Oracle® Hierarchical Storage Manager and StorageTek QFS Software

安装和配置指南

发行版 6.1

E56770-03

2016 年 3 月

ORACLE

Oracle® Hierarchical Storage Manager and StorageTek QFS Software
安装和配置指南

E56770-03

目录

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）安装和配置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。
- 粗体类型表示图形用户界面元素。
- 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* 安装和配置指南》是多卷 *Oracle HSM* 客户文档库的一部分，可以从 *docs.oracle.com/en/storage* 获取。

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

1

第 1 章 部署 Oracle HSM 解决方案

总的来说，Oracle Hierarchical Storage Manager and StorageTek QFS Software (Oracle HSM) 的部署是一个相当简单的过程。安装软件包，编辑几个配置文件，运行几条命令，然后挂载并使用新的文件系统。不过，Oracle HSM 还提供了各种选项和调整参数。这些额外的功能可让您解决几乎任何特殊需求。但不需要的功能也会使部署变得复杂并使生成的解决方案不尽人意。

因此，本文档旨在指导您完成严格遵守详细解决方案要求的 Oracle HSM 部署。首先介绍基本 QFS 和 Oracle HSM 文件系统的工作原理、安装和配置。这些文件系统将凭借自身满足您的所有要求，或形成更专业的解决方案的基础。在基础工作完成后，您可以转到附加功能的配置过程，以支持特定环境和专业的业务需求。您将执行以下核心任务：

- 配置硬件和操作系统软件以满足要求。
- 配置所需的基础 QFS 和/或 Oracle HSM 文件系统，尽可能接受默认设置。
- 配置需求中所要求的任何附加 Oracle HSM 功能。
- 备份您的最终配置并移交供测试和生产使用。

在整个计划和部署过程中，记住 QFS 和 Oracle HSM 的目的在于提供一个简单的 UNIX 文件系统接口，而将性能优化、数据保护和归档的复杂性隐藏起来。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。文件系统和存储资源过于复杂的实施和过多的微观管理都将危害 Oracle HSM 部署的关键目标，同时通常会损害性能、容量利用率和/或数据保护。

本简介的其余部分建议介绍了 QFS 文件系统和 Oracle HSM 归档文件系统。对此信息有个基本了解，会使后续配置步骤的作用理解起来更加轻松。

QFS 文件系统

QFS 文件系统使您能够将完全优化的定制存储解决方案与标准 UNIX 接口相结合。在内部，它们管理物理存储设备以满足严格且通常高度专业化的性能要求。但它们作为普通 UNIX 文件系统呈现给外部用户、应用程序和操作系统。因此，您可以使用复杂的存储硬件系列满足专业的性能和数据保护要求，并且仍能保证与现有应用程序和业务流程的直接集成。

QFS 文件系统使用不可分割的 QFS 卷管理器管理其自己的物理存储。QFS 软件将标准物理存储设备组织成高度优化且与标准接口完全兼容的逻辑设备。软件封装特定功能和定制设置，以便它们仍在操作系统和应用程序中隐藏。对于后者，QFS 软件通过标准的 Solaris 设备驱动程序呈现处理 I/O 请求的逻辑系列集设备（类似单一磁盘）。这种标准符合性和可协调性的结合将 QFS 与其他 UNIX 文件系统区分开来。

本节的其余部分从 QFS 默认值和 I/O 性能调整的简短讨论开始，然后描述了核心工具，可用于控制您创建的文件系统的 I/O 行为：

- 灵活的磁盘分配单元和逻辑设备类型使您能够将读取和写入大小与文件大小相匹配。
- 分散读写和循环文件分配方法使您能够控制文件 I/O 与设备交互的方式。
- 完全可配置的存储分配和集成卷管理使您能够控制文件系统与底层物理存储交互的方式。
- 一般用途和高性能文件系统的选择为您提供了在单独设备上执行数据和元数据 I/O 的选项。

QFS 默认值和 I/O 性能调整目标

磁盘 I/O（输入/输出）涉及 CPU 密集型操作系统请求和耗时的机械过程。因此 I/O 性能调整重点是最大程度地减少 I/O 相关系统开销，并且使传输指定数量的数据所需的机械工作保持绝对最少。这意味着减少每次数据传输的独立 I/O 数量（因而减少 CPU 执行的操作数量），并最大程度减少在每个独立 I/O 期间的寻道（重新定位读/写头）。因而 I/O 调整的基本目标如下：

- 在均匀地划分为平均文件大小的块中读取和写入数据。
- 读取和写入大型数据块。
- 在与底层介质的 512 字节的扇区边界对齐的单元中写入块，以便磁盘控制器在写入新数据之前不必读取和修改现有数据。
- 使小型 I/O 在高速缓存中排队等候并将较大的合并 I/O 写入磁盘。

Oracle HSM 默认设置为大部分通用文件系统的一系列典型应用程序和使用模式提供了最佳整体性能。但在必要时，您可调整默认行为以更好地与您的应用程序生成的 I/O 类型相匹配。可以指定最少连续读取或写入的大小。可以优化在设备上存储文件的方法。可以从针对通用性或高性能优化的文件系统之间进行选择。

磁盘分配单元和逻辑设备类型

文件系统按统一大小的块分配磁盘存储。此大小即磁盘分配单元 (disk allocation unit, DAU)，将确定每个 I/O 操作使用的最小连续空间量（无论写入多少数据量）以及在传输指定大小的文件时所需的 I/O 操作最小数量。如果块大小与文件的平均大小相比太大，则将浪费磁盘空间。如果块大小太小，则每次文件传输将需要更多 I/O 操作，这将损害性能。因此，在文件大小为基本块大小的偶数倍时，I/O 性能和存储效率最高。

出于这个原因，QFS 软件支持各种可配置 DAU 大小。创建 QFS 文件系统时，首先确定需要访问和存储的数据文件的平均大小。然后指定最均匀地划分为平均文件大小的 DAU。

首先选择最适合您的数据的 QFS 设备类型。有三种类型：

- *md* 设备
- *mr* 设备
- *gXXX* 分散读写组设备（其中 *XXX* 是范围 *[0-127]* 内的整数。

当文件系统将主要包含小文件或包含混合大小文件时，通常 *md* 设备是最佳选择。*md* 设备类型使用灵活的双重分配方案。将文件写入设备时，文件系统将对前八次写入使用 *4* KB 的小 DAU。接下来，它使用 *16*、*32* 或 *64* KB 的用户选择的大 DAU 写入任何剩余数据。小文件因而将写入合适的小型块，而大文件将写入针对其平均大小定制的大型块。

当文件系统将主要包含大文件和/或大小一致的文件时，*mr* 设备可能是更好的选择。*mr* 设备类型使用在范围 *[8-65528]* KB 中可按 8 KB 增量调整的 DAU。文件将写入非常接近平均文件大小的大型统一块，因此将最大程度降低读取/修改/写入开销并最大程度地提高性能。

分散读写组是均被视为单一逻辑设备的多达 128 个设备的聚合。类似于 *mr* 设备类型，分散读写组使用在范围 *[8-65528]* KB 中可按 8 KB 增量调整的 DAU。文件系统按每个磁盘一个 DAU 的方式，将数据并行写入分散读写组成员。因此聚合写入可能非常大。这使分散读写组在必须处理极大数据文件的应用程序中可能非常有用。

文件分配方法

默认情况下，非共享 QFS 文件系统使用分散读写分配，而共享文件系统使用循环分配。但您可在必要时更改分配。每种方法在一些情况下具有优势。

分散读写分配

指定分散读写分配之后，文件系统将在所有可用设备上并行分配空间。文件系统将数据文件分段并将一个段写入每个设备。每个段的大小由分散读写宽度（每个设备写入的 DAU 数）乘以系列集中的设备数量确定。设备可以是 *md* 磁盘设备、*mr* 磁盘设备或分散读写组。

分散读写一般会提高性能，因为文件系统并发读取而不是顺序读取多个文件段。单个设备上将并行出现多个 I/O 操作，从而减少每个设备的寻道开销。

不过，一次写入多个文件时，分散读写分配会产生高得多的寻道。过多寻道可能会导致性能严重降级，因此，如果您打算进行多个文件的同步 I/O，应考虑循环分配。

循环分配

指定循环分配之后，文件系统将连续分配存储空间，一次一个文件，并且一次一个设备。文件系统将文件写入具有可用空间的第一个设备。如果文件大于设备上剩余的空

间，则文件系统会将超过的大小写入下一个具有可用空间的设备。对于之后的每个文件，文件系统将移至下一个可用设备并重复此过程。当使用完最后一个可用设备时，文件系统将重新开始使用第一个设备。设备可以是 *md* 磁盘设备、*mr* 磁盘设备或分散读写组。

当应用程序同步执行多个文件的 I/O 时，循环分配可提高性能。此外，它对于共享 QFS 文件系统是默认设置（有关共享文件系统的更多信息，请参见“使用 Oracle HSM 软件从多台主机访问文件系统” 和 *mount_samfs* 手册页）。

存储分配和集成卷管理

与仅为一个设备或设备的一部分寻址的 UNIX 文件系统不同，QFS 文件系统执行其自己的卷管理。每个文件系统在内部处理提供物理存储的设备之间的关系，然后作为单一系列集 向操作系统提供存储。I/O 请求是通过标准 Solaris 设备驱动程序接口发出的，这与任何 UNIX 文件系统相同。

文件系统类型

有两种类型的 QFS 文件系统。每种文件系统有其自身的优势：

通用 ms 文件系统

QFS *ms* 文件系统实施起来最简单，非常适合大多数常见用途。这些文件系统将文件系统元数据与文件数据一起存储在相同的双重分配 *md* 磁盘设备上。该方法可简化硬件配置并能满足大多数需求。

高性能 ma 文件系统

QFS *ma* 文件系统可以提高严苛的应用中的数据传输速率。这些文件系统在专用设备上分开存储元数据和数据。元数据保存在 *mm* 设备上，而数据保存在一组 *md* 磁盘设备、*mr* 磁盘设备或分散读写组上。因此，元数据更新不会与用户和应用程序 I/O 竞争，并且设备配置无需适应两种不同种类的 I/O 工作负荷。例如，您可以将元数据放置在 RAID-10 镜像磁盘上以实现高冗余和快速读取，并将数据保存在空间效率更高的 RAID-5 磁盘阵列上。

Oracle HSM 归档文件系统

归档文件系统将一个或多个 QFS *ma* 或 *ms* 类型的文件系统与归档存储和 Oracle Hierarchical Storage Manager 软件组合在一起。Oracle HSM 软件将文件从文件系统的磁盘高速缓存复制到辅助磁盘存储和/或可移除介质中。该软件将副本作为文件系统不可分割的一部分进行管理。因此，文件系统将提供连续数据保护以及灵活有效地存储超大文件的能力，这些文件存储在磁盘或固态介质上的开销会非常昂贵。

正确配置的 Oracle HSM 文件系统将在不需要独立备份应用程序的情况下提供连续数据保护。该软件在文件创建或更改时按照用户定义的策略中的指定自动复制文件数据。在混用磁盘和磁带介质的情况下，使用本地和远程资源可以保留多达四个副本。

文件系统元数据将记录文件和所有副本的位置。软件将提供一系列快速查找副本的工具。因此，丢失或损坏的文件可从归档中轻松恢复。备份副本将使用符合 POSIX 标准的 *tar*（磁带归档）标准格式保留，这还使您能够在即使 Oracle HSM 软件不可用的情况下恢复数据。Oracle HSM 通过动态检测 I/O 错误并从此类错误恢复，随时保持文件系统元数据的一致性。因此，您可以恢复文件系统而无需执行耗时的完整性检查，这在存储成千上万个文件和 PB 级数据时是需要特别注意的事项。如果文件系统元数据存储在单独的设备上，并且仅涉及到数据存储磁盘，则在将替换磁盘配置为文件系统时，会执行完全恢复。当用户请求位于失败磁盘上的文件时，Oracle HSM 会自动将磁带中的备份副本回写到替换磁盘。如果元数据也丢失了，则管理员可使用 *samfsrestore* 命令从 *samfsdump* 备份文件恢复它。在恢复元数据后，文件可在用户请求它们时重新从磁带中恢复。由于文件仅在请求时才恢复到磁盘，因此恢复过程将有效使用网络带宽并且对正常操作的影响极小。

这种在高性能的主磁盘或固态介质和低成本、高密度的辅助磁盘、磁带或光盘介质上同步管理文件的能力，使得 Oracle HSM 文件系统非常适合经济地存储非常大和/或很少使用的文件。非常大、连续访问的数据文件（例如卫星图像和视频文件）可专门存储在磁带上。当用户或应用程序访问文件时，文件系统会自动将文件回写到磁盘或直接从磁带将其读入内存中，这取决于所选文件配置。对于主要出于历史记录或符合性目的而保留的记录，可以使用与文件生命周期的某个指定时间的用户访问模式和成本限制最一致的介质，按分层结构方式进行存储。最初，当用户仍偶尔访问某个文件时，可以将其归档在低成本的辅助磁盘设备上。随着需求减少，您可以仅在磁带或光盘介质上保留副本。但是，例如，当用户需要数据来响应法律取证或法规流程时，文件系统可自动将所需材料回写到延迟最小的主磁盘上，就像它一直在这里。出于法律和法规目的，Oracle HSM 文件系统可启用 WORM。启用 WORM 的文件系统支持默认的和可定制的文件保留期、数据和路径的不可更改性以及 WORM 设置的子目录继承性。可使用手动和/或自动介质验证来监视长期数据完整性。

管理和维护归档文件系统有四个基本的 Oracle HSM 过程：

- 归档
- 回写
- 释放
- 回收。

归档

归档过程将文件从文件系统复制到保留用来存储活动文件副本的归档介质。归档介质可以包括可移除介质卷，如盒式磁带和/或一个或多个位于磁盘或固态存储设备上的文件系统。归档文件副本可以为活动文件、长期保留的不活动文件或这两种文件的某些组合提供备份冗余。

在 Oracle HSM 归档文件系统中，活动的联机文件、归档副本和关联存储资源将构成一个逻辑资源，即归档集。归档文件系统中的每个活动文件均完全属于一个归档集。每个归档集可包括每个文件的多达四个归档副本以及控制该归档集的归档过程的策略。

归档过程由 UNIX 守护进程（服务）*sam-archiverd* 管理。此守护进程计划归档活动并调用执行所需任务的进程 *archiver*、*sam-arfind* 和 *sam-arcopy*。

archiver 进程读取可编辑配置文件 *archiver.cmd* 中的归档策略，并按规定设置剩余归档进程。此文件中的指令控制归档过程的常规行为，按文件系统定义归档集并且指定制作的副本数量以及用于每个副本的归档介质。

接下来，*sam-archiverd* 守护进程将为当前挂载的每个文件系统启动 *sam-arfind* 进程。*sam-arfind* 进程将扫描其为新文件、已修改文件、已重命名文件以及要重新归档或取消归档的文件分配的文件系统。默认情况下，此进程将不断扫描文件和目录的更改，因为这将提供最佳的整体性能。但是，例如，如果您必须维持与较旧 StorageTek Storage Archive Manager 实现的兼容性，则可编辑 *archiver.cmd* 文件中的归档集规则，以使用多种方法之一来计划扫描（有关详细信息，请参见 *sam-archiverd* 手册页）。

在它标识候选文件之后，*sam-arfind* 将标识定义文件的归档策略的归档集。*sam-arfind* 进程会通过将文件的属性与每个归档集定义的选择条件进行比较来标识归档集。这些条件可包括下列一个或多个文件属性：

- 文件的目录路径以及（可选）与一个或多个候选文件名匹配的正则表达式
- 与一个或多个候选文件的所有者匹配的指定用户名
- 与文件的关联组匹配的指定组名称
- 小于或等于候选文件大小的指定最小文件大小
- 大于或等于候选文件大小的指定最大文件大小。

在它找到正确的归档集和对应的归档参数之后，*sam-arfind* 将检查文件的归档时限等于还是超出归档集指定的阈值。文件的归档时限是自文件创建、上次修改（默认值）或上次访问以来经过的秒数。如果归档时限满足策略中指定的时限条件，则 *sam-arfind* 会将文件添加到归档集的归档请求队列并为其分配优先级。优先级基于归档集中指定的规则和已存在的归档副本数量、文件大小、任何未解决的操作员请求以及任何其他依赖归档副本创建的操作等因素。

sam-arfind 标识需要归档的文件之后，将对它们按优先级排序，然后将它们添加到每个归档集的归档请求，它会将请求返回给 *sam-archiverd* 守护进程。守护进程将编写每个归档请求。它会将数据文件安排到按大小排列的归档文件中，以便有效利用介质，并将文件有效地写入可移除介质，以后再从可移除介质中重新调用。此守护进程将接受任何文件排序参数或您在 *archiver.cmd* 文件中设置的介质限制（有关详细信息，请参见 *archiver.cmd* 手册页）。但请注意，限制软件自由选择介质的能力通常会降低性能和介质利用率。在组装归档文件之后，*sam-archiverd* 将对归档请求按优先级排序，以便复制进程可通过最少的挂载操作传输最多的文件（有关详细信息，请参见 *sam-archiverd* 手册页的调度部分）。*sam-archiverd* 将调度复制操作，以便在任何指定时间，它们需要的设备数量不超过归档集策略和/或机械装置磁带库允许的最大数量。

在调度归档请求之后，*sam-archiverd* 将为调度的每个归档请求和驱动器调用 *sam-arcopy* 进程的实例。然后，*sam-arcopy* 实例会将数据文件复制到归档介质上的归档文件，更新归档文件系统的元数据以反映新副本的存在并更新归档日志。

当 *sam-arcopy* 进程退出时，*sam-archiverd* 守护进程将检查归档请求是否存在由高速缓存磁盘的读取错误、可移除介质卷的写入错误以及已打开、已修改或已删除文件导致的错误或疏忽。如果任何文件尚未归档，则 *sam-archiverd* 将重新编写归档请求。

sam-arfind 和 *sam-arcopy* 进程可以使用 *syslog* 工具和 *archiver.sh* 创建归档活动、警告和信息性消息的连续记录。生成的归档程序日志包含有价值的诊断和历史信息，包括每个归档文件每个副本的位置和处理的详细记录。因此，在灾难恢复过程中，通常可以使用归档日志恢复丢失的数据文件，否则这些文件是不可恢复的（有关详细信息，请参见客户文档库中的《*Oracle Hierarchical Storage Manager and StorageTek QFS Software* 文件系统恢复指南》）。文件系统管理员在 *archiver.cmd* 文件中使用 *logfile=* 指令启用归档程序日志记录并定义日志文件。有关日志文件的更多信息，请参见 *archiver.cmd* 手册页。

回写

回写过程将文件数据从归档存储中复制回主磁盘高速缓存中。当应用程序尝试访问脱机文件（当前在主存储中不可用的文件）时，会有一个归档副本将自动回写到（复制回）主磁盘。然后，应用程序可快速访问此文件，即使尚未将完整的数据写回到磁盘中也是如此，因为读取操作可紧随在回写操作之后执行。如果出现介质错误或如果特定介质卷不可用，则回写过程将使用第一个可用设备自动装入下一可用副本（如果有）。这样，回写过程使得归档存储对用户和应用程序是透明的。所有文件看起来似乎在磁盘上一直可用。

默认回写行为适合大多数文件系统。不过，您可以通过在配置文件 */etc/opt/SUNWsamfs/stager.cmd* 中插入或修改指令来更改默认行为，您还可以从命令行按照每个目录或每个文件覆盖这些指令。例如，要访问大文件中的小型记录，您可选择直接从归档介质访问数据，而不回写文件。或者，您可以只要在一组相关文件中的任一文件回写时就对该组文件进行回写（使用关联回写功能）。有关详细信息，请参见 *stage* 和 *stager.cmd* 手册页。

释放

释放过程将通过删除当前未在使用的以前归档文件的联机副本，来释放主磁盘高速缓存空间。在文件复制到归档介质（例如磁盘归档或磁带卷）之后，可在应用程序访问文件时对它进行回写。因此当其他文件需要空间时，无需将它保留在磁盘高速缓存中。释放操作通过从磁盘高速缓存中删除不需要的副本，来确保主高速缓存存储始终可供新创建和当前所用的文件使用，即使文件系统在主存储容量未有任何相应增加的情况下发生增长也是如此。

当高速缓存利用率超出上限并持续在下限（您在挂载归档文件系统时设置的两个可配置阈值）之上时，会自动执行释放操作。上限可确保始终提供足够的空闲空间，而下

限可确保始终在高速缓存中提供合理数量的文件并且介质挂载操作因而保持必要的最低数量。上限值的典型值为 80%，下限值的典型值为 70%。

使用默认行为的按水位标志释放适合大多数文件系统。不过，您可以通过在配置文件 */etc/opt/SUNWsamfs/releaser.cmd* 中修改或添加指令来更改默认行为，您还可以从命令行按照每个目录或每个文件覆盖这些指令。例如，您可以部分释放按顺序访问的大文件，这样应用程序首先读取始终保留在磁盘上的文件部分，而文件的其余部分将从归档介质中回写。有关详细信息，请参见 *release* 和 *releaser.cmd* 手册页。

回收

通过删除不再使用的归档副本，回收过程可释放归档介质上的空间。当用户修改文件后，与文件的较旧版本关联的归档副本最终将会过期。回收程序将标识保留最大比例的已过期归档副本的介质卷。如果过期文件存储在归档磁盘卷上，则回收程序进程将删除它们。如果这些文件位于可移除介质（如磁带卷）上，则回收程序会将保留在目标卷上的任何未过期的副本重新归档至其他介质。然后，回收程序调用可编辑的脚本 */etc/opt/SUNWsamfs/scripts/recycler.sh*，以便为回收的卷重新设置标签，将其从库中导出，或执行用户定义的其他某些操作。

默认情况下，回收过程不会自动运行。您可以配置 Solaris *crontab* 文件，以便在合适的时间运行回收过程。您也可以根据需要使用命令 */opt/SUNWsamfs/sbin/sam-recycler* 从命令行运行回收过程。要修改默认回收参数，请编辑文件 */etc/opt/SUNWsamfs/archiver.cmd* 或创建单独的 */etc/opt/SUNWsamfs/recycler.cmd* 文件。有关详细信息，请参见相应的手册页。

2

第 2 章 配置主机系统

先配置 Oracle Hierarchical Storage Manager and StorageTek QFS Software 的主机操作系统，然后再继续安装和配置。本章概述了以下主题：

- 针对 Oracle HSM 配置 Oracle Solaris
- 针对 Oracle HSM 客户机配置 Linux

针对 Oracle HSM 配置 Oracle Solaris

要配置 Solaris 主机以便与 Oracle HSM 软件和 QFS 文件系统配合使用，请执行以下任务：

- 安装最新的操作系统更新
- 针对预测的文件系统 I/O 调整 Solaris 系统和驱动程序参数

安装最新的操作系统更新

如果可以，请始终为 Solaris 操作系统安装最新的修补程序和更新。如果您需要使用 Oracle Hierarchical Storage Manager and StorageTek QFS Software 发行版 6.1 中提供的最新功能，则必须在所有 Solaris 主机上安装 Oracle Solaris 11 操作系统软件。有关与软件配合使用的建议最低操作系统发行版的完整信息，请参见发行说明和 `support.oracle.com`。

有关选择的 Solaris 版本的安装和更新说明，请参见相应的客户文档库中的安装和管理文档、Oracle 技术网 (Oracle Technical Network, OTN) 和 `support.oracle.com` 上的知识。如果您刚接触映像包管理系统 (Image Packaging System, IPS)，则以下 OTN 文章对您非常有帮助：

- 《*Introducing the Basics of Image Packaging System (IPS) on Oracle Solaris 11*》（《Oracle Solaris 11 映像包管理系统 (Image Packaging System, IPS) 基础知识简介》），作者 Glynn Foster（2011 年 11 月）
- 《*How to Update Oracle Solaris 11 Systems From Oracle Support Repositories*》（《如何从 Oracle 支持系统信息库更新 Oracle Solaris 11 系统》），作者 Glynn Foster（2012 年 3 月）
- 《*More Tips for Updating Your Oracle Solaris 11 System from the Oracle Support Repository*》（《有关从 Oracle 支持系统信息库更新 Oracle Solaris 11 系统的更多提示》），作者 Peter Dennis（2012 年 5 月）。

针对预测的文件系统 I/O 调整 Solaris 系统和驱动程序参数

系统端到端输入/输出 (input/output, I/O) 性能在操作系统、驱动程序、文件系统和应用程序以无需进行不必要分段和重新缓存的单元传输数据时最高。因此，请针对应用程序和文件系统可以实现的最大数据传输设置 Solaris。执行如下操作：

1. 以 *root* 用户身份登录到 Oracle HSM 文件系统主机。

```
root@solaris:~#
```

2. 制作 */etc/system* 文件的备份副本，然后在文本编辑器中打开 */etc/system*。

   在示例中，使用 *vi* 编辑器。

```
root@solaris:~# cp /etc/system /etc/system.backup
root@solaris:~# vi /etc/system
*ident  "%Z%%M% %I%     %E% SMI" /* SVR4 1.5 */
* SYSTEM SPECIFICATION FILE
...
```

3. 在 */etc/system* 文件中，设置 *maxphys*，即任何驱动程序能够作为单个单元处理的最大物理 I/O 请求大小，它等于应用程序和文件系统能够实现的最大数据传输。以 *set maxphys = 0xvalue* 的形式输入一行，其中 *value* 是代表字节数的十六进制数字。然后保存文件并关闭编辑器。

   驱动程序将超过 *maxphys* 的请求拆分成 *maxphys* 大小的片段。默认值根据操作系统发行版而不同，但是通常大约为 128 千字节。在示例中，我们将 *maxphys* 设置为 *0x800000*（8,388,608 字节或 8 兆字节）：

```
root@solaris:~# vi /etc/system
*ident  "%Z%%M% %I%     %E% SMI" /* SVR4 1.5 */
* SYSTEM SPECIFICATION FILE
...
set maxphys = 0x800000
:wq
root@solaris:~#
```

4. 在文本编辑器中打开 */kernel/drv/sd.conf* 文件。

   在示例中，使用 *vi* 编辑器：

```
root@solaris:~# vi /kernel/drv/sd.conf
# Copyright (c) 1991, 2010, Oracle and/or its affiliates. All rights reserved.
name="sd" class="scsi" target=0 lun=0;
name="sd" class="scsi" target=1 lun=0;
...
```

```
# Associate the driver with devid resolution.
ddi-devid-registrant=1;
```

5. 在 */kernel/drv/sd.conf* 文件中，将 *sd_max_xfer_size*（即 SCSI 磁盘 (*sd*) 驱动程序可以处理的最大数据传输大小）设置为您为 *maxphys* 设置的值。以 *sd_max_xfer_size=0xvalue;* 的形式输入一行，其中 *value* 为代表字节数的十六进制数字。保存文件并关闭编辑器。

   默认值为 *0x100000*（1048576 字节或一兆字节）。在示例中，我们添加注释并将 *sd_max_xfer_size* 设置为 *0x800000*（8,388,608 字节或 8 兆字节）。

```
...
# Associate the driver with devid resolution.
ddi-devid-registrant=1;
# Set SCSI disk maximum transfer size
sd_max_xfer_size=0x800000;
:wq
root@solaris:~#
```

6. 在文本编辑器中打开 */kernel/drv/ssd.conf* 文件。

   在示例中，使用 *vi* 编辑器。

```
root@solaris:~# vi /kernel/drv/ssd.conf
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
name="ssd" parent="sf" target=0;
name="ssd" parent="fp" target=0;
...
name="ssd" parent="ifp" target=127;
```

7. 在 */kernel/drv/ssd.conf* 文件中，将 *ssd_max_xfer_size*（即光纤通道磁盘 (*ssd*) 驱动程序可以处理的最大数据传输大小）设置为您为 *maxphys* 设置的值。以 *ssd_max_xfer_size=0xvalue;* 的形式输入一行，其中 *value* 是代表字节数的十六进制数字。然后保存文件并关闭编辑器。

   默认值为 *0x100000*（1048576 字节或一兆字节）。在示例中，我们添加注释并将 *ssd_max_xfer_size* 设置为 *0x800000*（8,388,608 字节或 8 兆字节）：

```
...
name="ssd" parent="ifp" target=127;
# Set Fibre Channel disk maximum transfer size
ssd_max_xfer_size=0x800000;
:wq
root@solaris:~#
```

8. 重新启动系统。使用命令 *init 6*。

```
root@solaris:~# init 6
```

9. 如果您准备的解决方案包含其他 Solaris 主机，则重复执行“针对 Oracle HSM 配置 Oracle Solaris” 中指定的任务，直到已配置所有 Solaris 主机。
10. 如果您准备的解决方案包含一个或多个 Linux 客户机，则转至"针对 Oracle HSM 客户机配置 Linux"。
11. 否则，请转至第 3 章 配置存储主机和设备。

针对 Oracle HSM 客户机配置 Linux

在您安装 Oracle HSM 客户机软件之前，必须按如下方式准备 Linux 操作系统：

- 禁用不兼容的操作系统功能
- 安装必需的内核开发和实用程序软件包

禁用不兼容的操作系统功能

1. 以 *root* 用户身份登录 Oracle HSM 客户机主机。

```
[root@linux ~]#
```

2. 如果安装了 SELinux（安全 Linux），则禁用它。在文本编辑器中打开文件 */etc/selinux/config*，将 *SELINUX* 标志设置为 *disabled*，保存文件，关闭编辑器，然后重新引导。

   Oracle HSM 不支持 SELinux，默认情况下，该功能在 Oracle Linux 和 Red Hat Enterprise Linux 上处于启用状态。在示例中，用 *vi* 编辑器打开文件。

```
[root@linux ~]# vi /etc/selinux/config
# This file controls the state of SELinux on the system.
...
#SELINUX=enforcing
#SELINUX=permissive
SELINUX=disabled
SELINUXTYPE=targeted
:wq
[root@linux ~]# reboot
```

3. 如果安装了 AppArmor，则使用适用于您的 Linux 分发包的文档中建议的过程禁用它。

   AppArmor 有时用作 SELinux 的替代功能。Oracle HSM 不支持 AppArmor。

4. 接下来，安装必需的内核开发和实用程序软件包。

安装必需的内核开发和实用程序软件包

在安装 Oracle HSM 客户机软件之前，必须先安装 Linux 内核开发软件包以及一些指定的实用程序软件包。要确定并安装必需的软件包，请使用下面的过程：

1. 以 *root* 用户身份登录 Linux 客户机主机。

   在示例中，客户机在 Oracle Linux 上托管：

```
[root@linux ~]#
```

2. 确定客户机上安装的内核版本。使用命令 *uname -r*。

   在示例中，内核版本为 *2.6.9-89.0.0.0.1.EL*：

```
[root@linux ~]# uname -r
2.6.9-89.0.0.0.1.EL
[root@linux ~]#
```

3. 安装内核开发工具包 *kernel-devel-kernel-version*，其中 *kernel-version* 是您在前面步骤中确定的版本字符串。

   Oracle HSM 客户机安装要求此软件包中包含 *Module.symvers*。在示例中，使用带有参数 *-y install*（*-y* 用于确保针对所有提示自动回答“是”）的 Oracle Linux 命令 *yum*：

```
[root@linux ~]# yum -y install / kernel-devel-2.6.9-89.0.0.0.1.EL.i686.rpm
[root@linux ~]#
```

4. 查看 Korn shell *ksh* 是否已安装。如果未安装，则安装它。

   在示例中，我们将 Oracle Linux 命令 *rpm -qa* 的输出通过管道传输至 *grep* 命令并搜索字符串 *ksh*。该命令返回的输出为 no，这表示 *ksh* 未安装。因此我们使用命令 *yum install ksh* 安装它：

```
[root@linux ~]# rpm -qa | grep ksh
[root@linux ~]#
[root@linux ~]# yum install ksh
...
--> Running transaction check
---> Package ksh-20100621-19.e16.x86_64 set to be installed

================================================================================
Package            Arch          Version                  Repository      Size
```

```
================================================================================
Installing:
 ksh               i686          2.6.9-89.0.0.0.1.EL       updates        506 k
...
Installed:
  ksh-2.6.9-89.0.0.0.1.EL.i686
Complete!
[root@linux ~]#
```

5. 查看 *cpio* 实用程序是否已安装。如果未安装，则安装它。

   在示例中，我们将 Oracle Linux 命令 *rpm -qa* 的输出通过管道传输至 *grep* 命令并搜索字符串 *cpio*。该命令返回版本信息，因此 *cpio* 实用程序已安装：

```
[root@linux ~]#  rpm -qa | grep cpio
cpio-2.10-10.e16.x86_64
[root@linux ~]#
```

6. 查看 *find* 实用程序是否已安装。如果未安装，则安装它们。

   在示例中，我们将 Oracle Linux 命令 *rpm -qa* 的输出通过管道传输至 *grep* 命令并搜索字符串 *findutils*。该命令返回版本信息，说明 *findutils* 软件包已安装：

```
[root@linux ~]#  rpm -qa | grep findutils
findutils-4.4.2-6.e16.x86_64
[root@linux ~]#
```

7. 查看 *gcc* 编译器是否已安装。如果未安装，则安装它。

   在示例中，我们将 Oracle Linux 命令 *rpm -qa* 的输出通过管道传输至 *grep* 命令并搜索字符串 *gcc*。该命令返回版本信息，说明 *gcc* 编译器已安装：

```
[root@linux ~]#  rpm -qa | grep gcc
gcc-4.4.7-3.e16.x86_64
libgcc-4.4.7-3.e16.x86_64
[root@linux ~]#
```

8. 查看 *make* 实用程序是否已安装。如果未安装，则安装它。

   在示例中，我们将 Oracle Linux 命令 *rpm -qa* 的输出通过管道传输至 *grep* 命令并搜索字符串 *make*。该命令返回版本信息，说明 *make* 实用程序已安装：

```
[root@linux ~]#  rpm -qa | grep make
make-4.4.7-3.e16.x86_64
libmake-3.81.20.e16.x86_64
```

```
[root@linux ~]#
```

9. 查看 *binutils* 软件包是否已安装。如果未安装，则安装它。

   如果 Oracle HSM 安装软件需要构建 Linux 内核，则它需要 *nm* 实用程序，该实用程序包含在此软件包中。在本示例中，我们将 Oracle Linux 命令 *rpm -qa* 的输出通过管道传输至 *grep* 命令并搜索字符串 *nm*。该命令返回版本信息，说明 *nm* 实用程序已安装：

```
[root@linux ~]#  rpm -qa | grep nm
binutils-2.20.51.0.2-5.34.e16.x86_64
[root@linux ~]#
```

10. 查看 *rpmbuild* 软件包是否已安装。如果未安装，则安装它。

    在示例中，我们将 Oracle Linux 命令 *rpm -qa* 的输出通过管道传输至 *grep* 命令并搜索字符串 *rpmbuild*。该命令返回版本信息，说明 *rpmbuild* 软件包已安装：

```
[root@linux ~]#  rpm -qa | grep rpmbuild
rpm-build-4.8.0-37.el6.x86_64
[root@linux ~]#
```

11. 查看 *rpm* 软件包是否已安装。如果未安装，则安装它。

    如果 Oracle HSM 安装软件需要构建 Linux 内核，则它需要 *rpm2cpio* 实用程序，该实用程序包含在此软件包中。在示例中，我们将 Oracle Linux 命令 *rpm -qa* 的输出通过管道传输至 *grep* 命令并搜索字符串 *rpm*。该命令返回版本信息，说明该实用程序已安装：

```
[root@linux ~]#  rpm -qa | grep rpm
rpm-4.8.0-27.e16.x86_64
rpm-libs-4.8.0-27.e16.x86_64
rpm-python-4.8.0-27.e16.x86_64
[root@linux ~]#
```

12. 如果您准备的解决方案包含其他 Linux 客户机，则重复执行“针对 Oracle HSM 客户机配置 Linux” 中指定的任务，直到已配置所有 Linux 客户机。
13. 接下来，转至第 3 章 配置存储主机和设备。

3

第 3 章 配置存储主机和设备

在执行 Oracle HSM 安装和配置之前，请先执行本章概述的存储配置任务。本章概述了以下主题：

- 配置主要存储
- 配置归档存储
- 为高可用性系统配置存储

配置主要存储

在 Oracle HSM 文件系统中，主要磁盘或固态磁盘设备用于存储目前使用和修改的文件。请按照以下准则为高速缓存配置磁盘或固态磁盘设备。

为主要高速缓存配置设备

1. 要估算主高速缓存的起始容量，请确定每个文件系统在填满时会存储多少数据。
2. 将该起始容量增加 10% 以便为文件系统元数据留出空间。
3. 如果您正在为高性能 *ma* 类型文件系统做准备，请为 *mm* 元数据设备配置硬件。理想情况是为每个 *mm* 元数据设备配置一个硬件控制的四磁盘 RAID 10 (1+0) 卷组。为实现最高性能，请考虑使用固态磁盘设备。

   条带镜像 RAID 10 阵列的特征适合存储 Oracle HSM 元数据。RAID 10 存储硬件高度冗余，因此关键元数据将得到保护。与大多数其他 RAID 配置相比，吞吐量更高且延迟更低。

   与由使用共享的通用处理器运行的软件所控制的阵列相比，由专用控制器硬件所控制的阵列通常可提供更高性能。

   固态设备特别适合存储本质上会频繁更新并被频繁读取的元数据。
4. 如果您使用外部磁盘阵列进行主要高速缓存存储，请为文件系统配置中的每台 *md* 或 *mr* 设备配置 3+1 或 4+1 RAID 5 卷组。在每个卷组上配置一个逻辑卷 (LUN)。

   对于给定数量的磁盘，较小的 3+1 和 4+1 RAID 5 卷组比较大的卷组提供的并行性很好，因此可提供更高的输入/输出 (input/output, I/O) 性能。RAID 5 卷组中的各个磁盘设备不会独立操作－从 I/O 角度看，每个卷组就像一台设备一样操作。因此，与等效的较大配置相比，将给定数量的磁盘划分为 3+1 和 4+1 卷组可产生更独立的设备、更好的并行性和更少的 I/O 争用。

由于奇偶校验与存储的比率较高，较小的 RAID 组的容量也较低。但是，对于大多数用户而言，性能的提高足以抵消这点。在归档文件系统中，磁盘高速缓存容量的小幅降低通常会被归档中相对而言无限的可用容量所完全抵消。

在一个卷组上配置多个逻辑卷 (LUN) 会使逻辑上分离的卷的 I/O 争用一次只能为一个 I/O 服务的一组资源。这会增加 I/O 相关系统开销并降低吞吐量。

5. 接下来，开始配置归档存储。

配置归档存储

执行以下任务：

- 对 SAN 连接的设备进行区域划分
- 配置归档磁盘存储
- 配置归档磁带存储

对 SAN 连接的设备进行区域划分

确保对存储区域网络 (storage area network, SAN) 进行区域划分，以便可以在驱动器与 Oracle HSM 主机上的主机总线适配器之间进行通信。要检查区域划分，请执行如下操作：

对 Oracle HSM 配置中的所有设备正确进行区域划分

1. 确保主机可以发现 SAN 上的设备。输入 Solaris 配置管理命令 *cfgadm* 以及 *-al*（挂接点列表）和 *-o show_SCSI_LUN* 选项。检查驱动器端口的全局名称 (World Wide Name, WWN) 的输出。

   输出的第一列显示挂接点 ID (*Ap_id*)，其中包括主机总线适配器的控制器编号和 WWN（以冒号分隔）。如果节点是通过 ADI 接口控制介质转换器的桥接驱动器，*-o show_SCSI_LUN* 选项会显示节点上的所有 LUN。

```
root@solaris:~# cfgadm -al -o show_SCSI_LUN
Ap_Id      Type Receptacle Occupant   Condition
c2::500104f000937528   tape connected  configured   unknown
c3::50060160082006e2,0 tape connected  unconfigured unknown
```

2. 如果驱动器的 WWN 未列在 *cfgadm -al -o show_SCSI_LUN* 的输出中，则表明驱动器不可见。SAN 配置出现错误。因此，请再次检查 SAN 连接和区域划分配置。然后重复上述步骤。
3. 如果 *cfgadm -al* 命令的输出表明某驱动器未配置，请再次运行该命令，但这次使用 *-c*（配置）开关。

   该命令会在 */dev/rmt* 中生成必要的设备文件：

```
root@solaris:~# cfgadm -al
Ap_Id     Type Receptacle Occupant   Condition
c2::500104f000937528   tape connected  configured   unknown
c3::50060160082006e2,0 tape connected  unconfigured unknown
root@solaris:~# cfgadm -c configure 50060160082006e2,0
```

4. 验证设备名称与全局名称之间的关联。使用命令 *ls -al /dev/rmt | grep WWN*，其中 *WWN* 是全局名称。

```
root@solaris:~# ls -al /dev/rmt | grep 50060160082006e2,0
lrwxrwxrwx 1 root root 94 May 20 05:05 3un -> /
../../devices/pci@1f,700000/SUNW,qlc@2/fp@0,0/st@w50060160082006e2,0:
```

5. 如果您具有建议的最低 Solaris 修补程序级别，请现在配置磁盘存储。
6. 否则，请获取您设备的目标 ID。
7. 编辑 */kernel/drv/st.conf*。在 *tape-config-list* 中添加供应商指定的条目，以指定上面所确定的目标 ID。
8. 强制重新装入 *st* 模块。使用命令 *update_drv -f st*。

```
root@solaris:~# update_drv -f st
root@solaris:~#
```

9. 接下来，配置磁盘存储。

配置归档磁盘存储

Oracle HSM 归档文件系统可以将文件归档到磁盘以及磁带介质。磁盘文件系统配置为磁盘归档时，软件使用该文件系统有些类似于使用盒式磁带。它按卷序列号 (volume serial number, VSN) 处理文件系统并将文件副本存储在磁带归档 (*tar*) 文件中。

基于磁盘的归档存储可以提高归档解决方案的灵活性和冗余。随机访问磁盘设备不会引发与顺序磁盘设备关联的挂载和定位开销。所以，当归档和检索大量小文件的解决方案在磁盘上存储每个文件的第一个副本时，可以更快速且可靠地执行此操作。必须在异地介质上维护副本的归档解决方案可以经常执行此操作，只需要将副本写入远程主机上已挂载 NFS 的磁盘驻留文件系统。

通过将远程主机的有限本地磁盘空间用作基本上无限量的、基于云的存储的前端高速缓存，Oracle Storage Cloud Software Appliance (OSCSA) 可以进一步扩展此类已挂载 NFS 的归档存储的实用性。该设备包含随网络文件系统版本 4 (Network File System version 4, NFSv4) 配置的 Oracle Linux 7（或更高版本）主机、开源 Docker Engine 1.6（或更高版本）容器管理软件以及 Oracle Storage Cloud Software Appliance Docker 映像。

如果您计划使用归档磁盘存储，请首先确定所需的总容量、归档卷数以及文件系统数。然后，如果您计划在 Oracle 存储云中配置归档磁盘存储，请置备所需的 Oracle Storage Cloud Software Appliances。

确定容量、卷和文件系统要求

计划足够的硬件资源来处理预期的工作负荷。如果并发的 Oracle HSM 归档和回写操作必须互相竞争或者与相同物理设备集的其他应用程序竞争，则会降低性能。因此，请遵循下面列出的准则。

1. 每个 Oracle HSM 操作以及每 10 到 20 TB 的归档数据允许使用一个磁盘卷（磁盘或 RAID 组）。

   如果磁盘卷是磁盘设备池的动态分配的存储，则设置配额。确保未超额订阅底层物理存储。
2. 每个磁盘卷允许一个文件系统。

   请勿在位于相同物理驱动器或 RAID 组上的 LUN 上配置两个或更多个文件系统。
3. 计划将每个文件系统用作单个磁盘归档。

   请勿将子目录用作单独的归档卷。
4. 计划将每个文件系统直接用于归档。

   请勿将通用文件系统用作磁盘归档。
5. 接下来，创建将用作已挂载 NFS 的磁盘归档的任何远程文件系统。

创建远程文件系统以用作已挂载 NFS 的磁盘归档

1. 创建要用作磁盘归档的任何远程文件系统。

   创建新的专用文件系统。请勿使用必须与其他应用程序共享的现有通用文件系统。

   请注意，稍后您配置 Oracle HSM 服务器时，将创建在本地挂载的磁盘归档文件系统。
2. 如果您计划将 Oracle Storage Cloud 用作磁盘归档解决方案的一部分，则现在配置 Oracle Storage Cloud Software Appliance 主机。
3. 否则，配置归档磁带存储。

配置 Oracle Storage Cloud Software Appliance 主机

1. 从 Oracle Cloud > Public > Infrastructure > Storage > Storage Cloud Software Appliance (*`http://docs.oracle.com/cloud/latest/storagecs_common/CSSGU/`*) 下载最新的 OSCSA 文档。

   为方便起见，此过程汇总了配置过程和系统要求。但是，请始终参考 OSCSA 产品文档和 *README* 文件以了解完整的最新信息。

2. 请联系您的 Oracle 销售团队。购买 Oracle Storage Cloud Service 订阅并请求 Oracle Storage Cloud Software Appliance 映像。
3. 对于每个设备主机，提供通用 x86 服务器，该服务器至少具有两个双核中央处理器 (central processor, CPU) 和 4 GB 随机存取存储器 (random access memory, RAM)。
4. 在每个 OSCSA 主机上安装 Oracle Linux 7（内核版本 3.10 或更高版本）。

   可以从 Oracle Software Delivery Cloud (*https://edelivery.oracle.com/*) 获取 Oracle Linux。
5. 在每个 OSCSA 主机上安装 Docker 1.6.1 或更高版本。

   Docker 是软件容器的开源分布平台。可以从 Docker (*https://www.docker.com*) 获取 Docker。
6. 在每个 OSCSA 主机上安装网络文件系统版本 4 (Network File System version 4, NFSv4) 服务。

   Oracle HSM 主机使用 NFSv4 来远程挂载构成 OSCSA 前端高速缓存的 Linux 文件系统。
7. 根据 OSCSA 文档 *Using Oracle Storage Cloud Software Appliance* (*http://docs.oracle.com/cloud/latest/storagecs_common/CSSGU/*)（使用 Oracle Storage Cloud Software Appliance）中的说明安装和配置 Oracle Storage Cloud Software Appliance。
8. 按照 OSCSA 文档中所述创建 OSCSA 高速缓存文件系统。
9. 接下来，配置磁带存储。

配置归档磁带存储

执行以下任务：

- 确定驱动器在库中的安装顺序
- 配置直接连接的库（如果有）。

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 Oracle HSM 主配置文件 (*mcf*) 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 */var/adm/messages* 文件中报告这些设备的顺序。

对于每台 Oracle HSM 元数据服务器和数据移动器主机，请执行下列任务以决定驱动器顺序：

- 从库和 Solaris 主机收集驱动器的标识信息。
- 然后，按照适用于直接连接或 ACSLS 连接的库的过程，将驱动器映射到 Solaris 设备名称。

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/scsi/changer/* 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，将 Solaris 设备名称映射到您的直接连接或 ACSLS 连接的库中的设备。

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. 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。将信息组织成简单的表。

   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
LIBRARY SOLARIS      SOLARIS
DEVICE  LOGICAL      PHYSICAL
NUMBER  DEVICE       DEVICE
------- ----------   ------------------------------------------
        /dev/rmt/0 -> ../../devices/pci@1f,4000/scsi@2,1/st@2,0:
        /dev/rmt/1 -> ../../devices/pci@1f,4000/scsi@4,1/st@5,0:
        /dev/rmt/2 -> ../../devices/pci@1f,4000/scsi@4,1/st@6,0:
        /dev/rmt/3 -> ../../devices/pci@1f,4000/scsi@4/st@1,0:
```

```
lrwxrwxrwx 1 root root 40 Mar 18 2014 /dev/rmt/4 -> ../../devices/pci@1f,4000/scsi@4/st@2,0:
```

3. 在库中，确保所有驱动器都是空的。
4. 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。
5. 确定与挂载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
```

6. 在设备映射文件中，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/0 -> ../../devices/pci@1f,4000/scsi@2,1/st@2,0:
   1    /dev/rmt/1 -> ../../devices/pci@1f,4000/scsi@4,1/st@5,0:
        /dev/rmt/2 -> ../../devices/pci@1f,4000/scsi@4,1/st@6,0:
        /dev/rmt/3 -> ../../devices/pci@1f,4000/scsi@4/st@1,0:
:w
```

7. 卸载磁带。
8. 重复该过程，直到设备映射文件中包含了库为 Oracle HSM 主机分配的所有设备的 Solaris 逻辑设备名称。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/0 -> ../../devices/pci@1f,4000/scsi@2,1/st@2,0:
   1    /dev/rmt/1 -> ../../devices/pci@1f,4000/scsi@4,1/st@5,0:
   3    /dev/rmt/2 -> ../../devices/pci@1f,4000/scsi@4,1/st@6,0:
   4    /dev/rmt/3 -> ../../devices/pci@1f,4000/scsi@4/st@1,0:
:wq
root@solaris:~#
```

9. 保留映射文件。

   配置文件系统时将需要该信息，并且备份最终 Oracle HSM 配置时您可能希望包括该信息。

10. 接下来，转至“配置直接连接的库”。

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. 在文本编辑器中，打开您在“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，将其整理为一个简单表。

   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
LOGICAL DEVICE  DEVICE SERIAL NUMBER  ACSLS DEVICE ADDRESS
--------------  --------------------  --------------------------------
/dev/rmt/0
/dev/rmt/1
/dev/rmt/2
/dev/rmt/3
```

3. 对于 */dev/rmt/* 中所列的每个逻辑设备名称，显示设备序列号。使用命令 *luxadm display /dev/rmt/number*，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:~#
```

4. 在 *device-mappings.txt* 文件的相应行中输入序列号。

   在示例中，在逻辑设备 */dev/rmt/0* 的对应行中输入设备 */dev/rmt/0* 的序列号 *HU92K00200*。

```
root@solaris:~# vi /root/device-mappings.txt
LOGICAL DEVICE  DEVICE SERIAL NUMBER  ACSLS DEVICE ADDRESS
--------------  --------------------  ---------------------------------
/dev/rmt/0      HU92K00200
/dev/rmt/1
/dev/rmt/2
/dev/rmt/3
:wq
root@solaris:~#
```

5. 重复前面两个步骤，直到您为 */dev/rmt/* 中所列的所有逻辑设备标识设备序列号，并将结果记录在 *device-mappings.txt* 文件中。

   在示例中，有四个逻辑设备：

```
root@solaris:~# vi /root/device-mappings.txt
LOGICAL DEVICE  DEVICE SERIAL NUMBER  ACSLS DEVICE ADDRESS
--------------  --------------------  ---------------------------------
/dev/rmt/0      HU92K00200
/dev/rmt/1      HU92K00208
/dev/rmt/2      HU92K00339
/dev/rmt/3      HU92K00289
:w
root@solaris:~#
```

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    12    331000049255
```

```
 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
```

7. 在 *device-mappings.txt* 文件的相应行中记录每个 ACSLS 驱动器地址。保存文件并关闭文本编辑器。

```
root@solaris:~# vi /root/device-mappings.txt
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
```

8. 保留映射文件。

   配置文件系统时将需要该信息，并且备份最终 Oracle HSM 配置时您可能希望包括该信息。
9. 您在配置归档文件系统时需要配置 Oracle StorageTek ACSLS 网络连接库。因此，如果您要规划高可用性文件系统，请转至“为高可用性系统配置存储”。否则，请转至第 4 章 安装 *Oracle HSM and QFS Software* 。

配置直接连接的库

要配置直接连接的磁带库，您必须物理连接硬件并（在某些情况下）配置 SCSI 驱动程序（Oracle HSM 通过通用 *sgen* 驱动程序（而不是发行版 5.4 之前的 SAM-QFS 使用的 *samst* 驱动程序）控制库机械装置）。执行如下操作：

1. 以物理方式将库和驱动器连接到 Oracle HSM 服务器主机。
2. 如果您是第一次安装 Oracle HSM 或是在 Solaris 11 上升级 Oracle HSM 或 SAM-QFS 5.4 配置，请在以物理方式连接硬件后就停止。

   在 Solaris 11 下，*sgen* 是默认的 SCSI 驱动程序，因此 Oracle HSM 安装软件可以自动更新驱动程序别名和配置文件。
3. 如果您要在 Solaris 10 系统上安装 Oracle HSM，则查看是否为 *sgen* 驱动程序分配了以下列表中的驱动程序别名之一。使用命令 *grep scs.*,08 /etc/driver_aliases*。

   可以为 *sgen* 驱动程序分配以下任何别名：
   - *scsa,08.bfcp"* 和/或 *scsa,08.bvhci*

- *scsiclass,08*

在示例中，Solaris 将 *scsiclass,08* 别名用于 *sgen* 驱动程序：

```
root@solaris:~# grep scs.*,08 /etc/driver_aliases
sgen "scsiclass,08"
root@solaris:~#
```

4. 如果 *grep* 命令返回 *sgen "alias"*（其中 *alias* 是上面的列表中的别名），则安装 *sgen* 驱动程序并将其正确地分配给该别名。因此执行如下操作：
   - 如果您要配置高可用性文件系统，请参见“为高可用性系统配置存储”。
   - 否则，请转至第 4 章 安装 *Oracle HSM and QFS Software* 。
5. 如果 *grep* 命令返回 *some-driver "alias"*，其中 *some-driver* 是 *sgen* 之外的某个驱动程序，*alias* 是上述列表中的一个别名，则该别名已分配给其他驱动程序。所以，为 *sgen* 驱动程序创建面向路径的别名。
6. 如果命令 *grep scs.*,08 /etc/driver_aliases* 未返回任何输出，则表明 *sgen* 驱动程序并未安装。因此，请安装该驱动程序。使用命令 *add_drv -i scsiclass,08 sgen*。

   在示例中，*grep* 命令未返回任何输出。因此，我们安装 *sgen* 驱动程序：

```
root@solaris:~# grep scs.*,08 /etc/driver_aliases
root@solaris:~# add_drv -i scsiclass,08 sgen
```

7. 如果命令 *add_drv -i scsiclass,08 sgen* 返回消息 *Driver (sgen) is already installed*，则表明该驱动程序已安装但未连接。因此，请立即连接该驱动程序。使用命令 *update_drv -a -i scsiclass,08 sgen*。

   在示例中，*add_drv* 命令表明该驱动程序已安装。因此，我们附加驱动程序：

```
root@solaris:~# add_drv -i scsiclass,08 sgen
Driver (sgen) is already installed.
root@solaris:~# update_drv -a -i scsiclass,08 sgen
```

8. 如果命令 *grep scs.*,08 /etc/driver_aliases* 显示已将别名 *scsiclass,08* 分配给 *sgen* 驱动程序，则表明已正确配置该驱动程序。

```
root@solaris:~# grep scs.*,08 /etc/driver_aliases
sgen "scsiclass,08"
root@solaris:~#
```

9. 如果您要配置高可用性文件系统，请参见“为高可用性系统配置存储”。
10. 否则，请转至第 4 章 安装 *Oracle HSM and QFS Software* 。

为 sgen 驱动程序创建面向路径的别名

如果所需的 *sgen* 别名已分配给其他驱动程序，您需要创建面向路径的别名来使用 *sgen* 连接指定的库，而不会干扰现有的驱动程序分配。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. 显示系统配置。使用命令 *cfgadm -vl*。

   请注意，*cfgadm* 输出会使用双行标题（每条记录两行）的格式：

```
root@solaris:~# cfgadm -vl
Ap_Id                Receptacle  Occupant     Condition Information  When
Type        Busy  Phys_Id
c3                   connected   configured   unknown   unavailable
scsi-sas    n     /devices/pci@0/pci@0/pci@2/scsi@0:scsi
c5::500104f0008e6d78 connected   configured   unknown   unavailable
med-changer y     /devices/pci@0/.../SUNW,qlc@0,1/fp@0,0:fc::500104f0008e6d78
...
root@solaris:~#
```

3. 在 *cfgadm -vl* 的输出中，找到库的记录。在每个记录第二行的 *Type* 列中查找 *med-changer*。

   在示例中，我们在第二条记录中找到了该库：

```
root@solaris:~# cfgadm -vl
Ap_Id                Receptacle  Occupant     Condition Information  When
Type        Busy  Phys_Id
c3                   connected   configured   unknown   unavailable
scsi-sas    n     /devices/pci@0/pci@0/pci@2/scsi@0:scsi
c5::500104f0008e6d78 connected   configured   unknown   unavailable
med-changer y     /devices/pci@0/.../SUNW,qlc@0,1/fp@0,0:fc::500104f0008e6d78
...
root@solaris:~#
```

4. 获取将作为新的面向路径的别名的物理路径。在 *cfgadm -vl* 的输出中，从 *Phys_Id* 列的条目中，删除子字符串 */devices*。

   在示例中，介质转换器记录的 *Phys_Id* 列包含路径 */devices/pci@0/pci@0/pci@9/SUNW,qlc@0,1/fp@0,0:fc::500104f0008e6d78*，因此，我们选择 */devices/* 后面的部分字符串作为别名（请注意，为适合下面的可用空间，该物理路径已缩写）：

```
root@solaris:~# grep scsiclass,08 /etc/driver_aliases
sdrv "scsiclass,08"
root@solaris:~# cfgadm -vl
Ap_Id                Receptacle  Occupant     Condition Information  When
Type        Busy  Phys_Id
c3                      connected   configured   unknown   unavailable
scsi-sas    n     /devices/pci@0/pci@0/pci@2/scsi@0:scsi
c5::500104f0008e6d78 connected   configured   unknown   unavailable
med-changer y     /devices/pci@0/.../SUNW,qlc@0,1/fp@0,0:fc::500104f0008e6d78
...
root@solaris:~#
```

5. 创建面向路径的别名，并将其分配给 *sgen* 驱动程序。使用命令 *update_drv -d -i '"/path-to-library"' sgen*，其中 *path-to-library* 是您在上一步中标识的路径。

   在示例中，使用库路径创建面向路径的别名 *'"/pci@0/pci@0/pci@9/SUNW,qlc@0,1/fp@0,0:fc::500104f0008e6d78"'*（请注意单引号和双引号）。该命令在一行中，但为了适合页面布局，已处理为两行格式：

```
root@solaris:~# update_drv -d -i / '"/pci@0/pci@0/pci@9/SUNW,qlc@0,1/fp@0,0:fc::500104f0008e6d78"' sgen
root@solaris:~#
```

   现在，库已使用 *sgen* 驱动程序进行配置。
6. 如果您要配置高可用性文件系统，请转至"为高可用性系统配置存储"。
7. 否则，请转至第 4 章 安装 *Oracle HSM and QFS Software* 。

为高可用性系统配置存储

高可用性文件系统需要冗余硬件和多个独立 I/O 路径，从而使文件系统不会因单点硬件故障而变为无法访问。执行以下任务：

- 针对多路径 I/O 配置 Solaris Cluster 节点
- 针对多路径 I/O 配置 Linux 客户机

针对多路径 I/O 配置 Solaris Cluster 节点

要配置高可用性共享文件系统，必须小心地遵循您的 Solaris Cluster 软件版本硬件管理手册中的建议。提供冗余主存储设备和冗余 I/O 路径。

在硬件 RAID 设备上或 Solaris Volume Manager (SVM) 软件 RAID 卷上存储文件系统数据和元数据。将 Oracle HSM 元数据和配置文件放置在 RAID-10 卷组上，或放置在镜像的 SVM 卷上。将文件系统数据放置在硬件控制的 RAID-10 或 RAID-5 卷组上，

或放置在镜像的 SVM 卷上。请注意，当前 Solaris 发行版不再包括 SVM。您必须下载并安装 Solaris 10 9/10 发行版随附的软件版本。

确保存储区域网络连接不会遇到单点故障。在每个群集节点上安装多个主机总线适配器 (host bus adapter, HBA)。使用多个互连和冗余交换机配置存储区域网络 (Storage Area Network, SAN)。使用 Oracle Solaris I/O 多路径软件管理路径故障转移（有关其他详细信息，请参见 Oracle Solaris 客户文档库中的 *Oracle Solaris SAN* 配置和多路径指南和 *stmsboot* 手册页）。

针对多路径 I/O 配置 Linux 客户机

在 Linux 客户机上，使用设备映射器多路径 (Device Mapper Multipath, DMM) 软件包配置路径故障转移的冗余存储设备。DMM 软件管理所有主机总线适配器、电缆、交换机和控制器，它们将主机和存储设备链接作为单个虚拟 I/O 设备，即多路径。

链接主机和存储设备等作为单个虚拟设备的所有 I/O 路径。单独的电缆、交换机和控制器。多路径聚合 I/O 路径，创建由聚合路径构成的新设备。要启用多路径，请执行以下操作：

- 安装设备映射器多路径软件包
- 配置设备映射器多路径软件。

安装设备映射器多路径软件包

遵循下面的说明来配置运行 Oracle Linux 6.*x* 的客户机（对于其他 Linux 版本，请参考供应商的文档）。

1. 以 *root* 用户身份登录 Linux 主机。

```
[root@linux ~]#
```

2. 转至 */etc/yum.repos.d* 子目录并列出目录内容。

```
[root@linux ~]# cd /etc/yum.repos.d
[root@linux ~]# ls -l
total 4
-rw-r--r--. 1 root root 1707 Jun 25  2012 public-yum-ol6.repo
[root@linux ~]#
```

3. 如果 */etc/yum.repos.d* 子目录不包含 *public-yum-ol6.repo* 文件，请使用 *wget* 命令从 Oracle YUM 系统信息库下载一个该文件。

```
[root@linux ~]# wget http://public-yum.oracle.com/public-yum-ol6.repo
-- 2013-02-25 12:50:32 --   http://public-yum.oracle.com/public-yum-ol6.repo
Resolving public-yum.oracle.com... 14 1.146.44.34
Connecting to public-yum.oracle.com|141.146.44.34|:80... connected.
```

```
HTTP request sent, awaiting response... 200 OK
Length: 2411 (2.4K) [text/plain]
Saving to: ”public-yum-ol6.repo”
100%[======================================>] 2,411   -- . - K/s   in 0.001s
2013-02-25 12:50:32 (3.80 MB/s) - ”public-yum-ol6.repo” saved
[2411/2411]
[root@linux ~]#
```

4. 使用文本编辑器，打开 *public-yum-ol6.repo* 文件。确保第一个条目 *[ol6_latest]* 包含行 *enabled=1*。

   在示例中，使用 *vi* 编辑器。所需的行已存在，所以我们关闭该文件：

```
[root@linux ~]# vi public-yum-ol6.repo
[ol6_latest]
name=Oracle Linux $releasever Latest ($basearch)
baseurl=http://public-yum.oracle.com/repo/OracleLinux/OL6/latest/$basearch/
gpgkey=http://public-yum.oracle.com/RPM-GPG-KEY-oracle-ol6
gpgcheck=1
enabled=1
...
:q
[root@linux ~]#
```

5. 查找设备映射器多路径软件包。使用命令 *yum search multipath*。

```
[root@linux ~]# yum search multipath
Loaded plugins: refresh-packagekit, security
========================== N/S Matched: multipath ==========================
device-mapper-multipath.x86_64 : Tools to manage multipath devices using
                               : device-mapper
device-mapper-multipath-libs.x86_64 : The device-mapper-multipath modules and
                                    : shared library
  Name and summary matches only, use "search all" for everything.
[root@linux ~]#
```

6. 安装设备映射器多路径软件。使用命令 *yum install device-mapper-multipath*。当系统提示时，输入 *y*（是）接受列出的软件包及其相关项。

```
[root@linux ~]# yum install device-mapper-multipath
Loaded plugins: refresh-packagekit, security
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package device-mapper-multipath.x86_64 0:0.4.9-56.el6_3.1 will be
```

```
installed
--> Processing Dependency: device-mapper-multipath-libs = 0.4.9-56.el6_3.1
for package: device-mapper-multipath-0.4.9-56.el6_3.1.x86_64
--> Processing Dependency: libmultipath.so()(64bit)
for package: device-mapper-multipath-0.4.9-56.el6_3.1.x86_64
--> Running transaction check
---> Package device-mapper-multipath-libs.x86_64 0:0.4.9-56.el6_3.1 will be
installed
--> Finished Dependency Resolution
Dependencies Resolved
================================================================================
 Package                      Arch   Version           Repository   Size
================================================================================
Installing:
 device-mapper-multipath      x86_64 0.4.9-56.el6_3.1  ol6_latest    96 k
Installing for dependencies:
 device-mapper-multipath-libs x86_64 0.4.9-56.el6_3.1  ol6_latest   158 k
Transaction Summary
================================================================================
Install       2 Package(s)
Total download size: 254 k
Installed size: 576 k
Is this ok [y/N]: y
Downloading Packages:
(1/2): device-mapper-multipath-0.4.9-56.el6_3.1.x86_64.r |  96 kB     00:00
(2/2): device-map
per-multipath-libs-0.4.9-56.el6_3.1.x86 | 158 kB     00:00
--------------------------------------------------------------------------------
Total                                           104 kB/s | 254 kB     00:02
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : device-mapper-multipath-libs-0.4.9-56.el6_3.1.x86_64     1/2
  Installing : device-mapper-multipath-0.4.9-56.el6_3.1.x86_64          2/2
  Verifying  : device-mapper-multipath-0.4.9-56.el6_3.1.x86_64          1/2
  Verifying  : device-mapper-multipath-libs-0.4.9-56.el6_3.1.x86_64     2/2
Installed:
  device-mapper-multipath.x86_64 0:0.4.9-56.el6_3.1
Dependency Installed:
  device-mapper-multipath-libs.x86_64 0:0.4.9-56.el6_3.1
Complete!
[root@linux ~]#
```

7. 启动多路径守护进程。使用命令 *chkconfig multipathd on*。

```
[root@linux ~]# chkconfig multipathd on
[root@linux ~]#
```

8. 接下来，配置设备映射器多路径软件

配置设备映射器多路径软件

通过编辑 */etc/multipath.conf* 文件来配置设备映射器多路径软件。该文件包含一系列部分，每部分包含一组相关属性、值和子部分：

- *default* 部分配置多路径软件自身。它指定记录的详细级别、定义故障转移行为以及指定所需的操作系统命令和目录的位置。
- *blacklist* 部分标识您需要从多路径配置中排除的设备，例如本地系统磁盘。可以通过以下方式标识设备：通过全局名称/全局标识符 (World Wide Name/World Wide Identifier, WWN/WID)，或者通过指定设备节点名称或供应商和产品设备字符串的正则表达式。
- *blacklist_exceptions* 部分允许您在多路径配置中特别包括某些设备，否则 *blacklist* 部分中的常规规则会排除这些设备。
- *multipaths* 部分允许您定义一个或多个 *multipath* 子部分，每个子部分对您通过全局名称指定的多路径应用特殊配置。
- *devices* 部分允许您定义一个或多个 *device* 子部分，每个子部分对设备应用特殊多路径配置。

有关各个默认值的详细说明，请参见带注释的样例文件 */usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf.annotated*。

blacklist_exceptions 列出应该使用的所有设备（即使黑名单标识了这些设备）。defaults：设备映射器多路径 (DM-Multipath) 的常规默认设置。multipaths：单个多路径设备的特征设置。这些值将覆盖配置文件的 defaults 和 devices 部分中指定的内容。devices：单个存储控制器的设置。这些值将覆盖配置文件的 defaults 节中指定的内容。如果您使用的是默认情况下不支持的存储阵列，则可能需要为阵列创建 devices 子部分。系统确定多路径设备的属性时，它首先检查多路径设置，然后检查每个设备的设置，再然后检查多路径系统默认值。

4

第 4 章 安装 Oracle HSM and QFS Software

Oracle HSM 使用在 Oracle Solaris 11 中成为标准配置的映像包管理系统 (Image Packaging System, IPS)。IPS 是以网络为中心的软件包管理系统，可简化和协调软件包的安装、升级和删除。IPS 可显著简化修补程序管理并便于部署到生产环境。

使用 Solaris Package Manager图形桌面应用程序或 IPS 终端命令，管理员可以访问 Oracle Solaris 软件系统信息库，并找到、下载和安装所需的软件包，而 IPS 会自动处理依赖性检查和软件包验证。IPS 会更改系统快照，以便新软件可以在维护期间无干扰地部署。因此可以在必要时回退更改。这样，安装和更新便可安全地应用于运行中的生产系统。

要安装 Oracle HSM 软件，请执行以下任务：

- 获取软件
- 安装 Solaris Cluster 软件（仅适用于高可用性配置）
- 升级 Oracle HSM 共享文件系统（如果适用）
- 在主机上安装、升级或降级 Oracle HSM 软件

本章最后简要介绍了卸载 Oracle HSM 软件。

获取软件

本节概述了获取所需的安装软件和软件更新的过程。请参见以下各节：

- 查看安装要求
- 下载软件安装包。

查看安装要求

有关安装要求的最新信息（包括 Oracle Solaris 和 Linux 操作系统、Oracle Cluster 软件和其他要求或支持的软件包的受支持版本），请参见 Oracle HSM 发行说明、Oracle 支持服务 (*support.oracle.com*) 以及 Oracle HSM wiki 页面 (*wikis.oracle.com/display/hsmqfs/Home*)。

下载软件安装包

从 Oracle Software Delivery Cloud 下载 Oracle 软件产品的安装包。所有 Oracle 产品的基本过程都相似。

要下载 Oracle HSM 发行版 6.1 软件包，请执行如下操作：

1. 在 Web 浏览器窗口中打开 *edelivery.oracle.com*。
2. 如果您尚未使用过该站点，请先注册。
3. 使用您的注册凭证进行注册。
4. 选中确认适当软件许可证的复选框。
5. 选中同意软件适用的出口限制的复选框。
6. 在 "Media Pack Search"（介质包搜索）页面中，从 "Select a Product Pack"（选择产品程序包）控件中的列表中选择 "Oracle StorageTek Products"（Oracle StorageTek 产品）。
7. 从 "Platform"（平台）列表中，选择将托管 Oracle HSM 软件的平台体系结构上的 Oracle Solaris。
8. 按 "Go"（查找）按钮。
9. 显示结果列表时，单击与 Oracle Hierarchical Storage Manager 介质包对应的单选按钮，然后按 "Continue"（继续）。
10. 显示 "StorageTek Storage Archive Manager and StorageTek QFS Software Media Pack for Oracle Solaris"（适用于 Oracle Solaris 的 Oracle Hierarchical Storage Manager and StorageTek QFS Software 介质包）页面时，按 "Readme"（自述文件）按钮并阅读下载说明。
11. 仍在 "StorageTek Storage Archive Manager and StorageTek QFS Software Media Pack for Oracle Solaris"（适用于 Oracle Solaris 的 Oracle Hierarchical Storage Manager and StorageTek QFS Software 介质包）页面中时，按 "View Digest"（查看摘要）按钮并保存摘要值。

    摘要是由加密散列函数创建的校验和。通过比较发布的摘要和从下载的文件本地计算的摘要，您可以确保下载的文件是完整无缺的。有关从文件计算检验和的说明，请参见 Solaris *dgst* 和 *md5* 手册页。
12. 仍在 "StorageTek Storage Archive Manager and StorageTek QFS Software Media Pack for Oracle Solaris"（适用于 Oracle Solaris 的 Oracle Hierarchical Storage Manager and StorageTek QFS Software 介质包）页面中时，按与您已许可的产品相对应的 "Download"（下载）按钮。

    列表包含 Oracle Hierarchical Storage Manager and StorageTek QFS Software 的单独条目。Oracle Hierarchical Storage Manager 介质包包含归档和文件系统软件。Oracle StorageTek QFS Software 介质包仅包含文件系统软件。
13. 出现提示时，将 ZIP 归档保存到本地目录中，如 "Readme"（自述文件）页面所述。

    选择的目录应该可以从所有 Oracle HSM 主机上通过本地网络进行访问。在本章中的示例中，我们将文件下载到名为 *sw_install* 的网络文件服务器上的 */hsmqfs* 目录中。
14. 如果您多次尝试都未能下载所需文件，请联系软件交付客户服务中心 (*edelivery_ww@oracle.com*) 寻求帮助。

15. 下载 ZIP 文件后，将其解压缩到本地目录中。

    在本示例中，我们在 */hsmqfs* 子目录中解压缩 Oracle Hierarchical Storage Manager and StorageTek QFS Software 文件 *Q12345-01.zip*，然后列出内容：

```
[sw_install]root@solaris:~# cd /hsmqfs
[sw_install]root@solaris:~# unzip Q12345-01.zip
[sw_install]root@solaris:~# ls Q12345-01/
./       COPYRIGHT.txt       linux.iso             README.txt
../      iso.md5             Oracle-HSM_6.0/
[sw_install]root@solaris:~# ls Oracle-HSM_6.0/
total 42
./        COPYRIGHT.txt      linux1/       solaris_sparc/../       README.txt         linux2/
 solaris_x64/
```

16. 如果您正在准备高可用性文件系统，请转至"安装 Solaris Cluster 软件（仅适用于高可用性配置）"。
17. 如果您要升级多主机共享文件系统，请转至"升级 Oracle HSM 共享文件系统"。
18. 否则，请直接转至“在主机上安装、升级或降级 Oracle HSM 软件”。

安装 Solaris Cluster 软件（仅适用于高可用性配置）

如果您要准备高可用性 Oracle HSM 配置，请执行如下操作：

1. 按照 Solaris Cluster 软件的联机信息库中的安装和数据服务管理文档所述，在每台主机上，安装 Oracle Solaris Cluster 和 *SUNW.HAStoragePlus* 数据服务软件。
2. 然后转至“在主机上安装、升级或降级 Oracle HSM 软件”。

升级 Oracle HSM 共享文件系统

如果您要针对需要在升级过程中保持可用的共享文件系统升级软件，则考虑滚动升级。配置一个或多个潜在元数据服务器后，除活动服务器之外，您可以更新非活动服务器，激活更新的服务器，然后在升级其余的潜在元数据服务器和客户机之前配置并重新激活主服务器。该滚动升级过程会使活动 Oracle HSM 元数据服务器在所有时间都保持可用，从而使客户机仍可以访问文件系统数据。

要执行滚动升级，请执行以下任务：

- 升级 Oracle HSM 的任何过早发行版
- 执行滚动升级

升级 Oracle HSM 的任何过早发行版

在任何给定时刻，共享文件系统上元数据服务器和客户机的 Oracle HSM 软件最多只能相差一个发行版。如果您的共享文件系统配置中包括运行的 Oracle HSM（或 SAM-

QFS）软件比目标升级发行版早一个发行版以上的主机，则在执行更正操作之前，您将无法升级至所需发行版。

执行如下操作：

1. 如果任何客户机主机与元数据服务器运行的 Oracle HSM（或 SAM-QFS）软件发行版不同，请将它们升级至服务器所使用的发行版，然后再继续。
2. 如果活动元数据服务器上的 Oracle HSM（或 SAM-QFS）软件比目标升级发行版早一个发行版以上，且在升级期间文件系统需要保持挂载，请重复执行滚动升级（每次一个发行版级别），直到所有主机都为最新。
3. 如果活动元数据服务器上的 Oracle HSM（或 SAM-QFS）软件比目标升级发行版早一个发行版以上，但在升级期间文件系统不需要保持挂载，请勿尝试进行滚动升级。停止归档和回写进程、卸载文件系统、并分别升级每台主机，如“在主机上安装、升级或降级 Oracle HSM 软件”中所述。

执行滚动升级

1. 确保升级 Oracle HSM 的任何过早发行版，然后再继续！

   在尝试进行滚动升级时，如果任何主机比目标升级发行版早一个发行版以上，升级将会失败，最多会使文件系统处于不一致的状态。
2. 以 *root* 用户身份登录到当前处于活动状态的（第一个）元数据服务器。然后，同样以 *root* 用户身份登录到当前潜在的（第二个）元数据服务器。

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

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

[first-mds]root@solaris:~# ssh root@second-mds
Password:
[second-mds]root@solaris:~#
```

3. 升级当前非活动的第二个元数据服务器。使用“在主机上安装、升级或降级 Oracle HSM 软件”中的过程安装更新的 Oracle HSM 软件。
4. 升级步骤完成后，准备激活第二个服务器。如果第一个活动元数据服务器挂载了 Oracle HSM 或 SAM-QFS 归档文件系统，请停止任何新的归档和回写活动，使介质驱动器空闲并等待当前作业完成。然后停止库控制守护进程。

   有关如何停止归档活动的完整描述，请参见《Oracle Hierarchical Storage Manager and StorageTek QFS Software 维护和管理指南》。

```
[first-mds]root@solaris:~# samcmd aridle
[first-mds]root@solaris:~# samcmd stidle
[first-mds]root@solaris:~# samcmd 901 idle
```

```
...
[first-mds]root@solaris:~# samcmd a
 ...
Waiting for :arrun
[first-mds]root@solaris:~# samcmd r
...
ty   eq status      act  use  state  vsn
li  801  ---------p    0   0%  off
         empty
...
[first-mds]root@solaris:~# samd stop
[first-mds]root@solaris:~#
```

5. 在第二个元数据服务器上，装入 Oracle HSM 配置文件并启动 Oracle HSM 进程。使用命令 *samd config*。

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

6. 在第二个元数据服务器上，挂载 Oracle HSM 文件系统。

```
[second-mds]root@solaris:~# mount sharefs1
[second-mds]root@solaris:~#
```

7. 激活新更新的第二个元数据服务器。从第二个元数据服务器上发出命令 *samsharefs -s server file-system*，其中 *server* 是新更新的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，潜在元数据服务器为 *second-mds*，文件系统名称为 *sharefs1*：

```
[second-mds]root@solaris:~# samsharefs -s second-mds sharefs1
[second-mds]root@solaris:~#
```

8. 升级第一个现在非活动元数据服务器。使用“在主机上安装、升级或降级 Oracle HSM 软件”中的过程安装更新的 Oracle HSM 软件。
9. 升级步骤完成后，准备重新激活第一个元数据服务器。如果当前第二个活动元数据服务器挂载了 Oracle HSM 归档文件系统，请停止任何新的归档和回写活动，使介质驱动器空闲并等待当前作业完成。然后停止库控制守护进程。

```
[second-mds]root@solaris:~# samcmd aridle
[second-mds]root@solaris:~# samcmd stidle
...
[second-mds]root@solaris:~# samd stop
[second-mds]root@solaris:~#
```

10. 在第一个元数据服务器上，装入 Oracle HSM 配置文件并启动 Oracle HSM 进程。使用命令 *samd config*。

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

11. 在第一个元数据服务器上，挂载 Oracle HSM 文件系统。

```
[first-mds]root@solaris:~# mount sharefs1
[first-mds]root@solaris:~#
```

12. 重新激活第一个元数据服务器。从第一个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，潜在元数据服务器为 *first-mds*，文件系统名称为 *sharefs1*：

```
[first-mds]root@solaris:~# samsharefs -s first-mds sharefs1
[first-mds]root@solaris:~#
```

13. 更新其余的客户机。使用“在主机上安装、升级或降级 Oracle HSM 软件”中的过程安装更新的 Oracle HSM 软件。
14. 在此处停止。升级已完成。

在主机上安装、升级或降级 Oracle HSM 软件

要在单个主机上安装、升级或降级 Oracle HSM 软件，请执行以下任务：

- 在 Oracle Solaris 主机上安装、升级或降级 Oracle HSM 软件。
- 在 Linux 主机上安装或更新 Oracle HSM 客户机软件（如果有）。

在 Oracle Solaris 主机上安装、升级或降级 Oracle HSM 软件

要在 Solaris 主机上安装、升级或降级 Oracle HSM 软件包，请首先执行以下任务：

- 针对软件更改准备主机。
- 找到您的主机体系结构的软件包。

然后执行最适合您的情况的安装任务：

- 如果您正在安装新软件并且主机操作系统是 Solaris 11 或更高版本，则使用 Solaris 映像包管理系统 (Image Packaging System, IPS) 命令 *pkg install*。
- 如果您正在升级或降级使用 IPS 命令 *pkg install* 安装的软件，则使用映像包管理系统 (Image Packaging System, IPS) 命令 *pkg update*。

- 如果您正在 Solaris 10 主机上安装新软件，则使用 SVR4 *pkgrm* 和 *pkgadd* 命令。
- 如果您正在升级使用 SVR4 命令 *pkgadd* 安装的软件，则使用 SVR4 *pkgrm* 和 *pkgadd* 命令。

针对软件更改准备主机

1. 如果当前未在主机系统上安装 Oracle HSM 软件，则转至"找到您的主机体系结构的软件包"。
2. 否则，以 *root* 用户身份登录到 Oracle HSM 服务器。

```
[samqfs1host]root@solaris:~#
```

3. 如果当前已在主机系统上安装 Oracle HSM 软件，则闲置所有归档进程。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：

```
[samqfs1host]root@solaris:~# samcmd aridle
[samqfs1host]root@solaris:~#
```

4. 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：

```
[samqfs1host]root@solaris:~# samcmd stidle
[samqfs1host]root@solaris:~#
```

5. 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：

```
[samqfs1host]root@solaris:~# samcmd a
Archiver status samcmd     6.0 10:20:34 Feb 20 2015
samcmd on samqfs1host
sam-archiverd:  Waiting for :arrun
sam-arfind: ...
Waiting for :arrun
```

6. 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：

```
[samqfs1host]root@solaris:~# samcmd u
Staging queue samcmd      6.0  10:20:34 Feb 20 2015
samcmd on solaris.demo.lan
```

```
Staging queue by media type: all
sam-stagerd:  Waiting for :strun
[samqfs1host]root@solaris:~#
```

7. 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：

```
[samqfs1host]root@solaris:~# samcmd 801 idle
[samqfs1host]root@solaris:~# samcmd 802 idle
[samqfs1host]root@solaris:~# samcmd 803 idle
[samqfs1host]root@solaris:~# samcmd 804 idle
[samqfs1host]root@solaris:~#
```

8. 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。

```
[samqfs1host]root@solaris:~# samcmd r
Removable media samcmd     6.0 10:37:09 Feb 20 2014
samcmd on samqfs1host
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
[samqfs1host]root@solaris:~#
```

9. 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*。

```
[samqfs1host]root@solaris:~# samd stop
[samqfs1host]root@solaris:~#
```

10. 如果文件系统通过 NFS 或 SMB/CIFS 进行共享，请取消共享文件系统。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。

在第一个示例中，停止 Oracle HSM 独立文件系统 *samqfs1* 的 NFS 共享。

```
[samqfs1host]root@solaris:~# unshare /hsmqfs1
[samqfs1host]root@solaris:~#
```

在第二个示例中，停止 Oracle HSM 共享文件系统 *samqfs2* 的 NFS 共享：

```
[samqfs2server]root@solaris:~# unshare /hsmqfs2
[samqfs2server]root@solaris:~#
```

11. 卸载所有 Oracle HSM 文件系统。

    在第一个示例中，卸载非共享的独立文件系统 *samqfs1*：

```
[samqfs1host]root@solaris:~# umount samqfs1
```

    在第二个示例中，首先从客户机卸载共享文件系统 *samqfs1*（允许客户机在 *60* 秒内进行卸载），然后从服务器卸载该系统。

```
[samqfs2server]root@solaris:~# ssh root@samqfs2client1
Password:
[samqfs2client1]root@solaris:~# umount /hsmqfs2
[samqfs2client1]root@solaris:~# exit
[samqfs2server]root@solaris:~#

[samqfs2server]root@solaris:~# ssh root@samqfs2client1
Password:
[samqfs2client2]root@solaris:~# umount /hsmqfs2
[samqfs2client2]root@solaris:~# exit
[samqfs2server]root@solaris:~# umount -o await_clients=60 /sharefs2
```

12. 如果您当前安装了 SAM-QFS 5.3 或更低版本，请卸载所有软件包。使用命令 *pkgrm SUNWsamfsu SUNWsamfsr*（如果只安装了 QFS，则使用命令 *pkgrm SUNWqfsu SUNWqfsr*）。

    按照指定顺序删除软件包：从 *SUNWsamfsu* 开始，到 *SUNWsamfsr* 结束。在示例中，将回复 *yes* 通过管道传输到该命令，以便自动回答所有问题：

```
[host1]root@solaris:~# yes | pkgrm SUNWsamfsu SUNWsamfsr
```

13. 接下来，找到您的主机体系结构 Oracle HSM 软件包。

找到您的主机体系结构的软件包

1. 以 *root* 用户身份登录 Oracle HSM 主机。

```
root@solaris:~#
```

2. 转至解压缩 Oracle HSM 下载文件的目录，并找到存储所需版本的软件包的子目录。

   初始发行的软件包存储在 *Oracle_HSM_release-number*（或 *STK_QFS_release-number*）子目录中，其中 *release-number* 是主要发行版本号和次要发行版本号（由点进行连接）：*Oracle_HSM_6.0+/*。修补程序发行版（如果有）位于类似的子目录中，但该子目录具有附加的 *-patch-number* 后缀，其中 *patch-number* 是两位数的修补程序序列号：*Oracle_HSM_6.0-01/*。

   在本示例中，我们转至软件初始发行版的下载目录 *Oracle_HSM_6.0/* 并列出内容：

```
root@solaris:~# cd /net/sw-install/hsmqfs/Oracle_HSM_6.0/
root@solaris:~# ls -1
./
../
linux1/
linux2/
Notices/
README.txt
solaris_sparc/
solaris_x64/
```

3. 转至与您的主机体系结构相对应的子目录 *solaris_sparc/* 或 *solaris_x64/* 并列出内容。

   在示例中，我们转至 *solaris_sparc/* 子目录：

```
root@solaris:~# cd solaris_sparc/
root@solaris:~# ls -1
./
../
S10/
S11/
S11_ips/
fsmgr_6.1.zip
fsmgr_setup*
```

4. 在主机上安装 Solaris 11 或更高版本后，您可以使用映像包管理系统安装、升级或降级软件。访问以下内容之一：
   - "使用映像包管理系统 (Image Packaging System, IPS) 安装软件".
   - “使用映像包管理系统 (Image Packaging System, IPS) 升级或降级软件”.
5. 在主机上安装 Solaris 11 或更高版本后，您还可以选择使用 *pkgadd* 方法安装、升级或降级软件。请参见“使用 SVR4 *pkgrm* 和 *pkgadd* 命令升级或降级软件”。
6. 在主机上安装 Solaris 10 后，您可以使用 *pkgadd* 方法安装、升级或降级软件。转至“使用 SVR4 *pkgrm* 和 *pkgadd* 命令升级或降级软件”。

使用映像包管理系统 (Image Packaging System, IPS) 安装软件

通常，您应使用映像包管理系统 (Image Packaging System, IPS) 命令在运行 Solaris 11 或更高版本的主机上安装、升级或降级 Oracle HSM 软件。针对每台主机（包括元数据服务器和共享文件系统客户机 [如果有]），请执行如下操作：

1. 如果您尚未执行此操作，则找到您的主机体系结构的 Oracle HSM 软件包。
2. 转至 Solaris 11 IPS 软件包的系统信息库目录 *repo.samqfs/*。

   在本示例中，我们转至 Oracle HSM 6.0 的系统信息库目录 *Oracle_HSM_6.0/solaris_sparc/S11_ips/repo.samqfs*：

```
root@solaris:~# cd repo.samqfs/
root@solaris:~#
```

3. 要安装 Oracle Hierarchical Storage Manager and StorageTek QFS Software 软件包，请使用命令 *pkg install -g .--accept SUNWsamfs SUNWsamqassy*，其中 *.* 是当前目录（系统信息库），*SUNWsamfs* 和 *SUNWsamqassy* 是 Oracle HSM 映像包管理系统软件包名称。

```
root@solaris:~# pkg install -g . --accept SUNWsamfs SUNWsamqassy
Creating plan
...
* The licence and distribution terms for any publically available version or
 * derivative of this code cannot be changed.  i.e. this code cannot simply be
 * copied and put under another distribution licence
 * [including the GNU Public Licence.]
 */
           Packages to install:   2
       Create boot environment:  No
Create backup boot environment: Yes
DOWNLOAD                                  PKGS         FILES    XFER (MB)   SPEED
Completed                                  2/2       520/520    21.4/21.4    0B/s
PHASE                                        ITEMS
Installing new actions                   693/693
Updating package state database            Done
```

```
Updating image state                            Done
Creating fast lookup database                   Done
```

4. 要仅安装 QFS Software 软件包，请使用命令 *pkg install -g .--accept SUNWqfs SUNWsamqassy*，其中 *.* 是当前目录（系统信息库），*SUNWqfs* 和 *SUNWsamqassy* 是 Oracle HSM 映像包管理系统软件包名称。

```
root@solaris:~# pkg install -g . --accept SUNWqfs SUNWsamqassy
Creating plan
...
* The licence and distribution terms for any publically available version or
 * derivative of this code cannot be changed.  i.e. this code cannot simply be
 * copied and put under another distribution licence
 * [including the GNU Public Licence.]
 */
           Packages to install:   2
       Create boot environment:  No
Create backup boot environment: Yes
DOWNLOAD                                  PKGS         FILES    XFER (MB)   SPEED
Completed                                  2/2       520/520    21.4/21.4    0B/s
PHASE                                        ITEMS
Installing new actions                     693/693
Updating package state database              Done
Updating image state                         Done
Creating fast lookup database                Done
```

5. 在软件包完成安装后，运行安装后脚本 *sam-qfs-post-install*。它位于 Oracle HSM 安装目录（*/opt/SUNWsamfs/* 或 */opt/SUNWqfs/*） 的 *util/* 子目录中。

   在示例中，运行 */opt/SUNWsamfs/util/sam-qfs-post-install*：

```
root@solaris:~# /opt/SUNWsamfs/util/sam-qfs-post-install
SUNWsamfs IPS package installed.

inquiry.conf may have been updated for this release.
...
root@solaris:~#
```

6. 将 Oracle HSM 目录 */opt/SUNWsamfs/bin* 和 */opt/SUNWsamfs/sbin*（或 */opt/SUNWqfs/bin* 和 */opt/SUNWqfs/sbin*）添加到系统 *PATH* 变量（如果还不在路径中）。
7. 将 Oracle HSM 目录 */opt/SUNWsamfs/man*（或 */opt/SUNWqfs/man*）添加到系统 *MANPATH* 变量（如果还不在 man 路径中）。

8. 如果计划的 Oracle HSM 配置包括其他 Solaris 主机，请从头开始重复该过程，直到将该软件安装在所有主机上。
9. 如果计划的 Oracle HSM 配置包括 Linux 主机作为共享文件系统客户机，请转至“在 Linux 主机上安装或更新 Oracle HSM 客户机软件”。
10. 否则，请转至第 5 章 使用 ***samsetup*** 配置向导或第 6 章 配置基本文件系统 。

使用映像包管理系统 (Image Packaging System, IPS) 升级或降级软件

使用映像包管理系统 (Image Packaging System, IPS) 命令升级或降级最初使用 IPS 安装的 Oracle HSM 软件。

针对每台主机（包括元数据服务器和共享文件系统客户机 [如果有]），请执行如下操作：

1. 如果您尚未执行此操作，则找到您的主机体系结构的 Oracle HSM 软件包。
2. 要将 Oracle Hierarchical Storage Manager and StorageTek QFS Software 软件包升级至系统信息库中的最新版本，请使用命令 *`pkg update -g . --accept SUNWsamfs SUNWsamqassy`*，其中 *`.`* 是当前目录（系统信息库），*`SUNWsamfs`* 和 *`SUNWsamqassy`* 是 Oracle HSM 映像包管理系统软件包名称。

```
root@solaris:~# pkg update -g . --accept SUNWsamfs SUNWsamqassy
...
root@solaris:~#
```

3. 要仅将 QFS Software 软件包升级至系统信息库中的最新版本，请使用命令 *`pkg update -g . --accept SUNWqfs SUNWsamqassy`*，其中 *`.`* 是当前目录（系统信息库），*`SUNWqfs`* 和 *`SUNWsamqassy`* 是 Oracle HSM 映像包管理系统软件包名称。

```
[host1]root@solaris:~# pkg update -g . --accept SUNWqfs SUNWsamqassy
...
root@solaris:~#
```

4. 要降级 Oracle HSM 软件包或将其升级至指定的版本，请首先获取所需软件包的故障管理资源标识符 (fault managed resource identifier, FMRI)。使用命令 *`pkg info -r -g . package-name`*，其中 *`.`* 指定当前目录，*`package-name`* 是 Oracle HSM 软件包的名称。

   在本示例中，主机上安装了 Oracle HSM 版本 6.0.0：

```
root@solaris:~# samcmd l
Usage information samcmd      6.0.0 14:06:20 Feb  20 2015 ...
root@solaris:~#
```

我们需要降级至 SAM-QFS 5.4.6。因此我们针对版本 5.4.6 的 IPS 系统信息库 *Oracle_HSM_6.0/solaris_sparc/S11_ips/repo.samqfs* 中的 *SUNWsamfs* 和 *SUNWsamqassy* 运行 *pkg info* 命令：

```
root@solaris:~# pwd
/net/Oracle_HSM_6.0/solaris_sparc/S11_ips/repo.samqfs
root@solaris:~# pkg info -r -g . SUNWsamfs
          Name: SUNWsamfs
       Summary: StorageTek SAM and StorageTek SAM-QFS software
   Description: StorageTek Storage and Archive Manager File System
      Category: System/File System
         State: Not installed
     Publisher: samqfs
       Version: 5.4
 Build Release: 5.11
        Branch: None
Packaging Date: Tue Jul 08 22:56:56 2014
          Size: 88.64 MB
          FMRI: pkg://hsmqfs/SUNWsamfs@5.4,5.11:20140708T225656Z

root@solaris:~# pkg info -r -g . SUNWsamqassy
          Name: SUNWsamqassy
       Summary: StorageTek QFS and Storage Archive Manager SAM-QFS IPS assembly
 services
   Description: SAM-QFS IPS Assembly Services
      Category: System/File System
         State: Installed
     Publisher: samqfs
       Version: 5.4
 Build Release: 5.11
        Branch: None
Packaging Date: Fri Sep 26 17:21:35 2014
          Size: 15.15 kB
          FMRI: pkg://hsmqfs/SUNWsamqassy@5.4,5.11:20140926T172135Z
root@solaris:~#
```

5. 然后，要降级 Oracle HSM 软件包或将其升级至指定的版本，请运行命令 *pkg update -g . fmri*，其中 *.* 指定当前目录，*fmri* 指定所需软件版本的故障管理资源标识符。

   在本示例中，我们指定 5.4.6 版本的 *SUNWsamfs* 和 *SUNWsamqassy* 软件包的 FMRI：

```
root@solaris:~# pkg update -g . SUNWsamfs@5.4,5.11:20140708T225656Z
```

```
           Packages to update:   1
       Create boot environment:  No
Create backup boot environment: Yes
DOWNLOAD                                PKGS         FILES    XFER (MB)
 SPEEDCompleted                            1/1        160/160    19.2/19.2  3.4M/s
PHASE                                      ITEMS
Updating modified actions                172/172
Updating package state database             Done
Updating package cache                       1/1
Updating image state                        Done
Creating fast lookup database               Done
Updating package cache                       3/3
root@solaris:~# pkg update -g . SUNWsamqassy@5.4,5.11:20140926T172135Z
...
root@solaris:~#
```

6. *pkg update* 命令完成后，重新启动系统。使用 Solaris *reboot* 命令。

   ```
   root@solaris:~# reboot
   ```

7. 如果计划的 Oracle HSM 配置包括其他 Solaris 主机，请从头开始重复该过程，直到在所有主机上更新或降级该软件。
8. 如果计划的 Oracle HSM 配置包括 Linux 主机作为共享文件系统客户机，请转至“在 Linux 主机上安装或更新 Oracle HSM 客户机软件”。

使用 SVR4 *pkgrm* 和 *pkgadd* 命令安装软件

当您要在运行 Solaris 10 的主机上安装 Oracle HSM 软件和您要升级最初使用 SVR4 命令安装的软件时，请使用 SVR4 软件包命令。

针对每台 Oracle HSM Solaris 主机（包括元数据服务器和共享文件系统客户机 [如果有]），请执行如下操作：

1. 如果您尚未执行此操作，则找到您的主机体系结构的 Oracle HSM 软件包。
2. 要同时安装 Oracle Hierarchical Storage Manager and StorageTek QFS Software 软件包，请使用命令 *pkgadd -d . SUNWsamfsr SUNWsamfsu* 并接受所有默认值。

   请注意，您必须安装 *SUNWsamfsr* 软件包，才能安装 *SUNWsamfsu* 软件包。在本示例中，我们确保位于我们操作系统的目录 *Oracle_HSM_6.0/solaris_sparc/S10* 中。然后，将回复 *yes* 通过管道传输到该命令，以便自动回答所有问题：

```
root@solaris:~# pwd
/net/Oracle_HSM_6.0/solaris_sparc/s10
root@solaris:~# yes | pkgadd -d . SUNWsamfsr SUNWsamfsu
```

3. 要仅安装 QFS Software 软件包，请使用命令 *pkgadd -d . SUNWqfsr SUNWqfsu* 并接受所有默认值。

   请注意，您必须先安装 *SUNWqfsr* 软件包，才能安装 *SUNWqfsu* 软件包。在示例中，将回复 *yes* 通过管道传输到该命令，以便自动回答所有问题：

```
root@solaris:~# yes | pkgadd -d . SUNWqfsr SUNWqfsu
```

4. 如果计划的 Oracle HSM 配置包括 Linux 主机作为共享文件系统客户机，请转至“在 Linux 主机上安装或更新 Oracle HSM 客户机软件”。
5. 否则，请转至第 5 章 使用 *samsetup* 配置向导或第 6 章 配置基本文件系统 。

使用 SVR4 *pkgrm* 和 *pkgadd* 命令升级或降级软件

当您要在运行 Solaris 10 的主机上升级或降级 Oracle HSM 软件和您要升级或降级最初使用 SVR4 命令安装的软件时，请使用 SVR4 软件包命令。

针对每台 Oracle HSM Solaris 主机（包括元数据服务器和共享文件系统客户机 [如果有]），请执行如下操作：

1. 如果要将 Oracle HSM 软件降级至 SAM-QFS 5.3，请首先将配置文件恢复至较旧的软件指定的位置。使用命令 */opt/SUNWsamfs/sbin/backto 5.3*。

   backto 命令会将文件恢复到其之前的位置和格式。有关更多信息，请参见 *backto* 手册页。

   在本示例中，我们转换 Oracle HSM 6.0 配置文件以便与 Oracle SAM 5.3 配合使用：

```
root@solaris:~# /opt/SUNWsamfs/sbin/backto 5.3 ...
root@solaris:~#
```

2. 卸载当前安装的所有 Oracle HSM 软件包。使用命令 *pkgrm SUNWsamfsu SUNWsamfsr*（如果只安装了 QFS，则使用命令 *pkgrm SUNWqfsu SUNWqfsr*）。

   按照指定顺序删除软件包：从 *SUNWsamfsu* 开始，到 *SUNWsamfsr* 结束。在示例中，将回复 *yes* 通过管道传输到该命令，以便自动回答所有问题：

```
root@solaris:~# yes | pkgrm SUNWsamfsu SUNWsamfsr
```

3. 如果您尚未执行此操作，则找到您的主机体系结构的 Oracle HSM 软件包。
4. 要同时安装 Oracle Hierarchical Storage Manager and StorageTek QFS Software 软件包，请使用命令 *pkgadd -d . SUNWsamfsr SUNWsamfsu* 并接受所有默认值。

请注意，您必须安装 *SUNWsamfsr* 软件包，才能安装 *SUNWsamfsu* 软件包。在本示例中，我们确保位于我们操作系统的正确目录 *Oracle_HSM_6.0/solaris_sparc/S10* 中。然后，将回复 *yes* 通过管道传输到该命令，以便自动回答所有问题：

```
root@solaris:~# pwd
/net/Oracle_HSM_6.0/solaris_sparc/s10
root@solaris:~# yes | pkgadd -d . SUNWsamfsr SUNWsamfsu
```

5. 要仅安装 QFS Software 软件包，请使用命令 *pkgadd -d . SUNWqfsr SUNWqfsu* 并接受所有默认值。

   请注意，您必须先安装 *SUNWqfsr* 软件包，才能安装 *SUNWqfsu* 软件包。在示例中，将回复 *yes* 通过管道传输到该命令，以便自动回答所有问题：

```
root@solaris:~# pwd
/net/Oracle_HSM_6.0/solaris_sparc/s10
root@solaris:~# yes | pkgadd -d . SUNWqfsr SUNWqfsu
```

6. 如果计划的 Oracle HSM 配置包括 Linux 主机作为共享文件系统客户机，请转至"在 Linux 主机上安装或更新 Oracle HSM 客户机软件"。
7. 否则，请转至第 5 章 使用 ***samsetup*** 配置向导或第 6 章 配置基本文件系统 。

在 Linux 主机上安装或更新 Oracle HSM 客户机软件

针对 Oracle HSM 共享文件系统的每台 Linux 客户机，请执行如下操作：

1. 以 *root* 用户身份登录 Linux 客户机。

```
[root@linux ~]#
```

2. 卸载所有挂载的 Oracle HSM 文件系统。
3. 卸载旧版本的 Oracle HSM 软件包。运行脚本 */var/opt/SUNWsamfs/Uninstall*：

```
[root@linux ~]# /var/opt/SUNWsamfs/Uninstall
```

4. 找到 Linux 客户机 ISO 映像。该 ISO 映像位于您下载 Oracle HSM 安装软件的目录中（请参见“获取软件”）。

   在示例中，使用 *ssh* 登录至系统信息库主机 *sw-install*（IP 地址 *192.168.0.2*）。我们在目录 */hsmqfs* 中找到软件：

```
[root@linux ~]# ssh root@sw-install
Password:
```

```
[sw_install]root@solaris:~# ls -1 /hsmqfs
./       COPYRIGHT.txt       linux.iso             README.txt
../      iso.md5             Oracle-HSM_6.0/
```

5. 在 Linux 主机上，创建临时目录。

   在示例中，我们创建目录 */hsmtemp*：

   ```
   [root@linux ~]# mkdir /hsmtemp
   [root@linux ~]#
   ```

6. 使 *linux.iso* 映像可用于 Linux 主机。NFS 挂载将映像存放在您刚创建的临时目录上的远程目录。使用命令 *mount -t nfs repository-host-IP:hsm-repository-dir temp-dir*，其中：
   - *-t nfs* 标识要挂载的文件系统的类型。
   - *repository-host-IP* 是托管安装软件的服务器的 IP 地址。
   - *hsm-repository-dir* 是存放 Oracle HSM 安装软件的目录。
   - *temp-dir* 是您在 Linux 主机上创建的临时目录。

   在本示例中，我们在挂载点目录 */hsmtemp* 上以 NFS 方式挂载主机 *sw-install* (*192.168.0.2*) 的目录 */hsmqfs*：

```
[root@linux ~]# mount -t nfs 192.168.0.2:/hsmqfs /hsmtemp
[root@linux ~]#
```

7. 在 Linux 主机上挂载 *linux.iso* 映像。使用命令 *mount -o ro,loop -t iso9660 temp-dir/linux.iso /mnt*，其中：
   - *-o* 指定挂载选项列表。
   - *ro* 以只读方式挂载映像。
   - *loop* 将映像作为循环设备进行挂载。
   - *-t iso9660* 标识要挂载的文件系统的类型。
   - *temp-dir* 是挂载远程映像系统信息库目录的临时目录。
   - */mnt* 是 Linux 系统上的标准临时挂载点目录。

   在本示例中，ISO 映像位于 */hsmtemp* 中：

```
[root@linux ~]# mount -o ro,loop -t iso9660 /hsmtemp/linux.iso /mnt
[root@linux ~]#
```

8. 运行安装程序。使用命令 */mnt/linux1/Install*。

```
[root@linux ~]# /mnt/linux1/Install
```

9. 如果安装程序无法识别已安装的 Linux 内核版本，则会提示您创建一个定制内核。输入 *yes*。

```
[root@linux ~]# ./Install
...
A direct match for your kernel wasn't found. Attempt creating a custom rpm for your kernel (yes/
no)? yes
```

   Linux 内核有许多变体。Oracle HSM 安装程序会编译定制内核模块，使其可以支持尽可能多数的变体。

10. 按照屏幕上的说明操作。
11. 如果您要安装 SuSE Linux 客户机，请配置系统以识别手册页。在文本编辑器中打开 */etc/manpath.config* 文件，并将 *1m* 添加到 *SECTION* 参数的值中。

    在示例中，使用 *vi* 编辑器：

```
[root@linux ~]# vi /etc/manpath.config
...
#-------------------------------------------------------------
# Section names. Manual sections will be searched in the order listed here;
# the default is 1, n, l, 8, 3, 2, 5, 4, 9, 6, 7. Multiple SECTION
# directives may be given for clarity, and will be concatenated together in
# the expected way.
# If a particular extension is not in this list (say, 1mh), it will be
# displayed with the rest of the section it belongs to. The effect of this
# is that you only need to explicitly list extensions if you want to force a
# particular order. Sections with extensions should usually be adjacent to
# their main section (e.g. "1 1mh 8 ...").
SECTION 1 1m n l 8 3 2 3posix 3pm 3perl 5 4 9 6 7
```

12. 如果计划的 Oracle HSM 配置包括其他 Linux 客户机主机，请从头开始重复该过程，直到将该客户机软件安装在所有主机上。
13. 否则，请转至第 5 章 使用 ***samsetup*** 配置向导或第 6 章 配置基本文件系统 。

卸载 Oracle HSM 软件

本节概述了以下过程：

- 卸载 Solaris 主机上的 Oracle HSM
- 卸载 Linux 主机上的 Oracle HSM 客户机。

注意:

如果您打算使用现有配置升级或重新安装 Oracle HSM，请勿卸载软件！卸载会删除所有配置文件。相反，应使用“在 Oracle Solaris 主机上安装、升级或降级 Oracle HSM 软件” 中概述的升级方法之一。

卸载 Solaris 主机上的 Oracle HSM

要完全卸载软件并删除配置文件，请执行如下操作。

1. 以 *root* 用户身份登录主机。

```
root@solaris:~#
```

2. 如果使用 Solaris 映像包管理系统在 Solaris 11 或更高版本上安装了软件，则使用命令 *pkg uninstall SUNWsamfs SUNWsamqassy*（或 *pkg uninstall SUNWqfs SUNWsamqassy*（如果仅安装了 QFS 软件））卸载软件。

```
root@solaris:~#  pkg uninstall SUNWsamfs SUNWsamqassy
```

3. 如果使用 SVR4 *pkginstall* 方法在 Solaris 10 或 Solaris 11 上安装了软件，则使用命令 *pkgrm SUNWsamfsu SUNWsamfsr*（*pkgrm SUNWqfsu SUNWqfsr*（如果仅安装了 QFS 软件）卸载软件。

   按照指定顺序删除软件包：从 *SUNWsamfsu* 开始，到 *SUNWsamfsr* 结束。在示例中，将回复 *yes* 通过管道传输到该命令，以便自动回答所有问题：

```
root@solaris:~# yes | pkgrm SUNWsamfsu SUNWsamfsr
```

4. 如果使用 SVR4 *pkginstall* 方法在 Solaris 10 或 Solaris 11 上安装了软件，则删除不再需要的配置文件和日志文件。

```
root@solaris:~# rm -R /var/opt/SUNWsamfs/
root@solaris:~# rm -R /etc/opt/SUNWsamfs/
root@solaris:~# rm -R /var/adm/sam-log/
root@solaris:~#
```

5. 重新引导主机。

```
root@solaris:~#  reboot
```

6. 在此处停止。

卸载 Linux 主机上的 Oracle HSM 客户机

要卸载并完全删除 Linux 客户机软件，请执行如下操作。

1. 以 *root* 用户身份登录 Linux 客户机主机。

```
[root@linux ~]#
```

2. 运行 Oracle HSM 脚本 */var/opt/SUNWsamfs/Uninstall*（如果只安装了 QFS，则运行 */var/opt/SUNWqfs/Uninstall*）。

   请勿使用任何其他方法！*rpm -e* 等其他方法可能会导致在卸载或重新安装该软件时出现意外结果和问题。因此，请始终使用脚本：

```
[root@linux ~]#  /var/opt/SUNWsamfs/Uninstall
```

5

第 5 章 使用 samsetup 配置向导

samsetup 向导是一个简单的基于文本的菜单驱动实用程序，使您能够快速创建和配置 Oracle HSM 文件系统来满足最常遇到的要求。该向导将指导您完成下列所有基本任务：

- 创建在单个主机上挂载的 QFS 独立文件系统
- 创建在多个主机上挂载的 QFS 共享文件系统
- 为 QFS 文件系统配置 Oracle HSM 归档
- 配置存储硬件，包括主（高速缓存）磁盘存储、归档磁盘存储和可移除介质库、驱动器和介质。

该向导的输出－有效的 Oracle HSM 配置脚本－还可在您创建更专业的解决方案时作为有用的起点。

samsetup 向导基本上是自记录菜单和提示，可指导您完成此过程，并且上下文相关联机帮助将立即可用。因此本章未重复包含工具自身提供的信息。

但是，您在使用此向导之前应查看本书的其他部分，特别是在您对 Oracle HSM 不熟悉时：

- 第 6 章 配置基本文件系统 提供了有关 Oracle HSM 工作方式的基本信息，并介绍了创建文件系统的配置文件和流程。您将需要此信息来完全了解该向导为您提供的选项，即使您从未觉得需要自行创建和编辑配置文件也是如此。
- 如果您需要 Oracle HSM 归档文件系统，您将需要“配置文件系统保护” 中的信息。*samsetup* 向导将不会为关键文件系统元数据和日志配置调度备份。
- 如果您需要 Oracle HSM 共享文件系统，则还请查看第 7 章 从多台主机访问文件系统。“使用 Oracle HSM 软件从多台主机访问文件系统”和“配置 Oracle HSM 共享文件系统”这两个部分将特别重要。
- 如果您需要使用 Oracle HSM 的附加功能，则应查阅本书的相关部分来了解附加配置说明。例如，请参见“配置归档介质验证”、“启用对一次写入多次读取 (Write Once Read Many, WORM) 文件的支持”、“启用对 Linear Tape File System (LTFS) 的支持”、第 9 章 准备高可用性解决方案 、第 8 章 配置 *SAM-Remote* 和第 10 章 配置报告数据库。
- 最终，无论您使用 *samsetup*、命令行还是 Oracle HSM Manager 用户界面来配置文件系统，您均应按照第 13 章 备份 *Oracle HSM* 配置 中所述保护您的工作。

6

第 6 章 配置基本文件系统

QFS 文件系统是所有 Oracle HSM 解决方案的基本构建块。在单独使用时，它们可以为特大文件提供高性能、有效的无限容量和支持。与 Oracle Hierarchical Storage Manager 和经过适当配置的归档存储一起使用时，它们将成为 Oracle HSM 归档文件系统。归档 QFS 文件系统和非归档 QFS 文件系统之后都可以构成更加复杂的多主机、高可用性配置的基础。因此，本章概述了在创建和配置这些文件系统时所涉及的基本任务：

- 配置 QFS 文件系统
- 配置 Oracle HSM 归档文件系统

配置 QFS 文件系统

创建和配置基本 QFS 文件系统是非常简单的。对于每种情况，都需要执行以下任务：

- 准备用于支持该文件系统的磁盘设备。
- 创建主配置文件 (*mcf*)。
- 使用 */opt/SUNWsamfs/sbin/sammkfs* 命令创建文件系统。
- 通过编辑 */etc/vfstab* 文件，将新文件系统添加到主机的虚拟文件系统配置中。
- 挂载新文件系统。

执行此过程时，可以使用 Oracle HSM Manager 图形界面，也可以使用文本编辑器和命令行终端。但在示例中我们使用编辑器和命令行这种方法，使此过程的各个部分比较明晰，从而更便于理解。

在初始 Oracle HSM 配置期间为了简单和方便起见，本节中的过程将在 Solaris 虚拟文件系统的配置文件 (*/etc/vfstab*) 中设置文件系统挂载选项。但是，大多数选项也可以通过可选的 */etc/opt/SUNWsamfs/samfs.cmd* 文件或命令行进行设置。有关详细信息，请参见 *samfs.cmd* 和 *mount_samfs* 手册页。

准备 QFS 文件系统的磁盘存储

在开始配置过程之前，需要选择计划配置所需的磁盘资源。可以使用原始设备分片、ZFS *zvol* 卷或 Solaris Volume Manager 卷。

配置通用 ms 文件系统

1. 以 *root* 用户身份登录到文件系统主机。登录全局区域（如果主机已配置区域）。

```
root@solaris:~#
```

2. 创建文件 */etc/opt/SUNWsamfs/mcf*。

   mcf（主配置文件）是一个表，该表包含以空格分隔的六个列，每个列表示一个用于定义 QFS 文件系统的参数：*Equipment Identifier*、*Equipment Ordinal*、*Equipment Type*、*Family Set*、*Device State* 和 *Additional Parameters*。表中的行表示文件系统设备，这包括存储设备和设备组（系列集）。

   要创建 *mcf* 文件，可以通过在 Oracle HSM Manager 图形用户界面中选择选项来完成，也可以使用文本编辑器。在下面的示例中，我们使用 *vi* 文本编辑器：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
~
~
"/etc/opt/SUNWsamfs/mcf"  [New File]
```

3. 为清晰起见，输入列标题作为注释。

   注释行以井号 (*#*) 开头：

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   ---------------
```

4. 在第一行的 *Equipment Identifier* 字段（第一列）中，输入新文件系统的名称。

   在本示例中，文件系统名为 *qfsms*：

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   ---------------
qfsms
```

5. 在 *Equipment Ordinal* 字段（第二列）中，输入一个用于唯一标识该文件系统的编号。

   设备序号可唯一标识由 Oracle HSM 控制的所有设备。在本示例中，我们用 *100* 代表 *qfsms* 文件系统：

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   ---------------
```

```
qfsms                100
```

6. 在 *Equipment Type* 字段（第三列）中，输入通用 QFS 文件系统的设备类型 (*ms*)：

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   ----------------
qfsms                100        ms
```

7. 在 *Family Set* 字段（第四列）中，输入文件系统的名称。

   Family Set 参数用于定义配置在一起构成一个单元的设备组，例如，机械装置磁带库及其驻留磁带机或者文件系统及其组件磁盘设备。

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   ----------------
qfsms                100        ms         qfsms
```

8. 在 *Device State* 列中输入 *on*，并将 *Additional Parameters* 列留空。

   此行已完成：

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   ----------------
qfsms                100        ms         qfsms      on
```

9. 开始新行。在 *Equipment Identifier* 字段（第一列）中，输入所选的其中一个磁盘设备的标识符，然后在 *Equipment Ordinal* 字段（第二列）中输入唯一编号。

   在示例中，我们缩进了设备行，以强调设备是 *qfsms* 文件系统系列集的一部分，然后通过递增系列集的设备编号来创建设备编号，在本例中为 *101*：

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   ----------------
qfsms                100        ms         qfsms      on
  /dev/dsk/c1t3d0s3   101
```

10. 在磁盘设备行的 *Equipment Type* 字段（第三列）中，输入磁盘设备的设备类型 *md*：

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   ---------------
qfsms                100        ms         qfsms      on
  /dev/dsk/c1t3d0s3   101        md
```

11. 在磁盘设备行的 *Family Set* 字段（第四列）中输入文件系统的系列集名称，在 *Device State* 字段（第五列）中输入 *on*，然后将 *Additional Parameters* 字段（第六列）留空。

    系列集名称 *qfsms* 用于标识磁盘设备是文件系统硬件的一部分。

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   ---------------
qfsms                100        ms         qfsms      on
  /dev/dsk/c1t3d0s3   101        md         qfsms      on
```

12. 接下来添加任何剩余磁盘设备的条目，然后保存文件并退出编辑器。

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   ---------------
qfsms                100        ms         qfsms      on
  /dev/dsk/c1t3d0s3   101        md         qfsms      on
  /dev/dsk/c1t4d0s5   102        md         qfsms      on
:wq
root@solaris:~#
```

13. 运行 *sam-fsd* 命令检查 *mcf* 文件是否存在错误。

    sam-fsd 命令会读取 Oracle HSM 配置文件并初始化文件系统。该命令会在遇到以下错误时停止：

```
root@solaris:~# sam-fsd
```

14. 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：

```
root@solaris:~# sam-fsd
Problem in mcf file /etc/opt/SUNWsamfs/mcf for filesystem qfsms
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 *102*（第二个 *md* 设备）设备名称的分片编号中键入了一个字符 *o* 而非 0：

```
qfsms                100        ms         qfsms      on
  /dev/dsk/c0t0d0s0   101        md         qfsms      on
  /dev/dsk/c0t3d0so   102        md         qfsms      on
```

15. 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：

```
root@solaris:~# sam-fsd
Trace file controls:
sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
              cust err fatal ipc misc proc date
              size    10M  age 0
sam-archiverd /var/opt/SUNWsamfs/trace/sam-archiverd
              cust err fatal ipc misc proc date module
              size    10M  age 0
sam-catserverd /var/opt/SUNWsamfs/trace/sam-catserverd
              cust err fatal ipc misc proc date module
              size    10M  age 0
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
```

16. 为新文件系统创建挂载点目录，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 */qfsms* 挂载点目录并将权限设置为 *755* (*-rwxr-xr-x*)：

```
root@solaris:~# mkdir /qfsms
root@solaris:~# chmod 755 /qfsms
```

17. 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*。

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

18. 如果 *samd config* 命令失败并显示消息 *You need to run /opt/SUNWsamfs/util/SAM-QFS-post-install*，则表示您在安装软件时忘记了运行安装后脚本。请立即运行此脚本。

```
root@solaris:~# /opt/SUNWsamfs/util/SAM-QFS-post-install
- The administrator commands will be executable by root only (group bin).
If this is the desired value, enter "y".  If you want to change
the specified value enter "c".
...
root@solaris:~#
```

19. 使用 */opt/SUNWsamfs/sbin/sammkfs* 命令和文件系统的系列集名称创建文件系统。

    对于 *md* 设备，Oracle HSM 软件使用双重分配和默认磁盘分配单元 (Disk Allocation Unit, DAU) 大小。对于通用文件系统来说，这是一个很好的选择，因为它可以容纳大文件和小文件以及 I/O 请求。在示例中，我们接受默认值：

```
root@solaris:~# sammkfs qfsms
Building 'qfsms' will destroy the contents of devices:
  /dev/dsk/c1t3d0s3
  /dev/dsk/c1t4d0s5
Do you wish to continue? [y/N]yes
total data kilobytes       = ...
```

    如果我们使用的是指定更符合我们的 I/O 需求的非默认 DAU 大小所需的 *mr* 设备，则可以将 *sammkfs* 命令与 *-a* 选项结合使用来实现此目的：

```
root@solaris:~# sammkfs -a 16 qfs2ma
```

    有关其他信息，请参见 *sammkfs* 手册页。

20. 备份操作系统的 */etc/vfstab* 文件。

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
```

21. 将新文件系统添加到操作系统的虚拟文件系统配置。在文本编辑器中打开此文件，并为 *qfsms* 系列集设备添加一行：

```
#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       -
...
qfsms      -       /qfsms    samfs   -
```

22. 在 */etc/vfstab* 文件的第六列 *Mount at Boot* 中，输入 *no*（适用于大多数情况）。

```
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       -
...
qfsms      -        /qfsms    samfs   -     no
```

23. 要指定循环分配，请添加 *stripe=0* 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       -
...
qfsms      -        /qfsms    samfs   -     no       stripe=0
```

24. 要指定分散读写分配，请添加 *stripe=stripe-width* 挂载选项，其中 *stripe-width* 是应写入分散读写中的每个磁盘的磁盘分配单元 (Disk Allocation Unit, DAU) 数目。

    在示例中，我们将分散读写宽度设置为一个 DAU：

```
#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       -
...
qfsms      -        /qfsms    samfs   -     no       stripe=1
```

这里，*stripe=1* 选项用于指定分散读写宽度为 1 个 DAU 且写入大小为两个 DAU。因此，当文件系统同时写入两个 DAU 时，它会向 *qfsms* 系列集中的每个 *md* 磁盘设备（共两个）各写入一个 DAU。

25. 对 */etc/vfstab* 文件进行所需的其他更改。

    例如，如果元数据服务器没有响应，要在后台挂载文件系统，则应将 *bg* 挂载选项添加到 *Mount Options* 字段：

```
#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       -
...
qfsms      -        /qfsms    samfs   -     no       stripe=1,bg
```

26. 保存 *vfstab* 文件并关闭编辑器。

```
...
qfsms      -        /qfsms    samfs   -     no       stripe=1
:wq
root@solaris:~#
```

27. 挂载新文件系统：

```
root@solaris:~# mount /qfsms
```

28. 文件系统现已完成，可以使用了。

    从此处执行以下各操作并参考相应的内容：

    - 如果要使用 Oracle Hierarchical Storage Manager 设置归档文件系统，请参见“配置 Oracle HSM 归档文件系统”。
    - 如果需要在文件系统上启用 WORM（Write Once Read Many，一写多读）功能，请参见“启用对一次写入多次读取 (Write Once Read Many, WORM) 文件的支持”。
    - 如果需要与使用 LTFS 的系统交互，或者需要在远程站点之间传输大量数据，请参见“启用对 Linear Tape File System (LTFS) 的支持”。
    - 如果有其他需求（例如，多主机文件系统访问或高可用性配置），请参见“进阶知识”。

配置高性能 ma 文件系统

在文件系统主机上安装 Oracle HSM 软件之后，可按以下所述对 *ma* 文件系统进行配置。

1. 以 *root* 用户身份登录到文件系统主机。登录全局区域（如果主机已配置区域）。

```
root@solaris:~#
```

2. 选择用于存储元数据的磁盘设备。
3. 选择用于存储数据的磁盘设备。
4. 创建 *mcf* 文件。

   要创建 *mcf* 文件，可以通过在 Oracle HSM Manager 图形用户界面中选择选项来完成，也可以使用文本编辑器。在下面的示例中，我们使用 *vi* 文本编辑器：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
~
"/etc/opt/SUNWsamfs/mcf"  [New File]
```

5. 为清晰起见，输入列标题作为注释。

   注释行以井号 (*#*) 开头：

```
# Equipment           Equipment  Equipment  Family  Device   Additional
# Identifier          Ordinal    Type       Set     State    Parameters
#------------------   ---------  ---------  ------  ------   -----------------
```

6. 为文件系统系列集创建一个条目。

   在本示例中，我们将文件系统标识为 *qfsma*，将设备序号递增到 *200*，将设备类型设置为 *ma*，将系列集名称设置为 *qfsma*，并将设备状态设置为 *on*：

```
# Equipment           Equipment  Equipment  Family  Device   Additional
# Identifier          Ordinal    Type       Set     State    Parameters
#------------------   ---------  ---------  ------  ------   -----------------
qfsma          200        ma         qfsma   on
```

7. 为每个元数据设备添加一个条目。在设备标识符列中输入所选磁盘设备的标识符，设置设备序号，并将设备类型设置为 *mm*。

   添加足够的元数据设备，以存储文件系统大小所需的元数据。在示例中，我们添加单个元数据设备：

```
# Equipment           Equipment  Equipment  Family  Device   Additional
```

```
# Identifier          Ordinal    Type       Set     State    Parameters
#------------------   ---------  ---------  ------  ------   -----------------
qfsma                 200        ma         qfsma   on
  /dev/dsk/c0t0d0s0    201        mm         qfsma   on
```

8. 接下来添加数据设备的条目，然后保存文件并退出编辑器。

   这些可以是 *md*、*mr* 或分散读写组 (*gXXX*) 设备。在本示例中，我们将指定 *md* 设备：

```
# Equipment          Equipment  Equipment  Family  Device   Additional
# Identifier         Ordinal    Type       Set     State    Parameters
#------------------  ---------  ---------  ------  ------   -----------------
qfsma                200        ma         qfsma   on
  /dev/dsk/c0t0d0s0   201        mm         qfsma   on
  /dev/dsk/c0t3d0s0   202        md         qfsma   on
  /dev/dsk/c0t3d0s1   203        md         qfsma   on
:wq
root@solaris:~#
```

9. 运行 *sam-fsd* 命令检查 *mcf* 文件是否存在错误。

   sam-fsd 命令会读取 Oracle HSM 配置文件并初始化文件系统。该命令会在遇到以下错误时停止：

```
root@solaris:~# sam-fsd
```

10. 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：

```
root@solaris:~# sam-fsd
Problem in mcf file /etc/opt/SUNWsamfs/mcf for filesystem qfsma
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 *202*（第一个 *md* 设备）的设备名称的分片编号部分中键入了一个感叹号 *!* 而非 1：

```
sharefs1              200         ma         qfsma   on
  /dev/dsk/c0t0d0s0    201         mm         qfsma   on
  /dev/dsk/c0t0d0s!    202         md         qfsma   on
  /dev/dsk/c0t3d0s0    203         md         qfsma   on
```

11. 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：

```
root@solaris:~# sam-fsd
Trace file controls:
sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
              cust err fatal ipc misc proc date
              size    10M  age 0
sam-archiverd /var/opt/SUNWsamfs/trace/sam-archiverd
              cust err fatal ipc misc proc date module
              size    10M  age 0
sam-catserverd /var/opt/SUNWsamfs/trace/sam-catserverd
              cust err fatal ipc misc proc date module
              size    10M  age 0
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
```

12. 使用 */opt/SUNWsamfs/sbin/sammkfs* 命令和文件系统的系列集名称创建文件系统。

    在示例中，我们按以下方式创建文件系统：对具有 *md* 设备的 *ma* 文件系统使用默认磁盘分配单元 (Disk Allocation Unit, DAU) 大小（即 *64* 千字节）：

```
root@solaris:~# sammkfs qfsma
Building 'qfsma' will destroy the contents of devices:
  /dev/dsk/c0t0d0s0
  /dev/dsk/c0t3d0s0
  /dev/dsk/c0t3d0s1
Do you wish to continue? [y/N]yes
total data kilobytes       = ...
```

    默认磁盘分配单元大小是一个很好的通用选择。但是，如果文件系统主要是用于支持较小的文件或读写数据量较少的应用程序，则我们也可以将 DAU 大小指定为 *16* 或 *32* 千字节。要指定 16 千字节 DAU，需将 *sammkfs* 命令与 *-a* 选项结合使用：

```
root@solaris:~# sammkfs -a 16 qfsma
```

    mr 设备和 *gXXX* 分散读写组的 DAU 在范围 *8-65528* 千字节内完全可调（增量为 8 千字节）。对于 *mr* 设备，默认值为 *64* 千字节；对于 *gXXX* 分散读写组，默认值为 *256* 千字节。有关其他详细信息，请参见 *sammkfs* 手册页。

13. 备份操作系统的 */etc/vfstab* 文件。

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
```

14. 将新文件系统添加到操作系统的虚拟文件系统配置。在文本编辑器中打开 */etc/vfstab* 文件，并为 *qfsma* 系列集添加一行。

```
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       -
...
qfsma      -        /qfsma    samfs   -
```

15. 在 */etc/vfstab* 文件的第六列 *Mount at Boot* 中，输入 *no*。

```
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       -
...
qfsma      -        /qfsma    samfs   -     no
```

16. 要指定循环分配，请添加 *stripe=0* 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       -
...
qfsma      -        /qfsma    samfs   -     no       stripe=0
```

17. 要指定分散读写分配，请添加 *stripe=stripe-width* 挂载选项，其中 *stripe-width* 是 *[1-255]* 范围内的一个整数，表示应写入分散读写中的每个磁盘的磁盘分配单元 (Disk Allocation Unit, DAU) 数目。

    指定分散读写分配时，数据将并行写入设备。因此，为获得最佳性能，在选择分散读写宽度时应确保充分利用存储硬件所提供的带宽。请注意，给定分散读写宽度传输的数据量取决于硬件的配置情况。对于在单个磁盘卷上实施的 *md* 设备，当分散读写宽度为 **1** 时，将向每个磁盘（共两个）写入一个 64 千字节的 DAU（共计

128 千字节）。对于在 3+1 RAID 5 卷组上实施的 *md* 设备，使用这一相同的分散读写宽度时，将向每个设备（共两个）上的每个数据磁盘（共三个）传输一个 64 千字节的 DAU（每次传输共计六个 DAU 或 384 千字节）。在示例中，我们将分散读写宽度设置为一个 DAU：

```
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  -------------------------
/devices   -        /devices  devfs   -     no       -
...
qfsma      -        /qfsma    samfs   -     no       stripe=1
```

18. 可以尝试调整分散读写宽度，以更好地利用可用的硬件。在文件系统的 *Mount Options* 字段中，设置 *stripe=n* 挂载选项，其中 *n* 是为文件系统指定的 DAU 大小的倍数。测试文件系统的 I/O 性能，并根据需要重新调整设置。

    设置 *stripe=0* 时，Oracle HSM 会使用循环分配将文件写入设备。系统会将每个文件尽可能分配在一个设备上，直至填满该设备为止。循环分配是共享文件系统和多流环境的首选设置。

    在示例中，我们已确定分散读写宽度为一时 RAID-5 卷组的带宽并未充分利用，因此我尝试 *stripe=2*：

```
#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       -
...
qfsma      -        /qfsma    samfs   -     no       ...,stripe=2
```

19. 否则保存 *vfstab* 文件。

```
...
qfsma      -        /qfsma    samfs   -     no       stripe=1
:wq
root@solaris:~#
```

20. 挂载新文件系统：

```
root@solaris:~# mount /qfsms
```

基本文件系统现已完成，可以使用了。

21. 如果要使用 Oracle Hierarchical Storage Manager 设置归档文件系统，请参见“配置 Oracle HSM 归档文件系统”。
22. 如果需要在文件系统上启用 WORM（Write Once Read Many，一写多读）功能，请参见“启用对一次写入多次读取 (Write Once Read Many, WORM) 文件的支持”。
23. 如果需要与使用 LTFS 的系统交互，或者需要在远程站点之间传输大量数据，请参见“启用对 Linear Tape File System (LTFS) 的支持”。
24. 如果有其他需求（例如，多主机文件系统访问或高可用性配置），请参见“进阶知识”。
25. 否则，请转至第 11 章 配置通知和日志记录。

配置 Oracle HSM 归档文件系统

归档文件系统将一个或多个 QFS *ma* 或 *ms* 类型的文件系统与归档存储和 Oracle HSM 软件组合在一起。Oracle HSM 软件将辅助磁盘存储和/或可移除介质集成到基本文件系统操作之中，因此文件可在各种不同的介质上保留多个副本。这种冗余不但可提供连续数据保护，还可以为超大文件提供策略驱动的保留和有效的存储。

- 将磁盘归档文件系统添加到 Oracle HSM 主机配置
- 配置归档文件系统
- 挂载归档文件系统
- 配置归档过程
- 存储在网络连接磁带库中的目录归档介质
- 配置文件系统保护
- 配置归档介质验证
- 在 Oracle HSM 文件系统上启用 WORM 支持

将磁盘归档文件系统添加到 Oracle HSM 主机配置

在 Oracle HSM 主机上创建任何所需的磁盘归档文件系统并将本地和远程磁盘归档文件添加到主机配置。使用下面列出的过程：

- 创建本地文件系统以用作磁盘归档
- 将磁盘归档添加到 Oracle HSM 主机配置

创建本地文件系统以用作磁盘归档

如果您要将 Oracle HSM 服务器上的文件系统用作磁盘归档，请执行如下操作。

1. 为 Oracle HSM 服务器上本地挂载的每个磁盘归档卷创建 QFS、ZFS 或 UFS 文件系统。

   请勿使用必须与其他应用程序共享的现有通用文件系统。

2. 如果将一个或多个 QFS 文件系统配置为磁盘归档卷，则应为每个文件系统分配一个系列集名称和一组用于将其明确标识为归档存储卷的设备序号。

   明确区分 QFS 归档存储文件系统与其他 Oracle HSM 主文件系统可以使配置更易于理解和维护。在本示例中，新文件系统的名称 *DISKVOL1* 表明了它的功能。在 *mcf* 文件中，此名称和设备序号 *800* 将该磁盘归档与 *samms* 和 *100* 区别开来（后者是我们在后续示例中创建归档 Oracle HSM 文件系统时要使用的系列集名称和序号）：

```
# Archiving file systems:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ -----------------

# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ -----------------
DISKVOL1                 800       ms        DISKVOL1  on
 /dev/dsk/c6t0d1s7         801       md        DISKVOL1  on
 /dev/dsk/c4t0d2s7         802       md        DISKVOL1  on
```

3. 接下来，将磁盘归档添加到 Oracle HSM 主机系统配置。

将磁盘归档添加到 Oracle HSM 主机配置

1. 在 Oracle HSM 主机上，创建一个父目录，用于存放磁盘归档卷的挂载点，就像在物理磁带库中存放归档磁带卷一样。

   在示例中，我们创建目录 */diskvols*。

```
root@solaris:~# mkdir /diskvols
```

2. 在父目录中，为每个归档文件系统创建一个挂载点目录。

   在示例中，我们创建挂载点目录 *DISKVOL1* 和 *DISKVOL2* 到 *DISKVOL15*：

```
root@solaris:~# mkdir /diskvols/DISKVOL1
root@solaris:~# mkdir /diskvols/DISKVOL2
...
root@solaris:~# mkdir /diskvols/DISKVOL15
```

3. 在 Oracle HSM 主机上，备份 */etc/vfstab* 文件。

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
```

4. 在编辑器中打开 */etc/vfstab* 文件。

```
root@solaris:~# vi /etc/vfstab
```

5. 为用作磁盘归档的每个本地 QFS 文件系统添加一个条目。使用 *samfs* 文件系统类型标识每个条目并添加挂载选项 *nosam*（除非您希望 Oracle HSM 归档该归档）。

   nosam 挂载选项可确保存储在 QFS 文件系统上的归档副本本身不进行归档。

   在本示例中，使用 *vi* 编辑器为一个本地 QFS 文件系统 *DISKVOL1* 添加条目。

```
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       -
...
DISKVOL1          -        /diskvols/DISKVOL1  samfs   -     no       nosam
```

6. 为用作磁盘归档的每个 NFS 文件系统添加条目。使用 *nfs* 文件系统类型标识每个条目。

   在本示例中，使用 *vi* 编辑器添加条目 *DISKVOL2* 到 *DISKVOL15*，其中 *nfs1* 是托管磁盘归档 *DISKVOL2* 到 *DISKVOL13* 的 NFS 服务器，*oscsa1* 是托管其余磁盘归档的 Oracle 存储云软件设备主机的名称：

```
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       -
...
DISKVOL1          -        /diskvols/DISKVOL1  samfs   -     no       nosam
nfs1:/DISKVOL2    -        /diskvols/DISKVOL2  nfs     -     yes      -
nfs1:/DISKVOL3    -        /diskvols/DISKVOL3  nfs     -     yes      -
...
oscsa1:/DISKVOL14 -        /diskvols/DISKVOL3  nfs     -     yes      -
oscsa1:/DISKVOL15 -        /diskvols/DISKVOL15 nfs     -     yes      -
```

7. 保存 */etc/vfstab* 文件并关闭编辑器。

```
...
oscsa1:/DISKVOL14 -        /diskvols/DISKVOL3  nfs     -     yes      -
oscsa1:/DISKVOL15 -        /diskvols/DISKVOL15 nfs     -     yes      -
:wq
root@solaris:~#
```

8. 在 Oracle HSM 主机上，挂载磁盘归档文件系统。

   在示例中，我们挂载 *DISKVOL1* 和 *DISKVOL2* 到 *DISKVOL15*：

```
root@solaris:~# mount /diskvols/DISKVOL1
root@solaris:~# mount /diskvols/DISKVOL2
...
root@solaris:~# mount /diskvols/DISKVOL14
root@solaris:~# mount /diskvols/DISKVOL15
```

9. 现在，准备可移除介质库和驱动器。

准备可移除介质库和驱动器

本节介绍以下任务：

- 配置 Oracle StorageTek ACSLS 网络连接自动化库
- 为带有条码的可移除介质配置标记行为
- 设置驱动器计时值

配置 Oracle StorageTek ACSLS 网络连接自动化库

如果您有 Oracle StorageTek ACSLS 网络连接库，则可以按以下步骤进行配置，也可以使用 Oracle HSM Manager图形用户界面来自动发现和配置此库（有关使用 Oracle HSM Manager 的说明，请参见联机帮助）。

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. 转到 */etc/opt/SUNWsamfs* 目录。

```
root@solaris:~# cd /etc/opt/SUNWsamfs
```

3. 在文本编辑器中，使用与要配置的网络连接库的类型相对应的名称创建一个新文件。

在示例中，我们为 Oracle StorageTek ACSLS 网络连接库创建一个参数文件：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/acsls1params
# Configuration File for an ACSLS Network-Attached Tape Library 1
```

4. 输入 Oracle HSM 软件在与 ACSLS 连接库进行通信时将使用的参数和值。

   Oracle HSM 软件使用以下 Oracle StorageTek Automated Cartridge System 应用编程接口 (Automated Cartridge System Application Programming Interface, ACSAPI) 参数来控制 ACSLS 管理库（有关更多信息，请参见 *stk* 手册页）：

   - *access=user-id* 指定用于访问控制的可选用户标识值。默认情况下，没有基于用户标识的访问控制。
   - *hostname=hostname* 指定运行 StorageTek ACSLS 接口的服务器的主机名。
   - *portnum=portname* 指定用于在 ACSLS 和 Oracle HSM 软件之间进行通信的端口号。
   - *ssihost=hostname* 指定标识连接到 ACSLS 主机的网络的多宿主 Oracle HSM 服务器的主机名。其默认值为本地主机的名称。
   - *ssi_inet_port=ssi-inet-port* 指定 ACSLS 服务器系统接口必须用于传入 ACSLS 响应的固定防火墙端口。指定 *0* 或 [*1024-65535*] 范围内的一个值。默认值为 *0*，即允许动态端口分配。
   - *csi_hostport=csi-port* 指定 ACSLS 服务器上 Oracle HSM 将其 ACSLS 请求发送到的客户机系统接口端口号。指定 *0* 或 [*1024-65535*] 范围内的一个值。默认值为 *0*，表示系统将在 ACSLS 服务器上的端口映射器中查询端口。
   - *capid=(acs=acsnum, lsm=lsmnum, cap=capnum)* 指定磁带存取口 (cartridge access port, CAP) 的 ACSLS 地址，其中 *acsnum* 是库的 Automated Cartridge System (ACS) 编号，*lsmnum* 是用于存放 CAP 的模块的库存储模块 (Library Storage Module, LSM) 编号，*capnum* 是所需 CAP 的标识号。整个地址用括号括起来。
   - *capacity=(index-value-list)* 指定可移除介质磁带的容量，其中 *index-value-list* 是以逗号分隔的 *index=value* 对的列表。此列表中的每个 *index* 是 ACSLS 定义的介质类型的索引，每个 *value* 是相应的卷容量（以 1024 字节为单位）。

     ACSLS 文件 */export/home/ACSSS/data/internal/mixed_media/media_types.dat* 定义介质类型索引。通常，只有对于新磁带类型或者需要覆盖所支持的容量时，才需要提供容量条目。
   - *device-path-name=(acs=ACSnumber, lsm=LSMnumber, panel=Panelnumber, drive=Drivenumber)[shared]* 指定附加到客户机的驱动器的 ACSLS 地址，其中 *device-path-name* 标识 Oracle HSM 服务器上的设备，*acsnum* 是库的 Automated Cartridge System (ACS) 编号，*lsmnum* 是控制驱动器的模块的库存储模块 (Library Storage Module, LSM)

编号，*Panelnumber* 是安装驱动器的面板的标识号，*Drivenumber* 是驱动器的标识号。整个地址用括号括起来。

在 ACSLS 地址后面添加可选 *shared* 关键字将允许两个或更多 Oracle HSM 服务器共享驱动器，但每个服务器均保留对各自介质的独占控制。默认情况下，共享驱动器中的磁带可以在卸载之前空闲 60 秒。

在示例中，我们将 *acslserver1* 标识为 ACSLS 主机，限制 *sam_user* 的访问权限，指定动态端口分配，并映射一个磁带存取口和两个驱动器：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/acsls1params
# Configuration File for an ACSLS Network-Attached Tape Library 1
hostname = acslserver1
portnum = 50014
access = sam_user
ssi_inet_port = 0
csi_hostport = 0
capid = (acs=0, lsm=1, cap=0)
/dev/rmt/0cbn = (acs=0, lsm=1, panel=0, drive=1)
/dev/rmt/1cbn = (acs=0, lsm=1, panel=0, drive=2)
```

5. 保存文件并关闭编辑器。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/acsls1params
# /etc/opt/SUNWsamfs/acslibrary1
# Configuration File for an ACSLS Network-Attached Tape Library
...
/dev/rmt/0cbn = (acs=0, lsm=1, panel=0, drive=1)
/dev/rmt/1cbn = (acs=0, lsm=1, panel=0, drive=2)
:wq
root@solaris:~#
```

6. 如果库或应用程序软件对带有条码的可移除介质使用非标准标签，则现在配置标记行为。
7. 如果已知驱动器或应用程序软件与 Oracle HSM 默认值兼容，则现在设置驱动器计时值。
8. 否则，请转至“配置归档文件系统”。

为带有条码的可移除介质配置标记行为

默认情况下，如果库中包含条码读取器和带条码的介质，则 Oracle HSM 软件会自动使用条码中的前六位字符来标记卷。但是，您可以配置 Oracle HSM 以使卷标签基于条码的替代读取。为此，请执行如下操作：

1. 以 *root* 用户身份登录 Oracle HSM 主机。

```
root@solaris:~#
```

2. 如果您需要非默认行为或者之前覆盖了默认值并需要重置该默认值，请在文本编辑器中打开 */etc/opt/SUNWsamfs/defaults.conf* 文件。

   在示例中，用 *vi* 编辑器打开文件：

```
root@solaris:~# vi /opt/SUNWsamfs/examples/defaults.conf
...
```

3. 找到行 *labels* =（如果存在），否则请添加此行。

   在示例中，添加指令：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.
...
labels =
```

4. 要重新启用基于条码前六位字符的默认自动标记行为，请将 *labels* 指令的值设置为 *barcodes*。保存文件并关闭编辑器。

   现在，Oracle HSM 软件会自动使用磁带条码的前六位字符作为标签来重新标记无标签磁带：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
...
labels = barcodes
:wq
root@solaris:~#
```

5. 要启用基于磁盘上条码的后六位字符的自动标记行为，请将 *labels* 指令的值设置为 *barcodes_low*。保存文件并关闭编辑器。

   将 *labels* 指令设置为 *barcodes_low* 时，Oracle HSM 软件会自动使用磁带条码的后六位字符作为标签来重新标记无标签磁带：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
...
labels = barcodes_low
:wq
root@solaris:~#
```

6. 要禁用自动标记并将 Oracle HSM 配置为从磁带读取标签，请将 *labels* 指令的值设置为 *read*。保存文件并关闭编辑器。

   将 *labels* 指令设置为 *read* 值时，Oracle HSM 软件将无法自动重新标记磁带：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
...
labels = read
idle_unload = 0
...
:wq
root@solaris:~#
```

7. 如果已知驱动器或应用程序软件与 Oracle HSM 默认值兼容，则现在设置驱动器计时值。
8. 否则，请转至“配置归档文件系统”。

设置驱动器计时值

默认情况下，Oracle HSM 软件按以下方式设置驱动器计时参数：

- 指定设备类型卸载介质之前必须经过的最短时间是 *60* 秒。
- Oracle HSM 软件在向库（将响应 SCSI *unload* 命令）发出新命令之前所等待的时间为 *15* 秒。
- Oracle HSM 软件在卸载空闲驱动器之前所等待的时间为 *600* 秒（即 10 分钟）。
- Oracle HSM 软件在卸载由两个或更多 Oracle HSM 服务器共享的空闲驱动器之前所等待的时间为 *600* 秒（即 10 分钟）。

要更改默认计时值，请执行如下操作：

1. 如果尚未登录，请以 *root* 用户身份登录到 Oracle HSM 主机。

```
root@solaris:~#
```

2. 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：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character and change the value.
...
```

3. 如果需要，指定特定设备类型在卸载介质之前必须经历的最短时间。在 *defaults.conf* 文件中，添加一个 *equipment-type_delay* = *number-of-*

seconds 形式的指令，其中 *equipment-type* 是标识所配置驱动器类型的双字符 Oracle HSM 代码，*number-of-seconds* 是一个表示该设备类型的默认秒数的整数。

有关设备类型代码和相应设备的列表，请参见附录 A, 设备类型词汇表。在示例中，我们将 LTO 驱动器（设备类型 *li*）的卸载延迟从默认值（*60* 秒）更改为 *90* 秒：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character and change the value.
...
li_delay = 90
```

4. 如果需要，指定 Oracle HSM 软件在向库（将响应 SCSI *unload* 命令）发出新命令之前所等待的时间。在 *defaults.conf* 文件中，添加一个 *equipment-type* *_unload* = *number-of-seconds* 形式的指令，其中 *equipment-type* 是标识所配置驱动器类型的双字符 Oracle HSM 代码，*number-of-seconds* 是一个表示该设备类型的秒数的整数。

   有关设备类型代码和相应设备的列表，请参见附录 A, 设备类型词汇表。设置在最糟糕情况下响应 *unload* 命令时库可能需要的最长时间。在示例中，我们将 LTO 驱动器（设备类型 *li*）的卸载延迟从默认值（*15* 秒）更改为 *35* 秒：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character and change the value.
...
li_delay = 90
li_unload = 35
```

5. 如果需要，指定 Oracle HSM 在卸载空闲驱动器之前所等待的时间。在 *defaults.conf* 文件中，添加一个 *idle_unload* = *number-of-seconds* 形式的指令，其中 *number-of-seconds* 是一个表示指定秒数的整数。

   指定 *0* 可禁用此功能。在示例中，我们通过将默认值（*600* 秒）更改为 *0* 来禁用此功能：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character and change the value.
...
li_delay = 90
li_unload = 35
idle_unload = 0
```

6. 如果需要，指定 Oracle HSM 软件在卸载共享空闲驱动器之前所等待的时间。在 *defaults.conf* 文件中，添加一个 *shared_unload* = *number-of-seconds* 形式的指令，其中 *number-of-seconds* 是一个表示指定秒数的整数。

   您可以将 Oracle HSM 服务器配置为共享可移除介质驱动器。当拥有所装入介质的服务器并未实际使用驱动器时，该指令会释放驱动器，以供其他服务器使用。指定 *0* 可禁用此功能。在示例中，我们通过将默认值（*600* 秒）更改为 *0* 来禁用此功能：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character and change the value.
...
idle_unload = 600
li_delay = 90
li_unload = 35
idle_unload = 0
shared_unload = 0
```

7. 保存文件并关闭编辑器。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character and change the value.
...
idle_unload = 600
li_delay = 90
li_unload = 35
idle_unload = 0
shared_unload = 0
:wq
root@solaris:~#
```

8. 现在，配置归档文件系统。

配置归档文件系统

创建归档文件系统的过程与创建非归档文件系统的过程相同，不同之处在于我们添加了用于存储额外数据文件副本的设备：

1. 首先配置 QFS 文件系统。可以配置通用 *ms* 或高性能 *ma* 文件系统。

   虽然可以使用 Oracle HSM Manager 图形用户界面来创建文件系统，但在本节的示例中，我们使用 *vi* 编辑器。在这里，我们使用系列集名称 *samms* 和设备序号 *100* 创建一个通用的 *ms* 文件系统：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Archiving file systems:
#
# Equipment              Equipment Equipment Family  Device Additional
# Identifier             Ordinal   Type      Set     State  Parameters
#----------------------- --------- --------- ------  ------ ------------------
samms                   100       ms        samms   on
 /dev/dsk/c1t3d0s3       101       md        samms   on
 /dev/dsk/c1t3d0s4       102       md        samms   on
```

2. 要添加归档磁带存储，请先添加一个库条目。在设备标识符字段中，输入库的设备 ID 并分配设备序号：

   在本示例中，库设备标识符为 */dev/scsi/changer/c1t0d5*。我们将设备序号设置为 *900*（即所选磁盘归档范围的下一个范围）：

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ -----------------
DISKVOL1                800       ms        DISKVOL1  on
 /dev/dsk/c6t0d1s7       801       md        DISKVOL1  on
 /dev/dsk/c4t0d2s7       802       md        DISKVOL1  on
/dev/scsi/changer/c1t0d5 900
```

3. 将设备类型设置为 *rb*（一个通用 SCSI 连接磁带库），为磁带库系列集提供一个名称，并将设备状态设置为 *on*。

   在本示例中，我们将使用库 *library1*：

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ -----------------
DISKVOL1                800       ms        DISKVOL1  on
 /dev/dsk/c6t0d1s7       801       md        DISKVOL1  on
 /dev/dsk/c4t0d2s7       802       md        DISKVOL1  on
/dev/scsi/changer/c1t0d5 900       rb        library1  on
```

4. （可选）在 *Additional Parameters* 列中，输入将用于存储库目录的路径。

   如果未选择提供目录路径，则软件将会为您设置一个默认路径。

请注意，由于文档布局的限制，示例中缩写了库目录的长路径 *var/opt/SUNWsamfs/catalog/library1cat*：

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ -----------------
DISKVOL1                 800       ms        DISKVOL1  on
 /dev/dsk/c6t0d1s7        801       md        DISKVOL1  on
 /dev/dsk/c4t0d2s7        802       md        DISKVOL1  on
/dev/scsi/changer/c1t0d5 900       rb        library1  on     ...catalog/library1cat
```

5. 接下来，为属于库系列集的每个磁带机添加一个条目。按照在库中的实际安装顺序添加每个驱动器。

   请遵循在“确定驱动器在库中的安装顺序”中创建的驱动器映射文件中所列的驱动器顺序。在示例中，附加到 Solaris */dev/rmt/1*、*/dev/rmt/0*、*/dev/rmt/2* 和 */dev/rmt/3* 的驱动器分别是库中的驱动器 *1*、*2*、*3* 和 *4*。因此，*/dev/rmt/1* 是 *mcf* 文件中列出的第一个磁带机（作为设备 *901*）。*tp* 设备类型指定通用 SCSI 连接磁带机：

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------  ------ -----------------
DISKVOL1                  800       ms        DISKVOL1 on
 /dev/dsk/c6t0d1s7         801       md        DISKVOL1 on
 /dev/dsk/c4t0d2s7         802       md        DISKVOL1 on
/dev/scsi/changer/c1t0d5  900       rb        library1  on     ...catalog/library1cat
 /dev/rmt/1cbn            901       tp        library1  on
 /dev/rmt/0cbn            902       tp        library1  on
 /dev/rmt/2cbn            903       tp        library1  on
 /dev/rmt/3cbn            904       tp        library1  on
```

6. 最后，如果您希望自行配置 Oracle HSM 历史记录，请使用设备类型 *hy* 添加一个条目。在系列集和设备状态列中输入一个连字符，并在附加参数列中输入历史记录目录的路径。

   历史记录是一个虚拟库，用于对已从归档中导出的卷进行编录。如果您未配置历史记录，则软件会使用指定的最高设备序号增加一来自动创建一个历史记录。

请注意，由于页面布局的限制，我们在示例中缩写了历史记录目录的长路径。完整路径是 */var/opt/SUNWsamfs/catalog/historian_cat*：

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ -----------------
DISKVOL1                 800       ms        DISKVOL1  on
 /dev/dsk/c6t0d1s7        801       md        DISKVOL1  on
 /dev/dsk/c4t0d2s7        802       md        DISKVOL1  on
/dev/scsi/changer/c1t0d5 900       rb        library1  on     ...catalog/SL150cat
 /dev/rmt/0cbn            901       tp        library1  on
 /dev/rmt/1cbn            902       tp        library1  on
 /dev/rmt/2cbn            903       tp        library1  on
 /dev/rmt/3cbn            904       tp        library1  on
historian                999       hy        -         -      ...catalog/historian_cat
```

7. 保存 *mcf* 文件并关闭编辑器。

```
...
 /dev/rmt/3cbn            904       tp        library1  on
historian                999       hy        -      -      ...catalog/historian_cat
:wq
root@solaris:~#
```

8. 运行 *sam-fsd* 命令检查 *mcf* 文件是否存在错误。更正发现的任何错误。

   sam-fsd 命令会读取 Oracle HSM 配置文件并初始化文件系统。该命令会在遇到以下错误时停止：

```
root@solaris:~# sam-fsd
Trace file controls:
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
root@solaris:~#
```

9. 如果要使用一个或多个文件系统作为归档存储卷，请在文本编辑器中创建 */etc/opt/SUNWsamfs/diskvols.conf* 文件，并为每个文件系统分配一个卷序列号 (volume serial number, VSN)。对于每个文件系统，创建一个新行，该行包含所需的卷序列号、空格和文件系统挂载点的路径。然后保存文件。

在示例中，我们有三个基于磁盘的归档卷：*DISKVOL1* 是我们本地创建的专用于此目的的 QFS 文件系统。*DISKVOL2* 到 *DISKVOL15* 是 UFS 文件系统。全部都挂载在 */diskvols/* 目录中：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/diskvols.conf
# Volume
# Serial     Resource
# Number     Path
# ------     ---------------------
DISKVOL1     /diskvols/DISKVOL1
DISKVOL2     /diskvols/DISKVOL2
...
DISKVOL15     /diskvols/DISKVOL3
```

10. 为新文件系统创建挂载点目录，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 */samms* 挂载点目录并将权限设置为 *755* (*-rwxr-xr-x*)：

```
root@solaris:~# mkdir /samms
root@solaris:~# chmod 755 /samms
```

11. 指示 Oracle HSM 软件重新读取 *mcf* 文件并相应地重新配置自身。更正报告的任何错误并根据需要重复

```
root@solaris:~# /opt/SUNWsamfs/sbin/samd config
Configuring SAM-FS
root@solaris:~#
```

12. 接下来，挂载归档文件系统。

挂载归档文件系统

1. 以 *root* 用户身份登录到文件系统主机。登录全局区域（如果主机已配置区域）。
2. 备份 Solaris */etc/vfstab* 文件，然后在文本编辑器中将其打开。

   在示例中，使用 *vi* 编辑器。

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
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       -
```

```
...
samms      -        /samms    samfs   -     yes      -
```

3. 设置上限，即导致 Oracle HSM 从磁盘中释放先前归档文件的磁盘高速缓存利用率百分比。在 Oracle HSM 文件系统条目的最后一列中，输入挂载选项 *high=percentage*，其中 *percentage* 是 [*0-100*] 范围内的一个数字。

   根据磁盘存储容量、平均文件大小以及在任意给定时间所访问的估计文件数量设置该值。您希望确保始终有足够的高速缓存空间用于存储用户创建的新文件以及用户需要访问的归档文件。但您还希望尽量减少回写操作，以避免产生与挂载可移除介质卷相关的开销。

   如果使用最新高速磁盘或固态设备来实施主高速缓存，请将上限值设置为 95%。否则使用 80-85%。在示例中，我们将上限设置为 85%：

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
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       -
...
samms      -        /samms    samfs   -     yes      high=85
```

4. 设置下限，即导致 Oracle HSM 从磁盘中停止释放先前归档文件的磁盘高速缓存利用率百分比。在 Oracle HSM 文件系统条目的最后一列中，输入挂载选项 *low=percentage*，其中 *percentage* 是 [*0-100*] 范围内的一个数字。

   根据磁盘存储容量、平均文件大小以及在任意给定时间所访问的估计文件数量设置该值。出于性能原因，您希望尽量将最近的活动文件保留在高速缓存中，特别是在经常请求和修改文件的情况下。这样可以将与回写相关的开销降至最低。但您不希望先前高速缓存的文件占用新文件所需的空间以及必须从归档副本回写到磁盘的新访问文件所需的空间。

   如果使用最新高速磁盘或固态设备来实施主高速缓存，请将下限值设置为 90%。否则使用 70-75%。在示例中，我们根据本地要求将上限设置为 75%：

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
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       -
```

```
...
samms      -        /samms    samfs   -     yes      high=85,low=75
```

5. 如果在从磁盘释放先前归档的文件时用户需要将某些文件数据保留在磁盘中，请在 Oracle HSM 文件系统条目的最后一列中输入部分释放挂载选项。

   部分释放可以使 Oracle HSM 在释放归档文件以恢复磁盘空间时将指定文件的第一部分保留在磁盘高速缓存中。该方法将使应用程序能够即时访问文件开始部分的数据，而其余内容将从归档介质（如磁带）进行回写。以下挂载选项用于控制部分释放：

   - *maxpartial=value* 设置在文件部分释放到 *value* 时可以保留在磁盘高速缓存中的最大文件数据量，其中 *value* 是范围在 *0-2097152* 范围内的千字节数（为 *0* 时将禁用部分释放）。 默认值为 *16*。
   - *partial=value* 设置将某个文件部分释放到 *value* 之后仍位于磁盘高速缓存内的默认文件数据量，其中 *value* 是一个介于 [*0-maxpartial*] 范围内的千字节数。默认值为 *16*。但请注意，某个文件的保留部分始终使用一个至少等于一个磁盘分配单元 (Disk Allocation Unit, DAU) 的千字节数。
   - *partial_stage=value* 设置将整个部分释放的文件回写到 *value* 之前必须读取的最低文件数据量，其中 *value* 是一个介于 [*0-maxpartial*] 范围内的千字节数。默认值为 *-o partial* 指定的值（如果设置）或 *16*。
   - *stage_n_window=value* 设置可在任何时间从文件（直接从磁带介质中读取该文件）中读取而无需自动回写的最大数据量。指定的 *value* 是一个介于 [*64-2048000*] 范围内的千字节数。默认值为 *256*。

     有关从磁带介质中直接读取的文件的更多信息，请参见 *stage* 手册页 *-n* 下面的 *OPTIONS*（选项）一节。

   在示例中，基于应用程序的特征，我们将 *maxpartial* 设置为 *128*，将 *partial* 设置为 *64*；否则接受默认值：

```
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       -
...
samms      -        /samms    samfs   -     yes      ... maxpartial=128,partial=64
```

6. 如果需要从归档中排除 QFS 文件系统，请将 *nosam* 挂载选项添加到每个 QFS 文件的 */etc/vfstab* 条目。

   在本示例中，我们为 *DISKVOL1* 文件系统（此为磁盘归档）设置了 *nosam* 选项。在此处，*nosam* 挂载选项可确保归档副本不会对自身进行归档：

```
#File
#Device           Device   Mount               System  fsck  Mount    Mount
#to Mount         to fsck  Point               Type    Pass  at Boot  Options
#--------         -------  ------------------  ------  ----  -------  ---------
/devices          -        /devices            devfs   -     no       -
...
samms             -        /samms              samfs   -     yes      ... ,partial=64
DISKVOL1          -        /diskvols/DISKVOL1  samfs   -     yes      nosam
server:/DISKVOL2  -        /diskvols/DISKVOL2  nfs     -     yes
...
server:/DISKVOL15 -        /diskvols/DISKVOL15 nfs     -     yes
```

7. 保存 */etc/vfstab* 文件并关闭编辑器。

```
...
server:/DISKVOL15 -        /diskvols/DISKVOL15 nfs     -     yes
:wq
root@solaris:~#
```

8. 挂载 Oracle HSM 归档文件系统

```
root@solaris:~# mount /samms
```

9. 接下来，配置归档过程。

配置归档过程

创建并挂载归档文件系统后，通常只需完成少量的其他配置即可满足您的所有或大多数归档要求。在大多数情况下，您只需创建一个文本文件 *archiver.cmd*，该文件可标识文件系统，为您指定各个归档副本数量并为每个副本分配介质卷。

虽然 Oracle HSM 归档过程提供了许多可调参数，但如果没有定义明确的特殊要求，则通常应接受默认设置。这些默认设置均经过仔细挑选，可以在尽可能广泛的情况下最大限度减少介质挂载数量，最大限度提高介质利用率以及优化端到端归档性能。因此，如果需要进行调整，请特别注意会不必要地限制归档程序自由安排工作和选择介质的任何更改。如果尝试对存储操作进行微观管理，有时可能会大幅降低性能和总体效率。

但是，几乎在所有情况下，您都应启用归档日志记录。默认情况下不启用归档日志记录，因为日志文件如未得到妥善管理，可能会过大（在《Oracle Hierarchical Storage Manager and StorageTek QFS Software 维护和管理指南》中介绍了如何管理日志文件）。但是，如果文件系统损坏或丢失，归档日志文件可帮助您恢复文件，而您通常无法通过其他方式轻松恢复此类文件。如果您配置文件系统保护，则恢复点文件中的文件系统元数据可帮助您基于归档副本中存储的数据快速重新构建一个文件系统。但是，在文件系统损坏或丢失之前但在最后一个恢复点生成之后，不可避免地会归档几

个文件。在这种情况下，归档介质会保留有效副本，但如果缺少文件系统元数据，则无法自动定位这些副本。由于文件系统的归档日志中记录了保留每个归档副本的介质的卷序列号以及每个卷内相应 *tar* 文件的位置，因此可使用 *tar* 实用程序来恢复这些文件并完全恢复文件系统。

要创建 *archiver.cmd* 文件并配置归档过程，请执行如下操作：

1. 以 *root* 用户身份登录主机。

```
root@solaris:~#
```

2. 在文本编辑器中打开新的 */etc/opt/SUNWsamfs/archiver.cmd* 文件。

   archiver.cmd 中的每一行都由通过空格分隔的一个或多个字段组成（忽略前导空格）。

   在示例中，使用 *vi* 编辑器打开该文件并输入一条注释：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
# Configuration file for archiving file systems
```

3. 在 *archiver.cmd* 文件的开头，输入所需的任何常规归档指令。

   常规指令在第二个字段内包含等号 (=) 字符或不包含其他字段。在大多数情况下，您可以使用默认值而无需设置常规指令（有关详细信息，请参见 *archiver.cmd* 手册页的 *GENERAL DIRECTIVES SECTION*（常规指令部分））。

   在示例中，虽然可以将此部分留空，但我们还是输入了两个常规指令的默认值以便说明此类指令的格式：

   - *archivemeta = off* 指令告知归档过程不应归档元数据。
   - *examine = noscan* 指令告知归档过程当文件系统报告文件被更改时检查需要归档的文件（默认值）。

     早期版本的 Oracle HSM 会定期扫描整个文件系统。通常，您不应更改此指令，除非为与先前 Oracle HSM 配置相兼容而必须这样做。

```
# Configuration file for archiving file systems
#---------------------------------------------------------------------
# General Directives
archivemeta = off                                                 # default
examine = noscan                                                  # default
```

4. 输入所需的所有常规归档指令后，着手将文件分配到归档集。在一个新行上，输入分配指令 *fs = filesystem-name*，其中 *filesystem-name* 是 */etc/opt/SUNWsamfs/mcf* 文件中定义的某个文件系统的系列集名称。

分配指令将指定文件系统中的一组文件映射到归档介质上的一组副本。一组文件既可以像所有文件系统那样大，也可以像几个文件那样小。但为获得最佳性能和效率，不应过度指定。所创建的归档集数量不应多于实际需求量，否则会导致过多的介质挂载量、不必要的介质重新定位以及较低的总体介质利用率。在大多数情况下，只需为每个文件系统分配一个归档集。

在示例中，我们针对归档文件系统 *samms* 启动归档集分配指令：

```
# Configuration file for archiving file systems
#---------------------------------------------------------------------
# General Directives
archivemeta = off                                                  # default
examine = noscan                                                   # default
#---------------------------------------------------------------------
# Archive Set Assignments
fs = samms                                          # Archiving File System
```

5. 在下一行上，启用归档日志记录。输入 *logfile = path/filename* 指令，其中 *path/filename* 用于指定位置和文件名。

   如上文所述，归档日志数据对完全恢复丢失的文件系统至关重要。因此，将 Oracle HSM 配置为向一个非 Oracle HSM 指令（例如 */var/adm/*）写入归档程序日志并定期保存副本。虽然您可以创建一个全局 *archiver.log* 为所有文件系统一起记录归档程序活动，但分别为每个文件系统配置一个日志更便于在文件恢复过程中搜索相关日志。因此，在示例中，使用文件系统分配指令来指定 */var/adm/samms.archiver.log*：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
#---------------------------------------------------------------------
# Archive Set Assignments
fs = samms                                           # Archiving File System
logfile = /var/adm/samms.archiver.log
```

6. 在下一行上，将此文件系统中的文件分配到归档集。针对需要创建的每个归档集，输入指令 *archiveset-name starting-directory expression*，其中：
   - *archiveset-name* 是您为新归档集选择的名称。
   - *starting-directory* 是 Oracle HSM 从中开始搜索文件的目录的路径（相对于文件系统挂载点而言）。
   - *expression* 是由 Solaris *find* 命令定义的一个布尔型表达式。

   在大多数情况下，应使归档集定义尽可能宽泛而简单。但请注意，如遇特殊情况，可通过指定其他更加严格的限定符（例如用户或组文件拥有权、文件大小、文件

日期/时间戳和文件名（使用正则表达式））对归档集成员加以限制。有关完整信息，请参见 *archiver.cmd* 手册页。

在示例中，我们将在 *samms* 文件系统中找到的所有文件都输入到一个名为 *allsamms* 的归档集中。我们使用一个点 (*.*) 来指定路径，以便在自身挂载点目录 (*/samms*) 中进行搜索。

```
...
#-----------------------------------------------------------------------
# Archive Set Assignments
fs = samms                                      # Archiving File System
logfile = /var/adm/samms.archiver.log
allsamms .
```

7. 接下来，为 *samms* 文件系统的 *allsamms* 归档集添加复制指令。针对每个副本，以一个或多个空格开始行输入，然后输入指令 *copy-number -release -norelease archive-age unarchive-age*，其中：
   - *copy-number* 是一个整数。
   - *-release* 和 *-norelease* 是可选参数，用于控制进行复制后如何控制磁盘高速缓存空间。生成相应副本之后，*-release* 会自行释放磁盘空间。生成设有 *-norelease* 的所有副本并运行释放程序进程之前，*-norelease* 会自行阻止磁盘空间释放。生成设有 *-norelease* 的所有副本之后，*-release* 和 *-norelease* 会联合自动释放磁盘高速缓存空间。
   - *archive-age* 是从文件上次修改时间到文件归档之前必须经过的时间。将时间表示为整数和标识符 *s*（秒）、*m*（分）、*h*（小时）*d*（天）、*w*（周）和*y*（年）的任意组合。默认设置为 *4m*。
   - *unarchive-age* 是从文件上次修改时间到文件可取消归档之前必须经过的时间。默认值为从不取消归档副本。

   为实现完全冗余，始终至少为每个归档集指定两个副本（最多四个）。在示例中，我们指定三个副本，其中每个副本都设有 *-norelease*，直到该副本达到归档时间（15 分钟）。将使用磁盘归档卷生成副本 *1*，而副本 *2* 和副本 *3* 则生成到磁带介质：

```
...
#-----------------------------------------------------------------------
# Archive Set Assignments
fs = samms                                      # Archiving File System
logfile = /var/adm/samms.archiver.log
allsamms .
    1 -norelease 15m
    2 -norelease 15m
    3 -norelease 15m
```

8. 为所有剩余文件系统定义归档集。

   在示例中，我们将 QFS 文件系统 *DISKVOL1* 配置为复制过程的归档介质。因此，我们为 *fs = DISKVOL1* 启动一个条目。但我们不希望生成归档副本的归档副本。所以我们不指定日志文件，我们使用一个名为 *no_archive* 的特殊归档集，以阻止归档此文件系统中的文件：

```
...
#------------------------------------------------------------------------
# Archive Set Assignments
fs = samms                                        # Archiving File System
logfile = /var/adm/samms.archiver.log
allsamms .
    1 -norelease 15m
    2 -norelease 15m
    3 -norelease 15m
fs = DISKVOL1                             # QFS File System (Archival Media)
no_archive .
```

9. 接下来，我们输入一些用于控制副本创建方式的指令。在一个新行上，通过输入关键字 *params* 开始复制参数部分。

```
...
fs = DISKVOL1                              # QFS File System (Archival Media)
no_archive .
#------------------------------------------------------------------------
# Copy Parameter Directives
params
```

10. 如果您需要设置适用于所有归档集的所有副本的任何通用复制参数，请按以下格式输入一行：*allsets -param value ...*。其中，*allsets* 是代表所有已配置归档集的特殊归档集，*-param value ...* 代表通过空格分隔的一个或多个参数/值对。

    有关参数及参数值的完整说明，请参见 *archiver.cmd* 手册页的 *ARCHIVE SET COPY PARAMETERS SECTION*（归档集复制参数部分）一节。

    示例中的指令对于大多数文件系统而言都是最理想的选择。特殊 *allsets* 归档集可确保统一处理所有归档集，从而获得最佳性能并简化管理。*-sort path* 参数可确保适用于所有归档集的所有副本的磁带归档 (*tar*) 文件按路径进行排序，这样一来，位于相同目录中的文件在归档介质上仍然保存在一起。*-offline_copy stageahead* 参数可在归档脱机文件时改善性能：

```
...
#------------------------------------------------------------------------
# Copy Parameter Directives
```

```
params
allsets -sort path -offline_copy stageahead
```

11. 如果您需要为所有归档集中的特定副本设置复制参数，请按以下格式输入一行：*allfiles.copy-number -param value ...*。其中，*allsets* 是代表所有已配置归档集的特殊归档集，*copy-number* 是该指令将应用到的副本的数量，*-param value ...* 代表通过空格分隔的一个或多个参数/值对。

    有关参数及参数值的完整说明，请参见 *archiver.cmd* 手册页的 *ARCHIVE SET COPY PARAMETERS SECTION*（归档集复制参数部分）一节。

    在示例中，指令 *allfiles.1 -startage 10m -startsize 500M -drives 10 -archmax 1G* 可优化磁盘卷的副本 *1*。当为归档选择的第一个文件已等待 10 分钟，或所有等待文件的总大小至少为 500 兆字节时，该指令开始归档。最多可使用 10 个驱动器生成副本，并且副本中的每个 *tar* 文件不得大于一千兆字节。

    其余两个指令 *allfiles.2 -startage 24h -startsize 20G -drives 2 -archmax 24G -reserve set* 和 *allfiles.3 -startage 48h -startsize 20G -drives 2 -archmax 24G -reserve set* 可分别优化磁带介质的副本 *2* 和副本 *3*。当为归档选择的第一个文件已分别等待 24 或 48 小时，或所有等待文件的总大小至少为 20 千兆字节时，这两个指令开始归档。最多可使用 2 个驱动器生成这些副本（可根据您的基础结构调整此数值），并且副本中的每个 *tar* 文件不得大于 24 千兆字节。*-reserve set* 可确保使用磁带介质生成每个归档集的副本 *2* 和副本 *3*，并且仅包含来自同一个归档集的副本：

```
...
#------------------------------------------------------------------------
# Copy Parameter Directives
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 10m -startsize 500M -drives 10 -archmax 1G
allfiles.2 -startage 24h -startsize 20G -drives 2 -archmax 24G -reserve set
allfiles.3 -startage 48h -startsize 20G -drives 2 -archmax 24G -reserve set
```

    请注意，本节中的示例假定使用磁盘卷进行归档。如果只使用磁带卷，请指定两个副本并且更加频繁地归档到磁带。根据您的基础结构调整指定的驱动器数量之后，以下配置对于大多数文件系统而言都是最理想的选择：

```
allsets -sort path -offline_copy stageahead -reserve set
allfiles.1 -startage 8h -startsize 8G -drives 2 -archmax 10G
allfiles.2 -startage 24h -startsize 20G -drives 2 -archmax 24G
```

12. 如果您需要为特定归档集设置一个指令并复制，请按以下格式输入一行：*archive-set-name.copy-number -param value ...*。其中，*archive-*

set-name 是用于该归档集的名称，*copy-number* 是该指令将应用到的副本的数量，*-param value ...* 代表通过空格分隔的一个或多个参数/值对。

有关参数及参数值的完整说明，请参见 *archiver.cmd* 手册页的 *ARCHIVE SET COPY PARAMETERS SECTION*（归档集复制参数部分）一节。

在以下示例中，我们为 *corpfs* 文件系统定义了两个归档集：*hq* 和 *branches*。请注意，*hq.1* 和 *hq.2* 的复制指令仅适用于归档集 *hq*。归档集 *branches* 不受影响：

```
#------------------------------------------------------------------------
# Archive Set Assignments
fs = corpfs
logfile = /var/adm/corporatefs.archive.log
hq /corpfs/hq/
    1 -norelease 15m
    2 -norelease 15m
branches /corpfs/branches/
    1 -norelease 15m
    2 -norelease 15m
#------------------------------------------------------------------------
# Copy Parameter Directives
params
hq.1 -drives 4
hq.2 -drives 2
```

13. 设置完所有所需的复制参数之后，通过在一个新行上输入 *endparams* 关键字来关闭复制参数列表：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
#------------------------------------------------------------------------
# Copy Parameter Directives
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 10m -startsize 500M -drives 10 -archmax 1G
allfiles.2 -startage 24h -startsize 20G -drives 2 -archmax 24G -reserve set
allfiles.3 -startage 48h -startsize 20G -drives 2 -archmax 24G -reserve set
endparams
```

14. 或者，您也可以通过输入 *vsnpools* 关键字和以下格式的一个或多个指令来定义介质池：*pool-name media-type volumes*。其中，*pool-name* 是您为该介质池分配的名称，*media-type* 是在附录 A, 设备类型词汇表中定义的一个介质类型代码，*volumes* 是一个与一个或多个卷序列号 (Volume Serial Number, VSN) 相匹配的正则表达式。使用 *endvsnpools* 关键字关闭指令列表。

介质池可选，而且您通常不希望限制可供归档过程使用的介质。因此，在这些示例中，我们不定义介质池。有关更多信息，请参见 *archiver.cmd* 手册页的 *VSN POOL DEFINITIONS SECTION*（VSN 池定义部分）。

15. 接下来，开始标识归档集副本应使用的归档介质。在一个新行上，输入关键字 *vsns*：

```
...
#------------------------------------------------------------------------
# Copy Parameter Directives
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 10m -startsize 500M -drives 10 -archmax 1G
allfiles.2 -startage 24h -startsize 20G -drives 2 -archmax 24G -reserve set
allfiles.3 -startage 48h -startsize 20G -drives 2 -archmax 24G -reserve set
endparams
#------------------------------------------------------------------------
# VSN Directives
vsns
```

16. 通过以 *archive-set-name.copy-number media-type volumes* 格式输入一行来指定每个归档集副本的介质，其中，*archive-set-name.copy-number* 指定指令所应用于的归档集和副本，*media-type* 是在附录 A, 设备类型词汇表中定义的介质类型代码之一，*volumes* 是与一个或多个卷序列号 (volume serial number, VSN) 相匹配的正则表达式。

    为实现完全冗余，始终将每个归档集副本分配给不同的介质范围，以便两个副本永不驻留在同一物理卷上。如果可能，始终将至少一个副本分配给可移除介质（如磁带）。

    在示例中，我们将每个归档集的第一个副本发送到卷序列号介于 *DISKVOL1* 到 *DISKVOL15* 范围内的归档磁盘介质（类型 *dk*）。我们将每个归档集的第二个副本发送到卷序列号介于 *VOL000* 到 *VOL199* 范围内的磁带介质（类型 *tp*），将第三个副本发送到卷序列号介于 *VOL200* 到 *VOL399* 范围内的磁带介质（类型 *tp*）：

```
...
#------------------------------------------------------------------------
# Copy Parameter Directives
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 10m -startsize 500M -drives 10 -archmax 1G
allfiles.2 -startage 24h -startsize 20G -drives 2 -archmax 24G -reserve set
allfiles.3 -startage 48h -startsize 20G -drives 2 -archmax 24G -reserve set
endparams
#------------------------------------------------------------------------
```

```
# VSN Directives
vsns
allfiles.1 dk DISKVOL[1-15]
allfiles.2 tp VOL[0-1][0-9][0-9]
allfiles.2 tp VOL[2-3][0-9][0-9]
```

17. 指定所有归档集副本的介质后，通过在一个新行上输入 *endvsns* 关键字来关闭 *vsns* 指令。保存文件并关闭编辑器。

```
...
#------------------------------------------------------------------------
# Copy Parameter Directives
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 10m -startsize 500M -drives 10 -archmax 1G
allfiles.2 -startage 24h -startsize 20G -drives 2 -archmax 24G -reserve set
allfiles.3 -startage 48h -startsize 20G -drives 2 -archmax 24G -reserve set
endparams
#------------------------------------------------------------------------
# VSN Directives
vsns
allfiles.1 dk DISKVOL[1-15]
allfiles.2 tp VOL[0-1][0-9][0-9]
allfiles.2 tp VOL[2-3][0-9][0-9]
endvsns
:wq
root@solaris:~#
```

18. 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。示例中有一个错误：

```
root@solaris:~# archiver -lv
Reading '/etc/opt/SUNWsamfs/archiver.cmd'.
...
13: # File System Directives
14: #
15: fs = samms
16: logfile = /var/adm/samms.archiver.log
17: all .
18:     1 -norelease 15m
19:     2 -norelease 15m
20: fs=DISKVOL1                          # QFS File System (Archival Media)
21:
```

```
...
42: endvsns
DISKVOL1.1 has no volumes defined
1 archive set has no volumes defined
root@solaris:~#
```

19. 如果在 *archiver.cmd* 文件中发现错误，请对其进行更正，然后重新检查该文件。

    在上面的示例中，我们忘记将 *no_archive* 指令输入到文件系统指令 *DISKVOL1* 中（即配置为磁盘归档的 QFS 文件系统）。更正该疏忽后，*archiver -lv* 运行且无错误：

```
root@solaris:~# archiver -lv
Reading '/etc/opt/SUNWsamfs/archiver.cmd'.
...
20: fs=DISKVOL1                          # QFS File System (Archival Media)
21: no_archive .
...
42: endvsns
Notify file: /etc/opt/SUNWsamfs/scripts/archiver.sh
...
allfiles.1
    startage: 10m startsize: 500M drives 10: archmax: 1G
 Volumes:
  DISKVOL1 (/diskvols/DISKVOL15)
  ...
  DISKVOL15 (/diskvols/DISKVOL3)
 Total space available:   150T
allfiles.2
    startage: 24h startsize: 20G drives: 2 archmax: 24G reserve: set
 Volumes:
   VOL000
...
   VOL199
 Total space available:  300T
allfiles.3
    startage: 48h startsize: 20G drives: 2 archmax: 24G reserve: set
 Volumes:
   VOL200
...
   VOL399
 Total space available:  300T
root@solaris:~#
```

20. 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 *samd config* 命令。

```
root@solaris:~# /opt/SUNWsamfs/sbin/samd config
Configuring SAM-FS
root@solaris:~#
```

21. 在文本编辑器中打开 */etc/opt/SUNWsamfs/releaser.cmd* 文件，添加 *list_size = 300000* 行，保存文件并关闭编辑器。

    list_size 指令将可从文件系统同时释放的文件数设置为一个介于 [*10-2147483648*] 范围内的整数。如果 *.inodes* 文件中有足够的空间供一百万个 inode 使用（每个 inode 允许 512 个字节），默认值则为 *100000*。否则，默认值为 *30000*。将此数字增加到 *300000* 可以更好地符合包含大量小文件的典型文件系统。

    在示例中，使用 *vi* 编辑器：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/releaser.cmd
# releaser.cmd
logfile = /var/opt/SUNWsamfs/releaser.log
list_size = 300000
:wq
root@solaris:~#
```

22. 在文本编辑器中打开 */etc/opt/SUNWsamfs/stager.cmd* 文件，添加 *maxactive = stage-requests* 行，其中，在 RAM 大于等于 8 千兆字节和小于 8 千兆字节的主机上，*stage-requests* 分别为 *500000* 和 *100000*。保存文件并关闭编辑器。

    maxactive 指令将可同时激活的最大回写请求数设置为一个介于 [*1-500000*] 范围内的整数。默认情况是允许每千兆字节主机内存 5000 个回写请求。

    在示例中，使用 *vi* 编辑器：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/stager.cmd
# stager.cmd
logfile = /var/opt/SUNWsamfs/stager.log
maxactive = 300000
:wq
root@solaris:~#
```

23. 默认情况下不启用回收。因此，如果需要回收可移除介质卷，请转到“配置回收过程”。
24. 如果归档 Oracle HSM 文件系统的 *mcf* 文件在归档设备部分中包括网络连接磁带库，请转到“存储在网络连接磁带库中的目录归档介质”。

25. 如果您需要具备验证归档磁带卷完整性的能力，请转到“配置归档介质验证”。
26. 否则，“配置文件系统保护”。

配置回收过程

当可移除介质卷包含的有效归档集少于用户指定的有效归档集时，回收程序将合并其他卷上的有效数据，以便原始卷可以导出以供长期存储或可重新添加标签以供重新使用。可通过两种方式之一配置回收：

- 配置按归档集回收

  按归档集回收介质时，会将回收指令添加到 *archiver.cmd* 文件中。您可以精确地指定如何回收每个归档集中的介质。回收条件的应用范围更小，因为仅考虑归档集成员。

  如果可能，请按归档集而非库回收介质。在 Oracle HSM 归档文件系统中，从逻辑上看，回收属于文件系统操作而非库管理的一部分。回收补充归档、释放和回写。因此，可以将其配置为归档过程的一部分。请注意，如果您的配置包括磁盘归档卷和/或 SAM-Remote，则必须按归档集配置回收。
- 配置按库回收

  按库回收介质时，会将回收指令添加到 *recycler.cmd* 文件中。因此，您可以为指定的库中包含的所有介质设置公共回收参数。回收指令适用于库中的所有卷，因此，它们的固有细化程度低于归档集特定指令。您可以从检查中显式排除指定的卷序列号 (volume serial number, VSN)。但在其他方面，回收过程只寻找包含未识别为当前有效归档文件的卷。

  因此，按库进行回收会销毁不是待回收文件系统组成部分的文件。如果回收指令未显式排除它们，有用数据（例如，来自其他文件系统的归档日志和库目录或归档介质的备份副本）可能面临风险。为此，如果您使用的是 SAM-Remote，则无法按库进行回收。库中由 SAM-Remote 服务器控制的卷包含客户机而非服务器所拥有的外部归档文件。

配置按归档集回收

1. 以 *root* 用户身份登录到 Oracle HSM 文件系统主机。

```
root@solaris:~#
```

2. 在文本编辑器中打开 */etc/opt/SUNWsamfs/archiver.cmd* 文件，向下滚动到副本 *params* 部分。

   在示例中，使用 *vi* 编辑器。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
```

```
#------------------------------------------------------------------------
# Copy Parameter Directives
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 6h  -startsize 6G  -startcount 500000
allfiles.2 -startage 24h -startsize 20G -startcount 500000 -drives 5
```

3. 在 *archiver.cmd* 文件的 *params* 部分中，按归档集输入您的回收程序指令，格式为 *archive-set directive-list*，其中，archive-set 是其中一个归档集，*directive-list* 是指令名称/值对的空格分隔列表（有关回收指令的列表，请参见 *archiver.cmd* 手册页）。然后保存文件并关闭编辑器。

   在示例中，我们为归档集 *allfiles.1* 和 *allfiles.2* 添加回收指令。*-recycle_mingain 30* 和 *-recycle_mingain 90* 指令不回收卷，除非可分别恢复至少 30% 和 90% 的卷容量。当可移除介质的已用容量达到 60% 时，*-recycle_hwm 60* 指令开始回收。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
#------------------------------------------------------------------------
# Copy Parameters Directives
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 6h  -startsize 6G -startcount 500000
allfiles.1 -recycle_mingain 30 -recycle_hwm 60
allfiles.2 -startage 6h -startsize 6G -startcount 500000
allfiles.2 -recycle_mingain 90 -recycle_hwm 60
endparams
#------------------------------------------------------------------------
# VSN Directives
vsns
allfiles.1 dk DISKVOL1
allfiles.2 tp VOL0[0-1][0-9]
endvsns
:wq
[root@solaris:~#
```

4. 检查 *archiver.cmd* 文件中的错误。使用命令 *archiver -lv*。

   archiver -lv 命令读取 *archiver.cmd*，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。在本示例中，该文件不包含任何错误：

```
root@solaris:~# archiver -lv
Reading '/etc/opt/SUNWsamfs/archiver.cmd'.
...
```

```
   VOL399
 Total space available:  300T
root@solaris:~#
```

5. 如果在 *archiver.cmd* 文件中发现错误，请对其进行更正，然后重新检查该文件。
6. 在文本编辑器中创建 *recycler.cmd* 文件。指定回收程序日志的路径和文件名。然后保存文件并关闭编辑器。

   配置 Oracle HSM 以将日志写入到非 Oracle HSM 目录，例如 */var/adm/*。在示例中，使用 *vi* 编辑器并指定 */var/adm/recycler.log*：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
logfile = /var/adm/recycler.log
:wq
root@solaris:~#
```

7. 在文本编辑器中打开 */etc/opt/SUNWsamfs/scripts/recycler.sh* 脚本，然后输入用于处理已回收的可移除介质卷的 shell 命令。

   当回收过程标识已耗尽有效归档副本的可移除介质卷时，将调用 *recycler.sh* 文件（用于处理已回收介质的 C-shell 脚本）。

   编辑该文件以执行所需任务，从通知管理员卷已准备就绪可进行回收到重新为卷设置标签以供重新使用或从库导出卷以便长期历史保留。

   默认情况下，脚本提醒 *root* 用户设置该脚本。
8. 如果归档 Oracle HSM 文件系统的 *mcf* 文件在归档设备部分中包括网络连接磁带库，请转到“存储在网络连接磁带库中的目录归档介质”。
9. 否则，请转至“配置文件系统保护”。

配置按库回收

1. 以 *root* 用户身份登录到 Oracle HSM 文件系统主机。

```
root@solaris:~#
```

2. 在文本编辑器中创建 */etc/opt/SUNWsamfs/recycler.cmd* 文件。

   在示例中，使用 *vi* 编辑器。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
# Configuration file for archiving file systems
#----------------------------------------------------------------------
```

3. 使用 *logfile* 指令指定回收程序日志的路径和文件名。

配置 Oracle HSM 以将日志写入到非 Oracle HSM 目录，例如 */var/adm/*。在示例中，我们指定 */var/adm/recycler.log*：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
# Configuration file for archiving file systems
#---------------------------------------------------------------------
logfile = /var/adm/recycler.log
```

4. 如果归档介质库中存在任何不得回收的卷，请输入指令 *no_recycle media-type volumes*，其中，*media-type* 是在附录 A, 设备类型词汇表中定义的介质类型代码之一，*volumes* 是与一个或多个卷序列号 (volume serial number, VSN) 相匹配的正则表达式。

   在示例中，我们对 [*VOL020-VOL999*] 范围内的卷禁用回收：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
# Configuration file for archiving file systems
#---------------------------------------------------------------------
logfile = /var/adm/recycler.log
no_recycle tp VOL[0-9][2-9][0-9]
```

5. 在一个新行上，输入指令 *library parameters*，其中，*library* 是 */etc/opt/SUNWsamfs/mcf* 文件分配给可移除介质库的系列集名称，*parameters* 是从以下列表提取的参数/值对的空格分隔列表：
   - *-dataquantity size* 将可计划用于同时重新归档的最大数据量设置为 *size*，其中，*size* 是字节数。默认值为 1 GB。
   - *-hwm percent* 设置库的上限（触发回收时已用的总介质容量百分比）。上限以 *percent*（[*0-100*] 范围内的数字）指定。默认值为 *95*。
   - *-ignore* 防止回收此库，因此您可以对 *recycler.cmd* 文件进行无损测试。
   - *-mail address* 向 *address* 发送回收消息，其中，*address* 是有效电子邮件地址。默认情况下，不发送任何消息。
   - *-mingain percent* 将回收限制为可至少按最小量增加可用空间的卷，以总容量的百分比表示。最小增益以 *percent*（[*0-100*] 范围内的一个数字）指定。对于总容量低于 200 千兆字节的卷，默认值为 *60*；对于容量大于等于 200 千兆字节的卷，默认值则为 *90*。
   - *-vsncount count* 将可计划用于同时重新归档的最大卷数设置为 *count*。默认值为 *1*。

   在示例中，我们将库 *library1* 的上限设置为 95%，而且要求每个磁带的最小容量增益为 60%：

```
root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
# Configuration file for archiving file systems
```

```
#------------------------------------------------------------------------
logfile = /var/adm/recycler.log
no_recycle tp VOL[0-9][2-9][0-9]
library1 -hwm 95 -mingain 60
```

6. 对 Oracle HSM 配置中的任何其他库重复之前步骤。然后保存 *recycler.cmd* 文件并关闭编辑器。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
# Configuration file for archiving file systems
#------------------------------------------------------------------------
logfile = /var/adm/recycler.log
no_recycle tp VOL[0-9][2-9][0-9]
library1 -hwm 95 -mingain 60
:wq
root@solaris:~#
```

7. 如果归档 Oracle HSM 文件系统的 *mcf* 文件在归档设备部分中包括网络连接磁带库，请转到"存储在网络连接磁带库中的目录归档介质"。
8. 否则，请转至“配置文件系统保护”。

存储在网络连接磁带库中的目录归档介质

挂载文件系统后，Oracle HSM 软件会为 *mcf* 文件中配置的每个自动化库创建目录。不过，如果您具有网络连接库，则必须执行一些额外的步骤以填充其目录。

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. 如果归档文件系统使用 Oracle StorageTek ACSLS 连接磁带库，请从库的临时池提取所需 Oracle HSM 归档介质并自动生成目录。使用 *samimport -c volumes -s pool* 命令，其中，*volumes* 是所需卷数，*pool* 是为库定义的临时介质池的名称。在此处停止。

   在示例中，我们请求从名为 *scratch* 的池提取 *20* 个磁带卷：

```
root@solaris:~# samimport -c 20 -s scratch
```

3. 如果归档文件系统使用配置为单个非共享逻辑库的 IBM 3494 网络连接库，请将所需磁带卷置于库中转槽中并让库自动为其创建目录。在此处停止。

当 *mcf* 文件的 *Additional Parameters* 字段指定 *access=private* 时，IBM 3494 库被配置为单个逻辑库。如果 *access=shared*，IBM 3494 库则分成多个逻辑库，而且您必须使用下面指定的方法。

4. 否则，如果归档文件系统使用共享 IBM 3494 网络连接库或任何其他网络连接库，请使用文本编辑器创建目录输入文件。

   在示例中，使用 *vi* 编辑器创建文件 *input3494cat*：

```
root@solaris:~# vi input3494cat
~
"~/input3494cat" [New File]
```

5. 通过输入记录 *index* 来启动记录。始终针对第一条记录输入 *0*（零），然后递增每个后续记录的索引。输入空格以指示字段的结尾。

   行定义 *build_cat* 输入文件中的记录和空格分隔字段。第一个字段 *index* 的值只是一个从 *0* 开始的连续整数，用于标识 Oracle HSM 目录内的记录。在示例中，这是第一条记录，因此我们输入 *0*：

```
0
~
"~/input3494cat" [New File]
```

6. 在记录的第二个字段中，输入磁带卷的卷序列号 (volume serial number, VSN)；如果没有 VSN，请输入一个 *?*（问号）。然后输入空格以指示字段的结尾。

   将包含空格字符（如果有）的值用双引号括住：*"VOL 01"*。在本示例中，第一个卷的 VSN 不包含空格：

```
0 VOL001
~
"~/input3494" [New File]
```

7. 在第三个字段中，输入卷的条形码（如果与卷序列号不同）、卷序列号；如果没有卷序列号，请输入字符串 *NO_BAR_CODE*。然后输入空格以指示字段的结尾。

   在示例中，第一个卷的条形码与 VSN 的值相同：

```
0 VOL001 VOL001
~
"~/input3494cat" [New File]
```

8. 最后，在第四个字段中，输入卷的介质类型。然后输入空格以指示字段的结尾。

介质类型为双字母代码，如 *li*（对于 LTO 介质）（有关介质设备类型的综合列表，请参见附录 A, 设备类型词汇表）。在示例中，使用包含 LTO 磁带机的 IBM 3494 网络连接磁带库，因此我们输入 *li* （包含终止空格）：

```
0 VOL001 VOL001 li
~
"~/input3494cat" [New File]
```

9. 针对要与 Oracle HSM 配合使用的每个卷，重复步骤 3-6 以创建其他记录。然后保存文件。

```
0 VOL001 VOL001 li
1 VOL002 VOL002 li
...
13 VOL014 VOL014 li
:wq
root@solaris:~#
```

10. 使用 *build_cat input-file catalog-file* 命令创建目录，其中，*input-file* 是输入文件的名称，*catalog-file* 是库目录的完整路径。

    如果已在 *mcf* 文件的 *Additional Parameters* 字段中指定目录名称，请使用该名称。否则，如果未创建目录，Oracle HSM 软件将使用文件名 *family-set-name* 在 */var/opt/SUNWsamfs/catalog/* 目录中创建默认目录，其中，*family-set-name* 是您在 *mcf* 文件中用于库的设备名称。在示例中，使用系列集 i*3494*：

```
root@solaris:~# build_cat input_vsns /var/opt/SUNWsamfs/catalog/i3494
```

11. 如果归档文件系统已共享，请对每个潜在元数据服务器重复之前步骤。

    归档文件系统现已完成，可以使用了。

12. 接下来，配置文件系统保护。

配置文件系统保护

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。

因此，需要对文件系统保护进行一些规划。另一方面，恢复点和日志文件副本的创建最好足够频繁，而且保留时间足够长，以便您能够尽可能恢复丢失或损坏的文件和文件系统。 另一方面，不需要在数据文件主动发生更改时创建恢复点和日志文件副本，而需要知道它们使用的磁盘空间（恢复点文件和日志非常大）。本节会相应地建议一种广泛适用的配置，该配置可以在不经过修改的情况下用于许多文件系统配置。需要进行更改时，建议的配置会说明一些问题并可作为良好的起点。本节其余部分提供创建和管理恢复点的说明。包含以下小节：

- 创建用于存储恢复点文件和归档程序日志副本的位置
- 自动创建恢复点并保存归档程序日志

创建用于存储恢复点文件和归档程序日志副本的位置

对于已配置的各个归档文件系统，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. 为恢复点文件选择存储位置。选择可挂载在文件系统主机上的独立文件系统。
3. 请确保选定文件系统具有足够的空间来存储新的恢复点文件以及计划在任何给定时间保留的恢复点文件数。

   恢复点文件非常大，必须存储大量此类文件，具体取决于这些文件的创建频率以及保留时间。
4. 确保所选文件系统未与归档文件系统共享任何物理设备。

   请勿将恢复点文件存储在用于起保护作用的文件系统中。请勿将恢复点文件存储在也托管归档文件系统的物理设备的逻辑设备（如分区或 LUN）上。
5. 在选定的文件系统中，创建用于保存恢复点文件的目录。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，*path* 是所选目录的路径和名称。

   请勿在一个 catch-all 目录中存储多个归档文件系统的恢复点文件。为每个恢复点文件创建一个单独的目录，以便组织恢复点文件并在必要时轻松找到它们。

   在示例中，要为归档文件系统 */samms* 配置恢复点。因此，在独立文件系统 */zfs1* 上创建了目录 */zfs1/samms_recovery*：

```
root@solaris:~# mkdir /zfs1/samms_recovery
```

6. 如果文件系统未与归档文件系统共享任何物理设备，则为文件系统创建用于存储归档程序日志的时间点副本的子目录。

   在示例中，我们选择在主机根文件系统的 */var* 目录中存储日志副本。我们为归档文件系统 */samms* 配置文件系统保护。因此，创建目录 */var/samms_archlogs*：

```
root@solaris:~# mkdir /var/samms_archlogs
```

7. 接下来，自动创建恢复点并保存归档程序日志。

自动创建恢复点并保存归档程序日志

虽然通过在 *crontab* 文件中创建条目或者通过使用 Oracle HSM Manager 图形用户界面的调度功能可以自动创建元数据恢复点文件，但后一种方法不会自动保存归档程序日志数据。因此，本节重点介绍 *crontab* 方法。如果您希望使用该图形用户界面来调度恢复点，请参阅 Manager 联机帮助。

以下过程将创建两个每天运行的 *crontab* 条目：一条用于删除过时的恢复点文件，然后创建新恢复点；另一条用于保存归档程序日志。对于已配置的各个归档文件系统，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. 打开 *root* 用户的 *crontab* 文件进行编辑。使用命令 *crontab -e*。

   crontab 命令可在 *EDITOR* 环境变量指定的文本编辑器中打开 *root* 用户 *crontab* 文件的可编辑副本（有关完整的详细信息，请参见 Solaris *crontab* 手册页）。在示例中，使用 *vi* 编辑器：

```
root@solaris:~# crontab -e
...
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
15 3 * * 0 [ -x /usr/lib/fs/nfs/nfsfind ] && /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
```

3. 首先，创建用于删除过期恢复点文件并创建新恢复点的条目。在新行中，指定工作的完成时间。输入 *minutes hour * * **，其中：
   - *minutes* 是一个介于 [*0-59*] 范围内的整数，用于指定作业开始时的分钟。
   - *hour* 是一个介于 [*0-23*] 范围内的整数，用于指定作业开始时的小时。
   - *（星号）指定未使用的值。

     对于每天运行的任务，不使用月中日期 [*1-31*]、月份 [*1-12*] 以及星期几 [*0-6*] 的值。
   - 空格用于分隔时间规范中的字段。
   - *minutes hour* 用于指定未创建或修改文件的时间。

     在文件系统活动量最少时创建恢复点文件可确保文件尽可能准确完整地反映归档状态。理想情况下，所有新文件和修改的文件都将在指定的时间之前进行归档。

   在示例中，我们将工作安排在每天凌晨 2:10 开始：

```
...
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 2 * * *
```

4. 继续在同一行中，输入用来清理旧恢复点文件的 shell 命令。输入文本 *(find directory -type f -mtime +retention -print | xargs -l1 rm -f;*，其中：
   - *(*（左括号）标记 *crontab* 条目将执行的命令序列开始。
   - *directory* 是存储恢复点文件的目录的路径和目录名，因此，也是 Solaris *find* 命令的搜索起点。
   - *-type f* 是用于指定文本文件（与块特殊文件、字符特殊文件、目录、管道等相对）的 *find* 命令选项。

- *-mtime +retention* 是用于指定超过 *retention*（一个整数，表示恢复点文件保留的小时数）还未加以修改的文件的 *find* 命令选项。
- *-print* 是用于将找到的所有文件进行标准输出的 *find* 命令选项。
- *|xargs -l1 rm -f* 可将输出从 *-print* 通过管道传输到 Solaris 命令 *xargs -l1*，该命令每次将一行作为参数发送到 Solaris 命令 *rm -f*，进而删除找到的所有文件。
- *;*（分号）标记命令行结尾。

在示例中，*crontab* 条目在目录 */zfs1/samms_recovery* 中搜索至少 72 小时（3 天）未进行修改的所有文件并将找到的所有文件删除。请注意，*crontab* 条目仍是单行－使用反斜杠对换行符进行转义：

```
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
15 3 * * 0 [ -x /usr/lib/fs/nfs/nfsfind ] && /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 2 * * * ( find /zfs1/samms_recovery -type f -mtime +72 -print | /
xargs -l1 rm -f;
```

5. 继续在同一行中，输入用来切换到要创建恢复点的目录的 shell 命令。输入文本 *cd mount-point;*，其中，*mount-point* 是归档文件系统的根目录，分号 (;) 标记命令行的结尾。

   用于创建恢复点文件的命令 *samfsdump* 可对当前目录及所有子目录中的所有文件的元数据进行备份。在本示例中，我们转至 */samms* 目录，这是我们要保护的文件系统的挂载点：

```
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
15 3 * * 0 [ -x /usr/lib/fs/nfs/nfsfind ] && /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 2 * * * ( find /zfs1/samms_recovery -type f -mtime +72 -print | /
xargs -l1 rm -f; cd /samms;
```

6. 继续在同一行中，输入用来每天创建新恢复点的 shell 命令。输入文本 */opt/SUNWsamfs/sbin/samfsdump -f directory/'date +/%y/%m/%d')*，其中：
   - */opt/SUNWsamfs/sbin/samfsdump* 是用于创建恢复点的命令（有关完整的详细信息，请参见手册页）。
   - *-f* 是用于指定将保存恢复点文件的位置的 *samfsdump* 命令选项。
   - *directory* 是我们创建的用于保存该文件系统恢复点的目录。

- *'date +/%y/%m/%d'* 是 Solaris *date* 命令和格式化模板，用于创建恢复点文件的名称：*YYMMDD*，其中 *YYMMDD* 是当前年份的最后两位数、当前月份的两位数编号以及两位数的日期（例如 *150122* 表示 2015 年 1 月 22 日）。
- ;（分号）标记命令行结尾。
- *)*（右括号）标记 *crontab* 条目将执行的命令序列的结尾。

在示例中，我们指定上面创建的恢复点目录 */zfs1/samms_recovery*。请注意，*crontab* 条目仍是单行－使用反斜杠对换行符进行转义：

```
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
15 3 * * 0 [ -x /usr/lib/fs/nfs/nfsfind ] && /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 2 * * * ( find /zfs1/samms_recovery -type f -mtime +72 -print | /
xargs -l1 rm -f; cd /samms ; /opt/SUNWsamfs/sbin/samfsdump /
-f /zfs1/samms_recovery/'date +/%y/%m/%d')
```

7. 现在创建用于保存归档程序日志的条目。在一个新行中输入 *minutes hour * * **来指定工作的完成时间，其中：
   - *minutes* 是一个介于 [*0-59*] 范围内的整数，用于指定作业开始时的分钟。
   - *hour* 是一个介于 [*0-23*] 范围内的整数，用于指定作业开始时的小时。
   - *（星号）指定未使用的值。

     对于每天运行的任务，不使用月中日期 [*1-31*]、月份 [*1-12*] 以及星期几 [*0-6*] 的值。
   - 空格用于分隔时间规范中的字段。
   - *minutes hour* 用于指定未创建或修改文件的时间。

   在示例中，我们将工作安排在每个星期天的凌晨 3:15 开始：

```
...
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 2 * * * ( find /zfs1/samms_recovery -type f -mtime +72 -print | /
xargs -l1 rm -f; cd /samms ; /opt/SUNWsamfs/sbin/samfsdump /
-f /zfs1/samms_recovery/'date +/%y/%m/%d')
15 3 * * 0
```

8. 继续在同一行中，输入用来将当前归档程序日志移动到某个备份位置并为其指定唯一名称的 shell 命令。输入文本 *(mv /var/adm/samms.archive.log /var/samms_archlogs/"date +%y%m%d";*。

   此步骤将保存活动日志文件中将被覆盖的日志条目。在本示例中，我们将 *samms* 文件系统的归档程序日志移动到我们选择的位置 */var/samms_archlogs/* 并将其

重命名为 *YYMMDD*，其中 *YYMMDD* 是当前年份的最后两位数、当前月份的两位数编号以及两位数的日期（例如 *150122* 表示 2015 年 1 月 22 日）：

```
...
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 2 * * * ( find /samms_recovery/dumps -type f -mtime +72 -print | xargs -l1 rm -f; / cd /samms ; /
opt/SUNWsamfs/sbin/samfsdump -f /zfs1/samms_recovery/'date +/%y/%m/%d')
15 3 * * 0 ( mv /var/adm/samms.archiver.log /var/samms_archlogs/"date +%y%m%d";
```

9. 继续在同一行中，输入用来对归档程序日志文件重新初始化的 shell 命令。输入文本 *touch /var/adm/samms.archive.log)*。

   在本示例中，请注意，*crontab* 条目仍是单行－使用反斜杠对换行符进行转义：

```
...
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 2 * * * ( find /samms_recovery/dumps -type f -mtime +72 -print | xargs -l1 rm -f; / cd /samms ; /
opt/SUNWsamfs/sbin/samfsdump -f /zfs1/samms_recovery/'date +/%y/%m/%d')
15 3 * * 0 ( mv /var/adm/samms.archive.log /var/samms_archlogs/"date +%y%m%d";/
touch /var/adm/samms.archiver.log )
```

10. 保存文件并关闭编辑器。

```
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
15 3 * * 0 [ -x /usr/lib/fs/nfs/nfsfind ] && /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 2 * * * ( find /samms_recovery/dumps -type f -mtime +72 -print | xargs -l1 rm -f; / cd /samms ; /
opt/SUNWsamfs/sbin/samfsdump -f /zfs1/samms_recovery/'date +/%y/%m/%d')
15 3 * * 0 ( mv /var/adm/samms.archive.log /var/samms_archlogs/"date +%y%m%d";/ touch /var/adm/samms
.archive.log )
:wq
root@solaris:~#
```

11. 如果需要在文件系统上启用 WORM（Write Once Read Many，一写多读）功能，请参见“启用对一次写入多次读取 (Write Once Read Many, WORM) 文件的支持”。
12. 如果需要与使用 LTFS 的系统交互，或者需要在远程站点之间传输大量数据，请参见“启用对 Linear Tape File System (LTFS) 的支持”。
13. 如果您需要具备验证归档磁带卷完整性的能力，请转到“配置归档介质验证”。
14. 如果有其他需求（例如，多主机文件系统访问或高可用性配置），请参见“进阶知识”。
15. 否则，请转至第 11 章 配置通知和日志记录。

配置归档介质验证

介质验证是一种使用 SCSI *verify* 命令对磁带介质的数据完整性进行评估的技术。主机上的 SCSI 驱动程序可计算其写入驱动器的逻辑数据块的 CRC 校验和并发送 *verify* 命令。驱动器读取数据块、计算其自己的校验和并将结果与驱动程序提供的值进行比较。如果存在差异，则返回错误。只要校验和完成，驱动器便会放弃所读取的数据，因此，主机上不会有与 I/O 相关的额外开销。

Oracle HSM 支持以两种方式进行介质验证：

- 可以配置 Oracle HSM 以支持数据完整性验证 (Data Integrity Validation, DIV)，从而验证 StorageTek T10000 磁带介质上的数据，即在 Oracle HSM 定期介质验证中手动或自动执行。
- 也可以配置 Oracle HSM 定期介质验证来自动验证 StorageTek T10000 磁带介质上的数据和其他格式（例如 LTO Ultrium）的数据。

配置 Oracle HSM 以支持数据完整性验证 (Data Integrity Validation, DIV)

数据完整性验证 (Data Integrity Validation, DIV) 是 Oracle StorageTek 磁带机的一项功能，可与 Oracle HSM 软件配合使用来确保存储数据的完整性。启用该功能时（*div = on* 或 *div = verify*），服务器主机和驱动器都会在 I/O 期间计算并比较校验和。写入操作期间，服务器计算每个数据块的四字节校验和，并将该校验和连同数据一起传递到驱动器。磁带机随后重新计算校验和并将结果与服务器提供的值进行比较。如果值一致，驱动器便将数据块和校验和写入磁带。在读取操作期间，驱动器和主机从磁带读取数据块及其关联校验和。每一项都会对数据块中的校验和进行重新计算，然后将结果与存储的校验和进行比较。如果校验和在任何时间都不匹配，驱动器会通知应用程序软件出现了错误。

div = verify 选项可在写入数据时提供附加保护层。完成写入操作后，主机会要求磁带机重新验证数据。因此，驱动器会重新扫描数据、重新计算校验和，然后将结果与磁带上存储的校验和进行比较。驱动器在内部执行所有操作，不需要额外的 I/O（放弃数据），因此，主机系统没有额外的开销。也可以使用 Oracle HSM *tpverify*（磁带检验）命令根据需要执行此步骤。

要配置数据完整性验证，请执行如下操作：

1. 以 *root* 用户身份登录到 Oracle HSM 服务器。

   在示例中，元数据服务器名为 *samfs-mds*：

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

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

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

```
5.11
[samfs-mds]root@solaris:~#
```

3. 请确保 Oracle HSM *mcf* 文件中定义的归档存储设备包含兼容的磁带机：StorageTek T10000C（最低固件级别为 1.53.315）或 T10000D。
4. 使所有归档进程（如果有）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：

   ```
   [samfs-mds]root@solaris:~# samcmd aridle
   [samfs-mds]root@solaris:~#
   ```

5. 使所有回写进程（如果有）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：

   ```
   [samfs-mds]root@solaris:~# samcmd stidle
   [samfs-mds]root@solaris:~#
   ```

6. 等待任何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：

   ```
   [samfs-mds]root@solaris:~# samcmd a
   Archiver status samcmd     6.0 14:20:34 Feb 22 2015
   samcmd on samfs-mds
   sam-archiverd:  Waiting for :arrun
   sam-arfind: ...
   Waiting for :arrun
   ```

7. 等待任何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：

   ```
   [samfs-mds]root@solaris:~# samcmd u
   Staging queue samcmd      6.0 14:20:34 Feb 22 2015
   samcmd on solaris.demo.lan
   Staging queue by media type: all
   sam-stagerd:  Waiting for :strun
   root@solaris:~#
   ```

8. 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：

```
[samfs-mds]root@solaris:~# samcmd 801 idle
[samfs-mds]root@solaris:~# samcmd 802 idle
[samfs-mds]root@solaris:~# samcmd 803 idle
[samfs-mds]root@solaris:~# samcmd 804 idle
[samfs-mds]root@solaris:~#
```

9. 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。

```
[samfs-mds]root@solaris:~# samcmd r
Removable media samcmd     6.0 14:20:34 Feb 22 2015
samcmd on samqfs1host
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
[samfs-mds]root@solaris:~#
```

10. 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*。

```
[samfs-mds]root@solaris:~# samd stop
[samfs-mds]root@solaris:~#
```

11. 在文本编辑器中打开 */etc/opt/SUNWsamfs/defaults.conf* 文件。如果需要，取消注释 *#div = off* 行，如果它不存在，则添加该行。

    默认情况下，*div*（数据完整性验证）设置为 *off*（禁用）。

    在示例中，在 *vi* 编辑器中打开文件并取消注释该行：

```
[samfs-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.
```

```
...
div = off
```

12. 要启用数据完整性验证的读取、写入和验证操作，请将行 *#div = off* 更改为 *div = on* 并保存文件。

    在写入和读取各个块时，将对数据进行验证，但在完整的文件副本归档后，Oracle HSM 归档程序软件将不对其进行验证。

```
[samfs-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.
...
div = on
:wq
[samfs-mds]root@solaris:~#
```

13. 要启用数据完整性验证功能的写入后验证选项，请将行 *#div = off* 更改为 *div = verify* 并保存文件。

    在写入或读取各个块时，主机和驱动器会执行数据完整性验证。此外，只要将完整的归档请求写出磁带，驱动器便会重新读取新存储的数据和校验和、重新计算并比较存储结果与计算结果。

```
[samfs-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.
...
div = verify
:wq
[samfs-mds]root@solaris:~#
```

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

```
[samfs-mds]root@solaris:~# /opt/SUNWsamfs/sbin/samd config
```

15. 如果在之前的步骤中已将 Oracle HSM 操作停止，现在可使用 *samd start* 命令将其重新启动。

```
[samfs-mds]root@solaris:~# samd start
[samfs-mds]root@solaris:~#
```

现在已配置数据完整性验证。

16. 如果需要自动执行数据完整性验证，请转到“配置 Oracle HSM 定期介质验证”。
17. 如果需要在文件系统上启用 WORM（Write Once Read Many，一写多读）功能，请参见“启用对一次写入多次读取 (Write Once Read Many, WORM) 文件的支持”。
18. 如果需要与使用 LTFS 的系统交互，或者需要在远程站点之间传输大量数据，请参见“启用对 Linear Tape File System (LTFS) 的支持”。
19. 如果有其他需求（例如，多主机文件系统访问或高可用性配置），请参见“进阶知识”。

配置 Oracle HSM 定期介质验证

可以为 Oracle HSM 归档文件系统设置定期介质验证 (Periodic Media Verification, PMV)。定期介质验证会自动检查文件系统中可移除介质的数据完整性。该功能使用 StorageTek 数据完整性验证来检查 StorageTek T10000 介质，使用广受支持的 SCSI *`verify(6)`* 命令来检查其他驱动器。

定期介质验证功能添加了 Oracle HSM 守护进程 *`verifyd`*，用于定期应用 *`tpverify`* 命令、记录检测到的所有错误、通知管理员以及自动执行指定的恢复操作。通过设置配置文件 *`verifyd.cmd`* 中的策略指令来配置定期介质验证。策略可以指定验证扫描的运行时间、扫描类型、可以使用的磁带库和驱动器、应当扫描的磁带卷，以及在检测到错误时 Oracle HSM 采取的操作。例如，Oracle HSM 可以自动重新归档包含错误的文件和/或回收包含错误的磁带卷。

1. 以 *`root`* 用户身份登录到 Oracle HSM 服务器。

   在示例中，元数据服务器名为 *`samfs-mds`*：

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

2. 如果您尚未执行此操作，则配置 Oracle HSM 以支持数据完整性验证 (Data Integrity Validation, DIV)，然后再继续。
3. 确保元数据服务器运行的是 Oracle Solaris 11 或更高版本。

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

4. 在文本编辑器中打开 */etc/opt/SUNWsamfs/verifyd.cmd* 文件。

   在示例中，用 *`vi`* 编辑器打开文件：

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
# For additional information about the format of the verifyd.cmd file,
```

```
# type "man verifyd.cmd".
# Enable Oracle HSM Periodic Media Validation (PMV)
pmv = off
```

5. 要启用定期介质验证，请输入行 *pmv = on*。

   默认情况下，定期介质验证设置为 *off*。在示例中，我们将其设置为 *on*：

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
# For additional information about the format of the verifyd.cmd file,
# type "man verifyd.cmd".
# Enable Oracle HSM Periodic Media Validation (PMV)
pmv = on
```

6. 设置运行时间。输入行 *run_time = always* 可连续运行验证，或者输入 *run_time = HHMM hhmm DD dd*，其中，*HHMM* 和 *hhmm* 分别是开始时间和结束时间，而 *DD dd* 是可选的开始日期和结束日期。

   HH 和 *hh* 是一天中的小时，介于 *00-24* 范围内；*MM* 和 *mm* 是分钟数，介于 *00-60* 范围内；*DD* 和 *dd* 是一周中的星期几，介于 *[0-6]* 范围内，其中 *0* 表示星期日，*6* 表示星期六。默认值为 *2200 0500 6 0*。

   但是，验证具有更为重要的文件系统操作，因而不具有竞争力。验证过程会自动产生归档程序和回写程序所需的磁带卷和/或驱动器。因此，在示例中，我们将运行时间设置为 *always*：

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
# For additional information about the format of the verifyd.cmd file,
# type "man verifyd.cmd".
# Enable Oracle HSM Periodic Media Validation (PMV)
pmv = on
# Run all of the time.  PMV will yield VSNs and drives when
# resources are wanted by the SAM-QFS archiver and stager.
run_time = always
```

7. 指定验证方法。输入行 *pmv_method = specified-method*，其中 *specified-method* 是下列选项之一：
   - *standard* 方法专用于 Oracle StorageTek T10000C 和更高版本的磁带机。*standard* 方法在速度方面经过优化，可对介质的边缘、开头、结尾和前 1,000 个块进行验证。
   - *complete* 方法也用于 Oracle StorageTek T10000C 和更高版本的磁带机。该方法验证介质上每个块的介质纠错码 (error correction code, ECC)。
   - *complete plus* 方法也用于 Oracle StorageTek T10000C 和更高版本的磁带机。该方法验证介质上每个块的介质纠错码 (error correction code, ECC) 和数

据完整性验证校验和（请参见“配置 Oracle HSM 以支持数据完整性验证 (Data Integrity Validation, DIV)”）。

- *legacy* 方法可用于其他所有磁带机，当介质在目录中标记为损坏以及磁带机不支持 *verifyd.cmd* 文件中指定的方法时，系统会自动使用该方法。该方法运行 6 字节固定块模式 SCSI Verify 命令，从而跳过之前记录的缺陷。找到新的永久性介质错误时，*legacy* 方法会跳到下一个文件，并将新发现的错误记录到介质缺陷数据库中。
- *mir rebuild* 方法可在 Oracle StorageTek 盒式磁带的介质信息区域 (media information region, MIR) 缺失或损坏时重新构建 MIR。它适用于介质目录中标记为已损坏的介质，并且会在检测到 MIR 损坏时自动指定。

在示例中，使用 LTO 驱动器，因此我们指定 *legacy*：

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
# resources are wanted by the SAM-QFS archiver and stager.
run_time = always
pmv_method = legacy
```

8. 要使用所有可用磁带库和驱动器进行验证，请输入行 *pmv_scan = all*。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_method = legacy
pmv_scan = all
```

9. 要使用指定磁带库中所有可用驱动器进行验证，请输入行 *pmv_scan = library equipment-number*，其中 *equipment-number* 是在文件系统的 *mcf* 文件中分配给磁带库的设备编号。

   在示例中，我们让验证过程使用磁带库 *800* 中的所有驱动器。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_method = legacy
pmv_scan = library 800
```

10. 要对验证过程可使用的指定磁带库中的驱动器数量进行限制，请输入行 *pmv_scan = library equipment-number max_drives number*，其中 *equipment-number* 是在文件系统的 *mcf* 文件中分配给磁带库的设备编号，而 number 是可以使用的最大驱动器数量。

    在示例中，我们让验证过程最多使用磁带库 *800* 中的 *2* 个驱动器。

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

```
...
pmv_method = legacy
pmv_scan = library 800 max_drives 2
```

11. 要指定验证过程可使用的指定磁带库中的驱动器，请输入行 *pmv_scan = library equipment-number drive drive-numbers*，其中 *equipment-number* 是在文件系统的 *mcf* 文件中分配给磁带库的设备编号，而 *drive-numbers* 是在 *mcf* 文件中分配给指定驱动器的以空格分隔的设备编号列表。

    在示例中，我们让验证过程使用磁带库 *900* 中的驱动器 *903* 和 *904*：

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_method = legacy
pmv_scan = library 900 drive 903 904
```

12. 要指定验证过程可使用的两个或多个磁带库中的驱动器，请输入行 *pmv_scan = library-specification library-specification...*，其中 *equipment-number* 是在文件系统的 *mcf* 文件中分配给磁带库的设备编号，而 *drive-numbers* 是在 *mcf* 文件中分配给指定驱动器的以空格分隔的设备编号列表。

    在示例中，我们让验证过程最多使用磁带库 *800* 中的 *2* 个驱动器，以及磁带库 *900* 中的驱动器 *903* 和 *904*：

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_method = legacy
pmv_scan = library 800 max_drives 2 library 900 drive 903 904
```

13. 要禁用定期介质验证并防止其使用任何设备，请输入行 *pmv_scan = off*。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_method = legacy
pmv_scan = off
```

14. 要在定期介质验证检测到指定数量的永久性错误时自动标记介质进行回收，请输入行 *action = recycle perms number-errors*，其中 *number-errors* 是错误的数量。

    在示例中，我们将 Oracle HSM 配置为在检测到 *10* 个错误后将介质标记为进行回收：

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

```
pmv_scan = all
action = recycle perms 10
```

15. 要在错误累积了指定时间段后自动对包含错误块的文件进行重新归档，请输入行 *action = rearch age time*，其中 *time* 是以空格分隔的 *SECONDSs*、*MINUTESm*、*HOURSh*、*DAYSd* 和/或 *YEARSy* 的任意组合的列表，并且 *SECONDS*、*MINUTES*、*HOURS*、*DAYS* 和 *YEARS* 为整数。

    在对文件系统进行扫描以查找需要归档的文件之前，最早的介质缺陷必须已经过指定的一段时间。在示例中，我们将重新归档时限设置为 *1*（一）分钟：

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_scan = all
action = rearch age 1m
```

16. 要在定期介质验证检测到永久介质错误时将介质标记为损坏但不执行任何操作，请执行行 *action = none*。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_scan = all
action = none
```

17. 指定应定期验证的磁带卷。输入行 *pmv_vsns = selection-criterion*，其中 *selection-criterion* 为 *all* 或指定一个或多个卷序列号 (volume serial number, VSN) 的以空格分隔的正则表达式列表。

    缺省值为 *all*。在示例中，我们提供三个正则表达式：*^VOL0[01][0-9]* 和 *^VOL23[0-9]* 指定两组卷，这两组卷的卷序列号范围分别介于 *VOL000* 到 *VOL019* 和 *VOL230* 到 *VOL239*，而 *VOL400* 指定具有特定卷序列号的卷：

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_scan = all
action = none
pmv_vsns = ^VOL0[01][0-9] ^VOL23[0-9] VOL400
```

    在以下情况下，Oracle HSM 不会尝试对卷进行验证：卷需要审计、已计划进行回收、不可用、外部（非 Oracle HSM）卷或不包含数据。此外，也不会对清洗磁带、无标签的卷以及卷序列号重复的卷进行验证。

18. 定义所需的验证策略。输入行 *pmv_policy = verified age vertime [modified age modtime] [mounted age mnttime]*，其中：
    - *verified age* 指定自上次验证卷以来必须经过的最短时间。

- *modified age*（可选）指定自上次修改卷以来必须经过的最短时间。
- *mounted age*（可选）指定自上次挂载卷以来必须经过的最短时间。
- 参数值 *vertime*、*modtime* 和 *mnttime* 是非负整数和以下时间单位的组合：*y*（年）、*m*（月）、*d*（天）、*H*（小时）、*M*（分钟）和 *S*（秒）。

Oracle HSM 将根据卷上次验证后所经过的时间（还可以选择根据卷上次修改和/或挂载后所经过的时间）来识别候选卷并对其进行排名。默认策略为单一参数 *verified age 6m*（六个月）。在示例中，我们将上次验证时限设置为三个月，将上次修改时限设置为十五个月：

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_scan = all
action = none
pmv_vsns = ^VOL0[01][0-9] ^VOL23[0-9] VOL400
pmv_policy = verified age 3m modified age 15m
```

19. 保存 */etc/opt/SUNWsamfs/verifyd.cmd* 文件，然后关闭编辑器。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_vsns = ^VOL0[01][0-9] ^VOL23[0-9] VOL400
pmv_policy = verified age 3m modified age 15m
:wq
root@solaris:~#
```

20. 通过输入 *tpverify -x* 命令，检查 *verifyd.cmd* 文件中的错误。更正发现的任何错误。

    tpverify -x 命令读取 *verifyd.cmd* 并在遇到错误时停止：

```
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: legacy
STA Scan: off
Action: none
PMV VSNs: all
PMV Policy:
     Last Verified Age: 6m
root@solaris:~#
```

21. 使用新的 *verifyd.cmd* 文件重新启动验证服务。输入命令 *tpverify -r*。

```
root@solaris:~# tpverify -r
root@solaris:~#
```

    您已完成对定期介质验证的配置。

22. 如果需要在文件系统上启用 WORM（Write Once Read Many，一写多读）功能，请参见“启用对一次写入多次读取 (Write Once Read Many, WORM) 文件的支持”。
23. 如果需要与使用 LTFS 的系统交互，或者需要在远程站点之间传输大量数据，请参见“启用对 Linear Tape File System (LTFS) 的支持”。
24. 如果有其他需求（例如，多主机文件系统访问或高可用性配置），请参见“进阶知识”。
25. 否则，请转至第 11 章 配置通知和日志记录。

启用对一次写入多次读取 (Write Once Read Many, WORM) 文件的支持

出于法律和归档原因，要在许多应用程序中使用一次写入多次读取 (Write-once read-many, WORM) 文件。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。您可以使用以下两种 WORM 模式之一：

- 标准符合性模式（默认）

  标准 WORM 模式在用户设置对目录或不可执行文件 (*chmod 4000 directory|file*) 的 UNIX *setuid* 权限时开始 WORM 保留期。由于对可执行文件设置 *setuid*（在执行时设置用户 *ID*）会带来安全风险，因此无法使用此模式保留同时具有 UNIX 执行权限的文件。
- 仿真模式

  WORM 仿真模式在用户使可写入文件或目录变为只读 (*chmod 444 directory|file*) 时开始 WORM 保持期，因此可以保留可执行文件。

标准模式和仿真模式都有严格 WORM 实施和不太严格的轻量级 (*lite*) 实施，后者会放松对 *root* 用户的一些限制。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。严格实施不允许在保持期结束之前缩短指定的保持期（默认为 43,200 分钟/30 天）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法律和法规遵从性要求。轻量级实施允许 *root* 用户使用文件系统创建命令 *sammkfs* 缩短保持期、删除文件和目录。当数据完整性和灵活管理均为主要要求时，轻量级实施可能是更好的选择。

在选择 WORM 实施和启用文件保留时需要格外谨慎。通常，使用可满足要求的限制最少的选项。您不能从标准模式更改为仿真模式，反之亦然。因此，请谨慎选择。如果管理灵活性是优先的或者保留要求以后可能会发生变化，请选择轻量级实施。您可以从 WORM 模式的轻量级版本升级到严格版本（如果以后有必要）。但您不能从严格实施更改为轻量级实施。一旦严格的 WORM 实施生效后，文件就必须在整个指定保持期内都保留。因此，请将保持期设置为可满足需求的最小值。

在 Oracle HSM 文件系统上启用 WORM 支持

使用挂载选项在文件系统上启用 WORM 支持。请执行如下操作。

1. 以 *root* 用户身份登录。

```
root@solaris:~#
```

2. 备份操作系统的 */etc/vfstab* 文件。

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
```

3. 在文本编辑器中打开 */etc/vfstab* 文件并找到要启用 WORM 支持的 Oracle HSM 文件系统的条目。

   在示例中，我们在 *vi* 编辑器中打开 */etc/vfstab* 文件并找到归档文件系统 *worm1*：

```
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      -
...
worm1     -       /worm1   samfs  -    yes     -
```

4. 要启用标准 WORM 符合性模式的严格实施，请在 *vfstab* 文件的 *Mount Options* 列中输入 *worm_capable* 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      -
...
worm1     -       /worm1   samfs  -    yes     worm_capable
```

5. 要启用标准 WORM 符合性模式的轻量级实施，请在 *vfstab* 文件的 *Mount Options* 列中输入 *worm_lite* 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      -
...
worm1     -       /worm1   samfs  -    yes     worm_lite
```

6. 要启用 WORM 仿真模式的严格实施，请在 *vfstab* 文件的 *Mount Options* 列中输入 *worm_emul* 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      -
...
worm1     -       /worm1   samfs  -    yes     worm_emul
```

7. 要启用 WORM 仿真模式的轻量级实施，请在 *vfstab* 文件的 *Mount Options* 列中输入 *emul_lite* 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      -
...
worm1     -      /worm1    samfs  -    yes     emul_lite
```

8. 要更改未显式分配保持期的文件的默认保持期，请将 *def_retention=period* 选项添加到 *vfstab* 文件的 *Mount Options* 列，其中 *period* 采用下一段介绍的其中一种形式。

   period 的值可采用以下三种形式中的任意一种：

   - *permanent* 或 *0* 指定永久保留。

- *YEARSyDAYSdHOURShMINUTESm*，其中 *YEARS*、*DAYS*、*HOURS* 和 *MINUTES* 为非负整数并且可以忽略说明符。因此，举例来说，*5y3d1h4m*、*2y12h* 和 *365d* 都是有效的。
- *MINUTES*，其中 *MINUTES* 为介于 *[1-2147483647]* 范围内的整数。

如果您必须设置超过 2038 年的保持期，请设置默认保持期。UNIX 实用程序（如 *touch*）使用带符号的 32 位整数将时间表示为自 1970 年 1 月 1 日以来的秒数。32 位整数可表示的最大秒数可转换为 2038 年 1 月 18 日晚上 10 点 14 分

如果未提供值，则 *def_retention* 默认为 *43200* 分钟（30 天）。在示例中，我们将具有标准 WORM 功能的文件系统的保持期设置为 *777600* 分钟（540 天）：

```
#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      -
...
worm1      -       /worm1   samfs  -    no     worm_capable,def_retention=777600
```

9. 保存 *vfstab* 文件并关闭编辑器。

   文件系统已启用 WORM 功能。一旦一个或多个 WORM 文件驻留在文件系统中，Oracle HSM 软件就会更新文件系统超级块以反映 WORM 功能。如果文件系统已使用严格的 *worm_capable* 或 *worm_emul* 挂载选项挂载，则随后通过 *sammkfs* 重新构建文件系统的任何尝试都将失败。
10. 如果需要与使用 LTFS 的系统交互，或者需要在远程站点之间传输大量数据，请参见“启用对 Linear Tape File System (LTFS) 的支持”。
11. 如果有其他需求（例如，多主机文件系统访问或高可用性配置），请参见“进阶知识”。
12. 否则，请转至第 11 章 配置通知和日志记录。

启用对 Linear Tape File System (LTFS) 的支持

Oracle HSM 可从 Linear Tape File System (LTFS) 卷导入数据，也可以将数据导出到 LTFS 卷。此功能有助于与使用 LTFS 作为标准磁带格式的系统进行互操作。此外，当典型的广域网 (Wide Area Network, WAN) 连接太慢或对于任务来说成本过高时，它还能方便地在远程 Oracle HSM 站点之间传输大量数据。

请注意，Oracle HSM 软件支持但不包括 LTFS 功能。要使用 LTFS 文件系统，主机的 Solaris 操作系统必须包括 *SUNWltfs* 软件包。如果需要，请先下载和安装 *SUNWltfs* 软件包，然后再继续。

有关使用和管理 LTFS 卷的信息，请参见 *samltfs* 手册页和《Oracle Hierarchical Storage Manager and StorageTek QFS Software 维护和管理指南》。

要启用 Oracle HSM LTFS 支持，请执行如下操作：

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

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

2. 使所有归档进程（如果有）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：

   ```
   [samfs-mds]root@solaris:~# samcmd aridle
   [samfs-mds]root@solaris:~#
   ```

3. 使所有回写进程（如果有）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：

   ```
   [samfs-mds]root@solaris:~# samcmd stidle
   [samfs-mds]root@solaris:~#
   ```

4. 等待任何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：

   ```
   [samfs-mds]root@solaris:~# samcmd a
   Archiver status samcmd     6.0 14:20:34 Feb 22 2015
   samcmd on samfs-mds
   sam-archiverd:  Waiting for :arrun
   sam-arfind: ...
   Waiting for :arrun
   ```

5. 等待任何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：

   ```
   [samfs-mds]root@solaris:~# samcmd u
   Staging queue samcmd      6.0 14:20:34 Feb 22 2015
   samcmd on solaris.demo.lan
   Staging queue by media type: all
   sam-stagerd:  Waiting for :strun
   root@solaris:~#
   ```

6. 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：

   ```
   [samfs-mds]root@solaris:~# samcmd 801 idle
   [samfs-mds]root@solaris:~# samcmd 802 idle
   [samfs-mds]root@solaris:~# samcmd 803 idle
   [samfs-mds]root@solaris:~# samcmd 804 idle
   [samfs-mds]root@solaris:~#
   ```

7. 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。

   ```
   [samfs-mds]root@solaris:~# samcmd r
   Removable media samcmd     6.0 14:20:34 Feb 22 2015
   samcmd on samqfs1host
   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
   [samfs-mds]root@solaris:~#
   ```

8. 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*。

   ```
   [samfs-mds]root@solaris:~# samd stop
   [samfs-mds]root@solaris:~#
   ```

9. 在文本编辑器中打开 */etc/opt/SUNWsamfs/defaults.conf*。

   在示例中，用 *vi* 编辑器打开文件：

```
[samfs-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.
...
```

10. 在 *defaults.conf* 文件中，添加行 *ltfs = mountpoint workers volumes*，其中挂载点为应挂载 LTFS 文件系统的主机文件系统中的目录，*workers* 为要用于 LTFS 的最大驱动器数（可选），*volumes* 为每个驱动器的最大磁带卷数（可选）。然后保存文件并关闭编辑器。

    在示例中，我们指定 LTFS 挂载点 */mnt/ltfs* 并接受其他参数的默认值：

```
[samfs-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.
...
ltfs = /mnt/ltfs
:wq
[samfs-mds]root@solaris:~#
```

11. 指示 Oracle HSM 软件重新读取 *defaults.conf* 文件并相应地重新配置自身。更正报告的任何错误并根据需要重复。

```
[samfs-mds]root@solaris:~# /opt/SUNWsamfs/sbin/samd config
```

12. 如果在之前的步骤中已将 Oracle HSM 操作停止，现在可使用 *samd start* 命令将其重新启动。

```
[samfs-mds]root@solaris:~# samd start
```

13. 现在已启用 Oracle HSM 对 LTFS 的支持。如果有其他需求（例如，多主机文件系统访问或高可用性配置），请参见“进阶知识”。
14. 否则，请转至第 11 章 配置通知和日志记录。

进阶知识

至此，Oracle HSM 文件系统的基本安装和配置已经完成。目前，您已设置了功能全面的文件系统，并已针对各种用途进行了最优配置。

本书的其他章节将满足更专业的需要。因此，在您开始执行下面所述的其他调整和功能实施任务之前，请仔细评估您的需求。之后，如果您需要其他功能，例如，高可用性或共享文件系统配置，则可以从基本配置开始审慎地实施其他功能。但是，如果您发现目前所做的工作能够满足您的需要，则执行额外的更改可能不会有任何改善，而只会使维护和管理工作更为复杂。

- 如果应用程序要将海量数据或非常不统一的数据传输到文件系统，则可能需要设置其他挂载选项来提高文件系统性能。有关详细信息，请参见第 12 章 针对特殊需求调整 *I/O* 特征。
- 如果需要配置对文件系统的共享访问，请参见“使用 Oracle HSM 软件从多台主机访问文件系统”和/或“使用 NFS 和 SMB/CIFS 从多台主机访问文件系统”。
- 如果需要配置高可用性 QFS 文件系统或 Oracle HSM 归档文件系统，请参见第 9 章 准备高可用性解决方案 。
- 如果需要配置 Oracle HSM 归档文件系统以共享远程位置所托管的归档存储，请参见第 8 章 配置 *SAM-Remote* 。
- 如果计划使用边带数据库功能，请转至第 10 章 配置报告数据库。
- 否则，请转至第 11 章 配置通知和日志记录。

7

第 7 章 从多台主机访问文件系统

Oracle HSM 文件系统可以使用多种方法在多台主机之间共享。每种方法都在某些情况下有特定的优势，而在其他情况下有明显的缺点。因此您选择的方法取决于您的特定要求。共享方法包括：

- 使用 Oracle HSM 软件从多台主机访问文件系统
- 使用 NFS 和 SMB/CIFS 从多台主机访问文件系统

使用 Oracle HSM 软件从多台主机访问文件系统

Oracle HSM 通过配置同时挂载文件系统的一台服务器以及一台或多台客户机，将文件系统供多台主机使用。然后，文件数据将通过高性能的本地路径 I/O 直接从磁盘设备传递到主机，而不会发生与 NFS 和 CIFS 共享相关的网络和中间服务器延迟。一次只能有一台主机作为活动元数据服务器，但可将任意数量的客户机配置为潜在元数据服务器，以用于冗余用途。对文件系统挂载点的数量没有限制。

Oracle HSM 在多读取器/单写入器配置和共享配置（归档或不归档）中，都支持对高性能 (*ma*) 和通用 (*ms*) 文件系统进行多主机访问。只有一些限制：

- 块 (*b–*) 特殊文件不受支持。
- 字符 (*c–*) 特殊文件不受支持。
- FIFO 命名管道 (*p–*) 特殊文件不受支持。
- 分段文件不受支持。

  您不能在分段文件环境中实现 Oracle HSM 共享文件系统。
- 强制性锁定不受支持。

  如果设置了强制锁定，则系统会返回 *EACCES* 错误。但系统支持咨询锁定。有关建议性锁定的更多信息，请参见 `fcntl` 手册页。

Oracle HSM 软件主机可以使用两种配置中的任一种访问文件系统数据，在任何给定的应用中，每种配置都有其自己的优势和限制。

在多读取器单写入器配置中，一台主机会以读取/写入访问权限挂载文件系统，所有其他主机会以只读访问权限挂载文件系统。配置就是设置挂载点的简单操作。由于单台主机对文件进行所有更改，因此可确保文件一致性和数据完整性，而无需执行其他文件锁定或一致性检查。为实现最佳性能，所有主机会直接从磁盘读取元数据以及数

据。但所有主机都必须有权访问文件系统元数据，因此，*ma* 文件系统中的所有主机都必须有权访问数据和元数据设备。

在共享配置中，所有主机可使用租约读取、写入和附加文件数据，从而允许单台主机在给定的时间段内以给定方式访问文件。元数据服务器会发放读取、写入和附加租约，并管理续订和冲突的租约请求。共享文件系统可提供更高灵活性，但配置要复杂一些且需要更多文件系统开销。所有主机会直接从磁盘读取文件数据，但客户机通过网络访问元数据。因此，没有元数据设备访问权限的客户机可以共享 *ma* 文件系统。

要配置从多个主机对数据的访问，请选择以下两种方法之一：

- 配置 Oracle HSM 单写入器多读取器文件系统
- 配置 Oracle HSM 共享文件系统。

配置 Oracle HSM 单写入器多读取器文件系统

要配置单写入器多读取器文件系统，请执行以下任务：

- 在写入器上创建文件系统
- 配置读取器

在写入器上创建文件系统

执行如下操作：

1. 以 *root* 帐户登录到将用作 *writer* 的主机。

   在本示例中，*writer* 主机名为 *swriterfs-mds-writer*：

```
[swriterfs1-mds-writer]root@solaris:~#
```

2. 在将用作 *writer* 的主机上，在文本编辑器中打开 */etc/opt/SUNWsamfs/mcf* 文件并添加一个 QFS 文件系统。可以配置通用 *ms* 或高性能 *ma* 文件系统。

   在拥有独立元数据设备的 *ma* 文件系统上，将文件系统的元数据服务器配置为写入器。在下面的示例中，使用 *vi* 文本编辑器来编辑主机 *swriterfs1-mds-writer* 上的 *mcf* 文件。示例使用设备标识符和系列集名称 *swriterfs1* 以及设备序号 *300* 指定 *ma* 文件系统：

```
[swriterfs1-mds-writer]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment          Equipment  Equipment  Family      Device   Additional
# Identifier         Ordinal    Type       Set         State    Parameters
#------------------  ---------  ---------  ---------   ------   ---------------
swriterfs1           300        ma         swriterfs1  on
 /dev/dsk/c0t0d0s0   301        mm         swriterfs1  on
 /dev/dsk/c0t3d0s0   302        mr         swriterfs1  on
```

```
 /dev/dsk/c0t3d0s1    303        mr         swriterfs1  on
```

3. 保存 */etc/opt/SUNWsamfs/mcf* 文件并退出编辑器。

   在示例中，保存更改并退出 *vi* 编辑器：

```
# Equipment           Equipment  Equipment  Family      Device   Additional
# Identifier          Ordinal    Type       Set         State    Parameters
#------------------  ---------  ---------  ---------   ------   ---------------
swriterfs1            300        ma         swriterfs1  on
 /dev/dsk/c0t0d0s0     301        mm          swriterfs1  on
 /dev/dsk/c0t3d0s0     302        mr          swriterfs1  on
 /dev/dsk/c0t3d0s1     303        mr          swriterfs1  on
:wq
[swriterfs1-mds-writer]root@solaris:~#
```

4. 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 命令会读取 Oracle HSM 配置文件并初始化文件系统。该命令会在遇到以下错误时停止：

```
[swriterfs1-mds-writer]root@solaris:~# sam-fsd
...
Would start sam-stagerd()
Would start sam-amld()
[swriterfs1-mds-writer]root@solaris:~#
```

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

```
[swriterfs1-mds-writer]root@solaris:~# samd config
Configuring SAM-FS
[swriterfs1-mds-writer]root@solaris:~#
```

6. 使用 *sammkfs* 命令和文件系统的系列集名称创建文件系统，如“配置高性能 **ma** 文件系统” 中所述。

   在示例中，该命令创建了单写入器/多读取器文件系统 *swriterfs1*：

```
[swriterfs1-mds-writer]root@solaris:~# sammkfs swriterfs1
Building 'swriterfs1' will destroy the contents of devices:
  /dev/dsk/c0t0d0s0
  /dev/dsk/c0t3d0s0
  /dev/dsk/c0t3d0s1
Do you wish to continue? [y/N]yes ...
```

7. 备份操作系统的 */etc/vfstab* 文件。

```
[swriterfs1-mds-writer]root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
[swriterfs1-mds-writer]root@solaris:~#
```

8. 将新的文件系统添加到操作系统的 */etc/vfstab* 文件，如“配置高性能 **ma** 文件系统”中所述。

   在示例中，在 *vi* 文本编辑器中打开 */etc/vfstab* 文件，并为 *swriterfs1* 系列集设备添加一行：

```
[swriterfs1-mds-writer]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       -
...
swriterfs1 -        /swriterfs1 samfs   -     no
```

9. 在 */etc/vfstab* 文件的 *Mount Options* 列中，输入*writer* 挂载选项。

   注意:

   确保在任何给定时间只有一个主机为 *writer*。允许多台主机使用 *writer* 选项挂载一个多读取器单写入器文件系统可能会损坏文件系统！

```
#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       -
...
swriterfs1 -        /swriterfs1 samfs   -     no       writer
```

10. 对 */etc/vfstab* 文件进行所需的其他更改。使用逗号作为分隔符来添加挂载选项。

    例如，要在首次尝试不成功时在后台挂载文件系统，请将 *bg* 挂载选项添加到 *Mount Options* 字段中（有关可用挂载选项的综合列表，请参见 *mount_samfs* 手册页）：

```
#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       -
...
swriterfs1 -        /swriterfs1 samfs   -     no       writer,bg
```

11. 保存 */etc/vfstab* 文件并退出编辑器。

```
#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       -
...
swriterfs1 -        /swriterfs1 samfs   -     no       writer,bg
:wq
[swriterfs1-mds-writer]root@solaris:~#
```

12. 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。

    所有主机上的挂载点权限必须相同，且用户必须拥有执行 (*x*) 权限以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 */swriterfs1* 挂载点目录并将权限设置为 *755* (*-rwxr-xr-x*)：

```
[swriterfs1-mds-writer]root@solaris:~# mkdir /swriterfs1
[swriterfs1-mds-writer]root@solaris:~# chmod 755 /swriterfs1
[swriterfs1-mds-writer]root@solaris:~#
```

13. 挂载新文件系统：

```
[swriterfs1-mds-writer]root@solaris:~# mount /swriterfs1
[swriterfs1-mds-writer]root@solaris:~#
```

14. 创建了共享文件系统后，配置读取器。

配置读取器

reader 是以只读方式挂载文件系统的主机。对于您要配置为 reader 的每个主机，请执行如下操作：

1. 以 *root* 用户身份登录主机。

   在本示例中，*reader* 主机名为 *swriterfs-reader1]*：

```
[swriterfs-reader1]root@solaris:~#
```

2. 在终端窗口中，使用 *samfsconfig device-path* 命令检索多读取器、单写入器文件系统的配置信息，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/**）。

   samfsconfig 实用程序会读取 *sammkfs* 在 Oracle HSM 文件系统中包括的每台设备上写入的标识超级块，以检索文件系统配置信息。该命令会返回配置中从当前主机开始的每台设备的正确路径，并标记无法连接的设备（有关命令语法和参数的完整信息，请参见 *samfsconfig* 手册页）。

   在示例中，*samfsconfig* 输出显示 *swriterfs1-mds-writer* 上的 *mcf* 文件中所列的相同设备，只是设备的路径是从主机 *swriterfs1-reader1* 开始指定的：

```
[swriterfs1-reader1]root@solaris:~# samfsconfig /dev/dsk/*
# Family Set 'swriterfs1' Created Thu Nov 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   -
```

3. 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文件 */etc/opt/SUNWsamfs/mcf*，并将复制的条目粘贴到文件中。

   或者，您可以将 *samfsconfig* 的输出重定向至 *mcf* 文件。或者，您可以使用 *samd buildmcf* 命令运行 *samfsconfig* 并自动创建客户机 *mcf* 文件。

   在示例中，在我们添加注释掉的列标题之后，主机的 *mcf* 文件 *swriterfs1-reader1* 就会如下所示：

```
[swriterfs1-reader1]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    -
```

4. 确保所有设备的 *Device State* 字段均设置为 *on*。然后保存 *mcf* 文件。

```
[swriterfs1-reader1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family     Device   Additional
```

```
# Identifier          Ordinal    Type       Set        State    Parameters
#------------------   ---------  ---------  ---------  ------   ---------------
sharefs               300        ma         sharefs    on
 /dev/dsk/c1t0d0s0     301        mm         sharefs    on
 /dev/dsk/c1t3d0s0     302        mr         sharefs    on
 /dev/dsk/c1t3d0s1     303        mr         sharefs    on
:wq
[swriterfs1-reader1]root@solaris:~#
```

5. 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 命令会读取 Oracle HSM 配置文件并初始化文件系统。该命令会在遇到以下错误时停止：

```
[swriterfs1-reader1]root@solaris:~# sam-fsd
...
Would start sam-stagerd()
Would start sam-amld()
[swriterfs1-reader1]root@solaris:~#
```

6. 备份操作系统的 */etc/vfstab* 文件。

```
[swriterfs1-reader1]root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
[swriterfs1-reader1]root@solaris:~#
```

7. 将单写入器多读取器文件系统添加到主机操作系统的 */etc/vfstab* 文件。

   在示例中，在 *vi* 文本编辑器中打开 */etc/vfstab* 文件，并为 *swriterfs1* 系列集设备添加一行：

```
[swriterfs1-reader1]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       -
...
swriterfs1 -        /swriterfs1 samfs   -     no
```

8. 在 */etc/vfstab* 文件的 *Mount Options* 列中，输入 *reader* 选项。

   注意:

   请确保主机使用 *reader* 选项挂载文件系统！无意中在多台主机上使用 *writer* 挂载选项可能会损坏文件系统！

```
#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       -
...
swriterfs1 -        /swriterfs1 samfs   -     no       reader
```

9. 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。

```
#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       -
...
swriterfs1 -        /swriterfs1 samfs   -     no       writer,bg
:wq
[swriterfs1-reader1]root@solaris:~#
```

10. 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。

    所有主机上的挂载点权限必须相同，且用户必须拥有执行 (*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，就像在写入器主机上一样，创建 */swriterfs1* 挂载点目录，并将权限设置为 *755* (*-rwxr-xr-x*)：

```
[swriterfs1-reader1]root@solaris:~# mkdir /swriterfs1
[swriterfs1-reader1]root@solaris:~# chmod 755 /swriterfs1
[swriterfs1-reader1]root@solaris:~#
```

11. 挂载新文件系统：

```
[swriterfs1-reader1]root@solaris:~# mount /swriterfs1
[swriterfs1-reader1]root@solaris:~#
```

12. 重复该过程，直到所有读取器主机已配置为以只读方式挂载文件系统。
13. 如果计划使用边带数据库功能，请转至第 10 章 配置报告数据库。
14. 否则，请转至第 11 章 配置通知和日志记录。

配置 Oracle HSM 共享文件系统

Oracle HSM 共享文件系统可向多台 Oracle HSM 主机授予文件的读取、写入和附加访问权限。所有主机都会挂载该文件系统且可直接连接存储设备。此外，元数据服务器 (metadata server, MDS) 这台主机可独占控制文件系统元数据，并在尝试访问相同文件的主机之间起中介作用。该服务器通过以太网本地网络为客户机主机提供元数据更新，并通过发放、续订和撤销读取、写入和附加租约来控制文件访问。高性能 *ma* 或通用 *ms* 类型的非归档和归档文件系统都可共享。

要配置共享文件系统，请执行以下任务：

- 配置文件系统元数据服务器以进行共享
- 配置文件系统客户机以进行共享
- 为共享文件系统配置归档存储

配置文件系统元数据服务器以进行共享

要配置元数据服务器以支持共享文件系统，请执行下列任务：

- 在活动元数据服务器和潜在元数据服务器上创建 Hosts 文件
- 在活动服务器上创建共享文件系统
- 在活动服务器上挂载共享文件系统

在活动元数据服务器和潜在元数据服务器上创建 Hosts 文件

在活动元数据服务器和潜在元数据服务器上，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. 以 *root* 用户身份登录服务器。

   在示例中，服务器名为 *sharefs-mds*：

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

2. 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.sharefs*。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：

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

```
#                                          Server   On/  Additional
#Host Name            Network Interface     Ordinal  Off  Parameters
#------------------   --------------------  -------  ---  ----------
```

3. 在两列中添加元数据服务器的主机名和 IP 地址或域名，以空格字符分隔。

```
# /etc/opt/SUNWsamfs/hosts.sharefs
#                                          Server   On/  Additional
#Host Name            Network Interface     Ordinal  Off  Parameters
#------------------   --------------------  -------  ---  ----------
sharefs-mds          10.79.213.117
```

4. 添加第三列，用空格字符将其与网络地址隔开。在该列中，输入 *1*（活动元数据服务器的序号）。

   在此示例中，只有一台元数据服务器，因此输入 *1*：

```
# /etc/opt/SUNWsamfs/hosts.sharefs
#                                          Server   On/  Additional
#Host Name            Network Interface     Ordinal  Off  Parameters
#------------------   --------------------  -------  ---  ----------
sharefs-mds          10.79.213.117         1
```

5. 添加第四列，用空格字符将其与网络地址隔开。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 *samsharefs* 手册页）。

```
# /etc/opt/SUNWsamfs/hosts.sharefs
#                                          Server   On/  Additional
#Host Name            Network Interface     Ordinal  Off  Parameters
#------------------   --------------------  -------  ---  ----------
sharefs-mds          10.79.213.117         1        0
```

6. 添加第五列，用空格字符将其与网络地址隔开。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：

```
# /etc/opt/SUNWsamfs/hosts.sharefs
#                                          Server   On/  Additional
#Host Name            Network Interface     Ordinal  Off  Parameters
#------------------   --------------------  -------  ---  ----------
sharefs-mds          10.79.213.117         1        0    server
```

7. 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 *sharefs-mds_alt* 是服务器序号为 *2* 的一台潜在元数据服务器：

```
# /etc/opt/SUNWsamfs/hosts.sharefs
#                                          Server   On/  Additional
#Host Name            Network Interface    Ordinal  Off  Parameters
#------------------   ---------------------  -------  ---  ----------
sharefs-mds           10.79.213.117          1        0    server
sharefs-mds_alt       10.79.213.217          2        0
```

8. 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 *sharefs-client1* 和 *sharefs-client2*。

```
# /etc/opt/SUNWsamfs/hosts.sharefs
#                                          Server   On/  Additional
#Host Name            Network Interface    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       10.79.213.147          0        0
```

9. 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.sharefs* 的更改并退出 *vi* 编辑器：

```
# /etc/opt/SUNWsamfs/hosts.sharefs
#                                          Server   On/  Additional
#Host Name            Network Interface    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       10.79.213.147          0        0
:wq
[sharefs-mds]root@solaris:~#
```

10. 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。

11. 现在，在活动元数据服务器上创建共享文件系统。

在活动服务器上创建共享文件系统

执行如下操作：

1. 以 *root* 用户身份登录服务器。

   在示例中，服务器名为 *sharefs-mds*：

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

2. 在元数据服务器 (metadata server, MDS) 上，在文本编辑器中打开 */etc/opt/SUNWsamfs/mcf* 文件并添加一个 QFS 文件系统。可以配置通用 *ms* 或高性能 *ma* 文件系统。

   在下面的示例中，使用 *vi* 文本编辑器编辑主机 *sharefs-mds* 上的 *mcf* 文件。示例使用设备标识符和系列集名称 *sharefs* 以及设备序号 *300* 指定 *ma* 文件系统：

```
[sharefs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#------------------   ---------  ---------  ---------  ------   ---------------
sharefs               300        ma         sharefs    on
 /dev/dsk/c0t0d0s0    301        mm         sharefs    on
 /dev/dsk/c0t3d0s0    302        mr         sharefs    on
 /dev/dsk/c0t3d0s1    303        mr         sharefs    on
```

3. 在 *ma* 文件系统设备所在行的 *Additional Parameters* 字段中，输入 *shared* 参数：

```
# Equipment           Equipment   Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#------------------   ---------  ---------  ---------  ------   ---------------
sharefs               300         ma        sharefs    on       shared
 /dev/dsk/c0t0d0s0     301         mm        sharefs    on
 /dev/dsk/c0t3d0s0     302         mr        sharefs    on
 /dev/dsk/c0t3d0s1     303         mr        sharefs    on
```

4. 保存 */etc/opt/SUNWsamfs/mcf* 文件并退出编辑器。

   在示例中，保存更改并退出 *vi* 编辑器：

```
sharefs               300         ma         sharefs    on      shared
 /dev/dsk/c0t0d0s0     301         mm         sharefs    on
 /dev/dsk/c0t3d0s0     302         mr         sharefs    on
```

```
 /dev/dsk/c0t3d0s1     303         mr         sharefs    on
:wq
[sharefs-mds]root@solaris:~#
```

5. 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 命令会读取 Oracle HSM 配置文件并初始化文件系统。该命令会在遇到以下错误时停止：

```
[sharefs-mds]root@solaris:~# sam-fsd
...
Would start sam-stagerd()
Would start sam-amld()
[sharefs-mds]root@solaris:~#
```

6. 指示 Oracle HSM 服务重新读取 *mcf* 文件并相应地重新配置自身。更正报告的任何错误并根据需要重复。

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

7. 使用 *sammkfs -S* 命令和文件系统的系列集名称创建文件系统，如“配置高性能 ma 文件系统”中所述。

   sammkfs 命令读取 *hosts.family-set-name* 和 *mcf* 文件，并创建具有指定属性的共享文件系统。在示例中，该命令会从 *hosts.sharefs* 文件读取共享参数，并创建共享文件系统 *sharefs*：

```
[sharefs-mds]root@solaris:~# sammkfs -S sharefs
Building 'sharefs' will destroy the contents of devices:
  /dev/dsk/c0t0d0s0
  /dev/dsk/c0t3d0s0
  /dev/dsk/c0t3d0s1
Do you wish to continue? [y/N]yes ...
[sharefs-mds]root@solaris:~#
```

8. 接下来，在活动元数据服务器上挂载共享文件系统。

在活动服务器上挂载共享文件系统

1. 以 *root* 用户身份登录服务器。

   在示例中，服务器名为 *sharefs-mds*：

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

2. 备份操作系统的 */etc/vfstab* 文件。

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

3. 将新的文件系统添加到操作系统的 */etc/vfstab* 文件，如“配置高性能 **ma** 文件系统”中所述。

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

```
[sharefs-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       -
...
sharefs    -        /sharefs  samfs   -     no
```

4. 在 *Mount Options* 列中，输入 *shared* 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
```

5. 对 */etc/vfstab* 文件进行所需的其他更改。

   例如，要在首次尝试不成功时在后台重试挂载文件系统，请将 *bg* 挂载选项添加到 *Mount Options* 字段中（有关可用挂载选项的完整描述，请参见 *mount_samfs* 手册页）：

```
#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,bg
```

6. 保存 */etc/vfstab* 文件并退出编辑器。

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

7. 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。

   元数据服务器以及所有客户机上的挂载点权限必须相同，且用户必须拥有执行 (*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-mds]root@solaris:~# mkdir /sharefs
[sharefs-mds]root@solaris:~# chmod 755 /sharefs
[sharefs-mds]root@solaris:~#
```

8. 挂载新文件系统：

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

9. 如果您的主机配置了多个网络接口，可能需要使用本地 hosts 文件来路由网络通信。
10. 否则，在元数据服务器上创建了共享文件系统后，配置文件系统客户机以进行共享。

配置文件系统客户机以进行共享

客户机包括仅配置为客户机的主机以及配置为潜在元数据服务器的主机。在大多数方面，配置客户机与配置服务器基本相同。每台客户机与服务器包括的设备完全相同，只是挂载选项和设备的具体路径会更改（控制器编号由每台客户机主机分配，因此可能有所不同）。

要配置一台或多台客户机以支持共享文件系统，请执行下列任务：

- 在 Solaris 客户机上创建共享文件系统

- 在 Solaris 客户机上挂载共享文件系统
- 在 Linux 客户机上创建共享文件系统（如果有）
- 在 Linux 客户机上挂载共享文件系统（如果有）。

在 Solaris 客户机上创建共享文件系统

针对每个客户机，执行如下操作：

1. 在客户机上，以 *root* 用户身份登录。

   在示例中，服务器名为 *sharefs-client1*：

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

2. 在终端窗口中，输入命令 *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/**）。

   samfsconfig 命令检索共享文件系统的配置信息。

```
[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。

    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 命令会读取 Oracle HSM 配置文件并初始化文件系统。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
...
Would start sam-stagerd()
Would start sam-amld()
[sharefs-client1]root@solaris:~#
```

12. 此时，如果您的主机配置了多个网络接口，可能需要使用本地 hosts 文件来路由网络通信。
13. 接下来，在 Solaris 客户机上挂载共享文件系统。

在 Solaris 客户机上挂载共享文件系统

针对每个客户机，执行如下操作：

1. 在 Solaris 客户机上，以 *root* 用户身份登录。

在示例中，服务器名为 *sharefs-client1*：

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

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

5. 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:~#
```

6. 挂载共享文件系统：

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

7. 如果共享文件系统包括 Linux 客户机，请在 Linux 客户机上创建共享文件系统。
8. 如果您要配置 Oracle HSM 共享的归档文件系统，请转至下一个任务：“为共享文件系统配置归档存储”。
9. 否则，请在此处停止。您已配置 Oracle HSM 共享文件系统。

在 Linux 客户机上创建共享文件系统

针对每个客户机，执行如下操作：

1. 在 Linux 客户机上，以 *root* 用户身份登录。

   在本示例中，Linux 客户机主机名为 *sharefs-clientL*：

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

2. 在终端窗口中，输入命令 *samfsconfig device-path*，其中 *device-path* 是该命令应开始搜索文件系统磁盘设备的位置（如 */dev/**）。

   samfsconfig 命令检索共享文件系统的配置信息。因为 Linux 主机对文件系统的元数据设备没有访问权限，*samfsconfig* 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 ~]#
```

8. 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 命令会读取 Oracle HSM 配置文件并初始化文件系统。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
...
Would start sam-stagerd()
Would start sam-amld()
[sharefs-clientL][root@linux ~]#
```

9. 现在，在 Linux 客户机上挂载共享文件系统。

在 Linux 客户机上挂载共享文件系统

针对每个客户机，执行如下操作：

1. 在 Linux 客户机上，以 *root* 用户身份登录。

   在本示例中，Linux 客户机主机名为 *sharefs-clientL*：

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

2. 备份操作系统的 */etc/fstab* 文件。

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

3. 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，在备份 *sharefs-clientL* 上的 */etc/fstab* 文件后，在 *vi* 文本编辑器中打开该文件并为 *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
```

4. 在文件的第四列中，添加必需的 *shared* 挂载选项。

```
#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
```

8. 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。

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

9. 如果您要配置 Oracle HSM 共享的归档文件系统，请转至下一个任务：“为共享文件系统配置归档存储”
10. 如果计划使用边带数据库功能，请转至第 10 章 配置报告数据库。
11. 否则，请转至第 11 章 配置通知和日志记录。

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。文件系统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。

每个文件系统主机在元数据服务器上查找其他主机的网络接口。文件系统的全局 hosts 文件 */etc/opt/SUNWsamfs/hosts.family-set-name* 中列出了主机名和 IP 地址，其中 *family-set-name* 是共享文件系统的系列集编号。然后，主机查找本地 hosts 文件 */etc/opt/SUNWsamfs/hosts.family-set-name.local*。

如果没有本地 hosts 文件，则主机使用在全局 hosts 文件中指定的接口地址。按照全局文件指定的顺序使用主机。

如果有本地 hosts 文件，则主机将其与全局文件进行比较并仅使用在两个文件中都列出的接口。按照本地文件中指定的顺序使用主机。

因此，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. 在每个活动元数据服务器主机和潜在元数据服务器主机上，编辑共享文件系统的全局 hosts 文件，以使其按照所需的方法路由服务器和主机通信。

   对于本节中的示例，共享文件系统 *sharefs2nic* 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。

   因此我们编辑 */etc/opt/SUNWsamfs/hosts.sharefs2*（文件系统的全局 hosts 文件）。我们指定活动服务器和潜在服务器的专用网络接口地址。但是，对于客户机，我们提供主机名（而不是地址）：

```
[sharefs2-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sharefs2
# /etc/opt/SUNWsamfs/hosts.sharefs2
#                                    Server   On/  Additional
#Host Name        Network Interface  Ordinal  Off  Parameters
#--------------  -----------------  -------  ---  ----------
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
:wq
[sharefs2-mds]root@solaris:~#
```

2. 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，使用路径和文件名 */etc/opt/SUNWsamfs/hosts.family-set-name.local*，其中 *family-set-name* 是共享文件系统的设备标识符。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。

   在我们的示例中，我们希望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，每个服务器上的本地 hosts 文件 *hosts.sharefs2.local* 仅列出两个主机（活动元数据服务器和潜在元数据服务器）的专用地址：

```
[sharefs2-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sharefs2.local
# /etc/opt/SUNWsamfs/hosts.sharefs2 on sharefs2-mds
#                                    Server   On/  Additional
#Host Name        Network Interface  Ordinal  Off  Parameters
```

```
#--------------  -----------------  -------  ---  ----------
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.local on sharefs2-mds_alt
#                                   Server   On/  Additional
#Host Name        Network Interface  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:~#
```

3. 在每台客户机上创建一个本地 hosts 文件，使用路径和文件名 */etc/opt/SUNWsamfs/hosts.family-set-name.local*，其中 *family-set-name* 是共享文件系统的设备标识符。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，文件仅包含两个主机（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.local on sharefs2-client1
#                                   Server   On/  Additional
#Host Name        Network Interface  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.local on sharefs2-client2
#                                   Server   On/  Additional
#Host Name        Network Interface  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，请转至“在活动服务器上挂载共享文件系统”。
5. 如果您在配置客户机时启动该过程，现在应“在 Solaris 客户机上挂载共享文件系统”。

为共享文件系统配置归档存储

要为 Oracle HSM 共享文件系统设置归档存储，请执行以下任务：

- 使用永久绑定将磁带机连接到服务器和数据移动器主机
- 配置归档文件系统的主机以使用归档存储
- 将磁带 I/O 分布到共享归档文件系统的主机上（如果需要）。

使用永久绑定将磁带机连接到服务器和数据移动器主机

在共享归档文件系统中，所有潜在元数据服务器必须有权访问库和磁带机。如果您决定将磁带 I/O 分布到共享归档文件系统的主机上，则一个或多个客户机还将需要访问驱动器。因此，您必须配置其中的每台主机，以使用一致的方式对每个驱动器进行寻址。

Solaris 操作系统按照启动时发现设备的顺序连接系统设备树中的驱动器。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可移动介质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永久绑定到每台主机，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（有关创建永久绑定的完整信息，请参见 Solaris *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）：

1. 以 *root* 用户身份登录到活动元数据服务器。

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

2. 如果您不知道库中驱动器的当前物理顺序，请按“确定驱动器在库中的安装顺序”所述创建映射文件。

   在示例中，*device-mappings.txt* 文件类似于以下内容：

```
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，向 */etc/devlink.tab* 文件中添加一行，以将 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设备。以 *type=ddi_byte:tape; addr=device_address,0; rmt/node-number/M0* 形式输入该行，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，该位置足够高，因而可以避免与 Solaris 自动配置的任何设备发生冲突（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* 中创建一行来将非冲突节点 *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，我们将设备地址映射到接下来的三个 Solaris 设备节点（**rmt/61**、**rmt/62** 和 **rmt/63**），保持与在库中相同的顺序：

```
[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个潜在元数据服务器和数据移动器上创建相同的永久磁带设备链接。将相同的行添加至 */etc/devlink.tab* 文件，删除 */dev/rmt* 中的链接，并运行 *devfsadm -c tape*。

   在示例中，潜在元数据服务器是 *sharefs-mds_alt*，数据移动器客户机是 *sharefs-client1*。因此，编辑这两者中的 */etc/devlink.tab* 文件，使其与活动服务器 *sharefs-mds* 中的文件匹配。然后，删除 *sharefs-mds_alt* 和 *sharefs-client1* 上 */dev/rmt* 中的现有链接，并针对这两者运行 *devfsadm -c tape*：

```
[sharefs-mds]root@solaris:~# ssh 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 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. 现在，配置归档文件系统的主机以便其可以使用归档存储。

配置归档文件系统的主机以使用归档存储

针对活动元数据服务器以及每台潜在元数据服务器和数据移动器客户机，执行如下操作：

1. 以 *root* 用户身份登录主机。

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

2. 在文本编辑器中打开 */etc/opt/SUNWsamfs/mcf* 文件。

   在示例中，使用 *vi* 编辑器。

```
[sharefs-host]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
sharefs                  100       ms        sharefs on
 /dev/dsk/c1t3d0s3        101       md        sharefs on
 /dev/dsk/c1t3d0s4        102       md        sharefs on
...
```

3. 在 */etc/opt/SUNWsamfs/mcf* 文件中的文件系统定义之后，开始归档存储设备的部分。

   在示例中，添加一些标题进行说明：

```
[sharefs-host]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ ----------------
```

4. 要添加归档磁带存储，请先添加一个库条目。在设备标识符字段中，输入库的设备 ID 并分配设备序号：

   在本示例中，库设备标识符为 */dev/scsi/changer/c1t0d5*。我们将设备序号设置为 *900*（即所选磁盘归档范围的下一个范围）：

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ ----------------
/dev/scsi/changer/c1t0d5 900
```

5. 将设备类型设置为 *rb*（一个通用 SCSI 连接磁带库），为磁带库系列集提供一个名称，并将设备状态设置为 *on*。

   在本示例中，我们将使用库 *library1*：

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
```

```
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ ----------------
/dev/scsi/changer/c1t0d5 900       rb        library1  on
```

6. 在 *Additional Parameters* 列中，可以输入可选的用户定义的库目录路径和名称。

   可选的非默认路径不能超过 127 个字符。在示例中，使用默认路径 *var/opt/SUNWsamfs/catalog/*，用户定义的目录文件名为 *library1cat*。请注意，由于文档布局的限制，示例中对路径进行了缩写：

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ ----------------
/dev/scsi/changer/c1t0d5 900       rb        library1  on     .../library1cat
```

7. 接下来，针对每个磁带机添加一个条目。使用我们在“使用永久绑定将磁带机连接到服务器和数据移动器主机 ”过程中生成的永久设备标识符。

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------  ------ -----------------
DISKVOL1                  800       ms        DISKVOL1  on
 /dev/dsk/c6t0d1s7         801       md        DISKVOL1  on
 /dev/dsk/c4t0d2s7         802       md        DISKVOL1  on
/dev/scsi/changer/c1t0d5  900       rb        library1  on     .../library1cat
 /dev/rmt/60cbn            901       tp        library1 on
 /dev/rmt/61cbn            902       tp        library1  on
 /dev/rmt/62cbn            903       tp        library1  on
 /dev/rmt/63cbn            904       tp        library1  on
```

8. 最后，如果您希望自行配置 Oracle HSM 历史记录，请使用设备类型 *hy* 添加一个条目。在系列集和设备状态列中输入一个连字符，并在附加参数列中输入历史记录目录的路径。

   历史记录是一个虚拟库，用于对已从归档中导出的卷进行编录。如果您未配置历史记录，则软件会使用指定的最高设备序号增加一来自动创建一个历史记录。

   请注意，由于页面布局的限制，我们在示例中缩写了历史记录目录的路径。完整路径是 */var/opt/SUNWsamfs/catalog/historian_cat*：

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ ---------------
/dev/scsi/changer/c1t0d5 900       rb        library1  on     ...catalog/library1cat
 /dev/rmt/60cbn           901       tp        library1  on
 /dev/rmt/61cbn           902       tp        library1  on
 /dev/rmt/62cbn           903       tp        library1  on
 /dev/rmt/63cbn           904       tp        library1  on
historian                999       hy        -         -      .../historian_cat
```

9. 保存 *mcf* 文件并关闭编辑器。

```
...
 /dev/rmt/3cbn            904       tp        library1  on
historian                999       hy        -      -      .../historian_cat
:wq
[sharefs-host]root@solaris:~#
```

10. 运行 *sam-fsd* 命令检查 *mcf* 文件是否存在错误。更正发现的任何错误。

    sam-fsd 命令会读取 Oracle HSM 配置文件并初始化文件系统。该命令会在遇到以下错误时停止：

```
[sharefs-host]root@solaris:~# sam-fsd
...
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
[sharefs-host]root@solaris:~#
```

11. 指示 Oracle HSM 服务重新读取 *mcf* 文件并相应地重新配置自身。更正报告的任何错误并根据需要重复。

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

12. 重复该过程，直到所有活动元数据服务器和潜在元数据服务器以及所有数据移动器客户机都已配置为使用归档存储。
13. 如果需要，将磁带 I/O 分布到共享归档文件系统的主机上。
14. 如果计划使用边带数据库功能，请转至第 10 章 配置报告数据库。

15. 否则，请转至第 11 章 配置通知和日志记录。

将磁带 I/O 分布到共享归档文件系统的主机上

从 Oracle HSM 发行版 6.1 开始，在 Oracle 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）。

要将磁带 I/O 分布到共享归档文件系统的主机上，请执行如下操作：

1. 将要用于分布式 I/O 的所有设备连接到文件系统元数据服务器和将处理磁带 I/O 的所有文件系统客户机。
2. 如果您尚未执行此操作，则使用永久绑定将磁带机连接到将用作数据移动器的每个客户机。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。

   在本示例中，服务器的主机名是 *samsharefs-mds*：

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

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

   在示例中，使用 *ssh* 远程登录到客户机主机 *samsharefs-client1* 和 *samsharefs-client2* 并从登录标题处获取 Solaris 版本：

```
[samsharefs-mds]root@solaris:~# ssh root@samsharefs-client1
Password:
Oracle Corporation      SunOS 5.11      11.1    September 2013
[samsharefs-client1]root@solaris:~# exit
[samsharefs-mds]root@solaris:~# ssh root@samsharefs-client2
Password:
Oracle Corporation      SunOS 5.11      11.1    September 2013
[samsharefs-client2]root@solaris:~# exit
[samsharefs-mds]root@solaris:~#
```

6. 计算可以为分布式 I/O 配置中的每个磁带机分配作为缓冲区空间的系统内存量。将可用总内存除以磁带机数量并减去合理安全边界：

   ```
   (total-memory bytes)/(drive-count drives) = memory bytes/drive
   (memory bytes/drive) - (safe-margin bytes/drive) = buffsize bytes/drive
   ```

   Oracle HSM 为使用的每个磁带机分配一个缓冲区。所以，确保不会在无意中配置大于系统内存可以提供的缓冲区空间。在本示例中，我们发现每个磁带机可以分配不超过 224 KB。所以，我们向下舍入到 128 以允许一个安全边界。

   ```
   ((3584 kilobytes)/(16 drives)) = 224 kilobytes/drive
   buffsize = 128 kilobytes/drive
   ```

7. 计算了可以分配给每个磁带机的缓冲区大小后，计算 Oracle HSM 设备块大小以及将适合指定大小的缓冲区的块数。

   ```
   (number blocks/buffer)*block-size bytes/block/drive = buffersize bytes/drive
   ```

   改变块数和块大小，直到两者的积小于或等于计算的缓冲区大小。块数必须处于 *[2-8192]* 范围内。在本示例中，我们确定每个缓冲区有两个块，每块为 64 KB：

   ```
   (2 blocks/buffer)*(64 kilobytes/block/drive) = 128 kilobytes/drive
   ```

8. 在元数据服务器上，在文本编辑器中打开 */etc/opt/SUNWsamfs/archiver.cmd* 文件。在文件顶部的常规指令部分中的新行中，输入 *bufsize = media-type media-blocks*，其中：
   - *media-type* 是 *mcf* 文件为用于分布式 I/O 的磁带机和介质分配的类型代码。
   - *media-blocks* 是您上面计算的每个缓冲区包含的块数。

   保存文件并关闭编辑器。

   在本示例中，我们登录到服务器 *samsharefs-mds* 并使用 *vi* 编辑器添加行 *bufsize = ti 2*，其中 *ti* 是我们使用的 Oracle StorageTek T10000 磁带机的介质类型，*2* 是我们计算的每个磁带机缓冲区包含的块数：

```
[samsharefs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
# archiver.cmd
#----------------------------------------------------------------------
# General Directives
archivemeta = off
examine = noscan
bufsize = ti 2
:wq
[samsharefs-mds]root@solaris:~#
```

9. 在元数据服务器上，在文本编辑器中打开 */etc/opt/SUNWsamfs/defaults.conf* 文件。对于将参与分布式 I/O 的每种介质类型，输入格式为 *media-type_blksize=size* 的行，其中：
   - *media-type* 是 *mcf* 文件为用于分布式 I/O 的磁带机和介质分配的类型代码。
   - *size* 是您在此过程中先前计算的块大小。

   默认情况下，StorageTek T10000 磁带机的设备块大小为 2 MB 或 2048 KB (*ti_blksize = 2048*)。所以，在本示例中，我们使用计算的块大小（64 KB）覆盖默认值：

```
[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.
...
#li_blksize = 256
ti_blksize = 64
[samsharefs-mds]root@solaris:~#
```

10. 仍在 */etc/opt/SUNWsamfs/defaults.conf* 文件中时，如果需要，取消注释行 *#distio = off*，如果它不存在，则添加该行。

    默认情况下，*distio* 处于 *off*（禁用）状态。在本示例中，我们添加行 *distio = on*：

```
...
distio = on
```

11. 仍在 */etc/opt/SUNWsamfs/defaults.conf* 文件中时，启用应参与分布式 I/O 的每种设备类型。在新行中，输入 *media-type _distio = on*，其中 *media-type* 是 *mcf* 文件为磁带机和介质分配的类型代码。

    默认情况下，允许 StorageTek T10000 驱动器和 LTO 驱动器参与分布式 I/O（*ti_distio = on* 且 *li_distio = on*），并排除所有其他类型。在本示例中，我们显式包括 StorageTek T10000 磁带机：

```
...
distio = on
ti_distio = on
```

12. 仍在 */etc/opt/SUNWsamfs/defaults.conf* 文件中时，禁用不应参与分布式 I/O 的每种设备类型。在新行中，输入 *media-type _distio = off*，其中 *media-type* 是 *mcf* 文件为磁带机和介质分配的类型代码。

在示例中，排除 LTO 驱动器：

```
...
distio = on
ti_distio = on
li_distio = off
```

13. 编辑完 */etc/opt/SUNWsamfs/defaults.conf* 文件后，保存内容并关闭编辑器。

```
...
distio = on
ti_distio = on
li_distio = off
:wq
[samsharefs-mds]root@solaris:~#
```

14. 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。
15. 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，并更新该文件以包括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* 编辑器配置主机 *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
```

16. 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 *samsharefs-client1* 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     .../library1cat
 /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:~#
```

17. 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。然后保存文件并关闭编辑器。

    在示例中，未在主机 *samsharefs-client2* 上配置磁带库，因此使用连字符作为磁带设备的系列集名称：

```
[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:~#
```

18. 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，请登录服务器，在文本编辑器中打开 */etc/opt/SUNWsamfs/archiver.cmd* 文件并向 copy 指令添加 *-distio* 参数。设置 *-distio on* 以启用分布式 I/O 或设置 *-distio off* 以禁用分布式 I/O。保存文件。

    在示例中，登录服务器 *samsharefs-mds*，并使用 *vi* 编辑器将副本 *1* 的分布式 I/O 设为 *off*：

```
[samsharefs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
# archiver.cmd
...
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 10m -startsize 500M -startcount 500000 -distio off
allfiles.2 -startage 24h -startsize 20G  -startcount 500000 -reserve set
:wq
[samsharefs-mds]root@solaris:~#
```

19. 运行 *sam-fsd* 命令以检查配置文件中是否有错误。更正发现的任何错误。

    sam-fsd 命令会读取 Oracle HSM 配置文件并初始化文件系统。如果遇到错误，该命令将停止。在示例中，在服务器 *sharefs-mds* 上运行命令：

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

20. 告知 Oracle HSM 服务读取修改的配置文件，并相应重新配置自身。更正报告的任何错误并根据需要重复。

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

21. 要验证是否已成功激活分布式 I/O，请使用命令 *samcmd g*。如果 *DATAMOVER* 标志显示在客户机的输出中，则表示已成功激活分布式 I/O。

    在示例中，显示了该标志：

```
[samsharefs-mds]root@solaris:~# samcmd g
Shared clients samcmd 6.0.dist_tapeio 11:09:13 Feb 20 2014
samcmd on samsharefs-mds
samsharefs is shared, server is samsharefs-mds, 2 clients 3 max
ord hostname             seqno nomsgs status   config  conf1  flags
  1 samsharefs-mds          14      0   8091  808540d   4051      0 MNT SVR

    config   :   CDEVID      ARCHIVE_SCAN    GFSID   OLD_ARCHIVE_FMT
    "        :   SYNC_META   TRACE   SAM_ENABLED     SHARED_MO
    config1  :   NFSV4_ACL   MD_DEVICES      SMALL_DAUS      SHARED_FS
    flags    :
    status   :   MOUNTED     SERVER  SAM     DATAMOVER
    last_msg :  Wed Jul  2 10:13:50 2014

  2 samsharefs-client1     127      0   a0a1  808540d   4041      0 MNT CLI

    config   :   CDEVID      ARCHIVE_SCAN    GFSID   OLD_ARCHIVE_FMT
```

```
    "         :   SYNC_META    TRACE    SAM_ENABLED      SHARED_MO
    config1   :   NFSV4_ACL    MD_DEVICES       SHARED_FS
    flags     :
    status    :   MOUNTED      CLIENT   SAM      SRVR_BYTEREV
    "         :   DATAMOVER
...
```

22. 如果计划使用边带数据库功能，请转至第 10 章 配置报告数据库。
23. 否则，请转至第 11 章 配置通知和日志记录。

使用 NFS 和 SMB/CIFS 从多台主机访问文件系统

多台主机可以使用网络文件系统 (Network File System, NFS) 或服务器消息块 (Server Message Block, SMB)/通用 Internet 文件系统 (Common Internet File System, CIFS) 来访问 Oracle HSM 文件系统，以取代或补充 Oracle HSM 软件对多主机文件系统访问的本机支持（请参见“使用 Oracle HSM 软件从多台主机访问文件系统”）。下面几节概述了基本配置步骤：

- 使用 NFS 共享 Oracle HSM 文件系统
- 使用 SMB/CIFS 共享 Oracle HSM 文件系统

使用 NFS 共享 Oracle HSM 文件系统

执行以下任务：

- 在使用 NFS 4 共享 Oracle HSM 共享文件系统之前禁用委托
- 配置 NFS 服务器和客户机以共享 WORM 文件和目录（如果需要）
- 在 Oracle HSM 主机上配置 NFS 服务器
- 以 NFS 共享来共享 Oracle HSM 文件系统
- 在 NFS 客户机上挂载 NFS 共享的 Oracle HSM 文件系统

在使用 NFS 4 共享 Oracle HSM 共享文件系统之前禁用委托

如果您使用 NFS 来共享 Oracle HSM 共享文件系统，需要确保 Oracle HSM 软件会控制对文件的访问而不受 NFS 的干扰。这通常没有问题，因为在 NFS 服务器代表其客户机访问文件时，它是作为 Oracle HSM 共享文件系统的客户机进行操作的。但是，如果 NFS 版本 4 服务器配置为将读写访问控制委托给其客户机，则可能会发生问题。委托比较有吸引力，因为服务器只需要为阻止可能的冲突而进行干涉。服务器的部分工作负荷会分布于 NFS 客户机，网络流量也会减小。但是，委托会授予独立于 Oracle HSM 服务器的权限（特别是写访问权限），该权限也用于控制从其自身的共享文件系统客户机进行的访问。为防止冲突和可能的文件损坏，您必须禁用委托。请执行如下操作。

1. 登录到您要配置为 NFS 共享的 Oracle HSM 文件系统的主机。以 *root* 用户身份登录。

如果文件系统为 Oracle HSM 共享文件系统，请登录到该文件系统的元数据服务器。在下面的示例中，服务器名称为 *qfsnfs*。

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

2. 如果您使用的是 NFS 版本 4 且 NFS 服务器运行的是 Solaris 11.1 或更高版本，请使用服务管理工具 (Service Management Facility, SMF) 的 *sharectl set -p* 命令将 NFS *server_delegation* 属性设置为 *off*。

```
[qfsnfs]root@solaris:~# sharectl set -p server_delegation=off
```

3. 如果您使用的是 NFS 版本 4 且 NFS 服务器运行的是 Solaris 11.0 或更低版本，请在文本编辑器中打开 */etc/default/nfs* 文件并将 *NFS_SERVER_DELEGATION* 参数设置为 *off*，以禁用委托。保存文件并关闭编辑器。

   在示例中，使用 *vi* 编辑器：

```
[qfsnfs]root@solaris:~# vi /etc/default/nfs
# ident "@(#)nfs        1.10    04/09/01 SMI"
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
...
NFS_SERVER_DELEGATION=off
:wq
[qfsnfs]root@solaris:~#
```

4. 如果您要共享的 Oracle HSM 文件系统支持一次写入多次读取 (Write-Once Read-Many, WORM) 功能，则您现在应该配置 NFS 服务器和客户机以共享 WORM 文件和目录。
5. 否则，在 Oracle HSM 主机上配置 NFS 服务器。

配置 NFS 服务器和客户机以共享 WORM 文件和目录

1. 登录到您想使用 NFS 进行共享的 Oracle HSM 文件系统的主机。以 *root* 用户身份登录。

   如果文件系统为 Oracle HSM 共享文件系统，请登录到该文件系统的元数据服务器。在下面的示例中，服务器名称为 *qfsnfs*，客户机名称为 *nfsclient1*。

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

2. 如果您要共享的 Oracle HSM 文件系统使用 WORM 功能，且托管于运行 Oracle Solaris 10 或更高版本的服务器上，请确保已在 NFS 服务器和所有客户机上启用 NFS 版本 4。

在示例中，检查服务器 *qfsnfs* 和客户机 *nfsclient1*。在每种情况下，先使用 *uname -r* 命令检查 Solaris 版本级别。然后，将 *modinfo* 命令的输出通过管道传输到 *grep* 和一个正则表达式，以查找 NFS 版本信息：

```
[qfsnfs]root@solaris:~# uname -r
5.11
[qfsnfs]root@solaris:~# modinfo | grep -i "nfs.* version 4"
258 7a600000  86cd0  28   1  nfs (network filesystem version 4)
[qfsnfs]root@solaris:~# ssh root@nfsclient1
Pasword: ...
[nfsclient1]root@solaris:~# uname -r
5.11
[nfsclient1]root@solaris:~# modinfo | grep -i "nfs.* version 4"
278 ffffffff8cba000  9df68  27   1  nfs (network filesystem version 4)
[nfsclient1]root@solaris:~# exit
[qfsnfs]root@solaris:~#
```

3. 如果未在运行 Oracle Solaris 10 或更高版本的服务器上启用 NFS 版本 4，请以 *root* 用户身份登录到服务器和每台客户机。然后，使用 *sharectl set* 命令启用 NFS 4：

```
[qfsnfs]root@solaris:~# sharectl set -p server_versmax=4 nfs
[qfsnfs]root@solaris:~# ssh root@nfsclient1
Password ...
[nfsclient1]root@solaris:~# sharectl set -p server_versmax=4 nfs
[nfsclient1]root@solaris:~# exit
[qfsnfs]root@solaris:~#
```

4. 接下来，在 Oracle HSM 主机上配置 NFS 服务器。

在 Oracle HSM 主机上配置 NFS 服务器

您必须配置网络文件系统 (Network File System, NFS) 服务器，使其不要在文件系统成功挂载到主机上之前尝试共享 Oracle HSM 文件系统，客户机才能成功使用 NFS 挂载 Oracle HSM 文件系统。在 Oracle Solaris 10 和操作系统的后续版本下，服务管理工具 (Service Management Facility, SMF) 会管理引导时文件系统的挂载。如果您不使用下面的过程配置 NFS，QFS 挂载或 NFS 共享中的一个会成功，另一个会失败。

1. 登录到您要配置为 NFS 共享的 Oracle HSM 文件系统的主机。以 *root* 用户身份登录。

   如果文件系统为 Oracle HSM 共享文件系统，请登录到该文件系统的元数据服务器。在下面的示例中，服务器名称为 *qfsnfs*。

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

2. 重定向 *svccfg export /network/nfs/server* 命令的输出，以将现有的 NFS 配置导出到一个 XML 清单文件。

   在示例中，将导出的配置定向到清单文件 */var/tmp/server.xml*：

```
[qfsnfs]root@solaris:~# svccfg export /network/nfs/server > /var/tmp/server.xml
[qfsnfs]root@solaris:~#
```

3. 在文本编辑器中打开该清单文件并找到 *filesystem-local* 依赖项。

   在示例中，用 *vi* 编辑器打开文件。*filesystem-local* 依赖项的条目就列在 *nfs-server_multi-user-server* 依赖项的条目前面：

```
[qfsnfs]root@solaris:~# vi /var/tmp/server.xml
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle type='manifest' name='export'>
  <service name='network/nfs/server' type='service' version='0'>
    ...
    <dependency name='filesystem-local' grouping='require_all' restart_on='error' type='service'>
      <service_fmri value='svc:/system/filesystem/local'/>
    </dependency>
    <dependent name='nfs-server_multi-user-server' restart_on='none'
        grouping='optional_all'>
      <service_fmri value='svc:/milestone/multi-user-server'/>
    </dependent>
    ...
```

4. 紧接着 *filesystem-local* 依赖项，添加一个 *qfs* 依赖项来挂载 QFS 共享文件系统。然后保存文件并退出编辑器。

   这样在挂载 Oracle HSM 共享文件系统之后，服务器才会尝试通过 NFS 共享该系统：

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle type='manifest' name='export'>
  <service name='network/nfs/server' type='service' version='0'>
    ...
    <dependency name='filesystem-local' grouping='require_all' restart_on='error' type='service'>
      <service_fmri value='svc:/system/filesystem/local'/>
    </dependency>
    <dependency name='qfs' grouping='require_all' restart_on='error' type='service'>
      <service_fmri value='svc:/network/qfs/shared-mount:default'/>
```

```
    </dependency>
    <dependent name='nfs-server_multi-user-server' restart_on='none'
        grouping='optional_all'>
      <service_fmri value='svc:/milestone/multi-user-server'/>
    </dependent>
:wq
[qfsnfs]root@solaris:~#
```

5. 使用 *svccfg validate* 命令验证清单文件。

```
[qfsnfs]root@solaris:~# svccfg validate /var/tmp/server.xml
```

6. 如果 *svccfg validate* 命令报告错误，请更正错误并重新验证该文件。

   在示例中，*svccfg validate* 命令返回了 XML 解析错误。我们在保存文件时无意中漏掉了结束标记 *</dependency>*。因此，我们再次在 *vi* 编辑器中打开该文件并更正问题：

```
[qfsnfs]root@solaris:~# svccfg validate /var/tmp/server.xml
/var/tmp/server.xml:75: parser error : Opening and ending tag mismatch: dependency line 29 and
 service
  </service>
            ^
/var/tmp/server.xml:76: parser error : expected '>'
</service_bundle>
         ^
/var/tmp/server.xml:77: parser error : Premature end of data in tag service_bundle line 3
^
svccfg: couldn't parse document
[qfsnfs]root@solaris:~# vi /var/tmp/server.xml
...
:wq
[qfsnfs]root@solaris:~#
```

7. 在 *svccfg validate* 命令无错误完成后，使用 *svcadm disable nfs/server* 命令禁用 NFS。

   在示例中，*svccfg validate* 命令未返回任何输出，因此文件有效，我们可以禁用 NFS：

```
[qfsnfs]root@solaris:~# svccfg validate /var/tmp/server.xml
[qfsnfs]root@solaris:~# svcadm disable nfs/server
```

8. 使用 *svccfg delete nfs/server* 命令删除现有的 NFS 服务器配置。

```
[qfsnfs]root@solaris:~# svccfg delete nfs/server
```

9. 使用 *svccfg import* 命令将清单文件导入到服务管理工具 (Service Management Facility, SMF) 中。

```
[qfsnfs]root@solaris:~# svccfg import /var/tmp/server.xml
```

10. 使用 *svcadm enable nfs/server* 命令重新启用 NFS。

    NFS 已配置为使用更新的配置。

```
[qfsnfs]root@solaris:~# svcadm enable nfs/server
```

11. 确认 *qfs* 依赖项已应用。确保命令 *svcs -d svc:/network/nfs/server:default* 显示 */network/qfs/shared-mount:default* 服务：

```
[qfsnfs]root@solaris:~# svcs -d svc:/network/nfs/server:default
STATE          STIME    FMRI
...
online         Nov_01   svc:/network/qfs/shared-mount:default
...
```

12. 接下来，以 NFS 共享来共享 Oracle HSM 文件系统。

以 NFS 共享来共享 Oracle HSM 文件系统

使用您的 Oracle Solaris 操作系统版本的管理文档中介绍的过程，来共享 Oracle HSM 文件系统。以下步骤概述了 Solaris 11.1 的过程：

1. 登录到您想使用 NFS 进行共享的 Oracle HSM 文件系统的主机。以 *root* 用户身份登录。

   如果文件系统为 Oracle HSM 共享文件系统，请登录到该文件系统的元数据服务器。在下面的示例中，服务器名称为 *qfsnfs*。

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

2. 输入命令行 *share -F nfs -o sharing-options sharepath*，其中 *-F* 开关指定 *nfs* 共享协议，*sharepath* 是共享资源的路径。如果使用可选的 *-o* 参数，则 *sharing-options* 可以包含以下任一项：
   - *rw* 会使所有客户机具有 *sharepath* 的读写权限。
   - *ro* 会使所有客户机具有 *sharepath* 的只读权限。
   - *rw=clients* 会使 *clients*（具有共享资源访问权限的一台或多台客户机的冒号分隔列表）具有 *sharepath* 的读写权限。

- *ro=clients* 会使 *clients*（具有共享资源访问权限的一台或多台客户机的冒号分隔列表）具有 *sharepath* 的只读权限。

在本示例中，我们以读取/写入方式与客户机 *nfsclient1* 和 *nfsclient2* 共享 */qfsms* 文件系统，以只读方式与 *nfsclient3* 共享该文件系统（请注意，下面的命令是作为单行输入的－使用反斜杠对换行符进行转义）：

```
[qfsnfs]root@solaris:~# share -F nfs -o rw=nfsclient1:nfsclient2 /
 ro=nfsclient3 /qfsms
```

输入命令后，系统会自动重新启动 NFS 服务器守护进程 *nfsd*。有关其他选项和详细信息，请参见 *share_nfs* 手册页。

3. 使用命令行 *share -F nfs* 检查共享参数。

   在示例中，命令输出表明我们已正确配置共享：

```
[qfsnfs]root@solaris:~# share -F nfs
/qfsms   sec=sys,rw=nfsclient1:nfsclient2,ro=nfsclient3
[qfsnfs]root@solaris:~#
```

4. 接下来，在 NFS 客户机上挂载 NFS 共享的 Oracle HSM 文件系统。

在 NFS 客户机上挂载 NFS 共享的 Oracle HSM 文件系统

在客户机系统中，将 NFS 服务器的文件系统挂载到近便的挂载点。针对每个客户机，执行如下操作：

1. 以 *root* 用户身份登录客户机。

   在示例中，NFS 客户机名称为 *nfsclient1*：

```
[nfsclient1]root@solaris:~#
```

2. 备份操作系统的 */etc/vfstab* 文件。

```
[nfsclient1]root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
[nfsclient1]root@solaris:~#
```

3. 在文本编辑器中打开 */etc/vfstab* 文件。

   在示例中，使用 *vi* 编辑器。

```
[nfsclient1]root@solaris:~# vi /etc/vfstab
#File         Device                         Mount
#Device       to      Mount     System fsck  at     Mount
```

```
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ---------------
/devices      -       /devices  devfs  -     no     -
...
```

4. 在 */etc/vfstab* 文件的第一列中，指定 NFS 服务器的名称和您要共享的文件系统的挂载点（以冒号分隔），来命名文件设备。

   在示例中，NFS 服务器名为 *qfsnfs*，共享文件系统名为 *qfsms*，服务器上的挂载点为 */qfsms*：

```
#File         Device                         Mount
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ---------------
/devices      -       /devices  devfs  -     no     -
...
qfsnfs:/qfsms
```

5. 在 */etc/vfstab* 文件的第二列中，输入连字符 (-)，使本地系统不会尝试检查远程文件系统是否一致：

```
#File         Device                         Mount
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ---------------
/devices      -       /devices  devfs  -     no     -
...
qfsnfs:/qfsms -
```

6. 在 */etc/vfstab* 文件的第三列中，输入您要挂载远程文件系统的本地挂载点。

   在示例中，挂载点将为目录 */qfsnfs*：

```
#File         Device                         Mount
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ---------------
/devices      -       /devices  devfs  -     no     -
...
qfsnfs:/qfsms -       /qfsnfs
```

7. 在 */etc/vfstab* 文件的第四列中，输入文件系统类型 *nfs*。

```
#File         Device                          Mount
```

```
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ---------------
/devices      -       /devices  devfs  -     no     -
...
qfsnfs:/qfsms -       /qfsnfs   nfs
```

将使用 *nfs* 文件系统类型，因为客户机会将远程 QFS 文件系统作为 NFS 文件系统进行挂载。

8. 在 */etc/vfstab* 文件的第五列中，输入连字符 (-)，因为本地系统不会尝试检查远程文件系统是否一致。

```
#File         Device                        Mount
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ---------------
/devices      -       /devices  devfs  -     no     -
...
qfsnfs:/qfsms -       /qfsnfs   nfs    -
```

9. 在 */etc/vfstab* 文件的第六列中，输入 *yes* 在引导时挂载远程文件系统，或输入 *no* 根据需要手动挂载。

   在示例中，输入 *yes*：

```
#File         Device                        Mount
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ---------------
/devices      -       /devices  devfs  -     no     -
...
qfsnfs:/qfsms -       /qfsnfs   nfs    -     yes
```

10. 在 */etc/vfstab* 文件的最后一列中，输入 *hard* 和 *intr* NFS 挂载以强制进行无限次不间断的重试，或输入 *soft*、*retrans* 和 *timeo* 挂载选项（将 *retrans* 设置为 *120* 或更大，将 *timeo* 设置为 *3000* 十分之一秒），以设置指定重试次数。

    设置 *hard* 重试选项或指定超时足够长和重试次数足够多的 *soft* 选项，可防止 NFS 请求在请求的文件位于无法立即挂载的可移除卷上时发生失败。有关这些挂载选项的更多信息，请参见 Solaris *mount_nfs* 手册页。

    在示例中，输入 *soft* 挂载选项：

```
#File         Device                        Mount
```

```
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ---------------
/devices      -       /devices  devfs  -     no     -
...
qfsnfs:/qfsms -       /qfsnfs   nfs    -     yes    soft,retrans=120,timeo=3000
```

11. 如果您使用的是 NFS 2，请将 *rsize* 挂载参数设置为 *32768*。

    对于其他版本的 NFS，请输入默认值。

    rsize 挂载参数会将读取缓冲区大小设置为 *32768* 字节（默认为 *8192* 字节）。示例显示了 NFS 2 配置：

```
#File         Device                         Mount
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ---------------
/devices      -       /devices  devfs  -     no     -
...
qfsnfs2:/qfs2 -       /qfsnfs2  nfs    -     yes    ...,rsize=32768
```

12. 如果您使用的是 NFS 2，请将 *wsize* 挂载参数设置为 *32768*。

    对于其他版本的 NFS，请输入默认值。

    wsize 挂载参数会将写入缓冲区大小设置为指定的字节数（默认为 *8192* 字节）。示例显示了 NFS 2 配置：

```
#File         Device                         Mount
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ---------------
/devices      -       /devices  devfs  -     no     -
...
qfsnfs2:/qfs2 -       /qfsnfs2  nfs    -     yes    ...,wsize=32768
```

13. 保存 */etc/vfstab* 文件并退出编辑器。

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

```
qfsnfs:/qfsms -       /qfsnfs   nfs    -     yes    soft,retrans=120,timeo=3000
:wq
[nfsclient1]root@solaris:~#
```

14. 在共享文件系统上创建挂载点目录。

    在示例中，将共享文件系统挂载到名为 */qfsnfs* 的目录上：

```
[nfsclient1]root@solaris:~# mkdir /qfsnfs
[nfsclient1]root@solaris:~#
```

15. 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。

    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 */qfsnfs* 挂载点目录并将权限设置为 *755* (*-rwxr-xr-x*)：

```
[nfsclient1]root@solaris:~# mkdir /qfsnfs
[nfsclient1]root@solaris:~# chmod 755 /qfsnfs
[nfsclient1]root@solaris:~#
```

16. 挂载共享文件系统：

```
[nfsclient1]root@solaris:~# mount /qfsnfs
[nfsclient1]root@solaris:~#
```

17. 如果计划使用边带数据库功能，请转至第 10 章 配置报告数据库。
18. 否则，请转至第 11 章 配置通知和日志记录。

使用 SMB/CIFS 共享 Oracle HSM 文件系统

SMB 使得 Oracle HSM 可供 Microsoft Windows 主机访问并提供了互操作性功能，例如不区分大小写、DOS 属性支持以及 NFSv4 访问控制列表 (Access Control List, ACL) 支持。Oracle Solaris OS 提供了一种服务器消息块 (Server Message Block, SMB) 协议服务器和客户机实现，该实现支持包含 NT LM 0.12 和通用 Internet 文件系统 (Common Internet File System, CIFS) 在内的众多 SMB 版本 (dialect)。

Oracle HSM 支持 Windows 安全标识符 (Security Identifier, SID)。Windows 标识符不再需要使用 *idmap* 服务显式定义，或由 Active Directory 服务提供。

要为 Oracle HSM 文件系统配置 SMB 服务，请执行以下任务：

- 查看 Oracle Solaris SMB 配置和管理文档。
- 为 SMB 服务器显式映射 Windows 标识符（可选）。
- 配置 Oracle HSM 文件系统以与 SMB/CIFS 共享。
- 针对 Windows Active Directory 域或工作组配置 SMB 服务器。

- 以 SMB/CIFS 共享来共享 Oracle HSM 文件系统。

查看 Oracle Solaris SMB 配置和管理文档

下面几节概述了适用于 Oracle HSM 文件系统的部分 SMB 配置过程。这些内容并不全面，并不涵盖所有可能的情况。因此，请查看配置 Oracle Solaris SMB 服务器的完整说明，将服务器集成到现有的 Windows 环境，并将 SMB 共享挂载到 Solaris 系统上。有关完整说明，请参见 *Oracle Solaris Information Library* 中的《*Managing SMB and Windows Interoperability in Oracle Solaris*》。

为 SMB 服务器显式映射 Windows 标识符（可选）

虽然 Oracle HSM 现在完全支持 Windows 安全标识符 (Security Identifier, SID)，但是在有些情况下，显式定义 UNIX 标识符和 SID 之间的关系仍然有一定优势。例如，在用户同时拥有 UNIX 和 Windows 标识符的异源环境中，您可能想要使用 *idmap* 服务或 Active Directory 服务创建显式映射。有关完整的 SMB 和 Windows 互操作性信息，请参见适用于您的 Oracle Solaris 版本的产品文档。

配置 Oracle HSM 文件系统以与 SMB/CIFS 共享

使用 SMB/CIFS 共享的 Oracle HSM 文件系统必须使用在网络文件系统 (Network File System, NFS) 版本 4 和 Oracle Solaris 11 中采用的新访问控制列表 (Access Control List, ACL) 实施。Solaris 和 NFS 的旧版本使用的 ACL 基于和 Windows ACL 实施不兼容的 POSIX 草案规范。

默认情况下，使用 Oracle HSM 创建的新文件系统会使用 Solaris 11 上的 NFS 版本 4 ACL。但是，如果您需要与 SMB/CIFS 客户机共享现有的 Oracle HSM 文件系统，则必须使用适当的过程转换现有的 POSIX 样式 ACL：

- 转换使用 POSIX 样式 ACL 的 Oracle HSM 非共享文件系统
- 转换使用 POSIX 样式 ACL 的 Oracle HSM 共享文件系统

转换使用 POSIX 样式 ACL 的 Oracle HSM 非共享文件系统

执行如下操作：

1. 以 *root* 用户身份登录主机。

   在示例中，登录到主机 *qfs-host*

```
[qfs-host]root@solaris:~#
```

2. 确保主机运行 Oracle Solaris 11.1 或更高版本。使用命令 *uname -r*。

```
[qfs-host]root@solaris:~# uname -r
5.11
[qfs-host]root@solaris:~#
```

3. 使用命令 *umount mount-point* 卸载文件系统，其中 *mount-point* 是 Oracle HSM 文件系统的挂载点。

   有关进一步的详细信息，请参见 *umount_samfs* 手册页。在下面的示例中，服务器名称为 *qfs-host*，文件系统为 */qfsms*：

```
[qfs-host]root@solaris:~# umount /qfsms
```

4. 使用 *samfsck -F -A file-system* 命令转换文件系统，其中 *-F* 选项指定对文件系统进行检查和修复，*-A* 选项指定转换 ACL，*file-system* 是您需要转换的文件系统的名称。

   如果指定 *-A* 选项，则必须指定 *-F* 选项。如果 *samfsck -F -A* 命令返回错误，进程会异常中止，且不会转换任何 ACL（有关这些选项的完整描述，请参见 *samfsck* 手册页）。

```
[qfs-host]root@solaris:~# samfsck -F -A /qfsms
```

5. 如果返回错误且 ACL 未转换，请使用 *samfsck -F -a file-system* 命令强制转换 ACL。

   -a 选项指定强制的转换。如果指定 *-a* 选项，则必须指定 *-F* 选项（有关这些选项的完整描述，请参见 *samfsck* 手册页）。

```
[qfs-host]root@solaris:~# samfsck -F -a /qfsms
```

6. 现在，针对 Windows Active Directory 域或工作组配置 SMB 服务器。

转换使用 POSIX 样式 ACL 的 Oracle HSM 共享文件系统

1. 以 *root* 用户身份登录到文件系统元数据服务器。

   在示例中，登录到元数据服务器 *sharedqfs-mds*：

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

2. 确保元数据服务器运行 Oracle Solaris 11.1 或更高版本。使用命令 *uname -r*。

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

3. 以 *root* 用户身份登录到每台 Oracle HSM 客户机，并确保每台客户机都运行 Oracle Solaris 11.1 或更高版本。

在示例中，打开终端窗口并使用 *ssh* 远程登录到客户机主机 *sharedqfs-client1* 和 *sharedqfs-client2*，以从登录标题处获取 Solaris 版本：

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

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

4. 使用命令 *umount mount-point* 从每台 Oracle HSM 客户机中卸载 Oracle HSM 共享文件系统，其中 *mount-point* 是 Oracle HSM 文件系统的挂载点。

   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（*sharedqfs-client1* 和 *sharedqfs-client2*）中卸载 */sharedqfs1*：

```
Oracle Corporation      SunOS 5.11      11.1    September 2013
[sharedqfs-client1]root@solaris:~# umount /sharedqfs
[sharedqfs-client1]root@solaris:~#

Oracle Corporation      SunOS 5.11      11.1    September 2013
[sharedqfs-client2]root@solaris:~# umount /sharedqfs
[sharedqfs-client1]root@solaris:~#
```

5. 使用命令 *umount -o await_clients=interval mount-point* 从元数据服务器中卸载 Oracle HSM 共享文件系统，其中 *mount-point* 是 Oracle HSM 文件系统的挂载点，interval 是由 *-o await_clients* 选项以秒为单位指定的用于延迟执行的延迟时间。

   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，从元数据服务器 *sharedqfs-mds* 中卸载 */sharedqfs* 文件系统，同时允许客户机在 *60* 秒内进行卸载：

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

6. 将文件系统从 POSIX 样式 ACL 转换为 NFS 版本 4 ACL。在元数据服务器上，使用命令 *samfsck -F -A file-system*，其中 *-F* 选项指定对文件系统进行检查和修

复，*-A* 选项指定对 ACL 进行转换，*file-system* 是您需要转换的文件系统的名称。

如果指定 *-A* 选项，则必须指定 *-F* 选项。如果 *samfsck -F -A file-system* 命令返回错误，进程会异常中止，且不会转换任何 ACL（有关这些选项的完整描述，请参见 *samfsck* 手册页）。在示例中，转换名为 */sharedqfs* 的 Oracle HSM 文件系统：

```
[sharedqfs-mds]root@solaris:~# samfsck -F -A /sharedqfs
```

7. 如果返回错误且 ACL 未转换，请强制转换 ACL。在元数据服务器上，使用 *samfsck -F -a file-system* 命令。

   -a 选项指定强制的转换。如果指定 *-a* 选项，则必须指定 *-F* 选项（有关这些选项的完整描述，请参见 *samfsck* 手册页）。在示例中，强制转换 Oracle HSM 文件系统 */qfsma*：

```
[sharedqfs-mds]root@solaris:~# samfsck -F -a /sharedqfs
```

8. 现在，针对 Windows Active Directory 域或工作组配置 SMB 服务器。

针对 Windows Active Directory 域或工作组配置 SMB 服务器

Oracle Solaris SMB 服务可以使用两种互斥的模式之一进行操作：域或工作组。请根据您的环境和验证需要选择其中一种模式：

- 如果您需要向 Active Directory 域用户授予 Solaris SMB 服务的访问权限，请在域模式下配置 SMB 服务器。
- 如果您需要向本地 Solaris 用户授予 SMB 服务的访问权，但没有 Active Directory 域或不需要向 Active Directory 域用户授予该服务的访问权限，请在工作组模式下配置 SMB 服务器。

在域模式下配置 SMB 服务器

1. 联系 Windows Active Directory 管理员并获取以下信息：
   - 在加入 Active Directory 域时您需要使用的经过验证的 Active Directory 用户帐户的名称
   - 您需要用来替代帐户的默认 *Computers* 容器的组织单位（如果有）
   - 要共享 Oracle HSM 文件系统的域的全限定 LDAP/DNS 域名。
2. 登录到您要配置为 SMB/CIFS 共享的 Oracle HSM 文件系统的主机。以 *root* 用户身份登录。

   如果文件系统为 Oracle HSM 共享文件系统，请登录到该文件系统的元数据服务器。在下面的示例中，服务器名称为 *qfssmb*。

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

3. 开源 Samba 和 SMB 服务器不能在单个 Oracle Solaris 系统中一起使用。因此，请查看 Samba 服务是否正在运行。将 *svcs* 服务状态命令的输出通过管道传输到 *grep* 和正则表达式 *samba*。

   在示例中，*svcs* 命令的输出包含该正则表达式的匹配项，因此 SMB 服务正在运行：

```
[qfssmb]root@solaris:~# svcs | grep samba
legacy_run     Nov_03   lrc:/etc/rc3_d/S90samba
```

4. 如果 Samba 服务 (*svc:/network/samba*) 正在运行，请禁用该服务以及 Windows Internet 命名服务/WINS (*svc:/network/wins*)（如果正在运行）。使用命令 *svcadm disable*。

```
[qfssmb]root@solaris:~# svcadm disable svc:/network/samba
[qfssmb]root@solaris:~# svcadm disable svc:/network/wins
```

5. 现在使用 *svcadm enable -r smb/server* 命令启动 SMB 服务器及其依赖的任何服务。

```
[qfssmb]root@solaris:~# svcadm enable -r smb/server
```

6. 确保 Oracle HSM 主机上的系统时钟与 Microsoft Windows 域控制器的系统时钟相差不超过五分钟：
   - 如果 Windows 域控制器使用网络时间协议 (Network Time Protocol, NTP) 服务器，请配置 Oracle HSM 主机以使用相同服务器。在 Oracle HSM 主机上创建 */etc/inet/ntpclient.conf* 文件，并使用命令 *svcadm enable ntp* 启动 *ntpd* 守护进程（有关完整信息，请参见 *ntpd* 手册页和您的 Oracle Solaris 管理文档）。
   - 否则，请运行命令 *ntpdate domain-controller-name* 来将 Oracle HSM 主机和域控制器同步（有关详细信息，请参见 *ntpdate* 手册页），或手动将 Oracle HSM 主机上的系统时钟设置为域控制器的系统时钟所显示的时间。
7. 使用命令 *smbadm join -u username -o organizational-unit domain-name* 加入 Windows 域，其中 *username* 是 Active Directory 管理员指定的用户帐户的名称，可选的 *organizational-unit* 是指定的帐户容器（如果有），*domain-name* 是指定的全限定 LDAP 或 DNS 域名。

   在示例中，使用用户帐户加入 Windows 域 *this.example.com*

```
[qfssmb]root@solaris:~# smbadm join -u admin -o smbsharing this.example.com
```

8. 现在，以 SMB/CIFS 共享来共享 Oracle HSM 文件系统。

在工作组模式下配置 SMB 服务器

1. 联系 Windows 网络管理员并获取 Oracle HSM 文件系统的主机应加入的 Windows 工作组的名称。

   默认工作组名为 *WORKGROUP*。
2. 登录到 Oracle HSM 文件系统的主机。以 *root* 用户身份登录。

   如果文件系统为 Oracle HSM 共享文件系统，请登录到该文件系统的元数据服务器。在下面的示例中，服务器名称为 *qfssmb*。

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

3. 开源 Samba 和 SMB 服务器不能在单个 Oracle Solaris 系统中一起使用。因此，请查看 Samba 服务是否正在运行。将 *svcs* 服务状态命令的输出通过管道传输到 *grep* 和正则表达式 *samba*。

   在示例中，*svcs* 命令的输出包含该正则表达式的匹配项，因此 SMB 服务正在运行：

```
[qfssmb]root@solaris:~# svcs | grep samba
legacy_run     Nov_03   lrc:/etc/rc3_d/S90samba
```

4. 如果 Samba 服务 (*svc:/network/samba*) 正在运行，请禁用该服务以及 Windows Internet 命名服务/WINS (*svc:/network/wins*) 服务（如果正在运行）。使用命令 *svcadm disable*。

   Samba 和 SMB 服务器不能在单个 Oracle Solaris 系统中一起使用。

```
[qfssmb]root@solaris:~# svcadm disable svc:/network/samba
[qfssmb]root@solaris:~# svcadm disable svc:/network/wins
```

5. 现在使用命令 *svcadm enable -r smb/server* 启动 SMB 服务器及其依赖的任何服务。

```
[qfssmb]root@solaris:~# svcadm enable -r smb/server
```

6. 加入工作组。使用命令 *smbadm join* 和 *-w*（工作组）开关以及 Windows 网络管理员指定的工作组的名称。

   在示例中，指定的工作组名为 *crossplatform*。

```
[qfssmb]root@solaris:~# smbadm join -w crossplatform
```

7. 配置 Oracle HSM 主机来为 SMB 密码加密。在文本编辑器中打开 */etc/pam.d/other* 文件，添加命令行 *password required pam_smb_passwd.so.1 nowarn* 并保存文件。

   在示例中，使用 *vi* 编辑器：

```
[qfssmb]root@solaris:~# vi /etc/pam.d/other
# Copyright (c) 2012, Oracle and/or its affiliates. All rights reserved.
#
# PAM configuration
#
# Default definitions for Authentication management
# Used when service name is not explicitly mentioned for authentication
#
auth definitive  pam_user_policy.so.1
...
password required pam_authtok_store.so.1
password required pam_smb_passwd.so.1 nowarn
:wq
[qfssmb]root@solaris:~#
```

   有关详细信息，请参见 *pam_smb_passwd* 手册页。

8. 安装 *pam_smb_passwd* 模块后，使用命令 *passwd local-username* 为用户 *local-username* 生成加密版密码，以便 SMB 服务器可以登录到 Windows 工作组。

   SMB 服务器无法使用 Solaris 操作系统使用的相同加密版密码来验证用户身份。在示例中，为用户 *smbsamqfs* 生成加密的 SMB 密码：

```
[qfssmb]root@solaris:~# passwd smbsamqfs
```

9. 现在，以 SMB/CIFS 共享来共享 Oracle HSM 文件系统。

以 SMB/CIFS 共享来共享 Oracle HSM 文件系统

使用您的 Oracle Solaris 操作系统版本的管理文档中介绍的过程，来共享 Oracle HSM 文件系统。以下步骤概述了 Solaris 11.1 的过程：

1. 登录到您要配置为 SMB/CIFS 共享的 Oracle HSM 文件系统的主机。以 *root* 用户身份登录。

   如果文件系统为 Oracle HSM 共享文件系统，请登录到该文件系统的元数据服务器。在下面的示例中，服务器名称为 *qfssmb*。

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

2. 配置共享。使用命令 *share -F smb -o specific-options sharepath sharename*，其中 *-F* 开关指定 *smb* 共享协议，*sharepath* 是共享资源的路径，*sharename* 是您要用于共享的名称。可选的 *-o* 参数的值 *sharing-options* 可以包括以下任一项：

   - *abe=[true|false]*

     当共享的基于访问的枚举 (access-based enumeration, ABE) 策略为 *true* 时，请求访问的用户无权访问的目录条目将从返回到客户机的目录列表中去掉。

   - *ad-container=cn=user,ou=organization,dc=domain-dns*

     Active Directory 容器将共享访问仅限为由轻量目录访问协议 (Lightweight Directory Access Protocol, LDAP) 相对标识名 (relative distinguished name, RDN) 属性值指定的域对象，这些属性值包括：*cn*（用户对象类）、*ou*（组织单位对象类）和 *dc*（域 DNS 对象类）。

     有关将 Active Directory 容器与 SMB/CIFS 配合使用的完整信息，请参见《*Internet Engineering Task Force Request For Comment (RFC) 2253*》以及您的 Microsoft Windows 目录服务文档。

   - *catia=[true|false]*

     当 CATIA 字符替换为 *true* 时，CATIA 版本 4 文件名中的所有在 Windows 中非法的字符都会被合法的等效字符替换。有关替换列表，请参见 *share_smb* 手册页。

   - *csc=[manual|auto|vdo|disabled]*

     客户端高速缓存 (*csc*) 策略控制文件的客户端高速缓存（供脱机使用）。*manual* 策略允许客户机在用户请求时高速缓存文件，但禁止自动基于每个文件重新整合（这是默认值）。*auto* 策略允许客户机自动高速缓存文件，并允许自动基于每个文件重新整合。*vdo* 策略允许客户机自动高速缓存文件以供脱机使用，允许基于每个文件重新整合，并允许客户机在脱机时从本地高速缓存工作。*disabled* 策略不允许客户端高速缓存。

   - *dfsroot=[true|false]*

     在 Microsoft 分布式文件系统 (Distributed File System, DFS) 中，根共享 (*dfsroot=true*) 是指将一组广泛分布的共享文件夹组织成可更方便管理的单个 DFS 文件系统的共享。有关完整信息，请参见您的 Microsoft Windows Server 文档。

   - *guestok=[true|false]*

     当 *guestok* 策略为 *true* 时，本地定义的 *guest* 帐户可以访问共享。当该策略为 *false* 或未定义（默认值）时，*guest* 帐户无法访问共享。该策略可让您将 Windows *Guest* 用户映射到本地定义的 UNIX 用户名，例如 *guest* 或 *nobody*：

```
# idmap add winname:Guest unixuser:guest
```

然后，本地定义的帐户可以通过存储在 */var/smb/smbpasswd* 中的密码进行身份验证（如果需要）。有关更多信息，请参见 *idmap* 手册页。

- *rw=[*|[[-]criterion][:[-]criterion*]...

  rw 策略可授予或拒绝对与提供的访问列表相匹配的任何客户机的访问权限。

  访问列表包含表示全部的单个星号 (*) 或冒号分隔的客户机访问条件列表，其中每个 *criterion* 包含一个可选的减号 (-)（表示拒绝），后跟一个主机名、一个网络组、一个完整的 LDAP 或 DNS 域名和/或符号 *@* 以及完整的 IP 地址或域名或其一部分。会从左向右评估访问列表，直到客户机满足其中一个条件。有关详细信息，请参见 *share_smb* 手册页。
- *ro=[*|[[-]criterion][:[-]criterion*]...

  ro 策略可授予或拒绝对与访问列表相匹配的任何客户机的只读访问权限。
- *none=[*|[[-]criterion][:[-]criterion*]...

  none 策略会拒绝对与访问列表相匹配的任何客户机的访问权限。如果访问列表为星号 (*)，*ro* 和 *rw* 策略可覆盖 *none* 策略。

在示例中，与客户机 *smbclient1* 和 *smbclient2* 共享 */qfsms* 文件系统读/写权限，与客户机 *smbclient3* 共享该文件系统的只读权限：

```
[qfssmb]root@solaris:~# share -F smb -o rw=smbclient1:smbclient2
ro=smbclient3 /qfsms
```

输入命令后，系统会自动重新启动 SMB 服务器守护进程 *smbd*。

3. 检查共享参数。使用命令 *share -F nfs*。

   在示例中，命令输出表明我们已正确配置共享：

```
[qfssmb]root@solaris:~# share -F smb /qfsms
sec=sys,rw=smbclient1:smbclient2,ro=smbclient3
[qfssmb]root@solaris:~#
```

4. 如果计划使用边带数据库功能，请转至第 10 章 配置报告数据库。
5. 否则，请转至第 11 章 配置通知和日志记录。

8

第 8 章 配置 SAM-Remote

Oracle Hierarchical Storage Manager 软件的 SAM-Remote 功能使 Oracle HSM 文件系统主机可以访问在远程 Oracle HSM 文件系统主机上托管的磁带介质和驱动器。本地主机作为远程主机（充当 SAM-Remote 服务器）的 SAM-Remote 客户机访问磁带资源。客户机的归档策略通常在本地磁盘或固态 (SSD) 磁盘归档中维护一个或两个副本，在服务器提供的远程磁带上维护一个或两个副本。每个主机上的主配置文件 */etc/opt/SUNWsamfs/mcf* 使用特殊的 SAM-Remote 设备类型定义共享资源和客户机/服务器关系。

您可以满足 SAM-Remote 客户机和服务器的大量归档和数据保护要求：

- 您可以将磁带归档的优势扩展至缺少库和驱动器的 Oracle HSM 主机。
- 您可以集中维护和管理区域办事处和卫星校园中托管的 Oracle HSM 文件系统的磁带资源。

  在总部中央数据中心中，Oracle HSM 文件系统主机已连接磁带库并作为 SAM-Remote 服务器运行。在较小的分布式办公室中，Oracle HSM 文件系统主机仅具有磁盘归档并充当 SAM-Remote 客户机。所有主机均维护其归档数据的本地和磁带副本。但硬件和介质库存集中在中央数据中心中，可以在此处以最有效率且成本最低的方式对其进行维护。
- 您可以自动创建和维护异地磁带副本以用于备份和灾难恢复。

  所有 Oracle HSM 文件系统主机均已连接磁带库。每个主机作为 SAM-Remote 客户机和服务器（就异地位置中的对应设备而言）运行。每个 Oracle HSM 主机使用本地资源创建本地磁盘和磁带副本。每个主机使用其对应设备提供的资源创建远程磁带副本，并且每个主机为其对应设备提供磁带资源。因此会在正常归档过程中自动创建两个文件系统的异地副本。
- 您可以配置 Oracle HSM 文件系统主机，以便在本地资源不可用时访问远程归档存储资源。

  再次重申，所有 Oracle HSM 文件系统主机已连接磁带库，并且每个主机作为 SAM-Remote 客户机和服务器（就其他位置中的对应设备而言）运行。每个 Oracle HSM 主机使用本地资源创建本地磁盘和磁带副本。但如果主机无法访问其本地库，则仍可以使用其远程对应设备提供的介质和资源归档和检索文件。

本章概述了配置 SAM-Remote 客户机/服务器网络的过程。本章涵盖以下任务：

- 确保所有 SAM-Remote 主机使用相同的软件

确保所有 SAM-Remote 主机使用相同的软件

SAM-Remote 客户机和服务器必须安装 Oracle HSM 软件的相同修订版。请使用以下过程检查修订版级别：

1. 以 *root* 用户身份登录到 SAM-Remote 服务器主机。

   在示例中，服务器主机为 *server1*：

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

2. 以 *root* 用户身份登录到 SAM-Remote 客户机主机。

   在示例中，打开终端窗口并使用 *ssh* 登录到主机 *client1*：

```
[server1]root@solaris:~# ssh root@client1
Password: ...
[client1]root@solaris:~#
```

3. 确保所有 SAM-Remote 服务器和客户机上的 Oracle HSM 软件包的修订版级别是相同的。在每台 SAM-Remote 主机上，使用命令 *samcmd l* 列出配置详细信息。比较结果。

   在示例中，将比较 *server1* 和 *client1* 上的结果。两者都使用 Oracle HSM 软件的相同修订版：

```
[server1]root@solaris:~# samcmd l
Usage information samcmd     6.0  10:20:34 Feb 20 2015
samcmd on server1
...
[server1]root@solaris:~#

[client1]root@solaris:~# samcmd l
Usage information samcmd     6.0  10:20:37 Feb 20 2015
samcmd on client1
...
[server1]root@solaris:~#
```

4. 使用第 4 章 安装 *Oracle HSM and QFS Software* 中的过程根据需要更新主机软件，直到所有 SAM-Remote 服务器和客户机都处于相同的修订版级别。
5. 接下来，停止 Oracle HSM 进程。

停止 Oracle HSM 进程

1. 以 *root* 用户身份登录到 SAM-Remote 服务器主机。

   在示例中，服务器名为 *server1*：

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

2. 获取已配置设备的设备序号。使用命令 *samcmd c*。

   在示例中，设备序号为 *801*、*802*、*803* 和 *804*：

```
[server1]root@solaris:~# samcmd c
Device configuration samcmd     6.0  10:20:34 Feb 20 2015
samcmd on server1
Device configuration:
ty  eq  state   device_name                       fs   family_set
rb  800 on      /dev/scsi/changer/c1t0d5          800  rb800
tp  801 on      /dev/rmt/0cbn                     801  rb800
tp  802 on      /dev/rmt/1cbn                     802  rb800
tp  803 on      /dev/rmt/2cbn                     803  rb800
tp  804 on      /dev/rmt/3cbn                     804  rb800
```

3.
4. 使所有归档进程（如果有）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：

```
[samfs-mds]root@solaris:~# samcmd aridle
[samfs-mds]root@solaris:~#
```

5. 使所有回写进程（如果有）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：

```
[samfs-mds]root@solaris:~# samcmd stidle
[samfs-mds]root@solaris:~#
```

6. 等待任何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：

```
[samfs-mds]root@solaris:~# samcmd a
Archiver status samcmd     6.0 14:20:34 Feb 22 2015
samcmd on samfs-mds
sam-archiverd:  Waiting for :arrun
sam-arfind: ...
Waiting for :arrun
```

7. 等待任何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：

```
[samfs-mds]root@solaris:~# samcmd u
Staging queue samcmd      6.0 14:20:34 Feb 22 2015
samcmd on solaris.demo.lan
Staging queue by media type: all
sam-stagerd:  Waiting for :strun
root@solaris:~#
```

8. 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：

```
[samfs-mds]root@solaris:~# samcmd 801 idle
[samfs-mds]root@solaris:~# samcmd 802 idle
[samfs-mds]root@solaris:~# samcmd 803 idle
[samfs-mds]root@solaris:~# samcmd 804 idle
[samfs-mds]root@solaris:~#
```

9. 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。

```
[samfs-mds]root@solaris:~# samcmd r
Removable media samcmd     6.0 14:20:34 Feb 22 2015
samcmd on samqfs1host
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
[samfs-mds]root@solaris:~#
```

10. 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*。

```
[samfs-mds]root@solaris:~# samd stop
[samfs-mds]root@solaris:~#
```

11. 接下来，配置 SAM-Remote 服务器。

配置 SAM-Remote 服务器

SAM-Remote 服务器是 Oracle HSM 文件系统主机，可允许其连接的机械装置磁带库和磁带机供本身也是 Oracle HSM 文件系统主机的远程客户机使用。SAM-Remote 服务器必须至少挂载一个 QFS 文件系统才能启动 Oracle HSM 进程。

要配置 SAM-Remote 服务器，请执行以下任务：

- 在 SAM-Remote 服务器的 **mcf** 文件中定义远程共享归档设备
- 创建 **samremote** 服务器配置文件

在 SAM-Remote 服务器的 mcf 文件中定义远程共享归档设备

1. 以 *root* 用户身份登录到 SAM-Remote 服务器主机。

   在示例中，服务器名为 *server1*：

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

2. 在服务器上，在文本编辑器中打开 */etc/opt/SUNWsamfs/mcf* 文件，并向下滚动到归档设备定义。

   在示例中，使用 *vi* 编辑器。该文件定义一个 Oracle HSM 归档文件系统 *fs600* 和一个包含四个驱动器的磁带库 *rb800*。请注意示例中包括实际文件中可能没有的说明标题，且会缩写较长的设备路径：

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
#================================================================================
# Oracle HSM archiving file system fs600
# Equipment               Equipment Equipment Family  Device Additional
# Identifier              Ordinal   Type      Set     State  Parameters
#-------------------------  --------- --------- ------  ------ ----------
```

```
fs600                     600       ms        fs600   on
 /dev/dsk/c9t60...F4d0s7    610       md        fs600   on
 /dev/dsk/c9t60...81d0s7    611       md        fs600   on
#-------------------------------------------------------------------------
# Local tape archive rb800
# Equipment                Equipment Equipment Family  Device Additional
# Identifier               Ordinal   Type      Set     State  Parameters
#------------------------  --------- --------- ------  ------ ----------
/dev/scsi/changer/c1t0d5   800       rb        rb800   on
 /dev/rmt/0cbn              801       tp        rb800   on
 /dev/rmt/1cbn              802       tp        rb800   on
 /dev/rmt/2cbn              803       tp        rb800   on
 /dev/rmt/3cbn              804       tp        rb800   on
```

3. 在归档设备定义的结尾，针对将使磁带资源可供客户机使用的服务器添加一个条目。在 *Equipment Identifier* 字段中输入 SAM-Remote 服务器配置文件的路径 */etc/opt/SUNWsamfs/samremote*，并分配设备序号。

   在示例中，添加一些标题作为注释并为服务器 *samremote* 分配设备序号 *500*：

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
#-------------------------------------------------------------------------
# Server samremote shares tape hardware and media with clients
# Equipment                  Equipment Equipment Family  Device Additional
# Identifier                 Ordinal   Type      Set     State  Parameters
#--------------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/samremote  500
```

4. 在新条目的 *Equipment Type* 字段中，输入 *ss* 表示 SAM-Remote 服务器设备。

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
#-------------------------------------------------------------------------
# Server samremote shares tape hardware and media with clients
# Equipment                  Equipment Equipment Family  Device Additional
# Identifier                 Ordinal   Type      Set     State  Parameters
#--------------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/samremote  500       ss
```

5. 分配一个在所有主机和服务器中唯一的 *Family Set* 名称，并将设备设置为 *on*。

   在示例中，为新设备分配系列集名称 *ss500*：

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

```
...
#========================================================================
# Server samremote shares tape hardware and media with clients
# Equipment                   Equipment Equipment Family  Device Additional
# Identifier                  Ordinal   Type      Set     State  Parameters
#--------------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/samremote  500       ss        ss500   on
```

6. 如果您打算配置十个以上的 SAM-Remote 客户机，请为包含一至十个客户机的每个后续组添加另一个服务器设备（类型 *ss*）。
7. 保存文件并关闭编辑器。

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
/etc/opt/SUNWsamfs/samremote  500       ss        ss500   on
:wq
[server1]root@solaris:~#
```

8. 接下来，创建 *samremote* 服务器配置文件。

创建 samremote 服务器配置文件

SAM-Remote 服务器配置文件用于定义每一个客户机将要使用的磁盘缓冲区特征和介质。对于需要配置的每个服务器，请执行如下操作：

1. 以 *root* 用户身份登录到 SAM-Remote 服务器主机。

   在示例中，服务器名为 *server1*：

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

2. 在服务器上，在文本编辑器中创建 */etc/opt/SUNWsamfs/samremote* 文件。

   在示例中，用 *vi* 编辑器创建文件。首先在文件中记录一些描述性注释，并以井号 (*#*) 表示：

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/samremote
# Server Configuration File:
# Defines the disk buffer and media that is available to each client.
```

3. 开始一个新行并在第一列中输入客户机的主机名、IP 地址或全限定域名，以开始第一个客户机条目。

   客户机标识符行必须以非空格字符开头。在示例中，使用主机名 *client1* 来标识客户机：

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/samremote
# Server Configuration File:
# Defines the disk buffer and media that is available to each client.
client1
```

4. 开始标识将与客户机共享的介质。以 *indent media* 形式开始一个新行，其中 *indent* 是一个或多个空格，*media* 是 SAM-Remote 关键字：

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/samremote
# Server Configuration File:
# Defines the disk buffer and media that is available to each client.
client1
        media
```

5. 以 *indent equipment-number media-type VSNs* 形式的新行标识每个介质类型和源，其中：
   - *indent* 是一个或多个空格。
   - *equipment-number* 是在 *mcf* 文件中标识归档存储设备的设备序号。
   - *media-type* 是该设备使用的磁带介质的介质标识符（有关 Oracle HSM 介质类型的完整列表，请参见附录 A, 设备类型词汇表）。
   - *VSNs* 是一个或多个卷序列号的空格分隔列表，这些序列号是最多包含 31 个字符的字母数字字符串。

   在示例中，标识一个共享介质源：位于设备序号为 *800* 的磁带库中的一系列磁带卷（类型为 *tp*）。可用卷以括号中的正则表达式指定：表达式 *VOL0[0-1][0-9]* 将 *client1* 限制为卷 *VOL000-VOL019*：

```
client1
       media
         800 tp (VOL0[0-1][0-9])
```

   请注意，每行只能指定一种介质类型。因此，如果库要支持多种介质类型，您应在新条目中指定每种类型：

```
      media
         800 ti VOL500 VOL501
         800 li (VOL0[0-1][0-9])
```

6. 标识好将与客户机共享的介质后，请输入 SAM-Remote 关键字 *endmedia* 来关闭该列表。

   在示例中，*client1* 现已完全配置好：

```
client1
        media
          800 tp (VOL0[0-1][0-9])
        endmedia
```

7. 如果您需要配置其他客户机，请现在进行操作。为每台客户机添加新的客户机配置记录，最多可以添加十 (10) 条记录。然后保存文件并关闭编辑器。

   为防止卷争用以及可能的数据丢失，请确保客户机不共享相同的可移除介质卷。

   在示例中，配置另一台客户机 *client2*。第二台客户机可访问与 *client1* 相同的磁带库（设备序号为 *800*）中的一系列磁带卷。但配置中的正则表达式指定不同的一组卷：*VOL020-VOL039*。

```
# Server Configuration File:
# Defines the disk buffer and media that is available to each client.
client1
        media
          800 tp (VOL0[0-1][0-9])
        endmedia
client2
        media
          800 tp (VOL02-3][0-9])
        endmedia
:wq
[server1]root@solaris:~#
```

8. 接下来，配置 SAM-Remote 客户机。

配置 SAM-Remote 客户机

针对每台 SAM-Remote 客户机，请执行以下任务：

- 在 SAM-Remote 客户机的 MCF 文件中定义远程归档设备
- 创建 SAM-Remote 客户机配置文件
- 在 SAM-Remote 客户机的 MCF 文件中定义远程归档设备
- 创建 SAM-Remote 客户机配置文件
- 在 SAM-Remote 客户机上配置 **archiver.cmd** 文件

在 SAM-Remote 客户机的 MCF 文件中定义远程归档设备

1. 以 *root* 用户身份登录到 SAM-Remote 客户机主机。

   在示例中，SAM-Remote 客户机名为 *client1*：

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

2. 在客户机上，在文本编辑器中打开 */etc/opt/SUNWsamfs/mcf* 文件，并向下滚动到归档设备定义。

   在示例中，使用 *vi* 编辑器。该文件定义一个 Oracle HSM 归档文件系统 *fs100*。本地副本存储在磁盘归档 *DISKVOL1*（一个本地 ZFS 文件系统）中。请注意，示例中包括实际文件中可能没有的说明标题，且会缩写较长的设备路径。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Client's /etc/opt/SUNWsamfs/mcf file
#================================================================
# Oracle HSM archiving file system "fs100"
# Equipment               Equipment Equipment Family  Device Additional
# Identifier              Ordinal   Type      Set     State  Parameters
#-----------------------  --------- --------- ------  ------ ----------
fs100                     100       ms        fs100   on
 /dev/dsk/c10t60...7Bd0s7   110       md        fs100   on
 /dev/dsk/c10t60...48d0s7   111       md        fs100   on
#================================================================
# Disk archive "/diskvols/DISKVOL1" stores local archive copies
```

3. 在归档设备定义的结尾，添加一个条目，指明服务器将允许客户机使用的设备。在 *Equipment Identifier* 字段中输入 SAM-Remote 服务器配置文件的路径，并分配设备序号。

   在本示例中，我们指定客户机配置 */etc/opt/SUNWsamfs/sc400*，并为客户机分配设备序号 *400*。再添加一些标题作为注释：

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
# Disk archive "/diskvols/DISKVOL1" stores local archive copies
#
#================================================================
# Client "sc400" accesses tape resources on server "samremote" (ss500)
# Equipment               Equipment Equipment Family  Device Additional
# Identifier              Ordinal   Type      Set     State  Parameters
#-----------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/sc400   400
```

4. 在新条目的 *Equipment Type* 字段中，输入 *sc* 表示 SAM-Remote 客户机设备。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
```

```
#========================================================================
# Client "sc400" accesses tape resources on server "samremote" (ss500)
# Equipment               Equipment Equipment Family  Device Additional
# Identifier              Ordinal   Type      Set     State  Parameters
#-----------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/ss500   400       sc
```

5. 分配一个在所有主机和服务器中唯一的 *Family Set* 名称，并将设备设置为 *on*。

   在示例中，为新设备分配系列集名称 *ss500*。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
#========================================================================
# Client "sc400" accesses tape resources on server "samremote" (ss500)
# Equipment               Equipment Equipment Family  Device Additional
# Identifier              Ordinal   Type      Set     State  Parameters
#-----------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/ss500   400       sc        ss500   on
```

6. 针对 SAM-Remote 服务器设为可用的每个磁带机，在 SAM-Remote 客户机 *sc* 设备中添加一个 SAM-Remote 伪设备。在 *Equipment Identifier* 字段中，添加一个 */dev/samrd/rddevice-number* 形式的条目，其中 *device-number* 是一个整数。

   在示例中，为两个伪设备 */dev/samrd/rd 0* 和 */dev/samrd/rd 1* 开始条目：

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
#========================================================================
# Client "sc400" accesses tape resources on server "samremote" (ss500)
# Equipment               Equipment Equipment Family  Device Additional
# Identifier              Ordinal   Type      Set     State  Parameters
#-----------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/sc400   400       sc        sc400   on
 /dev/samrd/rd0
 /dev/samrd/rd1
```

7. 在每个伪设备的 *Equipment Ordinal* 字段中，输入您分配给 *sc* 设备的范围中的数字。

   在示例中，将设备序号 *410* 分配给 */dev/samrd/rd0*，将设备序号 *420* 分配给 */dev/samrd/rd1*：

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
```

```
...
#==============================================================================
# Client "sc400" accesses tape resources on server "samremote" (ss500)
# Equipment                Equipment Equipment Family  Device Additional
# Identifier               Ordinal   Type      Set     State  Parameters
#------------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/ss500   400       sc        ss500   on
 /dev/samrd/rd0             410
 /dev/samrd/rd1             420
```

8. 在每个 SAM-Remote 伪设备的 *Equipment Type* 字段中，输入 *rd* 表示 SAM-Remote 伪设备的设备类型。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
#==============================================================================
# Client "sc400" accesses tape resources on server "samremote" (ss500)
# Equipment                Equipment Equipment Family  Device Additional
# Identifier               Ordinal   Type      Set     State  Parameters
#------------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/ss500   400       sc        ss500   on
 /dev/samrd/rd0             410       rd
 /dev/samrd/rd1             420       rd
```

9. 在每个伪设备的 *Family Set* 字段中，输入 *sc* 设备的系列集名称。

   在示例中，使用系列集名称 *ss500*：

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
#==============================================================================
# Client "sc400" accesses tape resources on server "samremote" (ss500)
# Equipment                Equipment Equipment Family  Device Additional
# Identifier               Ordinal   Type      Set     State  Parameters
#------------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/ss500   400       sc        ss500   on
 /dev/samrd/rd0             410       rd        ss500
 /dev/samrd/rd1             420       rd        ss500
```

10. 在每个伪设备的 *Device State* 字段中，输入 *on*。然后保存文件并关闭编辑器。

    在示例中，将设备序号 *410* 分配给 */dev/samrd/rd 0*，将设备序号 *420* 分配给 */dev/samrd/rd 1*：

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
```

```
...
#========================================================================
# Client "sc400" accesses tape resources on server "samremote" (ss500)
# Equipment                Equipment Equipment Family  Device Additional
# Identifier               Ordinal   Type      Set     State  Parameters
#------------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/ss500   400       sc        ss500   on
 /dev/samrd/rd0             410       rd        ss500   on
 /dev/samrd/rd1             420       rd        ss500   on
:wq
[client1]root@solaris:~#
```

11. 接下来，创建 SAM-Remote 客户机配置文件。

创建 SAM-Remote 客户机配置文件

对于每个SAM-Remote 客户机，请执行如下操作：

1. 以 *root* 用户身份登录到 SAM-Remote 客户机主机。

   在示例中，SAM-Remote 客户机名为 *client1*：

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

2. 在客户机上，在文本编辑器中创建 */etc/opt/SUNWsamfs/family-set-name* 文件，其中 *family-set-name* 是在 *mcf* 文件中使用的远程设备的系列集名称。

   在示例中，使用 *vi* 编辑器创建文件，并为其指定系列集名称 *ss500*。我们还在文件中记录一些以井号 (*#*) 开头的描述性注释：

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/sc400
# Client's SAM-Remote client configuration file: /opt/SUNWsamfs/sc400
# This file identifies the host of the SAM-Remote server.
```

3. 开始一个新行并在第一列中输入服务器的主机名、IP 地址或全限定域名，来为服务器添加一个条目。然后保存文件并关闭编辑器。

   该行必须以非空格字符开头。在示例中，使用主机名 *server1* 标识服务器：

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/samremote
# Client's SAM-Remote server configuration file: /opt/SUNWsamfs/sc400
# This file identifies the host of the SAM-Remote server.
server1
:wq
[client1]root@solaris:~#
```

4. 接下来，在 SAM-Remote 客户机上配置 *archiver.cmd* 文件。

在 SAM-Remote 客户机上配置 archiver.cmd 文件

1. 以 *root* 用户身份登录到 SAM-Remote 客户机主机。

   在示例中，SAM-Remote 客户机名为 *client1*：

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

2. 在文本编辑器中打开 */etc/opt/SUNWsamfs/archiver.cmd* 文件，向下滚动到副本参数指令（从关键字 *params* 开始，到关键字 *endparams* 结束）。

   在示例中，用 *vi* 编辑器打开文件：

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
#------------------------------------------------------------------------
# Copy Parameter Directives
params
allsets -sort path -offline_copy direct
allfiles.1 -startage 10m -startsize 500M -drives 10
allfiles.2 -startage 24h -startsize 20G -drives 2 -reserve set
endparams
```

3. 检查将在远程介质上归档的所有归档集的副本参数。如果其中任何参数包括 *-tapenonstop* 和/或 *-offline_copy direct* 指令，请立即删除这些指令。

   在示例中，*all* 参数为所有副本指定了 *-offline_copy direct* 指令。因此，为想要发送到远程介质的副本 *allfiles.3* 指定 *-offline_copy none* 来覆盖该指令：

```
#------------------------------------------------------------------------
# Copy Parameter Directives
# Copy Parameter Directives
params
allsets -sort path -offline_copy direct
allfiles.1 -startage 10m -startsize 500M -drives 10
allfiles.2 -startage 24h -startsize 20G -drives 2 -reserve set offline_copy none
endparams
```

4. 向下滚动到 VSN 指令（从 SAM-Remote 关键字 *vsns* 开始，到关键字 *endvsns* 结束）。

   在示例中，使用 *vi* 编辑器。当前已分配介质的唯一副本 *allfiles.1* 将使用本地磁盘归档卷 *qfs200* 创建：

```
...
endparams
#------------------------------------------------------------------------
# VSN Directives
vsns
allfiles.1 dk qfs200
endvsns
```

5. 将归档副本分配给在服务器的 */etc/opt/SUNWsamfs/samremote* 文件中为该客户机指定的远程介质。然后保存文件并关闭编辑器。

   在示例中，将配置 *client1*。副本 *allfiles.2* 将使用 *VOL000-VOL019* 范围内的远程磁带卷（在 *samremote* 服务器配置文件中指定）创建：

```
...
endparams
#------------------------------------------------------------------------
# VSN Directives
vsns
allfiles.1 dk qfs200
allfiles.2 tp VOL0[0-1][0-9]
endvsns
:wq
[client1]root@solaris:~#
```

6. 接下来，在 SAM-Remote 服务器上验证归档配置。

在 SAM-Remote 服务器上验证归档配置

1. 以 *root* 用户身份登录到 SAM-Remote 服务器主机。

   在示例中，SAM-Remote 服务器名为 *server1*：

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

2. 在服务器上启动 Oracle HSM 进程。使用命令 *samd start*：

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

3. 在服务器主机上，检查共享设备服务器的状态。使用命令 *samcmd s*。

   在示例中，设备序号为 *500* 的 SAM-Remote 服务器设备（类型 *ss*）处于 *on* 状态且在正常运行：

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

```
Device status samcmd     6.0  11:20:34 Feb 20 2015
samcmd on server1
ty    eq  state   device_name                    fs     status
rb    800 on      /dev/scsi/changer/c1t0d5       800    m--------r
tp    801 on      /dev/rmt/0cbn                  800    ---------p
  empty
tp    802 on      /dev/rmt/1cbn                  800    ---------p
  empty
tp    803 on      /dev/rmt/2cbn                  800    ---------p
  empty
tp    804 on      /dev/rmt/3cbn                  800    ---------p
  empty
ss    500 on      /etc/opt/SUNWsamfs/samremote     ss500  -------o-r
[server1]root@solaris:~#
```

4. 如果共享设备服务器未处于 *on* 状态，请确保已在服务器主机的 */etc/opt/SUNWsamfs/mcf* 文件中正确对其进行了定义。确保 */etc/opt/SUNWsamfs/samremote* 文件是正确的并处于正确的位置。

   请参见过程“在 SAM-Remote 服务器的 **mcf** 文件中定义远程共享归档设备”和“创建 **samremote** 服务器配置文件”。

5. 在服务器上，检查 SAM-Remote 客户机的连接状态。使用命令 *samcmd R*。

   在示例中，*client1* 和 *client2* 都处于 *0005* 状态，因此为 *connected* 状态（*0004* 状态表示无连接）：

```
[server1]root@solaris:~# samcmd R
Remote server eq: 500 addr: 00003858 samcmd 6.0  11:20:44 Feb 20 2015
samcmd on server1
message:
Client IPv4: client1 192.10.10.3 port - 5000
 client index - 0 port - 31842 flags - 0005 connected
Client IPv4: client2 10.1.229.97 port - 5000
 client index - 1 port - 32848 flags - 0005 connected
[server1]root@solaris:~#
```

6. 如果共享设备客户机未连接（*0004* 状态），请检查网络连接。确保服务器和客户机可以解析彼此的主机名和地址。确保服务器和客户机可以相互连接。

   在示例中，将 *ssh* 与 *getent* 和 *ping* 命令配合使用，以检查 SAM-Remote 配置中的每台主机与每台其他主机之间的连接：

```
[server1]root@solaris:~# getent hosts client1
192.10.10.3  client1
[server1]root@solaris:~# getent hosts 192.10.10.3
```

```
192.10.10.3  client1
[server1]root@solaris:~# ping 192.10.10.3
192.10.10.31 is alive
[server1]root@solaris:~# getent hosts client2
10.1.229.97 client2
[server1]root@solaris:~# getent hosts 10.1.229.97
10.1.229.97 client2
[server1]root@solaris:~# ping 10.1.229.97
192.10.10.31 is alive
[server1]root@solaris:~# ssh root@client1
Password: ...
[client1]root@solaris:~# getent hosts server1
192.10.201.12  server1
...
[client1]root@solaris:~# exit
[server1]root@solaris:~# ssh root@client2
Password: ...
[client2]root@solaris:~# getent hosts server1
192.10.201.12  server1
...
[client2]root@solaris:~# exit
[server1]root@solaris:~#
```

7. 如果共享设备客户机未连接（*0004* 状态），请确保它已在客户机主机的 */etc/opt/SUNWsamfs/mcf* 文件中正确定义。请确保服务器主机已在 */etc/opt/SUNWsamfs/family-set-name* 文件中正确标识，且该文件处于客户机主机中的正确位置。然后，确保客户机主机已在服务器主机上的 */etc/opt/SUNWsamfs/samremote* 文件中正确标识。

   请参见过程“在 SAM-Remote 客户机的 MCF 文件中定义远程归档设备”和“创建 SAM-Remote 客户机配置文件”。
8. 在客户机上，请确保服务器主机已在 */etc/opt/SUNWsamfs/family-set-name* 文件中正确标识，且该文件处于客户机主机中的正确位置。

   请参见过程“创建 SAM-Remote 客户机配置文件”。
9. 如果共享设备客户机未连接（*0004* 状态），且客户端配置文件没有问题，请检查服务器。确保客户机主机已在 */etc/opt/SUNWsamfs/samremote* 文件中正确标识。

   请参见过程“创建 **samremote** 服务器配置文件”。
10. 在服务器上，确保每台客户机可以访问共享磁带库的目录并查看可用的卷。使用命令 *samcmd v equipment-number*，其中 *equipment-number* 是客户机的 *mcf* 文件分配给 SAM-Remote 客户机设备的设备序号。

在示例中，检查 *client1*，显示 *400* 是 SAM-Remote 客户机设备 */etc/opt/SUNWsamfs/sc400* 的设备序号。输出正确列出了 *client1* 可以访问的卷（*VOL000* 到 *VOL019*）：

```
[server1]root@solaris:~# samcmd v 400
Robot catalog samcmd     6.0  12:20:40 Feb 20 2015
samcmd on server1
Robot VSN catalog by slot       : eq 400
slot     access time  count use flags         ty vsn
   3     none         0     0%  -il-o-b-----  li VOL000
   7     none         0     0%  -il-o-b-----  li VOL001
...
  24     none         0     0%  -il-o-b-----  li VOL019
[server1]root@solaris:~#
```

11. 如果共享设备客户机无法查看正确的卷，请检查主机文件。在服务器主机上，确保分配的卷已在 */etc/opt/SUNWsamfs/samremote* 文件中正确标识。在客户机主机上，确保 */etc/opt/SUNWsamfs/family-set-name* 文件已正确标识服务器主机。

    请参见过程“创建 **samremote** 服务器配置文件”和“创建 SAM-Remote 客户机配置文件”。
12. 接下来，在每台 SAM-Remote 客户机上验证归档配置。

在每台 SAM-Remote 客户机上验证归档配置

对于每个SAM-Remote 客户机，请执行如下操作：

1. 以 *root* 用户身份登录到 SAM-Remote 客户机主机。

   在示例中，SAM-Remote 客户机名为 *client1*：

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

2. 在客户机主机上启动 Oracle HSM 进程。使用命令 *samd start*：

```
[client1]root@solaris:~# samd start
[client1]root@solaris:~#
```

3. 在客户机主机上，检查共享设备客户机的状态。使用命令 *samcmd s*。

   在示例中，设备序号为 *400* 的 SAM-Remote 客户机设备（类型 *sc*）处于 *on* 状态且在正常运行：

```
[client1]root@solaris:~# samcmd s
Device status samcmd     6.0  12:20:49 Feb 20 2015
```

```
samcmd on client1
ty    eq  state   device_name                    fs     status
sc    400 on      /etc/opt/SUNWsamfs/sc400       sc400  -------o-r
```

4. 如果共享设备客户机未处于 *on* 状态，请确保 *sc* 设备已正确定义。在客户机主机上，检查 */etc/opt/SUNWsamfs/mcf* 文件，确保 */etc/opt/SUNWsamfs/family-set-name* 文件正确且处于正确位置。

   请参见过程“在 SAM-Remote 客户机的 MCF 文件中定义远程归档设备”和“创建 SAM-Remote 客户机配置文件”。
5. 在客户机主机上，确认 */etc/opt/SUNWsamfs/archiver.cmd* 文件为远程介质指定了正确的卷序列号。使用命令 *archiver -A* 列出文件。

   在示例中，将配置 *client1*。系统将使用 *VOL000-VOL019* 范围内的一个远程磁带卷（在 *samremote* 服务器配置文件中指定）创建副本 *allfiles.2*：

```
[client1]root@solaris:~# archiver -A
Reading '/etc/opt/SUNWsamfs/archiver.cmd'.
1: # archiver.cmd
2: #---------------------------------------------------------------------
3: # Global Directives
4: archivemeta = off
5: examine = noscan
...
30: #---------------------------------------------------------------------
31: # VSN Directives
32: vsns
33: allfiles.1 dk qfs200
34: allfiles.2 tp VOL0[0-1][0-9]
36: endvsns
[client1]root@solaris:~#
```

6. 如果您在 *archiver.cmd* 文件中发现了任何差异，请先更正问题再继续。
7. 如果您想配置回收，请参见为 SAM-Remote 配置回收。

为 SAM-Remote 配置回收

配置 SAM-Remote 时，必须确保一台主机上的回收不会销毁另一台主机上的有效数据。 您在 SAM-Remote 服务器上配置的任何回收指令必须仅回收该服务器用于其自己的归档集的介质。该服务器不得尝试回收其已供 SAM-Remote 客户机使用的介质卷。同样，您在 SAM-Remote 客户机上配置的任何回收指令必须仅回收存放归档的客户机数据的介质（在本地或在服务器设为可用的指定卷中）。

尝试在 SAM-Remote 环境中使用回收过程之前，您应全面了解回收过程。因此，请阅读“回收” 以及 *sam-recycler*、*archiver.cmd*、*recycler.cmd* 和 *recycler.sh* 手册页。

然后在熟悉回收过程后执行以下任务：

- 在 SAM-Remote 服务器上配置回收
- 在 SAM-Remote 客户机上配置回收。

在 SAM-Remote 服务器上配置回收

如果您需要为 SAM-Remote 服务器托管的文件系统配置回收，请执行如下操作：

1. 以 *root* 用户身份登录到 SAM-Remote 服务器。

   在示例中，SAM-Remote 服务器名为 *server1*：

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

2. 在文本编辑器中打开 */etc/opt/SUNWsamfs/archiver.cmd* 文件。向下滚动到 *params* 部分。

   在示例中，用 *vi* 编辑器打开文件：

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
#------------------------------------------------------------------------
# Copy Parameter Directives
params
allsets -sort path -offline_copy direct
allfiles.1 -startage 10m -startsize 500M -drives 10
allfiles.2 -startage 24h -startsize 20G -drives 2 -reserve set
endparams
```

3. 以 *archive-set directive-list* 形式按归档集输入您的回收程序指令，其中 archive-set 是归档集之一，*directive-list* 是指令名称/值对的空格分隔列表（有关回收指令的完整列表，请参见 *archiver.cmd* 手册页）。

   如果使用 SAM-Remote，必须在 *archiver.cmd* 文件的 *params* 部分配置按归档集回收。您无法指定按库回收。

   在示例中，我们为归档集 *allfiles.1* 和 *allfiles.2* 添加回收指令。仅当至少有 90% 的卷容量可以恢复时，*-recycle_mingain 90* 指令才会回收卷。当可移除介质的已用容量达到 60% 时，*-recycle_hwm 60* 指令开始回收。*-recycle_vsncount 1* 一次只能调度一个要回收的可移除介质卷：

```
#------------------------------------------------------------------------
```

```
#  Copy Parameters Directives
params
allsetsallfiles. -sort path -offline_copy direct
allfiles.1 -startage 10m -startsize 500M -drives 10
allfiles.1 -recycle_mingain 90
allfiles.2 -startage 24h -startsize 20G -drives 2 -reserve set offline_copy none
allfiles.2 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
endparams
```

请注意，在 SAM-Remote 服务器上定义的回收指令仅适用于该服务器用于其自己的归档集的归档卷。该服务器的回收指令不适用于可从客户机访问的卷。

在本示例中，该服务器针对副本 *allfiles.2* 的回收指令适用于 *VSN Directives* 部分中针对服务器使用列出的磁带卷 (*VOL100-VOL199*)。该服务器的回收指令不适用于卷 *VOL000-VOL019*（该卷是为 *client1* 保留的），也不适用于卷 *VOL020-VOL039*（该卷是为 *client2* 保留的）：

```
...
endparams
#------------------------------------------------------------------------
# VSN Directives
vsns
allfiles.1 dk DISKVOL1
allfiles.2 tp VOL1[0-9][0-9]
endvsns
```

4. 保存 *archiver.cmd* 文件并关闭编辑器。

```
...
endvsns
:wq
[server1]root@solaris:~#
```

5. 在服务器上，在文本编辑器中创建 *recycler.cmd* 文件。指定回收程序日志的路径和文件名。

   在示例中，使用 *vi* 编辑器。为日志文件指定默认位置：

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
logfile = /var/adm/recycler.log
```

6. 在服务器上的 *recycler.cmd* 文件中，添加 *no-recycle media-type volumes* 形式的指令，其中 *media-type* 是在附录 A, 设备类型词汇表中指定的介质类型

之一，*volumes* 是以空格分隔的列表或正则表达式，用于指定您已分配给 SAM-Remote 客户机的每个归档存储卷的卷序列号。保存文件并关闭编辑器。

no-recyle 指令为专供客户机使用的存储资源提供了额外的保护。该指令会显式命令主机回收过程跳过指定的卷。

在本示例中，我们针对处于范围 *VOL000-VOL019* 和 *VOL020-VOL039* 内的介质类型 *tp*（磁带）卷添加 *no-recyle* 指令：

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
logfile = /var/opt/SUNWsamfs/recycler/recycler.log
no_recycle tp VOL0[0-1][0-9] VOL0[2-3][0-9]
:wq
[server1]root@solaris:~#
```

7. 现在，在 SAM-Remote 客户机上配置回收。

在 SAM-Remote 客户机上配置回收

针对每个客户机，执行如下操作：

1. 以 *root* 用户身份登录到 SAM-Remote 客户机。

   在示例中，SAM-Remote 客户机名为 *client1*：

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

2. 在客户机上，在文本编辑器中打开 */etc/opt/SUNWsamfs/archiver.cmd* 文件，向下滚动到副本 *params* 部分。

   在示例中，用 *vi* 编辑器打开文件。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
#------------------------------------------------------------------------
# Copy Parameters Directives
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 6h  -startsize 6G  -startcount 500000
allfiles.2 -startage 24h -startsize 20G -startcount 500000 -archmax 24G
endparams
#------------------------------------------------------------------------
# VSN Directives
vsns
allfiles.1 dk qfs200
allfiles.2 tp VOL0[0-1][0-9]
```

```
endvsns
```

3. 在 *archiver.cmd* 文件的 *params* 部分中，按归档集输入您的回收程序指令，格式为 *archive-set directive-list*，其中，archive-set 是其中一个归档集，*directive-list* 是指令名称/值对的空格分隔列表（有关回收指令的列表，请参见 *archiver.cmd* 手册页）。然后保存文件并关闭编辑器。

   在使用 SAM-Remote 时，必须在 *archiver.cmd* 文件的 *params* 部分中配置按归档集回收。您无法指定按库回收。

   在示例中，我们为归档集 *allfiles.1* 和 *allfiles.2* 添加回收指令。仅当至少有 90% 的卷容量可以恢复时，*-recycle_mingain 90* 指令才会回收卷。当可移除介质的已用容量达到 60% 时，*-recycle_hwm 60* 指令开始回收。*-recycle_vsncount 1* 指令一次只能调度一个要回收的可移除介质卷。

```
#------------------------------------------------------------------------
# Copy Parameters Directives
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 6h  -startsize 6G  -startcount 500000
allfiles.1 -recycle_mingain 90
allfiles.2 -startage 24h -startsize 20G -startcount 500000 -archmax 24G
allsets.2 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
endparams
```

   请注意，在客户机上定义的回收指令仅适用于该客户机用于其自己的归档集的介质。在本示例中，客户机针对副本 *allfiles.2* 的回收指令适用于服务器提供的处于范围 *VOL000-VOL019* 内的远程磁带卷。这些指令不适用于处于范围 *VOL020-VOL039* 内的卷（这些卷是为 client2 保留的），也不适用于处于范围 *VOL100-VOL119* 内的卷（这些卷是为服务器保留的）：

```
...
endparams
#------------------------------------------------------------------------
# VSN Directives
vsns
allfiles.1 dk qfs200
allfiles.2 tp VOL0[0-1][0-9]
endvsns
:wq
[client1]root@solaris:~#
```

4. 保存 *archiver.cmd* 文件并关闭编辑器。

```
...
endvsns
:wq
[client]root@solaris:~#
```

5. 在客户机上，在文本编辑器中创建 *recycler.cmd* 文件。指定回收程序日志的路径和文件名。然后保存文件并关闭编辑器。

   已配置服务器和客户机，使客户机无法访问服务器或 *client2* 使用的任何归档介质。因此，不需要添加 *no-recyle* 指令。

   在示例中，使用 *vi* 编辑器。为日志文件指定默认位置：

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
logfile = /var/adm/recycler.log
:wq
[client1]root@solaris:~#
```

6. 重复此过程直到完成所有 SAM-Remote 客户机的配置。
7. 输入命令 *sam-recycler -dvxn*，其中的参数具有以下作用：
   - *-d* 会显示卷选择消息，指示每个卷为何已被选择或未被选择进行回收。
   - *-v* 会列出位于标记为要回收的每个卷中且需要移动的文件。
   - *-x* 会返回错误并停止（如果列出的任何归档副本比卷的标记时间要早而不可恢复）。
   - *-n* 会阻止实际回收。回收过程会表现得好像 *archiver.cmd* 文件中的所有归档集定义都包括 *-recycle_ignore* 一样，以便您可以对回收配置进行非破坏性测试。
8. 在配置好所有 SAM-Remote 客户机和服务器之后，如果您打算使用边带数据库功能，请转至第 10 章 配置报告数据库。
9. 否则，请转至第 11 章 配置通知和日志记录。

9

第 9 章 准备高可用性解决方案

Oracle Hierarchical Storage Manager and StorageTek QFS Software 高可用性配置设计用于维持不中断的文件系统和归档服务。在高可用性解决方案中，Oracle Hierarchical Storage Manager 或 QFS 软件与 Oracle Solaris Cluster 软件、冗余硬件和冗余通信相集成。因此，如果主机系统或组件出现故障或由管理员停止服务，则 Oracle HSM 服务将自动故障转移到用户和应用程序可以访问的备用主机。高可用性配置因而可最大程度地减少由设备和系统故障导致的停机时间。

但是，高可用性配置比较复杂，必须仔细进行设计和部署以防发生无法预料的交互和可能的数据损坏。因此，本章将首先对支持的配置进行说明。对本节进行学习并选择最能满足您的可用性要求的配置。后续各节将说明如何设置所选配置。

请注意，无法在共享 Oracle Solaris Cluster 配置中混用硬件体系结构。所有节点必须使用 SPARC 体系结构、x86-64 体系结构（仅限 Solaris 11.1）或 32 位 x86 体系结构（Solaris 10 和更早版本）。

了解支持的高可用性配置

在群集多主机解决方案中，必须仔细控制文件系统、应用程序、操作系统、群集软件和存储之间的交互，以确保存储数据的完整性。为使复杂性和潜在风险降至最低，专门针对四个特定的部署要求集量身定制了支持的高可用性 Oracle HSM 配置：

- HA-QFS，高可用性 QFS 非共享的独立文件系统配置
- HA-COTC，具有高可用性元数据服务器的 QFS 共享文件系统
- HA-SAM，高可用性、归档、QFS 共享文件系统配置
- SC-RAC，Oracle RAC 的高可用性 QFS 共享文件系统配置。

HA-QFS，高可用性 QFS 非共享的独立文件系统配置

高可用性 QFS (High Availability QFS, HA-QFS) 配置可确保 QFS 非共享的独立文件系统在主机发生故障时仍然可以访问。该文件系统在双节点群集中的两个节点上都进行配置，Solaris Cluster 软件将此群集作为 *SUNW.HAStoragePlus* 类型的资源来管理。但是，在任意给定时间，只有一个节点挂载 QFS 文件系统。如果挂载文件系统的节点出现故障，则群集软件将自动启动故障转移并在剩余节点上重新挂载文件系统。

客户机通过网络文件系统 (Network File System, NFS)、高可用性 NFS (High-Availability NFS, HA-NFS) 或 SMB/CIFS 共享访问数据，活动群集节点充当文件服务器。

有关实施说明，请参见“高可用性 QFS 非共享文件系统”。

HA-COTC，具有高可用性元数据服务器的 QFS 共享文件系统

群集外部的高可用性客户机 (High Availability-Clients Outside the Cluster, HA-COTC) 配置可维持 QFS 元数据服务器的可用性，以便 QFS 文件系统客户机可以在服务器出现故障时继续访问其数据。文件系统是共享的。在由 Solaris Cluster 软件管理的双节点群集上托管 QFS 活动元数据服务器和潜在元数据服务器。*SUNW.qfs* 类型的 Oracle HSM 高可用性资源可管理群集中共享文件系统服务器的故障转移。所有客户机均托管在群集外部。群集服务器可确保元数据的可用性、签发 I/O 许可证并维持文件系统的一致性。

如果托管活动元数据服务器的节点出现故障，则 Solaris Cluster 软件将自动激活运行正常的节点上的潜在 MDS 并启动故障转移。QFS 文件系统是共享的，因此它已经挂载到新激活的元数据服务器节点上，并在客户机上保持挂载状态。客户机继续接收元数据更新和 I/O 租约，因此，文件系统可无中断地继续运行。

HA-COTC 配置必须将高性能 *ma* 文件系统与物理独立的 *mm* 元数据设备和 *mr* 数据设备配合使用。通用 *ms* 文件系统和 *md* 设备不受支持。

您可以使用标准网络文件系统 (Network File System, NFS) 或 SMB/CIFS 与不运行 Oracle HSM 的客户机共享 HA-COTC 文件系统。但不支持 HA-NFS。

有关实施说明，请参见“高可用性 QFS 共享文件系统，群集外部的客户机”。

HA-SAM，高可用性、归档、QFS 共享文件系统配置

高可用性 Oracle Hierarchical Storage Manager (HA-SAM) 配置可通过确保 QFS 元数据服务器和 Oracle Hierarchical Storage Manager 应用程序在服务器主机发生故障的情况下仍能继续运行来维持归档文件系统的可用性。该文件系统在由 Solaris Cluster 软件管理的双节点群集上托管的活动 QFS 元数据服务器和潜在 QFS 元数据服务器之间共享。类型为 *SUNW.qfs* 的 Oracle HSM 高可用性资源管理服务器的故障转移。

如果活动 Oracle HSM 元数据服务器节点发生故障，则群集软件将自动激活潜在元数据服务器节点并启动故障转移。由于 QFS 文件系统是共享的并已挂载到所有节点上，因此仍可无中断地访问数据和元数据。

客户机通过高可用性网络文件系统 (high-availability Network File System, HA-NFS)、NFS 或 SMB/CIFS 共享访问数据，活动群集节点充当文件服务器。

有关实施说明，请参见“高可用性 Oracle HSM 共享归档文件系统”。

SC-RAC，Oracle RAC 的高可用性 QFS 共享文件系统配置

Solaris Cluster-Oracle Real Application Cluster (SC-RAC) 配置支持使用 QFS 文件系统的高可用性数据库解决方案。RAC 软件协调 I/O 请求、分配工作负荷，并且为群集节点上运行的多个 Oracle 数据库实例维护一组一致的数据库文件。在 SC-RAC 配置

中，Oracle 数据库、Oracle Real Application Cluster (RAC) 和 QFS 软件在群集中的两个或更多节点上运行。Solaris Cluster 软件将群集作为 *SUNW.qfs* 类型的资源进行管理。一个节点配置为 QFS 共享文件系统的元数据服务器 (metadata server, MDS)。剩余节点配置为将文件系统共享为客户机的潜在元数据服务器。如果活动元数据服务器节点发生故障，则 Solaris Cluster 软件将自动激活运行正常的节点上的潜在元数据服务器并启动故障转移。由于 QFS 文件系统是共享的并已挂载到所有节点上，因此仍可无中断地访问数据。

高可用性 QFS 非共享文件系统

要配置高可用性 QFS (high-availability QFS, HA-QFS) 文件系统，需在作为 *SUNW.HAStoragePlus* 类型的资源进行管理的双节点 Solaris Cluster 中设置两个相同的主机。然后，在两个节点上配置 QFS 非共享文件系统。在任意给定时间，只有一个节点挂载文件系统。但是，如果一个节点发生故障，则群集软件将自动启动故障转移并将文件系统重新挂载在留存下来的节点上。

要设置高可用性 QFS (high-availability QFS, HA-QFS) 文件系统，请执行如下操作：

- 在两个群集节点上创建非共享 QFS 文件系统
- 配置高可用性 QFS 文件系统
- 如果需要，配置高可用性网络文件系统 (High-Availability Network File System, HA-NFS) 共享。

  Oracle Solaris Cluster 联机文档库包含的《*Oracle Solaris Cluster Data Service for Network File System (NFS) Guide*》中提供了设置 HA-NFS 的详细步骤。

在两个群集节点上创建非共享 QFS 文件系统

1. 以 *root* 用户身份登录到其中一个群集节点。

   在示例中，主机为 *qfs1mds-node1* 和 *qfs1mds-node2*。登录主机 *qfs1mds-node1*：

```
[qfs1mds-node1]root@solaris:~#
```

2. 在主机上配置所需的 QFS 文件系统，但不进行挂载。

   使用“配置通用 **ms** 文件系统”或“配置高性能 **ma** 文件系统”中的说明配置文件系统。HA-QFS 配置不支持 QFS 共享文件系统。
3. 以 *root* 用户身份登录其余群集节点。

   在示例中，使用 *ssh* 登录到 *qfs1mds-node2* 主机：

```
[qfs1mds-node1]root@solaris:~# ssh root@qfs1mds-node2
Password:
[qfs1mds-node2]root@solaris:~#
```

4. 在第二个节点上配置相同的 QFS 文件系统。
5. 接下来，配置高可用性 QFS 文件系统。

配置高可用性 QFS 文件系统

执行如下操作：

1. 以 *root* 用户身份登录到其中一个群集节点。

   在示例中，主机为 *qfs1mds-node1* 和 *qfs1mds-node2*。登录主机 *qfs1mds-node1*：

```
[qfs1mds-node1]root@solaris:~#
```

2. 为 Solaris Cluster 软件定义 *SUNW.HAStoragePlus* 资源类型（如果尚未执行此操作）。使用命令 *clresourcetype register SUNW.HAStoragePlus*。

   HAStoragePlus 是用于定义和管理磁盘设备组、群集文件系统和本地文件系统之间依赖性的 Solaris Cluster 资源类型。它会协调故障转移后数据服务的启动，以便所有必需组件在服务尝试重新启动时准备就绪。有关详细信息，请参见 *SUNW.HAStoragePlus* 手册页。

```
[qfs1mds-node1]root@solaris:~# clresourcetype register SUNW.HAStoragePlus
[qfs1mds-node1]root@solaris:~#
```

3. 创建 *SUNW.HAStoragePlus* 类型的新 Solaris Cluster 资源以及包含该资源的新资源组。使用命令 */usr/global/bin/clresource create -g resource-group -t SUNW.HAStoragePlus -x FilesystemMountPoints=/global/mount-point -x FilesystemCheckCommand=/bin/true QFS-resource*，其中：
   - *resource-group* 是您为文件系统资源组选择的名称。
   - *mount-point* 是挂载 QFS 文件系统的目录。
   - *QFS-resource* 是您为 *SUNW.HAStoragePlus* 资源选择的名称。

   在示例中，使用挂载点目录 */global/qfs1* 和 *SUNW.HAStoragePlus* 资源 *haqfs* 创建资源组 *qfsrg*（请注意，下面的命令在一行中输入－换行符用反斜杠字符来转义）：

```
[qfs1mds-node1]root@solaris:~# clresource create -g qfsrg -t SUNW.HAStoragePlus /
-x FilesystemMountPoints=/global/hsmqfs1/qfs1 /
-x FilesystemCheckCommand=/bin/true haqfs
[qfs1mds-node1]root@solaris:~#
```

4. 显示群集中的节点。使用命令 *clresourcegroup status*。

在示例中，QFS 文件系统主机节点为 *qfs1mds-1* 和 *qfs1mds-2*。节点 *qfs1mds-1* 处于 *Online* 状态，因此它是挂载文件系统并托管 *qfsrg* 资源组的主节点：

```
[qfs1mds-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name    Node Name    Suspended   Status
----------    ---------    ---------   ------
qfsrg         qfs1mds-1    No          Online
              qfs1mds-2    No          Offline
[qfs1mds-node1]root@solaris:~#
```

5. 通过将资源组移动到辅助节点确保资源组正确地故障转移。使用 Solaris Cluster 命令 *clresourcegroup switch -n node2 group-name*，其中 *node2* 是辅助节点的名称，*group-name* 是您为 HA-QFS 资源组选择的名称。然后，使用 *clresourcegroup status* 检查结果。

   在示例中，我们将 *haqfs* 资源组移动到 *qfs1mds-node2* 并确认资源组在指定节点上处于联机状态：

```
[qfs1mds-node1]root@solaris:~# clresourcegroup switch -n qfs1mds-node2 qfsrg
[qfs1mds-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name    Node Name    Suspended   Status
----------    ---------    ---------   ------
qfsrg         qfs1mds-1    No          Offline
              qfs1mds-2    No          Online
[qfs1mds-node1]root@solaris:~#
```

6. 将资源组移回到主节点。使用 Solaris Cluster 命令 *clresourcegroup switch -n node1 group-name*，其中 *node1* 是主节点的名称，*group-name* 是您为 HA-QFS 资源组选择的名称。然后，使用 *clresourcegroup status* 检查结果。

   在示例中，我们将 *qfsrg* 资源组成功移回到 *qfs1mds-node1*：

```
[qfs1mds-node1]root@solaris:~# clresourcegroup switch -n qfs1mds-node1 qfsrg
[qfs1mds-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
qfsrg       qfs1mds-node1   No          Online
            qfs1mds-node2   No          Offline
[qfs1mds-node1]root@solaris:~#
```

7. 如果您需要配置高可用性网络文件系统 (High-Availability Network File System, HA-NFS) 共享，请立即执行此操作。有关说明，请参见Oracle Solaris Cluster 联机文档库包含的《Oracle Solaris Cluster Data Service for Network File System (NFS) Guide》。
8. 如果计划使用边带数据库功能，请转至第 10 章 配置报告数据库。
9. 否则，请转至第 11 章 配置通知和日志记录。

高可用性 QFS 共享文件系统，群集外部的客户机

群集外部的高可用性客户机 (High Availability-Clients Outside the Cluster, HA-COTC) 配置是非归档 QFS 共享文件系统，用于在 Solaris Cluster 软件所管理的高可用性群集的节点上托管至关紧要的元数据服务器 (metadata server, MDS)。这种安排可为 QFS 元数据和文件访问租约提供故障转移保护，因此，文件系统客户机在服务器出现故障时不会丢失对其数据的访问。但是，文件系统客户机和数据设备保留在群集外部，这样 Solaris Cluster 会与 QFS 软件争夺 QFS 共享数据的控制权。

要配置 HA-COTC 文件系统，请执行以下任务：

- 在两个 HA-COTC 群集节点上创建 QFS 共享文件系统 Hosts 文件
- 在 HA-COTC 群集外的 QFS 服务器和客户机上创建本地 Hosts 文件
- 在主 HA-COTC 群集节点上配置活动 QFS 元数据服务器
- 在辅助 HA-COTC 群集节点上配置潜在 QFS 元数据服务器
- 配置 HA-COTC 元数据服务器的故障转移
- 将 HA-COTC 群集外部的主机配置为 QFS 共享文件系统客户机
- 如果需要，按"使用 NFS 和 SMB/CIFS 从多台主机访问文件系统"中所述配置网络文件系统 (Network File System, NFS) 共享。不支持高可用性 NFS (High-Availability NFS, HA-NFS)。

在两个 HA-COTC 群集节点上创建 QFS 共享文件系统 Hosts 文件

在 QFS 共享文件系统中，您必须在元数据服务器上配置 hosts 文件，以便所有主机都能访问文件系统的元数据。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. 以 *root* 用户身份登录到 HA-COTC 群集的主节点。

   在示例中，主机为 *qfs1mds-node1* 和 *qfs1mds-node2*。登录主机 *qfs1mds-node1*：

```
[qfs1mds-node1]root@solaris:~#
```

2. 显示群集配置。使用 */usr/global/bin/cluster show* 命令。找出每个 *Node Name* 的记录，然后记下 *privatehostname*、*Transport Adapter* 名称和每个网络适配器的 *ip_address* 属性。

命令的输出可能非常冗长，因此，下面的示例将使用省略号 (...) 对较长的显示进行了缩写。

在示例中，每个节点都有两个网络接口，*qfe3* 和 *hme0*：

- *hme0* 适配器具有群集用于节点间内部通信的专用网络的 IP 地址。Solaris Cluster 软件会分配与每个专用地址相对应的专用主机名。

  默认情况下，主节点和辅助节点的专用主机名分别为 *clusternode1-priv* 和 *clusternode2-priv*。
- *qfe3* 适配器具有公共 IP 地址和主机名（即 *qfs1mds-node1* 和 *qfs1mds-node2*），群集使用它们来进行数据传输。

```
[qfs1mds-node1]root@solaris:~# cluster show
...
  === Cluster Nodes ===
  Node Name:                                    qfs1mds-node1...
    privatehostname:                               clusternode1-priv...
    Transport Adapter List:                        qfe3, hme0...
    Transport Adapter:                          qfe3...
      Adapter Property(ip_address):                172.16.0.12...
    Transport Adapter:                          hme0...
      Adapter Property(ip_address):                10.0.0.129...
  Node Name:                                    qfs1mds-node2...
    privatehostname:                               clusternode2-priv...
    Transport Adapter List:                        qfe3, hme0...
      Adapter Property(ip_address):                172.16.0.13...
    Transport Adapter:                          hme0
      Adapter Property(ip_address):                10.0.0.122
```

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*，其中 *family-set-name* 为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.qfs1*。我们添加一些可选标题以显示主机表中的列，每行以井号 (#) 开头表示该行为注释：

```
[qfs1mds-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.qfs1
# /etc/opt/SUNWsamfs/hosts.qfs1
#                                          Server   On/  Additional
#Host Name     Network Interface             Ordinal  Off  Parameters
#------------  ----------------------------  -------  ---  ----------
```

4. 在表的第一列中，输入主元数据服务器节点和辅助元数据服务器节点的主机名，后跟一些空格。将每个条目放在单独一行上。

在 hosts 文件中，行即为行（记录），空格为列（字段）分隔符。在本示例中，*Host Name* 列的前两行包含值 *qfs1mds-node1* 和 *qfs1mds-node2*，它们是托管文件系统元数据服务器的群集节点的主机名：

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------  -------  ---  ----------
qfs1mds-node1
qfs1mds-node2
```

5. 在每行的第二列，开始提供主机 *Host Name* 的 *Network Interface* 信息。输入各个 HA-COTC 群集节点的 Solaris Cluster 专用主机名或专用网络地址，后跟一个逗号。

   HA-COTC 服务器节点使用专用主机名在高可用性群集内进行服务器间的通信。在示例中，使用专用主机名 *clusternode1-priv* 和 *clusternode2-priv*，这些是 Solaris Cluster 软件分配的默认名称：

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------  -------  ---  ----------
qfs1mds-node1  clusternode1-priv,
qfs1mds-node2  clusternode2-priv,
```

6. 在每行第二列中的逗号后面，输入活动元数据服务器的虚拟公共主机名，后跟空格。

   HA-COTC 服务器节点使用公共数据网络与客户机通信，所有这些客户机都位于在群集之外。由于活动元数据服务器的 IP 地址和主机名在故障转移期间会发生更改（从 *qfs1mds-node1* 更改为 *qfs1mds-node2*，反之亦然），因此，我们对二者都使用虚拟主机名 *qfs1mds*。稍后，我们将配置 Solaris Cluster 软件配置为始终将 *qfs1mds* 的请求路由到活动元数据服务器：

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------  -------  ---  ----------
qfs1mds-node1  clusternode1-priv,qfs1mds
qfs1mds-node2  clusternode2-priv,qfs1mds
```

7. 在每行的第三列中，输入服务器的序号（*1* 表示活动元数据服务器，*2* 表示潜在元数据服务器），后跟空格。

   在本示例中，只有一个元数据服务器，主节点 *qfs1mds-node1* 是活动元数据服务器，因此其序号为 *1*，辅助节点 *qfs1mds-node2* 的序号为 **2**：

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------- -------  ---  ----------
qfs1mds-node1  clusternode1-priv,qfs1mds  1
qfs1mds-node2  clusternode2-priv,qfs1mds  2
```

8. 在每行的第四列中，输入 *0*（零），后跟空格。

   第四列中的值为 *0*（零）、*-*（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 *samsharefs* 手册页）。

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------- -------  ---  ----------
qfs1mds-node1  clusternode1-priv,qfs1mds  1        0
qfs1mds-node2  clusternode2-priv,qfs1mds  2        0
```

9. 在主节点行的第五列中，输入关键字 *server*。

   server 关键字用于标识默认的活动元数据服务器：

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------- -------  ---  ----------
qfs1mds-node1  clusternode1-priv,qfs1mds  1        0    server
qfs1mds-node2  clusternode2-priv,qfs1mds  2        0
```

10. 为每个客户机主机都添加一行，并将 *Server Ordinal* 值设置为 *0*。然后保存文件并关闭编辑器。

    服务器序号为 *0* 表示该主机是一个客户机而非服务器。HA-COTC 客户机不是群集成员，因此只能通过群集的公共数据网络进行通信。它们仅具有公共 IP 地址。在示例中，使用客户机的公共 IP 地址 *172.16.0.133* 和 *172.16.0.147*（而不是主机名）添加两个客户机 *qfs1client1* 和 *qfs1client2*：

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------- -------  ---  ----------
qfs1mds-node1  clusternode1-priv,qfs1mds  1        0    server
qfs1mds-node2  clusternode2-priv,qfs1mds  2        0
qfs1client1    172.16.0.133               0        0
qfs1client2    172.16.0.147               0        0
```

```
:wq
[qfs1mds-node1]root@solaris:~#
```

11. 将全局 */etc/opt/SUNWsamfs/hosts.family-set-name* 文件的一个副本放在 QFS 潜在元数据服务器（第二个 HA-COTC 群集节点）上。
12. 现在，在 HA-COTC 群集外的 QFS 服务器和客户机上创建本地 hosts 文件。

在 HA-COTC 群集外的 QFS 服务器和客户机上创建本地 Hosts 文件

在与群集外客户机共享文件系统的高可用性配置中，您需要确保客户机仅与使用 Solaris Cluster 软件定义的公共数据网络的文件系统服务器通信。为此，需要使用专门配置的 QFS 本地 hosts 文件选择性地在客户机和服务器的多个网络接口之间路由网络流量。

每个文件系统主机通过首先查看元数据服务器上的 */etc/opt/SUNWsamfs/hosts.family-set-name* 文件来确定用于其他主机的网络接口。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。通过在不同的安排下使用各文件中的 IP 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. 以 *root* 用户身份登录到 HA-COTC 群集的主节点。

   在示例中，主机为 *qfs1mds-node1* 和 *qfs1mds-node2*。登录主机 *qfs1mds-node1*：

```
[qfs1mds-node1]root@solaris:~#
```

2. 在每个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。使用路径和文件名 */etc/opt/SUNWsamfs/hosts.family-set-name.local*，其中 *family-set-name* 是共享文件系统的设备标识符。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，我们希望活动元数据服务器和潜在元数据服务器通过专用网络互相通信，并通过公共网络与客户机通信。因此活动服务器和潜在服务器上的本地 hosts 文件 *hosts.qfs1.local* 仅列出活动服务器和潜在服务器的群集专用地址：

```
[qfs1mds-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.qfs1.local
# /etc/opt/SUNWsamfs/hosts.qfs1.local
#                                         Server   On/  Additional
#Host Name     Network Interface           Ordinal  Off  Parameters
#------------  --------------------------  -------  ---  ----------
```

```
qfs1mds-node1  clusternode1-priv              1        0    server
qfs1mds-node2  clusternode2-priv              2        0
qfs1client1    172.16.0.133                     0        0
qfs1client2    172.16.0.147                     0        0
:wq
[qfs1mds-node1]root@solaris:~# ssh root@qfs1mds-node2
Password:


[qfs1mds-node2]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.qfs1.local
# /etc/opt/SUNWsamfs/hosts.qfs1.local
#                                            Server   On/  Additional
#Host Name     Network Interface              Ordinal  Off  Parameters
#------------  -----------------------------  -------  ---  ----------
qfs1mds-node1  clusternode1-priv              1        0    server
qfs1mds-node2  clusternode2-priv              2        0
qfs1client1    172.16.0.133                     0        0
qfs1client2    172.16.0.147                     0        0
:wq
[qfs1mds-node2]root@solaris:~# exit
[qfs1mds-node1]root@solaris:~#
```

3. 使用文本编辑器在每个客户机上创建一个本地 hosts 文件。使用路径和文件名 */etc/opt/SUNWsamfs/hosts.family-set-name.local*，其中 *family-set-name* 是共享文件系统的设备标识符。请仅包括您希望客户机使用的网络的接口。然后保存文件并关闭编辑器。

   在示例中，使用 *vi* 编辑器。我们希望客户机仅与服务器通信，并且仅通过公共数据网络通信。所以文件仅包括活动元数据服务器 *qfs1mds* 的虚拟主机名。Solaris Cluster 软件将 *qfs1mds* 的请求路由至处于活动状态的任何一个服务器节点：

```
[qfs1mds-node1]root@solaris:~# ssh root@qfsclient1
Password:
[qfs1client1]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.qfs1.local
# /etc/opt/SUNWsamfs/hosts.qfs1.local
#                                            Server   On/  Additional
#Host Name     Network Interface              Ordinal  Off  Parameters
#------------  -----------------------------  -------  ---  ----------
qfs1mds        qfs1mds                        1        0    server
:wq
qfs1client1]root@solaris:~# exit
[qfs1mds-node1]root@solaris:~# ssh root@qfs1client2
Password:
[qfs1client2]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.qfs1.local
# /etc/opt/SUNWsamfs/hosts.qfs1.local
#                                            Server   On/  Additional
```

```
#Host Name     Network Interface              Ordinal  Off  Parameters
#------------  -----------------------------  -------  ---  ----------
qfs1mds        qfs1mds                        1        0    server
:wq
[qfs1client2]root@solaris:~# exit
[qfs1mds-node1]root@solaris:~#
```

4. 接下来，在主 HA-COTC 群集节点上配置活动 QFS 元数据服务器。

在主 HA-COTC 群集节点上配置活动 QFS 元数据服务器

要配置活动元数据服务器，请执行以下任务：

- 在主 HA-COTC 节点上创建高性能的 QFS 文件系统
- 从群集控制中排除数据设备
- 在主 HA-COTC 节点上挂载 QFS 文件系统。

在主 HA-COTC 节点上创建高性能的 QFS 文件系统

1. 选择将同时充当 HA-COTC 群集的主节点和 QFS 共享文件系统的活动元数据服务器的群集节点。以 *root* 用户身份登录。

   在示例中，*qfs1mds-node1* 是主节点和活动元数据服务器：

```
[qfs1mds-node1]root@solaris:~#
```

2. 选择要用于 QFS 文件系统的全局存储设备。使用 Solaris Cluster 命令 */usr/global/bin/cldevice list -v*。

   Solaris Cluster 软件向连接到群集节点的所有设备分配唯一的设备标识符 (Device Identifier, DID)。全局设备可以从群集中的所有节点进行访问，而本地设备仅可以从挂载它们的主机进行访问。全局设备在故障转移后仍然可以访问。本地设备则不然。

   在示例中，请注意 *d1*、*d2*、*d7* 和 *d8* 设备从两个节点都无法访问。所以在配置高可用性 QFS 共享文件系统时，我们从 *d3*、*d4* 和 *d5* 设备中进行选择：

```
[qfs1mds-node1]root@solaris:~# cldevice list -v
DID Device          Full Device Path
----------          ----------------
d1                  qfs1mds-node1:/dev/rdsk/c0t0d0
d2                  qfs1mds-node1:/dev/rdsk/c0t6d0
d3                  qfs1mds-node1:/dev/rdsk/c1t1d0
d3                  qfs1mds-node2:/dev/rdsk/c1t1d0
d4                  qfs1mds-node1:/dev/rdsk/c1t2d0
d4                  qfs1mds-node2:/dev/rdsk/c1t2d0
```

```
d5                  qfs1mds-node1:/dev/rdsk/c1t3d0
d5                  qfs1mds-node2:/dev/rdsk/c1t3d0
d6                  qfs1mds-node2:/dev/rdsk/c0t0d0
d7                  qfs1mds-node2:/dev/rdsk/c0t1d0
```

3. 在选定的主节点上，创建使用 *md* 或 *mr* 数据服务的高性能 *ma* 文件系统。在文本编辑器中，打开 */etc/opt/SUNWsamfs/mcf* 文件。

   在示例中，我们配置文件系统 *qfs1*。我们将设备 *d3* 配置为元数据设备（设备类型 *mm*），并将 *d4* 和 *d5* 用作数据设备（设备类型 *mr*）：

```
[qfs1mds-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family   Device   Additional
# Identifier          Ordinal    Type       Set      State    Parameters
#-----------------   ---------  ---------  -------  ------   ----------------
qfs1                  100        ma         qfs1     -
 /dev/did/dsk/d3s0    101        mm         qfs1     -
 /dev/did/dsk/d4s0    102        mr         qfs1     -
 /dev/did/dsk/d5s1    103        mr         qfs1     -
```

4. 在 */etc/opt/SUNWsamfs/mcf* 文件的文件系统条目的 *Additional Parameters* 列中输入 *shared* 参数。保存文件。

```
[qfs1mds-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family   Device   Additional
# Identifier          Ordinal    Type       Set      State    Parameters
#-----------------   ---------  ---------  -------  ------   ----------------
qfs1                  100        ma         qfs1     -        shared
 /dev/did/dsk/d3s0     101        mm         qfs1     -
 /dev/did/dsk/d4s0     102        mr         qfs1     -
 /dev/did/dsk/d5s1     103        mr         qfs1     -
:wq
[qfs1mds-node1]root@solaris:~#
```

5. 检查 *mcf* 文件中是否存在错误。使用命令 */opt/SUNWsamfs/sbin/sam-fsd* 并更正发现的任何错误。

   sam-fsd 命令会读取 Oracle HSM 配置文件并初始化文件系统。如果遇到错误，该命令将停止。在示例中，检查主机 *qfs1mds-node1* 上的 *mcf* 文件：

```
[qfs1mds-node1]root@solaris:~# sam-fsd
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
```

```
Would start sam-amld()
[qfs1mds-node1]root@solaris:~#
```

6. 创建文件系统。使用命令 */opt/SUNWsamfs/sbin/sammkfs -S family-set-name*，其中 *family-set-name* 是文件系统的设备标识符。

   sammkfs 命令读取 *qfs1mds-node1* 主节点上的 *hosts.family-set-name* 和 *mcf* 文件，并创建具有指定属性的共享文件系统。

```
[qfs1mds-node1]root@solaris:~# sammkfs -S qfs1
Building 'qfs1' will destroy the contents of devices:
  ...
Do you wish to continue? [y/N]yes ...
[qfs1mds-node1]root@solaris:~#
```

7. 现在，从群集控制中排除数据设备。

从群集控制中排除数据设备

默认情况下，Solaris Cluster 软件隔离磁盘设备以专供群集使用。但是在 HA-COTC 配置中，仅元数据 (*mm*) 设备属于群集的一部分。数据 (*mr*) 设备与群集外的文件系统客户机共享，并且直接连接到客户机主机。因此您必须将数据 (*mr*) 设备置于群集软件的控制之外。这可以通过以下两种方法之一实现：

- 针对 HA-COTC 群集中的 QFS 数据设备禁用隔离 或
- 将共享数据设备放置在 HA-COTC 群集的 Local-Only 设备组中。

针对 HA-COTC 群集中的 QFS 数据设备禁用隔离

1. 登录到 HA-COTC 群集的主节点和 QFS 共享文件系统的活动元数据服务器。以 *root* 用户身份登录。

   在示例中，*qfs1mds-node1* 是主节点和活动元数据服务器：

```
[qfs1mds-node1]root@solaris:~#
```

2. 针对 */etc/opt/SUNWsamfs/mcf* 文件中定义的每台数据 (*mr*) 设备，禁用隔离。使用命令 *cldevice set -p default_fencing=nofencing-noscrub device-identifier*，其中 *device-identifier* 是针对 *mcf* 文件第一列中的设备列出的设备标识符。

   请勿针对元数据 (*mm*) 设备禁用隔离！在 HA-COTC 配置中，QFS 元数据 (*mm*) 设备属于群集的一部分，而 QFS 共享数据 (*mr*) 设备则不属于。数据设备直接连接到群集外的客户机。出于这个原因，HA-COTC 数据 (*mr*) 设备必须作为不受 Solaris

Cluster 软件管理的本地设备进行管理。否则，Solaris Cluster 软件和 QFS 可能会发生冲突并损坏数据。

在上面的示例中，我们将 *d4* 和 *d5* 设备配置为 *qfs1* 文件系统的数据设备。因此我们针对这些设备全局禁用隔离（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）：

```
[qfs1mds-node1]root@solaris:~# cldevice set -p /
default_fencing=nofencing-noscrub d4
[qfs1mds-node1]root@solaris:~# cldevice set -p /
default_fencing=nofencing-noscrub d5
```

3. 接下来，在主 HA-COTC 群集节点上挂载 QFS 文件系统。

将共享数据设备放置在 HA-COTC 群集的 Local-Only 设备组中

1. 登录到 HA-COTC 群集的主节点和 QFS 共享文件系统的活动元数据服务器。以 *root* 用户身份登录。

   在示例中，*qfs1mds-node1* 是主节点和活动元数据服务器：

```
[qfs1mds-node1]root@solaris:~#
```

2. 将属于文件系统的所有数据 (*mr*) 设备放置在 *localonly* 设备组中。使用命令 *cldevicegroup set -d device-identifier-list -p localonly=true -n active-mds-node device-group*，其中 *device-list* 是逗号分隔的设备标识符列表，*active-mds-node* 是活动元数据服务器通常所在的主节点，*device-group* 是您为设备组选择的名称。

   在以下示例中，我们将数据设备 *d4* 和 *d5*（*mcf* 设备编号 *102* 和 *103*）放置在主节点上的本地设备组 *mdsdevgrp* 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）：

```
[qfs1mds-node1]root@solaris:~# cldevicegroup set -d d4,d5 -p localonly=true /
-n node1mds mdsdevgrp
[qfs1mds-node1]root@solaris:~#
```

3. 接下来，在主 HA-COTC 群集节点上挂载 QFS 文件系统。

在主 HA-COTC 节点上挂载 QFS 文件系统

1. 登录到 HA-COTC 群集的主节点和 QFS 共享文件系统的活动元数据服务器。以 *root* 用户身份登录。

   在示例中，*qfs1mds-node1* 是主节点和活动元数据服务器：

```
[qfs1mds-node1]root@solaris:~#
```

2. 备份操作系统的 */etc/vfstab* 文件。

```
[qfs1mds-node1]root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
```

3. 在文本编辑器中打开操作系统的 */etc/vfstab* 文件，并为新的文件系统添加一行。在第一列 (*Device to Mount*) 中输入文件系统名称，后跟一个或多个空格。

   在示例中，使用 *vi* 文本编辑器。我们为 *qfs1* 文件系统添加了一行：

```
[qfs1mds-node1]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       -
...
qfs1       -
```

4. 在 */etc/vfstab* 文件的第二列 (*Device to fsck*) 中，输入一个连字符 (-)，后跟一个或多个空格。

   连字符告知操作系统跳过文件系统完整性检查。这些检查是针对 UFS 文件系统的，而不是针对 SAMFS 文件系统。

```
[qfs1mds-node1]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       -
...
qfs1       -
```

5. 在 */etc/vfstab* 文件的第三列中，输入相对于群集的文件系统挂载点。选择并非直接位于系统根目录下的子目录。

   将共享 QFS 文件系统直接挂载到根目录下可能会在使用 *SUNW.qfs* 资源类型时导致故障转移问题。在示例中，我们将群集上的挂载点设置为 */global/ha-cotc/qfs1*：

```
#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       -
...
qfs1       -        /global/ha-cotc/qfs1
```

6. 像处理任何共享 QFS 文件系统那样，填充 */etc/vfstab* 文件记录的剩余字段。然后保存文件并关闭编辑器。

```
#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       -
...
qfs1       -        /global/ha-cotc/qfs1  samfs   -     no       shared
:wq
[qfs1mds-node1]root@solaris:~#
```

7. 创建高可用性共享文件系统的挂载点。

   带有 *-p* (*parents*) 选项的 *mkdir* 命令可以在 */global* 目录不存在的情况下创建该目录：

```
[qfs1mds-node1]root@solaris:~# mkdir -p /global/ha-cotc/qfs1
```

8. 在主节点上挂载高可用性共享文件系统。

```
[qfs1mds-node1]root@solaris:~# mount /global/ha-cotc/qfs1
```

9. 接下来，在辅助 HA-COTC 群集节点上配置潜在 QFS 元数据服务器。

在辅助 HA-COTC 群集节点上配置潜在 QFS 元数据服务器

双节点群集的辅助节点充当潜在元数据服务器。潜在元数据服务器是可以访问元数据设备的主机，因此可以承担元数据服务器的职责。因此，如果主节点上的活动元数据服务器发生故障，则 Solaris Cluster 软件可以故障转移到辅助节点并激活潜在元数据服务器。要配置潜在元数据服务器，请执行以下任务：

- 在辅助 HA-COTC 节点上创建高性能的 QFS 文件系统
- 在辅助 HA-COTC 节点上挂载 QFS 文件系统。

在辅助 HA-COTC 节点上创建高性能的 QFS 文件系统

1. 以 *root* 用户身份登录到 HA-COTC 群集的辅助节点。

   在示例中，*qfs1mds-node2* 是辅助节点和潜在元数据服务器：

```
[qfs1mds-node2]root@solaris:~#
```

2. 将 */etc/opt/SUNWsamfs/mcf* 文件从主节点复制到辅助节点中。
3. 检查 *mcf* 文件中是否存在错误。使用命令 */opt/SUNWsamfs/sbin/sam-fsd* 并更正发现的任何错误。

   sam-fsd 命令会读取 Oracle HSM 配置文件并初始化文件系统。如果遇到错误，该命令将停止。在示例中，检查主机 *qfs1mds-node1* 上的 *mcf* 文件：

```
[qfs1mds-node2]root@solaris:~# sam-fsd
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
[qfs1mds-node2]root@solaris:~#
```

4. 接下来，在辅助 HA-COTC 群集节点上挂载 QFS 文件系统。

在辅助 HA-COTC 节点上挂载 QFS 文件系统

1. 以 *root* 用户身份登录到 HA-COTC 群集的辅助节点。

   在示例中，*qfs1mds-node2* 是辅助节点：

```
[qfs1mds-node2]root@solaris:~#
```

2. 备份操作系统的 */etc/vfstab* 文件。

```
[qfs1mds-node2]root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
```

3. 在文本编辑器中，打开操作系统的 */etc/vfstab* 文件，并为新的文件系统添加一行。然后保存文件并关闭编辑器。

   在示例中，使用 *vi* 编辑器：

```
[qfs1mds-node2]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       -
...
qfs1        -        /global/ha-cotc/qfs1  samfs    -     no       shared
:wq
[qfs1mds-node2]root@solaris:~#
```

4. 在辅助节点上创建高可用性共享文件系统的挂载点。

```
[qfs1mds-node2]root@solaris:~# mkdir -p /global/ha-cotc/qfs1
[qfs1mds-node2]root@solaris:~#
```

5. 在辅助节点上挂载高可用性共享文件系统。

```
[qfs1mds-node2]root@solaris:~# mount /global/ha-cotc/qfs1
[qfs1mds-node2]root@solaris:~#
```

6. 现在，配置 HA-COTC 元数据服务器的故障转移。

配置 HA-COTC 元数据服务器的故障转移

当您在 Solaris Cluster 软件管理的群集中托管 Oracle HSM 共享文件系统时，可以通过创建 *SUNW.qfs* 群集资源来配置元数据服务器的故障转移，该群集资源为 Oracle HSM 软件定义的一种资源类型（有关详细信息，请参见 *SUNW.qfs* 手册页）。要创建并配置 HA-COTC 配置的资源，请执行如下操作：

1. 以 *root* 用户身份登录到 HA-COTC 群集的主节点。

   在示例中，*qfs1mds-node1* 是主节点：

```
[qfs1mds-node1]root@solaris:~#
```

2. 为 Solaris Cluster 软件定义 QFS 资源类型 *SUNW.qfs*。使用命令 *clresourcetype register SUNW.qfs*。

```
[qfs1mds-node1]root@solaris:~# clresourcetype register SUNW.qfs
[qfs1mds-node1]root@solaris:~#
```

3. 如果注册因找不到注册文件而失败，则将 */opt/SUNWsamfs/sc/etc/* 目录的符号链接放置到 Solaris Cluster 存放资源类型注册文件的 */opt/cluster/lib/rgm/rtreg/* 目录中。

   您在安装 Oracle HSM 软件之前未安装 Oracle Solaris Cluster 软件。通常，Oracle HSM 会在安装期间检测到 Solaris Cluster 时自动提供 *SUNW.qfs* 注册文件的位置。因此您需要手动创建一个链接。

```
[qfs1mds-node1]root@solaris:~# cd /opt/cluster/lib/rgm/rtreg/
[qfs1mds-node1]root@solaris:~# ln -s /opt/SUNWsamfs/sc/etc/SUNW.qfs SUNW.qfs
[qfs1mds-node1]root@solaris:~#
```

4. 创建 QFS 元数据服务器的资源组。使用 Solaris Cluster 命令 *clresourcegroup create -n node-list group-name*，其中 *node-list* 是逗号分隔的两个群集节点名称的列表，*group-name* 是我们想要用于资源组的名称。

   在示例中，使用 HA-COTC 服务器节点作为成员创建资源组 *qfsrg*（请注意，下面的命令是作为单行输入的－使用反斜杠字符对换行符进行转义）：

```
[qfs1mds-node1]root@solaris:~# clresourcegroup create -n / qfs1mds-node1,qfs1mds-node2 qfsrg
[qfs1mds-node1]root@solaris:~#
```

5. 在新的资源组中，设置活动元数据服务器的虚拟主机名。使用 Solaris Cluster 命令 *clreslogicalhostname create -g group-name virtualMDS*，其中 *group-name* 是 QFS 资源组的名称，而 *virtualMDS* 是虚拟主机名。

   使用您在 hosts 文件中用于共享文件系统的相同虚拟主机名。在示例中，我们在 *qfsr* 资源组中创建 *qfs1mds* 虚拟主机：

```
[qfs1mds-node1]root@solaris:~# clreslogicalhostname create -g qfsrg qfs1mds
```

6. 向资源组中添加 QFS 文件系统资源。使用命令 *clresource create -g group-name -t SUNW.qfs -x QFSFileSystem=mount-point -y Resource_dependencies=virtualMDS resource-name*，其中：
   - *group-name* 是 QFS 资源组的名称。
   - *mount-point* 是群集中文件系统的挂载点，即并非直接位于系统根目录下的子目录。

     将共享 QFS 文件系统直接挂载到根目录下可能会在使用 *SUNW.qfs* 资源类型时导致故障转移问题。
   - *virtualMDS* 是活动元数据服务器的虚拟主机名。
   - *resource-name* 是您想要赋予资源的名称。

   在本示例中，我们在 *qfsrg* 资源组中创建名为 *hasqfs* 的 *SUNW.qfs* 类型的资源。我们将 *SUNW.qfs* 扩展属性 *QFSFileSystem* 设置为 */global/ha-cotc/qfs1* 挂载点，将标准属性 *Resource_dependencies* 设置为活动元数据服务器 *qfs1mds* 的逻辑主机（请注意，下面的命令是作为单行输入的－使用反斜杠字符对换行符进行转义）：

```
[qfs1mds-node1]root@solaris:~# clresource create -g qfsrg -t SUNW.qfs /
-x QFSFileSystem=/global/ha-cotc/qfs1 -y Resource_dependencies=qfs1mds hasqfs
```

7. 使资源组联机。使用命令 *clresourcegroup online -emM group-name*，其中 *group-name* 是 QFS 资源组的名称。

   在示例中，我们使 *qfsr* 资源组联机：

```
[qfs1mds-node1]root@solaris:~# clresourcegroup manage qfsrg
[qfs1mds-node1]root@solaris:~# clresourcegroup online -emM qfsrg
```

8. 确保 QFS 资源组处于联机状态。使用 Solaris Cluster *clresourcegroup status* 命令。

   在示例中，主节点 *sam1mds-node1* 上的 *qfsrg* 资源组处于 *online* 状态：

```
[qfs1mds-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
qfsrg       qfs1mds-node1   No          Online
            qfs1mds-node2   No            Offline
```

9. 通过将资源组移动到辅助节点确保资源组正确地故障转移。使用 Solaris Cluster 命令 *clresourcegroup switch -n node2 group-name*，其中 *node2* 是辅助节点的名称，*group-name* 是您为 HA-QFS 资源组选择的名称。然后，使用 *clresourcegroup status* 检查结果。

   在示例中，我们将 *qfsrg* 资源组移动到 *qfs1mds-node2* 并确认资源组在指定节点上处于联机状态：

```
[qfs1mds-node1]root@solaris:~# clresourcegroup switch -n qfs1mds-node2 qfsrg
[qfs1mds-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
qfsrg       qfs1mds-node1   No          Offline
            qfs1mds-node2   No          Online
```

10. 将资源组移回到主节点。使用 Solaris Cluster 命令 *clresourcegroup switch -n node1 group-name*，其中 *node1* 是主节点的名称，*group-name* 是您为 HA-QFS 资源组选择的名称。然后，使用 *clresourcegroup status* 检查结果。

    在示例中，我们将 *qfsrg* 资源组成功移回到 *qfs1mds-node1*：

```
[qfs1mds-node1]root@solaris:~# clresourcegroup switch -n qfs1mds-node1 qfsrg
[qfs1mds-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
```

```
----------  -------------   ---------   ------
qfsrg       qfs1mds-node1   No          Online
            qfs1mds-node2   No          Offline
```

11. 接下来，将 HA-COTC 群集外部的主机配置为 QFS 共享文件系统客户机。

将 HA-COTC 群集外部的主机配置为 QFS 共享文件系统客户机

将每台主机配置为不具有文件系统元数据设备访问权限的 QFS 客户机，以便这些客户机不会影响群集内元数据服务器的高可用性配置。

对于每台 HA-COTC 共享文件系统客户机，请执行如下操作：

1. 登录到 HA-COTC 群集的主节点。以 *root* 用户身份登录。

```
[qfs1mds-node1]root@solaris:~#
```

2. 显示群集的设备配置。使用 Solaris Cluster 命令 */usr/global/bin/cldevice list -v*。

```
[qfs1mds-node1]root@solaris:~# cldevice list -v
DID Device          Full Device Path
----------          ----------------
d1                  qfs1mds-node1:/dev/rdsk/c0t0d0
d2                  qfs1mds-node1:/dev/rdsk/c0t6d0
...
d7                  qfs1mds-node2:/dev/rdsk/c0t1d0
[qfs1mds-node1]root@solaris:~#
```

3. 检查 *cldevice list -v* 命令的输出。记下与每个 QFS 数据 (*mr*) 设备的设备标识符对应的 */dev/rdsk/* 路径。

   在示例中，QFS 数据设备为 *d4* 和 *d5*：

```
[qfs1mds-node1]root@solaris:~# cldevice list -v
DID Device          Full Device Path
----------          ----------------
d1                  qfs1mds-node1:/dev/rdsk/c0t0d0
d2                  qfs1mds-node1:/dev/rdsk/c0t6d0
d3                  qfs1mds-node1:/dev/rdsk/c1t1d0
d3                  qfs1mds-node2:/dev/rdsk/c1t1d0
d4                  qfs1mds-node1:/dev/rdsk/c1t2d0
d4                  qfs1mds-node2:/dev/rdsk/c1t2d0
d5                  qfs1mds-node1:/dev/rdsk/c1t3d0
d5                  qfs1mds-node2:/dev/rdsk/c1t3d0
```

```
d6                   qfs1mds-node2:/dev/rdsk/c0t0d0
d7                   qfs1mds-node2:/dev/rdsk/c0t1d0
[qfs1mds-node1]root@solaris:~#
```

4. 以 *root* 用户身份登录到 HA-COTC 群集的客户机主机。

   在示例中，*qfs1client1* 为客户机主机：

```
[qfs1mds-node1]root@solaris:~# ssh root@qfs1client1
[qfs1client1]root@solaris:~#
```

5. 在客户机主机上，检索共享文件系统的配置信息。使用 *samfsconfig /dev/rdsk/** 命令。

   *samfsconfig /dev/rdsk/** 命令在指定路径中搜索属于 QFS 文件系统的已连接设备。在示例中，该命令找到了 *qfs1* 数据 (*mr*) 设备的路径。如预期的那样，它找不到元数据 (*mm*) 设备，因此它在列出共享数据设备之前返回 *Missing slices* 和 *Ordinal 0* 消息：

```
[qfs1client1]root@solaris:~# samfsconfig /dev/rdsk/*
# Family Set 'qfs1' Created Thu Dec 21 07:17:00 2013
# Missing slices
# Ordinal 0
# /dev/rdsk/c1t2d0s0   102         mr       qfs1   -
# /dev/rdsk/c1t3d0s1   103         mr       qfs1   -
```

6. 将 *samfsconfig* 命令的输出与服务器上 Solaris Cluster *cldevice list* 命令的输出进行比较。确保两个输出针对 *mr* 数据设备报告相同的设备路径。

   samfsconfig 和 *cldevice list* 命令应该显示相同的设备，但是控制器编号 (*cN*) 可能不同。在示例中，*samfsconfig* 和 *cldevice list* 命令确实指向相同的设备。

   在元数据服务器节点上，*/etc/opt/SUNWsamfs/mcf* 文件使用群集设备标识符 *d4* 和 *d5* 标识共享 *mr* 数据设备 *102* 和 *103*：

```
/dev/did/dsk/d4s0     102        mr         qfs1  -
/dev/did/dsk/d5s1     103        mr         qfs1  -
```

   元数据服务器节点上的 *cldevice list* 命令将群集设备标识符 *d4* 和 *d5* 映射到路径 */dev/rdisk/c1t2d0* 和 */dev/rdisk/c1t3d0*：

```
d4                   qfs1mds-node1:/dev/rdsk/c1t2d0
d5                   qfs1mds-node1:/dev/rdsk/c1t3d0
```

在客户机节点上，*samfsconfig* 命令也通过路径 */dev/rdisk/c1t2d0* 和 */dev/rdisk/c1t3d0* 标识共享 *mr* 数据设备 *102* 和 *103*：

```
 /dev/rdsk/c1t2d0s0    102        mr       qfs1  -
 /dev/rdsk/c1t3d0s1    103        mr       qfs1  -
```

7. 在文本编辑器中打开客户机的 */etc/opt/SUNWsamfs/mcf* 文件。针对 HA-COTC 共享文件系统添加一个条目。该条目应与元数据服务器 *mcf* 文件中的对应条目精确匹配。

   在示例中，使用 *vi* 编辑器为文件 QFS 共享系统 *qfs1*（设备序号为 *100*）创建一个条目：

```
[qfs1client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family   Device   Additional
# Identifier          Ordinal    Type       Set      State    Parameters
#------------------   ---------  ---------  -------  ------   ----------------
qfs1                  100        ma         qfs1     -        shared
```

8. 在新行中，针对 HA-COTC 共享文件系统的元数据 (*mm*) 设备添加一个条目。在第一列 (*Equipment Identifier*) 中，输入关键字 *nodev*。

```
# Equipment           Equipment  Equipment  Family   Device   Additional
# Identifier          Ordinal    Type       Set      State    Parameters
#------------------   ---------  ---------  -------  ------   ----------------
qfs1                  100        ma         qfs1      -       shared
 nodev
```

9. 使用元数据服务器 *mcf* 文件中使用的相同设备序号、系列集和设备状态参数填充 HA-COTC 文件系统元数据 (*mm*) 设备的其余字段。

```
# Equipment           Equipment  Equipment  Family   Device   Additional
# Identifier          Ordinal    Type       Set      State    Parameters
#------------------   ---------  ---------  -------  ------   ----------------
qfs1                  100        ma         qfs1      -       shared
 nodev                 101        mm         qfs1     -
```

10. 从 *samfsconfig* 输出复制数据 (*mr*) 设备的完整条目。将这些条目粘贴到客户机的 */etc/opt/SUNWsamfs/mcf* 文件中。删除 *samfsconfig* 插入的前导注释 (*#*) 标记。然后保存文件并关闭编辑器。

```
# Equipment           Equipment  Equipment  Family   Device   Additional
# Identifier          Ordinal    Type       Set      State    Parameters
#------------------   ---------  ---------  -------  ------   -----------------
```

```
qfs1                  100         ma          qfs1       -       shared
 nodev                 101         mm          qfs1      -
 /dev/rdsk/c1t2d0s0    102         mr          qfs1      -
 /dev/rdsk/c1t3d0s1    103         mr          qfs1      -
:wq
[qfs1client1]root@solaris:~#
```

11. 检查 *mcf* 文件中是否存在错误。使用命令 */opt/SUNWsamfs/sbin/sam-fsd* 并更正发现的任何错误。

    sam-fsd 命令会读取 Oracle HSM 配置文件并初始化文件系统。如果遇到错误，该命令将停止。在示例中，检查主机 *qfs1client1* 上的 *mcf* 文件：

```
[qfs1client1]root@solaris:~# sam-fsd
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
[qfs1client1]root@solaris:~#
```

12. 在文本编辑器中打开客户机操作系统的 */etc/vfstab* 文件，并使用在服务器上使用的相同参数针对新文件系统添加一个条目。然后保存文件并关闭编辑器。

    在示例中，使用 *vi* 编辑器：

```
[qfs1client1]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       -
...
qfs1       -        /global/ha-cotc/qfs1  samfs   -     no       shared
:wq
[qfs1client1]root@solaris:~#
```

13. 在客户机上创建高可用性共享文件系统的挂载点。

```
[qfs1client1]root@solaris:~# mkdir -p /global/qfs1
[qfs1client1]root@solaris:~#
```

14. 在客户机上挂载高可用性共享文件系统。

在示例中

```
[qfs1client1]root@solaris:~# mount /global/qfs1
[qfs1client1]root@solaris:~#
```

15. 重复此过程直到完成所有 HA-COTC 客户机的配置。
16. 如果计划使用边带数据库功能，请转至第 10 章 配置报告数据库。
17. 否则，请转至第 11 章 配置通知和日志记录。

高可用性 Oracle HSM 共享归档文件系统

高可用性 Oracle Hierarchical Storage Manager (HA-SAM) 配置可通过确保 QFS 元数据服务器和 Oracle Hierarchical Storage Manager 应用程序在服务器主机发生故障的情况下仍能继续运行来维持归档文件系统的可用性。该文件系统在由 Solaris Cluster 软件管理的双节点群集上托管的活动 QFS 元数据服务器和潜在 QFS 元数据服务器之间共享。如果活动群集节点发生故障，则群集软件自动激活正常运行节点上的潜在 Oracle HSM 服务器并转移正在运行操作的控制权。由于 QFS 文件系统和 Oracle HSM 应用程序的本地存储目录是共享的并且已挂载，因此仍可无中断地访问数据和元数据。

凭借通过活动元数据服务器发送所有 I/O，HA-SAM 配置可确保群集环境中的文件系统一致性。仅出于可访问性原因共享 HA-SAM 文件系统。您无法像在其他 SAM-QFS 共享文件系统配置中那样将潜在元数据服务器主机用作文件系统客户机。除非在节点故障转移期间激活了潜在元数据服务器，否则潜在元数据服务器不执行 I/O。可以使用 NFS 与客户机共享 HA-SAM 文件系统。但您必须确保以独占方式从活动元数据服务器节点导出共享。

高可用性归档文件系统依赖三种 Solaris Cluster 资源类型：

- *SUNW.hasam*

  如果主要主机发生故障，则 *SUNW.hasam* 资源管理 Oracle Hierarchical Storage Manager 应用程序的故障转移。*SUNW.hasam* 软件随附在 Oracle HSM 软件分发中。
- *SUNW.qfs*

  如果主要主机发生故障，则 *SUNW.qfs* 资源管理 QFS 元数据服务器的故障转移。*SUNW.qfs* 软件随附在 Oracle HSM 软件分发中（有关更多信息，请参见 *SUNW.qfs* 手册页）。
- *SUNW.HAStoragePlus*

  如果主要主机发生故障，则 *SUNW.HAStoragePlus* 资源管理 Oracle Hierarchical Storage Manager 本地存储的故障转移。Oracle HSM 应用程序在服务器主机的本地文件系统中维护易失性归档信息（作业队列和可移除介质目录）。*SUNW.HAStoragePlus* 作为标准资源类型随附在 Solaris Cluster 软件中（有关资源类型

的更多信息，请参见 *Oracle Solaris Cluster* 文档库中的 *Data Services Planning and Administration*（数据服务规划和管理）文档。

要配置所需组件的实例并将它们整合成一个有效的 HA-SAM 归档配置，请执行以下任务：

- 在两个 HA-SAM 群集节点上创建全局 Hosts 文件
- 在两个 HA-SAM 群集节点上创建本地 Hosts 文件
- 在主 HA-SAM 群集节点上配置活动 QFS 元数据服务器
- 在辅助 HA-SAM 群集节点上配置潜在 QFS 元数据服务器
- 为 Oracle HSM 配置文件配置高可用性本地文件系统
- 将 Oracle HSM 配置文件重定位至高可用性本地文件系统
- 将 HA-SAM 群集配置为使用高可用性本地文件系统
- 配置 QFS 文件系统元数据服务器的故障转移
- 配置 Oracle Hierarchical Storage Manager 应用程序的故障转移
- 定义 HA-SAM 解决方案的群集资源依赖性
- 使 HA-SAM 资源组联机并测试配置
- 如果需要，配置高可用性网络文件系统 (High-Availability Network File System, HA-NFS) 共享。

  Oracle Solaris Cluster 联机文档库包含的《*Oracle Solaris Cluster Data Service for Network File System (NFS) Guide*》中提供了设置 HA-NFS 的详细步骤。

在两个 HA-SAM 群集节点上创建全局 Hosts 文件

在归档 Oracle HSM 共享文件系统中，您必须在元数据服务器上配置一个 hosts 文件，以便两个节点上的主机都能访问文件系统的元数据。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. 以 *root* 用户身份登录到 HA-SAM 群集的主节点。

   在示例中，*sam1mds-node1* 是主节点：

```
[sam1mds-node1]root@solaris:~#
```

2. 显示群集配置。使用 */usr/global/bin/cluster show* 命令。在输出中，找到每个 *Node Name* 的记录，并记下 *privatehostname* 和 *Transport Adapter* 名称，以及每个网络适配器的 *ip_address* 属性。

   在示例中，每个节点都有两个网络接口，*hme0* 和 *qfe3*：
   - *hme0* 适配器具有群集用于节点间内部通信的专用网络的 IP 地址。Solaris Cluster 软件会分配与每个专用地址相对应的 *privatehostname*。

默认情况下，主节点和辅助节点的专用主机名分别为 *clusternode1-priv* 和 *clusternode2-priv*。

- *qfe3* 适配器具有公共 IP 地址和公共主机名（即 *sam1mds-node1* 和 *sam1mds-node2*），群集使用它们来进行数据传输。

请注意，使用省略号 (*...*) 标记简化了显示内容：

```
[sam1mds-node1]root@solaris:~# cluster show
...
  === Cluster Nodes ===
  Node Name:                                    sam1mds-node1...
    privatehostname:                               clusternode1-priv...
    Transport Adapter List:                        qfe3, hme0...
    Transport Adapter:                          qfe3...
      Adapter Property(ip_address):                172.16.0.12...
    Transport Adapter:                          hme0...
      Adapter Property(ip_address):                10.0.0.129...
  Node Name:                                    sam1mds-node2...
    privatehostname:                               clusternode2-priv...
    Transport Adapter List:                        qfe3, hme0...
      Adapter Property(ip_address):                172.16.0.13...
    Transport Adapter:                          hme0...
      Adapter Property(ip_address):                10.0.0.122
```

3. 使用文本编辑器，创建文件 */etc/opt/SUNWsamfs/hosts.family-set-name*，其中 *family-set-name* 是 */etc/opt/SUNWsamfs/mcf* 文件分配给文件系统设备的系列集名称。

   在示例中，使用 *vi* 文本编辑器创建文件 *hosts.sam1*。我们添加一些可选标题以显示主机表中的列，每行以井号 (*#*) 开头表示该行为注释：

```
[sam1mds-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sam1
# /etc/opt/SUNWsamfs/hosts.sam1
#                                          Server   On/  Additional
#Host Name     Network Interface           Ordinal  Off  Parameters
#------------  --------------------------  -------  ---  ----------
```

4. 在表的第一列中，输入主元数据服务器节点和辅助元数据服务器节点的主机名称，后面跟有一些空格，让每个条目都独占一行。

   在 hosts 文件中，行即为行（记录），空格为列（字段）分隔符。在本示例中，*Host Name* 列的前两行包含值 *sam1mds-node1* 和 *sam1mds-node2*，它们是托管文件系统元数据服务器的群集节点的主机名：

```
#                                        Server   On/  Additional
#Host Name     Network Interface          Ordinal  Off  Parameters
#------------  -------------------------  -------  ---  ----------
sam1mds-node1
sam1mds-node2
```

5. 在每行的第二列中，开始为 *Host Name* 列中列出的主机提供 *Network Interface* 信息。输入各个 HA-SAM 群集节点的 Solaris Cluster 专用主机名或专用网络地址，后跟一个逗号。

   HA-SAM 服务器节点使用专用主机名在高可用性群集内进行服务器间的通信。在示例中，使用专用主机名 *clusternode1-priv* 和 *clusternode2-priv*，这些是 Solaris Cluster 软件分配的默认名称：

```
#                                        Server   On/  Additional
#Host Name     Network Interface          Ordinal  Off  Parameters
#------------  -------------------------  -------  ---  ----------
sam1mds-node1  clusternode1-priv,
sam1mds-node2  clusternode2-priv,
```

6. 在每行第二列中的逗号后面，输入活动元数据服务器的公共主机名，后跟空格。

   HA-SAM 服务器节点使用公共数据网络与群集外的主机通信。由于活动元数据服务器的 IP 地址和主机名在故障转移期间会发生更改（从 *sam1mds-node1* 更改为 *sam1mds-node2*，反之亦然），因此，我们对二者都使用虚拟主机名 *sam1mds*。稍后，我们将 Solaris Cluster 软件配置为始终将 *sam1mds* 的请求路由到活动元数据服务器：

```
#                                        Server   On/  Additional
#Host Name     Network Interface          Ordinal  Off  Parameters
#------------  -------------------------  -------  ---  ----------
sam1mds-node1  clusternode1-priv,sam1mds
sam1mds-node2  clusternode2-priv,sam1mds
```

7. 在每行的第三列中，输入服务器的序号（*1* 表示活动元数据服务器，*2* 表示潜在元数据服务器），后跟空格。

   在本示例中，只有一个元数据服务器，主节点 *sam1mds-node1* 是活动元数据服务器，因此其序号为 *1*，辅助节点 *sam1mds-node2* 的序号为 *2*：

```
#                                        Server   On/  Additional
#Host Name     Network Interface          Ordinal  Off  Parameters
#------------  -------------------------  -------  ---  ----------
sam1mds-node1  clusternode1-priv,sam1mds  1
```

```
sam1mds-node2  clusternode2-priv,sam1mds      2
```

8. 在每行的第四列中，输入 *0*（零），后跟空格。

   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 *samsharefs* 手册页）。

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------  -------  ---  ----------
sam1mds-node1  clusternode1-priv,sam1mds  1        0
sam1mds-node2  clusternode2-priv,sam1mds  2        0
```

9. 在主节点行的第五列中，输入关键字 *server*。然后保存文件并关闭编辑器。

   server 关键字用于标识默认的活动元数据服务器：

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------  -------  ---  ----------
sam1mds-node1  clusternode1-priv,sam1mds  1        0    server
sam1mds-node2  clusternode2-priv,sam1mds  2        0
:wq
[sam1mds-node1]root@solaris:~#
```

10. 将全局 */etc/opt/SUNWsamfs/hosts.family-set-name* 文件的副本放置在潜在元数据服务器上。
11. 现在，在两个 HA-SAM 群集节点上创建本地 hosts 文件。

在两个 HA-SAM 群集节点上创建本地 Hosts 文件

在高可用性归档共享文件系统中，您需要使用 Solaris Cluster 软件定义的专用网络确保服务器可以互相通信。为此，需要使用专门配置的本地 hosts 文件选择性地在服务器的网络接口之间路由网络流量。

每个文件系统主机通过首先查看元数据服务器上的 */etc/opt/SUNWsamfs/hosts.family-set-name* 文件来确定用于其他主机的网络接口。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。通过在不同的安排下使用各文件中的 IP 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. 以 *root* 用户身份登录到 HA-SAM 群集的主节点。

   在示例中，*sam1mds-node1* 是主节点：

```
[sam1mds-node1]root@solaris:~#
```

2. 使用文本编辑器通过路径和文件名 */etc/opt/SUNWsamfs/hosts.family-set-name.local* 在活动元数据服务器上创建本地 hosts 文件，其中 *family-set-name* 是 */etc/opt/SUNWsamfs/mcf* 文件分配给文件系统设备的系列集名称。仅包括您希望活动服务器与潜在服务器通信时使用的网络接口。然后保存文件并关闭编辑器。

   在示例中，我们希望活动元数据服务器和潜在元数据服务器通过专用网络互相通信。因此活动元数据服务器上的本地 hosts 文件 *hosts.sam1.local* 仅列出活动服务器和潜在服务器的群集专用地址：

```
[sam1mds-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sam1.local
#                                          Server   On/  Additional
#Host Name     Network Interface            Ordinal  Off  Parameters
#------------  ---------------------------  -------  ---  ----------
sam1mds-node1  clusternode1-priv            1        0    server
sam1mds-node2  clusternode2-priv            2        0
:wq
[sam1mds-node1]root@solaris:~#
```

3. 以 *root* 用户身份登录辅助群集节点。

   在示例中，*sam1mds-node2* 是辅助节点：

```
[sam1mds-node1]root@solaris:~# ssh root@sam1mds-node2
Password:
[sam1mds-node2]root@solaris:~#
```

4. 使用文本编辑器在潜在元数据服务器上创建一个本地 hosts 文件。使用路径和文件名 */etc/opt/SUNWsamfs/hosts.family-set-name.local*，其中 *family-set-name* 是 */etc/opt/SUNWsamfs/mcf* 文件分配给文件系统设备的系列集名称。仅包括您希望潜在服务器与活动服务器通信时使用的网络接口。然后保存文件并关闭编辑器。

   在示例中，我们希望活动元数据服务器和潜在元数据服务器通过专用网络互相通信。因此潜在元数据服务器上的本地 hosts 文件 *hosts.sam1.local* 仅列出活动服务器和潜在服务器的群集专用地址：

```
[sam1mds-node2]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sam1.local
#                                          Server   On/  Additional
#Host Name     Network Interface            Ordinal  Off  Parameters
```

```
#------------  ----------------------------  -------  ---  ----------
sam1mds-node1  clusternode1-priv              1        0    server
sam1mds-node2  clusternode2-priv              2        0
:wq
[sam1mds-node2]root@solaris:~# exit
[sam1mds-node1]root@solaris:~#
```

5. 接下来，在主 HA-SAM 群集节点上配置活动 QFS 元数据服务器。

在主 HA-SAM 群集节点上配置活动 QFS 元数据服务器

1. 选择将同时充当 HA-SAM 群集的主节点和 QFS 共享文件系统的活动元数据服务器的群集节点。以 *root* 用户身份登录。

   在示例中，*sam1mds-node1* 是主节点：

```
[sam1mds-node1]root@solaris:~#
```

2. 选择要用于 QFS 文件系统的全局存储设备。使用命令 */usr/global/bin/cldevice list -v*。

   Solaris Cluster 软件向连接到群集节点的所有设备分配唯一的设备标识符 (Device Identifier, DID)。全局设备可以从群集中的所有节点进行访问，而本地设备仅可以从挂载它们的主机进行访问。全局设备在故障转移后仍然可以访问。本地设备则不然。

   在示例中，请注意 *d1*、*d2*、*d7* 和 *d8* 设备从两个节点都无法访问。所以在配置高可用性 QFS 共享文件系统时，我们从 *d3*、*d4* 和 *d5* 设备中进行选择：

```
[sam1mds-node1]root@solaris:~# cldevice list -v
DID Device          Full Device Path
----------          ----------------
d1                  sam1mds-node1:/dev/rdsk/c0t0d0
d2                  sam1mds-node1:/dev/rdsk/c0t6d0
d3                  sam1mds-node1:/dev/rdsk/c1t1d0
d3                  sam1mds-node2:/dev/rdsk/c1t1d0
d4                  sam1mds-node1:/dev/rdsk/c1t2d0
d4                  sam1mds-node2:/dev/rdsk/c1t2d0
d5                  sam1mds-node1:/dev/rdsk/c1t3d0
d5                  sam1mds-node2:/dev/rdsk/c1t3d0
d6                  sam1mds-node2:/dev/rdsk/c0t0d0
d7                  sam1mds-node2:/dev/rdsk/c0t1d0
```

3. 在选定的主节点上，创建使用 *mr* 数据设备的高性能 *ma* 文件系统。在文本编辑器中，打开 */etc/opt/SUNWsamfs/mcf* 文件。

在示例中，我们配置文件系统 *sam1*。我们将设备 *d3* 配置为元数据设备（设备类型 *mm*），并将 *d4* 和 *d5* 用作数据设备（设备类型 *mr*）：

```
[sam1mds-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family   Device   Additional
# Identifier          Ordinal    Type       Set      State    Parameters
#------------------   ---------  ---------  -------  ------   -----------------
sam1                  100        ma         sam1     -
 /dev/did/dsk/d3s0    101        mm         sam1     -
 /dev/did/dsk/d4s0    102        mr         sam1     -
 /dev/did/dsk/d5s1    103        mr         sam1     -
```

4. 在 */etc/opt/SUNWsamfs/mcf* 文件的文件系统条目的 *Additional Parameters* 列中输入 *shared* 参数。保存文件。

```
[sam1mds-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family   Device   Additional
# Identifier          Ordinal    Type       Set      State    Parameters
#------------------   ---------  ---------  -------  ------   -----------------
sam1                  100        ma         sam1     -        shared
 /dev/did/dsk/d3s0     101        mm         sam1     -
 /dev/did/dsk/d4s0     102        mr         sam1     -
 /dev/did/dsk/d5s1     103        mr         sam1     -
:wq
[sam1mds-node1]root@solaris:~#
```

5. 检查 *mcf* 文件中是否存在错误。使用命令 */opt/SUNWsamfs/sbin/sam-fsd* 并更正发现的任何错误。

   sam-fsd 命令会读取 Oracle HSM 配置文件并初始化文件系统。如果遇到错误，该命令将停止。在示例中，检查主机 *sam1mds-node1* 上的 *mcf* 文件：

```
[sam1mds-node1]root@solaris:~# sam-fsd
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
[sam1mds-node1]root@solaris:~#
```

6. 创建文件系统。使用命令 */opt/SUNWsamfs/sbin/sammkfs -S family-set-name*，其中 *family-set-name* 是 */etc/opt/SUNWsamfs/mcf* 文件分配给文件系统设备的系列集名称。

sammkfs 命令读取 *hosts.family-set-name* 和 *mcf* 文件，并创建具有指定属性的 Oracle HSM 文件系统。

```
[sam1mds-node1]root@solaris:~# sammkfs -S sam1
Building 'sam1' will destroy the contents of devices:
  ...
Do you wish to continue? [y/N]yes ...
[sam1mds-node1]root@solaris:~#
```

7. 在文本编辑器中打开操作系统的 */etc/vfstab* 文件，并为新的文件系统添加一行。在第一列中输入文件系统名称，然后输入几个空格，在第二列中输入一个连字符，然后输入更多的空格。

   在示例中，使用 *vi* 文本编辑器。我们为 *sam1* 文件系统添加了一行。连字符阻止操作系统尝试使用 UFS 工具检查文件系统的完整性：

```
[sam1mds-node1]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       -
...
sam1       -
```

8. 在 */etc/vfstab* 文件的第三列中，输入相对于群集的文件系统挂载点。选择并非直接位于系统根目录下的子目录。

   将共享 QFS 文件系统直接挂载到根目录下可能会在使用 *SUNW.qfs* 资源类型时导致故障转移问题。在示例中，我们将群集上的挂载点设置为 */global/ha-sam/sam1*：

```
#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       -
...
sam1       -        /global/ha-sam/sam1
```

9. 像处理任何共享 Oracle HSM 共享文件系统那样，填充 */etc/vfstab* 文件记录的剩余字段。然后保存文件并关闭编辑器。

```
#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       -
...
sam1       -        /global/ha-sam/sam1 samfs   -     no       shared
:wq
[sam1mds-node1]root@solaris:~#
```

10. 创建高可用性文件系统的挂载点。

    带有 *-p* (*parents*) 选项的 *mkdir* 命令可以在 */global* 目录不存在的情况下创建该目录：

```
[sam1mds-node1]root@solaris:~# mkdir -p /global/ha-sam/sam1
```

11. 在主节点上挂载高可用性共享文件系统。

```
[sam1mds-node1]root@solaris:~# mount /global/ha-sam/sam1
```

12. 接下来，在辅助 HA-SAM 群集节点上配置潜在 QFS 元数据服务器。

在辅助 HA-SAM 群集节点上配置潜在 QFS 元数据服务器

双节点群集的辅助节点充当潜在元数据服务器。潜在元数据服务器是可以访问元数据设备的主机，因此可以承担元数据服务器的职责。因此，如果主节点上的活动元数据服务器发生故障，则 Solaris Cluster 软件可以故障转移到辅助节点并激活潜在元数据服务器。

1. 以 *root* 用户身份登录到 HA-SAM 群集的辅助节点。

   在示例中，*sam1mds-node2* 是辅助节点：

```
[sam1mds-node2]root@solaris:~#
```

2. 将 */etc/opt/SUNWsamfs/mcf* 文件从主节点复制到辅助节点中。
3. 检查 *mcf* 文件中是否存在错误。使用命令 */opt/SUNWsamfs/sbin/sam-fsd* 并更正发现的任何错误。

   sam-fsd 命令会读取 Oracle HSM 配置文件并初始化文件系统。如果遇到错误，该命令将停止。在示例中，检查主机 *sam1mds-node1* 上的 *mcf* 文件：

```
[sam1mds-node2]root@solaris:~# sam-fsd
```

```
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
[sam1mds-node2]root@solaris:~#
```

4. 创建文件系统。使用命令 */opt/SUNWsamfs/sbin/sammkfs -S family-set-name*，其中 *family-set-name* 是 */etc/opt/SUNWsamfs/mcf* 文件分配给文件系统设备的系列集名称。

   sammkfs 命令读取 *hosts.family-set-name* 和 *mcf* 文件，并创建具有指定属性的 Oracle HSM 文件系统。

```
[sam1mds-node2]root@solaris:~# sammkfs sam1
Building 'sam1' will destroy the contents of devices:
  ...
Do you wish to continue? [y/N]yes ...
[sam1mds-node2]root@solaris:~#
```

5. 在文本编辑器中，打开操作系统的 */etc/vfstab* 文件，并为新的文件系统添加一行。然后保存文件并关闭编辑器。

   在示例中，使用 *vi* 编辑器：

```
[sam1mds-node2]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       -
...
sam1       -        /global/ha-sam/sam1 samfs   -     no       shared
:wq
[sam1mds-node2]root@solaris:~#
```

6. 在辅助节点上创建高可用性共享文件系统的挂载点。

```
[sam1mds-node2]root@solaris:~# mkdir -p /global/ha-sam/sam1
```

7. 在辅助节点上挂载高可用性共享文件系统。

```
[sam1mds-node2]root@solaris:~# mount /global/ha-sam/sam1
```

8. 现在，创建 HA-SAM 群集资源组。

创建 HA-SAM 群集资源组

创建将管理 HA-SAM 解决方案的高可用性资源的资源组。

1. 以 *root* 用户身份登录到 HA-SAM 群集的主群集节点。

   在示例中，主节点是 *sam1mds-node1*：

```
[sam1mds-node1]root@solaris:~#
```

2. 创建 Solaris Cluster 资源组以管理 HA-SAM 解决方案资源。使用命令 *clresourcegroup create -n node1,node2 groupname*，其中：
   - *node1* 是主群集节点的主机名。
   - *node2* 是辅助群集节点的主机名。
   - *groupname* 是您为 HA-SAM 资源组选择的名称。

   在本示例中，我们创建名为 *has-rg* 的资源组并包含主机 *sam1mds-node1* 和 *sam1mds-node2*（请注意，下面的命令是作为单行输入的－使用反斜杠字符对换行符进行转义）：

```
[sam1mds-node1]root@solaris:~# clresourcegroup create /
-n sam1mds-node1,sam1mds-node2 has-rg
```

3. 接下来，配置高可用性本地文件系统来存放 Oracle HSM 配置文件。

为 Oracle HSM 配置文件配置高可用性本地文件系统

要成功恢复以下故障转移，Oracle HSM 软件必须重新启动在故障转移发生时运行的归档操作。要重新启动归档操作，该软件必须对系统配置和状态信息具有访问权限，该信息通常存储在活动元数据服务器的本地文件系统中。因此您必须将所需的信息移动到高可用性本地文件系统中，始终可以从群集中的两个节点访问该文件系统。

要创建所需的文件系统，请执行如下操作：

1. 以 *root* 用户身份登录到 HA-SAM 群集的主节点。

   在示例中，主节点是 *sam1mds-node1*：

```
[sam1mds-node1]root@solaris:~#
```

2. 在主群集节点上，在全局设备的空闲分片上创建 UFS 文件系统。使用命令 *newfs /dev/global/dsk/dXsY*，其中 *X* 是全局设备的设备标识符 (Device Identifier, DID) 编号，*Y* 是分片编号。

在本示例中，我们在 */dev/global/dsk/d10s0* 上创建新文件系统：

```
[sam1mds-node1]root@solaris:~# newfs /dev/global/dsk/d10s0
newfs: construct a new file system /dev/global/dsk/d10s0: (y/n)? y
/dev/global/dsk/d10s0: 1112940 sectors in 1374 cylinders of 15 tracks,
54 sectors 569.8MB in 86 cyl groups (16 c/g, 6.64MB/g, 3072 i/g)
super-block backups(for fsck -b #) at:
32, 13056, 26080, 39104, 52128, 65152, 78176, 91200, 104224, . . .
[sam1mds-node1]root@solaris:~#
```

3. 在主群集节点上，在文本编辑器中打开操作系统的 */etc/vfstab* 文件。针对新 UFS 文件系统添加一行。保存文件并关闭编辑器。

   新行应为 */dev/global/dsk/dXsY /dev/global/dsk/dXsY / global/mount_point ufs 5 no global* 形式的空格分隔的列表，其中：

   - *X* 是存放文件系统的全局设备的设备标识符 (Device Identifier, DID) 编号。
   - *Y* 是存放文件系统的分片的编号。
   - */dev/global/dsk/dXsY* 是将要挂载的文件系统设备的名称。
   - */dev/global/dsk/dXsY* 是将由 *fsck* 命令进行检查的文件系统设备的名称。
   - *mount_point* 是要挂载 UFS 文件的子目录的名称。
   - *ufs* 是文件系统类型。
   - *5* 是建议的 *fsck* 传递号。
   - *no* 告知操作系统不应在启动时挂载文件系统。
   - *global* 挂载文件系统，以便两个节点都具有访问权限。

   在示例中，使用 *vi* 编辑器。文件系统名称为 */dev/global/dsk/d10s0*，挂载点为 */global/hasam_cfg*（请注意，文件系统条目是单行－为了适应页面，插入了一个换行符并使用反斜杠字符对其进行了转义）：

```
[sam1mds-node1]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       -
...
sam1       -        /global/ha-samsam1  samfs   -     no       shared
/dev/global/dsk/d10s0   /dev/global/rdsk/d10s0  /global/hasam_cfg  ufs  5 /
     no   global
:wq
[sam1mds-node2]root@solaris:~#
```

4. 在主群集节点上，为高可用性本地文件系统创建挂载点。使用命令 *mkdir -p /global/mount_point*，其中 *mount_point* 是选择的挂载点目录。

   在本示例中，我们创建目录 */global/hasam_cfg*：

```
[sam1mds-node1]root@solaris:~# mkdir -p /global/hasam_cfg
```

5. 以 *root* 用户身份登录辅助群集节点。

   在示例中，辅助节点是 *sam1mds-node2*。我们使用 *ssh* 登录：

```
[sam1mds-node1]root@solaris:~# ssh root@sam1mds-node2
Password:
[sam1mds-node2]root@solaris:~#
```

6. 在辅助节点上，在文本编辑器中打开操作系统的 */etc/vfstab* 文件。针对新 UFS 文件系统添加一个相同的条目。保存文件并关闭编辑器。

```
[sam1mds-node1]root@solaris:~# ssh root@sam1mds-node2
Password:
[sam1mds-node2]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       -
...
sam1       -        /global/ha-samsam1  samfs   -     no       shared
/dev/global/dsk/d10s0   /dev/global/rdsk/d10s0  /global/hasam_cfg  ufs  5  /
    no   global
:wq
[sam1mds-node1]root@solaris:~#
```

7. 在辅助节点上，创建相同的挂载点。

   在本示例中，我们创建 */global/hasam_cfg* 目录。然后，我们关闭 *ssh* 会话并恢复在主节点上的工作：

```
[sam1mds-node2]root@solaris:~# mkdir -p /global/hasam_cfg
[sam1mds-node2]root@solaris:~# exit
[sam1mds-node1]root@solaris:~#
```

8. 在主节点上，挂载高可用性本地文件系统。使用命令 *mount /global/mount_point*，其中 *mount_point* 是选择的挂载点目录。

该命令在两个节点上挂载 UFS 文件系统。在本示例中，我们在 */global/hasam_cfg* 上挂载文件系统：

```
[sam1mds-node1]root@solaris:~# mount /global/hasam_cfg
[sam1mds-node1]root@solaris:~#
```

9. 在主节点上，创建用于存放 Oracle HSM 回写信息的子目录。使用命令 *mkdir -p /global/mount_point/catalog*，其中 *mount_point* 是选择的挂载点目录。

```
[sam1mds-node1]root@solaris:~# mkdir /global/hasam_cfg/catalog
[sam1mds-node1]root@solaris:~#
```

10. 在主节点上，创建用于存放 Oracle HSM 归档目录的子目录。使用命令 *mkdir -p /global/mount_point/stager*，其中 *mount_point* 是选择的挂载点目录。

```
[sam1mds-node1]root@solaris:~# mkdir /global/hasam_cfg/stager
[sam1mds-node1]root@solaris:~#
```

11. 接下来，将 Oracle HSM 配置文件重定位至高可用性本地文件系统。

将 Oracle HSM 配置文件重定位至高可用性本地文件系统

1. 以 *root* 用户身份登录到 HA-SAM 群集的主节点。

   在示例中，主节点是 *sam1mds-node1*：

```
[sam1mds-node1]root@solaris:~#
```

2. 在主节点上，将 *catalog/* 和 *stager/* 目录从其在 */var/opt/SUNWsamfs/* 中的默认位置复制到临时位置。

   在本示例中，我们将指令递归复制到 */var/tmp/* 中（请注意，下面的第一条命令是作为单行输入的－使用反斜杠字符对换行符进行转义）：

```
[sam1mds-node1]root@solaris:~# cp -r /var/opt/SUNWsamfs/catalog /
/var/tmp/catalog
[sam1mds-node1]root@solaris:~# cp -r /var/opt/SUNWsamfs/stager /var/tmp/stager
[sam1mds-node1]root@solaris:~#
```

3. 在主节点上，从 */var/opt/SUNWsamfs/* 中删除 *catalog/* 和 *stager/* 目录。

```
[sam1mds-node1]root@solaris:~# rm -rf /var/opt/SUNWsamfs/catalog
[sam1mds-node1]root@solaris:~# rm -rf /var/opt/SUNWsamfs/stager
[sam1mds-node1]root@solaris:~#
```

4. 在主节点上，创建从目录信息的默认位置到高可用性 UFS 本地文件系统中的新位置的符号链接。使用命令 *ln -s /global/mount_point/catalog /var/opt/SUNWsamfs/catalog*，其中：
   - *mount_point* 是将高可用性本地文件系统连接到节点根文件系统的子目录的名称。
   - */var/opt/SUNWsamfs/catalog* 是默认位置。

   符号链接会自动将对目录信息的请求重定向至新位置。在本示例中，我们创建一个指向新位置 */global/hasam_cfg/catalog* 的 *catalog* 链接（请注意，下面的命令是作为单行输入的－使用反斜杠字符对换行符进行转义）：

```
[sam1mds-node1]root@solaris:~# ln -s /global/hasam_cfg/catalog /
/var/opt/SUNWsamfs/catalog
[sam1mds-node1]root@solaris:~#
```

5. 在主节点上，创建从回写信息的默认位置到高可用性 UFS 本地文件系统中的新位置的符号链接。使用命令 *ln -s /global/mount_point/stager /var/opt/SUNWsamfs/stager*，其中：
   - *mount_point* 是将高可用性本地文件系统连接到节点根文件系统的子目录的名称。
   - */var/opt/SUNWsamfs/stager* 是默认位置。

   符号链接会自动将对回写程序信息的请求重定向至新位置。在本示例中，我们创建一个指向新位置 */global/hasam_cfg/stager* 的 *stager* 链接（请注意，下面的命令是作为单行输入的－使用反斜杠字符对换行符进行转义）：

```
[sam1mds-node1]root@solaris:~# ln -s /global/hasam_cfg/stager / /var/opt/SUNWsamfs/stager
[sam1mds-node1]root@solaris:~#
```

6. 在主节点上，确保符号链接已替换默认的 */var/opt/SUNWsamfs/catalog* 和 */var/opt/SUNWsamfs/stager* 目录。确保这些链接指向高可用性文件系统中的新位置。

   在示例中，链接正确：

```
[sam1mds-node1]root@solaris:~# ls -l /var/opt/SUNWsamfs/catalog
lrwxrwxrwx 1 root other ... /var/opt/SUNWsamfs/catalog -> /global/hasam_cfg/catalog
[sam1mds-node1]root@solaris:~# ls -l /var/opt/SUNWsamfs/stager
lrwxrwxrwx 1 root other ... /var/opt/SUNWsamfs/stager -> /global/hasam_cfg/stager
[sam1mds-node1]root@solaris:~#
```

7. 将 *catalog/* 和 *stager/* 目录的内容从临时位置复制到高可用性共享文件系统。

在本示例中，我们将 *catalog/* 和 *stager/* 目录从 */var/tmp/* 复制到新位置 */global/hasam_cfg/stager*（请注意，下面的命令是作为单行输入的－使用反斜杠字符对换行符进行转义）：

```
[sam1mds-node1]root@solaris:~# cp -rp /var/tmp/catalog/* /
/var/opt/SUNWsamfs/catalog
[sam1mds-node1]root@solaris:~# cp -rp /var/tmp/stager/* /
/var/opt/SUNWsamfs/stager
[sam1mds-node1]root@solaris:~#
```

8. 以 *root* 用户身份登录到 HA-SAM 群集的辅助节点。

   在示例中，使用 *ssh*（secure shell，安全 shell）登录到 *sam1mds-node2*（辅助节点）：

```
[sam1mds-node1]root@solaris:~# ssh root@sam1mds-node2
Password:
[sam1mds-node2]root@solaris:~#
```

9. 在辅助节点上，创建从目录信息的默认位置到高可用性 UFS 本地文件系统中的新位置的符号链接。使用命令 *ln -s /global/mount_point/catalog /var/opt/SUNWsamfs/catalog*，其中：
   - *mount_point* 是将高可用性本地文件系统连接到节点根文件系统的子目录的名称。
   - */var/opt/SUNWsamfs/catalog* 是默认位置。

   符号链接会自动将对目录信息的请求重定向至新位置。在本示例中，我们创建一个指向新位置 */global/hasam_cfg/catalog* 的 *catalog* 链接（请注意，下面的命令是作为单行输入的－使用反斜杠字符对换行符进行转义）：

```
[sam1mds-node2]root@solaris:~# ln -s /global/hasam_cfg/catalog /
/var/opt/SUNWsamfs/catalog
[sam1mds-node2]root@solaris:~#
```

10. 在辅助节点上，创建从回写信息的默认位置到高可用性 UFS 本地文件系统中的新位置的符号链接。使用命令 *ln -s /global/mount_point/stager /var/opt/SUNWsamfs/stager*，其中：
    - *mount_point* 是将高可用性本地文件系统连接到节点根文件系统的子目录的名称。
    - */var/opt/SUNWsamfs/stager* 是默认位置。

    符号链接会自动将对回写程序信息的请求重定向至新位置。在本示例中，我们创建一个指向新位置 */global/hasam_cfg/stager* 的 *stager* 链接（请注意，下面的命令是作为单行输入的－使用反斜杠字符对换行符进行转义）：

```
[sam1mds-node2]root@solaris:~# ln -s /global/hasam_cfg/stager / /var/opt/SUNWsamfs/stager
[sam1mds-node2]root@solaris:~#
```

11. 在辅助节点上，确保符号链接已替换默认的 */var/opt/SUNWsamfs/catalog* 和 */var/opt/SUNWsamfs/stager* 目录。确保这些链接指向高可用性文件系统中的新位置。

    在示例中，链接正确。因此，我们关闭 *ssh* 会话并恢复在主节点上的工作：

```
[sam1mds-node2]root@solaris:~# ls -l /var/opt/SUNWsamfs/catalog
lrwxrwxrwx 1 root other ... /var/opt/SUNWsamfs/catalog -> /global/hasam_cfg/catalog
[sam1mds-node2]root@solaris:~# ls -l /var/opt/SUNWsamfs/stager
lrwxrwxrwx 1 root other ... /var/opt/SUNWsamfs/stager -> /global/hasam_cfg/stager
[sam1mds-node2]root@solaris:~# exit
[sam1mds-node1]root@solaris:~#
```

12. 接下来，将 HA-SAM 群集配置为使用高可用性本地文件系统。

将 HA-SAM 群集配置为使用高可用性本地文件系统

1. 在 HA-SAM 群集的主节点上，注册 *SUNW.HAStoragePlus* 资源类型作为群集配置的一部分。使用 Solaris Cluster 命令 *clresourcetype register SUNW.HAStoragePlus*。

```
[sam1mds-node1]root@solaris:~# clresourcetype register SUNW.HAStoragePlus
[sam1mds-node1]root@solaris:~#
```

2. 在主节点上，创建 *SUNW.HAStoragePlus* 资源类型的新实例并将其与 Solaris Cluster 资源组相关联。使用命令 *clresource create -g groupname -t SUNW.HAStoragePlus -x FilesystemMountPoints=mountpoint -x AffinityOn=TRUE resourcename*，其中：
   - *groupname* 是您为 HA-SAM 资源组选择的名称。
   - *SUNW.HAStoragePlus* 是支持本地文件系统故障转移的 Solaris Cluster 资源类型。
   - *mountpoint* 是存放目录和回写程序文件的高可用性本地文件系统的挂载点。
   - *resourcename* 是您为资源本身选择的名称。

   在本示例中，我们创建类型为 *SUNW.HAStoragePlus* 且名为 *has-cfg* 的资源。我们将新资源添加到资源组 *has-rg* 中。然后我们配置资源扩展属性。我们将 *FilesystemMountPoints* 设置为 */global/hasam_cfg*，将 *AffinityOn* 设置为 *TRUE*（请注意，下面的命令是作为单行输入的—使用反斜杠字符对换行符进行转义）：

```
[sam1mds-node1]root@solaris:~# clresource create -g has-rg /
```

```
-t SUNW.HAStoragePlus -x FilesystemMountPoints=/global/hasam_cfg /
-x AffinityOn=TRUE has-cfg
[sam1mds-node1]root@solaris:~#
```

3. 接下来，配置 QFS 文件系统元数据服务器的故障转移。

配置 QFS 文件系统元数据服务器的故障转移

通过创建 *SUNW.qfs* 群集资源配置元数据服务器的故障转移，该群集资源是由 Oracle HSM 软件定义的资源类型（有关详细信息，请参见 *SUNW.qfs* 手册页）。要创建并配置 HA-SAM 配置的资源，请执行如下操作：

1. 以 *root* 用户身份登录到 HA-SAM 群集的主群集节点。

   在示例中，主节点是 *sam1mds-node1*：

```
[sam1mds-node1]root@solaris:~#
```

2. 为 Solaris Cluster 软件定义资源类型 *SUNW.qfs*。使用命令 *clresourcetype register SUNW.qfs*。

```
[sam1mds-node1]root@solaris:~# clresourcetype register SUNW.qfs
[qfs1mds-node1]root@solaris:~#
```

3. 如果注册因找不到注册文件而失败，则将 */opt/SUNWsamfs/sc/etc/* 目录的符号链接放置到 Solaris Cluster 存放资源类型注册文件的 */opt/cluster/lib/rgm/rtreg/* 目录中。

   如果在安装 Oracle HSM 软件之前未安装 Oracle Solaris Cluster 软件，则注册将失败。通常，Oracle HSM 会在安装期间检测到 Solaris Cluster 时自动提供 *SUNW.qfs* 注册文件的位置。在示例中，我们手动创建链接。

```
[qfs1mds-node1]root@solaris:~# cd /opt/cluster/lib/rgm/rtreg/
[qfs1mds-node1]root@solaris:~# ln -s /opt/SUNWsamfs/sc/etc/SUNW.qfs SUNW.qfs
[qfs1mds-node1]root@solaris:~#
```

4. 在新的资源组中，设置活动元数据服务器的虚拟主机名。使用 Solaris Cluster 命令 *clreslogicalhostname create -g group-name virtualMDS*，其中：
   - *group-name* 是 QFS 资源组的名称。
   - *virtualMDS* 是虚拟主机名。

   使用您在 hosts 文件中用于共享文件系统的相同虚拟主机名。在示例中，我们将虚拟主机名 *sam1mds* 添加到 *has-rg* 资源组中：

```
[sam1mds-node1]root@solaris:~# clreslogicalhostname create -g has-rg sam1mds
```

```
[qfs1mds-node1]root@solaris:~#
```

5. 向资源组中添加 Oracle HSM 文件系统资源。使用命令 *clresource create -g groupname -t SUNW.qfs -x QFSFileSystem=mount-point*，其中：
   - *groupname* 是您为 HA-SAM 资源组选择的名称。
   - *SUNW.qfs* 是支持 QFS 文件系统元数据服务器的故障转移的 Solaris Cluster 资源类型。
   - *mount-point* 是群集中文件系统的挂载点，即并非直接位于系统根目录下的子目录。

     将共享 QFS 文件系统直接挂载到根目录下可能会在使用 *SUNW.qfs* 资源类型时导致故障转移问题。
   - *resource-name* 是您为资源本身选择的名称。

   在示例中，我们在 *has-rg* 资源组中创建名为 *has-qfs* 的 *SUNW.qfs* 类型的资源。我们将 *SUNW.qfs* 扩展属性 *QFSFileSystem* 设置为 */global/ha-sam/sam1* 挂载点。我们将标准属性 *Resource_dependencies* 设置为 *sam1mds*，这是表示活动元数据服务器的虚拟主机名（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）：

```
[sam1mds-node1]root@solaris:~# clresource create -g has-rg -t SUNW.qfs /
-x QFSFileSystem=/global/ha-sam/sam1 -y Resource_dependencies=sam1mds has-qfs
[sam1mds-node1]root@solaris:~#
```

6. 接下来，配置 Oracle Hierarchical Storage Manager 应用程序的故障转移。

配置 Oracle Hierarchical Storage Manager 应用程序的故障转移

通过创建 Oracle HSM *SUNW.hasam* 资源来配置 Oracle Hierarchical Storage Manager 应用程序的故障转移。该资源类型协调按顺序执行的 Oracle HSM 关闭和重新启动过程。

要配置 Oracle HSM 应用程序的故障转移，请执行如下操作：

1. 以 *root* 用户身份登录到 HA-SAM 群集的主群集节点。

   在示例中，主节点是 *sam1mds-node1*：

```
[sam1mds-node1]root@solaris:~#
```

2. 为 Solaris Cluster 软件定义资源类型 *SUNW.hasam*。使用命令 *clresourcetype register SUNW.hasam*。

```
[sam1mds-node1]root@solaris:~# clresourcetype register SUNW.hasam
[sam1mds-node1]root@solaris:~#
```

3. 向资源组添加 Oracle HSM *SUNW.hasam* 资源。使用命令 *clresource create -g groupname -t SUNW.hasam -x QFSName=fs-name -x CatalogFileSystem=mount-point resource-name*，其中：
   - *groupname* 是您为 HA-SAM 资源组选择的名称。
   - *SUNW.hasam* 是支持 Oracle Hierarchical Storage Manager 应用程序故障转移的 Solaris Cluster 资源类型。
   - *mount-point* 是存放 Oracle HSM 归档目录的全局文件系统的挂载点。
   - *resource-name* 是您为资源本身选择的名称。

   在示例中，我们在 *has-rg* 资源组中创建名为 *has-sam* 的 *SUNW.hasam* 类型的资源。我们将 *SUNW.hasam* 扩展属性 *QFSName* 设置为在 *mcf* 文件中指定的 QFS 文件系统名称 *sam1*。我们将 *SUNW.hasam* 扩展属性 *CatalogFileSystem* 设置为 */global/hasam_cfg* 挂载点。

```
[sam1mds-node1]root@solaris:~# clresource create -g has-rg -t SUNW.hasam /
-x QFSName=sam1 -x CatalogFileSystem=/global/hasam_cfg has-sam
[sam1mds-node1]root@solaris:~#
```

4. 接下来，定义 HA-SAM 解决方案的群集资源依赖性。

定义 HA-SAM 解决方案的群集资源依赖性

1. 以 *root* 用户身份登录到 HA-SAM 群集的主群集节点。

   在示例中，主节点是 *sam1mds-node1*：

```
[sam1mds-node1]root@solaris:~#
```

2. 除非高可用性本地文件系统可用，否则 QFS 文件系统不应启动。因此使 *SUNW.qfs* 资源依赖于 *SUNW.HAStoragePlus* 资源。使用 Solaris Cluster 命令 *clresource set -p Resource_dependencies=dependency resource-name*，其中：
   - *dependency* 是 *SUNW.HAStoragePlus* 资源的名称。
   - *resource-name* 是 *SUNW.qfs* 资源的名称。

   在本示例中，我们使 *SUNW.qfs* 资源依赖于 *SUNW.HAStoragePlus* 资源 *has-cfg*（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）：

```
[sam1mds-node1]root@solaris:~# clresource set /
-p Resource_dependencies=has-cfg has-qfs
[sam1mds-node1]root@solaris:~#
```

3. 除非 QFS 活动元数据服务器联机，否则群集不应使虚拟主机名可用。因此使虚拟主机名依赖于 *SUNW.qfs* 资源。使用 Solaris Cluster 命令 *clresource set -p Resource_dependencies=virtualMDS resource-name*，其中：
   - *virtualMDS* 是表示活动 Oracle HSM 元数据服务器的虚拟主机名。
   - *resource-name* 是 *SUNW.qfs* 资源的名称。

   在本示例中，在我们设置 *SUNW.qfs* 资源时创建的虚拟主机名是 *sam1mds*。该资源本身名为 *has-qfs*（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）：

```
[sam1mds-node1]root@solaris:~# clresource set /
-p Resource_dependencies=sam1mds has-qfs
[sam1mds-node1]root@solaris:~#
```

4. 接下来，使 HA-SAM 资源组联机并测试配置。

使 HA-SAM 资源组联机并测试配置

1. 以 *root* 用户身份登录到 HA-SAM 群集的主群集节点。

   在示例中，主节点是 *sam1mds-node1*：

```
[sam1mds-node1]root@solaris:~#
```

2. 使资源组联机。使用 Solaris Cluster 命令 *clresourcegroup manage groupname*, and *clresourcegroup online -emM groupname*，其中 *groupname* 是 HA-SAM 资源组的名称。

   在示例中，我们将 *has-rg* 资源组联机：

```
[sam1mds-node1]root@solaris:~# clresourcegroup manage has-rg
[sam1mds-node1]root@solaris:~# clresourcegroup online -emM has-rg
[sam1mds-node1]root@solaris:~#
```

3. 确保 HA-SAM 资源组处于联机状态。使用 Solaris Cluster *clresourcegroup status* 命令。

   在示例中，*has-rg* 资源组在主节点 *sam1mds-node1* 上处于 *online* 状态：

```
[sam1mds-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
has-rg      sam1mds-node1   No          Online
            sam1mds-node2   No            Offline
[sam1mds-node1]root@solaris:~#
```

4. 接下来，确保资源组正确故障转移。将资源组移到辅助节点。使用 Solaris Cluster 命令 *clresourcegroup switch -n node2 groupname*，其中 *node2* 是辅助节点的名称，*groupname* 是您为 HA-SAM 资源组选择的名称。然后，使用 *clresourcegroup status* 检查结果。

   在示例中，我们将 *has-rg* 资源组移动到 *sam1mds-node2* 并确认资源组在指定节点上处于联机状态：

```
[sam1mds-node1]root@solaris:~# clresourcegroup switch -n sam1mds-node2 has-rg
[sam1mds-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
has-rg      sam1mds-node1   No          Offline
            sam1mds-node2   No          Online
[sam1mds-node1]root@solaris:~#
```

5. 将资源组移回到主节点。使用 Solaris Cluster 命令 *clresourcegroup switch -n node1 groupname*，其中 *node1* 是主节点的名称，*groupname* 是您为 HA-SAM 资源组选择的名称。然后，使用 *clresourcegroup status* 检查结果。

   在示例中，我们将 *has-rg* 资源组成功移回到 *sam1mds-node1*：

```
[sam1mds-node1]root@solaris:~# clresourcegroup switch -n sam1mds-node1 has-rg
[sam1mds-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
has-rg      sam1mds-node1   No          Online
            sam1mds-node2   No          Offline
[sam1mds-node1]root@solaris:~#
```

6. 如果需要，立即配置高可用性网络文件系统 (High-Availability Network File System, HA-NFS) 共享。

   Oracle Solaris Cluster 联机文档库包含的《*Oracle Solaris Cluster Data Service for Network File System (NFS) Guide*》中提供了设置 HA-NFS 的详细步骤。
7. 如果计划使用边带数据库功能，请转至第 10 章 配置报告数据库。
8. 否则，请转至第 11 章 配置通知和日志记录。

高可用性 QFS 共享文件系统和 Oracle RAC

在 Solaris Cluster-Oracle Real Application Cluster (SC-RAC) 配置中，Solaris Cluster 软件将 QFS 共享文件系统作为同时还托管 Oracle 数据库和 Oracle Real Application

Cluster (RAC) 软件的节点上挂载的 *SUNW.qfs* 资源进行管理。所有节点都配置为 QFS 服务器，其中一个为活动元数据服务器，其他的则为潜在元数据服务器。如果活动元数据服务器节点发生故障，则 Solaris Cluster 软件将自动激活运行正常的节点上的潜在元数据服务器并启动故障转移。通过 Oracle RAC 协调 I/O，QFS 文件系统是共享文件系统并且已挂载到所有节点上。因此仍可无中断地访问数据。

在 SC-RAC 配置中，RAC 软件协调 I/O 请求、分配工作负荷，并且为群集节点上运行的多个 Oracle 数据库实例维护一组一致的数据库文件。由于文件系统完整性在 RAC 下得到保证，因此 QFS 潜在元数据服务器可以作为共享文件系统的客户机执行 I/O。有关更多信息，请参见 *Oracle Solaris Cluster* 联机文档库中适用于 Oracle Real Application Clusters 的 Oracle Solaris Cluster 数据服务文档。

要配置 SC-RAC 文件系统，请执行以下任务：

- 在所有 SC-RAC 群集节点上创建 QFS 共享文件系统 Hosts 文件
- 在 HA-COTC 群集外的 QFS 服务器和客户机上创建本地 Hosts 文件
- 在主 SC-RAC 群集节点上配置活动 QFS 元数据服务器 或使用软件 RAID 存储在 SC-RAC 节点上配置 QFS 元数据服务器
- 在其余的 SC-RAC 群集节点上配置潜在 QFS 元数据服务器
- 配置 SC-RAC 元数据服务器的故障转移
- 如果需要，按“使用 NFS 和 SMB/CIFS 从多台主机访问文件系统”中所述配置网络文件系统 (Network File System, NFS) 共享。不支持高可用性 NFS (High-Availability NFS, HA-NFS)。

在所有 SC-RAC 群集节点上创建 QFS 共享文件系统 Hosts 文件

在 QFS 共享文件系统中，您必须在元数据服务器上配置 hosts 文件，以便所有主机都能访问文件系统的元数据。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. 以 *root* 用户身份登录到 SC-RAC 群集的主群集节点。

   在示例中，主节点是 *qfs1rac-node1*：

```
[qfs1rac-node1]root@solaris:~#
```

2. 显示群集配置。使用 */usr/global/bin/cluster show* 命令。在输出中，找到每个 *Node Name* 的记录，然后记下 *privatehostname* 和 *Transport Adapter* 名称，以及每个网络适配器的 *ip_address* 属性。

   在示例中，每个节点都有两个网络接口，*qfe3* 和 *hme0*：

   - *hme0* 适配器具有群集用于节点间内部通信的专用网络的 IP 地址。Solaris Cluster 软件会分配与每个专用地址相对应的 *privatehostname*。

默认情况下，主节点和辅助节点的专用主机名分别为 *clusternode1-priv* 和 *clusternode2-priv*。

- *qfe3* 适配器具有公共 IP 地址和公共主机名（即 *qfs1rac-node1* 和 *qfs1rac-node2*），群集使用它们来进行数据传输。

请注意，使用省略号 (*...*) 标记简化了显示内容：

```
[qfs1rac-node1]root@solaris:~# cluster show
...
  === Cluster Nodes ===
  Node Name:                                    qfs1rac-node1...
    privatehostname:                               clusternode1-priv...
    Transport Adapter List:                        qfe3, hme0...
    Transport Adapter:                          qfe3...
      Adapter Property(ip_address):                172.16.0.12...
    Transport Adapter:                          hme0...
      Adapter Property(ip_address):                10.0.0.129...
  Node Name:                                    qfs1rac-node2...
    privatehostname:                               clusternode2-priv...
    Transport Adapter List:                        qfe3, hme0...
      Adapter Property(ip_address):                172.16.0.13...
    Transport Adapter:                          hme0
      Adapter Property(ip_address):                10.0.0.122...
  Node Name:                                    qfs1rac-node3...
    privatehostname:                               clusternod3-priv...
    Transport Adapter List:                        qfe3, hme0...
      Adapter Property(ip_address):                172.16.0.33...
    Transport Adapter:                          hme0
      Adapter Property(ip_address):                10.0.0.092
```

3. 使用文本编辑器，创建文件 */etc/opt/SUNWsamfs/hosts.family-set-name*，其中 *family-set-name* 是 */etc/opt/SUNWsamfs/mcf* 文件分配给文件系统设备的系列集名称。

   在示例中，使用 *vi* 文本编辑器创建文件 *hosts.qfs1rac*。我们添加一些可选标题以显示主机表中的列，每行以井号 (*#*) 开头表示该行为注释：

```
[qfs1rac-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.qfs1rac
# /etc/opt/SUNWsamfs/hosts.qfs1rac
#                                              Server   On/  Additional
#Host Name     Network Interface                Ordinal  Off  Parameters
#------------  -------------------------------  -------  ---  ----------
```

4. 在表的第一列中，输入主元数据服务器节点和辅助元数据服务器节点的主机名，后跟一些空格。将每个条目放在单独一行上。

   在 hosts 文件中，行即为行（记录），空格为列（字段）分隔符。在本示例中，Host Name 列的前两行列出了群集节点 *qfs1rac-node1*、*qfs1rac-node2* 和 *qfs1rac-node3* 的主机名。

```
#                                          Server   On/  Additional
#Host Name     Network Interface            Ordinal  Off  Parameters
#------------  -------------------------------  -------  ---  ----------
qfs1rac-node1
qfs1rac-node2
qfs1rac-node3
```

5. 在每行的第二列，开始提供主机 *Host Name* 的 *Network Interface* 信息。输入各个 SC-RAC 群集节点的 Solaris Cluster 专用主机名或专用网络地址，后跟一个逗号。

   SC-RAC 服务器节点使用专用主机名在高可用性群集内进行服务器间的通信。在示例中，使用专用主机名 *clusternode1-priv*、*clusternode2-priv* 和 *clusternode3-priv*，这些是 Solaris Cluster 软件分配的默认名称：

```
#                                          Server   On/  Additional
#Host Name     Network Interface            Ordinal  Off  Parameters
#------------  -------------------------------  -------  ---  ----------
qfs1rac-node1  clusternode1-priv,
qfs1rac-node2  clusternode2-priv,
qfs1rac-node3  clusternode3-priv,
```

6. 在每行第二列中的逗号后面，输入活动元数据服务器的公共主机名，后跟空格。

   SC-RAC 服务器节点使用公共数据网络与客户机通信，所有这些客户机都位于在群集之外。由于活动元数据服务器的 IP 地址和主机名会在故障转移期间发生更改（例如，从 *qfs1rac-node1* 更改为 *qfs1rac-node2*），因此我们使用虚拟主机名 *qfs1rac-mds* 表示活动服务器。稍后，我们将配置 Solaris Cluster 软件，以便始终将 *qfs1rac-mds* 的请求路由到当前托管活动元数据服务器的节点：

```
#                                          Server   On/  Additional
#Host Name     Network Interface            Ordinal  Off  Parameters
#------------  -------------------------------  -------  ---  ----------
qfs1rac-node1  clusternode1-priv,qfs1rac-mds
qfs1rac-node2  clusternode2-priv,qfs1rac-mds
qfs1rac-node3  clusternode3-priv,qfs1rac-mds
```

7. 在每行的第三列中，输入服务器的序号（*1* 表示活动元数据服务器，*2* 表示潜在元数据服务器），后跟空格。

   在示例中，主节点 *qfs1rac-node1* 是活动元数据服务器。因此其序号为 *1*。第二个节点 *qfs1rac-node2* 的序号为 *2*，依此类推：

```
#                                          Server   On/  Additional
#Host Name     Network Interface            Ordinal  Off  Parameters
#------------  -------------------------------  -------  ---  ----------
qfs1rac-node1  clusternode1-priv,qfs1rac-mds     1
qfs1rac-node2  clusternode2-priv,qfs1rac-mds     2
qfs1rac-node3  clusternode3-priv,qfs1rac-mds     3
```

8. 在每行的第四列中，输入 *0*（零），后跟空格。

   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 *samsharefs* 手册页）。

```
#                                          Server   On/  Additional
#Host Name     Network Interface            Ordinal  Off  Parameters
#------------  -------------------------------  -------  ---  ----------
qfs1rac-node1  clusternode1-priv,qfs1rac-mds     1        0
qfs1rac-node2  clusternode2-priv,qfs1rac-mds     2        0
qfs1rac-node3  clusternode3-priv,qfs1rac-mds     3        0
```

9. 在主节点行的第五列中，输入关键字 *server*。保存文件并关闭编辑器。

   server 关键字用于标识默认的活动元数据服务器：

```
#                                          Server   On/  Additional
#Host Name     Network Interface            Ordinal  Off  Parameters
#------------  -------------------------------  -------  ---  ----------
qfs1rac-node1  clusternode1-priv,qfs1rac-mds     1        0    server
qfs1rac-node2  clusternode2-priv,qfs1rac-mds     2        0
qfs1rac-node3  clusternode3-priv,qfs1rac-mds     2        0
:wq
[qfs1rac-node1]root@solaris:~#
```

10. 将全局 */etc/opt/SUNWsamfs/hosts.family-set-name* 文件的副本放置在 SC-RAC 群集的每个节点上。
11. 现在，在主 SC-RAC 群集节点上配置活动 QFS 元数据服务器。

在主 SC-RAC 群集节点上配置活动 QFS 元数据服务器

1. 选择将同时充当 SC-RAC 群集的主节点和 QFS 共享文件系统的活动元数据服务器的群集节点。以 *root* 用户身份登录。

   在示例中，主节点是 *qfs1rac-node1*：

```
[qfs1rac-node1]root@solaris:~#
```

2. 选择要用于 QFS 文件系统的全局存储设备。使用命令 */usr/global/bin/cldevice list -v*。

   Solaris Cluster 软件向连接到群集节点的所有设备分配唯一的设备标识符 (Device Identifier, DID)。全局设备可以从群集中的所有节点进行访问，而本地设备仅可以从挂载它们的主机进行访问。全局设备在故障转移后仍然可以访问。本地设备则不然。

   在示例中，请注意 *d1*、*d2*、*d6*、*d7* 和 *d8* 设备从所有节点都无法访问。所以在配置高可用性 QFS 共享文件系统时，我们从 *d3*、*d4* 和 *d5* 设备中进行选择：

```
[qfs1rac-node1]root@solaris:~# cldevice list -v
DID Device          Full Device Path
----------          ----------------
d1                  qfs1rac-node1:/dev/rdsk/c0t0d0
d2                  qfs1rac-node1:/dev/rdsk/c0t6d0
d3                  qfs1rac-node1:/dev/rdsk/c1t1d0
d3                  qfs1rac-node2:/dev/rdsk/c1t1d0
d3                  qfs1rac-node3:/dev/rdsk/c1t1d0
d4                  qfs1rac-node1:/dev/rdsk/c1t2d0
d4                  qfs1rac-node2:/dev/rdsk/c1t2d0
d4                  qfs1rac-node3:/dev/rdsk/c1t2d0
d5                  qfs1rac-node1:/dev/rdsk/c1t3d0
d5                  qfs1rac-node2:/dev/rdsk/c1t3d0
d5                  qfs1rac-node3:/dev/rdsk/c1t3d0
d6                  qfs1rac-node2:/dev/rdsk/c0t0d0
d7                  qfs1rac-node2:/dev/rdsk/c0t1d0
d8                  qfs1rac-node3:/dev/rdsk/c0t1d0
```

3. 创建使用 *mr* 数据设备的共享高性能 *ma* 文件系统。在文本编辑器中，打开 */etc/opt/SUNWsamfs/mcf* 文件。

   在示例中，我们配置文件系统 *qfs1rac*。我们将设备 *d3* 配置为元数据设备（设备类型 *mm*），并将 *d4* 和 *d5* 用作数据设备（设备类型 *mr*）：

```
[qfs1rac-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family   Device   Additional
```

```
# Identifier          Ordinal    Type       Set      State    Parameters
#------------------   ---------  ---------  -------  ------   --------------
qfs1rac               100        ma         qfs1rac  -
 /dev/did/dsk/d3s0    101        mm         qfs1rac  -
 /dev/did/dsk/d4s0    102        mr         qfs1rac  -
 /dev/did/dsk/d5s0    103        mr         qfs1rac  -
...
```

4. 在 */etc/opt/SUNWsamfs/mcf* 文件的文件系统条目的 *Additional Parameters* 列中输入 *shared* 参数。保存文件。

```
[qfs1rac-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family   Device   Additional
# Identifier          Ordinal    Type       Set      State    Parameters
#------------------   ---------  ---------  -------  ------   --------------
qfs1rac               100        ma         qfs1rac  -        shared
 /dev/did/dsk/d3s0     101        mm         qfs1rac  -
 /dev/did/dsk/d4s0     102        mr         qfs1rac  -
 /dev/did/dsk/d5s0     103        mr         qfs1rac  -
...
:wq
[qfs1rac-node1]root@solaris:~#
```

5. 检查 *mcf* 文件中是否存在错误。使用命令 */opt/SUNWsamfs/sbin/sam-fsd* 并更正发现的任何错误。

   sam-fsd 命令会读取 Oracle HSM 配置文件并初始化文件系统。如果遇到错误，该命令将停止。在示例中，检查主机 *qfs1rac-node1* 上的 *mcf* 文件：

```
[qfs1rac-node1]root@solaris:~# sam-fsd
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
[qfs1rac-node1]root@solaris:~#
```

6. 创建文件系统。使用命令 */opt/SUNWsamfs/sbin/sammkfs -S family-set-name*，其中 *family-set-name* 是文件系统的设备标识符。

   sammkfs 命令读取 *hosts.family-set-name* 和 *mcf* 文件，并创建具有指定属性的共享文件系统。

```
[qfs1rac-node1]root@solaris:~# sammkfs -S qfs1rac
Building 'qfs1rac' will destroy the contents of devices:
```

```
  ...
Do you wish to continue? [y/N]yes ...
[qfs1rac-node1]root@solaris:~#
```

7. 在文本编辑器中打开操作系统的 */etc/vfstab* 文件，并为新的文件系统添加一行。在第一列中输入文件系统名称，然后输入几个空格，在第二列中输入一个连字符，然后输入更多的空格。

   在示例中，使用 *vi* 文本编辑器。我们为 *qfs1rac* 文件系统添加了一行。连字符阻止操作系统尝试使用 UFS 工具检查文件系统的完整性：

```
[qfs1rac-node1]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       -
...
qfs1rac    -
```

8. 在 */etc/vfstab* 文件的第三列中，输入相对于群集的文件系统挂载点。指定未直接位于系统根目录下的子目录。

   将共享 QFS 文件系统直接挂载到根目录下可能会在使用 *SUNW.qfs* 资源类型时导致故障转移问题。在示例中，*qfs1rac* 文件系统的挂载点为 */global/sc-rac/qfs1rac*：

```
#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       -
...
qfs1rac    -        /global/sc-rac/qfs1rac
```

9. 在第四列中输入文件系统类型 *samfs*，在第五列中输入 -（连字符），在第六列中输入 *no*。

```
#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       -
...
qfs1rac    -        /global/sc-rac/qfs1rac   samfs   -     no
:wq
[qfs1rac-node1]root@solaris:~#
```

10. 在 */etc/vfstab* 文件的第七列中，输入下面列出的挂载选项。然后保存文件并关闭编辑器。

    建议对 SC-RAC 群集配置使用以下挂载选项。它们可以在 */etc/vfstab* 中指定，如果更方便，也可以在文件 */etc/opt/SUNWsamfs/samfs.cmd* 中指定：

    - *shared*
    - *stripe=1*
    - *sync_meta=1*
    - *mh_write*
    - *qwrite*
    - *forcedirectio*
    - *notrace*
    - *rdlease=300*
    - *wrlease=300*
    - *aplease=300*

    在示例中，简化了此列表以适应页面布局：

```
#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      -
...
qfs1rac   -       /global/sc-rac/qfs1rac samfs  -    no      shared,...=300
:wq
[qfs1rac-node1]root@solaris:~#
```

11. 创建高可用性共享文件系统的挂载点。

```
[qfs1rac-node1]root@solaris:~# mkdir -p /global/sc-rac/qfs1rac
[qfs1rac-node1]root@solaris:~#
```

12. 在主节点上挂载高可用性共享文件系统。

```
[qfs1rac-node1]root@solaris:~# mount /global/sc-rac/qfs1rac
```

```
[qfs1rac-node1]root@solaris:~#
```

13. 接下来，在其余的 SC-RAC 群集节点上配置潜在 QFS 元数据服务器。

在其余的 SC-RAC 群集节点上配置潜在 QFS 元数据服务器

群集的其余节点充当潜在元数据服务器。潜在元数据服务器是可以访问元数据设备的主机，可以承担元数据服务器的职责。因此，如果主节点上的活动元数据服务器发生故障，则 Solaris Cluster 软件可以故障转移到辅助节点并激活潜在元数据服务器。

对于 SC-RAC 群集中其余的每个节点，请执行如下操作：

1. 以 *root* 用户身份登录节点。

   在示例中，当前节点是 *qfs1rac-node2*：

```
[qfs1rac-node2]root@solaris:~#
```

2. 将 */etc/opt/SUNWsamfs/mcf* 文件从主节点复制到当前节点中。
3. 检查 *mcf* 文件中是否存在错误。运行命令 */opt/SUNWsamfs/sbin/sam-fsd* 并更正发现的任何错误。

   sam-fsd 命令会读取 Oracle HSM 配置文件并初始化文件系统。如果遇到错误，该命令将停止。在示例中，检查主机 *qfs1rac-node2* 上的 *mcf* 文件：

```
[qfs1rac-node2]root@solaris:~# sam-fsd
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
[qfs1rac-node2]root@solaris:~#
```

4. 在文本编辑器中打开操作系统的 */etc/vfstab* 文件，并为新的文件系统添加一行。

   在示例中，使用 *vi* 编辑器：

```
[qfs1rac-node2]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       -
```

```
...
qfs1rac    -        /global/sc-rac/qfs1rac samfs   -     no
```

5. 在 */etc/vfstab* 文件的第七列中，输入下面列出的挂载选项。然后保存文件并关闭编辑器。

   建议对 SC-RAC 群集配置使用以下挂载选项。它们可以在 */etc/vfstab* 中指定，如果更方便，也可以在文件 */etc/opt/SUNWsamfs/samfs.cmd* 中指定：

   - *shared*
   - *stripe=1*
   - *sync_meta=1*
   - *mh_write*
   - *qwrite*
   - *forcedirectio*
   - *notrace*
   - *rdlease=300*
   - *wrlease=300*
   - *aplease=300*

   在示例中，简化了此列表以适应页面布局：

```
#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      -
...
qfs1rac   -       /global/sc-rac/qfs1rac samfs   -     no      shared,...=300
:wq
[qfs1rac-node2]root@solaris:~#
```

6. 在辅助节点上创建高可用性共享文件系统的挂载点。

```
[qfs1rac-node2]root@solaris:~# mkdir -p /global/sc-rac/qfs1rac
[qfs1rac-node2]root@solaris:~#
```

7. 在辅助节点上挂载高可用性共享文件系统。

```
[qfs1rac-node2]root@solaris:~# mount /global/sc-rac/qfs1rac
[qfs1rac-node2]root@solaris:~#
```

8. 现在，配置 SC-RAC 元数据服务器的故障转移。

配置 SC-RAC 元数据服务器的故障转移

当您在 Solaris Cluster 软件管理的群集中托管 Oracle HSM 共享文件系统时，可以通过创建 *SUNW.qfs* 群集资源来配置元数据服务器的故障转移，该群集资源为 Oracle HSM 软件定义的一种资源类型（有关详细信息，请参见 *SUNW.qfs* 手册页）。要创建并配置 SC-RAC 配置的资源，请执行如下操作：

1. 以 *root* 用户身份登录到 SC-RAC 群集的主节点。

   在示例中，主节点是 *qfs1rac-node1*：

```
[qfs1rac-node1]root@solaris:~#
```

2. 为 Solaris Cluster 软件定义 QFS 资源类型 *SUNW.qfs*。使用命令 *clresourcetype registerSUNW.qfs*。

```
[qfs1rac-node1]root@solaris:~# clresourcetype registerSUNW.qfs
[qfs1rac-node1]root@solaris:~#
```

3. 如果注册因找不到注册文件而失败，则将 */opt/SUNWsamfs/sc/etc/* 目录的符号链接放置到 Solaris Cluster 存放资源类型注册文件的 */opt/cluster/lib/rgm/rtreg/* 目录中。

   您在安装 Oracle HSM 软件之前未安装 Oracle Solaris Cluster 软件。通常，Oracle HSM 会在安装期间检测到 Solaris Cluster 时自动提供 *SUNW.qfs* 注册文件的位置。因此您需要手动创建一个链接。

```
[qfs1rac-node1]root@solaris:~# cd /opt/cluster/lib/rgm/rtreg/
[qfs1rac-node1]root@solaris:~# ln -s /opt/SUNWsamfs/sc/etc/SUNW.qfs SUNW.qfs
[qfs1rac-node1]root@solaris:~#
```

4. 创建 QFS 元数据服务器的资源组。使用 Solaris Cluster 命令 *clresourcegroup create -n node-list group-name*，其中 *node-list* 是逗号分隔的群集节点列表，*group-name* 是我们想要用于资源组的名称。

   在示例中，使用 SC-RAC 服务器节点作为成员创建资源组 *qfsracrg*（请注意，下面的命令是作为单行输入的－使用反斜杠字符对换行符进行转义）：

```
[qfs1rac-node1]root@solaris:~# clresourcegroup create /
-n qfs1rac-node1,qfs1rac-node2 qfsracrg
[qfs1rac-node1]root@solaris:~#
```

5. 在新的资源组中，设置活动元数据服务器的虚拟主机名。使用 Solaris Cluster 命令 *clreslogicalhostname create -g group-name*，其中 *group-name* 是 QFS 资源组的名称，*virtualMDS* 是虚拟主机名。

使用您在 hosts 文件中用于共享文件系统的相同虚拟主机名。在本示例中，我们在 *qfsracrg* 资源组中创建虚拟主机 *qfs1rac-mds*（请注意，下面的命令是作为单行输入的－使用反斜杠字符对换行符进行转义）：

```
[qfs1rac-node1]root@solaris:~# clreslogicalhostname create /
-g qfsracrg qfs1rac-mds
[qfs1rac-node1]root@solaris:~#
```

6. 向资源组中添加 QFS 文件系统资源。使用命令 *clresource create -g group-name -t SUNW.qfs -x QFSFileSystem=mount-point -y Resource_dependencies=virtualMDS resource-name*，其中：
   - *group-name* 是 QFS 资源组的名称。
   - *mount-point* 是群集中文件系统的挂载点，即并非直接位于系统根目录下的子目录。

     将共享 QFS 文件系统直接挂载到根目录下可能会在使用 *SUNW.qfs* 资源类型时导致故障转移问题。
   - *virtualMDS* 是活动元数据服务器的虚拟主机名。
   - *resource-name* 是您想要赋予资源的名称。

   在示例中，我们在 *qfsracrg* 资源组中创建名为 *scrac* 的 *SUNW.qfs* 类型的资源。我们将 *SUNW.qfs* 扩展属性 *QFSFileSystem* 设置为 */global/sc-rac/qfs1rac* 挂载点。我们将标准属性 *Resource_dependencies* 设置为活动元数据服务器 *qfs1rac-mds* 的虚拟主机（请注意，下面的命令是作为单行输入的－使用反斜杠字符对换行符进行转义）：

```
[qfs1rac-node1]root@solaris:~# create -g qfsracrg -t SUNW.qfs /
-x QFSFileSystem=/global/sc-rac/qfs1rac /
-y Resource_dependencies=qfs1rac-mds scrac
[qfs1rac-node1]root@solaris:~#
```

7. 使资源组联机。使用 Solaris Cluster 命令 *clresourcegroup manage group-name* and *clresourcegroup online -emM group-name*，其中 *group-name* 是 QFS 资源组的名称。

   在示例中，我们将 *qfsracrg* 资源组联机：

```
[qfs1rac-node1]root@solaris:~# clresourcegroup manage qfsracrg
[qfs1rac-node1]root@solaris:~# clresourcegroup online -emM qfsracrg
[qfs1rac-node1]root@solaris:~#
```

8. 确保 QFS 资源组处于联机状态。使用 Solaris Cluster 命令*clresourcegroup status*。

在示例中，*qfsracrg* 资源组在主节点 *qfs1rac-node1* 上处于 *online* 状态：

```
[qfs1rac-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
qfsracrg    qfs1rac-node1   No          Online
            qfs1rac-node2   No          Offline
            qfs1rac-node3   No          Offline
[qfs1rac-node1]root@solaris:~#
```

9. 确保资源组正确故障转移。将资源组移到辅助节点。使用 Solaris Cluster 命令 *clresourcegroup switch -n node2 group-name*，其中 *node2* 是辅助节点的名称，*group-name* 是您为 HA-SAM 资源组选择的名称。然后，使用 *clresourcegroup status* 检查结果。

   在示例中，我们将 *qfsracrg* 资源组移到 *qfs1rac-node2* 和 *qfs1rac-node3*，并确认资源组在指定的节点上处于联机状态：

```
[qfs1rac-node1]root@solaris:~# clresourcegroup switch -n qfs1rac-node2 qfsracrg
[qfs1rac-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
qfsracrg    qfs1rac-node1   No          Offline
            qfs1rac-node2   No          Online
            qfs1rac-node3   No          Offline
[qfs1rac-node1]root@solaris:~# clresourcegroup switch -n qfs1rac-node3 qfsracrg
[qfs1rac-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
qfsracrg    qfs1rac-node1   No          Offline
            qfs1rac-node2   No          Offline
            qfs1rac-node3   No          Online
[qfs1rac-node1]root@solaris:~#
```

10. 将资源组移回到主节点。使用 Solaris Cluster 命令 *clresourcegroup switch -n node1 group-name*，其中 *node1* 是主节点的名称，*group-name* 是您为 HA-SAM 资源组选择的名称。然后，使用 *clresourcegroup status* 检查结果。

    在示例中，我们将 *qfsracrg* 资源组成功移回到 *qfs1rac-node1*：

```
[qfs1rac-node1]root@solaris:~# clresourcegroup switch -n qfs1rac-node1 qfsracrg
```

```
[qfs1rac-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
samr        qfs1rac-node1   No          Online
            qfs1rac-node2   No          Offline
            qfs1rac-node3   No          Offline
[qfs1rac-node1]root@solaris:~#
```

11. 如果计划使用边带数据库功能，请转至第 10 章 配置报告数据库。
12. 否则，请转至第 11 章 配置通知和日志记录。

使用软件 RAID 存储在 SC-RAC 节点上配置 QFS 元数据服务器

高可用性文件系统必须在冗余主存储设备上存储数据和元数据。冗余磁盘阵列硬件可以通过将 RAID-1 或 RAID-10 用于存储元数据以及将 RAID-5 用于存储数据来提供此冗余。但是，如果您需要将普通的双端口 SCSI 磁盘设备或 JBOD（*just a bunch of disk*，简单磁盘捆绑）阵列用作主存储，则需要在软件中提供所需的冗余。

出于此原因，SC-RAC 配置支持基于 Oracle Solaris Volume Manager (SVM) 多属主磁盘集的软件 RAID 配置。此部分介绍设置 SC-RAC 文件系统配置的此变体时需要执行的基本步骤。

请注意，您应该将 Solaris Volume Manager 仅用于管理冗余存储阵列。请勿串联不同设备上的存储。这么做会导致将 I/O 低效分发到组件设备并使 QFS 文件系统性能降级。

执行以下任务：

- 在 Solaris 11+ 上安装 Solaris Volume Manager
- 创建 Solaris Volume Manager 多属主磁盘组
- 创建 QFS 数据和元数据的镜像卷
- 使用镜像卷在 SC-RAC 群集上创建 QFS 共享文件系统

在 Solaris 11+ 上安装 Solaris Volume Manager

从 Solaris 11 开始，Solaris Volume Manager (SVM) 不再包含在 Solaris 中。但 Solaris Cluster 4 软件继续支持 Solaris Volume Manager。因此要使用该软件，您必须下载并安装 Solaris 10 9/10 发行版随附的版本。对于群集中的每个节点，请执行如下操作：

1. 以 *root* 用户身份登录节点。

   在下面的示例中，我们使用 Solaris 映像包管理系统 (Image Packaging System, IPS) 配置群集节点 *qfs2rac-node1*：

```
[qfs2rac-node1]root@solaris:~#
```

2. 检查本地可用的 Solaris Volume Manager (SVM) 软件包。使用命令 *pkg info svm*。

```
[qfs2rac-node1]root@solaris:~# pkg info svm
pkg: info: no packages matching the following patterns you specified are
installed on the system.  Try specifying -r to query remotely:
        svm
[qfs2rac-node1]root@solaris:~#
```

3. 如果本地找不到软件包，则检查 Solaris 映像包管理系统 (Image Packaging System, IPS) 系统信息库。使用命令 *pkg -r svm*。

```
[qfs2rac-node1]root@solaris:~# pkg -r svm
          Name: storage/svm
       Summary: Solaris Volume Manager
   Description: Solaris Volume Manager commands
      Category: System/Core
         State: Not installed
     Publisher: solaris
       Version: 0.5.11
 Build Release: 5.11
        Branch: 0.175.0.0.0.2.1
Packaging Date: October 19, 2011 06:42:14 AM
          Size: 3.48 MB
          FMRI: pkg://solaris/storage/svm@0.5.11,5.11-0.175.0.0.0.2.1:20111019T064214Z
[qfs2rac-node1]root@solaris:~#
```

4. 安装软件包。使用命令 *pkg install storage/svm*：

```
[qfs2rac-node1]root@solaris:~# pkg install storage/svm
           Packages to install:   1
       Create boot environment:  No
Create backup boot environment: Yes
            Services to change:   1
DOWNLOAD      PKGS       FILES    XFER (MB)
Completed      1/1     104/104      1.6/1.6
PHASE            ACTIONS
Install Phase    168/168
PHASEITEMS
Package State Update Phase         1/1
Image State Update Phase           2/2
[qfs2rac-node1]root@solaris:~#
```

5. 当安装完成后，检查 *metadb* 的位置。使用命令 *which metadb*。

```
[qfs2rac-node1]root@solaris:~# which metadb
/usr/sbin/metadb
[qfs2rac-node1]root@solaris:~#
```

6. 检查安装。使用命令 *metadb*。

```
[qfs2rac-node1]root@solaris:~# metadb
[qfs2rac-node1]root@solaris:~#
```

7. 如果 *metadb* 返回错误，则查看 *kernel/drv/md.conf* 文件是否存在。

```
[qfs2rac-node1]root@solaris:~# metadb
metadb: <HOST>: /dev/md/admin: No such file or directory
[qfs2rac-node1]root@solaris:~# ls -l /kernel/drv/md.conf
-rw-r--r--   1 root     sys          295 Apr 26 15:07 /kernel/drv/md.conf
[qfs2rac-node1]root@solaris:~#
```

8. 如果 *kernel/drv/md.conf* 文件不存在，则创建该文件。使 *root* 用户成为该文件的所有者，并使 *sys* 成为组所有者。将权限设置为 *644*。

   在示例中，用 *vi* 编辑器创建文件。文件的内容应该如下所示：

```
[qfs2rac-node1]root@solaris:~# vi kernel/drv/md.conf
#####################################################
#pragma ident   "@(#)md.conf    2.1   00/07/07 SMI"
#
# Copyright (c) 1992-1999 by Sun Microsystems, Inc.
# All rights reserved.
#
name="md" parent="pseudo" nmd=128 md_nsets=4;
#####################################################
:wq
[qfs2rac-node1]root@solaris:~# chown root:sys kernel/drv/md.conf
[qfs2rac-node1]root@solaris:~# chmod 644
[qfs2rac-node1]root@solaris:~#
```

9. 动态重新扫描 *md.conf* 文件并确保设备树得到更新。使用命令 *update_drv -f md*：

   在示例中，设备树得到更新。因此 Solaris Volume Manager 已安装：

```
[qfs2rac-node1]root@solaris:~# update_drv -f md
[qfs2rac-node1]root@solaris:~# ls -l  /dev/md/admin
lrwxrwxrwx   1 root root 31 Apr 20 10:12 /dev/md/admin -> ../../devices/pseudo/md@0:admin
```

```
[qfs2rac-node1]root@solaris:~#
```

10. 接下来，创建 Solaris Volume Manager 多属主磁盘组。

创建 Solaris Volume Manager 多属主磁盘组

1. 以 *root* 用户身份登录到 SC-RAC 配置中的所有节点。

   在示例中，登录到节点 *qfs2rac-node1*。然后我们打开新的终端窗口并使用 *ssh* 登录到节点 *qfs2rac-node2* 和 *qfs2rac-node3*：

```
[qfs2rac-node1]root@solaris:~#

[qfs2rac-node1]root@solaris:~# ssh root@qfs2rac-node2
Password:
[qfs2rac-node2]root@solaris:~#

[qfs2rac-node1]root@solaris:~# ssh root@qfs2rac-node3
Password:
[qfs2rac-node3]root@solaris:~#
```

2. 如果您要在 Solaris 11.*x* 或更高版本上使用 Oracle Solaris Cluster 4.*x*（如果尚未执行），则在继续之前，先在每个节点上安装 Solaris Volume Manager。

   从 Solaris 11 开始，默认不安装 Solaris Volume Manager。

3. 在每个节点上，连接新的状态数据库设备并创建三个状态数据库副本。使用命令 *metadb -a -f -c3 device-name*，其中 *device-name* 是 *cXtYdYsZ* 形式的物理设备名称。

   请勿使用 Solaris Cluster 设备标识符 (Device Identifier, DID)。使用物理设备名称。在示例中，我们在全部三个群集节点上创建状态数据库设备。

```
[qfs2rac-node1]root@solaris:~# metadb -a -f -c3 /dev/rdsk/c0t0d0

[qfs2rac-node2]root@solaris:~# metadb -a -f -c3 /dev/rdsk/c0t6d0

[qfs2rac-node3]root@solaris:~# metadb -a -f -c3 /dev/rdsk/c0t4d0
```

4. 在一个节点上创建 Solaris Volume Manager 多属主磁盘组。使用命令 *metaset -sdiskset -M -a -h host-list*，其中 *host-list* 是空格分隔的属主列表。

   Solaris Volume Manager 最多支持每个磁盘集四个主机。在本示例中，我们在 *qfs2rac-node1* 上创建磁盘组 *datadisks* 并指定三个节点 *qfs2rac-node1*、*qfs2rac-node2* 和 *qfs2rac-node3* 作为属主（请注意，下面的命令是作为单行输入的－使用反斜杠字符对换行符进行转义）：

```
[qfs2rac-node1]root@solaris:~# metaset -s datadisks -M -a -h qfs2rac-node1 /
qfs2rac-node2 qfs2rac-node3
```

5. 列出其中一个节点上的设备。使用 Solaris Cluster 命令 *cldevice list -n -v*。

```
[qfs2rac-node1]root@solaris:~# cldevice list -n -v
DID Device  Full Device Path
----------  ----------------
d13         qfs2rac-node1:/dev/rdsk/c6t600C0FF00000000000332B62CF3A6B00d0
d14         qfs2rac-node1:/dev/rdsk/c6t600C0FF0000000000876E950F1FD9600d0
d15         qfs2rac-node1:/dev/rdsk/c6t600C0FF0000000000876E9124FAF9C00d0
...
[qfs2rac-node1]root@solaris:~#
```

6. 在 *cldevice list -n -v* 命令的输出中，选择要镜像的设备。

   在示例中，我们针对四个镜像选择四对设备：*d21* 和 *d13*、*d14* 和 *d17*、*d23* 和 *d16*，以及 *d15* 和 *d19*。

```
[qfs2rac-node1]root@solaris:~# cldevice list -n -v
DID Device  Full Device Path
----------  ----------------
d13         qfs2rac-node1:/dev/rdsk/c6t600C0FF00000000000332B62CF3A6B00d0
d14         qfs2rac-node1:/dev/rdsk/c6t600C0FF0000000000876E950F1FD9600d0
d15         qfs2rac-node1:/dev/rdsk/c6t600C0FF0000000000876E9124FAF9C00d0
d16         qfs2rac-node1:/dev/rdsk/c6t600C0FF00000000000332B28488B5700d0
d17         qfs2rac-node1:/dev/rdsk/c6t600C0FF000000000086DB474EC5DE900d0
d18         qfs2rac-node1:/dev/rdsk/c6t600C0FF0000000000876E975EDA6A000d0
d19         qfs2rac-node1:/dev/rdsk/c6t600C0FF000000000086DB47E331ACF00d0
d20         qfs2rac-node1:/dev/rdsk/c6t600C0FF0000000000876E9780ECA8100d0
d21         qfs2rac-node1:/dev/rdsk/c6t600C0FF000000000004CAD5B68A7A100d0
d22         qfs2rac-node1:/dev/rdsk/c6t600C0FF000000000086DB43CF85DA800d0
d23         qfs2rac-node1:/dev/rdsk/c6t600C0FF000000000004CAD7CC3CDE500d0
d24         qfs2rac-node1:/dev/rdsk/c6t600C0FF000000000086DB4259B272300d0
....
[qfs2rac-node1]root@solaris:~#
```

7. 将选定的设备添加到相同节点的磁盘集中。使用命令 *metaset -a devicelist*，其中 *devicelist* 是空格分隔的一个或多个群集设备标识符的列表。

   在本示例中，我们将列出的磁盘添加到多属主磁盘集 *dataset1* 中（请注意，下面的命令是作为单行输入的－使用反斜杠字符对换行符进行转义）：

```
[qfs2rac-node1]root@solaris:~# metaset -s dataset1 -M -a -h /dev/did/rdsk/d21 / /dev/did/rdsk/d13 /dev/did/
rdsk/d14 /dev/did/rdsk/d17 /dev/did/rdsk/d23 /
/dev/did/rdsk/d16 /dev/did/rdsk/d15 /dev/did/rdsk/d19
[qfs2rac-node1]root@solaris:~#
```

8. 接下来，创建 QFS 数据和元数据的镜像卷。

创建 QFS 数据和元数据的镜像卷

1. 为了使组件之间保持明确的关系，请为 RAID-0 逻辑卷和要创建的 RAID-1 镜像确定命名模式。

   一般而言，RAID-1 镜像命名为 *dn*，其中 *n* 是一个整数。构成 RAID-1 镜像的 RAID-0 卷命名为 *dnX*，其中 *X* 是代表镜像中设备位置的整数（对于双向镜像，通常为 *0* 或 *1*）。

   在本过程的示例中，我们基于 RAID-0 逻辑卷对创建双向 RAID-1 镜像。因此我们将镜像命名为 *d1*、*d2*、*d3*、*d4* 等等，依此类推。然后我们命名构成 RAID-1 镜像的每对 RAID-0 卷：*d10* 和 *d11*、*d20* 和 *d21*、*d30* 和 *d31*、*d40* 和 *d41* 等等。
2. 登录到您创建多属主磁盘集的节点。以 *root* 用户身份登录。

   在上面的示例中，我们在 *qfs2rac-node1* 上创建了磁盘集：

```
[qfs2rac-node1]root@solaris:~#
```

3. 创建第一个 RAID-0 逻辑卷。使用命令 *metainit -s diskset-name device-name number-of-stripes components-per-stripe component-names*，其中：
   - *diskset-name* 是您选择的磁盘集名称。
   - *device-name* 是您选择的 RAID-0 逻辑卷名称。
   - *number-of-stripes* 为 *1*。
   - *components-per-stripe* 为 *1*。
   - *component-name* 是要用于 RAID-0 卷的磁盘集组件的设备名称。

   在示例中，使用多属主磁盘集 *dataset1* 中的群集 (DID) 设备 */dev/did/dsk/d21s0* 创建 RAID-0 逻辑卷 *d10*：

```
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d10 1 1 /dev/did/dsk/d21s0
[qfs2rac-node1]root@solaris:~#
```

4. 创建其余的 RAID-0 逻辑卷。

```
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d11 1 1 /dev/did/dsk/d13s0
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d20 1 1 /dev/did/dsk/d14s0
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d21 1 1 /dev/did/dsk/d17s0
```

```
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d30 1 1 /dev/did/dsk/d23s0
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d31 1 1 /dev/did/dsk/d16s0
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d40 1 1 /dev/did/dsk/d15s0
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d41 1 1 /dev/did/dsk/d19s0
...
[qfs2rac-node1]root@solaris:~#
```

5. 创建第一个 RAID-1 镜像。使用命令 *metainit -s diskset-name RAID-1-mirrorname -m RAID-0-volume0*，其中：
   - *diskset-name* 是多属主磁盘集的名称。
   - *RAID-1-mirrorname* 是 RAID-1 镜像卷的名称。
   - *RAID-0-volume0* 是您添加到镜像的第一个 RAID-0 逻辑卷。

   在示例中，我们创建镜像 *d1* 并将第一个 RAID-0 卷 *d10* 添加到该镜像中：

```
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d1 -m d10
[qfs2rac-node1]root@solaris:~#
```

6. 将其余的 RAID-0 卷添加到第一个 RAID-1 镜像中。使用命令 *metattach -s diskset-name RAID-1-mirrorname RAID-0-volume*，其中：
   - *diskset-name* 是多属主磁盘集的名称
   - *RAID-1-mirrorname* 是 RAID-1 镜像卷的名称
   - *RAID-0-volume* 是您添加到镜像的 RAID-0 逻辑卷。

   在示例中，*d1* 是一个双向镜像，因此我们添加一个 RAID-0 卷 *d11*：

```
[qfs2rac-node1]root@solaris:~# metattach -s dataset1 d11 d1
[qfs2rac-node1]root@solaris:~#
```

7. 创建其余的镜像。

   在示例中，我们创建镜像 *d2*、*d3*、*d4* 等等。

```
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d2 -m d20
[qfs2rac-node1]root@solaris:~# metattach -s dataset1 d21 d2
[qfs2rac-node1]root@solaris:~# metainit -s dataset2 d3 -m d30
[qfs2rac-node1]root@solaris:~# metattach -s dataset2 d31 d3
[qfs2rac-node1]root@solaris:~# metainit -s dataset2 d4 -m d40
[qfs2rac-node1]root@solaris:~# metattach -s dataset2 d41 d4
...
[qfs2rac-node1]root@solaris:~#
```

8. 选择将存放 QFS 文件系统元数据的镜像。

对于下面的示例，我们选择镜像 *d1* 和 *d2*。

9. 在选定的镜像中，创建软分区来存放 QFS 元数据。对于每个镜像，使用命令 *metainit -s diskset-name partition-name -p RAID-1-mirrorname size*，其中：
   - *diskset-name* 是多属主磁盘集的名称。
   - *partition-name* 是新分区的名称。
   - *RAID-1-mirrorname* 是镜像的名称。
   - *size* 是分区的大小。

   在示例中，我们创建两个 500G 的分区：*d53*（位于镜像 *d1* 上）和 *d63*（位于镜像 *d2* 上）：

```
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d53 -p d1 500g
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d63 -p d2 500g
```

10. 接下来，使用镜像卷在 SC-RAC 群集上创建 QFS 共享文件系统。

使用镜像卷在 SC-RAC 群集上创建 QFS 共享文件系统

1. 如果您尚未“在所有 SC-RAC 群集节点上创建 QFS 共享文件系统 Hosts 文件”，请执行该过程。完成后，返回此处。
2. 选择将同时充当 SC-RAC 群集的主节点和 QFS 共享文件系统的活动元数据服务器的群集节点。以 *root* 用户身份登录。

   在示例中，我们选择节点 *qfs2rac-node1*：

```
[qfs2rac-node1]root@solaris:~#
```

3. 在主节点上，创建共享高性能 *ma* 文件系统。将 Solaris Volume Manager 镜像磁盘卷用作 *mm* 元数据设备和 *mr* 数据设备。在文本编辑器中，打开 */etc/opt/SUNWsamfs/mcf* 文件，进行必需的编辑后保存文件。

   在示例中，使用 *vi* 文本编辑器创建文件系统 *qfs2rac*。镜像卷 *d1* 和 *d2* 上的分区充当文件系统的两个 *mm* 元数据设备 *110* 和 *120*。镜像卷 *d3* 和 *d4* 充当文件系统的两个 *mr* 数据设备 *130* 和 *140*。

```
[qfs2rac-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# /etc/opt/SUNWsamfs/mcf file:
#
# Equipment               Equipment Equipment Family   Device  Additional
# Identifier              Ordinal   Type      Set      State   Parameters
# ----------------------- --------- --------  -------  ------  ----------
qfs2rac                   100         ma        qfs2rac  on      shared
 /dev/md/dataset1/dsk/d53  110         mm        qfs2rac  on
```

```
 /dev/md/dataset1/dsk/d63  120         mm        qfs2rac  on
 /dev/md/dataset1/dsk/d3   130         mr        qfs2rac  on
 /dev/md/dataset1/dsk/d4   140         mr        qfs2rac  on
:wq
[qfs2rac-node1]root@solaris:~#
```

4. 检查 *mcf* 文件中是否存在错误。使用命令 */opt/SUNWsamfs/sbin/sam-fsd* 并更正发现的任何错误。

   sam-fsd 命令会读取 Oracle HSM 配置文件并初始化文件系统。如果遇到错误，该命令将停止。在示例中，检查主机 *qfs2rac-node1* 上的 *mcf* 文件：

```
[qfs2rac-node1]root@solaris:~# sam-fsd
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
[qfs2rac-node1]root@solaris:~#
```

5. 创建文件系统。使用命令 */opt/SUNWsamfs/sbin/sammkfs -S family-set-name*，其中 *family-set-name* 是文件系统的设备标识符。

   sammkfs 命令读取 *hosts.family-set-name* 和 *mcf* 文件，并创建具有指定属性的共享文件系统。

```
[qfs2rac-node1]root@solaris:~# sammkfs -S qfs2rac
Building 'qfs2rac' will destroy the contents of devices:
  ...
Do you wish to continue? [y/N]yes ...
[qfs2rac-node1]root@solaris:~#
```

6. 在文本编辑器中打开操作系统的 */etc/vfstab* 文件，并为新的文件系统添加一行。在第一列中输入文件系统名称，然后输入几个空格，在第二列中输入一个连字符，然后输入更多的空格。

   在示例中，使用 *vi* 文本编辑器。我们为 *qfs2rac* 文件系统添加了一行。连字符阻止操作系统尝试使用 UFS 工具检查文件系统的完整性：

```
[qfs2rac-node1]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      -
...
qfs2rac   -
```

7. 在 */etc/vfstab* 文件的第三列中，输入相对于群集的文件系统挂载点。指定未直接位于系统根目录下的挂载点子目录。

   将共享 QFS 文件系统直接挂载到根目录下可能会在使用 *SUNW.qfs* 资源类型时导致故障转移问题。在示例中，*qfs2rac* 文件系统的挂载点为 */global/sc-rac/qfs2rac*：

```
#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      -
...
qfs2rac   -       /global/sc-rac/qfs2rac samfs  -    no
```

8. 在 */etc/vfstab* 文件的第四列中，输入文件系统类型 (*samfs*)。

```
#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      -
...
qfs2rac   -       /global/sc-rac/qfs2rac samfs
```

9. 在 */etc/vfstab* 文件的第五列中，输入 *fsck* 传递选项 (-)。

```
#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      -
...
qfs2rac   -       /global/sc-rac/qfs2rac samfs  -
```

10. 在 */etc/vfstab* 文件的第六列中，输入在引导时挂载选项 (*no*)。

```
#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      -
...
qfs2rac   -       /global/sc-rac/qfs2rac samfs  -    no
```

11. 在 */etc/vfstab* 文件的第七列中，输入 *sw_raid* 挂载选项和针对 SC-RAC 配置建议的挂载选项。然后保存文件并关闭编辑器。

    建议使用以下挂载选项。它们可以在 */etc/vfstab* 中指定，如果更方便，也可以在文件 */etc/opt/SUNWsamfs/samfs.cmd* 中指定：

    - *shared*
    - *stripe=1*
    - *sync_meta=1*
    - *mh_write*
    - *qwrite*
    - *forcedirectio*
    - *notrace*
    - *rdlease=300*
    - *wrlease=300*
    - *aplease=300*

    在示例中，简化了挂载选项列表以适应页面布局：

```
#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      -
...
qfs2rac   -       /global/sc-rac/qfs2rac samfs  -    no      shared,...sw_raid
:wq
[qfs2rac-node1]root@solaris:~#
```

12. 创建高可用性共享文件系统的挂载点。

```
[qfs2rac-node1]root@solaris:~# mkdir -p /global/sc-rac/qfs2rac
[qfs2rac-node1]root@solaris:~#
```

13. 在主节点上挂载高可用性共享文件系统。

```
[qfs2rac-node1]root@solaris:~# mount /global/sc-rac/qfs2rac
[qfs2rac-node1]root@solaris:~#
```

14. 设置第二个节点。使用“在其余的 SC-RAC 群集节点上配置潜在 QFS 元数据服务器 ” 过程。
15. 配置故障转移。使用“配置 SC-RAC 元数据服务器的故障转移” 过程。然后返回此处。

    您已使用 Solaris Volume Manager 镜像卷成功配置 SC-RAC 配置。
16. 如果计划使用边带数据库功能，请转至第 10 章 配置报告数据库。
17. 否则，请转至第 11 章 配置通知和日志记录。

10

第 10 章 配置报告数据库

Oracle HSM 支持一个可选报告数据库，该数据库存储指定文件系统中每个文件的当前元数据信息。此边带数据库对于文件和文件系统活动的管理和报告是非常重要的。

Oracle HSM 边带数据库的实施非常简单。使用 *samdb* 命令创建和配置 MySQL 数据库（使用提供的数据库方案或定制替代数据库方案）和 *samfsdump* 命令生成的恢复点文件。然后，Oracle HSM 守护进程在相应的文件系统发生更改时自动更新数据库。利用其他 *samdb* 命令，可以查询和管理数据库。有关命令和选项的完整详细信息，请参见 *samdb* 和 *samdb.conf* 手册页。

要使用边带数据库功能，请执行下列任务：

- 安装并配置 MySQL 服务器软件
- 创建数据库装入文件
- 创建边带数据库

安装并配置 MySQL 服务器软件

要启用 *samdb* 报告功能，您必须安装并配置 MySQL 数据库。请执行如下操作。

1. 从 *http://dev.mysql.com/doc/* 下载 MySQL 参考手册。

   使用以下过程可以确定在启用 *samdb* 报告时所需执行的 MySQL 任务。但请注意，下列步骤不一定是完整的或权威的。在查阅 MySQL 参考手册时将它们用作指南。
2. 以 *root* 用户身份登录到将托管 MySQL 服务器的系统。

   可以在 Oracle HSM 元数据服务器主机或独立 Solaris 或 Linux 主机上安装 MySQL 服务器。

   在示例中，我们将在 Solaris 主机 *samsql* 上安装 MySQL：

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

3. 按照 *MySQL* 参考手册中的指示，下载并安装 MySQL 服务器软件。启用自动启动。
4. 使用 *root* 用户帐户，通过 *mysql* 客户机连接到 MySQL 服务器。使用命令 *mysql --user=root -p*。当系统提示时，输入您在安装期间为 *root* 用户分配的密码。

mysql 命令 shell 将启动：

```
[samsql]root@solaris:~# mysql --user=root -p
Enter Password:
mysql>
```

5. 创建 Oracle HSM MySQL 用户。使用命令 *CREATE USER 'user_name'@'host_name' IDENTIFIED BY 'user-password'*，其中：
   - *user_name* 是 Oracle HSM MySQL 用户的名称
   - *host_name* 是在 Oracle HSM 元数据服务器主机上安装 MySQL 时的 *localhost*；否则，为元数据服务器的主机名或 IP 地址。
   - *user-password* 是您分配给 Oracle HSM MySQL 用户的密码。

   在示例中，我们将在 Oracle HSM 元数据服务器 *samqfs1mds* 上创建用户 *samsql*。我们为了演示目的而设置了用户密码 *samsqluserpassw0rd*（这对生产数据库使用来说并不是安全的选择）：

```
[samsql]root@solaris:~# mysql --user=root
Enter Password:
mysql> CREATE USER 'samsql'@'samqfs1mds' IDENTIFIED BY 'samsqluserpassw0rd'
mysql>
```

6. 向 Oracle HSM 用户授予必需的权限。使用命令 *GRANT CREATE,DROP,INDEX,SELECT,INSERT,UPDATE,DELETE ON host_name TO 'user_name'@'host_name'*。

   在示例中，我们将为元数据服务器 *samqfs1mds* 上的用户 *samsql* 授予权限：

```
[samsql]root@solaris:~# mysql --user=root -p
Enter Password:
mysql> CREATE USER 'samsql'@'host_name' IDENTIFIED BY 'samsqluserpassw0rd'
mysql> GRANT CREATE,DROP,INDEX,SELECT,INSERT,UPDATE,DELETE ON samqfs1mds TO /
'samsql'@'samqfs1mds'
mysql>
```

7. 关闭 MySQL 命令接口，返回到操作系统命令 shell。使用 MySQL 命令 *QUIT*。

```
[samsql]root@solaris:~# mysql --user=root -p
Enter Password:
mysql> CREATE USER 'samsql'@'host_name' IDENTIFIED BY 'samsqluserpassw0rd'
mysql> GRANT CREATE,DROP,INDEX,SELECT,INSERT,UPDATE,DELETE ON samqfs1mds TO /
'samsql'@'samqfs1mds'
mysql> QUIT
Bye
```

```
root@solaris:~#
```

8. 接下来，创建数据库装入文件。

创建数据库装入文件

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

   在示例中，我们登录到主机 *samqfs1mds*：

```
[samqfs1mds]root@solaris:~#
```

2. 如果您已具有当前恢复点文件，则从恢复点文件内容生成数据库装入文件。使用命令 *samfsrestore -SZ output-path-name -f recoverypoint-file*，其中：
   - *-f* 将 *recoverypoint-file* 指定为输入文件的路径和文件名。
   - *-SZ* 导致此命令扫描恢复点文件并使用 *output-path-name* 指定的路径和文件名输出数据库装入文件。

   有关其他详细信息，请参见 *samfsdump* 手册页。

   在示例中，我们将使用日常恢复点文件 */zfs1/hsmqfs1_recovery/140129*，该文件是我们在配置 *samqfs1* 文件系统时调度的（请参见“配置文件系统保护”）。我们将输出发送至数据库装入文件 */root/hsmqfs1dataload*（请注意，下面的命令是作为单行输入的－使用反斜杠字符对换行符进行转义）：

```
[samqfs1mds]root@solaris:~# samfsrestore -SZ /root/hsmqfs1dataload -f /
/zfs1/hsmqfs1_recovery/140129
```

3. 如果您没有当前恢复点文件，则现在创建数据库装入文件。转到 Oracle HSM 文件系统的根目录。然后使用命令 *samfsdump -SZ output-path-name*。

   有关其他详细信息，请参见 *samfsdump* 手册页。在示例中，我们转至 */hsmqfs1* 目录。我们将输出发送到数据库装入文件 */root/hsmqfs1dataload*：

```
[samqfs1mds]root@solaris:~# cd /hsmqfs1
[samqfs1mds]root@solaris:~# samfsdump -SZ /root/hsmqfs1dataload
```

4. 接下来，创建边带数据库。

创建边带数据库

1. 以 *root* 用户身份登录到 MySQL 服务器主机。

   在示例中，MySQL 服务器托管在 Solaris 主机 *samqfs1mds* 上：

```
[samqfs1mds]root@solaris:~#
```

2. 在文本编辑器中，打开文件 */etc/opt/SUNWsamfs/samdb.conf*。

   在示例中，使用 *vi* 编辑器。我们首先添加标题行作为注释：

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/samdb.conf
 #FS_NAME:HOST:USER:PASSWORD:NAME:PORT:CLIENT_FLAG:MOUNT_POINT
```

3. 在 *samdb.conf* 文件的第一列中，输入文件系统的系列集名称，后跟冒号 (*:*) 作为列分隔符。

   在示例中，我们输入系列集名称 *samqfs1*：

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/samdb.conf
# /etc/opt/SUNWsamfs/samdb.conf
 #FS_NAME:HOST:USER:PASSWORD:NAME:PORT:CLIENT_FLAG:MOUNT_POINT
samqfs1:
```

4. 在第二列中，输入 MySQL 数据库服务器的主机名，后跟冒号 (*:*) 作为列分隔符。

   在示例中，我们将数据库服务器共同托管在 Oracle HSM 元数据服务器主机 *samqfs1mds* 上。因此我们输入主机名 *localhost*：

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/samdb.conf
# /etc/opt/SUNWsamfs/samdb.conf
 #FS_NAME:HOST:USER:PASSWORD:NAME:PORT:CLIENT_FLAG:MOUNT_POINT
samqfs1:localhost:
```

5. 在第三列中，输入 Oracle HSM 软件在访问 MySQL 数据库时使用的用户名，后跟冒号 (*:*) 作为列分隔符。

   在示例中，为了登录数据库，我们已创建用户 *samqfs*：

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/samdb.conf
# /etc/opt/SUNWsamfs/samdb.conf
 #FS_NAME:HOST:USER:PASSWORD:NAME:PORT:CLIENT_FLAG:MOUNT_POINT
samqfs1:localhost:samqfs:
```

6. 在第四列中，输入 Oracle HSM 软件在访问 MySQL 数据库时使用的密码，后跟冒号 (*:*) 作为列分隔符。

   在示例中，使用伪密码 *P^ssw0rd*：

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/samdb.conf
# /etc/opt/SUNWsamfs/samdb.conf
 #FS_NAME:HOST:USER:PASSWORD:NAME:PORT:CLIENT_FLAG:MOUNT_POINT
```

```
samqfs1:localhost:samqfs:Pˆssw0rd:
```

7. 在第四列中，输入 MySQL 数据库的名称，后跟冒号 (*:*) 作为列分隔符。

   在示例中，我们将数据库命名为 *samqfs1db*：

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/samdb.conf
# /etc/opt/SUNWsamfs/samdb.conf
 #FS_NAME:HOST:USER:PASSWORD:NAME:PORT:CLIENT_FLAG:MOUNT_POINT
samqfs1:localhost:samqfs:Pˆssw0rd:samqfs1db:
```

8. 在第六列中，输入数据库服务器的 TCP/IP 端口，后跟冒号 (*:*) 作为列分隔符。

   在示例中，输入 *0*（零）。如果我们使用的是远程服务器，则零（或空）值将指定默认端口 *3306*。不过，由于我们使用的是 *localhost*，零只是充当占位符：

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/samdb.conf
# /etc/opt/SUNWsamfs/samdb.conf
 #FS_NAME:HOST:USER:PASSWORD:NAME:PORT:CLIENT_FLAG:MOUNT_POINT
samqfs1:localhost:samqfs:Pˆssw0rd:samqfs1db:0:
```

9. 在第七列中，输入 MySQL 客户机标志，后跟冒号 (*:*) 作为列分隔符。

   MySQL 客户机标志通常设置为 *0*（零）。不过，可以设置不同的值组合以便启用特定 MySQL 功能。有关详细信息，请参阅 MySQL 文档中的 *mysql_real_connect()* 函数。

   在示例中，输入 *0*（零）：

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/samdb.conf
# /etc/opt/SUNWsamfs/samdb.conf
 #FS_NAME:HOST:USER:PASSWORD:NAME:PORT:CLIENT_FLAG:MOUNT_POINT
samqfs1:localhost:samqfs:Pˆssw0rd:samqfs1db:0:0:
```

10. 在第八列和最后一列中，输入 Oracle HSM 文件系统的挂载点。保存文件并关闭编辑器。

    在示例中，文件系统在 */hsmqfs/hsmqfs1* 中挂载：

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/samdb.conf
# /etc/opt/SUNWsamfs/samdb.conf
 #FS_NAME:HOST:USER:PASSWORD:NAME:PORT:CLIENT_FLAG:MOUNT_POINT
samqfs1:localhost:samqfs:Pˆssw0rd:samqfs1db:0:0:/hsmqfs/hsmqfs1
:wq
[samqfs1mds]root@solaris:~#
```

11. 创建新的数据库和关联表。使用命令 *samdb create family_set*，其中 *family_set* 是在 */etc/opt/SUNWsamfs/mcf* 文件中为 Oracle HSM 文件系统指定的系列集名称。

    默认数据库方案为 */opt/SUNWsamfs/etc/samdb.schema*。您可以通过 *samdb create family_set -s schema* 的形式输入命令来指定替代方案，其中 *schema* 是方案文件的路径和名称。

    在示例中，使用默认方案为文件系统系列集 *samqfs1* 创建数据库。

```
[samqfs1mds]root@solaris:~# samdb create samqfs1
```

12. 使用您在前面的过程中创建的数据库装入文件中包含的数据填充数据库。使用命令 *samdb load family_set input_file*，其中 *family_set* 是在 */etc/opt/SUNWsamfs/mcf* 文件中为文件系统指定的系列集名称，而 *input_file* 是数据库装入文件的路径和名称。

    在示例中，使用数据库装入文件 */root/hsmqfs1dataload* 为文件系统系列集 *samqfs1* 装入数据库。

```
[samqfs1mds]root@solaris:~# samdb load samqfs1 /root/hsmqfs1dataload
```

13. 检查数据库的一致性。使用命令 *samdb check family_set*，其中 *family_set* 是在 */etc/opt/SUNWsamfs/mcf* 文件中为文件系统指定的系列集名称。

    samdb check 命令会将数据条目与当前文件系统元数据进行比较。它记录并尽力更正可能在装入过程中引发的不一致性。

    在示例中，使用数据库装入文件 */root/hsmqfs1dataload* 为文件系统系列集 *samqfs1* 装入数据库。

```
[samqfs1mds]root@solaris:~# samdb check samqfs1
```

14. 接下来，在启用数据库支持的情况下挂载 Oracle HSM 文件系统。

在启用数据库支持的情况下挂载 Oracle HSM 文件系统

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

```
[samqfs1mds]root@solaris:~#
```

2. 备份 */etc/vfstab* 文件。

```
[samqfs1mds]root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
```

3. 在文本编辑器中打开 */etc/vfstab* 文件，并向下滚动到您已为之创建数据库的文件系统的条目。

   在示例中，使用 *vi* 编辑器。我们将向下滚动到 *samqfs1* 文件系统的条目：

```
[samqfs1mds]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       -
...
samqfs1    -       /hsmqfs1  samfs   -     yes      ... ,partial=64
```

4. 在 */etc/vfstab* 文件的最后一列中，将 *sam_db* 添加到文件系统的挂载选项列表。然后保存文件并关闭编辑器。

   在示例中，在 *samqfs1* 文件系统上启用边带数据库：

```
[samqfs1mds]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       -
...
samqfs1    -        /hsmqfs1  samfs   -     yes      ... ,partial=64,sam_db
:wq
root@solaris:~#
```

5. 挂载 Oracle HSM 归档文件系统。

   使用 *sam_db* 选项挂载文件系统之后，Oracle HSM 软件将启动更新边带数据库的过程。

   在示例中，我们挂载文件系统 */hsmqfs1*：

```
[samqfs1mds]root@solaris:~# mount /hsmqfs1
```

6. 接下来，转至第 11 章 配置通知和日志记录

11

第 11 章 配置通知和日志记录

Oracle HSM 文件系统使用简单网络管理协议 (Simple Network Management Protocol, SNMP) 支持自动化远程通知，并提供全面的可配置日志记录工具。本章概述了以下主题：

- 配置简单网络管理协议 (Simple Network Management Protocol, SNMP)
- 启用 Oracle HSM 日志记录
- 配置设备日志记录
- 配置日志轮转
- 启用电子邮件警报。

配置简单网络管理协议 (Simple Network Management Protocol, SNMP)

网络管理应用程序可以使用简单网络管理协议 (Simple Network Management Protocol, SNMP) 监视 Oracle HSM 文件系统。可以将 SNMP 代理配置为自动发送陷阱，以向网络管理站发出警报，提示发生故障和配置更改。

Oracle HSM 管理信息库 (Management Information Base, MIB) 定义 SNMP 陷阱提供的信息类型。这包括配置错误、SCSI *tapealert* 事件，以及各种非典型的系统活动。有关更多信息，请参见管理信息库 (Management Information Base, MIB) 文件 */var/snmp/mib/SUN-SAM-MIB.mib*。

当 Oracle HSM 陷阱事件发生时，Solaris 内核系统事件通知守护进程 *sysevent* 调用脚本 */etc/opt/SUNWsamfs/scripts/sendtrap*。然后该脚本将陷阱发送到本地主机或您指定的管理站。该脚本支持 2c 版本的 SNMP 标准，并且可以向后兼容该标准的早期版本。请注意，2c 版本以明文形式交换验证凭证（团体字符串）和管理数据。有关其他详细信息，请参见 *sendtrap* 手册页。

要配置 SNMP 通知，请执行以下任务：

- 确保所有 SNMP 管理站都已在 **/etc/hosts** 文件中列出
- 启用 SNMP 支持
- 将管理站指定为陷阱接收方并配置验证。

本节还提供您在任何时候需要禁用 SNMP 支持时应该遵循的说明。

确保所有 SNMP 管理站都已在 /etc/hosts 文件中列出

1. 以 *root* 用户身份登录到 Oracle HSM 服务器。

   在示例中，Oracle HSM 服务器主机为 *samqfs1mds*：

```
[samqfs1mds]root@solaris:~#
```

2. 在文本编辑器中打开 */etc/hosts* 文件。确保该文件针对您要用作 SNMP 管理站的每个主机包含一个条目。

   在示例中，使用 *vi* 编辑器。打算使用的管理站之一 *management1* 已列出。但是另一台管理站 *management2* 则未列出：

```
[samqfs1mds]root@solaris:~# vi /etc/hosts
# Internet host table
::1 localhost
127.0.0.1 localhost loghost
10.0.0.10 server1
10.0.0.20 management1
```

3. 如果 */etc/hosts* 文件未包含某些或所有您打算使用的 SNMP 管理站主机的条目，则添加必需的条目，然后保存文件。

   在示例中，使用 *vi* 编辑器添加缺少的管理站 *management2*：

```
[samqfs1mds]root@solaris:~# vi /etc/hosts
# Internet host table
::1 localhost
127.0.0.1 localhost loghost
10.0.0.10 server1
10.0.0.20 management1
10.0.0.30 management2
```

4. 当 */etc/hosts* 文件包含所有您打算使用的 SNMP 管理站主机的条目时，请关闭编辑器。

```
[samqfs1mds]root@solaris:~# vi /etc/hosts
...
10.0.0.20 management1
10.0.0.30 management2
:wq
[samqfs1mds]root@solaris:~#
```

5. 接下来，启用 SNMP 支持。

启用 SNMP 支持

默认情况下启用对 SNMP 通知的支持，因此无需任何操作，除非在某种情况下禁用了 SNMP 支持。如果您需要重新启用 SNMP 支持，请执行如下操作：

1. 以 *root* 用户身份登录到 Oracle HSM 服务器。

   在示例中，Oracle HSM 服务器主机为 *samqfs1mds*：

```
[samqfs1mds]root@solaris:~#
```

2. 在文本编辑器中打开文件 */etc/opt/SUNWsamfs/defaults.conf*。查找 *alerts* = *off* 行。

   指令 *alerts* = *off* 禁用 SNMP 支持。在示例中，在 *vi* 编辑器中打开文件并找到该行：

```
[samqfs1mds]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.
...
alerts = off
```

3. 要启用对 SNMP 通知的支持，请将 *alerts* 指令的值更改为 *on*。然后保存文件并关闭编辑器。

```
[samqfs1mds]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.
...
alerts = on
:wq
[samqfs1mds]root@solaris:~#
```

4. 指示 Oracle HSM 服务重新读取 *defaults.conf* 文件并相应地重新配置自身。使用命令 *samd config*。

```
[samqfs1mds]root@solaris:~# 1mds]samd config
[samqfs1mds]root@solaris:~#
```

5. 接下来，将管理站指定为陷阱接收方并配置验证。

将管理站指定为陷阱接收方并配置验证

1. 以 *root* 用户身份登录到 Oracle HSM 服务器。

   在示例中，Oracle HSM 服务器主机为 *samqfs1mds*：

```
[samqfs1mds]root@solaris:~#
```

2. 在文本编辑器中打开 */etc/opt/SUNWsamfs/scripts/sendtrap* 文件，然后查找以 *TRAP_DESTINATION=* 开头的行。

   sendtrap 文件是可配置的 shell 脚本。在示例中，用 *vi* 编辑器打开文件：

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/scripts/sendtrap
# /etc/opt/SUNWsamfs/scripts/sendtrap#!/usr/bin/sh
# sendtrap:
#  This script gets invoked by the sysevent configuration file.
#  This is not expected to be run as a stand-alone program
...
# CONFIGURATION PARAMETERS:
TRAP_DESTINATION=`hostname`
```

3. 在 *TRAP_DESTINATION=`* 行中，将单引号中的文本替换为空格分隔的一个或多个陷阱接收方的列表，每个接收方采用形式 *hostname*:*port*，其中 *hostname* 是管理站的主机名（在 */etc/hosts* 中列出），而 *port* 是主机在其上侦听陷阱的端口。

   默认情况下，陷阱发送到 *localhost* 的 UDP 端口 *161*。在示例中，我们将 *management1* 和 *management2* 主机添加到默认的 *localhost* 中。*localhost* 和 *management1* 使用默认的端口，而 *management2* 使用定制端口 *1161*：

```
...
# CONFIGURATION PARAMETERS:
TRAP_DESTINATION=`localhost:161 management1:161 management1:1161`
```

4. 向下滚动到设置团体字符串 *COMMUNITY="public"* 的行。

   该团体字符串是共享的纯文本密码，对 SNMP 2c 版本中的代理和管理站进行验证。默认值为 SNMP 标准 *public*。

```
...
# CONFIGURATION PARAMETERS:
TRAP_DESTINATION=`localhost:161 management1:161 management1:1161`
...
COMMUNITY="public"
```

5. 将 *COMMUNITY=""* 指令设置为管理站使用的值。然后保存文件并关闭编辑器。

请勿编辑文件中的任何其他内容。*COMMUNITY=""* 和 *TRAP_DESTINATION=``* 是唯一可编辑的参数。

请注意，默认 SNMP 团体字符串 *public* 不安全。因此您的网络管理员可能会要求使用一个更安全的字符串。2c 版本的 SNMP 最多允许 32 个字母数字字符。在本示例中，我们将团体字符串设置为 *Iv0wQh2th74bVVt8of16t1m3s8it4wa9*。

```
...
# CONFIGURATION PARAMETERS:
TRAP_DESTINATION=`localhost:161 management1:163 management1:1162`
...
COMMUNITY="Iv0wQh2th74bVVt8of16t1m3s8it4wa9"
:wq
[samqfs1mds]root@solaris:~#
```

6. 接下来，启用 Oracle HSM 应用程序日志记录。

禁用 SNMP 支持

默认情况下，系统启用远程通知功能。如果您要禁用远程通知，请执行如下操作：

1. 以 *root* 用户身份登录到 Oracle HSM 服务器。

   在示例中，Oracle HSM 服务器主机为 *samqfs1mds*：

```
[samqfs1mds]root@solaris:~#
```

2. 在文本编辑器中打开 */etc/opt/SUNWsamfs/defaults.conf* 文件。查找 *#alerts = on* 行。

   在示例中，使用 *vi* 编辑器：

```
[samqfs1mds]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.
...
#alerts = on
[samqfs1mds]root@solaris:~#
```

3. 要禁用对 SNMP 通知的支持，请删除井号 (*#*) 以取消对该行的注释，然后将 *alerts* 的值更改为 *off*。然后保存文件并关闭编辑器。

```
[samqfs1mds]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.
...
alerts = off
:wq
[samqfs1mds]root@solaris:~#
```

4. 指示 Oracle HSM 服务重新读取 *defaults.conf* 文件并相应地重新配置自身。使用命令 *samd config*。

```
[samqfs1mds]root@solaris:~# samd config
[samqfs1mds]root@solaris:~#
```

5. 在此处停止。SNMP 支持已禁用。

启用 Oracle HSM 日志记录

/var/adm/sam-log 文件记录 Oracle HSM 应用程序及其组件守护进程和流程的状态和错误信息。要设置日志记录流程，请执行如下操作：

启用 Oracle HSM 应用程序日志记录

1. 以 *root* 用户身份登录到 Oracle HSM 服务器。

   在示例中，Oracle HSM 服务器主机为 *samqfs1mds*：

```
[samqfs1mds]root@solaris:~#
```

2. 在文本编辑器中打开 */etc/syslog.conf* 文件。

   在示例中，用 *vi* 编辑器打开文件：

```
[samqfs1mds]root@solaris:~# vi /etc/syslog.conf
# syslog configuration file ...
*.err;kern.notice;auth.notice                  /dev/sysmsg
*.err;kern.debug;daemon.notice;mail.crit       /var/adm/messages
*.alert;kern.err;daemon.err                    operator
*.alert                                        root
...
```

3. 在 */etc/syslog.conf* 文件中，添加由字符串 *local7.debug*、一个或多个制表符以及路径字符串 */var/adm/sam-log* 组成的行。然后保存文件并关闭编辑器。

   在示例中，我们还添加一个注释：

```
[samqfs1mds]root@solaris:~# vi /etc/syslog.conf
```

```
# syslog configuration file ...
*.err;kern.notice;auth.notice                  /dev/sysmsg
*.err;kern.debug;daemon.notice;mail.crit       /var/adm/messages
*.alert;kern.err;daemon.err                    operator
*.alert                                        root
...
# Oracle HSM logging
local7.debug    /var/adm/sam-log
:wq
[samqfs1mds]root@solaris:~#
```

4. 创建日志文件 */var/adm/sam-log*。使用命令 *touch /var/adm/sam-log*。

```
[samqfs1mds]root@solaris:~# touch /var/adm/sam-log
[samqfs1mds]root@solaris:~#
```

5. 告诉 Solaris *syslogd* 守护进程重新读取其配置文件，然后启动 Oracle HSM 日志记录。使用命令 *pkill -HUP syslogd*。

   不管何时它收到 HUP 信号，*syslogd* 日志记录服务都重新读取 */etc/syslog.conf* 配置文件、关闭所有打开的日志文件，然后打开 *syslog.conf* 中列出的日志文件。当命令运行时，Oracle HSM 日志记录随即启用：

```
[samqfs1mds]root@solaris:~# pkill -HUP syslogd
[samqfs1mds]root@solaris:~#
```

6. 转至配置设备日志记录。

配置设备日志记录

设备日志记录工具提供特定于各台硬件设备的错误信息（它不收集软介质错误）。每台设备都具有其自己的日志文件，以对应的设备序号命名，并存储在 */var/opt/SUNWsamfs/devlog/* 目录中。

设备日志增长很快。因此默认情况下，系统记录有限的事件数据集：*err*、*retry*、*syserr* 和 *date*。之后如果出现问题，则可以使用 *samset* 命令针对每个设备记录其他事件（有关详细信息，请参见 *samset* 手册页的 *devlog* 一节）。

在 `defaults.conf` 文件中启用设备日志

要启用基本设备日志记录，请执行如下操作：

1. 以 *root* 用户身份登录到 Oracle HSM 服务器。

   在示例中，Oracle HSM 服务器主机为 *samqfs1mds*：

```
[samqfs1mds]root@solaris:~#
```

2. 在文本编辑器中打开 */etc/opt/SUNWsamfs/defaults.conf*。

   在示例中，用 *vi* 编辑器打开文件：

```
[samqfs1mds]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.
...
```

3. 在 *defaults.conf* 文件中，添加定义默认所需设备日志记录级别的行。输入指令 *devlog equipment-number loggable-events*，其中：
   - *equipment-number* 要么是代表 */etc/opt/SUNWsamfs/mcf* 文件中定义的所有设备的关键字 *all*，要么是标识 *mcf* 中定义的具体设备的设备序号。
   - *loggable-events* 是空格分隔的默认值列表：*err retry syserr date*

     有关事件类型的完整列表，请参见 *samset* 手册页的 *devlog* 一节。但是，要最大限度地减小日志大小，请使用默认选项。进行诊断时，*samset* 命令可以根据需要选择性地启用其他事件。

   在示例中，我们针对所有设备使用默认日志记录级别启用设备日志记录：

```
[samfs-mds1]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.
...
devlog all err retry syserr date
```

4. 保存 *defaults.conf* 文件，然后关闭编辑器。

```
[samfs-mds1]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.
...
devlog all err retry syserr date
:wq
[samfs-mds1]root@solaris:~#
```

5. 指示 Oracle HSM 服务重新读取 *defaults.conf* 文件并相应地重新配置自身。使用命令 *samd config*。

```
[samfs-mds1]root@solaris:~# samd config
[samfs-mds1]root@solaris:~#
```

6. 接下来，设置 Oracle HSM 日志文件的自动轮转。

配置日志轮转

日志文件增长很快，占用大量的空间并使日志难以使用。因此您应该为 Oracle HSM 日志配置自动日志轮转。软件包括一个用于此目的的脚本 *log_rotate.sh*，您可以从 Solaris *crontab* 文件运行该脚本。

对于每个您打算轮转的日志，创建两个 *crontab* 条目。第一个在所需时间运行 *log_rotate.sh* 脚本。如果目标日志文件达到了指定的最小大小（默认值为 *100000* 字节），则脚本对其执行重命名，然后删除最早的现有副本（始终保留七个副本）。第二个 *crontab* 条目告诉 Solaris 日志记录守护进程 *syslogd* 使用新的日志文件重新启动日志记录。

设置 Oracle HSM 日志文件的自动轮转

考虑轮转以下日志：

- Oracle HSM 日志文件 *sam-log*，位于 */etc/syslog.conf* 文件中指定的位置。
- 位于 */var/opt/SUNWsamfs/devlog/* 目录中的设备日志文件。
- */etc/opt/SUNWsamfs/stager.cmd* 文件中指定的回写程序日志文件。
- */etc/opt/SUNWsamfs/releaser.cmd* 文件中指定的释放程序日志文件。
- */etc/opt/SUNWsamfs/recycler.cmd* 文件中指定的回收程序日志文件。

不应轮转归档程序日志文件！日志信息对于分析和文件系统恢复很有价值。有关归档程序日志的正确处理，请参见“配置文件系统保护”。

当您确定应该轮转的日志后，请针对每个日志执行如下操作：

1. 以 *root* 用户身份登录到 Oracle HSM 服务器。

   在示例中，Oracle HSM 服务器主机为 *samqfs1mds*：

```
[samqfs1mds]root@solaris:~#
```

2. 将脚本文件 *log_rotate.sh* 从 */opt/SUNWsamfs/examples/*（卸载位置）复制到 */etc/opt/SUNWsamfs/scripts/*。

   请注意，下面的命令是作为单行输入的－使用反斜杠对换行符进行转义：

```
[samfs-mds1]root@solaris:~# cp /opt/SUNWsamfs/examples/log_rotate.sh /  /etc/opt/SUNWsamfs/scripts/
```

3. 打开 *root* 用户的 *crontab* 文件进行编辑。使用命令 *crontab -e*。

   crontab 命令可在 *EDITOR* 环境变量指定的文本编辑器中打开 *root* 用户 *crontab* 文件的可编辑副本（有关完整的详细信息，请参见 Solaris *crontab* 手册页）。在示例中，使用 *vi* 编辑器：

```
[samfs-mds1]root@solaris:~# crontab -e
#ident "%Z%%M% %I% %E% SMI"
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
...
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
```

4. 在新的一行上，通过输入 *minutes hour * * day-of-the-week* 来指定日志文件将轮转的时间，其中：
   - *minutes* 是一个介于 [*0-59*] 范围内的整数，用于指定作业开始时的分钟。
   - *hour* 是一个介于 [*0-23*] 范围内的整数，用于指定作业开始时的小时。
   - *（星号）指定未使用的值。

     对于日常运行的任务，不使用日期值 [*1-31*] 以及月份值 [*1-12*]。
   - *day-of-the-week* 是范围 [*0-6*] 中的一个整数，从星期日 (*0*) 开始。
   - 空格用于分隔时间规范中的字段。

   在示例中，我们安排日志轮转在每个星期日的凌晨 3:10 开始。

```
[samfs-mds1]root@solaris:~# crontab -e
#ident "%Z%%M% %I% %E% SMI"
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
...
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 3 * * 0
```

5. 继续在相同的行上输入轮转 Oracle HSM 日志的 shell 脚本文件的路径和名称 */etc/opt/SUNWsamfs/scripts/log_rotate.sh*，后跟一个空格。

```
[samfs-mds1]root@solaris:~# crontab -e
#ident "%Z%%M% %I% %E% SMI"
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
```

```
# Use is subject to license terms.
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
...
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 3 * * 0 /etc/opt/SUNWsamfs/scripts/log_rotate.sh
```

6. 继续在相同的行上输入需要轮转的日志的名称以及要轮转的最小文件大小。输入文本 *samfslog* [*minimum-size*]，其中 *samfslog* 是 Oracle HSM 日志文件的路径，而 [*minimum-size*] 是一个可选的指定脚本轮转的最小文件大小（以字节为单位）的整数（默认值为 *100000*）。

   在示例中，我们需要轮转 */var/adm/sam-log*。我们接受默认的最小大小：

```
[samfs-mds1]root@solaris:~# crontab -e
#ident "%Z%%M% %I% %E% SMI"
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
...
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 3 * * 0 /etc/opt/SUNWsamfs/scripts/log_rotate.sh /var/adm/sam-log
```

7. 开始新行。创建在 *log_rotate.sh* 脚本 10 分钟后启动的 *crontab* 条目。该条目告知 Solaris *syslogd* 守护进程关闭其旧日志文件并恢复新文件日志记录。输入行 *minutes hour * * day-of-the-week /bin/kill -HUP `/bin/cat /etc/syslog.pid`*，其中 *minutes hour * * day-of-the-week* 指定的时间比上一步骤中指定的时间晚 10 分钟。

   在示例中，该条目在每个星期日的凌晨 3:20 重新启动 Oracle HSM 日志记录：

```
[samfs-mds1]root@solaris:~# crontab -e
#ident "%Z%%M% %I% %E% SMI"
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
...
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 3 * * 0 /etc/opt/SUNWsamfs/scripts/log_rotate.sh /var/adm/sam-log
20 3 * * 0  /bin/kill -HUP `/bin/cat /etc/syslog.pid`
```

8. 保存文件并关闭编辑器。

```
[samfs-mds1]root@solaris:~# crontab -e
#ident "%Z%%M% %I% %E% SMI"
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
...
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 3 * * 0 /etc/opt/SUNWsamfs/scripts/log_rotate.sh /var/adm/sam-log
20 3 * * 0  /bin/kill -HUP `/bin/cat /etc/syslog.pid`
:wq
[samfs-mds1]root@solaris:~#
```

9. 重复此过程，直到为所有需要的日志配置了日志轮转。
10. 现在，如果需要，启用电子邮件警报。
11. 否则，请转至第 13 章 备份 *Oracle HSM* 配置。

启用电子邮件警报

电子邮件警报最好从 Oracle HSM Manager 图形用户界面进行设置。有关详细信息，请参见联机帮助。

如果您必须从命令行界面配置电子邮件警报，请参见 *defaults.conf*、*archiver.sh*、*dev_down.sh*、*load_notify.sh*、*recycler.sh*、*archiver.cmd*、*recycler.cmd* 和 *notify.cmd* 手册页。

您的 Oracle HSM 系统现已配置。但是在您开始使用它之前，请保护您所做的工作。有关说明，请参见第 13 章 备份 *Oracle HSM* 配置。

12

第 12 章 针对特殊需求调整 I/O 特征

前面章节中介绍的基本文件系统配置步骤在大多数情况下提供了最佳平衡性能。因此，如果您完全不确定您的应用程序的行为方式，则通常最好将此部分中的设置保留为默认值。但是，如果您的应用程序发出非常一致或非常大的 I/O 请求，则整体性能可能会从调整或更改文件系统处理物理 I/O 的方式中受益。

当所有或大部分读取和写入完全在磁盘扇区的 512 字节边界上开始和结束时，物理 I/O 最有效。磁盘 I/O 只能在扇区大小的块中发生。因此，当 I/O 请求跨越扇区边界时，系统必须执行附加操作以便将应用程序数据与同一扇区中的无关数据分隔开。系统必须确保后者在此过程中不会损坏。在最坏的情况下，跨扇区写入时，文件系统必须读取扇区，修改内存中的扇区数据，然后将扇区写回磁盘。其他单独的机械活动会使此类读取、修改和写入操作对性能造成极大损害。

遗憾的是，大部分应用程序需要读取和写入扇区边界上未良好对齐的各种大小的数据。出于此原因，与许多文件系统一样，默认情况下，Oracle HSM 将使用分页 *I/O*。文件系统将通过在 Solaris 分页内存的数据高速缓存中读取或写入来处理应用程序的即时 I/O 请求。文件系统将使用大小更有效且对齐得更好的物理读取和写入来异步更新高速缓存。只要它从磁盘中读取数据，就可通过预见即将执行的读取并在同一操作中将对应数据装入到高速缓存来进行大多数物理 I/O。这样，将使用虚拟内存页中高速缓存的数据来满足大多数 I/O 请求，而无需其他物理磁盘活动。分页 I/O 使用内存并对系统 CPU 施加一些额外负载，不过在大多数情况下，这些成本可以被更高的物理 I/O 效率抵消。

但在少数情况下，与分页 I/O 相关的额外开销无法通过其优势抵消。对于始终执行对齐良好的 I/O 的应用程序以及可调整达到此目的的应用程序来说，页面高速缓存毫无裨益。执行超大 I/O 的应用程序也很难从页面高速缓存中获益，因为仅第一个和最后一个扇区未对齐，而且因为大型 I/O 可能在任何情况下都太大而无法保留在高速缓存中。最终，对于流式传输遥测数据、监视视频或其他类型的实时信息的应用程序，在写入未立即提交到非易失存储时可能面临着丢失不可恢复的数据的风险。在这些情况下，最好使用直接 *I/O*。指定直接 I/O 后，文件系统会直接在应用程序内存与磁盘设备之间传输数据，而绕过页面高速缓存。

Oracle HSM 在选择和调整 I/O 高速缓存行为方面会为您提供相当大的自由度。在了解您的应用程序的 I/O 特征并执行“针对预测的文件系统 I/O 调整 Solaris 系统和驱动程序参数”中介绍的任务之后，请按照如下所示选择您的方式：

- 如果您的应用程序一致地发出小的、大小不同和/或未对齐的 I/O 请求，则接受 Oracle HSM 默认设置。请勿在此部分中进行任何更改。

- 如果您的应用程序发出大小不同但大于平均大小的、未对齐的 I/O 请求，则针对大型数据传输优化分页 I/O。
- 如果您的应用程序发出的请求是对齐良好或超大的 I/O 请求与小型、未对齐的请求的混合，则允许在分页 I/O 和直接 I/O 之间切换。
- 如果您的应用程序一致地发出对齐良好或超大的 I/O 请求，则将文件系统配置为仅使用直接 I/O。
- 如果在共享文件系统客户机上运行的应用程序一致地打开大量文件，则增加目录名称查找高速缓存大小。

针对大型数据传输优化分页 I/O

分页 I/O 可进行调整，以便更好地与应用程序和硬件特征匹配。高速缓存的读取和写入应该足够大，以传输应用程序所传输的平均数据量或物理存储可传输的最大数据量，以两者中较大的为准。如果我们无法调整其中任一页面高速缓存行为，则高速缓存将得不到充分利用，应用程序 I/O 请求将需要更多物理 I/O，并且整体系统性能将变差。

例如，考虑在单一磁盘卷上实施的 *md* 数据设备与在 3+1 RAID 5 卷组上实施的 *md* 设备之间的差异。如果我们通过将单一 64 KB 磁盘分配单元 (disk allocation unit, DAU) 从高速缓存写入后者来处理应用程序的每个写入请求，从而避免多磁盘设备可能具有的其他带宽问题，则 RAID 设备必须将 I/O 拆分为三个更小的、效率更低的 21 和 22 KB 段，之后再将数据写出到 RAID 组中的三个数据磁盘。使用此配置完成应用程序的 64 KB I/O 请求需要的工作明显多于它在我们使用页面高速缓存将请求组装成一个 3-DAU、192 KB I/O 时需要的工作。如果应用程序可以（或者能够调整为）使用设备带宽的偶数倍（192、384 或 576 KB）发出 I/O 请求，则我们可以高速缓存更多数据并使用每个物理 I/O 传输更多数据，从而进一步减少开销并相应地提升性能。

因此，确定您的应用程序的 I/O 需求并了解您的硬件的 I/O 属性。然后执行如下操作。

1. 以 *root* 用户身份登录到文件系统主机。

```
root@solaris:~#
```

2. 备份操作系统的 */etc/vfstab* 文件。

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
root@solaris:~#
```

3. 在文本编辑器中打开 */etc/vfstab* 文件，并找到需要调整的文件系统对应的行。

   在本示例中，文件系统名为 *qfsma*：

```
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       -
...
qfsma      -        /qfsma    samfs   -     yes      ...
```

4. 在文件系统的 *Mount Options* 字段中，添加 *writebehind=n* 挂载选项，其中 *n* 是 *8* 千字节的倍数。使用逗号（无空格）分隔挂载选项。保存文件并关闭编辑器。

   writebehind 选项将确定在页面高速缓存刷新到磁盘之前，可在页面高速缓存中排队的指定文件的数量。将此参数设置为更高的值将改善性能，因为大型队列会将多个小应用程序写入合并到更少、更大、更有效的物理 I/O 中。将此参数设置为更低的值将更好地保护数据，因为更改将更快地写入非易失存储。

   默认值为 *512* KB（八个 64 KB DAU），此大小一般有利于大块、顺序 I/O。但在本示例中，系列集包含两个具有分散读写文件分配的 *md* 磁盘设备。分散读写宽度为一个 64 KB DAU，用于将 128 KB 写入到两个 *md* 设备。*md* 设备是 3+1 RAID 5 组。因此我们需要向三个数据轴中的每个轴写入至少 128 KB，写入总计至少为 *768* KB（96 个组，每组 8 KB）：

```
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  --------------------
/devices   -        /devices  devfs   -     no       -
...
qfsma      -        /qfsma    samfs   -     yes      ...,writebehind=768
:wq
root@solaris:~#
```

5. 测试文件系统的 I/O 性能，并根据需要调整 *writebehind* 设置。
6. 在文本编辑器中重新打开 */etc/vfstab* 文件。在文件系统的 *Mount Options* 字段中，添加 *readahead=n* 挂载选项，其中 *n* 是 8 KB 的倍数。使用逗号（无空格）分隔挂载选项。保存文件并关闭编辑器。

   readahead 选项将确定在一次物理读取过程中读入高速缓存的数据量。当应用程序看起来似乎要按顺序读取时，文件系统会在每次物理读取过程中高速缓存即将读取的文件数据块。接下来将从高速缓存内存中处理一系列应用程序读取请求，将若干应用程序读取请求合并到一个物理 I/O 请求中。

   默认值为 *1024* KB（十六个 64 KB DAU），此大小一般对大块、顺序 I/O 有利。如果数据库或类似应用程序执行其自己的 readahead，则将 Oracle HSM *readahead* 设置为 *0* 以避免冲突。否则，*readahead* 一般应设置为高速缓存一次物理 I/O 可传输的最大数据量。如果 *readahead* 设置小于应用程序通常请求和设

备可提供的数据量，则完成应用程序 I/O 请求将需要不必要的物理 I/O。但是，如果 *readahead* 设置过高，则它可能占用非常多的内存，导致整体系统性能降级。在示例中，我们将 *readahead* 设置为 *736* KB（三十六个 64 KB DAU）。

```
#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       -
...
qfsma      -        /qfsma    samfs   -     yes      ...,readahead=736
:wq
root@solaris:~#
```

7. 测试文件系统的 I/O 性能，并根据需要调整 *readahead* 设置。

   增加 *readahead* 参数的大小可增强大型文件的传输性能，但性能增强的幅度很小。因此在重置 *readahead* 大小之后测试系统的性能。然后向上调整 *readahead* 大小，直到传输速率不再提高为止。

允许在分页 I/O 和直接 I/O 之间切换

您可以将 Oracle HSM 文件系统配置为在分页 I/O 和直接 I/O 之间切换（如果这样做更符合应用程序的 I/O 行为的话）。指定可能受益于直接 I/O 的读取和写入的扇区对齐和最小大小特征，然后设置应触发切换的符合条件的读取和写入数量。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. 备份操作系统的 */etc/vfstab* 文件。

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
root@solaris:~#
```

3. 在文本编辑器中打开 */etc/vfstab* 文件并找到您要配置的文件系统的行。

   在本示例中，文件系统名为 *qfsma*：

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

```
/devices  -      /devices devfs  -    no    -
/proc     -      /proc    proc   -    no    -
...
qfsma     -      /qfsma   samfs  -    yes   stripe=1
```

4. 要为与 512 字节的扇区边界对齐良好的读取请求设置起始直接 I/O 的阈值大小，请将 *dio_rd_form_min=n* 挂载选项添加到文件系统的 *Mount Options* 字段，其中 *n* 是 KB 数。使用逗号（无空格）分隔挂载选项。

   默认情况下，*dio_rd_form_min=256* KB。在示例中，我们知道我们的应用程序直到请求一次读取至少 512 KB 才会一致地生成对齐良好的读取。因此，我们将对齐良好的直接读取的阈值大小更改为 *512*：

```
#File     Device                     Mount
#Device   to     Mount    System fsck at    Mount
#to Mount fsck   Point    Type   Pass Boot  Options
#-------- ------ -------- ------ ---- ----- ----------------------------
/devices  -      /devices devfs  -    no    -
/proc     -      /proc    proc   -    no    -
...
qfsma     -      /qfsma   samfs  -    yes   stripe=1,dio_rd_form_min=512
```

5. 要为与 512 字节的扇区边界对齐良好的读取请求设置起始直接 I/O 的阈值大小，请将 *dio_wr_form_min=n* 挂载选项添加到文件系统的 *Mount Options* 字段，其中 *n* 是 KB 数。使用逗号（无空格）分隔挂载选项。

   默认情况下，*dio_wr_form_min=256* KB。在示例中，我们知道我们的应用程序直到请求一次写入至少 1 MB 才会一致地生成对齐良好的写入。因此，我们将对齐良好的直接写入项的阈值大小更改为 *1024* KB：

```
#File     Device                     Moun
#Device   to     Mount    System fsck at    Mount
#to Mount fsck   Point    Type   Pass Boot  Options
#-------- ------ -------- ------ ---- ----- ----------------------------
/devices  -      /devices devfs  -    no    -
/proc     -      /proc    proc   -    no    -
...
qfsma     -      /qfsma   samfs  -    yes   ...,dio_wr_form_min=1024
```

6. 要为与 512 字节的扇区边界未对齐的读取请求设置起始直接 I/O 的阈值大小，请将 *dio_rd_ill_min=n* 挂载选项添加到文件系统的 *Mount Options* 字段，其中 *n* 是 KB 数。使用逗号（无空格）分隔挂载选项。

   默认情况下，*dio_rd_ill_min=0* KB，因此直接 I/O 不会用于未对齐的读取。在示例中，我们知道我们的应用程序通常会为小数据块发出未对齐的读取请求。将

按顺序重新读取大量此类数据。因此页面高速缓存可能对这些读取有益。切换到直接 I/O 将导致额外的无用物理 I/O 并降低性能。因此我们接受默认值并且不更改 *vfstab* 文件：

```
#File     Device                      Mount
#Device   to     Mount    System fsck at    Mount
#to Mount fsck   Point    Type   Pass Boot  Options
#-------- ------ -------- ------ ---- ----- ------------------------------
/devices  -      /devices devfs  -    no    -
/proc     -      /proc    proc   -    no    -
...
qfsma     -      /qfsma   samfs  -    yes   ...,dio_wr_form_min=1024
```

7. 要为与 512 字节的扇区边界未对齐的写入请求设置起始直接 I/O 的阈值大小，请将 *dio_wr_ill_min=n* 挂载选项添加到文件系统的 *Mount Options* 字段，其中 *n* 是 KB 数。使用逗号（无空格）分隔挂载选项。

   默认情况下，*dio_wr_ill_min=0* KB，因此直接 I/O 不会用于未对齐的写入。未对齐的写入尤其会损害性能，因为系统必须读取、修改和写入扇区。但在本示例中，我们知道我们的应用程序偶尔会发出不在扇区边界内的大型、单一写入请求。由于读取、写入和修改操作限制为连续扇区的大型块的开头和结尾，因此直接 I/O 的好处要超过分页 I/O。因此我们设置 *dio_wr_ill_min=2048* KB：

   在本示例中，我们将写入未对齐数据过程中使用直接 I/O 的默认阈值更改为 *2048* KB：

```
#File     Device                      Mount
#Device   to     Mount    System fsck at    Mount
#to Mount fsck   Point    Type   Pass Boot  Options
#-------- ------ -------- ------ ---- ----- ------------------------------
/devices  -      /devices devfs  -    no    -
/proc     -      /proc    proc   -    no    -
...
qfsma     -      /qfsma   samfs  -    yes   ...,dio_wr_ill_min=2048
```

8. 要为读取启用直接 I/O，请将 *dio_rd_consec=n* 挂载选项添加到 *Mount Options* 字段，其中 *n* 是连续 I/O 传输的数量，该数量必须满足上面指定的大小和对齐要求才会触发切换到直接 I/O 的操作。选择一个针对受益于直接 I/O 的应用程序操作而选择的值。使用逗号（无空格）分隔挂载选项。

   默认情况下 *dio_rd_consec=0*，因此将禁用 I/O 切换。在示例中，我们知道，我们的应用程序请求三个连续、对齐良好的读取（其大小至少为 *dio_rd_form_min* 指定的最小大小，即 512 KB）后，应用程序将继续请求直到足够长以使直接 I/O 比较有价值。*dio_rd_form_min* 指定的最小大小是默认值 *0*，因此启用直接 I/O 将不影响未对齐读取请求。因此我们设置 *dio_rd_consec=3*：

```
#File     Device                      Mount
#Device   to     Mount    System fsck at    Mount
#to Mount fsck   Point    Type   Pass Boot  Options
#-------- ------ -------- ------ ---- ----- ----------------------------
/devices  -      /devices devfs  -    no    -
/proc     -      /proc    proc   -    no    -
...
qfsma     -      /qfsma   samfs  -    yes   ...,dio_rd_consec=3
```

9. 要为写入启用直接 I/O，请将 *dio_wr_consec=n* 挂载选项添加到 *Mount Options* 字段，其中 *n* 是连续 I/O 传输的数量，该数量必须满足上面指定的大小和对齐要求才会触发切换到直接 I/O 的操作。选择一个针对受益于直接 I/O 的应用程序操作而选择的值。使用逗号（无空格）分隔挂载选项。

   默认情况下 *dio_wr_consec=0*，因此将禁用 I/O 切换。在示例中，我们知道，我们的应用程序请求两个连续、对齐良好的写入（其大小至少为 *dio_wr_form_min* 指定的最小大小，即 1024 KB）后，它将继续请求直到足够长以使直接 I/O 比较有价值。我们还知道，两个连续、未对齐写入将大于 *dio_wr_form_min* (2048 KB)，这已经足够大，未对齐的影响相对来说不大。因此我们设置 *dio_wr_consec=2*：

```
#File     Device                      Mount
#Device   to     Mount    System fsck at    Mount
#to Mount fsck   Point    Type   Pass Boot  Options
#-------- ------ -------- ------ ---- ----- ----------------------------
/devices  -      /devices devfs  -    no    -
/proc     -      /proc    proc   -    no    -
...
qfsma     -      /qfsma   samfs  -    yes   ...,dio_wr_consec=2
```

10. 保存 *vfstab* 文件并关闭编辑器。

```
#File     Device                      Mount
#Device   to     Mount    System fsck at    Mount
#to Mount fsck   Point    Type   Pass Boot  Options
#-------- ------ -------- -----  ---- ----- ----------------------------
/devices  -      /devices devfs  -    no    -
/proc     -      /proc    proc   -    no    -
...
qfsma     -      /qfsma   samfs  -    yes   ...,dio_wr_consec=2
:wq
root@solaris:~#
```

11. 挂载已修改的文件系统：

```
root@solaris:~# mount /qfsms
root@solaris:~#
```

将文件系统配置为仅使用直接 I/O

当应用程序的 I/O 特征需要仅使用直接 I/O 时，您可以使用 *forcedirectio* 挂载选项来挂载整个文件系统（有关如何为独立文件或目录指定直接 I/O 的信息，请参见 Oracle HSM *setfa* 手册页）。

要挂载文件系统以仅使用直接 I/O，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. 备份操作系统的 */etc/vfstab* 文件。

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
root@solaris:~#
```

3. 在文本编辑器中打开 */etc/vfstab* 文件，然后查找您要在其中使用直接 I/O 的文件系统对应的行。

   在本示例中，文件系统名为 *qfsma*：

```
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       -
...
qfsma      -        /qfsma    samfs   -     yes      stripe=1
```

4. 在文件系统的 *Mount Options* 字段中，添加 *forcedirectio* 挂载选项。使用逗号（无空格）分隔挂载选项。保存文件并关闭编辑器。

```
#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       -
...
```

```
qfsma      -        /qfsma    samfs   -     yes      stripe=1,forcedirectio
:wq
root@solaris:~#
```

5. 挂载已修改的文件系统：

```
root@solaris:~# mount /qfsms
root@solaris:~#
```

增加目录名称查找高速缓存大小

在共享文件系统的客户机一次打开大量文件时，元数据服务器上的 Oracle Solaris 目录名称查找高速缓存 (directory name lookup cache, DNLC) 的默认大小可能会显得不足。元数据服务器将代表所有客户机查找文件名，因此文件系统性能在这些情况下可能会下降。

如果您预期会有此类工作负载，则将目录名称查找高速缓存大小参数 *ncsize* 更改为默认大小的两倍或三倍。有关说明，请参见 *Oracle Solaris Information Library* 中的《*Oracle Solaris* 可调参数参考手册》（请参见前言的"可用文档"部分）。

13

第 13 章 备份 Oracle HSM 配置

完成 Oracle Hierarchical Storage Manager and StorageTek QFS Software 配置后，通过备份配置文件和相关信息来保护您的投资。执行以下任务：

- 为您的 Oracle HSM 配置创建备份位置
- 运行 **samexplorer** 并安全地存储报告
- 手动备份 Oracle HSM 配置。

为您的 Oracle HSM 配置创建备份位置

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. 为 Oracle HSM 配置的备份副本选择存储位置。选择可挂载在文件系统主机上的独立文件系统。
3. 确保所选文件系统未与归档文件系统共享任何物理设备。

   请勿将恢复点文件存储在用于起保护作用的文件系统中。请勿将恢复点文件存储在也托管归档文件系统的物理设备的逻辑设备（如分区或 LUN）上。
4. 在所选文件系统中，创建一个目录来存储配置信息。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，*path* 是所选目录的路径和名称。

   在示例中，我们已在独立文件系统 */zfs1* 上创建 */zfs1/sam_config* 目录：

```
root@solaris:~# mkdir /zfs1/sam_config
```

5. 接下来，运行 *samexplorer* 并安全地存储报告。

运行 samexplorer 并安全地存储报告

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。Oracle 支持服务人员在进行故障排除时使用该输出。因此，无论您

何时配置或重新配置 Oracle HSM 软件和文件系统，创建基线 *samexplorer* 报告都是一个很好的方法。

1. 以 *root* 用户身份登录到文件系统元数据主机。

   在本示例中，主机名为 *samqfs1mds*：

```
[samqfs1mds]root@solaris:~#
```

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，*path* 是所选目录的路径和名称。

   在示例中，我们创建目录 */zfs1/sam_config/explorer*：

```
[samqfs1mds]root@solaris:~# mkdir /zfs1/sam_config/explorer
[samqfs1mds]root@solaris:~#
```

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，我们在目录 */zfs1/sam_config/explorer/* 中创建文件 *samhost1.20140130.1659MST.tar.gz*（请注意，下面的命令是作为单行输入的－使用反斜杠字符对换行符进行转义）：

```
[samqfs1mds]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

[samqfs1mds]root@solaris:~#
```

4. 只要对文件系统配置做了重大更改，就应重复该过程。
5. 接下来，手动备份 Oracle HSM 配置。

手动备份 Oracle HSM 配置

虽然 *samexplorer* 实用程序可捕获许多 Oracle HSM 配置信息，但为了实现完全冗余，在每次执行重要的配置工作后，您应执行以下过程：

1. 以 *root* 用户身份登录到文件系统主机。

   在本示例中，主机名为 *samqfs1mds*：

```
[samqfs1mds]root@solaris:~#
```

2. 在保存备份配置信息的目录中，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，*path* 是所选目录的路径和名称。

   在示例中，要为归档文件系统 */hsmqfs1* 配置恢复点。因此创建了目录 */zfs1/sam_config/samconfig*：

```
[samqfs1mds]root@solaris:~# mkdir /zfs1/sam_config/samconfig
[samqfs1mds]root@solaris:~#
```

3. 在选择的目录中，为当前 Oracle HSM 配置创建一个子目录。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，将日期用于命名子目录：

```
samqfs1mdsroot@solaris:~# mkdir /zfs1/sam_config/samconfig/20140127
[samqfs1mds]root@solaris:~#
```

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
```

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－使用反斜杠对换行符进行转义）：

```
samqfs1mdsroot@solaris:~# dump_cat -V /var/opt/SUNWsamfs/catalog/library1cat /
> /zfs1/sam_config/20140513/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，如果您创建了 *crontab* 条目来管理恢复点的创建和日志轮转，则要立即保存副本。

8. 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。
9. 在所选子目录中，保存下载的 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。

10. 在此处停止。您的配置已经备份完毕，文件系统可以使用了。

附录 A

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

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* 文件系统的数据和元数据。*md* 设备将文件数据存储在 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
Cygnet 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

附录 B. 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 上的线程池中将始终至少具有 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 C

附录 C. 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与它们相关的归档集分配指令的后面。

归档副本指令以 *copy-number* 值 *1*、*2*、*3* 或 *4* 开头。数字后面是一个或多个用于指定该副本归档特征的参数。每一个归档副本指令均采用以下格式：

```
copy-number [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*) 的归档时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*) 的归档时限时才会释放该副本，并在副本 *2* 达到 7 年 6 个月 (*7y6m*) 的取消归档时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 软件会通过在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*]（-3.402823466x10^{38} 至 3.402823466x10^{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*、*queuewait*、*re-archive*、*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-amld*) 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

hlwm_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)+(1x100)=900*
- 优先级卷的回写挂载请求的聚合优先级为 *1000+0+(1x100)=1100*
- 非优先卷的回写挂载请求的聚合优先级为 *0+0+(1x100)=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+(1x100)=1600*
- 优先级卷的回写挂载请求的聚合优先级为 *1000+0+(1x100)=1100*
- 非优先卷的回写挂载请求的聚合优先级为 *0+0+(1x100)=100*

附录 D

附录 D. 示例

/opt/SUNWsamfs/examples/ 子目录包含样例 Oracle HSM 配置文件、shell 脚本和 *dtrace* 程序，用于说明各种功能以及各种要求的解决方案。其中包括以下文件：

```
01_example.archiver.cmd.simple.txt
01_example.mcf.simple.txt
01_example.vfstab.txt
02_example.archiver.cmd.disk.tape.txt
02_example.diskvols.conf.NFS.txt
02_example.mcf.shared.txt
02_example.vfstab.disk.archive.NFS.txt
03_example.archiver.cmd.dk.9840.9940.txt
03_example.diskvols.conf
03_example.mcf.dk.9840.9940.txt
03_example.stk.9840C_parms.txt
03_example.stk.9940B_parms.txt
03_example.vfstab.disk.archive.txt
04_example.archiver.cmd.9840.LTO.txt
04_example.mcf.ma.9840.LTO.txt
04_example.stk50c.txt
05_example.archiver.cmd.9840.9940.T10k.txt
05_example.mcf.veritas.9840.9940.T10K.txt
05_example.stk_params9840.txt
05_example.stk_params9940.txt
05_example.stk_paramsT10K.txt
05_example.vstab.txt
06_example.archiver.cmd.samremote.client.txt
06_example.local.copy.samremote.client.stk50.txt
06_example.mcf.samremote.client.txt
06_example.mcf.samremote.server.txt
06_example.samremote.client.setup.stk100.txt
06_example.samremote.client.vfstab.txt
06_example.samremote.configuration.samremote.server.txt
07_example.mcf.distio.client.txt
07_example.mcf.distio.mds.txt
archiver.sh
defaults.conf
dev_down.sh
dtrace/fs_mon
dtrace/ino_mon
hosts.shsam1
hosts.shsam1.local.client
hosts.shsam1.local.server
inquiry.conf
load_notify.sh
log_rotate.sh
metadata_config_samfs.xml
nrecycler.sh
preview.cmd
recover.sh
recycler.sh
```

```
restore.sh
samdb.conf
samfs.cmd
samst.conf
save_core.sh
sendtrap
ssi.sh
st.conf_changes
stageback.sh
syslog.conf_changes
tarback.sh
verifyd.cmd
```

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、异地和离线存储在内的存储空间，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，对涉及到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磁带，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） 装入磁带以验证其 VSN 的过程。对于磁光磁带，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，自动装入和卸载可移除介质磁带。自动化库包含两个部分：一个或多个驱动器，以及将磁带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（如硬盘或盒式磁带）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（盒式磁带）

一种数据存储介质容器，如磁带或光学介质。有时也称为卷、磁带或介质。请参见 **volume（卷）**、**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磁带是否存在。

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。*setfa* 命令的 *-D* 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）

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。

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。*md* 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。*mr* 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

	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大或最小级别。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 *archiver.cmd* 文件中的全局指令之后），这些指令专用于特定文件系统，以 *fs =* 开头。特定于文件系统的指令的应用范围到出现下一个 **fs =** 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 **fs=** 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）

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 *sam-releaser* 和 *mount_samfs* 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 */etc/opt/SUNWsamfs/mcf* 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 */var/opt/SUNWsamfs/catalog/historian* 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 */etc/opt/SUNWsamfs/hosts.fs-name*。*sammkfs* 命令会在创建文件系统时使用该 hosts 文件。以后，您可以使用 *samsharefs* 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 (*.inodes*)，其中包含驻留在文件系统中的所有文件的 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在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 *sam-releaser* 和 *mount_samfs* 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（例如固态磁盘或镜像设备）。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 *mcf* 文件中，元数据设备被声明为 *ma* 文件系统中的 *mm* 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 *mount* 命令中的 *-o reader* 选项指定多个读取

	器。使用 *mount* 命令中的 *-o writer* 选项指定单写入器主机。有关更多信息，请参见 *mount_samfs* 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（离线存储）	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	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 *setfa* 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、高容量 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。请参见 **samfsdump (qfsdump)**、**samfsrestore (qfsrestore)**。

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磁带空间。

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 *regex* 和 *regcmp* 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磁带（如磁带或光盘磁带）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磁带。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

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 *-u* 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 *qfsdump*。

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 *stripe=0* 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）	文件系统中的设备集合；在 *mcf* 文件中，它被定义成一个或多个 *gXXX* 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 *stripe=x* 挂载参数。硬分散读写在设置文件系统时启用，需要在 *mcf* 文件内定义分散读写组。软分散读写通过 *stripe=x* 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 *stripe=0* 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。*SUNW.qfs* 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）	文件系统中的一种数据结构，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）

`vfstab` 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 */etc/vfstab* 文件中指定的挂载选项，但是 */etc/vfstab* 文件中指定的挂载选项会覆盖 *samfs.cmd* 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（超过单个磁带的容量）的站点，卷溢出功能非常有用。

volume serial number (VSN)（卷序列号）

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（盒式磁带）**。

volume（卷）

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。存储磁盘和盒式磁带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（盒式磁带）**。
3. 在随机访问磁盘设备上，被当做顺序访问且可移除介质磁带（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

S

W