etc/opt/SUNWsamfs/archiver.cmd
# -----
...
# -----
# VSN Directives
vsns
allfiles.1 lt .*
allfiles.2 lt .*
endvsns
```

2. 在副本 2 的指令中，将指定的介质类型更改为新介质的标识符，保存文件并关闭文本编辑器。

在示例中，需要将旧 DLT 磁带上的数据迁移到新 LTO 磁带上。因此，在副本 2 中，将旧介质类型 `lt` (DLT) 更改为 `li` (LTO)：

```
root@solaris: vi /etc/opt/SUNWsamfs/archiver.cmd
# =====
# /etc/opt/SUNWsamfs/archiver.cmd
# -----
...
# -----
# VSN Directives
vsns
allfiles.1 lt .*
allfiles.2 li .*
endvsns
:wq
root@solaris:~#
```

3. 检查 `archiver.cmd` 文件是否存在语法错误。运行 `archiver -lv` 命令，更正错误，直至没有错误。

`archiver -lv` 命令将按行输出文件。如果有错误，会在出现错误的地方停止运行。

```
root@solaris:~# archiver -lv
Reading '/etc/opt/SUNWsamfs/archiver.cmd'.
1: # =====
```

```
2: # /etc/opt/SUNWsamfs/archiver.cmd
3: # -----
4: # Global Directives
5: logfile = /var/opt/SUNWsamfs/archiver.log
6: # -----
7: # File System Directives:
8: fs = samqfsms
9: all .
10: 1 5m ...
root@solaris:~#
```

4. 当修改后的 *archiver.cmd* 文件中没有错误后，使用命令 *samd config* 将其装入当前配置中：

```
root@solaris:~# samd config
Configuring SAM-FS
root@solaris:~#
```

5. 接着，在磁带盒之间迁移数据。

将数据迁移到替换介质

用来迁移数据的回写和归档方法使用 *sfind*，这是 GNU *find* 命令的 Oracle HSM 扩展。*sfind* 命令用于在指定磁带卷上查找文件，并对找到的所有文件启动 *stage* 和 *rearchive* 命令。

如果您不熟悉 *sfind*、*stage* 和/或 *rearchive* 命令，现在应查看各自的手册页。然后，对存储了必须迁移的数据的每个盒式磁带，执行如下操作：

将数据从一个磁带迁移到另一个磁带

1. 以 *root* 用户身份登录到文件系统主机。

```
root@solaris:~#
```

2. 移至要迁移的文件所在文件系统的挂载点目录。

在示例中，要迁移挂载在 */hsm/hsmfs1* 上的 *hsmfs1* 文件系统中所存储文件的归档副本：

```
root@solaris:~# cd /hsm/hsmfs1
root@solaris:~#
```

3. 选择磁带卷。

在介质类型之间迁移数据时，一次处理一个卷。在以下示例中，使用卷序列号 `VOL008`。

4. 首先，搜索无法成功回写的已损坏文件的选定卷。使用 Oracle HSM 命令 `sfind . -vsn volume-serial-number -damaged`，其中 `volume-serial-number` 是在库中唯一标识卷的字母数字字符串。

在示例中，从当前工作目录 (`.`) 开始搜索。`-vsn` 参数将搜索限定于当前磁带 `VOL008` 上找到的文件。`-damaged` 标志将搜索限定于无法成功回写的文件。

```
root@solaris:~# sfind . -vsn VOL008 -damaged
```

5. 如果 `sfind` 在搜索已损坏文件时返回任何结果，请尝试修复相应文件。使用命令 `undamage -m media-type -vsn volume-serial-number file`，其中：
 - `media-type` 为附录 A, 设备类型词汇表中列出的由两个字符表示的介质类型代码之一。
 - `volume-serial-number` 是唯一地标识卷的字母数字字符串。
 - `file` 是损坏的文件的路径和名称。

有时，短暂的 I/O 错误将导致副本被标记为已损坏。Oracle HSM `undamage` 命令可清除此状态。在示例中，归档文件副本 `/hsm/hsmfs1/data0008/20131025DAT` 被报告为已损坏。因此，我们将其设置为未损坏并重新尝试搜索已损坏的文件：

```
root@solaris:~# sfind . -vsn VOL008 -damaged
/hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# undamage -m lt -vsn VOL008 /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind . -vsn VOL008 -damaged
```

6. 如果 `sfind` 命令再次将文件列为已损坏，该副本将不可用。请查看归档中是否包含该文件的其他未损坏副本。要列出可用副本，请使用命令 `sls -D file`，其中 `file` 是文件的路径和名称。要检查所发现的任何副本的状态，请使用命令 `sfind file -vsn volume-serial-number`。

在本示例中，`undamage` 命令无法修复副本。因此，使用 `sls` 列出文件 `/hsm/hsmfs1/data0008/20131025DAT` 的所有副本：

```
root@solaris:~# undamage -m lt -vsn VOL008 /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind . -vsn VOL008 -damaged
/hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sls -D /hsm/hsmfs1/data0008/20131025DAT
20131025DAT:
mode: -rw-r--r--  links:  1  owner: root      group: other
      length:    319279  admin id:    7  inode: 1407.5
      project: system(0)
      offline; archdone; stage -n;
```

```

copy 1: ---- May 21 07:12      1e4b1.1    lt VOL008
copy 2: ---- May 21 10:29      109c6.1    lt VOL022
...

```

磁带卷 *VOL022* 保存文件的第二个副本。因此我们使用 *sfind* 检查第二个副本：

```
root@solaris:~# sfind /hsm/hsmfs1/data0008/20131025DAT -vsn VOL022 -damaged
```

7. 如果某个副本不可用并且存在文件的一个未损坏的副本，则重新归档文件。然后，归档保存两个正常的副本后，取消归档已损坏的副本。

在示例中，卷 *VOL008* 上的文件 */hsm/hsmfs1/data0008/20131025DAT* 的副本 1 不可用，但 *sfind* 命令未找到副本 2 的损坏情况。因此我们发出带 *-c* 选项的 *archive* 命令，以便在卷 *VOL008* 上取消归档已损坏的副本之前创建有效的副本 1：

```

root@solaris:~# sfind /hsm/hsmfs1/data0008/20131025DAT -vsn VOL022 -damaged
root@solaris:~# archive -c 1 /hsm/hsmfs1/data0008/20131025DAT
...
root@solaris:~# unarchive -m lt -vsn VOL008 /hsm/hsmfs1/data0008/20131025DAT

```

8. 如果不存在任何可用的副本，请查看文件是否位于高速缓存中。使用命令 *sfind . -vsn volume-serial-number -online*。

在本示例中，卷 *VOL008* 上的副本 1 和卷 *VOL022* 上的副本 2 已损坏且不可用。因此我们查看文件是否在磁盘高速缓存中联机可用：

```

root@solaris:~# undamage -m lt -vsn VOL008 /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind . -vsn VOL008 -damaged
/hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# undamage -m lt -vsn VOL022 /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind /hsm/hsmfs1/data0008/20131025DAT -vsn VOL022 -damaged
/hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind /hsm/hsmfs1/data0008/20131025DAT -online

```

9. 如果不存在任何可用的副本，但文件位于高速缓存中，则归档文件。然后，归档保存两个正常的副本后，取消归档已损坏的副本。

在本示例中，卷 *VOL008* 上的副本 1 和卷 *VOL022* 上的副本 2 均不可用，因此我们发出 *archive* 命令以创建两个有效的副本，然后取消归档卷 *VOL008* 上已损坏的副本：

```

root@solaris:~# undamage -m lt -vsn VOL008 /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind . -vsn VOL008 -damaged
/hsm/hsmfs1/data0008/20131025DAT

```

```

root@solaris:~# undamage -m lt -vsn VOL022 /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind /hsm/hsmfs1/data0008/20131025DAT -vsn VOL022 -damaged
/hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind /hsm/hsmfs1/data0008/20131025DAT -online
/hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# archive /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# unarchive -m lt -vsn VOL008 /hsm/hsmfs1/data0008/20131025DAT

```

10. 如果不存在任何可用的副本，并且该文件不在磁盘高速缓存中，则数据可能已丢失。如果数据非常重要，请咨询专业的数据恢复公司寻求帮助。否则，取消归档已损坏的副本。

在本示例中，卷 *VOL008* 上的副本 1 和卷 *VOL022* 上的副本 2 不可用。*sfind* 命令在磁盘高速缓存上找不到该文件。这些数据不重要。因此我们在卷 *VOL008* 上取消归档损坏的副本：

```

root@solaris:~# undamage -m lt -vsn VOL008 /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind . -vsn VOL008 -damaged
/hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# undamage -m lt -vsn VOL022 /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind /hsm/hsmfs1/data0008/20131025DAT -vsn VOL022 -damaged
/hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind /hsm/hsmfs1/data0008/20131025DAT -online
root@solaris:~# archive /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# unarchive -m lt -vsn VOL008 /hsm/hsmfs1/data0008/20131025DAT

```

11. 如果 *sfind* 在搜索已损坏文件时未返回任何结果，请将当前磁带中的文件回写到磁盘高速缓存。使用命令 *sfind . -vsn volume-serial-number -offline -exec stage {}/;*

-vsn 参数将搜索限定于在当前磁带上找到的文件（始终以一次一个磁带的方式迁移数据）。

-offline 参数进一步将 *sfind* 输出限定于尚不位于高速缓存中的文件，从而不会覆盖数据。

-exec stage {}/; 参数采用 *sfind* 返回的每个路径和文件名并将其用作 Oracle HSM *stage* 命令的参数。然后，*stage* 命令将指定的文件恢复到磁盘高速缓存中。该过程会不断重复，直到已回写所有符合条件的文件为止。

在本示例中，*sfind -vsn VOL008 -damaged* 命令未返回任何输出。因此我们使用 *sfind* 回写在 *VOL008* 上找到且尚未位于高速缓存中的所有文件：

```

root@solaris:~# sfind . -vsn VOL008 -damaged
root@solaris:~# sfind . -vsn VOL008 -offline -exec stage {}/;

```

12. 从磁带回写了文件后，有选择地对其重新归档。使用命令 `sfind . -vsn volume-serial-number -online -exec rearch -r -m media-type {}/;`，其中 `media-type` 是您要从中进行迁移的介质的类型。

`-vsn` 参数将搜索限定于还在当前磁带上找到的文件（始终以一次一个磁带的方式迁移数据）。

`-online` 参数进一步将 `sfind` 输出限定于位于高速缓存中的文件，从而不会覆盖数据。

`-exec rearch -r -m media-type {}/;` 参数采用 `sfind` 返回的每个路径和文件名并将其用作 Oracle HSM `rearch -r -m media-type` 命令的参数。`-r` 参数在子目录中递归运行该过程。`-m` 参数仅重新归档位于源介质上的文件。

在示例中，`-vsn` 参数值为 `VOL008`，`-m` 参数值指定 `lt` 表示 DLT 介质：

```
root@solaris:~# sfind . -vsn VOL008 -online -exec rearch -r -m lt {}/;
```

13. 重复前面的步骤，直至 `sfind` 搜索不再找到文件。
14. 重新归档了所有文件后，按照计划（请参见["迁移之后处置旧介质"](#)）处理磁带。
15. 重复该过程，直至所有旧介质上的数据都迁移至新介质。

迁移之后处置旧介质

在完成迁移之后，旧介质不一定会丢失所有值。因此，请认真考虑应当如何处置它们。

- 至少，在累积了足够多的新恢复点文件以便仅使用新的替换介质恢复文件系统中的任何文件之前，保留旧介质。
- 如果存储空间允许，请无限期保留旧介质。只要兼容的驱动器保持可用，旧介质就可能是宝贵的备份和恢复资源。
- 如果磁带库空间即将耗尽，请导出旧介质并将它们保存在非现场存储中。
- 如果旧介质可以重用而且您确信其中包含的数据不再有用，请为旧卷重新设置标签。例如，您可能要对较旧的 Oracle StorageTek T10000C 驱动器中的介质重新设置标签，并将其用于较新的 T10000D 驱动器。
- 否则，如果旧卷上的数据和介质上的数据都没有剩余价值，请从磁带库中导出卷并对它们进行相应的处置。

附录 A. 设备类型词汇表

主配置文件 (*mcf*) 的 *Equipment Type* 字段的值标识 Oracle Hierarchical Storage Manager and StorageTek QFS Software 内的设备和设备配置。将设备类型指定为两字符代码。此词汇表列出了当使用样例或解释现有 *mcf* 时用于快速参考的代码（有关完整详细信息，请参见 *mcf(4)* 手册页）。

为方便起见，将这些代码分为三部分，然后按字母顺序列出：

- [推荐的设备和介质类型](#)
- [其他设备和介质类型](#)

推荐的设备和介质类型

本节介绍了您通常需要的所有设备代码：通用设备代码 (*rb*、*tp* 和 *od*) 以及用于标识网络连接磁带库接口和 Oracle HSM 历史记录的代码。

通用设备代码 *rb*、*tp* 和 *od* 为所有 SCSI 连接磁带库、磁带机和光盘设备的首选设备类型代码。当您指定通用设备类型时，Oracle HSM 可基于 SCSI 供应商代码自动设置正确的类型。

gxxx

其中 *xxx* 为 $[0-127]$ 范围内的一个整数，表示属于 *ma* 磁盘高速缓存系列集的分散读写磁盘设备组。

hy

Oracle HSM 历史记录，是维护介质目录但没有关联硬件的可选虚拟库。用于跟踪导出的介质。

ma

在一个或多个专用 *mm* 磁盘设备上维护文件系统元数据的高性能 QFS 文件系统。文件数据驻留在单独的 *md*、*mr* 或 *gxxx* 数据设备上。

md

一个磁盘设备，该设备存储 *ma* 文件系统的文件数据，或 *ms* 文件系统的文件和元数据。*md* 设备将文件数据存储在小至 4 千字节的小型磁盘分配单元 (Disk Allocation Unit, DAU) 中及 16 千字节、32 千字节或 64 千字节的大型 DAU 中。默认 DAU 大小为 64 千字节。

mm

一个磁盘设备，该设备存储高性能 *ma* 文件系统的文件系统元数据。

mr

一个磁盘设备，该设备存储 *ma* 文件系统的文件数据。*mr* 设备将文件数据存储在完全可调整的大型磁盘分配单元 (Disk Allocation Units, DAU) 中，此类磁盘分配单元为 8 千字节的倍数，范围为 8-65528 千字节。默认 DAU 大小是 64 KB。

ms

一个 Oracle HSM 文件系统，该文件系统在存储文件数据的相同设备上维护文件系统元数据。

od

任何 SCSI 连接光盘。Oracle HSM 使用 SCSI 供应商代码自动设置相应的设备类型。

rb

任何 SCSI 连接磁带库。Oracle HSM 使用 SCSI 供应商代码自动设置相应的设备类型。

rd

SAM-Remote 伪设备。在主配置文件 (*mcf*) 中，对应的 *Equipment Identifier* (设备标识符) 字段必须包含伪设备的路径 (例如 */dev/samrd/rd2*)。对应的 *Family Set* (系列集) 字段必须包含 SAM-Remote 服务器的主机名。

sc

SAM-Remote 客户机系统。在主配置文件 (*mcf*) 中，对应的 *Equipment Identifier* (设备标识符) 字段必须包含客户机的 SAM-Remote 客户机配置文件的路径。对应的 *Family Set* (系列集) 字段必须包含服务器的系列集名称。*Additional Parameters* (其他参数) 字段必须包含客户机库目录文件的完整路径。

sk

指向网络连接磁带库的 Oracle StorageTek ACSLS 接口。在主配置文件 (*mcf*) 中，对应的 *Equipment Identifier* (设备标识符) 字段必须包含 ACSLS 接口的参数文件的路径。有关更多信息，请参见 *stk(7)* 手册页。

ss

SAM-Remote 服务器。在主配置文件 (*mcf*) 中，对应的 *Equipment Identifier* (设备标识符) 字段必须包含 SAM-Remote 服务器配置文件的路径。对应的 *Family Set* (系列集) 字段必须包含服务器的系列集名称，该名称必须与客户机上 *mcf* 的 *Family Set* (系列集) 字段中使用的名称一致。

tp

任何 SCSI 连接磁带机。Oracle HSM 使用 SCSI 供应商代码自动设置相应的设备类型。但是，请注意，如果您使用更具体的设备代码 (例如 *li* 和 *ti*)，您必须执行相应的操作。例如，如果在 *mcf* 文件中指定 *li* (LTO) 磁带设备，则不能在 *archiver.cmd* 文件中将同一设备称作 *tp* 设备。

其他设备和介质类型

还支持本节中列出的设备类型。

请注意，在大多数情况下，Oracle 建议使用通用设备类型 *rb*、*tp* 和 *od* 标识 SCSI 连接的磁带库、磁带机和光盘设备。通用设备类型告知 Oracle HSM 使用 SCSI 供应商 ID 动态确定硬件。从一种介质类型迁移到另一种介质类型时，下面的类型代码非常重要；这些代码有时还可用于满足管理需要。但是例如，在主配置文件 (*mcf*) 中使用这些代码会以硬编码方式写入可能与实际硬件不匹配的静态设备配置（不建议这么做）。

ac

Sun 1800、3500 或 L11000 磁带库。

at

Sony AIT-4 或 AIT-5 磁带机。

cy

Cygnnet 光盘库。

d3

StorageTek D3 磁带机。

dm

Sony DMF 磁带库。

ds

DocuStore 或 Plasmon 光盘库。

dt

DAT 4 毫米磁带机。

e8

Exabyte X80 库。

fd

Fujitsu M8100 128 磁轨磁带机。

h4

HP SL48 或 SL24 磁带库。

hc

Hewlett Packard L9-/L20-/L60 系列磁带库。

i7

IBM 3570 磁带机。

ic

IBM 3570 介质转换器。

il

IBM 3584 磁带库。

li

LTO-3 或更高版本的磁带机。

lt

数字线性磁带 (Digital Linear Tape, DLT)、Super DLT 或 DLT-S4 磁带机。

me

Metrum 磁带库。

mf

IBM 多功能光盘驱动器。

mo

5.25 英寸可擦光盘驱动器。

o2

12 英寸 WORM 驱动器。

ov

Overland Data Inc. Neo 系列磁带库。

pd

Plasmon D 系列 DVD-RAM 库。

q8

Qualstar 42xx、62xx 或 82xx 磁带库。

s3

StorageTek SL3000 磁带库。

s9

Oracle StorageTek 97xx 系列磁带库。

se

StorageTek 9490 磁带机。

sf

StorageTek T9940 磁带机。

sg

StorageTek 9840C 或更高版本的磁带机。

sl

Spectra Logic 或 Qualstar 磁带库。

st

StorageTek 3480 磁带机。

ti

StorageTek T10000 (Titanium) 磁带机。

vt

Metrum VHS (RSP-2150) 磁带机。

wo

5.25 英寸 WORM 光盘驱动器。

xt

Exabyte (850x) 8 毫米磁带机。

附录 B. 介质状态标志

介质标志含义如下：

- *A* 表示插槽需要审计。
- *C* 表示插槽中有清洗磁带。
- *D* 表示卷为介质迁移目标。
- *E* 表示卷损坏或清洗介质已过期。
- *L* 表示卷是 Linear Tape File System (LTFS) 卷。
- *N* 表示卷不是 Oracle HSM 格式。
- *R* 表示卷只读（软件标志）。
- *S* 表示卷为介质迁移源。
- *U* 表示卷不可用。
- *W* 表示卷处于物理写保护状态。
- *X* 表示插槽是导出插槽。
- *b* 表示卷有条形码。
- *c* 表示卷可回收。
- *f* 表示归档程序发现卷已满或损坏。
- *d* 表示卷有重复的卷序列号 (volume serial number, VSN)。
- *l* 表示卷有标签。
- *o* 表示插槽已占用。
- *p* 表示卷是高优先级卷。
- -, 显示时使用，表示未设置相应的标志。

例如，`samcmd v` 列出目录信息，包括每个加入目录的卷的介质标志：

```
root@solaris:~# samcmd v 800
Robot catalog samcmd      6.1      16:45:25 Feb 14 2016
samcmd on samqfshost      count 32
Robot VSN catalog by slot : eq 800
slot      access time count use  flags      ty vsn
  0      2014/03/14 11:23 875  0%  -i1-o-b-----  li VOL001
  1      2014/03/13 17:54 866  0%  -i1-o-b-----  li VOL002
  2      2014/03/14 11:26   3  0%  -i1-o-b-----  li VOL003
  3      2014/03/14 10:33   3  0%  -i1-o-b-----  li VOL004
  4      2014/03/14 11:34   5  0%  -i1-o-b-----  li VOL005
  5      2014/03/14 11:32   2  0%  -i1Eo-b----f  li VOL006 MEDIA ERROR
  6      2014/03/13 18:07   2  0%  -i1-o-b-----  li VOL007
  7      2014/03/13 18:07   1  0%  -i1-o-b-----  li VOL008
 10      2014/03/13 18:10   2  0%  -i1-o-b-----  li VOL011
```

附录 C. 共享文件系统中的挂载选项

Oracle Hierarchical Storage Manager and StorageTek QFS Software 共享文件系统可通过多个挂载选项进行挂载。本章将在这些选项的作用范围内对其加以描述。

共享文件系统挂载选项

通过使用 *mount* 命令、在 */etc/vfstab* 文件中输入挂载选项或在 *samfs.cmd* 文件中输入挂载选项，您可以指定大多数的挂载选项。例如，以下 */etc/vfstab* 文件包括了某个共享文件系统的挂载选项：

```
sharefs - /sfs samfs - no shared,mh_write
```

通过使用 *samu* 操作员实用程序，您可以动态更改某些挂载选项。有关这些选项的更多信息，请参见《Oracle Hierarchical Storage Manager and StorageTek QFS Software *samu* 命令参考》。

有关这些挂载选项的更多信息，请参见 *mount_samfs* 手册页。

bg：在后台挂载

bg 挂载选项指定，如果初次挂载操作失败，随后的挂载尝试应在后台进行。在默认情况下，*bg* 是无效的，挂载都将继续在前台进行。

retry：再次尝试文件系统挂载

retry 挂载选项可指定系统应尝试挂载文件系统的次数。默认值为 10000。

shared：声明 Oracle HSM 共享文件系统

shared 挂载选项可将文件系统声明为 Oracle HSM 共享文件系统。该选项必须在 */etc/vfstab* 文件中指定，以便将文件系统挂载为 Oracle HSM 共享文件系统。*samfs.cmd* 文件或 *mount* 命令中有该选项并不会导致产生错误，但它不会将文件系统挂载为共享文件系统。

minallocsz 和 **maxallocsz**：调整分配大小

mount 命令的 *minallocsz* 和 *maxallocsz* 选项可指定空间容量（单位为 KB）。这些选项设置块分配大小的最小值。如果文件不断增大，元数据服务器将在得到附加租约授权后分配块。使用 *-o minallocsz=n* 指定此分配的初始大小。元数据服务器可

根据应用程序的访问模式增加块分配的大小，但最大不超过 `-o maxallocsz=n` 的设置。

可以在 `mount` 命令行、`/etc/vfstab` 文件或 `samfs.cmd` 文件中指定这些 `mount` 选项。

rdlease、wrlease 和 aplease：在 Oracle HSM 共享文件系统中使用租约

默认情况下，主机共享文件时 Oracle HSM 元数据服务器通过向自身和其客户机发出 I/O 租约来维护文件系统一致性。租约用于向共享主机授予权限，使其可在指定期限内对文件执行操作。读取租约允许主机读取文件数据。写入租约允许主机覆盖现有文件数据。附加租约允许主机将添加的数据写到文件末尾。元数据服务器可以根据需要更新租约。

向 Oracle HSM 共享文件系统执行读写操作时，应对数据提供类似 POSIX 系统的操作方式。但是，对于元数据，访问时间的变化可能不会立即在其他主机上反映出来。文件的更改在写入租约将要结束时都被推入磁盘。当获得读取租约时，系统使所有过时的高速缓存页无效，这样就可以显示新写入的数据。

以下挂载选项设置租约的持续时间：

- `-o rdlease= number-seconds` 指定读取租约的最大时间量，以秒为单位。
- `-o wrlease= number-seconds` 指定写入租约的最大时间量，以秒为单位。
- `-o aplease= number-seconds` 指定附加租约的最大时间量，以秒为单位。

对于上述三种情况，`number-seconds` 为 `[15-600]` 范围内的一个整数。每种租约的默认时间均为 30 秒。不能在租约有效期间截取文件。有关设置这些租约的更多信息，请参见 `mount_samfs` 手册页。

如果由于当前元数据服务器停机而更改元数据服务器，您必须在转换时间中增加租约时间，因为只有当所有的租约都已过期，备用元数据服务器才能采取控制。

设置较短的租约时间将导致客户机主机与元数据服务器之间的通信量增大，这是因为租约到期后都必须进行续借。

mh_write：启用多台主机读写

`mh_write` 选项可控制多台主机对同一文件的写访问权限。如果在元数据服务器主机上指定了 `mh_write` 挂载选项，则 Oracle HSM 共享文件系统可允许多台主机同时对同一文件进行读写。如果未在元数据服务器主机上指定 `mh_write`，则在任意时刻，都只有一台主机可对文件执行写操作。

默认情况下 `mh_write` 被禁用，只有一台主机在 `wrlease` 挂载选项指定的持续时间内对文件有访问权限。如果元数据服务器上的 Oracle HSM 共享文件系统在挂载时启用了 `mh_write` 选项，则多台主机可同时对同一文件进行读写。

在元数据服务器上启用了 *mh_write* 时，Oracle HSM 支持以下操作：

- 多个读取器主机和分页 I/O
- 多个读取器主机和/或写入器主机，仅当有写入器时才允许直接 I/O
- 一个附加主机（其他主机执行读或写操作），仅当有写入器时才允许直接 I/O。

使用 *mh_write* 选项挂载文件不会更改锁定行为。无论 *mh_write* 是否有效，文件锁定行为都不变。但是在其他方面，行为可能有所不同。如果同时存在读取器和写入器，Oracle HSM 共享文件系统对文件的所有主机访问使用直接 I/O。因此，其他主机可立即看到页对齐 I/O。但是，非页对齐 I/O 将导致显示过时的数据，甚至将过时数据写入文件。这是因为禁用了可阻止这种情况发生的常规租约机制。

因此，只有当多台主机需要同时对同一文件执行写操作，且托管应用程序执行页对齐 I/O 并协调有冲突的写操作时，您才应当指定 *mh_write* 选项。在其他情况下，可能会发生数据不一致。将 *flock()* 与 *mh_write* 一起使用以便在主机间进行协调无法保证一致性。有关更多信息，请参见 *mount_samfs* 手册页。

min_pool: 设置最小并发线程数

min_pool 挂载选项可为 Oracle HSM 共享文件系统设置最小并发线程数。Oracle Solaris 系统上的默认设置为 *min_pool=64*。此设置意味着 Oracle Solaris 上的线程池中始终至少具有 64 个活动线程。您可以根据共享文件系统活动，在 [8-2048] 的范围内调整 *min_pool* 设置的值。

必须在 *samfs.cmd* 文件中设置 *min_pool* 挂载选项。如果在 */etc/vfstab* 文件或命令行中设置，此挂载选项将被忽略。

meta_timeo: 保留缓存属性

meta_timeo 挂载选项可决定系统在两次元数据信息校验之间的等待时间。默认情况下，系统每三秒刷新一次元数据信息。例如，如果在具有几个新创建文件的共享文件系统中输入 *ls* 命令，可能要等三秒后才能返回有关所有文件的信息。该选项的语法是 *meta_timeo=seconds*，其中 *seconds* 为 [0-60] 范围内的一个整数。

stripe: 指定分散读写分配

在默认情况下，系统使用循环文件分配方法对共享文件系统中的数据文件进行分配。要指定将文件数据分散读写到多个磁盘上，可以在元数据主机和所有潜在元数据主机上指定 *stripe* 挂载选项。请注意，在默认情况下，非共享文件系统使用分散读写方法分配文件数据。

在循环分配中，文件以循环方式在每个分片或分散读写组上创建。可按分片或分散读写组的大小实现对文件的最大处理速度。有关文件分配方法的更多信息，请参见《Oracle Hierarchical Storage Manager and StorageTek QFS Software 安装和配置指南》（Oracle HSM 客户文档库，docs.oracle.com/en/storage）。

sync_meta: 指定元数据写入频率

可以将 *sync_meta* 选项设置为 *sync_meta=1* 或 *sync_meta=0*。

默认设置为 *sync_meta=1*，这意味着每当元数据更改时，Oracle HSM 共享文件系统即将文件元数据写入磁盘。此设置虽然降低了数据处理性能，但可确保数据一致性。如果您要更改元数据服务器，则必须采用这种设置。

如果设置 *sync_meta=0*，则 Oracle HSM 共享文件系统会先将元数据写入缓冲区，然后再写入磁盘。这种延迟写入可提高性能，但如果突发计算机中断，则会降低数据的一致性。

worm_capable 和 def_retention: 启用 WORM 功能

worm_capable 挂载选项允许文件系统支持 WORM 文件。*def_retention* 挂载选项使用 *def_retention=MyNdOhPm* 格式设置默认保留时间。

在此格式中，*M*、*N*、*O* 和 *P* 为非负整数，*y*、*d*、*h* 和 *m* 分别代表年数、天数、小时数和分钟数。可以使用这些单位的任意组合。例如，*1y5d4h3m* 表示 1 年 5 天 4 小时零 3 分钟，*30d8h* 表示 30 天零 8 小时，*300m* 表示 300 分钟。此格式与以前软件版本中的公式向下兼容，在以前软件版本的公式中，保持期是以分钟为单位来指定的。

有关更多信息，请参见《Oracle Hierarchical Storage Manager and StorageTek QFS Software 安装和配置指南》（Oracle HSM 客户文档库，docs.oracle.com/en/storage）。

附录 D. 用于归档的配置指令

本附录列出了用于配置 Oracle Hierarchical Storage Manager 文件系统和相关软件操作的指令。每条指令是由一个或多个逗号分隔的字段构成的单个文本行。相关指令一起存储在 Oracle HSM 命令 (.cmd) 文件中。

本附录的其余部分提供三种主要指令的概述：

- [归档指令](#)
- [回写指令](#)
- [预览请求指令](#)

有关其他信息，请参见 Oracle HSM 手册页。

请注意，您可以从命令行配置 Oracle HSM 命令文件（如下所述），也可以使用 Oracle HSM Manager 软件配置这些命令文件。有关 Oracle HSM Manager 的信息，请参见联机帮助。

归档指令

本节提供了有关构成 *archiver.cmd* 文件的归档指令的用法信息。归档指令定义了用于控制以下各项的归档集：文件复制、使用的介质以及归档软件的整体行为。

有四种基本的归档指令类型：

- [全局归档指令](#)
- [文件系统指令](#)
- [副本参数](#)
- [卷序列号 \(Volume Serial Number, VSN\) 关联指令](#)

全局指令和文件系统指令均控制文件的归档方式。但是归档程序先检验特定于文件系统的指令，然后再检验全局指令。因此，出现冲突时，文件系统指令会覆盖全局指令。同样，在文件系统指令中，列出的第一条指令会覆盖后续的所有冲突指令。

全局归档指令

全局指令用于控制归档程序的整体操作，并可用于优化所有已配置文件系统的操作。全局指令包含单独的关键字或一个关键字后跟等号 (=) 和其他数据字段。全局指令启动 *archiver.cmd* 文件，在文件系统指令的第一个指令处结束。

archivemeta: 控制是否对元数据进行归档

archivemeta 指令控制是否对文件系统元数据进行归档。如果文件系统中的文件经常移动,且目录结构经常发生更改,则对文件系统元数据进行归档。但是,如果目录结构相当稳定,则可以禁用元数据归档,从而减少可移除介质驱动器所执行的操作。默认情况下,不会对元数据进行归档。

此指令的格式如下:

```
archivemeta=state
```

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

元数据的归档过程取决于您使用的超级块是第 1 版还是第 2 版,具体如下:

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

archmax: 控制归档文件的大小

archmax 指令用于指定归档 (.tar) 文件的最大大小。达到 *target-size* 值后,不再向归档文件添加任何用户文件。大型用户文件将写入单个归档文件中。

要更改默认值,请使用以下指令:

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

其中 *media* 是[附录 A, 设备类型词汇表](#)和 *mcf* 手册页中定义的一种介质类型,*target-size* 是归档文件的最大大小。该值与介质有关。默认情况下,写入光盘的归档文件不得超过 5 MB。对于磁带,归档文件的最大默认大小为 512 MB。

将归档文件的大小设置为较大的值或较小的值都各有优缺点。例如,在使用磁带进行归档时,将 *archmax* 设置成较大的值,可以减少磁带机停止和启动的次数。但是,在写入较大的归档文件时,提前到达磁带末尾会浪费大量磁带空间。最佳做法是,不要将 *archmax* 指令设置为超过介质容量的 5%。

此外,您还可为单个归档集设置 *archmax* 指令。

bufsize: 设置归档程序缓冲区大小

默认情况下,系统使用内存缓冲器将需要归档的文件复制到归档介质。可以使用 *bufsize* 指令来设置非默认的缓冲区大小和锁定缓冲区(可选)。这些操作在某些情况下可以提高性能。您可以尝试不同的 *number-blocks* 值。此指令的格式如下:

```
bufsize=media number-blocks [lock]
```

其中：

- *media* 是附录 A, [设备类型词汇表](#)和 *mcf* 手册页中定义的一种介质类型
- *number-blocks* 是 [2-1024] 范围内的数字。默认值为 4。此值乘以该介质类型的 *dev_blksize* 值, 计算结果即为所使用的缓冲区大小。*dev_blksize* 值在 *defaults.conf* 文件中指定。有关详细信息, 请参见 *defaults.conf* 手册页。
- *lock* 指明归档程序在创建归档副本时是否可以使用锁定的缓冲区。

如果指定 *lock*, 则归档程序将在 *sam-arcopy* 操作期间在内存中的归档缓冲区上设置文件锁定。此操作可以避免由于为每个 I/O 请求锁定和取消锁定缓冲区而造成的开销, 从而减少占用系统 CPU 的时间。

仅在配有大量内存的大型系统上, 才必须指定 *lock* 参数。如果内存不足, 则可能会导致内存用尽。只有已为需要归档的文件启用直接 I/O 时, *lock* 参数才有效。默认情况下, 不会指定 *lock* 参数, 并且文件系统会在所有直接 I/O 缓冲区 (包括用于归档的缓冲区) 上设置锁定。

您可以使用归档集副本参数 *-bufsize* 和 *-lock* 为每个归档集指定缓冲区大小和锁定。有关更多信息, 请参见[“归档副本指令”](#)。

drives: 控制用于归档的驱动器数

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

```
drives=media-library count
```

其中 *media-library* 是 *mcf* 文件中定义的自动化库的系列集名称, *count* 是允许用于归档的驱动器数量。

您还可以使用归档集副本参数 *-drivemax*、*-drivemin* 和 *-drives* 来实现此目的。有关更多信息, 请参见[“归档副本指令”](#)。

examine: 控制归档扫描

examine 指令用来设置归档程序用于标识已准备就绪进行归档的文件的 *method*。

```
examine=method
```

其中 *method* 是以下指令之一:

- *noscan* (默认值), 用于指定连续归档。在执行初次扫描后, 仅当目录内容发生更改以及需要进行归档时, 才会对其进行扫描。但不对目录和 inode 信息进行扫描。这种归档方法的性能要好于扫描归档, 尤其是当文件系统中文件的数量大于 1,000,000 时。

- *scan*，用于指定扫描归档。在初次扫描文件系统指令之后，始终会扫描 inode。
- *scandirs* 指定扫描归档。始终会扫描指令。不扫描 inode 信息。

归档程序不扫描设置了 *no_archive* 属性的指令。因此您可以通过在包含不发生更改的文件的指令上设置该属性来减少扫描时间。

- *scaninodes* 指定扫描归档。始终会扫描 inode。不对目录信息进行扫描。

interval：指定归档时间间隔

归档程序定期检查所有已启用归档的已挂载文件系统的状态。检查时间由归档时间间隔控制。归档时间间隔是指对每一个文件系统执行扫描操作的时间间隔。要更改归档时间间隔，请使用 *interval* 指令。

仅当未设置连续归档且未指定 *startage*、*startsize* 或 *startcount* 参数时，*interval* 指令才启动完全扫描。如果已设置连续归档 (*examine=noscan*)，则 *interval* 指令将用作默认的 *startage* 值。此指令的格式如下：

```
interval=time
```

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

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

如果为文件系统设置了 *hwm_archive* 挂载选项，则可以自动缩短归档时间间隔。归档程序在文件系统利用率超过上限时开始进行扫描。*high=percent* 挂载选项用于设置文件系统空间占用的上限。

有关指定归档时间间隔的更多信息，请参见 *archiver.cmd* 和 *mount_samfs* 手册页。

logfile：指定归档程序日志文件

归档程序可以生成一个日志文件，其中包含每一个归档、重新归档或取消归档的文件的有关信息。日志文件连续地记录归档操作。默认情况下不启用归档程序日志文件。要指定日志文件，请使用 *logfile* 指令。此指令的格式如下：

```
logfile=pathname
```

其中的 *pathname* 用于指定日志文件的绝对路径和名称。此外，还可以为单个文件系统设置 *logfile* 指令。

归档程序日志文件对于恢复已损坏或丢失的文件系统至关重要，对于监视和分析也非常有价值。因此，您应启用归档程序日志并对其进行备份。有关更多信息，请参见

《Oracle Hierarchical Storage Manager and StorageTek QFS Software 安装和配置指南》。

notify: 重命名事件通知脚本

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

```
notify=filename
```

其中的 *filename*，用于指定包含归档程序事件通知脚本的文件名称，或指定该文件的完整路径。默认文件名为 `/etc/opt/SUNWsamfs/scripts/archiver.sh`。

归档程序执行此脚本，来根据特定站点的要求处理各种事件。通过对第一个参数使用以下关键字之一来调用该脚本：*emerg*、*alert*、*crit*、*err*、*warning*、*notice*、*info* 和 *debug*。

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

ovflmin: 控制卷溢出

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

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

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

其中 *media* 是[附录 A, 设备类型词汇表](#)和 *mcf* 手册页中定义的一种介质类型，*minimum-file-size* 是应触发卷溢出的最小文件大小。此外，您还可为单个归档集设置 *ovflmin* 指令。

请慎用卷溢出功能，在评估其所产生的影响后，再使用该功能。对于跨多个卷的文件，灾难恢复和回收会比较困难。卷溢出文件不会生成校验和。有关如何使用校验和的更多信息，请参见 *ssum* 手册页。

scanlist_squash: 控制扫描列表合并

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

当为 *on* 时，该指令合并目录树中子目录的扫描列表，以便归档程序从公共父目录向下进行递归扫描。如果文件系统中有许多文件和子目录已发生更改，则扫描列表合并可能会极大地降低归档性能。

setarchdone: 控制 archdone 标志的设置

setarchdone 全局指令控制是否在从不归档的文件上设置 *archdone* 标志。此指令的格式如下:

```
setarchdone=state
```

其中的 *state*, 用于指定状态为 *on* 还是 *off*。如果将 *examine* 指令设置为 *scandirs* 或 *noscan*, 则默认值为 *off*。

archdone 标志告知归档过程忽略标记的文件。通常, 创建某个文件的所有指定副本后, 归档过程会设置 *archdone* 标志, 以便后续归档操作跳过该文件, 除非以后对其进行修改。

但是在将 *setarchdone* 设置为 *on* 时, 归档过程会标识并标记不符合任何归档条件从而从不归档的未归档文件。尽管这可以减少将来的归档开销, 但文件评估会立即增加开销并且可能对性能产生不利影响。

wait: 延迟归档程序启动

wait 指令会促使归档程序等待来自 *samcmd* 命令、*samu* 接口或 Oracle HSM Manager 的启动信号。此指令的格式如下:

```
wait
```

默认情况下, 归档程序会在 *sam-fsd* 初始化命令运行时自动启动。

此外, 还可以为单个文件系统设置 *wait* 指令。

文件系统指令

文件系统指令用于定义特定文件系统的归档行为:

- **fs**: 指定文件系统
- **copy-number [archive-age]**: 指定文件系统元数据的多个副本
- **interval**、**logfile** 和 **scanlist** 作为文件系统指令

fs: 指定文件系统

每条 *fs=file-system-name* 指令都会引入一系列仅适用于给定文件系统 *file-system-name* 的归档指令。此指令的格式如下:

```
fs=file-system-name
```

其中 *file-system-name* 是 *mcf* 文件中定义的文件系统名称。

fs= 指令后面出现的常规指令和归档集关联指令仅应用于指定的文件系统。

***copy-number* [*archive-age*]: 指定文件系统元数据的多个副本**

文件系统元数据包括文件系统中的路径名。如果需要多个元数据副本，可在 *archiver.cmd* 文件中放置副本定义，其位置是紧跟 *fs=* 指令之后。

```
copy-number [archive-age]
```

其中时间采用一个或多个由一个整数和一个时间单位组成的组合来表示。单位包括 *s* (秒)、*m* (分钟)、*h* (小时)、*d* (日)、*w* (周) 和 *y* (年)。如果指令经常发生更改，则指定多个元数据副本可能会导致文件系统过于频繁地挂载元数据磁带卷。因此，默认情况下，Oracle HSM 仅创建一个元数据副本。

在示例中，*fs=samma1* 文件系统的元数据副本 1 在 4 小时 (4h) 后生成，副本 2 在十二小时 (12h) 后生成：

```
# General Directives
archivemeta = off
examine = noscan
# Archive Set Assignments
fs = samma1
1 4h
2 12h
```

***interval*、*logfile* 和 *scanlist* 作为文件系统指令**

有几个指令既可以指定为适用于所有文件系统的全局指令，也可以指定为专用于某一文件系统的指令。以下各节对这些指令进行了说明：

- ***interval***: 指定归档时间间隔
- ***logfile***: 指定归档程序日志文件
- ***scanlist_squash***: 控制扫描列表合并
- ***wait***: 延迟归档程序启动

***archive-set-name*: 归档集分配指令**

归档集分配指令指定要一起归档的文件。您可以使用下面介绍的各种选择条件非常精准地指定文件。但是，除非绝对必要，否则最好不要这样做。通常，配置的归档集数量应尽可能最少而范围应尽可能最广。归档集会独占使用一组归档介质。因此，如果按照非常严格的分配条件定义大量的归档集，会导致介质利用率低、系统开销高以及性能降低。在极端情况下，尽管库中仍有足够的容量，作业也可能会由于缺少可用介质而失败。

每一个归档集分配指令均采用以下格式：

```
archive-set-name path [-access interval [-nftv]] [-after date-time] [-minsize size] [-maxsize size] [-user username] [-group groupname] [-name regex]
```

其中：

- *archive-set-name* 是管理员为归档集定义的名称。

这些名称最多可包含 29 个字符，可以是大写和/或小写字母 [A-Za-z]、数字 [0-9] 以及下划线 (`_`) 的任意组合，只要第一个字符为字母即可。不能包含其他任何字符（包括空格），不能将 Oracle HSM 特殊归档集 *no_archive* 和 *all* 的名称用于您自己的归档集。

- *path* 指定文件系统中从其开始进行归档的子目录相对于挂载点的路径。系统会归档起始目录及其子目录中的所有文件。要包含文件系统中的所有文件，请使用句点 (`.`)。不允许在路径的开头使用斜杠 (`/`)。
- *-access* 将重新归档在 *interval* 指定的时间内未访问的文件，其中 *interval* 是一个整数，后跟以下单位之一：*s*（秒）、*m*（分钟）、*h*（小时）、*d*（日）、*w*（周）和 *y*（年）。

通过此参数可以安排将很少使用的文件从成本较高的介质重新归档到成本较低的介质。软件将验证文件的访问和修改时间，以确保这些时间大于或等于文件的创建时间，且小于或等于文件的检查时间。*-nftv*（no file time validation，非文件时间验证）参数会禁用此验证。

- *-after* 仅归档在 *date-time* 之后创建或修改的文件，其中 *date-time* 是一个表达式，格式为：*YYYY-MM-DD [hh:mm:ss] [Z]*，其中 *YYYY*、*MM*、*DD*、*hh*、*mm* 和 *ss* 均为整数，分别表示年、月、日、小时、分钟和秒。可选的 *Z* 参数用于将时区设置为国际协调时间 (Coordinated Universal Time, UTC)。默认值为 *00:00:00* 和本地时间。
- *-minsize* 和 *-maxsize* 仅归档那些超出或未达到指定的 *size* 的文件，其中 *size* 是一个整数，后跟以下单位之一：*b*（字节）、*k*（千字节）、*M*（兆字节）、*G*（千兆字节）、*T*（兆兆字节）、*P*（千兆兆字节）、*E*（艾字节）。
- *-user username* 和 *-group groupname* 仅归档属于指定的用户和/或组的文件。
- *-name* 归档路径和文件名与正则表达式 *regex* 所定义的模式匹配的所有文件。

归档副本指令

默认情况下，当归档集中文件的归档时限为四分钟时，归档程序将为这些文件编写一个归档副本。要更改默认行为，请使用归档副本指令。归档副本指令必须紧跟在它们相关的归档集分配指令的后面。

归档副本指令以 *copy-number* 值 1、2、3 或 4 开头。数字后面是一个或多个用于指定该副本归档特征的参数。每一个归档副本指令均采用以下格式：

```
copy-number [archive-age] [-release [attribute] [-norelease]][-stage[attribute] [unarchive-age]
```

其中：

- 可选的 *archive-age* 参数是新的或修改后的文件在能够进行归档之前必须在磁盘高速缓存中经历的时间。*archive-age* 应指定为一个或多个由一个整数和一个时间单位组成的组合，其中单位包括 *s*（秒）、*m*（分钟）、*h*（小时）、*d*（日）、*w*（周）和 *y*（年）。默认值为 *4m*（4 分钟）。

- 可选的 `-release` 参数用于在创建归档副本之后立即清除 Oracle HSM 释放程序软件以释放文件所使用的磁盘空间。可选的 `release` 属性为 `-a`、`-n` 或 `-d`。`-a`（关联回写）属性要求软件在访问已从归档集中释放的任一文件时回写所有这些文件。`-n` 属性要求软件直接从归档介质进行读取而从不回写文件。`-d` 属性用于重置默认回写行为。
- 在创建 `-norelease` 所标记的所有副本之前，可选的 `-norelease` 参数不会清除 Oracle HSM 释放程序软件以释放文件使用的磁盘空间。
- `-release -norelease` 一起使用时，要求 Oracle HSM 软件在创建 `-release -norelease` 所标记的所有副本之后立即释放文件所使用的磁盘空间。Oracle HSM 不会等待释放程序进程运行。
- 可选的 `-stage` 参数属性为 `-a`、`-c copy-number`、`-f`、`-I`、`-i input_file`、`-w`、`-n`、`-p`、`-V`、`-x`、`-r`、`-d`，其中：
 - `-a` 要求软件在访问归档集中的任一文件时回写所有这些文件。
 - `-c copy-number` 要求软件根据指定的副本编号进行回写。
 - `-n` 要求软件直接从归档介质进行读取而从不回写文件。
 - `-w` 要求软件等待每个文件成功完成回写后再继续（与 `-d` 或 `-n` 一起使用时无效）。
 - `-d` 用于重置默认回写行为。
- `unarchive-age` 参数指定文件的归档副本在将其取消归档来释放介质空间以供重用之前在归档中所用的时间。时间表示为一个或多个由一个整数和一个时间单位组成的组合，其中单位包括 `s`（秒）、`m`（分钟）、`h`（小时）、`d`（日）、`w`（周）和 `y`（年）。

以下示例包括归档集 `allsamma1` 的两条副本指令。第一条指令直到副本 1 达到 5 分钟 (`5m`) 的归档限时才会释放该副本。第二条指令直到副本 2 达到 1 小时 (`1h`) 的归档限时才会释放该副本，并在副本 2 达到 7 年 6 个月 (`7y6m`) 的取消归档限时取消归档该副本。

```
# Archive Set Assignments
fs = samma1
logfile = /var/adm/samma1.archive.log
allsamma1 .
    1 -norelease 5m
    2 -norelease 1h 7y6m
```

副本参数

副本参数定义如何创建归档集所指定的副本。`archiver.cmd` 文件的归档集副本参数部分以 `params` 指令开头，以 `endparams` 指令结尾：

```
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 10m -startsize 10M -drives 10 -archmax 1G
allfiles.2 -startage 1h -startsize 1G -drives 2 -archmax 10G -reserve set
```

endparams

每个副本参数均采用以下格式：

```
archive-set-name[.copy-number][R] [-startage time] [-startcount count] [-startsize size] [-archmax maximum-size] [-bufsize=number-blocks] [-drivemax maximum-size] [-drivemin minimum-size] [-drives number] [-fillvsns] [-lock] [-offline_copy method] [-sort criterion] [-rsort criterion] [-recycle_dataquantity size] [-recycle_hwm percent] [-recycle_ignore] [-recycle_mailaddr mail-address] [-recycle_mingainpercentage] [-recycle_vsncountcount ] [-recycle_minobs percentage] [-unarchagetime_ref] [-tapenonstop] [-reserve keyword ] [-priority multiplier ranking]
```

其中：

- `archive-set-name` 是文件系统指令中的归档集分配指令或特殊指令 `allsets`（该指令会对所有定义的归档集应用指定的副本参数）所定义的归档集的名称。请先为 `allsets` 设置参数，再为各个归档集指定参数。否则，各个归档集的参数会覆盖为 `allsets` 指定的值，从而使其失去作用。
- `.copy-number` 将指定的副本参数的应用范围限定为 `copy-number` 所指定的归档副本，其中 `copy-number` 为 [1-4] 范围内的一个整数，可选的 `R` 将参数的应用范围限定为重新归档的副本。
- `-startage time` 指定将第一个文件添加到归档请求时与实际开始归档时之间的时间间隔。`time` 应指定为一个或多个由一个整数和一个时间单位组成的组合，其中单位包括 `s`（秒）、`m`（分钟）、`h`（小时）、`d`（日）、`w`（周）和 `y`（年）。默认值为 `2h`（二小时）。
- `-startcount count` 指定一个归档请求中的最小文件个数。当等待归档的文件个数达到此阈值时，便开始进行归档。默认情况下，不设置 `count`。
- `-startsize size` 指定一个归档请求的最小大小（以字节为单位）。当等待归档的文件总大小达到此阈值时，便开始进行归档。默认情况下，不设置 `size`。
- `-archmax` 将归档文件的大小限定为不大于 `maximum-size`，其中 `maximum-size` 取决于介质。对于磁带，归档文件的最大默认大小为 512 MB。写入光盘的归档文件不能超过 5 MB。

有关同名的全局归档指令的说明，请参见“[archmax：控制归档文件的大小](#)”。

- `-bufsize=media-type number-blocks` 将存放归档文件（将其写出到归档介质时）的缓冲区的大小设置为 `number-blocks*dev_blksize`，其中 `number-blocks` 为缓冲的磁带块数，是 [2-32] 范围内的一个整数，`dev_blksize` 是 `defaults.conf` 文件中为该介质类型指定的块大小。默认值为 4。
- `-drivemax` 将使用一个驱动器归档的数据量限定为不大于 `maximum-size` MB，其中 `maximum-size` 是一个整数。默认情况下，不指定 `maximum-size`。

使用 `-drives` 参数指定多个驱动器时，限制写入任意一个驱动器的数据量可以改善驱动器使用情况，这有助于平衡工作负载以及提高驱动器的整体利用率。

- `-drivemin minimum-size` 将使用一个驱动器归档的数据量限定为至少 `minimum-size` MB，其中 `minimum-size` 是一个整数。默认为 `-archmax` 的值（如果指定）或 `defaults.conf` 文件中为该介质类型列出的值。

对写入一个驱动器的数据量设置较低的限制可提高驱动器利用率和效率。设置的 *minimum-size* 应足够大，以便传输时间远远超过装入、定位和卸载介质所需的时间。指定 *-drivemin* 后，仅当数据传输足够大时才使用多个驱动器。

- *-drives number* 将用于归档的驱动器数量限定为最多 *number*，其中 *number* 是一个整数。默认值为 1。

设置较高的最大驱动器数可以在归档集包含大型文件或大量文件时提高性能。如果可用的不同驱动器以不同的速度运行，则指定多个驱动器还可以平衡这些差异并提高归档效率。

- *-fillvsns* 强制归档进程使用可较完整地填充归档介质卷的小型归档文件。

默认情况下，归档程序会选择一个包含足够多空间的卷以容纳归档副本中的所有文件。这样生成的大型归档文件可能无法纳入多数磁带上的剩余容量。由此导致介质的整体利用率较低。*-fillvsns* 参数可解决此问题，但代价是更多的介质挂载、定位操作和卸载，所有这些操作都会降低归档和回写性能。

- *-lock* 会在使用直接 I/O 创建归档副本时强制使用锁定的缓冲区。锁定的缓冲区可阻止对缓冲区进行分页并提高直接 I/O 性能。

如果在可用内存有限的系统上指定 *-lock* 参数，则可能导致内存用尽。默认情况下，不强制使用锁定的缓冲区，而由文件系统保留对归档缓冲区的控制权。

- *-offline_copy method* 指定在已从磁盘高速缓存中释放文件的情况下如何创建归档副本。指定的 *method* 可以为 *direct*、*stageahead*、*stageall* 或 *none*。

只要创建了一个归档副本，即可释放文件，因此其余副本必须通过脱机副本进行创建。通过指定的 *-offline_copy* 方法可以定制复制过程，使其符合可用驱动器的数量以及磁盘高速缓存中可用的空间量。

direct 使用两个驱动器将文件直接从脱机卷复制到归档卷。要确保有足够的缓冲区空间，请在使用此方法时增大 *stage_n_window* 挂载选项所设置的值。

stageahead 在将某归档文件写入目标位置的同时，回写下一个归档文件。

stageall 在进行归档之前使用一个驱动器将所有文件回写到磁盘高速缓存。使用此方法时，请确保磁盘高速缓存足以容纳这些文件。

none（默认值）在将文件复制到归档卷之前会根据需要将其回写到磁盘高速缓存。

- *-sort* 在归档文件之前按 *criterion* 对其进行排序，其中 *criterion* 为 *age*、*priority*、*size* 或 *none*。

age 指定按修改时间进行排序（从最早到最近）。

path（默认值）指定按全路径名进行排序，因此会在归档介质中将相同目录中的文件保存在一起。

priority 指定按归档优先级进行排序（从最高到最低）。

size 按文件大小进行排序（从最小到最大）。

none 指定不对文件进行排序，而是按照它们在文件系统中出现的顺序进行归档。

- *-rsort criterion* 按 *criterion* 对文件进行排序，与 *-sort* 类似，但排序方向相反。
- *-recycle_dataquantity size* 将回收程序计划重新归档的数据量限定为 *size* 字节，其中 *size* 是一个整数。

当回收程序需要排出有效归档文件的归档卷时，它会计划重新归档。请注意，选择进行回收的卷的实际数量可能还取决于 *-recycle_vsncount* 参数。默认值为 1073741824 (1 GB)。

- *-recycle_hwm percent* 设置用于启动可移除介质回收的最大介质利用率百分比（上限，即 *hwm*）。对于磁盘介质，将忽略此参数（请参见下面的 *-recycle_minobs*）。默认值为 95。
- *-recycle_ignore* 阻止实际回收归档集中的任何介质，同时允许回收进程正常运行。用于测试。
- *-recycle_mailaddr mail-address* 将信息性回收程序消息定向至 *mail-address*。默认情况下，不发送邮件。
- *-recycle_mingain* 限定在那些增加的空闲空间至少可达到指定的 *percentage* 的卷中选择进行回收的卷。默认值为 50。
- *-recycle_vsncount* 将回收程序计划进行重新归档的卷的数量限定为 *count*。请注意，选择进行回收的卷的实际数量可能还取决于 *-recycle_dataquantity* 参数。对于磁盘介质，将忽略此参数。默认值为 1。
- *-recycle_minobs* 设置磁盘中归档文件中的过时文件的 *percentage*，该值会触发对有效文件的重新归档以及对原始 *tar* 文件的最终删除。对于可移除介质，将忽略此参数（请参见上面的 *-recycle_hwm*）。默认值为 50。
- *-unarchage* 将用于计算取消归档时限的参考时间设置为 *time_ref*，其中 *time_ref* 为 *access*（文件访问时间，默认值）或 *modify*（修改时间）。
- *-tapenonstop* 在归档文件末尾写入一个磁带标记和一个文件结束 (end-of-file, EOF) 标签，但并不关闭可移除介质文件。这可以加速多个归档文件的传输，但在将整个归档集完全写入磁带之前不能卸载盒式磁带。默认情况下，Oracle HSM 软件会通过向归档文件结尾的文件结束标签后面写入两个额外的磁带标记来关闭磁带文件。
- *-reserve keyword* 会保留一个可移除介质卷以供指定的归档集独占使用。如果某卷最先用于存放归档集中的文件，则软件会为该卷分配一个唯一的保留名称，其依据为指定的一个或多个关键字：*fs*、*set* 和/或以下关键字之一：*dir*（目录）、*user* 或 *group*。

fs 将在保留名称中包括文件系统名：*arset.1 -reserve fs*。

set 将在保留名称中包括归档集分配指令中的归档集名称：*all -reserve set*。

dir 将在保留名称中包括归档集分配指令中指定的目录路径的前 31 个字符。

user 将包括与归档文件关联的用户名：*arset.1 -reserve user*。

group 将包括与归档文件关联的组：*arset.1 -reserve group*。

在某些情况下，按集保留卷会比较方便。但请注意，从本质上来说其效率要低于允许软件选择介质时的效率。保留卷时，系统必须更为频繁地挂载、卸载和定位磁带，这会增大开销而降低性能。非常严格的保留方案会导致可用介质的利用率较低，在极端情况下，还可能由于缺少可用介质而导致归档失败。

- *-priority multiplier ranking* 与上面列出的 *sort priority* 参数一起使用时可更改文件的归档优先级。*ranking* 为 $[(-3.400000000E+38)-3.400000000E+38]$ $(-3.402823466 \times 10^{38}$ 至 $3.402823466 \times 10^{38})$ 范围内的一个实数，*multiplier* 是要针对其更改相关 *ranking* 的归档特征，从以下列表中进行选择：*age*、*archive_immediate*、*archive_overflow*、*archive_loaded*、*copies*、*copy1*、*copy2*、*copy3*、*copy4*、*offline*、*queuwait*、*rearchive*、*reqrelease*、*size*、*stage_loaded* 和 *stage_overflow*。

有关属性的更多信息，请参见 *archiver* 和 *archiver.cmd* 手册页。

卷序列号 (Volume Serial Number, VSN) 池指令

archiver.cmd 文件的 VSN 池部分定义了可在卷序列号 (Volume Serial Number, VSN) 关联指令中指定为一个单位的已命名归档介质卷集合。

此部分以 *vsnpools* 指令开始，以 *endvsnpools* 指令或 *archiver.cmd* 文件的结尾结束。定义 VSN 池的语法如下：

```
vsn-pool-name media-type volume-specification
```

其中：

- *vsn-pool-name* 是您分配给池的名称。
- *media-type* 是附录 A, [设备类型词汇表](#) 和 *mcf* 手册页中列出的双字符 Oracle HSM 介质类型标识符之一。
- *volume-specification* 是一个以空格分隔的列表，其中含有一个或多个用于匹配卷序列号的正则表达式。有关正则表达式语法的详细信息，请参见 *Solaris regcmp* 手册页。

以下示例定义了四个 VSN 池：*users_pool*、*data_pool*、*proj_pool* 和 *scratch_pool*。临时池是指当 VSN 关联中的特定卷或另一个 VSN 池消耗殆尽时，系统临时使用的一组卷。如果这三个特定池中有一个池的卷已消耗殆尽，归档程序将选择临时池 VSN。

```
vsnpools
users_pool li ^VOL2[0-9][0-9]
data_pool li ^VOL3.*
scratch_pool li ^VOL4[0-9][0-9]
proj_pool li ^VOL[56].*
```

```
endvsnpools
```

卷序列号 (Volume Serial Number, VSN) 关联指令

archiver.cmd 文件的 VSN 关联部分用于为归档集分配归档介质卷。此部分以 *vsns* 指令开始，以 *endvsns* 指令结束。

卷分配指令采用以下格式：

```
archive-set-name.copy-number [media-type volume-specification] [-pool vsn-pool-name]
```

其中：

- *archive-set-name* 是归档集分配指令分配给要与指定卷关联的归档集的名称。
- *copy-number* 是归档副本指令分配给要与指定卷关联的副本的编号。它为 [1-4] 范围内的一个整数。
- *media-type* 是附录 A, [设备类型词汇表](#)和 *mcf* 手册页中列出的双字符 Oracle HSM 介质类型标识符之一。
- *volume-specification* 是一个以空格分隔的列表，其中含有一个或多个用于匹配卷序列号的正则表达式。有关正则表达式语法的详细信息，请参见 Solaris *regcmp* 手册页。
- *-pool vsn-pool-name* 是一个先前指定的已命名归档介质卷集合，该集合可作为一个单位进行指定。请参见卷序列号 (Volume Serial Number, VSN) 池指令。

以下示例说明了将介质与两行 VSN 指定内容关联的各种方法。

```
vsns
archiveset.1 lt VSN001 VSN002 VSN003 VSN004 VSN005
archiveset.2 lt VSN0[6-9] VSN10
archiveset.3 -pool data_pool
endvsns
```

回写指令

回写是将文件数据从近线或离线存储设备复制回在线存储设备的过程。

回写程序在 *samd* 守护进程运行时启动。回写程序具有以下默认行为：

- 回写程序尝试使用库中的所有驱动器。
- 回写缓冲区的大小由介质类型决定，并且不锁定回写缓冲区。
- 不写日志文件。
- 一次最多可以激活 1000 个回写请求。

可以通过在 */etc/opt/SUNWsamfs/stager.cmd* 文件中插入指令为站点定制回写程序的操作。

当应用程序需要脱机文件时，系统会将其归档副本回写到磁盘高速缓存，除非该文件归档时使用了 `-n`（从不回写）选项。要使文件立即可供应用程序使用，读取操作在回写操作之后立即进行跟踪，这样在整个文件回写完毕之前就可以开始进行访问。

回写错误包括介质错误、介质不可用、自动化库不可用以及其他错误。返回回写错误时，如果存在文件副本并有用来读取归档副本的介质的设备，则 Oracle HSM 软件将尝试找到下一个可用的文件副本。

stager.cmd 文件

在 `stager.cmd` 文件中，指定要覆盖默认行为的指令。您可以对回写程序进行配置，以便立即回写文件、从不回写文件、回写部分文件以及指定其他回写操作。例如，指定 `never-stage` 属性对访问大型文件中小型记录的应用程序有益，因为不必联机回写文件即可从归档介质直接访问数据。

本节将介绍回写程序指令。有关回写程序指令的其他信息，请参见 `stager.cmd` 手册页。如果您使用的是 Oracle HSM Manager 软件，则可以通过 "File System Summary"（文件系统摘要）页或 "File System Details"（文件系统详细信息）页来控制回写。您可以浏览文件系统并查看各个文件的状态，可以使用过滤器查看某些文件并选择要回写的特定文件。可以选择从哪个副本回写，也可以让系统选择该副本。

以下示例显示了设置所有可能指令后的 `stager.cmd` 文件。

```
drives=dog 1
bufsize=od 8 lock
logfile=/var/adm/stage.log
maxactive=500
```

drives: 指定进行回写的驱动器数量

默认情况下，回写程序在回写文件时使用所有可用的驱动器。如果回写程序使所有驱动器处于繁忙状态，则会影响归档程序的活动。`drives` 指令用于指定回写程序可用的驱动器数量。此指令的格式如下：

```
drives=library count
```

其中：

- `library` 是磁带库的系列集名称（与其在 `mcf` 文件中显示的名称一致）。
- `count` 是使用的最大驱动器数。默认情况下，此数量与在 `mcf` 文件中为该库配置的驱动器数量相同。

以下示例指定 `dog` 系列集的库中只有一个驱动器用于回写文件：

```
drives = dog 1
```

bufsize: 设置回写缓冲区大小

默认情况下，要回写的文件先被读取到缓冲区的内存中，然后再从归档介质恢复到磁盘高速缓存。使用 `bufsize` 指令可指定缓冲区大小，还可以选择锁定缓冲区。这些操作可以改善性能。您可以尝试各种 `number-blocks` 值。该指令的格式如下：

```
bufsize= media-type number-blocks [lock]
```

其中：

- `media-type` 是[附录 A, 设备类型词汇表](#)和 `mcf` 手册页中列出的双字符 Oracle HSM 介质类型标识符之一。
- `number-blocks` 为 [2-8192] 范围内的一个整数。此值将与 `defaults.conf` 文件中指定的 `media-type_blksize` 值相乘。为 `number-blocks` 指定的数值越大，所使用的内存就越多。默认值为 16。
- `lock` 在每个回写操作期间强制使用锁定的缓冲区。这可以避免由于为每个 I/O 请求锁定和取消锁定回写缓冲区而产生的开销，并且可以提高性能。如果在可用内存有限的系统上指定 `lock` 参数，则可能导致内存用尽。默认情况下，不强制使用锁定的缓冲区，而由文件系统保留对归档缓冲区的控制权。

只有已为回写的文件启用直接 I/O 时，`lock` 参数才有效。有关启用直接 I/O 的更多信息，请参见 `setfa`、`sam_setfa` 和 `mount_samfs` 手册页。

logfile: 指定回写日志文件

可以请求 Oracle HSM 软件收集文件回写事件信息，并将此信息写入日志文件中。默认情况下，不写入任何日志文件。`logfile` 指令用于指定回写程序可在其中写入记录信息的日志文件。回写程序在日志文件中写入一行或多行有关每个已回写文件的信息。该行中包括文件名、回写日期和时间以及卷序列号 (volume serial number, VSN) 等信息。该指令的格式如下：

```
logfile=filename [event-list]
```

其中 `filename` 是日志文件的全路径名，`event-list` 是要记录的事件类型的列表（以空格分隔）：

- `all` 记录所有回写事件。
- `start` 记录文件开始回写的时间。
- `finish`（默认值）记录文件回写结束时间。
- `cancel`（默认值）记录操作员取消回写操作的时间。
- `error`（默认值）记录回写错误。

以下指令会在 `/var/adm/` 目录中创建一条回写日志：

```
logfile=/var/adm/stage.log
```


回写程序日志条目采用以下格式：

```
status date time media-
type volume position.offset inode filesize filename copy user group requestor equipment-
number validation
```

其中：

- *status* 为：*S*（开始）、*C*（已取消）、*E*（出错）、*F*（已完成）。
- *date* 是采用以下格式的日期：*yyyy/mm/dd*，其中 *yyyy* 是表示年份的四位数字，*mm* 是表示月份的两位数字，*dd* 是表示一月中某一天的两位数字。
- *time* 是采用以下格式的时间：*hh:mm:ss*，其中 *hh*、*mm* 和 *ss* 分别是表示小时、分钟和秒的两位数字。
- *media-type* 是附录 A, [设备类型词汇表](#)和 *mcf* 手册页中列出的双字符 Oracle HSM 介质类型标识符之一。
- *volume* 是存放要回写的文件的介质的卷序列号 (volume serial number, VSN)。
- *position.offset* 是一对以点分隔的十六进制数字，表示归档 (*tar*) 文件在卷中的起始位置以及回写文件相对于归档文件起始位置的偏移量。
- *inode* 是回写文件的 inode 编号和生成号，以点分隔。
- *filesize* 是回写文件的大小。
- *filename* 是已回写文件的名称。
- *copy* 是包含回写文件的副本的归档副本编号。
- *user* 是拥有该文件的用户。
- *group* 是拥有该文件的组。
- *requestor* 是请求该文件的组。
- *equipment-number* 是 *mcf* 文件中为从其回写文件的驱动器定义的设备序号。
- *validation* 指明是要验证 (*v*) 回写文件还是不验证 (*-*)。

以下示例显示了一个典型回写程序日志的一部分：

```
S 2014/02/16 14:06:27 dk disk01 e.76d 2557.1759 1743132 /sam1/testdir0/filebu 1 root
other root 0 -
F 2014/02/16 14:06:27 dk disk01 e.76d 2557.1759 1743132 /sam1/testdir0/filebu 1 root
other root 0 -
S 2014/02/16 14:06:27 dk disk02 4.a68 1218.1387 519464 /sam1/testdir1/fileaq 1 root
other root 0 -
S 2014/02/16 14:06:43 dk disk01 13.ba5 3179.41 750880 /sam1/testdir0/filecl 1 root
other root 0 -
F 2014/02/16 14:06:43 dk disk01 13.ba5 3179.41 750880 /sam1/testdir0/filecl 1 root
other root 0 -
```

maxactive：指定回写请求的数量

可以使用 *maxactive* 指令指定一次可激活的回写请求的数量。该指令的格式如下：

```
maxactive=number
```

其中 *number* 为 [1-500000] 范围内的一个整数。默认值为 4000。

以下示例指定可以同时存在于队列中的回写请求数量不超过 500 个：

```
maxactive=500
```

copysel：指定回写期间的副本选择顺序。

回写指令 *copysel* 针对每个文件系统设置回写程序副本选择顺序。

```
copysel=selection-order
```

其中 *selection-order* 是以冒号分隔的副本编号列表（按先后顺序）。默认选择顺序为 1:2:3:4。

有关更多信息，请参见 *stager.cmd* 手册页。以下示例显示的 *stager.cmd* 文件为文件系统 *samfs1* 和 *samfs2* 设置了非默认的副本选择顺序：

```
logfile = /var/opt/SUNWsamfs/log/stager
drives = hp30 1
fs = samfs1
copysel = 4:3:2:1
fs = samfs2
copysel = 3:1:4:2
```

预览请求指令

当一个 Oracle HSM 进程请求一个当前未装入驱动器的可移除介质卷时，系统会将该请求添加到预览队列。默认情况下，系统按先进先出 (First In First Out, FIFO) 的顺序执行已排队的请求。不过，您可以通过编辑文件 */etc/opt/SUNWsamfs/preview.cmd* 来覆盖默认行为。Oracle HSM 磁带库控制守护进程 (*sam-aml*d) 在开始使用这些指令时会对其进行读取，直至该守护进程停止。不能动态更改队列优先级。

有两种类型的指令：

- 全局指令位于文件顶部，适用于所有文件系统。
- 文件系统指令采用 *fs=directive* 格式并特定于各个文件系统

以下各节介绍了如何编辑 *preview.cmd* 文件，以控制预览队列：

- [全局指令](#)
- [全局和/或特定于文件系统的指令](#)
- [preview.cmd 文件样例](#)

全局指令

以下全部是全局指令：

- **vsn_priority**: 调整卷优先级
- **age_priority**: 针对在队列中的等待时间调整优先级

vsn_priority: 调整卷优先级

vsn_priority 指令可提高卷 (VSN) 的优先级, 通过指定的值将卷标记为高优先级卷。指令采用以下形式:

```
vsn_priority=value
```

其中 *value* 是一个实数。默认值为 *1000.0*。

您可以使用以下命令为卷设置高优先级标志:

```
chmed +p media-type.volume-serial-number
```

其中 *media-type* 是附录 A, 设备类型词汇表和 *mcf* 手册页中列出的双字符 Oracle HSM 介质类型之一, 其中 *volume-serial-number* 是字母数字字符串, 用于唯一标识磁带库中的高优先级卷。有关完整信息, 请参见 *chmed* 手册页。

age_priority: 针对在队列中的等待时间调整优先级

age_priority 指令可根据请求在队列中所用时间更改指定的相对优先级, 例如, 为防止较旧的请求不停地被优先级较高的新请求抢先。该指令指定一个乘数来更改在队列中所用时间的相对权重。它采用以下形式:

```
age_priority=weighting-factor
```

其中 *weighting-factor* 是一个大于、小于或等于 *1.0* 的实数, 其中:

- 如果值大于 *1.0*, 则在计算聚合优先级时会增加为在队列中所用时间指定的权重。
- 如果值小于 *1.0*, 则在计算总优先级时会减少为在队列中所用时间指定的权重。
- 如果值等于 *1.0*, 则不会更改为在队列中所用时间指定的相对权重。

默认值为 *1.0*。

全局和/或特定于文件系统的指令

以下指令既可全局应用, 也可以按每个文件系统应用:

- **hwm_priority**: 在磁盘高速缓存几乎填满时调整优先级
- **lwm_priority**: 在磁盘高速缓存几乎为空时调整优先级
- **lhwm_priority**: 根据磁盘高速缓存的填充情况调整优先级
- **hlwm_priority**: 根据磁盘高速缓存为空的程度调整优先级

hwm_priority: 在磁盘高速缓存几乎填满时调整优先级

hwm_priority 指令可在文件系统利用率超出上限 (*hwm*, 释放程序进程启动并开始回收在归档介质中存在副本的文件所占用的磁盘空间的点) 时调整为归档请求指定的相对于回写请求的相对权重。在这种情况下, 通过增大为归档指定的相对权重, 释放进程可释放更多的空间以用于回写归档副本和新文件。指令采用以下形式:

```
hwm_priority=weighting-factor
```

其中 *weighting-factor* 是一个实数。默认值为 0.0 。

lwm_priority: 在磁盘高速缓存几乎为空时调整优先级

lwm_priority 指令可在文件系统利用率降至低于下限 (*lwm*, 释放程序进程停止的点) 时调整为归档请求指定的相对于回写请求的相对权重。在这种情况下, 通过减小为归档指定的相对权重 (从而增大回写请求的优先级), 可在磁盘高速缓存中放置更多的文件、降低对介质挂载的需求并提高文件系统性能。指令采用以下形式:

```
lwm_priority=weighting-factor
```

其中 *weighting-factor* 是一个实数。默认值为 0.0 。

lhwm_priority: 根据磁盘高速缓存的填充情况调整优先级

lhwm_priority 指令在磁盘高速缓存填满且高速缓存利用率介于下限 (*lwm*) 和上限 (*hwm*) 之间时, 调整为归档请求指定的相对于回写请求的相对权重。在这种情况下, 通过增大为归档指定的相对权重, 释放进程可释放更多的空间以用于回写归档副本和新文件。指令采用以下形式:

```
lhwm_priority=weighting-factor
```

其中 *weighting-factor* 是一个实数。默认值为 0.0 。

hlwm_priority: 根据磁盘高速缓存为空的程度调整优先级

hlwm_priority 指令在磁盘高速缓存将要为空且高速缓存利用率介于上限 (*hwm*) 和下限 (*lwm*) 之间时, 调整为归档请求指定的相对于回写请求的相对权重。在这种情况下, 通过减小为归档指定的相对权重 (从而增大回写请求的优先级), 可在磁盘高速缓存中放置更多的文件、降低对介质挂载的需求并提高文件系统性能。指令采用以下形式:

```
hlwm_priority=weighting-factor
```

其中 *weighting-factor* 是一个实数。默认值为 0.0 。

preview.cmd 文件样例

任何给定介质挂载请求的聚合优先级均根据以下公式使用由所有权重因子设置的值进行确定：

$$priority = vsn_priority + wm_priority + (age_priority * time-waiting-in-queue)$$

其中 *wm_priority* 是当前生效的水位标志优先级 (*hwm_priority*、*lwm_priority*、*hlwm_priority* 或 *lhwm_priority*)，*time-waiting-in-queue* 是卷请求已排队等待的秒数。有关优先级计算的详尽说明，请参见 *preview.cmd* 手册页的 *PRIORITY CALCULATION* (优先级计算) 部分。

在特殊情况下 (对数据的访问至关重要时或可移除介质驱动器供不应求时)，通过 *preview.cmd* 文件中的指令，可更好地将文件系统活动与操作要求和可用资源进行匹配。*preview.cmd* 文件中的设置不影响所存储数据的完整性，因此，您可以自由地尝试，直到找到归档请求与回写请求之间的合适平衡。

基于以下任一或全部原因，您可能需要调整默认的优先级计算：

- 确保先处理回写请求再处理归档请求，以便使文件在用户和应用程序访问时可用。
- 确保归档请求在文件系统将要充满时获得最高优先级

下面的 *preview.cmd* 文件样例用于处理上面重点介绍的情况：

```
# Use default weighting value for vsn_priority:
vsn_priority=1000.0
age_priority = 1.0
# Insure that staging requests are processed before archive requests:
lwm_priority = -200.0
lhwm_priority = -200.0
hlwm_priority = -200.0
# Insure that archive requests gain top priority when a file system is about to fill
up:
hwm_priority = 500.0
```

将 *lwm_priority*、*lhwm_priority* 和 *hlwm_priority* 的权重值设为负值可确保只要磁盘高速缓存中有可用空间，回写请求就优先于归档请求，这样，在请求数据时便始终可进行访问。如果队列中已经有多个请求的等待时间达到了 100 秒，且文件系统低于下限，则：

- 优先级卷的归档挂载请求的聚合优先级为 $1000 + (-200) + (1 \times 100) = 900$
- 优先级卷的回写挂载请求的聚合优先级为 $1000 + 0 + (1 \times 100) = 1100$
- 非优先卷的回写挂载请求的聚合优先级为 $0 + 0 + (1 \times 100) = 100$

但是，当磁盘高速缓存接近填满时，归档请求必须优先。如果在文件系统即将填满时归档的文件过少，则没有空间可用于回写已归档的文件或容纳新文件。如果队列中已经有多个请求的等待时间达到了 100 秒，且文件系统高于上限，则：

- 优先级卷的归档挂载请求的聚合优先级为 $1000+500+(1\times100)=1600$
- 优先级卷的回写挂载请求的聚合优先级为 $1000+0+(1\times100)=1100$
- 非优先卷的回写挂载请求的聚合优先级为 $0+0+(1\times100)=100$

附录 E

附录 E. 产品辅助功能

弱视、失明、色盲或其他视力障碍的用户可通过命令行界面访问 Oracle Hierarchical Storage Manager and StorageTek QFS Software (Oracle HSM)。此基于文本的界面与屏幕阅读器兼容，且可以使用键盘控制所有功能。

词汇表

该词汇表重点描述特定于 Oracle HSM 软件和文件系统的术语。有关行业标准定义，请参阅由全球网络存储工业协会 (Storage Networking Industry Association, SNIA) 维护的词典，网址为：<http://www.snia.org/education/dictionary/>。

addressable storage (可寻址存储)	包括在线、近线、异地和离线存储在的存储空间，用户可通过 Oracle HSM 文件系统进行引用。
admin set ID (管理集 ID)	由存储管理员定义、拥有共同特征的用户和/或组的集合。创建管理集的目的通常在于，对涉及来自多个组的用户以及跨多个文件和目录的项目存储进行管理。
archival media (归档介质)	写入归档文件的目标介质。归档介质同时包括可移除磁带或磁光盘卡盒，以及配置用于归档的磁盘文件系统。
archival storage (归档存储)	在归档介质上创建的数据存储空间。
archive set (归档集)	归档集标识要归档的一组文件，这些文件在大小、所有权、组或目录位置方面满足相同的条件。归档集可以跨任何文件系统组进行定义。
archiver (归档程序)	一种可以自动将文件复制到可移除磁带的归档程序。
associative staging (关联回写)	当对组的任一成员进行回写时回写一组相关文件。当多个文件位于同一目录并经常一起使用时，文件所有者可通过设置 Oracle HSM 关联回写文件属性来关联这些文件。这样，如果在组中的一个文件被某个应用程序访问时其中的任何文件脱机，则 Oracle HSM 会将整个组从归档介质回写到磁盘高速缓存中。这样可确保所有必要的文件同时重新可用。
audit (full) (完全审计)	装入卡盒以验证其 VSN 的过程。对于磁光盘卡盒，会确定容量和空间信息，并将这些信息输入到自动化库的目录中。请参见 volume serial number (VSN) (卷序列号)。
automated library (自动化库)	一款自动控制的设备，用于在无操作员参与的情况下，自动载入和卸载可移除介质卡盒。自动化库包含两个部分：一个或多个驱动器，以及将卡盒移入和移出存储插槽和驱动器的传输装置。
backup (备份)	用于防止意外丢失的文件集合的快照。一个备份同时包括文件的属性和关联数据。
block allocation map (块分配图)	一个显示磁盘中所有可用存储块的位图，它指出了每个块的状态为在使用中还是未使用。
block size (块大小)	块设备（如硬盘或磁带卡盒）上最小的可寻址数据单位的大小。在磁盘设备上，这相当于扇区大小，通常为 512 字节。

cartridge (卡盒)	一种数据存储介质容器，如磁带或光学介质。有时也称为 volume (卷) 、磁带或介质。请参见 volume serial number (VSN) (卷序列号) 。
catalog (目录)	自动化库中可移除介质卷的记录。每个自动化库均有一个目录，且一个站点拥有一个有关所有自动化库的历史记录。使用 volume serial number (VSN) (卷序列号) 对卷进行标识和跟踪。
client-server (客户机/服务器)	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响应。发送请求的程序称为客户机。提供响应的程序称为服务器。
connection (连接)	两个协议模块之间的通道，用于提供可靠的流传输服务。TCP 连接可从一台计算机上的 TCP 模块扩展至另一台计算机上的 TCP 模块。
data device (数据设备)	文件系统中，用于存储文件数据的设备或设备组。
DAU	请参见 disk allocation unit (DAU) (磁盘分配单元) 。
device logging (设备日志记录)	一种可配置功能，该功能可提供支持 Oracle HSM 文件系统的硬件设备的具体错误信息。
device scanner (设备扫描程序)	一种软件，该软件可定期监视所有手动挂载的可移除设备是否存在，并检测可由用户或其他进程请求的已挂载卡盒是否存在。
direct access (直接访问)	一种文件属性（永远不必回写），表示可直接从归档介质中访问近线文件，而不需要将近线文件检索到磁盘高速缓存。
direct attached library (直接连接式库)	一种自动化库，该库使用 SCSI 接口直接连接至服务器。SCSI 连接库直接受 Oracle HSM 软件控制。
direct I/O (直接 I/O)	用于大型块对齐的连续 I/O 的属性。 <i>setfa</i> 命令的 <i>-D</i> 选项为直接 I/O 选项。它可为文件或目录设置直接 I/O 属性。如果将直接 I/O 属性应用于目录，则可继承直接 I/O 属性。
directory (目录)	文件数据结构，该结构指向文件系统内的其他文件和目录。
disk allocation unit (DAU) (磁盘分配单元)	<p>在 Oracle HSM 文件系统中，每个 I/O 操作所占据的最少量连续空间（无论写入的数据量是多少）。因而磁盘分配单元可确定当传输给定大小的文件时所需的 I/O 操作最少数量。它应该是磁盘设备的 block size (块大小) 的倍数。</p> <p>磁盘分配单元会有所不同，具体取决于选定的 Oracle HSM 设备类型和用户要求。<i>md</i> 设备类型使用双分配单元：前八次文件写入的 DAU 是 4 KB，后续写入是用户指定的值 16、32 或 64 KB，因此小文件将以合适的小块写入，较大的文件以较大的块写入。<i>mr</i> 和 striped group (分散读写组) 设备类型使用可调整的 DAU，范围在 [8-65528] KB 内，增</p>

	量为 8 KB。因而文件中写入了大小统一的大块，这些大块的大小与统一的大文件的大小十分近似。
disk buffer (磁盘缓冲区)	在 SAM-Remote 配置中，用于将数据从客户机归档到服务器的服务器系统上的缓冲区。
disk cache (磁盘高速缓存)	文件系统软件的磁盘驻留部分，用于在联机磁盘高速缓存和归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可用作磁盘高速缓存。
disk space threshold (磁盘空间阈值)	由管理员定义的磁盘高速缓存利用率的最高或最低级别。释放程序基于这些预定义的磁盘空间阈值来控制磁盘高速缓存利用率。
disk striping (磁盘分散读写)	跨几个磁盘记录一个文件的过程，从而可改善访问性能并提高总体存储容量。另请参见 striping (分散读写) 。
drive (磁带机)	一种从可移除介质卷中传输数据或向可移除介质卷中传输数据的机制。
Ethernet (以太网)	一种包交换局域网技术。
extent array (范围阵列)	文件的 inode 内的阵列，用于定义分配给文件的每个数据块在磁盘上的位置。
family device set (设备系列集)	请参见 family set (系列集) 。
family set (系列集)	独立物理设备的逻辑分组，如自动化库中的磁盘或驱动器集合。另请参见 storage family set (存储系列集) 。
FDDI	fiber-distributed data interface (光纤分布式数据接口) 的缩写，为局域网中数据传输标准，最高可扩展至 200 公里 (124 英里)。FDDI 协议基于令牌环协议。
Fibre Channel (光纤通道)	由 ANSI 提出的标准，该标准规定在不同设备之间实行高速串行通信。光纤通道是在 SCSI-3 中使用的其中一个总线体系结构。
file system (文件系统)	一种由文件和目录组成的分层集合。
file-system-specific directives (特定于文件系统的指令)	归档程序和释放程序指令 (位于 <i>archiver.cmd</i> 文件中的全局指令之后)，这些指令专用于特定文件系统，以 <i>fs =</i> 开头。特定于文件系统的指令的应用范围到出现下一个 <i>fs =</i> 指令行或文件末尾结束。如果多个指令影响到一个文件系统，则特定于文件系统的指令会覆盖全局指令。
ftp	File Transfer Protocol (文件传输协议) 的缩写，用于在两台主机之间传输文件的网络协议。有关更安全的备选方法，请参见 sftp 。

global directives (全局指令)	归档程序和释放程序指令，这些指令应用于所有文件系统，位于首个 <code>fs=</code> 行之前。
grace period (宽限期)	对于 quota (配额) 而言，是指文件系统允许属于指定用户、组和/或 admin set ID (管理集 ID) 的文件大小总量超出配额中指定的 soft limit (软限制) 的时间期限。
hard limit (硬限制)	对于 quota (配额) 而言，是指定用户、组和/或 admin set ID (管理集 ID) 可使用的存储资源的绝对最大数量。请参见 soft limit (软限制)。
high-water mark (高水位线)	<ol style="list-style-type: none"> 1. 归档文件系统中磁盘高速缓存利用率百分比，达到此百分比时，Oracle HSM 文件系统启动释放程序进程，并从磁盘中删除以前归档的文件。正确配置的高水位标志可确保文件系统始终拥有足够的可用空间用于新文件和新回写的文件。有关更多信息，请参见 <i>sam-releaser</i> 和 <i>mount_samfs</i> 手册页。与 low-water mark (低水位标志) 相对。 2. 在包含在归档文件系统中的可移除介质库中，可启动回收程序流程的介质高速缓存利用率百分比。回收可部分清空当前数据的整卷，以便它们可由新介质进行替换或对它们重新设置标签。
historian (历史记录)	Oracle HSM 历史记录是卷的目录，已将这些卷从在 <code>/etc/opt/SUNWsamfs/mcf</code> 文件中定义的自动化介质库中导出。默认情况下，历史记录位于 <code>/var/opt/SUNWsamfs/catalog/historian</code> 中的 Oracle HSM 文件系统主机上。有关详细信息，请参见 Oracle HSM 历史记录手册页。
hosts file (hosts 文件)	hosts 文件包含共享文件系统中所有主机的列表。如果您要将某个文件系统初始化为 Oracle HSM 共享文件系统，则必须在创建该文件系统之前创建 hosts 文件 <code>/etc/opt/SUNWsamfs/hosts.fs-name</code> 。 <i>sammkfs</i> 命令会在创建文件系统时使用该 hosts 文件。以后，您可以使用 <i>samsharefs</i> 命令替换或更新 hosts 文件的内容。
indirect block (间接块)	包含存储块列表的磁盘块。文件系统最多可拥有三级间接块。第一级间接块包含用于数据存储的块列表。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。
inode file (inode 文件)	文件系统上的特殊文件 (<i>.inodes</i>)，其中包含驻留在文件系统的所有文件的 inode 结构。inode 的长度为 512 字节。inode 文件是一个元数据文件，它不同于文件系统中的其它文件数据。
inode (索引节点)	index node (索引节点) 的缩写。文件系统用来描述文件的数据结构。inode 描述与文件关联的所有属性 (名称属性除外)。这些属性包括所有权、访问权、权限、大小和文件在磁盘系统中的位置。
kernel (内核)	提供基本操作系统工具的程序。UNIX 内核可以创建和管理进程、提供访问文件系统的功能、提供基本安全性能，以及提供通信工具。

LAN	local area network (局域网) 的缩写。
lease (租约)	一种功能, 用于向客户机主机授予权限, 使其可在指定期限内对文件执行操作。元数据服务器向每一个客户机主机发放租约。根据需要, 可对租约进行续借以允许客户机主机继续进行文件操作。
library catalog (库目录)	请参见 catalog (目录) 。
library (磁带库)	请参见 automated library (自动化库) 。
local file system (本地文件系统)	安装在 Solaris Cluster 系统的一个节点上的文件系统, 该文件系统不太可用于其他节点。同时也指安装在服务器上的文件系统。
low-water mark (低水位标志)	归档文件系统中磁盘高速缓存利用率百分比, 达到此百分比时, Oracle HSM 文件系统停止释放程序进程, 并停止从磁盘中删除以前归档的文件。正确配置的下限可确保文件系统的高速缓存中尽可能保留文件, 以实现最佳性能, 同时为新文件和新回写的文件提供可用空间。有关更多信息, 请参见 <i>sam-releaser</i> 和 <i>mount_samfs</i> 手册页。与 high-water mark (高水位线) 相对。
LUN	Logical unit number (逻辑单元号)。
mcf	Master Configuration File (主配置文件)。在初始化时读取的文件, 定义文件系统环境中设备之间的关系 (拓扑)。
media recycling (介质回收)	回收或重新使用包含很少活动文件的归档介质的过程。
media (介质)	磁带或光盘磁带。
metadata device (元数据设备)	用于存储文件系统元数据的设备 (例如固态硬盘或镜像设备)。将文件数据和元数据存放在不同的设备上可以提高性能。在 <i>mcf</i> 文件中, 元数据设备被声明为 <i>ma</i> 文件系统中的 <i>mm</i> 设备。
metadata (元数据)	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。它由以下各项的相关信息组成: 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。
mirror writing (镜像写入)	在互不相连的磁盘组中保存两份文件副本的过程, 用于防止因单个磁盘故障而导致数据丢失。
mount point (挂载点)	挂载文件系统的目录。
multireader file system (多读取器文件系统)	一种单写入器、多读取器功能, 该功能使您能够指定可在多个主机上挂载的文件系统。多个主机可以读取该文件系统, 但只有一个主机可以写入该文件系统。使用 <i>mount</i> 命令中的 <i>-o reader</i> 选项指定多个读取

	器。使用 <code>mount</code> 命令中的 <code>-o writer</code> 选项指定单写入器主机。有关更多信息，请参见 <code>mount_samfs</code> 手册页。
name space (名称空间)	文件集合的元数据部分，用于标识文件、文件属性及其存储位置。
nearline storage (近线存储)	一种可移除介质存储，在访问该存储前，需要启用自动挂载功能。近线存储通常要比在线存储的成本低，但访问时间稍长。
network attached automated library (网络连接自动化库)	一种库（例如，来自 StorageTek、ADIC/Grau、IBM 或 Sony 的库），该库使用由供应商提供的软件包进行控制。QFS 文件系统使用专门针对自动化库设计的 Oracle HSM 介质转换器守护进程与供应商软件进行交互。
NFS	network file system (网络文件系统) 的缩写，该文件系统可对异构网络上远程文件系统进行透明访问。
NIS	Network Information Service (网络信息服务) 的缩写，一种分布式网络数据库，包含有关网络上系统和用户的关键信息。NIS 数据库存储于主服务器和所有从属服务器中。
offline storage (脱机存储)	需要操作员参与装载的存储。
offsite storage (异地存储)	远离服务器并用于灾难恢复的存储。
online storage (在线存储)	直接可用的存储，例如磁盘高速缓存存储。
Oracle HSM	<ol style="list-style-type: none"> 1. Oracle Hierarchical Storage Manager 的常用缩写。 2. 形容词，用于描述为归档配置并由 Oracle HSM 软件管理的 QFS 文件系统。
partition (分区)	设备的一部分或者磁光磁带的一面。
preallocation (预分配)	在磁盘高速缓存上预留连续的空间量用于写入文件的过程。只能为大小为零的文件指定预分配。有关更多信息，请参见 <code>setfa</code> 手册页。
pseudo device (伪设备)	未关联任何硬件的软件子系统或驱动程序。
QFS	Oracle HSM QFS 软件产品，是一种高性能、大容量 UNIX 文件系统，可独立使用，也可用作由 Oracle Hierarchical Storage Manager 控制的归档文件系统。
qfsdump	请参见 samfsdump (qfsdump) 。
qfsrestore	请参见 samfsrestore (qfsrestore) 。

quota (配额)	允许指定用户、组或 admin set ID (管理集 ID) 使用的存储资源量。请参见 hard limit (硬限制) 和 soft limit (软限制)。
RAID	redundant array of independent disks (独立磁盘的冗余阵列) 的缩写。一种使用若干独立磁盘来可靠存储文件的磁盘技术。它可以防止单个磁盘出现故障时发生数据丢失, 可提供容错磁盘环境, 以及提供比单个磁盘更高的吞吐量。
recovery point (恢复点)	一种压缩文件, 可存储 Oracle HSM 文件系统元数据的即时备份副本。 一旦发生数据丢失 (包括从意外删除用户文件到灾难性丢失整个文件系统在内的任何数据丢失情况), 通过以下方法, 管理员几乎可以立即恢复至文件或文件系统的上个已知良好状态: 找到文件或文件系统未受影响的上一个恢复点。然后, 管理员可恢复在该时间点记录的元数据, 并将元数据中指示的文件从归档介质回写到磁盘高速缓存中, 或者 (更适合) 让文件系统按需回写文件, 以供用户和应用程序访问这些文件。
recycler (回收程序)	一种 Oracle HSM 实用程序, 用于回收由过期归档副本占用的卡盒空间。
regular expression (正则表达式)	标准化模式匹配语言中的一个字符串, 设计用于搜索、选择和编辑其他字符串, 如文件名和配置文件。有关用于 Oracle HSM 文件系统操作的正则表达式语法的完整详细信息, 请参见 Oracle HSM Solaris <i>regex</i> 和 <i>regcmp</i> 手册页。
release priority (释放优先级)	一种优先级方法, 文件系统中的文件在归档后按照该优先级进行释放。释放优先级的计算方法是通过各种加权数乘以相应的文件属性, 然后得出各个结果之和。
releaser (释放程序)	一个 Oracle HSM 组件, 用于识别已归档的文件并释放它们的磁盘高速缓存副本, 从而腾出更多可用的磁盘高速缓存空间。释放程序可以根据阈值上限和阈值下限自动调整联机磁盘存储量。
remote procedure call (远程过程调用)	请参见 RPC 。
removable media file (可移除介质文件)	一种特殊类型的用户文件, 可以直接从它所在的可移除介质卡盒 (如磁带或光盘卡盒) 中进行访问。也可用于写入归档和回写文件数据。
robot (机械手)	一个 automated library (自动化库) 组件, 用于在存储插槽和驱动器之间移动卡盒。也称为 transport (传输)。
round-robin (循环)	一种按顺序将全体文件写入到多个逻辑磁盘的数据访问方法。将单个文件写入磁盘时, 此文件的全部内容都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。另请参见 disk striping (磁盘分散读写) 和 striping (分散读写)。

RPC	Remote procedure call（远程过程调用）。NFS 用以实施定制网络数据服务器的底层数据交换机制。
SAM	Storage Archive Manager（Oracle Hierarchical Storage Manager 产品的旧名称）的常用缩写。
SAM-QFS	<ol style="list-style-type: none"> 1. Oracle Hierarchical Storage Manager 产品的旧版本的常用缩写。 2. 形容词，用于描述为归档配置并由 Oracle HSM 软件管理的 QFS 文件系统。
SAM-Remote client (SAM-Remote 客户机)	具有客户机守护进程的 Oracle HSM 系统，其中包含许多伪设备，并且还可以具有自己的库设备。客户机用来存储一个或多个归档副本的归档介质由 SAM-Remote 服务器决定。
SAM-Remote server (SAM-Remote 服务器)	SAM-Remote 服务器不仅是功能完备的 Oracle HSM 存储管理服务器，而且还是用于定义各个 SAM-Remote 客户机所共享库的 SAM-Remote 服务器守护进程。
samfsdump (qfsdump)	一个程序，用于为给定的文件组创建控制结构转储并复制所有控制结构信息。它通常不复制文件数据。使用 <i>-U</i> 选项时，该命令还会复制数据文件。如果未安装 Oracle Hierarchical Storage Manager 软件包，则该命令称为 <i>qfsdump</i> 。
samfsrestore (qfsrestore)	一个程序，用于从控制结构转储中恢复 inode 和目录信息。另请参见 samfsdump (qfsdump) 。
SAN	Storage Area Network（存储区域网络）。
SCSI	Small Computer System Interface（小型计算机系统接口），一个通常用于外围设备（例如磁盘和磁带机以及自动化库）的电子通信规范。
seeking（寻道）	在随机访问 I/O 操作过程中，将磁盘设备的读/写头从一个磁盘位置移动到另一个磁盘位置。
sftp	Secure File Transfer Protocol（安全文件传输协议）的缩写，一种基于 ftp 的 ssh 安全实现。
shared hosts file（共享 hosts 文件）	当您创建共享文件系统时，系统会将信息从 hosts 文件复制到元数据服务器上的共享 hosts 文件中。当您发出 samsharefs -u 命令时更新此信息
Small Computer System Interface（小型计算机系统接口）	请参见 SCSI 。
soft limit（软限制）	对于 quota（配额） 而言，是指所指定用户、组和/或 admin set ID（管理集 ID） 可无限期填充的最大存储空间量。文件可使用超出软

	限制所允许的空间量，最多可达到硬限制，但是仅限于配额中定义的短 grace period (宽限期)。请参见 hard limit (硬限制)。
ssh	Secure Shell (安全 Shell) 的缩写，一种可实现安全、远程命令行登录和命令执行的加密网络协议。
staging (回写)	将近线或离线文件从归档存储复制回在线存储的过程。
Storage Archive Manager	Oracle Hierarchical Storage Manager 产品的旧名称。
storage family set (存储系列集)	统一表示为单个逻辑设备的磁盘集。
storage slots (存储插槽)	当磁带未在驱动器中使用时在自动化库中的存储位置。
stripe size (分散读写大小)	移至分散读写的下一个设备之前要分配的磁盘分配单元 (disk allocation unit, DAU) 的数量。如果使用 <i>stripe=0</i> 挂载选项，则文件系统将采用循环访问方式，而不采用分散读写访问方式。
striped group (分散读写组)	文件系统上的设备集合；在 <i>mcf</i> 文件中，它被定义成一个或多个 <i>gXXX</i> 设备。分散读写组作为一个逻辑设备使用，并且始终分散读写成大小等于磁盘分配单元 (disk allocation unit, DAU) 的空间。
striping (分散读写)	一种以交错方式将文件同时写入到多个逻辑磁盘的数据访问方法。Oracle HSM 文件系统提供两种类型的分散读写：“硬分散读写”，该类型使用分散读写组；以及“软分散读写”，该类型使用 <i>stripe=x</i> 挂载参数。硬分散读写在设置文件系统时启用，需要在 <i>mcf</i> 文件内定义分散读写组。软分散读写通过 <i>stripe=x</i> 挂载参数启用，可针对文件系统或单个文件进行更改。它是通过设置 <i>stripe=0</i> 来禁用的。如果文件系统由包含相同数量元素的多个分散读写组构成，则可同时使用硬分散读写和软分散读写。另请参见 round-robin (循环)。
SUNW.qfs	支持 Oracle HSM 共享文件系统的一种 Solaris Cluster 资源类型。 <i>SUNW.qfs</i> 资源类型可定义共享文件系统的元数据服务器 (metadata server, MDS) 的故障转移资源
superblock (超级块)	文件系统上的一种数据结构，可定义文件系统的基本参数。超级块写入到存储系列集中的所有分区，并可标识该集中分区的成员身份。
tar	tape archive (磁带归档) 的缩写。它是用来存储归档映像的标准文件和数据记录格式。
TCP/IP	Transmission Control Protocol/Internet Protocol (传输控制协议/Internet 协议) 的缩写。Internet 协议负责主机之间的寻址和路由以及包传送 (IP)，在各个应用点之间可靠地传递数据 (TCP)。

timer (计时器)	一种配额软件，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。
transport (传输)	请参见 robot (机械手) 。
vfstab file (vfstab 文件)	<i>vfstab</i> 文件包含文件系统的挂载选项。在命令行上指定的挂载选项会覆盖 <i>/etc/vfstab</i> 文件中指定的挂载选项，但是 <i>/etc/vfstab</i> 文件中指定的挂载选项会覆盖 <i>smfs.cmd</i> 文件中指定的挂载选项。
volume overflow (卷溢出)	一种允许系统在多个 volume (卷) 上存储单个文件的功能。对于使用大容量文件（超过单个卡盒的容量）的站点，卷溢出功能非常有用。
volume serial number (VSN) (卷序列号)	<ol style="list-style-type: none">1. 分配给磁带或磁盘存储卷的序列号。一个卷序列号最多可包含六个大写的字母数字字符，必须以字母开头，并且必须在给定上下文（例如磁带库或分区）中唯一标识卷。卷序列号写入到卷标签上。2. 不严格地讲，是指特定存储 volume (卷)，尤其是可移除介质 cartridge (卡盒)。
volume (卷)	<ol style="list-style-type: none">1. 存储介质中的单个可访问的逻辑存储区域，通常通过 volume serial number (VSN) (卷序列号) 和/或卷标签来寻址。存储磁盘和磁带卡盒可容纳一个或多个卷。为了便于使用，会将卷挂载在指定 mount point (挂载点) 中的某个文件系统上。2. 容纳单个逻辑卷的磁带 cartridge (卡盒)。3. 在随机访问磁盘设备上，被当做顺序访问且可移除介质卡盒（如磁带）而配置和使用的文件系统、目录或文件。
WORM	Write-Once-Read-Many（一次写入多次读取）。介质的一种存储类别，只能写入一次但可以多次读取。

索引

W

文档

可用性, 12
