

Oracle® Hierarchical Storage Manager and StorageTek QFS Software

保守および管理ガイド

リリース 6.1

E56773-03

2016 年 3 月

Oracle® Hierarchical Storage Manager and StorageTek QFS Software
保守および管理ガイド

E56773-03

Copyright © 2011, 2016, Oracle and/or its affiliates. All rights reserved.

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクルまでご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

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

このソフトウェアまたはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアまたはハードウェアは、危険が伴うアプリケーション (人的傷害を発生させる可能性があるアプリケーションを含む) への用途を目的として開発されていません。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性 (redundancy)、その他の対策を講じることは使用者の責任となります。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用したことに起因して損害が発生しても、Oracle Corporation およびその関連会社は一切の責任を負いかねます。

Oracle および Java はオラクルおよびその関連会社の登録商標です。その他の社名、商品名等は各社の商標または登録商標である場合があります。

Intel、Intel Xeon は、Intel Corporation の商標または登録商標です。すべての SPARC の商標はライセンスをもとに使用し、SPARC International, Inc. の商標または登録商標です。AMD、Opteron、AMD ロゴ、AMD Opteron ロゴは、Advanced Micro Devices, Inc. の商標または登録商標です。UNIX は、The Open Group の登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。適用されるお客様と Oracle Corporation との間の契約に別段の定めがある場合を除いて、Oracle Corporation およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。適用されるお客様と Oracle Corporation との間の契約に定めがある場合を除いて、Oracle Corporation およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

目次

はじめに	15
ドキュメントのアクセシビリティについて	15
このドキュメントを使用するための前提条件	15
表記規則	15
入手可能ドキュメント	16
1. Oracle HSM 解決法の保守	17
2. ファイルシステム操作のモニタリング	19
Oracle HSM Manager	19
samu	21
ログとトレースファイル	22
3. Oracle HSM ファイルシステムの管理	23
Oracle HSM ファイルシステムの管理	23
ファイルシステム割り当ての管理	23
ユーザー、グループ、および組織単位のストレージ要件の定義	25
複数のグループで使用されるプロジェクト用およびディレクトリ用の管理セットの作成	29
割り当てを使用するための新しいファイルシステムの構成	31
割り当てを使用するための既存ファイルシステムの構成	34
グループ、プロジェクト、ディレクトリ、およびユーザー用の割り当ての設定	39
整合性のない割り当ての修復	42
割り当ての確認	44
ファイルシステム管理者としての割り当てのモニタリング	44

ユーザー独自の割り当てのモニタリング	46
猶予期間の一時的な延長または取り消し	47
指定した分の猶予期間の延長	47
猶予期間の再開	49
猶予期間の早期終了	52
新しいリソース割り当ての停止	54
ファイルシステムの割り当ての削除	57
アーカイブおよびステージング操作の制御	59
アーカイブおよびステージングプロセスのアイドル状態	59
アーカイブおよびステージングプロセスの停止	61
アーカイブおよびステージングプロセスの再起動	62
ファイルシステム名の変更	63
ファイルシステム名の変更	63
ファイルシステムの修復	66
ファイルシステムの修復	66
ファイルシステムへのデバイスの追加	68
マウントしたファイルシステムへのデバイスの追加	68
共有ファイルシステムに追加された新しいデバイスの構成 の完了	73
アンマウントしたファイルシステムへのデバイスの追加	76
ファイルシステムからのデータデバイスの削除	79
ファイルシステムメタデータおよびデータがバックアップされ ていることの確認	80
samexplorer の実行	80
ファイルシステム用の回復ポイントファイルの作成	81
マウントした高パフォーマンスのファイルシステムからのデバ イスの削除	83
Oracle HSM 共有ファイルシステムの管理	88
共有ファイルシステムのマウントとマウント解除	88
共有ファイルシステムのマウント	89
共有ファイルシステムのアンマウント	91
共有ファイルシステムのホスト構成の変更	93

追加ファイルシステムクライアントの構成	93
共有ファイルシステム構成へのホスト情報の追加	93
Solaris クライアントでの共有ファイルシステムの構成	96
Solaris ホストでの共有ファイルシステムのマウント	101
Linux クライアントホストでの共有ファイルシステムの構成	103
Linux クライアントホストでの共有ファイルシステムのマウント	106
共有ファイルシステム構成からのホストの削除	109
ファイルシステムの hosts ファイルからのホストの削除	109
分散テープ入出力用のデータムーバークライアントの構成	113
データムーバークライアントの構成	113
永続的なバインドを使用したテープドライブの接続	118
ハードウェア構成への変更を反映させるための永続的なバインドの更新	119
リムーバブルメディアデバイスへの新しいファイルシステムホストの永続的なバインド	123
アクティブなメタデータサーバーから潜在的なメタデータサーバーへの切り替え	127
障害のあるアクティブなメタデータサーバーを交換するための潜在的なメタデータサーバーのアクティブ化	127
正常でアクティブなメタデータサーバーを交換するための潜在的なメタデータサーバーのアクティブ化	129
非共有ファイルシステムから共有ファイルシステムへの変換	130
アクティブおよび潜在的なメタデータサーバーでのホストファイルの作成	131
非共有ファイルシステムの共有とクライアントの構成	135
ローカル hosts ファイルを使用したホストネットワーク通信のルーティング	138
共有ファイルシステムから非共有ファイルシステムへの変換	141
共有メタデータサーバーの非共有システムへの変換	142
4. ファイルおよびディレクトリの管理	147

Oracle HSM ファイル属性の設定	147
デフォルトのファイル属性値の復元	148
ファイルシステム領域の事前割り当て	148
ファイルにラウンドロビン式またはストライプ化割り当てを指定	150
指定したストライプグループデバイスでのファイルストレージの割り 当て	151
拡張ファイル属性の使用	152
大容量ファイルの格納	152
非常に大きいファイルでのディスクキャッシュの管理	153
ファイルのセグメント化	153
ファイルのセグメント化	154
複数のボリュームにわたるセグメント化ファイルのストライプ 化	155
大規模なデータセットにリムーバブルメディアファイルを使用	157
リムーバブルメディアファイルまたはボリュームオーバーフ ローファイルの作成	158
リムーバブルメディアファイルとしての外部テープボリューム の読み取り	159
Linear Tape File System (LTFS) ボリュームの操作	160
ライブラリへの LTFS メディアのインポート	160
LTFS ディレクトリおよびファイルの Oracle HSM ファイルシステム への接続	160
要求時の LTFS ファイルのアクセス可能化	161
ディスクキャッシュで LTFS ファイルを即時にアクセス可能化	162
Oracle HSM ソフトウェアを使用した LTFS メディアへのアクセス	163
テープドライブへの LTFS ボリュームのロードおよび LTFS ファ イルシステムのマウント	163
LTFS ファイルシステムのアンマウントおよびテープドライブか らのボリュームのアンロード	164
Oracle HSM ソフトウェアを使用した LTFS メディアの管理	165
ボリュームを LTFS ファイルシステムとしてフォーマット	165
LTFS データの消去、およびボリュームからの LTFS フォーマッ トとパーティションの削除	166

LTFS ファイルシステムの整合性の検査	166
LTFS 構成およびステータス情報の表示	167
SMB/CIFS 共有でのディレクトリおよびファイルの管理	167
SMB/CIFS 共有でのシステム属性の管理	167
Oracle HSM でサポートされるシステム属性	168
システム属性の表示	169
システム属性の変更	169
アクセス制御リストの管理	169
5. ライブラリ、メディア、およびドライブの管理	171
自動メディアライブラリの管理	171
ライブラリのオンライン化およびオフライン化	171
ライブラリのオフライン化	171
ライブラリのオンライン化	172
リムーバブルメディアのインポートおよびエクスポート	173
リムーバブルメディアカートリッジのインポート	174
リムーバブルメディアカートリッジのエクスポート	174
ライブラリカタログの保持	176
ライブラリカタログの表示	176
ライブラリスロットの内容の監査	179
直接接続自動ライブラリ全体の監査	180
カタログからのメディアエラーのクリア	180
ヒストリアンカタログの管理	183
ヒストリアンカタログの表示	184
ヒストリアンカタログへのエントリの追加	184
ヒストリアンカタログからのエントリの削除	185
ヒストリアン情報の更新	185
ドライブをライブラリに取り付ける順序の判別	186
ライブラリと Solaris ホストのドライブ情報の収集	186
直接接続ライブラリ内のドライブを Solaris デバイス名にマッ プ	187

ACSLS 接続ライブラリ内のドライブを Solaris デバイス名にマッ プ	189
ドライブの管理	192
ドライブのロードおよびアンロード	192
自動ライブラリに取り付けられたドライブのロードおよびアン ロード	192
指定したライブラリ位置からのドライブのロード	193
指定したメディアタイプおよびボリュームシリアル番号の ドライブのロード	193
ライブラリで指定したドライブのアンロード	193
スタンドアロンドライブの手動でのロードおよびアンロード	193
スタンドアロンドライブへのカートリッジのロード	194
スタンドアロンドライブからのカートリッジのアンロー ド	194
ボリュームを手動でロードする必要があるときにオペレータに 通知	194
ロード通知の有効化	194
テープドライブのクリーニング	197
十分なクリーニングカートリッジの提供	197
自動テープドライブクリーニングの有効化 (推奨)	199
テープドライブの手動によるクリーニング	200
暗号化機能のあるドライブの使用	201
ドライブの問題の処理	201
保守または修復のためにドライブをオフラインにする	201
ドライブの問題のあとでメディアをライブラリに戻す	202
自動監査を実行していないライブラリにメディアに戻す	202
自動監査後のライブラリへのメディアの返却	203
リムーバブルメディアの管理	203
リムーバブルメディアのラベル付け	203
バーコードからのラベルの生成	204
新しいテープのラベル付けまたは既存のテープの再ラベル付 け	206

新しい光磁気ディスクのラベル付けまたは既存の光磁気ディスクの再ラベル付け	208
データの整合性の維持	210
Data Integrity Validation (DIV) 設定およびステータスの表示	210
DIV 設定の表示	211
アーカイブファイルの書き込みステータス後の検証のモニター	211
デバイスの書き込みステータス後の検証のモニター	211
特定のテープボリュームの整合性のチェック	211
ライブラリ位置で指定されたテープ上のデータの検証	212
メディアタイプおよびボリュームシリアル番号で指定されたテープ上のデータの検証	213
指定したドライブを使用したテープ上のデータの検証	213
テープの先頭からデータの検証を再開	213
T10000C/D テープ上にあるすべてのブロックの ECC の検証	214
T10000C/D テープ上にあるすべてのブロックの ECC および DIV チェックサムの検証	215
T10000C/D テープのメディア情報領域 (MIR) の再構築	215
指定したテープのデータ検証のキャンセル	216
テープの DIV ステータスおよび検証の進行状況の表示	216
自動整合性検証のモニタリング	217
verifyd.cmd 構成ファイルの表示および検証	217
verifyd.cmd 構成ファイルのリロード	218
定期的なメディア検証のテープの不具合データベースに一覧表示されたすべての不具合の表示	218
特定のボリュームに対して一覧表示された不具合の表示	218
定期的なメディア検証のテープの不具合データベースに一覧表示された不具合のクリア	219
6. デジタル保存のためのアーカイブの管理	221
保存用ファイルシステムの構成	222
メッセージダイジェスト (チェックサム) の使用	222

ファイルシステムホストのパフォーマンスが適切であることの確認	224
メッセージダイジェストの指定およびファイルの検証の有効化	225
メッセージダイジェストの生成およびファイルの検証の有効化	229
メッセージダイジェストの生成およびディレクトリ内にある各ファイルの検証の有効化	232
ステージング中のファイルのメッセージダイジェストの検証	235
ファイルがアーカイブされる前のメッセージダイジェストと検証属性の変更	236
ファイルを不変にする	239
メッセージダイジェストを指定してファイルを不変にする	239
メッセージダイジェストを生成してファイルを不変にする	240
ファイルのダイジェストと fixity 属性のチェック	242
メッセージダイジェストと検証属性の一覧表示	242
WORM ファイルシステムの使用	243
WORM ファイルシステムの理解	243
ディレクトリの WORM 対応化	245
ファイルの WORM 保護のアクティブ化	247
WORM ファイルの検索および一覧表示	249
7. 構成およびファイルシステムのバックアップ	253
ファイルシステムのバックアップ	254
回復ポイントおよびアーカイブログの理解	254
要求に応じた回復ポイントの作成	255
アーカイバログのバックアップ	257
Oracle HSM 構成のバックアップ	258
Oracle HSM 構成の手動バックアップ	259
samexplorer を使用した構成および診断情報の収集	261
samexplorer の実行	262
8. 新しいストレージメディアへの移行	265
移行の準備	266
ファイルシステムが常にバックアップされていることの確認	266

必要なメディアの用意	267
移行方式の選択	267
ニーズを満たす最適な移行方式の選択	267
ボリューム全体の移行	269
migrationd.cmd ファイルの作成	269
アクティブな移行ジョブの確認	278
ボリュームの移行	280
ファイルのステージングおよび交換用メディアへの再アーカイブ	286
使用可能なリソースの評価	286
新しいメディアを使用するためのアーカイブ処理の構成	287
交換用メディアへのデータの移行	289
別のカートリッジへのデータの移行	290
移行後の古いメディアの処分	295
A. 装置タイプの用語集	297
推奨される装置およびメディアのタイプ	297
その他の装置タイプとメディアタイプ	299
B. メディアステータスフラグ	303
C. 共有ファイルシステムでのマウントオプション	305
共有ファイルシステムのマウントオプション	305
bg : バックグラウンドでのマウント	305
retry : ファイルシステムのマウントの再試行	305
shared : Oracle HSM 共有ファイルシステムの宣言	306
minallocsz および maxallocsz : 割り当てサイズの調整	306
rdlease 、 wrlease 、および aplease : Oracle HSM 共有ファイルシステムでのリースの使用	306
mh_write : マルチホスト読み取りと書き込みの有効化	307
min_pool : 並行スレッドの最小数の設定	308
meta_timeo : キャッシュされた属性の保持	309
stripe : ストライプ化割り当ての指定	309

sync_meta : メタデータが書き込まれる頻度の指定	309
worm_capable および def_retention : WORM 機能の有効化	310
D. アーカイブのための構成ディレクティブ	311
アーカイブディレクティブ	311
グローバルアーカイブディレクティブ	312
archivemeta : メタデータをアーカイブするかどうかの制御	312
archmax : アーカイブファイルサイズの制御	313
bufsize : アーカイババッファサイズの設定	313
drives : アーカイブに使用するドライブ数の制御	314
examine : アーカイブスキャンの制御	315
interval : アーカイブ間隔の指定	315
logfile : アーカイバログファイルの指定	316
notify : イベント通知スクリプトの名前変更	317
ovflmin : ボリュームオーバーフローの制御	317
scanlist_squash : スキャンリストの連結の制御	318
setarchdone : archdone フラグ設定の制御	318
wait : アーカイバ起動の遅延	319
ファイルシステムディレクティブ	319
fs : ファイルシステムの指定	319
copy-number [archive-age] : ファイルシステムメタデータの複 数コピーの指定	320
ファイルシステムディレクティブとしての interval 、 logfile 、 scanlist	320
archive-set-name : アーカイブセット割り当てディレクティブ	321
アーカイブのコピーディレクティブ	322
コピーパラメータ	324
ボリュームシリアル番号 (VSN) プールディレクティブ	330
ボリュームシリアル番号 (VSN) 関連付けディレクティブ	331
ステージングディレクティブ	332
stager.cmd ファイル	332
drives : ステージングに使用するドライブ数の指定	333

bufsize: ステージングバッファサイズの設定	334
logfile: ステージングログファイルの指定	334
maxactive: ステージング要求数の指定	336
copysel: ステージング時のコピー選択順序の指定	337
プレビュー要求ディレクティブ	337
グローバルディレクティブ	338
vs_n_priority: ボリューム優先順位の調整	338
age_priority: キュー内で待機する時間に応じた優先順位の調整	338
グローバルディレクティブまたはファイルシステム固有のディレクティブ	339
hwm_priority: ディスクキャッシュがほぼ満杯になったときの優先順位の調整	339
hwm_priority: ディスクキャッシュがほぼ空になったときの優先順位の調整	340
lhwm_priority: ディスクキャッシュが満杯になったときの優先順位の調整	340
hlwm_priority: ディスクキャッシュが空になったときの優先順位の調整	340
preview.cmd ファイルのサンプル	341
E. 製品のアクセシビリティ機能	343
用語集	345
索引	359

はじめに

このドキュメントでは、Oracle Hierarchical Storage Manager (旧 StorageTek Storage Archive Manager) and Oracle StorageTek QFS Software を使用してファイルシステムをモニター、管理、および保守するシステム管理者、ストレージおよびネットワーク管理者、サービスエンジニアに必要な事項およびアーカイブ解決方法について説明します。

ドキュメントのアクセシビリティについて

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility Program の Web サイト (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>) を参照してください。

Oracle Support へのアクセス

サポートをご契約のお客様には、My Oracle Support を通して電子支援サービスを提供しています。詳細情報は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>) か、聴覚に障害のあるお客様は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>) を参照してください。

このドキュメントを使用するための前提条件

このドキュメントは、Oracle Solaris オペレーティングシステム、ディスクおよびテープストレージシステム、およびローカルエリアネットワークとストレージエリアネットワークの両方の管理に精通している読者を対象としています。関連するタスク、コマンド、および手順については、Solaris のドキュメントとマニュアルページ、およびストレージハードウェアのドキュメントを参照してください。

表記規則

このドキュメントでは、次の表記規則が使用されています。

- **イタリック**は、ドキュメントのタイトルおよび強調を表します。
- **等幅**は、コマンド、端末ウィンドウに表示されるテキスト、および構成ファイル、シェルスクリプト、ソースコードのファイル内容を表します。
- **等幅太字**は、ユーザー入力、コマンド行出力内の重要な変更、端末表示またはファイル内容を表します。ファイルまたは表示上で特に関連性の高い部分を強調する場合にも使用されます。

- **等幅太字斜体**は、端末表示またはファイル上の変数の入力または出力を表します。
- **等幅斜体**は、端末表示またはファイル上のその他の変数を表します。
- ... (3点省略記号) は、例とは関係のないため、簡潔性および明確性を高めるために省略されたファイル内容やコマンド出力を表しています。
- / (バックスラッシュ) が例内の行の末尾で使用されている場合、それは改行を回避するため、コマンドが次の行に続くことを表します。
- [-] (ハイフンで区切られた値を囲む大括弧) は値の範囲を区切ります。
- [] (大括弧) がコマンド構文の説明で使用されている場合は、オプションのパラメータであることを表します。
- `root@solaris:~#` および `[hostname]:root@solaris:~#` は、Solaris コマンドのシェルプロンプトを表します。
- `[root@linux ~]#` は、Linux コマンドシェルプロンプトを表します。

入手可能ドキュメント

『Oracle Hierarchical Storage Manager and StorageTek QFS Software 保守および管理ガイド』は、複数巻で構成されている『Oracle HSM お客様向けドキュメントライブラリ』(<http://docs.oracle.com/en/storage/#sw> から入手可能)の一部です。

Oracle Solaris オペレーティングシステムのドキュメントは、<http://docs.oracle.com/en/operating-systems/> で入手できます。

システム要件、新機能、およびバグ修正については、リリースノート (ダウンロード ZIP ファイル内の `README.txt`、またはファイルシステムサーバー上の `/opt/SUNWsamfs/doc/README.txt`) を参照してください。

第1章 Oracle HSM 解決法の保守

計画プロセスとデプロイメントプロセス全体を通じて、QFS ファイルシステムおよび Oracle Hierarchical Storage Manager software は、パフォーマンスの最適化、データ保護、およびアーカイブ処理の複雑性を単純な UNIX ファイルシステムインタフェースの背後に隠すように設計されていることを強調してきました。ユーザー、アプリケーション、およびほとんどの場合、管理者は、ディスクアレイとテープライブラリの混合に実装され、完全に最適化された Oracle HSM アーカイブ処理システムを、単一のローカルディスク上にある通常の UFS ファイルシステムと同様に取り扱うことができるべきです。Oracle HSM ソフトウェアは、インストールして構成したあとは、データとストレージリソースを可能なかぎりもっとも効率的かつ信頼できる方法で自動的に管理するはずであり、人間の介入は最小限で済みます。

とは言っても、UNIX ファイルシステムと同様に、引き続きモニタリングタスクと定期的な管理タスクを実行する必要があります。このマニュアルでは、これらのアクティビティの概要について説明します。

第2章 ファイルシステム操作のモニタリング

Oracle HSM ファイルシステムは正しく構成されていれば、日常的な管理者による介入はほとんど必要ありません。ただし、各システムに異常がないかどうかをモニターする必要はあります。一般に、可用性と利用率の2点についてモニターします。

可用性は概念的に単純であるため、簡単にモニターできます。ホストシステム、ネットワークインタフェース、ファイルシステム、ストレージサブシステムなどの主要なコンポーネントが使用不可になると、中心的な機能が突然失われたり、低下したりし、管理インタフェースおよびログに警告が表示されます。

利用率の算出は難しく、ユーザー側での判断がより多く必要になります。使用率が高くなり過ぎると、ストレージメディアなどのリソースが不足することで、システムが通常通りに機能しているにもかかわらず、アーカイブプロセスが停止し、事実上コンポーネントの故障と同じ状態になる可能性があります。ある組織では差し迫ったトラブルの前兆となる利用率レベルが、別の組織では何年間もトラブルなしで操作できるレベルと完全に一致する場合があります。したがって、利用率をモニターする際には、傾向と比率を認識することが重要となります。利用率が1年あたり1%増加する場合は、リソースが80%使用されていても問題ありませんが、毎週の増加率が1%を超える場合は危機的な状況です。

Oracle HSM には、次の3つのモニタリングインタフェースが備わっています。

- [Oracle HSM Manager](#)
- [samu](#)
- [ログとトレースファイル](#)

作業スタイルや習慣に応じて、それぞれの長所があります。

Oracle HSM Manager

Oracle HSM Manager は、管理者がファイルシステム操作のあらゆる点についてモニターおよび制御できるブラウザベースのグラフィカルユーザーインタフェースで

す。Oracle HSM Manager のブラウザインタフェースページは、3つのセクションに分割されています。

- バナー
- ナビゲーションツリー
- コンテンツペイン

バナーには、アプリケーションの名前とステータス情報 (ブラウザインタフェースでデータが最後に更新された時間など)、現在ログインしているユーザーの名前と役割、Oracle HSM Manager ソフトウェアをホストしている管理ステーションの名前、および未確認の障害の現在の数とタイプが表示されます。

インタフェースの左側にあるナビゲーションツリーには、サーバーメニューおよび使用可能な表示の階層一覧が表示されます。ナビゲーションツリー内のリンクをクリックすると、対応する表示がコンテンツペインに表示されます。

ナビゲーションツリーの「モニタリング」ノードは、主要なモニタリングリソースです。モニター対象のファイルシステムおよび装置上で検出されたすべての障害を一覧表示およびソートできます。自動電子メール警告を構成できます。現在実行中のすべてのジョブを一覧表示できます。「モニタリング」ノードには、モニタリングダッシュボードへのリンク、問題を警告するポップアップウィンドウ、次の関心分野に関するサマリー情報へのクイックリンクも含まれています。

- デーモン
- ファイルシステム
- アーカイブメディアの利用率
- テープライブラリ
- ライブラリドライブ
- ロード要求保留中のボリューム
- 使用不可のボリューム
- アーカイブコピーキュー
- ステージングキュー

ナビゲーションツリーの「メトリック」および「レポート」ノードでは、包括的な範囲のステータスおよび利用率レポート、および「システムの詳細」へのリンクが提供されます。「システム詳細」ページを使用すると、Oracle HSM の構成をすばや

く確認したり、ログおよびトレースファイルにすばやく便利にアクセスしたりできます。

Oracle HSM Manager の使用については、包括的なヘルプシステムで詳細に説明されています。

samu

samu オペレータユーティリティーは、コマンド行から起動するテキストベースで、メニュー駆動型の構成および管理用インタフェースです。Oracle HSM デバイス、ファイルシステムアクティビティー、およびエラーメッセージをモニターするための便利で、簡単な方法です。

samu ユーティリティーには、UNIX *vi* エディタと似ている点はいくつかあります。同様の制御キーシーケンスを使用して、表示の選択、表示オプションの設定、表示内および表示間のナビゲーション、コマンドの入力、表示のリフレッシュ、およびユーティリティーの終了を行います。各表示ウィンドウの最終行には、エラーメッセージが表示されます。エラーが発生しなければ、表示は自動的にリフレッシュされます。エラーが発生した場合は、オペレータが追加アクションを取るまで表示が停止します。必要に応じて、あとで参照するために、表示ウィンドウのスナップショットを取得できます。

h コマンドは、すべてのキーボードショートカット、コマンド、およびパラメータを一覧表示したヘルプ画面を開きます。追加情報については、*samu* のマニュアルページおよび *Oracle HSM* お客様向けドキュメントライブラリ (<http://docs.oracle.com/en/storage/#sw>) の『*Oracle Hierarchical Storage Manager and StorageTek QFS samu* コマンドリファレンス』も参照できます。

samu による一般的なモニタリングは、次のように表示されます。

```

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

sam-archiverd: Waiting for resources

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

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

                                samu on samqfshost1

```

ログとトレースファイル

Oracle HSM ソフトウェアでは、包括的なロギングおよびトレース (構成されている場合) が実行されます。そのため、特に問題が発生したときに、次のファイルをモニターすることがあります。

- `/var/adm/messages`
- `/var/adm/sam-log`
- `/var/opt/SUNWsamfs/trace/` (デーモンおよびプロセスのトレースファイルを保持)
- `/var/opt/SUNWsamfs/devlog/` (`/etc/opt/SUNWsamfs/mcf` ファイルに構成されているデバイスデバイスのログを保持)
- `/var/opt/SUNWsamfs/archiver.log`
- `/var/opt/SUNWsamfs/stager.log`
- `/var/opt/SUNWsamfs/recycler.log`
- ファイルシステムに固有の追加アーカイブログ (構成されている場合)。

ロギングおよびトレースの構成の詳細は、*Oracle HSM* お客様向けドキュメントライブラリ (<http://docs.oracle.com/en/storage/#sw>) の『Oracle Hierarchical Storage Manager and StorageTek QFS インストールおよび構成ガイド』を参照してください。

第3章 Oracle HSM ファイルシステムの管理

この章では、ファイルシステムの保守および再構成タスクについて説明します。最初のセクション[Oracle HSM ファイルシステムの管理](#)では、すべての Oracle HSM ファイルシステム (アーカイブと非アーカイブ、共有と非共有 (スタンドアロン)) の保守を扱います。2つ目のセクション[Oracle HSM 共有ファイルシステムの管理](#)では、共有ファイルシステムに影響を与える特殊な考慮事項を扱います。

Oracle HSM ファイルシステムの管理

このセクションでは、次のタスクの概要について説明します。

- [ファイルシステム割り当ての管理](#)
- [アーカイブおよびステージング操作の制御](#)
- [ファイルシステム名の変更](#)
- [ファイルシステムの修復](#)
- [ファイルシステムへのデバイスの追加](#)
- [ファイルシステムからのデータデバイスの削除](#)

ファイルシステム割り当ての管理

特定のユーザーまたはユーザーの集合がファイルシステム内で消費できるオンラインの合計ストレージ領域を制御するように、ファイルシステム割り当てを設定します。ユーザー ID、グループ ID、または特定のプロジェクトへの参加などの共通する特性別にユーザーをグループ化する、管理者が定義した管理セット ID 別に割り当てを設定できます。管理セット ID は、プロジェクトに複数のグループのユーザーが含まれる場合や、プロジェクトが複数のディレクトリおよびファイルにまたがる場合に特に便利です。

割り当てを有効にするには、*quota* マウントオプション (デフォルト設定) を使用してファイルシステムをマウントし、割り当てを無効にするには、*noquota* マウン

トオプションを使用してマウントします。割り当ては、ファイルシステムのルートディレクトリに1つ以上の割り当てファイル置いて定義します。割り当てファイル `.quota_u`、`.quota_g`、および `.quota_a` は、それぞれユーザー、グループ、および管理セットの割り当てを設定します。各ファイルの最初のレコード (レコード 0) には、デフォルト値が設定されます。後続のレコードには、特定のユーザー、グループ、または管理セットに固有の値が設定されます。

割り当てによって、ストレージ領域だけでなく、使用可能なファイルシステム領域も割り当てられます。したがって、メディアに割り当てられた 512 バイトのブロック数と、ファイルシステムに割り当てられた i ノード数の両方に上限が設定されます。ブロック数で、ストレージ領域自体が測定されます。i ノード数で、そのストレージにアクセスできるリソースが測定されます。したがって、非常に多くのブロックのストレージ領域を使用している i ノードは1つしか使用しない単一のファイルと、多数の i ノードを使用し、ブロックがなく空で長さゼロの非常に多くのファイルが使用するファイルシステム領域の量は同じです。

各割り当てには、弱い制限値と強い制限値の両方を含めることができます。強い制限値には、特定の所有者のすべてのファイルが一時的に使用できるファイルシステムリソースの最大数を定義します。弱い制限値には、所有者のファイルが無制限に使用できるファイルシステムリソースの最大数を定義します。割り当ての猶予時間で定義されているように、リソースの使用量は、ほんの短期間で弱い制限値と強い制限値の間の量にまで増加する可能性があります。

このセクションでは、次の管理タスクについて説明します。

- ユーザー、グループ、および組織単位のストレージ要件の定義
- 複数のグループで使用されるプロジェクト用およびディレクトリ用の管理セットの作成
- 割り当てを使用するための新しいファイルシステムの構成
- 割り当てを使用するための既存ファイルシステムの構成
- グループ、プロジェクト、ディレクトリ、およびユーザー用の割り当ての設定
- 整合性のない割り当ての修復
- 割り当ての確認
- 猶予期間の一時的な延長または取り消し
- 新しいリソース割り当ての停止。

ユーザー、グループ、および組織単位のストレージ要件の定義

持続的な割り当てを設定するには、ユーザーの要件に対応する制限値を、管理可能かつ拡張可能な方法で設定する必要があります。したがって、割り当てを設定する前に、ユーザーのストレージ要件を見積もります。プロセスを管理可能に保つには、最小限の管理労力で最大数の要件に対処できるように、ユーザーの要件をできるだけ幅広く分類することから始めます。その後、幅広いカテゴリに当てはまらない少数のユーザー要件を特別に評価できます。結果には、設定する割り当ての概要と制限値のタイプが示されます。

ほとんどの組織ではアクセス制御グループがすでに定義されているため、次に示すアプローチでは、これらのグループのファイルシステム要件を特定することから始めます。次に、要件が標準グループと一致しない特別なユーザーのセットを定義します。この時点ではじめて、個別のユーザーに固有の要件に対処し始めます。次のように進めます。

1. 既存のアクセス制御グループが同様のリソース要件を持つユーザーをすでに収集しているため、ファイルシステムを使用するグループの平均的なストレージ要件を定義することから始めます。使用されるストレージ領域の平均量 (512 キロバイトブロック単位) と、格納されるファイルの平均数 (使用される i ノードの平均数と同等) の両方を見積もります。

通常、グループメンバーは同様の組織上の役割や業務上の責任を持っているため、同じディレクトリやファイルに頻繁にアクセスする必要があり、一般に、ストレージに対して同様の要求を行います。この例では、ファイルシステム `/hsm/hsmfs1` を使用する 3 つのグループ `dev` (製品開発)、`cit` (会社情報技術)、`pgmt` (プログラム管理) を特定します。次の簡単なスプレッドシートに、グループ、それぞれのメンバー数、個人およびグループの平均的な要件を一覧表示します。

グループ	ユーザー	1 ユーザーあたりの平均ブロック数	1 ユーザーあたりの平均ファイル数	1 グループあたりの平均ブロック数	1 グループあたりの平均ファイル数
dev	30	67108864	500	2013265920	15000
cit	15	10485760	50	157286400	750
pgmt	6	20971520	200	125829120	1200
合計ブロック数/ファイル数 (平均)					

- 次に、グループメンバーが常時格納するストレージ領域の最大量およびファイルの最大数について、同じ計算を実行します。結果を記録します。

この例では、新しいスプレッドシートに結果を記録します。

グループ	ユーザー	1ユーザーあたりの最大ブロック数	1ユーザーあたりの最大ファイル数	1グループあたりの最大ブロック数	1グループあたりの最大ファイル数
dev	30	100663296	1000	3019898880	30000
cit	15	15728640	100	235929600	1500
pmgt	6	31457280	400	188743680	2400
合計ブロック数/ファイル数(最大)					

- ここで、さまざまなグループに属するが、グループメンバーシップに基づいて対処できない個別のストレージ要件を共有するユーザーのセットを特定します。特定された組織ごとに、各アクセス制御グループに対して行なったときと同じ見積もりと同じ計算を実行します。

この例では、ストレージの割り当てが必要な2つの会社プロジェクト(コード名 *portal* と *lockbox*)を特定します。これらのプロジェクトでは、エンジニアリング、マーケティング、コンプライアンス、テスト、およびドキュメントの各グループメンバーが共同で作業し、同じディレクトリや多数の同じファイルを使用します。したがって、これらのプロジェクトを要件スプレッドシートに追加します。

グループ	ユーザー	1ユーザーあたりの平均ブロック数	1ユーザーあたりの平均ファイル数	1グループあたりの平均ブロック数	1グループあたりの平均ファイル数
dev	30	67108864	500	2013265920	15000
cit	15	10485760	50	157286400	750
pmgt	6	20971520	200	125829120	1200
portal	10	31457280	400	314572800	4000
lockbox	12	31457280	500	377487360	6000

グループ	ユーザー	1ユーザーあたりの平均ブロック数	1ユーザーあたりの平均ファイル数	1グループあたりの平均ブロック数	1グループあたりの平均ファイル数
合計ブロック数/ファイル数(平均)					
グループ	ユーザー	1ユーザーあたりの最大ブロック数	1ユーザーあたりの最大ファイル数	1グループあたりの最大ブロック数	1グループあたりの最大ファイル数
dev	30	100663296	1000	3019898880	30000
cit	15	15728640	100	235929600	1500
pmgt	6	31457280	400	188743680	2400
portal	10	37748736	700	377487360	7000
lockbox	12	45613056	600	547356672	7200
合計ブロック数/ファイル数(最大)					

4. ここで、まだ対処していない要件を持つ個別のユーザーを特定します。ユーザーごとに、各アクセス制御グループおよびグループ以外の組織に対して行なったときと同じ見積もりと同じ計算を実行します。

可能な場合は、ユーザー要件にまとめて対処します。これにより、ポリシーが統一され、管理のオーバーヘッドが最小限になります。ただし、個別の要件が固有のものである場合は、個別に対処する必要があります。この例では、特殊なストレージ割り当てが必要となる特殊な責任を持つユーザーとして、*pmgt* グループの *jr23547* を確認します。したがって、このユーザーを要件スプレッドシートに追加します。

グループ	1セットあたりユーザー数	1ユーザーあたりの平均ブロック数	1ユーザーあたりの平均ファイル数	平均ブロック数	平均ファイル数
dev	30	67108864	500	2013265920	15000
cit	15	10485760	50	157286400	750

グループ	1 セットあたり のユーザー数	1 ユーザーあたりの平均ブロック数	1 ユーザーあたりの平均ファイル数	平均ブロック数	平均ファイル数
pmgt	6	20971520	200	125829120	1200
portal	10	31457280	400	314572800	4000
lockbox	12	31457280	500	377487360	6000
jr23547	1	10485760	600	10485760	600
合計ブロック数/ファイル数 (平均)					

グループ	ユーザー	1 ユーザーあたりの最大ブロック数	1 ユーザーあたりの最大ファイル数	1 グループあたりの最大ブロック数	1 グループあたりの最大ファイル数
dev	30	100663296	1000	3019898880	30000
cit	15	15728640	100	235929600	1500
pmgt	6	31457280	400	188743680	2400
portal	10	37748736	700	377487360	7000
lockbox	12	45613056	600	547356672	7200
jr23547	1	100663296	2000	100663296	2000
合計ブロック数/ファイル数 (最大)					

5. 最後に、すべてのユーザーが必要とする平均および最大のブロック数とファイル数を計算します。

グループ	ユーザー	1 ユーザーあたりの平均ブロック数	1 ユーザーあたりの平均ファイル数	1 グループあたりの平均ブロック数	1 グループあたりの平均ファイル数
dev	30	67108864	500	2013265920	15000
cit	15	10485760	50	157286400	750
pmgt	6	20971520	200	125829120	1200

グループ	ユーザー	1ユーザーあたりの平均ブロック数	1ユーザーあたりの平均ファイル数	1グループあたりの平均ブロック数	1グループあたりの平均ファイル数
portal	10	31457280	400	314572800	4000
lockbox	12	31457280	500	377487360	6000
jr23547	1	10485760	600	10485760	600
合計ブロック数/ファイル数(平均)				2998927360	27550

グループ	ユーザー	1ユーザーあたりの最大ブロック数	1ユーザーあたりの最大ファイル数	1グループあたりの最大ブロック数	1グループあたりの最大ファイル数
dev	30	100663296	1000	3019898880	30000
cit	15	15728640	100	235929600	1500
pmgt	6	31457280	400	188743680	2400
portal	10	37748736	700	377487360	7000
lockbox	12	45613056	600	547356672	7200
jr23547	1	100663296	2000	100663296	2000
合計ブロック数/ファイル数(平均)				4470079488	50100

- アクセス制御グループおよびユーザー ID 別に定義できないプロジェクトベースの割り当てやその他の割り当てを管理する必要がある場合は、複数のグループで使用されるプロジェクト用およびディレクトリ用の管理セットを作成します。
- 新たに作成した空のファイルシステム上で割り当てを設定する場合は、「[割り当てを使用するための新しいファイルシステムの構成](#)」に進みます。
- すでにファイルを保持しているファイルシステム上で割り当てを設定する場合は、「[割り当てを使用するための既存ファイルシステムの構成](#)」に進みます。

複数のグループで使用されるプロジェクト用およびディレクトリ用の管理セットの作成

管理セットとは、割り当ての目的で管理セット ID によって特定されるディレクトリ階層または個別のディレクトリやファイルです。実際にファイルを所有するユー

ザー ID やグループ ID とは関係なく、指定した管理セット ID を使用して作成されたファイル、または指定した管理セット ID を含むディレクトリに保存されたファイルは、すべて同じ割り当てを持っています。管理セットを定義するには、次のように進めます。

1. ファイルシステムサーバーに *root* としてログインします。

この例では、サーバーの名前は *server1* です。

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

2. 管理セットを使用して新しいプロジェクトまたはチーム用にストレージの割り当てを構成する場合は、このプロジェクトまたはチーム用のファイルシステム内の任意の場所に、新しいディレクトリを作成します。

この例では、*/hsm/hsmfs1* ファイルシステムにディレクトリを作成し、同じ名前のプロジェクト用に *portalproject/* という名前を付けます

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

3. 割り当てを設定する必要があるディレクトリやファイルに、管理セット ID を割り当てます。コマンド *samchaid [-fhR] admin-set-id directory-or-file-name* を使用します。ここでは:
 - *-f* は、強制的に割り当てを実行し、エラーを報告しません。
 - *-h* は、管理セット ID をシンボリックリンクに割り当てます。このオプションを指定しないと、シンボリックリンクで参照されるファイルのグループが変更されます。
 - *-R* は、管理セット ID をサブディレクトリおよびファイルに再帰的に割り当てます。
 - *admin-set-id* は、一意の整数値です。
 - *directory-or-file-name* は、管理セット ID を割り当てるディレクトリまたはファイルの名前です。

この例では、管理 ID *1* をディレクトリ */hsm/hsmfs1/portalproject/* およびそのサブディレクトリとファイルすべてに割り当てます。

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

- 必要に応じて、割り当てを確認できます。`command sls -D directory-path` を使用します。ここで `-D` を指定すると、`directory-path` 内のファイルとディレクトリに関する詳細な Oracle HSM ディレクトリ一覧が表示されます。

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

- 新たに作成した空のファイルシステム上で割り当てを設定する場合は、「[割り当てを使用するための新しいファイルシステムの構成](#)」に進みます。
- すでにファイルを保持しているファイルシステム上で割り当てを設定する場合は、「[割り当てを使用するための既存ファイルシステムの構成](#)」に進みます。

割り当てを使用するための新しいファイルシステムの構成

これから新しいファイルシステムを作成し、現在はファイルシステムにファイルが常駐していない場合は、この手順を使用します。

- ファイルシステムサーバーに `root` としてログインします。

この例では、サーバーは `server2` という名前です。

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

- 新しいファイルシステムが現在マウントされていない場合は、マウントしてから続行します。
- グループ用の割り当てを設定する必要がある場合は、ファイルシステムのルートディレクトリにグループ割り当てファイルを作成します (`.quota_g`)。Solaris コマンド `dd if=/dev/zero of=mountpoint/.quota_g bs=4096 count=number-blocks` を使用します。ここでは:

- `if=/dev/zero` では、入力として UNIX 特殊ファイル `/dev/zero` からの NULL 文字が指定されます。
- `of=mountpoint/.quota_g` には、出力ファイルを指定します。ここで `mountpoint` はファイルシステム用のマウントポイントディレクトリです。
- `bs=4096` は、書き込み用のブロックサイズが `4096` バイトに設定されます。
- `count=number-blocks` には、書き込まれるブロック数を指定します。この値は、ファイルが保持されるレコードの数によって異なります。指定された割り当てごとに、128 バイトのレコードが 1 つあるため、1 つのブロックに 32 個のレコードを格納できます。

この例では、`/newsamfs` でマウントされたファイルシステム `newsamfs` 用のグループ割り当てファイルを作成します。要件の収集フェーズ中に、ファイルシステム上で割り当てが必要な 3 つのグループ `dev`、`cit`、および `pgmt` を特定しました。その他のグループ割り当てを追加する予定はないため、1 つのブロックでファイルサイズを計算します。

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

4. 管理セット用の割り当てを設定する必要がある場合は、ファイルシステムのルートディレクトリに管理セット割り当てファイルを作成します (`.quota_a`)。Solaris コマンド `dd if=/dev/zero of=mountpoint/.quota_a bs=4096` を使用します。ここでは:

- `mountpoint` は、ファイルシステムのマウントポイントディレクトリです。
- `.quota_a` は、出力ファイルの名前です。
- `4096` はバイト単位の書き込み用のブロックサイズです。
- `number-blocks` は、書き込まれるブロック数です。

この例では、`/newsamfs` でマウントされたファイルシステム `newsamfs` 用の管理セット割り当てファイルを作成します。要件の収集フェーズ中に、ファイルシステム上で割り当てが必要な 2 つのプロジェクト `portal` (管理セット ID 1) と `lockbox` (管理セット ID 2) を特定しました。その他の管理セット割り当てを追加する予定はないため、1 つのブロックでファイルサイズを計算します。

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


5. ユーザー用の割り当てを設定する必要がある場合は、ファイルシステムのルートディレクトリにユーザー割り当てファイル `.quota_u` を作成します。Solaris コマンド `dd if=/dev/zero of=mountpoint/.quota_u bs=4096 count=number-blocks` を使用します。ここでは:

- `mountpoint` は、ファイルシステムのマウントポイントディレクトリです。
- `.quota_u` は、出力ファイルの名前です。
- `4096` はバイト単位の書き込み用のブロックサイズです。
- `number-blocks` は、書き込まれるブロック数です。

この例では、`/newsamfs` でマウントされたファイルシステム `newsamfs` 用のユーザー割り当てファイルを作成します。要件の収集フェーズ中に、ファイルシステム上で特定の割り当てが必要な1つのユーザー `jr23547` を特定しました。その他の個別のユーザー割り当てを追加する予定はないため、1つのブロックでファイルサイズを計算します。

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

6. ファイルシステムをマウント解除します。

ファイルシステムを再マウントし、割り当てファイルを有効にする前に、ファイルシステムをアンマウントする必要があります。

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

7. ファイルシステムチェックを実行します。

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

8. ファイルシステムを再マウントします。

割り当ては、ファイルシステムのルートディレクトリで1つ以上の割り当てファイルが検出されると有効になります。

デフォルトでファイルシステムは割り当てが有効な状態でマウントされるため、`quota` マウントオプションを `/etc/vfstab` または `samfs.cmd` ファイルに含める必要はありません。

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

9. 次に、必要に応じて割り当てを設定または更新します。「[グループ、プロジェクト、ディレクトリ、およびユーザー用の割り当ての設定](#)」を参照してください。

割り当てを使用するための既存ファイルシステムの構成

この手順は、すでにファイルを保持しているファイルシステムに対して割り当てを作成する場合に使用します。

1. ファイルシステムサーバーに *root* としてログインします。

この例では、サーバーの名前は *server1* です。

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

2. テキストエディタで */etc/vfstab* ファイルを開き、*noquota* マウントオプションが設定されていないことを確認します。

この例では、*vi* テキストエディタでファイルを開きます。*noquota* マウントオプションが設定されています。

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

3. *noquota* マウントオプションが */etc/vfstab* ファイルに設定されている場合は、削除してからファイルを保存します。

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

```

#to Mount      to fsck Point          Type   Pass  at Boot  Options
#-----
/devices      -          /devices      devfs  -     no      -
/proc         -          /proc         proc   -     no      -
...
hsmfs1       -          /hsm/hsmfs1   samfs  -     no      -
:wq
[server1]root@solaris:~#

```

4. テキストエディタで `/etc/opt/SUNWsamfs/samfs.cmd` ファイルを開き、`noquota` マウントオプションが設定されていないことを確認します。

この例では、`vi` テキストエディタでファイルを開きます。`noquota` マウントオプションは設定されていません。

```

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

```

5. *noquota* マウントオプションが `/etc/opt/SUNWsamfs/samfs.cmd` ファイルに設定されている場合は、削除してからファイルを保存します。
6. *noquota* マウントオプションを `/etc/vfstab` ファイルまたは `/etc/opt/SUNWsamfs/samfs.cmd` ファイル (あるいは両方) から削除する場合は、ファイルシステムをアンマウントします。

noquota マウントオプションを削除する場合は、割り当てが有効になっている状態で再マウントできるように、ファイルシステムをアンマウントする必要があります。

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

7. ファイルシステムが現在マウントされていない場合は、ここでマウントします。
割り当てを有効にする前に、ファイルシステムをマウントする必要があります。

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

8. ファイルシステムのルートディレクトリに移動して、既存の割り当てファイルを確認します。Solaris コマンド `ls -a` を使用して、`.quota_g`、`.quota_a`、または `.quota_u` ファイル (あるいはすべて) を検索します。

この例では、割り当てファイルは現在存在しません。

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

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

```
.                .archive        .fuid            .stage           portalproject
..               .domain         .inodes         lost+found
```

9. 割り当てファイルが存在する場合は、変更しないでください。
10. グループ用の割り当てを設定する必要がある、ファイルシステムのルートディレクトリにグループ割り当てファイル `.quota_g` が存在していない場合は、ここでファイルを作成します。Solaris コマンド `dd if=/dev/zero of=mountpoint/.quota_g bs=4096 count=number-blocks` を使用します。ここでは:
 - `if=/dev/zero` では、入力として UNIX 特殊ファイル `/dev/zero` からの NULL 文字が指定されます。

- `of=mountpoint/.quota_g` には、出力ファイルを指定します。ここで `mountpoint` はファイルシステム用のマウントポイントディレクトリです。
- `bs=4096` は、書き込み用のブロックサイズが `4096` バイトに設定されます。
- `count=number-blocks` には、書き込まれるブロック数を指定します。この値は、ファイルが保持されるレコードの数によって異なります。指定された割り当てごとに、`128` バイトのレコードが `1` つあるため、`1` つのブロックに `32` 個のレコードを格納できます。

この例では、`/hsm/hsmfs1` でマウントされたファイルシステム `/hsm/hsmfs1` 用のグループ割り当てファイルを作成します。要件の収集フェーズ中に、ファイルシステム上で割り当てが必要な `3` つのグループ `dev`、`cit`、および `pgmt` を特定しました。その他のグループ割り当てを追加する予定はないため、`1` つのブロックでファイルサイズを計算します。

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

11. 管理セット用の割り当てを設定する必要があり、ファイルシステムのルートディレクトリに管理セット割り当てファイル `.quota_a` が存在していない場合は、ここでファイルを作成します。Solaris コマンド `dd if=/dev/zero of=mountpoint/.quota_a bs=4096 count=number-blocks` を使用します。ここでは:

- `mountpoint` は、ファイルシステムのマウントポイントディレクトリです。
- `.quota_a` は、出力ファイルの名前です。
- `4096` はバイト単位の書き込み用のブロックサイズです。
- `number-blocks` は、書き込まれるブロック数です。

この例では、`/hsm/hsmfs1` でマウントされたファイルシステム `/hsm/hsmfs1` 用の管理セット割り当てファイルを作成します。要件の収集フェーズ中に、ファイルシステム上で割り当てが必要な `2` つのプロジェクト `portal` (管理セット ID `1`) と `lockbox` (管理セット ID `2`) を特定しました。その他の管理セット割り当てを追加する予定はないため、`1` つのブロックでファイルサイズを計算します。

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

12. ユーザー用の割り当てを設定する必要があり、ファイルシステムのルートディレクトリにユーザー割り当てファイル `.quota_u` が存在していない場合は、ここ

でファイルを作成します。Solaris コマンド `dd if=/dev/zero of=mountpoint/.quota_u bs=4096 count=number-blocks` を使用します。ここでは:

- `mountpoint` は、ファイルシステムのマウントポイントディレクトリです。
- `.quota_u` は、出力ファイルの名前です。
- `4096` はバイト単位の書き込み用のブロックサイズです。
- `number-blocks` は、書き込まれるブロック数です。

この例では、`/hsm/hsmfs1` でマウントされたファイルシステム `/hsm/hsmfs1` 用のユーザー割り当てファイルを作成します。要件の収集フェーズ中に、ファイルシステム上で特定の割り当てが必要な 1 つのユーザー `jr23547` を特定しました。その他の個別のユーザー割り当てを追加する予定はないため、1 つのブロックでファイルサイズを計算します。

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

13. ファイルシステムをマウント解除します。

ファイルシステムを再マウントし、割り当てファイルを有効にする前に、ファイルシステムをアンマウントする必要があります。

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

14. ファイルシステムチェックを実行します。

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

15. ファイルシステムを再マウントします。

割り当ては、ファイルシステムのルートディレクトリで 1 つ以上の割り当てファイルが検出されると有効になります。

デフォルトでファイルシステムは割り当てが有効な状態でマウントされるため、`quota` マウントオプションを `/etc/vfstab` または `samfs.cmd` ファイルに含める必要はありません。

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

16. 次に、グループ、プロジェクト、ディレクトリ、およびユーザー用の割り当てを設定します。

グループ、プロジェクト、ディレクトリ、およびユーザー用の割り当ての設定

`samquota` コマンドを使用して、新しい割り当てを設定し、既存の割り当てを調整します。次の手順に従います。

1. ストレージ要件を定義したら、グループ、ユーザー、およびグループ以外の組織ごとに適切な割り当てを決定します。次の要素を検討し、必要に応じて調整します。
 - すべてのユーザーが必要とする平均および最大のブロック数と比較したファイルシステムのサイズ
 - すべてのユーザーが必要とする平均および最大の i ノード数と比較したファイルシステム内の i ノード数
 - いつでも最大の要件に近くなる可能性のあるユーザーの数とタイプ。
2. ファイルシステムサーバーに `root` としてログインします。

この例では、サーバーの名前は `server1` です。

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

3. 制限が必要なグループごとに、制限値を設定します。コマンド `samquota -b number-blocks:type[:scope] -f number-files:type[:scope] -t interval[:scope] -G groupID [directory-or-file]` を使用します。ここでは:
 - `-b number-blocks` は、ファイルシステムに格納できる 512K バイトの最大ブロック数を整数値 `number-blocks` に設定します (サイズを指定する代替の方法については、`samquota` のマニュアルページを参照)。値を `0` (ゼロ) にすると、無制限のブロック数が指定されます。
 - `:` はフィールドセパレータです。
 - `type` には、制限値の種類 (強い制限値の場合は `h`、弱い制限値の場合は `s`) を指定します。
 - `scope` (オプション) は、制限対象のストレージのタイプを識別します。オンライン (ディスクキャッシュ) ストレージのみの場合は `o`、ディスクキャッシュと

アーカイブストレージの両方を含むストレージ全体 (デフォルト) の場合は *t* にします。

- *-f number-files* は、ファイルシステムに格納できる最大ファイル数を整数値 *number-files* に設定します。値を 0 (ゼロ) にすると、無制限のファイル数が指定されます。
- *-t number-seconds* は、猶予期間 (弱い制限値を超過できる時間) を秒数を表す整数値 *number-seconds* に設定します (時間を指定する代替の方法については、*samquota* のマニュアルページを参照)。
- *-G groupID* には、グループ名またはグループを表す整数の識別子を指定します。値を 0 (ゼロ) にすると、すべてのグループにデフォルトの制限値が設定されます。
- *directory-or-file* (オプション) は、割り当てを設定する必要のある特定のファイルシステム用または特定のディレクトリやファイル用のマウントポイントディレクトリです。

この例では、要件の収集フェーズでの見積もりを使用して、グループ *dev* が使用できる */hsm/hsmfs1* ファイルシステム内のストレージ領域と格納できるファイル数の両方に、強い制限値と弱い制限値の両方を設定します。オンラインストレージのみの場合は猶予期間を 43200 秒 (12 時間) に設定します (次のコマンドは 1 行で入力し、改行はバックスラッシュ文字でエスケープします)。

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

4. 制限が必要な管理セットごとに、制限値を設定します。コマンド *samquota -b number-blocks:type[:scope] -f number-files:type[:scope] -t interval[:scope] -A adminsetID [directory-or-file]* を使用します。ここで *-A adminsetID* は、管理セットを一意に識別する整数値です。

adminsetID を 0 (ゼロ) に設定すると、すべての管理セットにデフォルトの制限値が設定されます。

この例では、要件の収集フェーズでの見積もりを使用して、*portal* プロジェクト (管理セット ID 1) が使用できる */hsm/hsmfs1* ファイルシステム内のストレージ領域と格納できるファイル数の両方に、強い制限値と弱い制限値の両方を設定します。ストレージ全体を使用する場合、猶予期間はデフォルト範囲である 43200 秒 (12 時間) に設定します (次のコマンドは 1 行で入力し、改行はバックスラッシュ文字でエスケープします)。

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

5. 制限が必要な個別のユーザーごとに、制限値を設定します。コマンド *samquota -b number-blocks:type[:scope] -f number-files:type[:scope] -t interval[:scope] -U userID [directory-or-file]* を使用します。ここで *-U userID* は、ユーザー名またはユーザーを表す整数の識別子です。

userID を 0 (ゼロ) に設定すると、すべてのユーザーにデフォルトの制限値が設定されます。

この例では、要件の収集フェーズでの見積もりを使用して、ユーザー *jr23547* が使用できる */hsm/hsmfs1* ファイルシステム内のストレージ領域と *jr23547* が格納できるファイル数の両方に、強い制限値と弱い制限値の両方を設定します。ストレージ全体を使用する場合、猶予期間はデフォルト範囲である 1209600 秒 (2 週間) に設定します (次のコマンドは 1 行で入力し、改行はバックスラッシュ文字でエスケープします)。

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

6. ここで停止します。

整合性のない割り当ての修復

ルートディレクトリ内に割り当てファイルが存在するときに、*noquota* マウントオプションを使用して Oracle HSM ファイルシステムをマウントする場合、ブロックまたはファイルが割り当てられたり、解放されたりすると、割り当てレコードの整合性がなくなります。このような状況では、次のように進めます。

1. ファイルシステムサーバーに *root* としてログインします。

この例では、サーバーの名前は *server1* です。

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

2. 影響を受けるファイルシステムをアンマウントします。

この例では、ファイルシステム *samfs2* をアンマウントします。

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

3. テキストエディタで */etc/vfstab* ファイルを開き、*noquota* マウントオプションが設定されていないことを確認します。

この例では、*vi* テキストエディタでファイルを開きます。*noquota* マウントオプションが設定されています。

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

4. *noquota* マウントオプションが */etc/vfstab* ファイルに設定されている場合は、削除してからファイルを保存します。

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

5. テキストエディタで `/etc/opt/SUNWsamfs/samfs.cmd` ファイルを開き、`noquota` マウントオプションが設定されていないことを確認します。

この例では、`vi` テキストエディタでファイルを開きます。`noquota` マウントオプションは設定されていません。

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

```
# shared_reader (ma filesystem, default no shared_reader)
```

6. *noquota* マウントオプションが */etc/opt/SUNWsamfs/samfs.cmd* ファイルに設定されている場合は、削除してからファイルを保存します。
7. 整合性のない割り当てレコードを修復します。コマンド *samfsck -F family-set-name* を使用します。ここで *family-set-name* は、*/etc/opt/SUNWsamfs/mcf* ファイル内のファイルシステムのファミリーセット名です。

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

8. ファイルシステムを再マウントします。

割り当ては、ファイルシステムのルートディレクトリで1つ以上の割り当てファイルが検出されると有効になります。

デフォルトでファイルシステムは割り当てが有効な状態でマウントされるため、*quota* マウントオプションを */etc/vfstab* または *samfs.cmd* ファイルに含める必要はありません。

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

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

9. ここで停止します。

割り当ての確認

管理者とユーザーの両方が、割り当ておよびリソース使用量をモニターできます。*root* ユーザーは *samquota* コマンドを使用して、ユーザー、グループ、または管理セットに関する割り当てレポートを生成できます。ファイルシステムユーザーは *squota* コマンドを使用して、独自の割り当てを確認できます。

次の手順を参照してください。

- [ファイルシステム管理者としての割り当てのモニタリング](#)
- [ユーザー独自の割り当てのモニタリング](#)

ファイルシステム管理者としての割り当てのモニタリング

1. ファイルシステムサーバーに *root* としてログインします。

この例では、サーバーの名前は *server1* です。

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

2. すべてのグループの割り当て統計を表示するには、コマンド *samquota -g [directory-or-file]* を使用し、オプションの *directory-or-file* パラメータで、指定されたディレクトリにマウントされたファイルシステム、指定されたディレクトリ自体、または指定されたファイルにレポートの範囲を限定します。

この例では、*/hsm/hsmfs1* にマウントされた *hsmfs1* ファイルシステムのレポートを要求します。

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

3. すべての管理セットの割り当て統計を表示するには、コマンド *samquota -a [directory-or-file]* を使用し、オプションの *directory-or-file* パラメータで、指定されたディレクトリにマウントされたファイルシステム、指定されたディレクトリ自体、または指定されたファイルにレポートの範囲を限定します。

この例では、*/hsm/hsmfs1* にマウントされた *hsmfs1* ファイルシステムのレポートを要求します。

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

4. すべてのユーザーの割り当て統計を表示するには、コマンド *samquota -u [directory-or-file]* を使用し、オプションの *directory-or-file* パラメータで、指定されたディレクトリにマウントされたファイルシステム、指定されたディレクトリ自体、または指定されたファイルにレポートの範囲を限定します。

この例では、*/hsm/hsmfs1* にマウントされた *hsmfs1* ファイルシステムのレポートを要求します。

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

5. 特定のグループの割り当て統計を表示するには、コマンド *samquota -G groupID [directory-or-file]* を使用して、*groupID* でグループのグループ名または整

数の識別子を指定し、オプションの *directory-or-file* パラメータで、指定されたディレクトリにマウントされたファイルシステム、指定されたディレクトリ自体、または指定されたファイルにレポートの範囲を限定します。

この例では、*/hsm/hsmfs1* にマウントされた *hsmfs1* ファイルシステム内にある、*dev* グループの割り当てのレポートを要求します。

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

6. 特定の管理セットの割り当て統計を表示するには、コマンド *samquota -A adminsetID [directory-or-file]* を使用して、*adminsetID* で管理セットの整数の識別子を指定し、オプションの *directory-or-file* パラメータで、指定されたディレクトリにマウントされたファイルシステム、指定されたディレクトリ自体、または指定されたファイルにレポートの範囲を限定します。

この例では、*/hsm/hsmfs1* にマウントされた *hsmfs1* ファイルシステム内にある、管理セット *1* の割り当てのレポートを要求します。

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

7. 特定のユーザーの割り当て統計を表示するには、コマンド *samquota -U userID [directory-or-file]* を使用して、*userID* でユーザーのユーザー名または整数の識別子を指定し、オプションの *directory-or-file* パラメータで、指定されたディレクトリにマウントされたファイルシステム、指定されたディレクトリ自体、または指定されたファイルにレポートの範囲を限定します。

この例では、*/hsm/hsmfs1* にマウントされた *hsmfs1* ファイルシステム内にある、ユーザー *jr23547* の割り当てのレポートを要求します。

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

8. ここで停止します。

ユーザー独自の割り当てのモニタリング

1. 自分のユーザー ID を使用して、ファイルシステムホストにログインします。

この例では、ホスト *server1* にユーザー *od447* としてログインします。

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

- すべてのグループの割り当て統計を表示するには、コマンド `squota` `[directory-or-file]` を使用し、オプションの `directory-or-file` パラメータで、指定されたディレクトリにマウントされたファイルシステム、指定されたディレクトリ自体、または指定されたファイルにレポートの範囲を限定します。

この例では、すべてのファイルシステムに関するレポートを要求します。

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

```

                                Limits
      Type   ID   In Use   Soft   Hard
/hsm/hsmfs1
Files group 101     1    1000   1200
Blocks group 101     8   20000  30000
Grace period                25920
No user quota entry.
```

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

- ここで停止します。

猶予期間の一時的な延長または取り消し

猶予期間を一時的に延長する必要がある場合や、猶予期間を短縮する必要がある場合は、次の手順を実行します。

- [指定した分の猶予期間の延長](#)
- [猶予期間の再開](#)
- [猶予期間の早期終了](#)。

指定した分の猶予期間の延長

グループ、ユーザー、または管理セットがその割り当てに指定された弱い制限値を超過していて、弱い制限値を上回る状態を一時的に保持する必要がある場合は、現在の猶予期間で許可される期間よりも長くならなければ、次のように延長を付与できます。

1. ファイルシステムサーバーに `root` としてログインします。

この例では、ホスト `server1` にログインします。

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

2. 延長が必要な割り当てを確認します。コマンド `samquota -quota-type ID [directory-or-file]` を使用します。ここでは:

- `quota-type ID` は、`G` とグループ名または ID 番号、`A` と管理セット ID 番号、または `U` とユーザー名または ID 番号です。
- `directory-or-file` (オプション) は、猶予期間を延長する必要がある特定のファイルシステム用または特定のディレクトリやファイル用のマウントポイントディレクトリです。

この例では、`dev` グループは弱い制限値を大幅に上回り、猶予期間内に数時間しか残っていません。

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

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

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

3. 保証されていれば、猶予期間を延長します。コマンド `samquota -quota-type ID -x number-seconds [directory-or-file]` を使用します。ここでは:
 - `quota-type ID` は、`G` とグループ名または ID 番号、`A` と管理セット ID 番号、または `U` とユーザー名または ID 番号です。
 - `directory-or-file` (オプション) は、猶予期間を延長する必要がある特定のファイルシステム用または特定のディレクトリやファイル用のマウントポイントディレクトリです。
 - `number-seconds` は延長する秒数を表す整数です (時間を指定する代替の方法については `samquota` のマニュアルページを参照)。

プロンプトが表示されたら「y」（はい）を入力して、続行します。

この例では、*hsmfs1* ファイルシステム内のファイルについて、*dev* グループの猶予期間を 2678400 秒 (31 日) に延長します。

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

dev グループの割り当てを再確認すると、猶予期間が延長されています。

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

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

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

4. グループ、管理セット、またはユーザーが定期的に延長する必要がある場合は、ストレージ要件を再評価したり、猶予期間を永続的に延長することを検討したりしてください。「[グループ、プロジェクト、ディレクトリ、およびユーザー用の割り当ての設定](#)」の手順を使用します。
5. ここで停止します。

猶予期間の再開

グループ、ユーザー、または管理セットがその割り当てに指定された弱い制限値を超過していて、現在の猶予期間が切れる前に、弱い制限値を下回るのに十分な速さで容量を解放できない場合は、猶予期間を再開できます。次のように進めます。

1. ファイルシステムサーバーに *root* としてログインします。

この例では、ホスト *server1* にログインします。

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

2. 延長が必要な割り当てを確認します。コマンド `samquota -quota-type ID [directory-or-file]` を使用します。ここでは:
- `quota-type ID` は、`G` とグループ名または ID 番号、`A` と管理セット ID 番号、または `U` とユーザー名または ID 番号です。
 - `directory-or-file` (オプション) は、猶予期間を延長する必要がある特定のファイルシステム用または特定のディレクトリやファイル用のマウントポイントディレクトリです。

この例では、`cit` グループは `hsmfs1` ファイルシステムの弱い制限値を上回り、猶予期間が 1 時間強残っています。

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

			Online Limits			Total Limits		
Type	ID	In Use	Soft	Hard	In Use	Soft	Hard	
/hsm/hsmfs1								
Files	group	119	762	750	1500	762	750	1500
Blocks	group	119	3109330961	2013265920	3019898880	120096782	157286400	235929600
Grace period			4320			4320		

```
---> Warning: soft limits to be enforced in 1h11m23s
[server1]root@solaris:~#
```

3. 次回、ファイルまたはブロックが割り当てられるときに、猶予期間を開始時のフルサイズにリセットするには、猶予期間タイマーをクリアします。コマンド `samquota -quota-type ID -x clear [directory-or-file]` を使用します。ここでは:
- `quota-type ID` は、`G` とグループ名または ID 番号、`A` と管理セット ID 番号、または `U` とユーザー名または ID 番号です。
 - `directory-or-file` (オプション) は、猶予期間を延長する必要がある特定のファイルシステム用または特定のディレクトリやファイル用のマウントポイントディレクトリです。

プロンプトが表示されたら「y」(はい)を入力して、続行します。

この例では、*hsmfs1* ファイルシステム上の *cit* グループの割り当て用の猶予期間タイマーをクリアします。

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

cit グループの割り当てを再確認すると、ファイルが割り当てられ、猶予期間が *12h* (12 時間、4320 秒) にリセットされています。

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

			Online Limits			Total Limits		
Type	ID	In Use	Soft	Hard	In Use	Soft	Hard	
/hsm/hsmfs1								
Files	group	119	763	750	1500	763	750	1500
Blocks	group	119	3109330961	2013265920	3019898880	120096782	157286400	235929600
Grace period			4320			4320		
---> Warning: soft limits to be enforced in 12h								

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

- また、すぐに猶予期間を開始時のフルサイズにリセットするには、猶予期間タイマーをリセットします。コマンド `samquota -quota-type ID -x reset [directory-or-file]` を使用します。
 - `quota-type ID` は、*G* とグループ名または ID 番号、*A* と管理セット ID 番号、または *U* とユーザー名または ID 番号です。
 - `directory-or-file` (オプション) は、猶予期間を延長する必要がある特定のファイルシステム用または特定のディレクトリやファイル用のマウントポイントディレクトリです。

プロンプトが表示されたら「y」(はい)を入力して、続行します。

この例では、*hsmfs1* ファイルシステム上の *cit* グループの割り当て用の猶予期間タイマーをクリアします。

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

```
Setting Grace Timer: continue? y
```

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

cit グループの割り当てを再確認すると、ファイルが割り当てられ、猶予期間が *12h* (12 時間、4320 秒) にリセットされています。

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

			Online Limits			Total Limits		
Type	ID	In Use	Soft	Hard	In Use	Soft	Hard	
/hsm/hsmfs1								
Files	group 119	762	750	1500	762	750	1500	
Blocks	group 119 3109330961	2013265920	3019898880	120096782	157286400	235929600		
Grace period		4320			4320			

```
---> Warning: soft limits to be enforced in 12h
[server1]root@solaris:~#
```

5. ここで停止します。

猶予期間の早期終了

1. ファイルシステムサーバーに *root* としてログインします。

この例では、ホスト *server1* にログインします。

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

2. 短縮する必要がある猶予期間を確認します。コマンド *samquota -quota-type ID [directory-or-file]* を使用します。ここでは:

- *quota-type ID* は、*G* とグループ名または ID 番号、*A* と管理セット ID 番号、または *U* とユーザー名または ID 番号です。
- *directory-or-file* (オプション) は、猶予期間を延長する必要がある特定のファイルシステム用または特定のディレクトリやファイル用のマウントポイントディレクトリです。

この例では、*cit* グループは弱い制限値を上回り、猶予期間内に 11 時間残っていますが、猶予期間を早期に終了する必要があります。

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

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

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

3. 猶予期間を期限切れにします。コマンド `samquota -quota-type ID -x expire [directory-or-file]` を使用します。ここでは:
 - *quota-type ID* は、*G* とグループ名または ID 番号、*A* と管理セット ID 番号、または *U* とユーザー名または ID 番号です。
 - *directory-or-file* (オプション) は、猶予期間を延長する必要のある特定のファイルシステム用または特定のディレクトリやファイル用のマウントポイントディレクトリです。

この例では、*cit* グループの猶予期間を期限切れにします。

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

割り当てを再確認すると、*cit* グループの弱い制限値が強制的に強い制限値として適用されています。

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

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

```
/hsm/hsmfs1
Files group 119      762      750      1500      762      750  1500
Blocks group 119 3109330961 2013265920 3019898880 120096782 157286400
      235929600
Grace period          4320          4320
---> Online soft limits under enforcement (since 6s ago)
[server1]root@solaris:~#
```

4. ここで停止します。

新しいリソース割り当ての停止

矛盾する割り当て値を作成して、ファイルシステムリソースの割り当てを禁止できます。ファイルシステムは、ユーザー、グループ、または管理セットに対して割り当て値が矛盾していることを検出した場合に、そのユーザー、グループ、または管理セットに追加のシステムリソースの使用を禁止します。したがって、対応する弱い制限値を下回る強い制限値を割り当てに設定すると、それ以降の割り当てが停止します。このテクニックを使用するには、次のように進めます。

1. ファイルシステムサーバーに *root* としてログインします。

この例では、ホスト *server1* にログインします。

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

2. あとで復元できるように、割り当てをバックアップします。現在の構成をエクスポートし、情報をファイルにリダイレクトします。コマンド `samquota -quota-type ID [directory-or-file] > file` を使用します。ここでは:
 - *quota-type ID* は、*G* とグループ名または ID 番号、*A* と管理セット ID 番号、または *U* とユーザー名または ID 番号です。
 - *directory-or-file* (オプション) は、猶予期間を延長する必要がある特定のファイルシステム用または特定のディレクトリやファイル用のマウントポイントディレクトリです。
 - *file* は、出力ファイルの名前です。

この例では、*root* ユーザーのホームディレクトリ内のファイル *restore* *.hsmfs1.quota_g.cit* に *cit* グループの割り当てをエクスポートします (次のコマンドは 1 行で入力し、改行はバックスラッシュ文字でエスケープします)。

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

3. 出力を確認します。Solaris コマンド *more < file* を使用します。ここで *file* は出力ファイルの名前です。

```
[server1]root@solaris:~# more < /root/restore.hsmfs1.quota_g.cit
# Type ID
#
#           Online Limits
#           soft          hard
# Files
# Blocks
# Grace Periods
samquota -G 119 /
-f          750:s:o -f          1500:h:o      -f          750:s:t -f          1500:h:t /
-b 157286400:s:o -b 235929600:h:o      -b 157286400:s:t -b 235929600:h:t /
-t 4320:o      -t 4320:t
```

4. 割り当ての強い制限値を *0* (ゼロ)、弱い制限値を *1* (またはゼロ以外の値) に設定します。コマンド *samquota -quota-type ID -f 1:s -f 0:h -b 1:s -b 0:h [directory-or-file]* を使用します。

- *quota-type ID* は、*G* とグループ名または ID 番号、*A* と管理セット ID 番号、または *U* とユーザー名または ID 番号です。
- *directory-or-file* (オプション) は、猶予期間を延長する必要がある特定のファイルシステム用または特定のディレクトリやファイル用のマウントポイントディレクトリです。

この例では、*/hsm/hsmfs1* ファイルシステム内の *cit* グループ用の割り当て設定を不整合にし、それによって新しいリソース割り当てが停止します。

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

cit グループ用の割り当てを確認すると、ゼロの割り当てが有効になっています。感嘆符 (!) は、現在の使用量がすべて割り当てを上回っているため、それ以上の割り当てが行われないことを示します。

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

- 変更された割り当てを元の状態に復元して、通常の割り当てを再開できる状態になっている場合は、シェルスクリプトとして作成したバックアップファイルを実行します。Solaris コマンド *sh file* を使用します。ここで *file* はバックアップファイルの名前です。

この例では、ファイル */root/restore.hsmfs1.quota_g.cit* を実行して、*cit* グループ用の割り当てを復元します。

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

割り当てを確認すると、通常の制限値が復元され、割り当てがブロックされていません。

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



```

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

```

6. ここで停止します。

ファイルシステムの割り当ての削除

ファイルシステム割り当てを削除または無効にするには、マウントプロセスで割り当てを無効にします。

1. ファイルシステムサーバーに *root* としてログインします。

この例では、ホスト *server1* にログインします。

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

2. テキストエディタで */etc/vfstab* ファイルを開き、ファイルシステム行のマウントポイント列に *noquota* マウントオプションを追加して、ファイルを保存します。

この例では、*vi* テキストエディタでファイルを開き、*hsmfs1* ファイルシステムに *noquota* マウントオプションを設定します。

```

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

```

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

3. ファイルシステムがマウントされている場合は、アンマウントします。

オペレーティングシステムが `/etc/vfstab` ファイルをリロードし、指定された変更を行うことができるように、ファイルシステムをアンマウントしてから再マウントする必要があります。この例では、`hsmfs1` ファイルシステムをアンマウントします。

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

4. ファイルシステムをマウントします。

この例では、`hsmfs1` ファイルシステムをマウントします。

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

5. あとで割り当てを元に戻す予定がある場合は、割り当てファイルを残しておきます。

割り当てを元に戻すことができる状態になっている場合は、ファイルシステムをアンマウントし、ファイルシステム上でコマンド `samfsck -F` を実行し、`noquota` マウントオプションを削除してから、ファイルシステムを再マウントするだけです。

6. 割り当てを元に戻す予定がない場合や、割り当てファイルで消費されている容量を再利用する必要がある場合は、Solaris コマンド `rm` を使用して、ファイルシステムのルートディレクトリから `.quota_g`、`.quota_a`、または `.quota_u` ファイル (あるいはすべて) を削除します。

この例では、`/hsm/hsmfs1` ファイルシステムのルートディレクトリから割り当てファイルをすべて削除します。

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

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

7. ここで停止します。

アーカイブおよびステージング操作の制御

一般に、アーカイブファイルシステムは、非アーカイブファイルシステムとほぼ同じ方法で管理します。ただし、ほとんどのファイルシステム管理タスクを実行する前には、アーカイブプロセスを停止する必要があります。アクティブになっていると、アーカイブ処理によってファイルシステムのプライマリディスクキャッシュが変更されます。したがって、ディスクキャッシュ上で保守作業を行う前に、これらのプロセスを休止する必要があります。このセクションでは、次のタスクを扱います。

- [アーカイブおよびステージングプロセスのアイドル状態](#)
- [アーカイブおよびステージングプロセスの停止](#)
- [アーカイブおよびステージングプロセスの再起動](#)。

アーカイブおよびステージングプロセスのアイドル状態

1. ファイルシステムホストに *root* としてログインします。

この例では、ホスト *server1* にログインします。

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

2. すべてのアーカイブプロセスをアイドル状態にします。コマンド *samcmd aridle* を使用します。

このコマンドは現在のアーカイブおよびステージングを完了できますが、新しいジョブは開始されません。

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

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

3. すべてのステージングプロセスをアイドル状態にします。コマンド *samcmd stidle* を使用します。

このコマンドは現在のアーカイブおよびステージングを完了できますが、新しいジョブは開始されません。

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

4. アクティブなアーカイブジョブが完了するまで待機します。コマンド `samcmd a` を使用して、アーカイブ処理のステータスを確認します。

アーカイブ処理が *Waiting for :arrun* の場合、アーカイブ処理はアイドル状態になっています。

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

5. アクティブなステージングジョブが完了するまで待機します。コマンド `samcmd u` を使用して、ステージングプロセスのステータスを確認します。

ステージングプロセスが *Waiting for :strun* の場合、ステージングプロセスはアイドル状態になっています。

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

6. システムを完全に休止するには、アーカイブおよびステージングプロセスの停止も行います。

アーカイブおよびステージングプロセスの停止

1. まだ実行していない場合には、アーカイブおよびステージングプロセスをアイドル状態にします。
2. まだ実行していない場合には、ファイルシステムホストに *root* としてログインします。

この例では、ホスト *server1* にログインします。

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

3. すべてのリムーバブルメディアドライブをアイドル状態にしてから、続行します。ドライブごとに、コマンド *samcmd equipment-number idle* を使用します。ここで *equipment-number* は、*/etc/opt/SUNWsamfs/mcf* ファイル内のドライブに割り当てられている装置の順序番号です。

このコマンドはドライブを「*off*」にする前に、現在のアーカイブジョブおよびステージングジョブを完了できますが、新しいジョブは開始されません。この例では、4つのドライブ(順序番号 *801*、*802*、*803*、*804*)をアイドル状態にします。

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

4. 実行中のジョブが完了するまで待機します。

コマンド *samcmd r* を使用すると、ドライブのステータスを確認できます。すべてのドライブが「*notrdy*」または「*empty*」の場合は、続行できる状態になっています。

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

```
empty
li 802 -----p 0 0% notrdy
empty
li 803 -----p 0 0% notrdy
empty
li 804 -----p 0 0% notrdy
empty
[server1]root@solaris:~#
```

5. アーカイバおよびステージャープロセスがアイドル状態で、テープドライブがすべて「*notrdy*」になっている場合は、ライブラリ制御デーモンを停止します。コマンド *samd stop* を使用します。

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

6. ファイルシステムの保守を続行します。
7. 保守が完了したら、アーカイブおよびステーキングプロセスを再起動します。

操作を再開すると、保留中のステーキングリクエストが再発行され、アーカイブが再開されます。

8. ここで停止します。

アーカイブおよびステーキングプロセスの再起動

準備ができれば、通常の自動操作を再開し、次の手順に従います。

1. ファイルシステムホストに *root* としてログインします。

この例では、ホスト *server1* にログインします。

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

2. Oracle HSM ライブラリ制御デーモンを再起動します。コマンド *samd start* を使用します。

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

3. ここで停止します。

ファイルシステム名の変更

ファイルシステムの名前は、2段階のプロセスで変更します。まず、`/etc/opt/SUNWsamfs/mcf` ファイルを編集して、ファイルシステムのファミリーセット名を変更します。次に、`samfsck -R -F` コマンドで新しい名前を読み込み、対応するディスクデバイス上のスーパーブロックを更新します。ファイルシステムの名前を変更するには、次の手順を使用します。

ファイルシステム名の変更

1. ファイルシステムサーバーに `root` としてログインします。

この例では、ホスト `server1` にログインします。

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

2. アーカイブファイルシステムを修復する場合は、手順「[アーカイブおよびステージングプロセスのアイドル状態](#)」を実行してから続行します。
3. 名前を変更する必要があるファイルシステムをアンマウントします。

この例では、ファイルシステム `hsmfs1` をアンマウントします。

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

4. テキストエディタで `/etc/opt/SUNWsamfs/mcf` ファイルを開き、名前を変更する必要があるファイルシステムを検索します。

次の例では、`vi` エディタを使用します。`hsmfs1` ファイルシステムの名前を変更する必要があります。

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

# Equipment	Equipment	Equipment	Family	Device	Additional
# Identifier	Ordinal	Type	Set	State	Parameters
#-----	-----	-----	-----	-----	-----
hsmfs1	100	ms	hsmfs1	on	
/dev/dsk/c1t3d0s3	101	md	hsmfs1	on	

```
/dev/dsk/c1t4d0s5 102 md hsmfs1 on
```

5. ファイルの 4 列目で、ファイルシステムのファミリーセット名を新しい値に変更します。1 列目でファイルシステム装置の識別子も変更できますが、それ以外は変更しないでください。ファイルを保存して、エディタを閉じます。

この例では、装置識別子とファイルシステムのファミリーセット名の両方を *hsmfs1* から *samqfs-hpcc* に変更します。

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

:wq
root@solaris:~#
```

6. ファイルシステムのスーパーブロックを再度書き込んで、新しいファミリーセット名を反映させます。コマンド *samfsck -R -F family-set-name* を使用します。ここで *family-set-name* は、*/etc/opt/SUNwsamfs/mcf* ファイル内で指定したファミリーセット名です。

-R および *-F* オプションを付けて *samfsck* コマンドを発行すると、*/etc/opt/SUNwsamfs/mcf* ファイルから新しいファミリーセット名および対応するディスクストレージ装置の識別子が読み込まれます。その後、新しいファミリーセット名で、指定したディスクデバイス上のスーパーブロックが再度書き込まれます。この例では、新しいファミリーセット名 *samqfs-hpcc* でコマンドを実行します。

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

7. テキストエディタで */etc/vfstab* ファイルを開き、名前を変更するファイルシステムのエントリを検索します。

この例では、*vi* テキストエディタでファイルを開きます。新しい名前を使用するには、*hsmfs1* ファイルシステムのエントリを変更する必要があります。


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

8. 名前を変更したファイルシステムの `/etc/vfstab` エントリで、1 列目のファイルシステム名、および (必要に応じて) 3 列目のマウントポイントディレクトリ名を変更して、ファイルを保存します。

この例では、`hsmfs1` ファイルシステムの名前を `samqfs-hpcc` に変更し、マウントポイントも変更して一致させます。

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

9. 必要に応じて新しいファイルシステム用の新しいマウントポイントディレクトリを作成し、マウントポイントに対するアクセス権を設定します。

ユーザーはマウントポイントディレクトリに移動し、マウントしたファイルシステム内のファイルにアクセスするための実行権 (x) を持っている必要があります。この例では、`/samqfs-hpcc` マウントポイントディレクトリを作成し、アクセス権を `755 (-rwxr-xr-x)` に設定します。

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

10. *sam-fsd* コマンドを実行して、*mcf* ファイルにエラーがないかどうかを確認し、検出された場合は修正します。

sam-fsd は、Oracle HSM 構成ファイルを読み取る初期化コマンドです。エラーを検出すると停止します。

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

11. *mcf* ファイルを再度読み取り、ソフトウェア自体を適宜再構成するように Oracle HSM ソフトウェアに指示します。コマンド *samd config* を使用します。

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

12. *samd config* によりエラーが報告された場合、修正し、エラーがなくなるまでコマンドを再発行します。
13. ファイルシステムをマウントします。

この例では、新しいマウントポイントディレクトリを使用します。

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

14. ここで停止します。

ファイルシステムの修復

ファイルシステムが *samu*、Oracle HSM Manager、または */var/adm/sam-log* ファイル経由でエラーをレポートした場合は、次の手順に従います。

ファイルシステムの修復

1. ファイルシステムサーバーに *root* としてログインします。

この例では、ホスト *server1* にログインします。

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

2. アーカイブファイルシステムを修復する場合は、手順「[アーカイブおよびステージングプロセスのアイドル状態](#)」を実行してから続行します。
3. 影響を受けるファイルシステムをアンマウントします。

アーカイブが停止するまで待機している場合は、複数回試す必要がある場合もあります。この例では、ファイルシステム *hsmfs1* をアンマウントします。

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

4. ファイルシステムを修復します。コマンド `samfsck -F -V family-set-name` を使用します。ここで *family-set-name* は、`/etc/opt/SUNWsamfs/mcf` ファイルでファイルシステムに対して指定されたファミリセット名です。

必要に応じて、今後の参照や診断の目的のために、修復結果を日付の付いたファイルに保存しておくことを推奨します。そのため、この例では、`samfsck` の出力をコマンド `tee /var/tmp/samfsck-FV.family-set-name.`date +%Y%m%d.%H%M%S`` にパイプして結果を保存します (次のコマンドは 1 行で入力し、改行はバックスラッシュ文字でエスケープします)。

```
[server1]root@solaris:~# samfsck -F -V hsmfs1 | tee /
/var/tmp/samfsck-FV.hsmfs1.`date +%Y%m%d.%H%M%S`
name:      /hsm/hsmfs1      version:      2A
First pass
Second pass
Third pass
NOTICE: ino 2.2,  Repaired link count from 8 to 14
Inodes processed: 123392
total data kilobytes      = 1965952
total data kilobytes free = 1047680
total meta kilobytes      = 131040
total meta kilobytes free = 65568
INFO:  FS samma1 repaired:
      start:  May 19, 2014 10:57:13 AM MDT
```

```
finish: May 19, 2014 10:57:37 AM MDT
NOTICE: Reclaimed 70057984 bytes
NOTICE: Reclaimed 9519104 meta bytes
[server1]root@solaris:~#
```

5. ファイルシステムを再マウントします。

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

6. ここで停止します。

ファイルシステムへのデバイスの追加

既存のファイルシステムにデバイスを追加する前に、要件や代替方法を検討することをお勧めします。増大する容量の要件を満たすための最適な方法は、既存のファイルシステムを拡張することです。新しいプロジェクトやユーザーコミュニティに対応するために、より大きな物理ストレージ領域が必要である場合は、1つ以上の新しい Oracle HSM ファイルシステムを作成することが適切な選択肢となる場合があります。一般に、小さいファイルシステムが複数ある方が大きいファイルシステムが1つあるよりもパフォーマンスが高く、小さいファイルシステムの方が簡単に作成および保守できます。

ファイルシステムを拡張する必要があると判断したら、次の方法のいずれかを行います。

- [マウントしたファイルシステムへのデバイスの追加 \(推奨\)](#)
- [アンマウントしたファイルシステムへのデバイスの追加](#)

マウントしたファイルシステムへのデバイスの追加

次のように進めます。

1. ファイルシステムサーバーに *root* としてログインします。

この例では、ホスト *server1* にログインします。

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

2. テキストエディタで `/etc/opt/SUNWsamfs/mcf` ファイルを開き、拡張する必要のあるファイルシステムを検索します。

この例では、`vi` エディタを使用します。汎用の `samqfsms` ファイルシステムと高性能 `samqfs2ma` ファイルシステムの2つのファイルシステムを拡張する必要があります。

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

3. 汎用の `ms` ファイルシステムにデバイスを追加する場合は、追加のデータまたはメタデータデバイスを `mcf` ファイル内のファイルシステム定義の末尾に追加します。次に、ファイルを保存し、エディタを閉じます。

最大 252 個の論理デバイスを追加できます。この例では、2つのデバイス (103 と 104) を `samqfsms` ファイルシステムに追加します。

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

- 高パフォーマンスの *ma* ファイルシステムにデバイスを追加する場合は、データデバイスおよび 1 つ以上の *mm* ディスクデバイスを *mcf* ファイル内のファイルシステム定義の末尾に追加します。次に、ファイルを保存し、エディタを閉じます。

必ず、新しいデバイスは既存のデバイスリストの末尾に追加します。データデバイスの追加に比例して、最大 252 個のメタデータデバイスを追加できます。この例では、1 つの *mm* メタデータデバイス 204、および 2 つの *md* データデバイス (205 と 206) を *samqfs2ma* ファイルシステムに追加します。

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

- sam-fsd* コマンドを実行して、*mcf* ファイルにエラーがないかどうかを確認し、検出された場合は修正します。

sam-fsd は、Oracle HSM 構成ファイルを読み取る初期化コマンドです。エラーを検出すると停止します。

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

- sam-fsd* コマンドによって *mcf* ファイルでエラーが見つかった場合は、ファイルを編集してエラーを修正し、前の手順の説明に従って再確認します。

次の例では、*sam-fsd* によって、デバイスに関して何らかの問題があることが指摘されています。

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

通常、このようなエラーの原因は不注意なタイプミスです。ここで、エディタで *mcf* ファイルを開くと、デバイス *104* (2 番目に新しい *md* デバイス) の装置名に、*0* の代わりに文字 *o* が入力されていることがわかります。

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

7. *sam-fsd* コマンドがエラーなしで実行された場合、*mcf* ファイルは正確です。次の手順に進みます。

この例は、エラーのない出力の一部です。

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

8. *mcf* ファイルを再度読み取り、ソフトウェア自体を適宜再構成するように Oracle HSM ソフトウェアに指示します。コマンド *samd config* を使用します。

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

9. `samd config` によって、新しいデバイスが含まれるように Oracle HSM ファイルシステムの構成が更新されたことを確認します。コマンド `samcmd f` を使用します。

デバイスの状態は「*off*」になっているはずですが、この例では、`samcmd f` は新しいデバイス `103` と `104`、および両方とも「*off*」であることが表示されます。

```
[server1]root@solaris:~# samcmd f
File systems samcmd      5.4 16:57:35 Feb 27 2014
samcmd on server1

```

ty	eq	state	device_name	status	high	low	mountpoint	server
ms	100	on	samqfsms	m----2-----	80%	70%	/samqfsms	
md	101	on	/dev/dsk/c1t3d0s3					
md	102	on	/dev/dsk/c1t4d0s5					
md	103	off	/dev/dsk/c1t3d0s7					
md	104	off	/dev/dsk/c1t4d0s7					

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

10. 新たに追加されたデバイスを有効にします。デバイスごとに、コマンド `samcmd add equipment-number` を使用します。ここで `equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイル内のデバイスに割り当てられている装置の順序番号です。

この例では、新しいデバイス (`103` と `104`) を有効にします。

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

11. 共有ファイルシステムにデバイスを追加する場合は、「共有ファイルシステムに追加された新しいデバイスの構成の完了」に進みます。

12. 非共有のスタンドアロンファイルシステムにデバイスを追加する場合は、デバイスが追加され、ファイルシステムで使用できる状態になっていることを確認します。コマンド `samcmd m` を使用して、結果を確認します。

デバイスの状態が「*on*」である場合は、デバイスが正常に追加され、使用できる状態になっています。この例では、デバイス *103* と *104* が正常に追加されています。

```
[server1]root@solaris:~# samcmd f
Mass storage status samcmd      5.4 17:17:08 Feb 27 2014
samcmd on server1

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

13. ここで停止します。

共有ファイルシステムに追加された新しいデバイスの構成の完了

共有ファイルシステムにデバイスを追加する場合は、すべてのファイルシステムホスト上でデバイスが構成される前に、追加の手順をいくつか実行する必要があります。次のように進めます。

1. ファイルシステムのメタデータサーバーホストに `root` としてログインします。
この例では、メタデータサーバーホストは `metadata-server` という名前です。

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

2. メタデータサーバーに新しいデバイスが追加されたことを確認します。コマンド `samcmd m` を使用します。

デバイスの状態が「*unavail*」である場合は、デバイスが正常に追加されましたが、使用できる状態にはなっていません。この例では、デバイス *103* と *104* が正常に追加されています。

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

3. 各ファイルシステムのクライアントホストに *root* としてログインします。

潜在的なメタデータサーバーもクライアントであるため、忘れずに含めてください。この例では、*potential-metadata-server* という名前の潜在的なメタデータサーバー、*client1* および *client2Linux* の2つのクライアントにログインする必要があります。そのため、3つの端末ウィンドウを開き、セキュアシェル (*ssh*) を使用します。

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

4. クライアントが Linux クライアントである場合は、共有ファイルシステムをアンマウントします。

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

- クライアントごとに、テキストエディタで `/etc/opt/SUNWsamfs/mcf` ファイルを開き、サーバー上で行なったときと同様に、新しいデバイスをファイルシステム定義の末尾に追加します。

この例では、デバイス `103` と `104` を `client1` 上の `mcf` ファイルに追加します。

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

- クライアントごとに、`sam-fsd` コマンドを実行して、`mcf` ファイルにエラーがないかどうかを確認し、検出された場合は修正します。

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

- 各クライアント上で、`mcf` ファイルを再度読み取り、ソフトウェア自体を適宜再構成するように Oracle HSM ソフトウェアに指示します。

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

- クライアントが Linux クライアントである場合は、共有ファイルシステムをマウントします。

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

- すべてのクライアントが構成されたら、メタデータサーバーに戻って、新しいデバイス上でストレージの割り当てを有効にします。デバイスごとに、コマンド `samcmd alloc equipment-number` を使用します。ここで `equipment-number`

は、`/etc/opt/SUNWsamfs/mcf` ファイル内のデバイスに割り当てられている装置の順序番号です。

この例では、デバイス `103` と `104` 上でストレージの割り当てを有効にします。

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

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

10. 最後に、デバイスがファイルシステムで使用できる状態になっていることを確認します。コマンド `samcmd m` を使用して、結果を確認します。

デバイスの状態が「`on`」である場合は、デバイスが正常に追加され、使用できる状態になっています。この例では、デバイス `103` と `104` が正常に追加されています。

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

```
Mass storage status samcmd      5.4 17:17:08 Feb 27 2014
```

```
samcmd on metadata-server
```

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

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

11. ここで停止します。

アンマウントしたファイルシステムへのデバイスの追加

次のように進めます。

1. ファイルシステムのサーバーホストに `root` としてログインします。

この例では、メタデータサーバーホストは `server1` という名前です。

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

2. アーカイブファイルシステムをアンマウントする前に、手順「[アーカイブおよびステージングプロセスのアイドル状態](#)」を実行する必要があります。
3. ファイルシステムをマウント解除します。

ファイルシステムをアンマウントするまでは次に進まないでください。この例では、ファイルシステム *hsmfs1* をアンマウントします。

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

4. テキストエディタで */etc/opt/SUNWsamfs/mcf* ファイルを開き、拡張する必要のあるファイルシステムを検索します。

次の例では、*vi* エディタを使用します。*hsmfs1* ファイルシステムを拡張する必要があります。

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

# Equipment	Equipment	Equipment	Family	Device	Additional
# Identifier	Ordinal	Type	Set	State	Parameters
#-----	-----	-----	-----	-----	-----
hsmfs1	100	ms	hsmfs1	on	
/dev/dsk/c1t3d0s3	101	md	hsmfs1	on	
/dev/dsk/c1t4d0s5	102	md	hsmfs1	on	

5. 高パフォーマンスの *ma* ファイルシステムにデバイスを追加する場合は、データストレージとともにメタデータストレージを追加する必要があります。追加するデータデバイスのメタデータを格納するのに十分な *mm* ディスクデバイスを追加します。次に、ファイルを保存し、エディタを閉じます。

最大 252 個の論理デバイスを追加できます。この例では、1 つの *mm* メタデータデバイスを *samqfs2ma* ファイルシステムに、2 つのデータデバイスを *samqfs2ma* ファイルシステムに追加します。

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

# Equipment	Equipment	Equipment	Family	Device	Additional
# Identifier	Ordinal	Type	Set	State	Parameters
#-----	-----	-----	-----	-----	-----
samqfs2ma	200	ma	samqfs2ma	on	

```

/dev/dsk/c1t3d0s3    201      mm      samqfs2ma  on
/dev/dsk/c1t5d0s6    204      mm      samqfs2ma  on
/dev/dsk/c1t3d0s5    202      md      samqfs2ma  on
/dev/dsk/c1t4d0s5    203      md      samqfs2ma  on
/dev/dsk/c1t3d0s7    205      md      samqfs2ma  on
/dev/dsk/c1t4dos7    206      md      samqfs2ma  on
:wq
[server1]root@solaris:~#

```

6. 汎用の *ms* ファイルシステムにデバイスを追加する場合は、追加のデータデバイスまたはメタデータデバイスを *mcf* ファイル内のファイルシステム定義に追加します。次に、ファイルを保存し、エディタを閉じます。

最大 252 個の論理デバイスを追加できます。この例では、2 つのデバイスを *hsmfs1* ファイルシステムに追加します。

```

[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment      Equipment  Equipment  Family    Device    Additional
# Identifier      Ordinal    Type       Set       State     Parameters
#-----
hsmfs1           100       ms        hsmfs1    on
/dev/dsk/c1t3d0s3  101       md        hsmfs1    on
/dev/dsk/c1t4d0s5  102       md        hsmfs1    on
/dev/dsk/c1t3d0s7  103       md        hsmfs1    on
/dev/dsk/c1t4dos7  104       md        hsmfs1    on
:wq
[server1]root@solaris:~#

```

7. *sam-fsd* コマンドを実行して、*mcf* ファイルにエラーがないかどうかを確認し、検出された場合は修正します。

sam-fsd は、Oracle HSM 構成ファイルを読み取る初期化コマンドです。エラーを検出すると停止します。

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

8. *mcf* ファイルを再度読み取り、ソフトウェア自体を適宜再構成するように Oracle HSM software に指示します。

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

9. 新しいデバイスをファイルシステムに組み込みます。コマンド *samgrowfs family-set-name* を使用します。ここで *family-set-name* は、*/etc/opt/SUNWsamfs/mcf* ファイル内でファイルシステムに指定されたファミリーセット名です。

この例では、*hsmfs1* ファイルシステムを拡張します。

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

10. ファイルシステムを再マウントします。

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

11. アーカイブファイルシステムにデバイスを追加した場合は、Oracle HSM ライブラリ管理デーモンを再起動します。コマンド *samd start* を使用します。

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

12. 変更を行う前にファイルシステムをアンマウントし忘れたために、ファイルシステムがマウントされない場合は、追加したデバイスへの参照を削除して、元の *mcf* ファイルを復元します。次に、*samd config* を実行して構成を復元し、ファイルシステムをアンマウントしてから、もう一度やり直します。

13. ここで停止します。

ファイルシステムからのデータデバイスの削除

必要に応じて、マウントした Oracle HSM ファイルシステムからデータデバイスを削除できます。一般に、障害の発生したユニットを交換する必要がある場合、または十分に利用されていないデバイスをほかのユーザー用に解放する必要がある場合に、この操作が必要になります。ただし、いくつかの制限があります。

データデバイスのみを削除できます。メタデータにはファイルシステム自体の構成が定義されているため、メタデータの保持に使用されているデバイスは削除できま

せん。つまり、高パフォーマンスの *ma* ファイルシステムから削除できるデバイスは、*md*、*mr*、およびストライプグループデバイスのみです。*ma* ファイルシステムから *mm* メタデータデバイスを削除することはできません。また、*md* デバイスには、データとメタデータの両方が格納されているため、これらのデバイスを汎用の *ms* ファイルシステムから削除することもできません。

デバイスを削除するには、ターゲットデバイス上に存在する有効なデータファイルを移動するための場所も必要です。つまり、すべてのデバイスを削除することはできません。必ず、1つのデバイスがファイルシステムに残っている必要があり、削除するデバイスに存在するすべてのファイルを保持するための十分な空き容量が必要です。したがって、ストライプグループを削除する必要がある場合は、同数のメンバーデバイスを持つ別の使用可能なストライプグループを構成する必要があります。

デバイスを削除するには、次のように進めます。

- [ファイルシステムメタデータおよびデータがバックアップされていることの確認](#)
- [マウントした高パフォーマンスのファイルシステムからのデバイスの削除](#)。

ファイルシステムメタデータおよびデータがバックアップされていることの確認

次のタスクを実行します。

- [samexplorer の実行](#)
- [ファイルシステム用の回復ポイントファイルの作成](#)。

samexplorer の実行

1. ファイルシステムのサーバーホストに *root* としてログインします。

この例では、メタデータサーバーホストは *server1* という名前です。

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

2. *samexplorer* レポートを作成します。コマンド *samexplorer path/hostname.YYYYMMDD.hhmmz.tar.gz* を使用します。ここでは:
 - *path* は選択したディレクトリへのパスです。
 - *hostname* は Oracle HSM ファイルシステムのホスト名です。

- `YYYYMMDD.hmmz` は日付とタイムスタンプです。

このファイルはデフォルトで `/tmp/`

`SAMreport.hostname.YYYYMMDD.hmmz.tar.gz` と呼ばれます。この例では、ディレクトリ `/zfs1/tmp/` を使用します。ここで `/zfs1` は、Oracle HSM ファイルシステムと共通のコンポーネントを持たないファイルシステムです (次のコマンドは 1 行で入力し、改行はバックスラッシュ文字でエスケープします)。

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

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

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

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

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

3. ファイルシステム用の回復ポイントファイルを作成します。

ファイルシステム用の回復ポイントファイルの作成

1. ファイルシステムのサーバーホストに `root` としてログインします。

この例では、メタデータサーバーホストは `server1` という名前です。

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

2. 回復ポイントが格納される場所を選択します。選択した場所では、バックアップするファイルシステムでデバイスを共有してはならず、異常に大きいファイルを格納できる容量が必要です。

削除する予定のデバイスには、アーカイブされていないファイルが含まれていることがあります。このようなファイルは単一のコピーとして存在するだけであるため、少なくとも一部のデータおよびメタデータを格納する回復ポイントファイルを作成する必要があります。これにより、回復ポイントファイルのサイズが大幅に増加する可能性があります。

この例では、Oracle HSM ファイルシステム `/zfs1` と共通のコンポーネントを持たないファイルシステムに、サブディレクトリ `tmp/` を作成します。

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

3. ファイルシステムのルートディレクトリに移動します。

この例では、マウントポイントディレクトリ `/hsm/hsmfs1` に移動します。

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

4. ファイルシステムメタデータおよびアーカイブされていないデータをバックアップします。コマンド `samfsdump -f -u recovery-point` を使用します。ここで `recovery-point` は、最終的な回復ポイントのパスおよびファイル名です。

`-u` オプションを付けると、回復ポイントにアーカイブされていないファイルのデータ部分が追加されます。これにより、ファイルのサイズが大幅に増加する可能性があります。

この例では、ディレクトリ `/zfs1/tmp/` に、`hsmfs1-20140313.025215` という `hsmfs1` ファイルシステム用の回復ポイントファイルを作成します。コマンド `ls -l` を使用して結果を確認します (次の2つ目のコマンドは1行で入力し、改行はバックスラッシュ文字でエスケープします)。

```
[server1]root@solaris:~# cd /hsm/hsmfs1
[server1]root@solaris:~# samfsdump -f /
/zfs1/tmp/hsm/hsmfs1-`date '+%Y%m%d.%H%M%S'` -T /hsm/hsmfs1
samfsdump statistics:
    Files:                10010
```

```

Directories:          2
Symbolic links:      0
Resource files:      0
Files as members of hard links :    0
Files as first hard link :    0
File segments:       0
File archives:       10010
Damaged files:       0
Files with data:     0
File warnings:       0
Errors:              0
Unprocessed dirs:   0
File data bytes:     0

[server1]root@solaris:~# ls -l /zfs1/tmp/hsmfs1*
-rw-r--r-- 1 root other 5376517 Mar 13 02:52 /zfs1/tmp/hsm/hsmfs1-20140313.025215
[server1]root@solaris:~#

```

5. ここで、マウントした高パフォーマンスのファイルシステムからデバイスを削除します。

マウントした高パフォーマンスのファイルシステムからのデバイスの削除

デバイスは一度に1つずつ削除する必要があります。デバイスごとに、次のように進めます。

1. ファイルシステムのサーバーホストに *root* としてログインします。

この例では、メタデータサーバーホストは *server1* という名前です。

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

2. */etc/opt/SUNWsamfs/mcf* ファイルを開き、削除する必要のあるデバイスの装置の順序番号を書き留めます。

次の例では、*vi* エディタを使用します。*hsmfs1* ファイルシステムの装置一覧からデバイス */dev/dsk/c1t4d0s7* を削除する必要があります。装置の順序番号は *104* です。

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

3. デバイスの削除を試みる前に、ファイルシステム内の残りのデバイスが、削除する予定のデバイスから移動する必要のあるファイルを受け入れることができることを確認します。

- 残りのデバイスに十分な容量があることを確認します。
- デバイスがストライプグループである場合は、ファイルシステムに同等の構成を持つ別のストライプグループが含まれていることを確認します。

たとえば、削除する予定のストライプグループが4つの装置番号を持っている場合は、状態が「ON」で、4つの装置番号を持っている別のストライプグループが必要です。

4. 変更する予定のファイルシステムにバージョン 2A スーパーブロックが含まれていることを確認します。コマンド `samfsinfo filesystem-name` を使用します。ここで `filesystem-name` はファイルシステムの名前です。

この例では、ファイルシステム `hsmfs1` で `version:2A` スーパーブロックが使用されています。

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

...

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

5. ファイルシステムにバージョン 2A スーパーブロックが含まれていない場合は、ここで終了します。このファイルシステムがマウントされている場合は、デバイスを削除できません。
6. Oracle HSM アーカイブファイルシステムからデバイスを削除する場合は、削除するディスクデバイスからアーカイブ済みファイルをすべて解放します。コマンド `samcmd release equipment-number` を使用します。ここで `equipment-number` は、`/etc/opt/SUNwsamfs/mcf` ファイル内のデバイスを識別する装置の順序番号です。

デバイスがストライプグループである場合は、グループ内の任意のデバイスの装置番号を指定します。

Oracle HSM ソフトウェアは、新しいファイルが格納されないように、指定したデバイスの状態を「`noalloc`」(割り当てなし)に変更し、事前にアーカイブされたファイルの解放を開始します。デバイスにアーカイブされていないファイルが1つも含まれなくなると、ソフトウェアはファイルシステム構成からデバイスを削除して、状態を「`off`」に変更します。

この例では、アーカイブファイルシステム `hsmfs1` 内のデバイス `104` からファイルを解放します。

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

7. Oracle HSM 非アーカイブファイルシステムからデバイスを削除する場合は、削除するディスクデバイス外に残りの有効なファイルをすべて移動します。コマンド `samcmd remove equipment-number` を使用します。ここで `equipment-number` は、`/etc/opt/SUNwsamfs/mcf` ファイル内のデバイスを識別する装置の順序番号です。

Oracle HSM ソフトウェアは、新しいファイルが格納されないように、指定したデバイスの状態を「`noalloc`」(割り当てなし)に変更し、有効なデータが含まれるファイルをファイルシステム内の残りのデバイスに移動し始めます。すべてのファイルが移動されると、ソフトウェアはファイルシステム構成からデバイスを削除して、状態を「`off`」に変更します。

この例では、デバイス `104` 外にファイルを移動します。

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

8. 選択したプロセス `samcmd remove` または `samcmd release` の進捗状況をモニターします。コマンド `samcmd m` を使用するか、またはログファイルと `/var/opt/SUNWsamfs/trace/sam-shrink` ファイルを監視します (あるいは両方を行います)。

`release` プロセスは、アーカイブメディアにコピーされたファイルに関連付けられている容量のみを解放するため、すべてのファイルがアーカイブされている場合は瞬時に完了します。`remove` プロセスでは、ディスクデバイス間でファイルを移動する必要があるため、データ量およびファイル数によっては非常に長い時間がかかります。

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

ty	eq	status	use	state	ord	capacity	free	ra	part	high	low
ms	100	m----2-----	27%	on		3.691G	2.628G	1M	16	80%	70%
md	101		27%	on	0	959.938M	703.188M				
md	102		28%	on	1	899.938M	646.625M				
md	103		13%	on	2	959.938M	834.250M				
md	104		0%	noalloc	3	959.938M	959.938M				

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

9. `samcmd release` を使用していて、ターゲットデバイスの状態が「`off`」になっていない場合は、デバイス上にアーカイブされていないファイルが存在します。アーカイバが実行され、アーカイブが完了するまで待機します。次に、コマンド `samcmd release` を再度使用します。コマンド `samcmd a` を使用すると、アーカイブの進捗状況を確認できます。

`release` プロセスは、アーカイブされていないファイルがアーカイブされるまでディスク領域を解放できません。

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

```
Archiver status samcmd      5.4 14:12:14 Mar  1 2014
samcmd on server1
```

```

sam-archiverd: Waiting for resources
sam-arfind: hsmfs1 mounted at /hsm/hsmfs1
Files waiting to start      4  schedule      2  archiving      2
[server1]root@solaris:~#

```

10. 1つ以上のアーカイブされていないファイルをアーカイブできないために `samcmd release` に失敗した場合は、アーカイブされていないファイルを別のデバイスに移動します。スタンドアロンの非アーカイブファイルシステムからデバイスを削除するときと同様に、コマンド `samcmd remove equipment-number` を使用します。

この例では、デバイス 104 外にファイルを移動します。

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

11. デバイスの状態が「*off*」に変更されたら、テキストエディタで `/etc/opt/SUNWsamfs/mcf` ファイルを開き、ファイルシステムを検索し、変更が反映されるように装置一覧を更新します。ファイルを保存して、エディタを閉じます。

この例では、`samcmd m` は 104 が *off* であることを示しています。したがって、`vi` エディタを使用して `mcf` ファイルを開きます。`hsmfs1` ファイルシステムの装置一覧からデバイス 104 のエントリを削除して、変更を保存します。

```

[server1]root@solaris:~# samcmd m

```

ty	eq	status	use	state	ord	capacity	free	ra	part	high	low
ms	100	m----2-----	27%	on		3.691G	2.628G	1M	16	80%	70%
md	101		27%	on	0	959.938M	703.188M				
md	102		28%	on	1	899.938M	646.625M				
md	103		13%	on	2	959.938M	834.250M				
md	104		0%	off	3	959.938M	959.938M				

```

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

```

# Equipment	Equipment	Equipment	Family	Device	Additional
# Identifier	Ordinal	Type	Set	State	Parameters
#-----	-----	-----	-----	-----	-----
hsmfs1	100	ms	hsmfs1	on	
/dev/dsk/c1t3d0s3	101	md	hsmfs1	on	
/dev/dsk/c1t4d0s5	102	md	hsmfs1	on	

```
/dev/dsk/c1t3d0s7 103 md hsmfs1 on
:wq
[server1]root@solaris:~#
```

12. `sam-fsd` コマンドを実行して、変更した `mcf` ファイルにエラーがないかどうかを確認し、検出された場合はエラーを修正します。

エラーが発生した場合は、`sam-fsd` コマンドが停止します。

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

13. `mcf` ファイルを再度読み取り、ソフトウェア自体を適宜再構成するように Oracle HSM software に指示します。

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

14. ここで停止します。

Oracle HSM 共有ファイルシステムの管理

このセクションでは、次のタスクの概要について説明します。

- [共有ファイルシステムのマウントとマウント解除](#)
- [共有ファイルシステムのホスト構成の変更](#)
- [アクティブなメタデータサーバーから潜在的なメタデータサーバーへの切り替え](#)
- [非共有ファイルシステムから共有ファイルシステムへの変換](#)
- [共有ファイルシステムから非共有ファイルシステムへの変換。](#)

共有ファイルシステムのマウントとマウント解除

共有ファイルシステムをマウントまたはマウント解除するときは、メタデータサーバーとクライアントをマウントまたはマウント解除する順序が重要です。

フェイルオーバーを行うため、メタデータサーバーとすべての潜在的なメタデータサーバーでマウントオプションは同じにしておく必要があります。たとえば、マウントオプションを含む `samfs.cmd` ファイルを作成し、そのファイルをすべてのホストにコピーできます。

共有ファイルシステムのマウントの詳細については、*mount_samfs* のマニュアルページを参照してください。

共有ファイルシステムのマウント

1. Oracle HSM メタデータサーバーおよびクライアントホストに *root* としてログインします。

この例では、*sharefs* ファイルシステムのメタデータサーバーホスト *sharefs-mds* にログインします。次に各クライアント *sharefs-client1* および *sharefs-client2* に対して端末ウィンドウを開きます。ssh (セキュアシェル) を使用してログインします。

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

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

2. ファイルシステムの Solaris */etc/vfstab* ファイルにエントリが存在する場合は、コマンド *mount mountpoint* を使用して、メタデータサーバーホスト上に共有ファイルシステムをマウントします。ここで *mountpoint* は、ホストのルートファイルシステム上のマウントポイントディレクトリです。

必ず、クライアント上でファイルシステムをマウントする前に、まずメタデータサーバーホスト上にファイルシステムをマウントします。

この例では、*sharefs* ファイルシステムの */etc/vfstab* ファイルに次のエントリが存在します。

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

したがって、マウントポイントパラメータを指定するだけでファイルシステムをマウントできます。

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

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

3. ファイルシステムの Solaris `/etc/vfstab` ファイルにエントリが存在しない場合は、コマンド `mount -F samfs -o shared mountpoint` を使用して、メタデータサーバーホスト上に共有ファイルシステムをマウントします。ここで `mountpoint` は、ホストのルートファイルシステム上のマウントポイントディレクトリです。

必ず、クライアント上でファイルシステムをマウントする前に、まずメタデータサーバーホスト上にファイルシステムをマウントします。

この例では、`sharefs` ファイルシステムの `/etc/vfstab` ファイルに次のエントリが存在しません。

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

4. ファイルシステムの Solaris `/etc/vfstab` ファイルにエントリが存在する場合は、コマンド `mount mountpoint` を使用して、各クライアントホスト上に共有ファイルシステムをマウントします。ここで `mountpoint` は、ホストのルートファイルシステム上のマウントポイントディレクトリです。

クライアントホストには、任意の順序でファイルシステムをマウントできます。

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

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

5. ファイルシステムの Solaris `/etc/vfstab` ファイルにエントリが存在しない場合は、コマンド `mount -F samfs -o shared mountpoint` を使用して、各クライアントホスト上に共有ファイルシステムをマウントします。ここで `mountpoint` は、ホストのルートファイルシステム上のマウントポイントディレクトリです。

クライアントホストには、任意の順序でファイルシステムをマウントできます。

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

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

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

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

6. ここで停止します。

共有ファイルシステムのアンマウント

1. Oracle HSM メタデータサーバーおよびクライアントホストに *root* としてログインします。

この例では、*sharefs* ファイルシステムのメタデータサーバーホスト *sharefs-mds* にログインします。次に各クライアント *sharefs-client1* および *sharefs-client2* に対して端末ウィンドウを開き、*ssh* (セキュアシェル) を使用してログインします。

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

```
Password:
```

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

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

```
Password:
```

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

2. ファイルシステムが NFS または SAMBA で共有されている場合は、マウントを解除する前に、そのファイルシステムの共有を解除します。メタデータサーバーでコマンド *unshare mount-point* を使用します。ここで *mount-point* は、Oracle HSM ファイルシステムのマウントポイントディレクトリです。

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

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

3. 各クライアントから Oracle HSM 共有ファイルシステムをアンマウントします。コマンド *umount mount-point* を使用します。ここで *mount-point* は、Oracle HSM ファイルシステムのマウントポイントディレクトリです。

詳細については、*umount_samfs* のマニュアルページを参照してください。この例では、2つのクライアント (*sharefs-client1* と *sharefs-client2*) から *sharedqfs1* をアンマウントします。

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

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

4. メタデータサーバーから Oracle HSM 共有ファイルシステムをアンマウントします。コマンド *umount -o await_clients=interval mount-point* を使用します。ここで、*mount-point* は Oracle HSM ファイルシステムのマウントポイントディレクトリ、*interval* は *-o await_clients* オプションで実行を遅延させる秒数です。

Oracle HSM 共有ファイルシステムのメタデータサーバー上で *-o await_clients* オプションを付けて *umount* コマンドを発行すると、*umount* はクライアントが共有をアンマウントする時間を持てるように、指定された秒数間待機します。非共有ファイルシステムをアンマウントする場合や、Oracle HSM クライアント上でコマンドを発行する場合は影響がありません。詳細については、*umount_samfs* のマニュアルページを参照してください。

この例では、サーバーから */sharefs* ファイルシステムをアンマウントします。クライアントにアンマウントする時間を、60 秒間与えます。

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

5. ここで停止します。

共有ファイルシステムのホスト構成の変更

このセクションでは、共有ファイルシステムのクライアントとして追加ホストを構成する手順、および既存のクライアントの構成を解除する手順について説明します。次のセクションが含まれます。

- [追加ファイルシステムクライアントの構成](#)
- [共有ファイルシステム構成からのホストの削除](#)
- [分散テープ入出力用のデータムーバークライアントの構成](#)
- [永続的なバインドを使用したテープドライブの接続。](#)

追加ファイルシステムクライアントの構成

クライアントホストを共有ファイルシステムに追加するプロセスは、3つの部分に分かれています。

- まず、共有ファイルシステム構成にホスト情報を追加します。
- 次に、ホストのオペレーティングシステムに固有の手順 (Solaris または Linux) を使用して、ホスト上に共有ファイルシステムを構成します。
- 最後に、ホストのオペレーティングシステムに固有の手順 (Solaris または Linux) を使用して、ホスト上に共有ファイルシステムをマウントします。

共有ファイルシステム構成へのホスト情報の追加

1. Oracle HSM メタデータサーバーに *root* としてログインします。

この例では、Oracle HSM 共有ファイルシステムは *sharefs*、メタデータサーバーホストは *sharefs-mds* です。

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

2. ファイル */etc/opt/SUNWsamfs/hosts.filesystem* をバックアップします。ここで *filesystem* はクライアントホストの追加先であるファイルシステムの名前です。

次のコマンドは1行で入力し、改行はバックスラッシュ文字でエスケープします。

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

- 共有ファイルシステムがマウントされている場合は、アクティブなメタデータサーバーからコマンド `samsharefs filesystem` を実行して、出力をファイル `/etc/opt/SUNWsamfs/hosts.filesystem` にリダイレクトします。ここで `filesystem` は、クライアントホストの追加先であるファイルシステムの名前です。

`samsharefs` コマンドは、Oracle HSM 共有ファイルシステムのホスト構成を表示します。出力をファイルにリダイレクトすると、新しい `hosts` ファイルが作成されます(次のコマンドは1行で入力し、改行はバックスラッシュ文字でエスケープします)。

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

- 共有ファイルシステムがマウントされていない場合は、アクティブまたは潜在的なメタデータサーバーからコマンド `samsharefs -R filesystem` を実行して、出力をファイル `/etc/opt/SUNWsamfs/hosts.filesystem` にリダイレクトします。ここで `filesystem` は、クライアントホストの追加先であるファイルシステムの名前です。

`samsharefs -R` コマンドは、アクティブまたは潜在的なメタデータサーバーからのみ実行できます(詳細については、`samsharefs` のマニュアルページを参照)。`samsharefs` コマンドは、Oracle HSM 共有ファイルシステムのホスト構成を表示します。出力をファイルにリダイレクトすると、新しい `hosts` ファイルが作成されます。この例では、メタデータサーバー `sharefs-mds` からコマンドを実行します(次のコマンドは1行で入力し、改行はバックスラッシュ文字でエスケープします)。

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

- 新たに作成された `hosts` ファイルをテキストエディタで開きます。

次の例では、`vi` エディタを使用します。ホスト構成には、アクティブなメタデータサーバー `sharefs-mds`、潜在的なメタデータサーバーでもある 1 つのクライアント `sharefs-mds_alt`、および他の 2 つのクライアント `sharefs-client1` と `sharefs-client2` が含まれています。

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

- hosts ファイルに新しいクライアントホストの行を追加し、ファイルを保存して、エディタを閉じます。

この例では、ホスト `sharefs-client3` のエントリを追加します。

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

- ファイルシステムがマウントされている場合は、アクティブなメタデータサーバーからファイルシステムを更新します。コマンド `samsharefs -u filesystem` を使用します。ここで `filesystem` はクライアントホストの追加先であるファイルシステムの名前です。

`samsharefs` コマンドは、修正されたホストファイルを再度読み取り、構成を更新します。

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

8. ファイルシステムがマウントされていない場合は、アクティブなメタデータサーバーまたは潜在的なメタデータサーバーからファイルシステムを更新します。コマンド `samsharefs -R -u filesystem` を使用します。ここで `filesystem` はクライアントホストの追加先であるファイルシステムの名前です。

`samsharefs` コマンドは、修正されたホストファイルを再度読み取り、構成を更新します。

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

9. クライアントとして Solaris ホストを追加する場合は、[「Solaris クライアントでの共有ファイルシステムの構成」](#)に進みます。
10. クライアントとして Linux ホストを追加する場合は、[「Linux クライアントホストでの共有ファイルシステムの構成」](#)に進みます。

Solaris クライアントでの共有ファイルシステムの構成

1. 共有ファイルシステムクライアントで、`root` としてログインします。

この例では、Oracle HSM 共有ファイルシステムは `sharefs`、クライアントホストは `sharefs-client1` です。

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

2. 端末ウィンドウで、共有ファイルシステムの構成情報を取得します。コマンド `samfsconfig device-path` を使用します。ここで `device-path` は、コマンドがファイルシステムディスクデバイスの検索を開始する場所 (`/dev/dsk/*` や `/dev/zvol/dsk/rpool/*` など) です。

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


3. ホストがファイルシステムのメタデータデバイスへのアクセス権を持っているため、潜在的なメタデータサーバーとしての使用に適している場合は、*samfsconfig* の出力が、ファイルシステムのメタデータサーバーで作成された *mcf* ファイルに酷似しています。

この例では、ホスト *sharefs-client1* がメタデータデバイス (装置タイプ *mm*) へのアクセス権を持っているため、コマンドの出力にサーバー *sharefs-mds* 上の *mcf* ファイルに一覧表示されるものと同じ装置が表示されます。ホストで割り当てられたデバイスコントローラ番号のみが異なります。

```
[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. ホストがファイルシステムのメタデータデバイスへのアクセス権を持っていない場合は、*samfsconfig* コマンドではメタデータデバイスを検索できません。したがって、検出された Oracle HSM デバイスをファイルシステム構成に合わせることはできません。コマンドの出力では、*Missing Slices* の下に *Ordinal 0* (メタデータデバイス) が一覧表示されますが、ファイルシステムファミリセットを識別する行を含めることができず、データデバイスの一覧がコメントアウトされています。

この例では、ホスト *sharefs-client2* はデータデバイスへのアクセス権のみを持っています。したがって、*samfsconfig* の出力は次のように表示されます。

```
[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. `samfsconfig` の出力から共有ファイルシステムのエントリをコピーします。次に、2つ目のウィンドウからテキストエディタで `/etc/opt/SUNWsamfs/mcf` ファイルを開き、コピーしたエントリをファイルにペーストします。

1つ目の例では、ホスト `sharefs-client1` がファイルシステムのメタデータデバイスへのアクセス権を持っているため、`mcf` ファイルの始まりは次のように表示されます。

```
[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   -
```

2つ目の例では、ホスト `sharefs-client2` がファイルシステムのメタデータデバイスへのアクセス権を持っていないため、`mcf` ファイルの始まりは次のように表示されます。

```
[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. ホストがファイルシステムのメタデータデバイスへのアクセス権を持っている場合は、共有ファイルシステムのエントリの「*Additional Parameters*」フィールドに、`shared` パラメータを追加します。

1つ目の例では、ホスト `sharefs-client1` はメタデータへのアクセス権を持っています。

```
[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. ホストがファイルシステムのメタデータデバイスへのアクセス権を持っていない場合は、共有ファイルシステムの行を追加し、*shared* パラメータを追加します。

```

[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. ホストがファイルシステムのメタデータデバイスへのアクセス権を持っていない場合は、メタデータデバイスの行を追加します。*Equipment Identifier* フィールドを *nodev* (デバイスなし) に設定し、残りのフィールドはメタデータサーバーの場合とまったく同じ値に設定します。

```

[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. ホストがファイルシステムのメタデータデバイスへのアクセス権を持っていない場合は、データデバイスのエントリのコメントを解除します。

```
[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. すべてのデバイスで「*Device State*」フィールドが「*on*」に設定されていることを確認し、*mcf* ファイルを保存して、エディタを閉じます。

1つ目の例では、ホスト *sharefs-client1* がファイルシステムのメタデータデバイスへのアクセス権を持っているため、*mcf* ファイルの終わりは次のように表示されます。

```
[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:~#
```

2つ目の例では、ホスト *sharefs-client2* がファイルシステムのメタデータデバイスへのアクセス権を持っていないため、*mcf* ファイルの終わりは次のように表示されます。

```
[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. `sam-fsd` コマンドを実行して、`mcf` ファイルにエラーがないかどうかを確認し、エラーがあれば修正します。

`sam-fsd` は、Oracle HSM 構成ファイルを読み取る初期化コマンドです。エラーが発生した場合は停止します。この例では、`sharefs-client1` 上で `mcf` ファイルを確認します。

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

12. 次に、Solaris ホストに共有ファイルシステムをマウントします。

Solaris ホストでの共有ファイルシステムのマウント

1. 共有ファイルシステムホストで、`root` としてログインします。

この例では、Oracle HSM 共有ファイルシステムは `sharefs`、ホストは `sharefs-client1` という名前のクライアントです。

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

2. オペレーティングシステムの `/etc/vfstab` ファイルをバックアップします。

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

3. テキストエディタで `/etc/vfstab` ファイルを開き、共有ファイルシステムの行を追加します。

この例では、`vi` テキストエディタでファイルを開き、`sharefs` ファミリセットデバイスの行を追加します。

```
[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

```

- クライアント上のファイルシステムを共有ファイルシステムとしてマウントするには、共有ファイルシステムの *vfstab* エントリの「*Mount Options*」列に *shared* オプションを入力します。

現在のクライアントで共有ファイルシステム *sharefs* を読み取り専用でマウントする必要がある場合は、*vfstab* エントリを次の例で示すように編集します。

```

#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

```

- セパレータとしてコンマを使用して、その他の必要なマウントオプションを追加し、その他の必要な変更を */etc/vfstab* ファイルに加えます。次に、*/etc/vfstab* ファイルを保存します。

この例では、追加のマウントオプションはありません。

```

#File
#Device    Device    Mount    System  fsck  Mount    Mount
#to Mount  to fsck  Point    Type    Pass  at Boot  Options
#-----  -----  -----  -----  ----  -----  -----
/devices   -        /devices devfs   -     no       -
/proc     -        /proc    proc    -     no       -

```

```
...
sharefs - /sharefs samfs - no shared
:wq
[sharefs-client1]root@solaris:~#
```

6. `/etc/vfstab` ファイルで指定されたマウントポイントを作成し、そのマウントポイントに対するアクセス権を設定します。

マウントポイントのアクセス権は、メタデータサーバーおよびその他のすべてのクライアントと同じにする必要があります。ユーザーはマウントポイントポイントディレクトリに移動し、マウントしたファイルシステム内のファイルにアクセスするための実行権 (x) を持っている必要があります。この例では、`/sharefs` マウントポイントディレクトリを作成し、アクセス権を `755 (-rwxr-xr-x)` に設定します。

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

7. 共有ファイルシステムをマウントします。

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

8. 分散テープ入出力データムーバーとして潜在的なメタデータサーバーホストを追加する場合は、「[分散テープ入出力用のデータムーバークライアントの構成](#)」に進みます。
9. ここで停止します。

Linux クライアントホストでの共有ファイルシステムの構成

1. Linux クライアント上で、`root` としてログインします。

この例では、Oracle HSM 共有ファイルシステムは `sharefs`、ホストは `sharefs-clientL` という名前の Linux クライアントです。

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

2. 端末ウィンドウで、`samfsconfig device-path` コマンドを使用して共有ファイルシステムに関する構成情報を取得します。ここで `device-path` は、コマンドがファイルシステムディスクデバイスの検索を開始する場所 (`/dev/*` など) です。

Linux ホストがファイルシステムのメタデータデバイスへのアクセス権を持っていないため、`samfsconfig` コマンドではメタデータデバイスを検索できません。したがって、検出された Oracle HSM デバイスをファイルシステム構成に合わせることはできません。コマンドの出力では、*Missing Slices* の下に *Ordinal 0* (メタデータデバイス) が一覧表示されますが、ファイルシステムファミリーセットを識別する行を含めることができず、データデバイスの一覧がコメントアウトされています。

この例では、Linux ホスト `sharefs-clientL` 用の `samfsconfig` 出力が次のように表示されます。

```
[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. `samfsconfig` の出力から共有ファイルシステムのエントリをコピーします。次に、2つ目のウィンドウからテキストエディタで `/etc/opt/SUNWsamfs/mcf` ファイルを開き、コピーしたエントリをファイルにペーストします。

この例では、Linux ホスト `sharefs-clientL` 用の `mcf` ファイルの始まりが次のように表示されます。

```
[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. *mcf* ファイルに共有ファイルシステムの行を挿入し、*shared* パラメータを追加します。

```
# 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. *mcf* ファイルに、ファイルシステムのメタデータデバイスの行を挿入します。Linux ホストはメタデータデバイスへのアクセス権を持っていないため、*Equipment Identifier* フィールドを *nodev* (デバイスなし) に設定し、残りのフィールドはメタデータサーバーの場合とまったく同じ値に設定します。

```
# 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. *mcf* ファイルで、Linux データデバイスのエントリをコメント解除します。

```
# 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  -
```

- すべてのデバイスで「*Device State*」フィールドが「*on*」に設定されていることを確認し、*mcf* ファイルを保存します。

```
# 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 ~]#
```

- sam-fsd* コマンドを実行して、*mcf* ファイルにエラーがないかどうかを確認し、エラーがあれば修正します。

sam-fsd は、Oracle HSM 構成ファイルを読み取る初期化コマンドです。エラーが発生した場合は停止します。この例では、Linux クライアント *sharefs-clientL* 上で *mcf* ファイルを確認します。

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

- 次に、Linux ホストに共有ファイルシステムをマウントします。

Linux クライアントホストでの共有ファイルシステムのマウント

- Linux クライアント上で、*root* としてログインします。

この例では、Oracle HSM 共有ファイルシステムは *sharefs*、ホストは *sharefs-clientL* という名前の Linux クライアントです。

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

- オペレーティングシステムの */etc/fstab* ファイルをバックアップします。

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

3. テキストエディタで `/etc/fstab` ファイルを開き、共有ファイルシステムの行を開始します。

この例では、`vi` テキストエディタを使用して、`sharefs` ファミリセットデバイスの行を追加します。

```
[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. ファイルの 4 列目で、必須の `shared` マウントオプションを追加します。

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

5. ファイルの 4 列目で、セパレータとしてコンマを使用して、その他の必要なマウントオプションを追加します。

Linux クライアントでは、次の追加マウントオプションがサポートされています。

- `rw, ro`
- `retry`
- `meta_timeo`
- `rdlease, wrlease, aplease`
- `minallocsz, maxallocsz`

- *noauto, auto*

この例では、オプション *noauto* を追加します。

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

6. ファイルの残りの 2 列には、それぞれゼロ (0) を入力します。次に、*/etc/fstab* ファイルを保存します。

```
#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. */etc/fstab* ファイルで指定されたマウントポイントを作成し、そのマウントポイントに対するアクセス権を設定します。

マウントポイントのアクセス権は、メタデータサーバーおよびその他のすべてのクライアントと同じにする必要があります。ユーザーはマウントポイントポイントディレクトリに移動し、マウントしたファイルシステム内のファイルにアクセスするための実行権 (x) を持っている必要があります。この例では、*/sharefs* マウントポイントディレクトリを作成し、アクセス権を *755 (-rwxr-xr-x)* に設定します。

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

- 共有ファイルシステムをマウントします。コマンド `mount mountpoint` を使用します。ここで `mountpoint` は、`/etc/fstab` ファイルで指定されたマウントポイントディレクトリです。

例で示すように、`mount` コマンドは警告を生成します。これは通常の動作であり、無視できます。

```
[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
```

- ここで停止します。

共有ファイルシステム構成からのホストの削除

次で説明するように、共有ファイルシステムからホストを削除しても、単にサーバー構成からホストが削除されるだけです (完全にホストの構成を解除するには、ソフトウェアおよび構成ファイルをアンインストールしてください)。

ファイルシステムの `hosts` ファイルからのホストの削除

- Oracle HSM メタデータサーバーに `root` としてログインします。

この例では、Oracle HSM 共有ファイルシステムは `sharefs`、メタデータサーバーホストは `sharefs-mds` です。

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

- 各クライアントに `root` としてログインし、共有ファイルシステムをアンマウントします。

潜在的なメタデータサーバー自体もクライアントであることを忘れないください。この例には、`sharefs-client1`、`sharefs-client2`、および `sharefs-mds_alt` の3つのクライアントと、潜在的なメタデータサーバーがあります。各クライアントに対して、`ssh` を使用してログインし、ファイルシステム `sharefs` をアンマウントし、`ssh` セッションを終了します。

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

3. メタデータサーバー上で、共有ファイルシステムをアンマウントします。

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

4. メタデータサーバー上で、ファイル `/etc/opt/SUNWsamfs/hosts.filesystem` の名前を `/etc/opt/SUNWsamfs/hosts.filesystem.bak` に変更します。ここで `filesystem` は、クライアントホストの削除元であるファイルシステムの名前です。

次のコマンドは1行で入力し、改行はバックスラッシュ文字でエスケープします。

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

5. 現在の共有ファイルシステムのホスト構成をファイル上に取得します。メタデータサーバーからコマンド `samsharefs -R filesystem` を実行して、出力をファイル `/etc/opt/SUNWsamfs/hosts.filesystem` にリダイレクトします。ここで `filesystem` は、クライアントホストの追加先であるファイルシステムの名前です。

`samsharefs` コマンドは、指定した Oracle HSM 共有ファイルシステムのホスト構成を表示します。出力をファイルにリダイレクトすると、新しい `hosts` ファイルが作成されます。この例では、メタデータサーバー `sharefs-mds` からコマンドを実行します。

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

6. 新たに作成された `hosts` ファイルをテキストエディタで開きます。

次の例では、`vi` エディタを使用します。クライアント `sharefs-client3` を削除する必要があります。

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

7. `hosts` ファイルで、削除する必要のあるクライアントホストに対応する行を削除します。次に、ファイルを保存し、エディタを閉じます。

この例では、ホスト `sharefs-client3` のエントリを削除します。

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

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

- 修正されたホストファイルを使用してファイルシステムを更新します。メタデータサーバーからコマンド `samsharefs -R -u filesystem` を使用します。ここで `filesystem` はクライアントホストの削除元であるファイルシステムの名前です。

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

- メタデータサーバーホスト上で、共有ファイルシステムをマウントします。

この例では、`/etc/vfstab` ファイルに `sharefs` ファイルシステムのエンタリが含まれているため、単純なマウント構文を使用します (詳細な情報については、`mount_samfs` のマニュアルページを参照)。

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

- 各クライアントホスト上で、共有ファイルシステムをマウントします。

潜在的なメタデータサーバー自体もクライアントであることを忘れないください。この例には、`sharefs-client1`、`sharefs-client2`、および `sharefs-mds_alt` の3つのクライアントと、潜在的なメタデータサーバーがあります。各クライアントに対して、`ssh` を使用してログインし、ファイルシステム `sharefs` をアンマウントし、`ssh` セッションを終了します。

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



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

11. ここで停止します。

分散テープ入出力用のデータムーバークライアントの構成

Oracle HSM リリース 6.1 以降、Solaris 11 以上が実行されている共有アーカイブファイルシステムのクライアントは、テープドライブを接続して、ファイルシステムの代わりにテープ入出力を実行できます。これらのデータムーバークライアント間でテープ入出力を分散させると、サーバーのオーバーヘッドが大幅に削減されるため、ファイルシステムのパフォーマンスが向上し、Oracle HSM 実装のスケールリング時の柔軟性が大幅に向上します。アーカイブの必要性が高くなれば、Oracle HSM メタデータサーバーをより強力なシステムに交換するか (垂直スケールリング)、より多くのクライアント間で負荷を分散させるか (水平スケールリング) の選択肢があります。

データムーバークライアントの構成

分散テープ入出力用のクライアントを構成するには、次のように進めます。

1. 分散入出力用に使用されるすべてのデバイスをクライアントに接続します。
2. まだ実行していない場合は、手順「[永続的なバインドを使用したテープドライブの接続](#)」を実行します。その後、ここに戻ります。
3. 共有アーカイブファイルシステムのメタデータサーバーに *root* としてログインします。

この例では、ホスト名は *samsharefs-mds* です。

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

4. メタデータサーバーで Oracle HSM Solaris 11 以上が実行されていることを確認します。

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

5. データムーバーとして動作するすべてのクライアントで、Oracle HSM Solaris 11 以上が実行されていることを確認します。

この例では、各クライアントホスト `samsharefs-client1` および `samsharefs-client2` に対して端末ウィンドウを開き、`ssh` を使用してリモートでログインします。ログインバナーに Solaris バージョンが表示されます。

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

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

6. メタデータサーバーでファイル `/etc/opt/SUNWsamfs/defaults.conf` をテキストエディタで開き、行 `distio =` をコメント解除し、値を `on` に設定して分散入出力を有効にします。

デフォルトでは、「`distio`」は「`off`」(無効)になっています。

この例では、`vi` エディタでファイルを開き、行を追加します。

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

7. 次に、分散入出力に追加するデバイスタイプを特定します。分散入出力でデバイスタイプ `dev` を使用するには、`defaults.conf` ファイルに行 `dev_distio = on` を追加します。分散入出力からデバイスタイプ `dev` を除外するには、行 `dev_distio = off` を追加します。ファイルを保存して、エディタを閉じます。

デフォルトでは、Oracle HSM T10000 ドライブおよび LTO ドライブを分散入出力に追加できません (*ti_distio = on* および *li_distio = on*) が、その他のタイプはすべて除外されています。この例では、LTO ドライブを除外します。

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

8. データムーバーとして動作する各クライアント上で、サーバー上のファイルと一致するように *defaults.conf* ファイルを編集します。

この例では、クライアント *samsharefs-client1* 上の *defaults.conf* ファイルを *vi* を使用して編集し、このファイルを保存してエディタを閉じます。

```
[samsharefs-mds]root@solaris:~# ssh root@samsharefs-client1
Password:
[samsharefs-client1]root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
...
distio = on
li_distio = off
:wq
[samsharefs-client1]root@solaris:~#
[samsharefs-mds]root@solaris:~#
```

9. データムーバーとして動作する各クライアント上で、テキストエディタで */etc/opt/SUNWsamfs/mcf* ファイルを開きます。分散テープ入出力用にメタデータサーバーが使用しているテープデバイスをすべて追加します。デバイスの順序お

よび装置番号がメタデータサーバー上の *mcf* ファイルのものと同一であることを確認します。

この例では、*vi* を使用してクライアント *samsharefs-client1* 上の *mcf* ファイルを編集します。

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

10. データムーバーとして動作するクライアント上に、メタデータサーバー上の */etc/opt/SUNWsamfs/mcf* ファイルに一覧表示されたテープライブラリが構成されている場合は、分散テープ入出力用に使用されているテープデバイスのファミリセット名としてライブラリファミリセットを指定します。ファイルを保存します。

この例では、ホスト上にライブラリが構成されているため、テープデバイスのファミリセット名 *library1* を使用します。

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

```

/dev/rmt/62cbn          903      ti      library1  on
/dev/rmt/63cbn          904      ti      library1  on
:wq
[samsharefs-client1]root@solaris:~#

```

11. データムーバーとして動作するクライアント上に、メタデータサーバー上の `/etc/opt/SUNWsamfs/mcf` ファイルに一覧表示されたテープライブラリが構成されていない場合は、分散テープ入出力用に使用されているテープデバイスのファミリーセット名としてハイフン (-) を使用します。

この例では、ライブラリはホスト上で構成されていません。

```

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

```

12. 特定のアーカイブセットのコピーで分散テープ入出力を有効または無効にする必要がある場合は、テキストエディタでサーバーの `/etc/opt/SUNWsamfs/archiver.cmd` ファイルを開き、`-distio` パラメータをコピーディレクティブに追加します。分散入出力を有効にするには「`-distio on`」、無効にするには「`off`」を設定します。ファイルを保存して、エディタを閉じます。

この例では、`vi` エディタを使用して、コピー 1 の分散入出力を「`off`」、コピー 2 の分散入出力を「`on`」に設定します。

```

[samsharefs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
# archiver.cmd

```

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

13. 各ホストで、*sam-fsd* コマンドを実行して、*mcf* ファイルにエラーがないかどうかを確認し、見つかったエラーを修正します。

sam-fsd は、Oracle HSM 構成ファイルを読み取る初期化コマンドです。エラーが発生した場合は停止します。この例では、Linux クライアント *sharefs-clientL* 上で *mcf* ファイルを確認します。

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

14. サーバーで、変更した構成ファイルを読み取り、ソフトウェア自体を適宜再構成するように、Oracle HSM ソフトウェアに指示します。コマンド *samd config* を使用し、見つかったエラーを修正します。

この例では、サーバー *sharefs-mds* 上でコマンド *samd config* を実行します。

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

15. ここで停止します。

永続的なバインドを使用したテープドライブの接続

潜在的なメタデータサーバーまたは分散入出力データムーバークライアントとして機能するホストを追加する際には、永続的なバインドを使用してリムーバブルメディアデバイスを構成する必要があります。Solaris オペレーティングシステムは、起動時にデバイスが検出される順序でドライブをシステムデバイスツリーに追加します。この順序によって、その他のファイルシステムホストでデバイスが検出される順序や、テープライブラリに物理的にインストールされる順序が反映される場合

と、反映されない場合があります。したがって、その他のホストにバインドするときと同じ方法、およびリムーバブルメディアライブラリにインストールされる時と同じ順序で、デバイスを新しいホストにバインドする必要があります。

次の手順では、必要なステップの概要を示します (完全な情報については、*devfsadm* および *devlinks* のマニュアルページ、および使用している Solaris オペレーティングシステムのバージョンに対応した管理ドキュメントを参照)。

- ライブラリ内のドライブを移動、追加、または削除した場合や、アーカイブ Oracle HSM 共有ファイルシステムに関連付けられているライブラリを交換または再構成した場合は、変更を反映させるために永続的なバインドを更新します。
- 新しいメタデータサーバーまたはデータムーバークライアントをアーカイブ Oracle HSM 共有ファイルシステムに追加する場合は、新しいファイルシステムホストをリムーバブルメディアデバイスに永続的にバインドします

ハードウェア構成への変更を反映させるための永続的なバインドの更新

1. アクティブなメタデータサーバーホストに *root* としてログインします。

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

2. 「[ドライブをライブラリに取り付ける順序の判別](#)」で説明するように、新しいドライブマッピングファイルを作成します。

この例では、*device-mappings.txt* ファイルは次のように表示されます。

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

3. */etc/devlink.tab* ファイルをテキストエディタで開きます。

この例では、*vi* エディタを使用します。

```
[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. ガイドとして *device-mappings.txt* ファイルを使用して、Solaris テープデバイスツリー内の開始ノードをライブラリ内の 1 番目のドライブに再マッピングします。*/etc/devlink.tab* ファイルに、*type=ddi_byte:tape; addr=device_address,0; rmt/node-number/M0* という形式の行を追加します。ここで、*device_address* はデバイスの物理アドレス、*node-number* は Solaris デバイスツリー内のデバイスの位置です。Solaris で自動的に構成されるデバイスとの競合を回避するために、十分に大きいノード番号を選択してください (Solaris ではノード *0* から始まります)。

この例では、ライブラリ内の 1 番目のデバイス *1* のデバイスアドレス *w500104f0008120fe* を書き留め、デバイスが *rmt/1* にあるホストに現在接続されていることを確認します。

```
[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
```

/etc/devlink.tab に、*rmt/60* をライブラリ内の番号 *1* のドライブ *w500104f0008120fe* に再マッピングする 1 行を作成します。

```
[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. メタデータサーバー上のデバイスツリー内のドライブ順序とライブラリへのインストール順序が一致するように、Oracle HSM アーカイブ用に割り当てられているテープデバイスごとに、`/etc/devlink.tab` ファイルへの行の追加を繰り返します。ファイルを保存して、エディタを閉じます。

この例では、残りの3つのデバイス (`w500104f00093c438` にあるライブラリドライブ 2、`w500104f000c086e1` にあるライブラリドライブ 3、`w500104f000c086e1` にあるライブラリドライブ 4) の順序とアドレスを書き留めます。

```
[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
```

次に、デバイスのアドレスを次の3つの Solaris デバイスノードにマッピングして、ライブラリ内と同じ順序を保持します。

```
[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. `/dev/rmt` 内のテープデバイスへの既存のリンクをすべて削除します。

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

7. `/etc/devlink.tab` ファイル内のエントリから、新しい永続的なテープデバイスのリンクを作成します。コマンド `devfsadm -c tape` を使用します。

`devfsadm` コマンドを実行するたびに、`/etc/devlink.tab` ファイルで指定された構成を使用して、そのファイルに指定されたデバイス用に新しいテープデバイスのリンクが作成されます。`-c tape` オプションを指定すると、テープクラスデバイス用の新しいリンクのみが作成されるようにコマンドが制限されます。

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

8. 潜在的なメタデータサーバーおよび共有ファイルシステム構成のデータムーバーで操作を繰り返します。いずれの場合も、`/etc/devlink.tab` ファイルに同じ行を追加し、`/dev/rmt` 内のリンクを削除し、`devfsadm -c tape` を実行します。

この例では、`ssh` を使用して各ホストに順にログインし、同じ 4 つの論理デバイス `rmt/60/M0`、`rmt/61/M0`、`rmt/62/M0`、および `rmt/63/M0` を構成します。

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

```

type=ddi_byte:tape;addr=w500104f00093c438,0;    rmt/61/M0
type=ddi_byte:tape;addr=w500104f000c086e1,0;    rmt/62/M0
type=ddi_byte:tape;addr=w500104f000b6d98d,0;    rmt/63/M0
:wq
[sharefs-client1]root@solaris:~# rm /dev/rmt/*
[sharefs-client1]root@solaris:~# devfsadm -c tape
[sharefs-client1]root@solaris:~# exit
[sharefs-mds]root@solaris:~#

```

9. 「分散テープ入出力用のデータムーバークライアントの構成」または「追加ファイルシステムクライアントの構成」に戻ります。

リムーバブルメディアデバイスへの新しいファイルシステムホストの永続的なバインド

1. ホストに *root* としてログインします。

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

2. 既存のファイルシステムホストが構成されたために、メディアライブラリ内のデバイスの物理的な順序が変更された場合は、「[ドライブをライブラリに取り付ける順序の判別](#)」で説明するように、新しいマッピングファイルを作成します。

この例では、*device-mappings.txt* ファイルは次のように表示されます。

```

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

```

3. テキストエディタで */etc/devlink.tab* ファイルを開きます。

この例では、*vi* エディタを使用します。

```
[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.
...
```

- ガイドとして *device-mappings.txt* ファイルを使用して、Solaris テープドライブツリー内の開始ノード *rmt/node-number* をライブラリ内の 1 番目のドライブに再マッピングします。*/etc/devlink.tab* ファイルに行を、*type=ddi_byte:tape; addr=device_address,0; rmt/node-number/M0* の形式で入力します。ここでは:*device_address* はデバイスの物理アドレス、*node-number* は Solaris デバイスツリー内のデバイスの位置です。Solaris で自動的に構成されるデバイスとの競合を回避するために、十分に大きいノード番号を選択してください (Solaris ではノード 0 から始まります)。

この例では、ライブラリ内の 1 番目のデバイス 1 のデバイスアドレス *w500104f0008120fe* を書き留め、デバイスが *rmt/1* にあるホストに現在接続されていることを確認します。

```
[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
```

/etc/devlink.tab に、*rmt/60* をライブラリ内の番号 1 のドライブ *w500104f0008120fe* に再マッピングする 1 行を作成します。

```
[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. メタデータサーバー上のデバイスツリー内のドライブ順序とライブラリへのインストール順序が一致するように、Oracle HSM アーカイブ用に割り当てられているテープデバイスごとに、`/etc/devlink.tab` ファイルへの行の追加を繰り返します。ファイルを保存します。

この例では、残りの3つのデバイス (`w500104f00093c438` にあるライブラリドライブ 2、`w500104f000c086e1` にあるライブラリドライブ 3、`w500104f000c086e1` にあるライブラリドライブ 4) の順序とアドレスを書き留めます。

```
[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
```

次に、デバイスのアドレスを次の3つの Solaris デバイスノードにマッピングして、ライブラリ内と同じ順序を保持します。

```
[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. `/dev/rmt` 内のテープデバイスへの既存のリンクをすべて削除します。

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

7. `/etc/devlink.tab` ファイル内のエントリから、新しい永続的なテープデバイスのリンクを作成します。コマンド `devfsadm -c tape` を使用します。

`devfsadm` コマンドを実行するたびに、`/etc/devlink.tab` ファイルで指定された構成を使用して、そのファイルに指定されたデバイス用に新しいテープデバイスのリンクが作成されます。`-c tape` オプションを指定すると、テープクラスデバイス用の新しいリンクのみが作成されるようにコマンドが制限されます。

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

8. 潜在的なメタデータサーバーおよび共有ファイルシステム構成内のデータムーバ-のそれぞれで、`/etc/devlink.tab` ファイルに同じ行を追加し、`/dev/rmt` 内のリンクを削除し、`devfsadm -c tape` を実行します。

この例では、`ssh` を使用して潜在的なメタデータサーバーホスト `sharefs-mds_alt` およびクライアントホスト `sharefs-client1` にログインします。次に、それぞれについて同じ 4 つの論理デバイス `rmt/60/M0`、`rmt/61/M0`、`rmt/62/M0`、および `rmt/63/M0` を構成します。

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

```

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

```

9. 「分散テープ入出力用のデータムーバークライアントの構成」または「追加ファイルシステムクライアントの構成」に戻ります。

アクティブなメタデータサーバーから潜在的なメタデータサーバーへの切り替え

このセクションの手順を実行すると、ファイルシステム用のメタデータサービスが現在のホスト (アクティブなメタデータサーバー) からスタンバイホスト (潜在的なメタデータサーバー) に移動します。使用する手順は、交換するサーバーホストの状態によって異なります。

- 障害のあるアクティブなメタデータサーバーを交換するための潜在的なメタデータサーバーのアクティブ化
- 正常でアクティブなメタデータサーバーを交換するための潜在的なメタデータサーバーのアクティブ化

障害のあるアクティブなメタデータサーバーを交換するための潜在的なメタデータサーバーのアクティブ化

この手順を実行すると、機能を停止したアクティブなメタデータサーバーホスト外にメタデータサービスを移動できます。ファイルシステムがマウントされている場合でも、潜在的なメタデータサーバーはアクティブになります。次のように進めます。

注意:

障害のあるメタデータサーバーを停止、無効化、または切断するまで、潜在的なメタデータサーバーをアクティブにしないでください。

ファイルシステムがマウントされていて、アクティブなメタデータサーバーが停止しているときに、潜在的なサーバーをアクティブにするには、**-R** オプションを付けて **samsharefs** コマンドを起動する必要があります。これにより、コマンドがファイルシステムインタフェース上ではなく、**raw** デバイス上で実行されます。そのため、障害のあるサーバーがデバイスに接続されているときに、潜在的なメタデータサーバーをアクティブにすると、障害のあるサーバーによってファイルシステムが壊れる可能性があります。

1. アクティブなメタデータサーバーに障害がある場合は、何らかの操作を行う前に、メタデータデバイスにアクセスできないことを確認します。影響を受けるホストの電源をオフにして、ホストを停止するか、またはメタデータデバイスから障害のあるホストを切断します。
2. すべてのクライアントの読み取り、書き込み、および追加のリース期限が切れるように、少なくとも最大のリースタイムが経過するまで待機します。
3. 潜在的なメタデータサーバーに **root** としてログインします。

この例では、潜在的なメタデータサーバー **sharefs-mds_alt** にログインします。

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

4. 潜在的なメタデータサーバーをアクティブにします。潜在的なメタデータサーバーから、コマンド **samsharefs -R -s server file-system** を発行します。ここで **server** は潜在的なメタデータサーバーのホスト名、**file-system** は Oracle HSM 共有ファイルシステムの名前です。

この例では、潜在的なメタデータサーバーは **sharefs-mds_alt**、ファイルシステム名は **sharefs** です。

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

5. ファイルシステムの整合性を確認し、発生する可能性のある問題を修復する必要がある場合は、ここで手順「[共有ファイルシステムのアンマウント](#)」を使用してファイルシステムをアンマウントします。

6. ファイルシステムをアンマウントした場合は、ファイルシステムの確認を実行します。コマンド `samfsck -F file-system` を使用します。ここで `-F` を指定するとエラーが修復され、`file-system` はファイルシステムの名前です。

この例では、ファイルシステム名が `sharefs` であることを確認してから、修復します。

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

7. ここで停止します。

正常でアクティブなメタデータサーバーを交換するための潜在的なメタデータサーバーのアクティブ化

必要に応じて、メタデータサービスを正常でアクティブなメタデータサーバーホストから、新たにアクティブにした潜在的なメタデータサーバーに移動できます。たとえば、元のサーバーホストやそのコンポーネントの一部をアップグレードまたは交換するときに、ファイルシステムの可用性を保持するために、メタデータサービスを代替のホストに転送することがあります。次のように進めます。

1. アクティブなメタデータサーバーと潜在的なメタデータサーバーの両方に `root` としてログインします。

この例では、アクティブなメタデータサーバー `sharefs-mds` にログインします。次に、2つ目の端末ウィンドウでセキュアシェル (`ssh`) を使用して、潜在的なメタデータサーバー `sharefs-mds_alt` にログインします。

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

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

```
Password:
```

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

2. アクティブなメタデータサーバー上で Oracle HSM アーカイブファイルシステムがマウントされている場合は、アクティブなアーカイブおよびステージングジョブを終了し、新しいアクティビティを停止してから続行してください。「[アーカイブおよびステージングプロセスのアイドル状態](#)」を参照してください。

3. アクティブなメタデータサーバー上で Oracle HSM アーカイブファイルシステムがマウントされている場合は、リムーバルメディアドライブをアイドル状態にして、ライブラリ制御デーモンを停止します。「[アーカイブおよびステージングプロセスの停止](#)」を参照してください。
4. `crontab` エントリを使用してリサイクラプロセスを実行する場合は、そのエントリを削除して、リサイクラが現在実行されていないことを確認します。
5. 潜在的なメタデータサーバーをアクティブにします。潜在的なメタデータサーバーから、コマンド `samsharefs -s server file-system` を発行します。ここで `server` は潜在的なメタデータサーバーのホスト名、`file-system` は Oracle HSM 共有ファイルシステムの名前です。

この例では、潜在的なメタデータサーバーは `sharefs-mds_alt`、ファイルシステム名は `sharefs` です。

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

6. 構成ファイルをロードして、潜在的なメタデータサーバー上で Oracle HSM プロセスを起動します。コマンド `samd config` を使用します。

アーカイブ共有ファイルシステムで、`samd config` コマンドを使用すると、アーカイブプロセスおよびライブラリ制御デーモンが再起動されます。ただし、ファイルがテープからプライマリディスクキャッシュにステージングされるまで待機している共有ファイルシステムクライアントは、ステージング要求を再発行する必要があります。

7. `crontab` エントリを使用してリサイクラプロセスを実行する必要がある場合は、そのエントリを復元します。
8. ここで停止します。

非共有ファイルシステムから共有ファイルシステムへの変換

非共有ファイルシステムを共有ファイルシステムに変換するには、次のタスクを実行します。

- [アクティブおよび潜在的なメタデータサーバーでのホストファイルの作成](#)
- [非共有ファイルシステムの共有とクライアントの構成](#)

アクティブおよび潜在的なメタデータサーバーでのホストファイルの作成

各メタデータサーバー上で、共有ファイルシステムのサーバーおよびクライアントに関するネットワークアドレス情報を一覧表示する `hosts` ファイルを作成する必要があります。`hosts` ファイルは、`/etc/opt/SUNWsamfs/` ディレクトリに `mcf` ファイルとともに格納されています。共有ファイルシステムの初期作成中に、`sammkfs -S` コマンドを実行すると、このファイルに格納されている設定を使用して共有が構成されます。ここで、次の手順を使用して作成します。

1. クライアントとしてファイルシステムを共有するホストのネットワークホスト名と IP アドレスを収集します。

次の例では、`hsmfs1` ファイルシステムをクライアント `hsmfs1-mds_alt` (潜在的なメタデータサーバー)、`hsmfs1-client1`、および `hsmfs1-client2` と共有します。

2. メタデータサーバーに `root` としてログインします。

この例では、ホスト `hsmfs1-mds` にログインします。

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

3. テキストエディタを使用して、メタデータサーバー上で `/etc/opt/SUNWsamfs/hosts.family-set-name` を作成します。`family-set-name` は、共有する予定のファイルシステムのファミリーセット名で置き換えます。

この例では、`vi` テキストエディタを使用してファイル `hosts.hsmfs1` を作成します。いくつかのオプションの見出しを追加します。各行は、コメントを示すシャープ記号 (`#`) で始めます。

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

4. 1 列目にメタデータサーバーのホスト名を入力し、2 列目に対応する IP アドレスまたはドメイン名を入力します。列は空白文字で区切ります。

この例では、メタデータサーバーのホスト名および IP アドレスをそれぞれ *hsmfs1-mds*、*10.79.213.117* と入力します。

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

- 3 列目を、空白文字でネットワークアドレスと区切って追加します。この列には、サーバーの装置番号 (アクティブなメタデータサーバーの場合は 1、1 台目の潜在的なメタデータサーバーの場合は 2 など) を入力します。

この例では、メタデータサーバーは 1 つだけであるため、*1* を入力します。

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

- 4 列目を、空白文字でサーバーの順序番号と区切って追加します。この列には、*0* (ゼロ) を入力します。

4 列目の *0*、*-* (ハイフン)、または空白値は、ホストが「*on*」(共有ファイルシステムへのアクセスありで構成)であることを示します。*1* (数字の 1) は、ホストが「*off*」(ファイルシステムへのアクセスなしで構成)であることを示します (共有ファイルシステムを管理する際のこれらの値の使用については、*samhsmfs1* のマニュアルページを参照)。

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

```
#-----
hsmfs1-mds          10.79.213.117          1          0
```

7. 5列目を、空白文字で on/off ステータス列と区切って追加します。この列には、現在アクティブなメタデータサーバーを示すキーワード「`server`」を入力します。

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

8. 潜在的なメタデータサーバーとして1つ以上のホストを追加する予定である場合は、それぞれのエントリを作成します。そのたびに、サーバー番号を増分します。ただし、「`server`」キーワードは含めないでください(アクティブなメタデータサーバーは、ファイルシステムごとに1つのみです)。

この例では、ホスト `hsmfs1-mds_alt` は、サーバー番号が2の潜在的なメタデータサーバーです。メタデータサーバーとしてアクティブにしないかぎり、クライアントとなります。

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

9. クライアントホストごとに1行追加して、それぞれのサーバー番号の値を0に指定します。

サーバー番号0は、クライアントとしてのホストを示します。この例では、2つのクライアント (`hsmfs1-client1` と `hsmfs1-client2`) を追加します。

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

10. `/etc/opt/SUNWsamfs/hosts.family-set-name` ファイルを保存して、エディタを終了します。

この例では、`/etc/opt/SUNWsamfs/hosts.hsmfs1` への変更を保存して、`vi` エディタを終了します。

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

11. 共有ファイルシステムの構成に含まれる任意の潜在的なメタデータサーバー上に、新しい `/etc/opt/SUNWsamfs/hosts.family-set-name` ファイルのコピーを配置します。

この例では、ホスト `hsmfs1-mds_alt` 上にコピーを配置します。

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

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

12. ここで、[非共有ファイルシステムの共有とクライアントの構成](#)を実行します。

非共有ファイルシステムの共有とクライアントの構成

1. メタデータサーバーに *root* としてログインします。

この例では、ホスト *hsmfs1-mds* にログインします。

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

2. システムファイルおよび構成ファイルの最新のバックアップコピーがない場合は、ここでバックアップを作成します。[「Oracle HSM 構成のバックアップ」](#)を参照してください。
3. 最新のファイルシステムの回復ポイントファイルおよびアーカイブログの最新コピーがない場合も、ここで作成します。[「ファイルシステムのバックアップ」](#)を参照してください。

初期構成時にファイルシステムの自動バックアッププロセスを設定する場合は、追加のバックアップが必要ないこともあります。

4. アーカイブファイルシステムを変換する場合は、アクティブなアーカイブおよびステージングジョブを終了し、新しいアクティビティを停止してから続行してください。[「アーカイブおよびステージングプロセスのアイドル状態」](#)および[「アーカイブおよびステージングプロセスの停止」](#)を参照してください。
5. ファイルシステムをマウント解除します。コマンド *umount family-set-name* を使用します。ここで *family-set-name* は、共有する予定のファイルシステムのファミリセット名です。

Oracle HSM ファイルシステムのマウントおよびアンマウントの詳細については、*mount_samfs* のマニュアルページを参照してください。この例では、*hsmfs1* ファイルシステムをアンマウントします。

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

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

6. ファイルシステムを Oracle HSM 共有ファイルシステムに変換します。コマンド `samfscck -S -F file-system-name` を使用します。ここで `file-system-name` はファイルシステムのファミリセット名です。

この例では、`hsmfs1` というファイルシステムを変換します。

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

7. テキストエディタで `/etc/opt/SUNWsamfs/mcf` ファイルを開き、該当するファイルシステムの行を探します。

この例では、`vi` エディタを使用します。

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

8. `mcf` ファイルで、ファイルシステムエントリの最終列にある「Additional Parameters」フィールドに、`shared` パラメータを追加します。その後、ファイルを保存してエディタを閉じます。

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



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

9. テキストエディタで `/etc/vfstab` ファイルを開き、該当するファイルシステムの行を探します。

この例では、`vi` エディタを使用します。

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

10. `/etc/vfstab` ファイルで、ファイルシステムエントリの最終列にある「Mount Options」フィールドに、`shared` マウントオプションを追加します。その後、ファイルを保存してエディタを閉じます。

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

11. 共有ファイルシステムおよびホスト構成を初期化します。コマンド `samsharefs -u -R family-set-name` を使用します。ここで `family-set-name` はファイルシステムのファミリーセット名です。

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

12. *mcf* ファイルを再度読み取り、ソフトウェア自体を適宜再構成するように Oracle HSM software に指示します。

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

13. メタデータサーバー上で共有ファイルシステムをマウントします。

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

14. 複数のネットワークインタフェースを持つホストを構成する場合は、「[ローカル hosts ファイルを使用したホストネットワーク通信のルーティング](#)」を参照してください。
15. 「[追加ファイルシステムクライアントの構成](#)」で概要を示した手順を使用して、新たに共有したファイルシステムに必要なクライアントを追加します。

ローカル hosts ファイルを使用したホストネットワーク通信のルーティング

個別のホストには、ローカル hosts ファイルは必要ありません。メタデータサーバー上にあるファイルシステムのグローバルファイルは、すべてのファイルシステムホストについて、アクティブなメタデータサーバーとアクティブおよび潜在的なメタデータサーバーのネットワークインタフェースを識別します（「[アクティブおよび潜在的なメタデータサーバーでのホストファイルの作成](#)」を参照）。ただし、複数のネットワークインタフェースを持つファイルシステムホスト間で、ネットワークトラフィックを選択的にルーティングする必要がある場合は、ローカル hosts ファイルが役立ちます。

各ファイルシステムホストは、メタデータサーバー上の `/etc/opt/SUNWsamfs/hosts.family-set-name` ファイルを最初に確認することで、その他のホストのネットワークインタフェースを識別します。ここで `family-set-name` は、`/etc/opt/SUNWsamfs/mcf` ファイル内で指定されたファイルシステムのファミリー名です。次に、個別の `/etc/opt/SUNWsamfs/hosts.family-set-name.local` ファイルを確認します。ローカル hosts ファイルが存在しない場合、ホストはグローバル hosts ファイルに指定されたインタフェースアドレスをグローバルファイルに指定された順序で使用します。ただし、ローカル hosts ファイルが存在する場合、ホストはグロー

バルファイルと比較して、両方のファイルに一覧表示されたインタフェースのみをローカルファイルに指定された順序で使用します。各ファイルでさまざまなアドレスを使用すると、さまざまなホストで使用されているインタフェースを制御できません。

ローカル `hosts` ファイルを構成するには、次に概要を示す手順を使用します。

1. 「アクティブおよび潜在的なメタデータサーバーでのホストファイルの作成」で説明するように、メタデータサーバーホスト上およびそれぞれの潜在的なメタデータサーバーホスト上で、グローバル `hosts` ファイル `/etc/opt/SUNWsamfs/hosts.family-set-name` のコピーを作成します。

このセクションの例では、共有ファイルシステム `sharefs2` に、アクティブなメタデータサーバー `sharefs2-mds`、および潜在的なメタデータサーバー `sharefs2-mds_alt` (それぞれが2つのネットワークインタフェースを持つ) が含まれています。また、2つのクライアント (`sharefs2-client1` と `sharefs2-client2`) も存在します。

アクティブおよび潜在的なメタデータサーバーが、プライベートネットワークアドレスを使用して相互に通信し、DNS (Domain Name Service) でパブリック LAN (Local Area Network) 上のアドレスに解決できるホスト名を使用してクライアントと通信する必要があります。したがって、ファイルシステムのグローバル `hosts` ファイル `/etc/opt/SUNWsamfs/hosts.sharefs2` には、アクティブおよび潜在的サーバーのエントリの「*Network Interface*」フィールド内のプライベートネットワークアドレス、および各クライアントのインタフェースアドレス用のホスト名が指定されています。このファイルは、次のようになっています。

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

2. パスとファイル名 `/etc/opt/SUNWsamfs/hosts.family-set-name.local` を使用して、それぞれのアクティブおよび潜在的なメタデータサーバー上でロー

カル hosts ファイルを作成します。ここで *family-set-name* は、*/etc/opt/SUNWsamfs/mcf* ファイルで共有ファイルシステムに対して指定された名前です。アクティブおよび潜在的なサーバーで使用するネットワーク用のインタフェースのみを含めてください。

この例では、アクティブおよび潜在的なメタデータサーバーがプライベートネットワークを介して相互に通信するようにするため、各サーバー上のローカル hosts ファイル *hosts.sharefs2.local* には、アクティブおよび潜在的なサーバーのプライベートアドレスのみが一覧表示されています。

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

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

3. パスとファイル名 */etc/opt/SUNWsamfs/hosts.family-set-name.local* を使用して、それぞれのクライアント上でローカル hosts ファイルを作成します。ここで *family-set-name* は、*/etc/opt/SUNWsamfs/mcf* ファイルで共有ファイルシステムに対して指定された名前です。クライアントで使用するネットワーク用のインタフェースのみを含めてください。

この例では、クライアントはパブリックネットワーク経由でのみサーバーと通信する必要があります。したがって、アクティブおよび潜在的なメタデータサーバーのホスト名のみがファイルに含まれています。

```
[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
#
#                               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
#
#                               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. サーバーの構成が完了したときに、この手順を開始した場合は、クライアントを追加します。「[追加ファイルシステムクライアントの構成](#)」を参照してください。

共有ファイルシステムから非共有ファイルシステムへの変換

ファイルシステムの共有を解除する必要がある場合は、次のように進めます。

共有メタデータサーバーの非共有システムへの変換

1. メタデータサーバーに *root* としてログインします。

この例では、ホスト *hsmfs1-mds* にログインします。

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

2. 手順「[ファイルシステムの hosts ファイルからのホストの削除](#)」を使用して、メタデータサーバーの構成からクライアントを削除します。
3. システムファイルおよび構成ファイルの最新のバックアップコピーがない場合は、ここでバックアップを作成します。「[Oracle HSM 構成のバックアップ](#)」を参照してください。
4. 最新のファイルシステムの回復ポイントファイルおよびアーカイブログの最新コピーがない場合も、ここで作成します。「[ファイルシステムのバックアップ](#)」を参照してください。

初期構成時にファイルシステムの自動バックアッププロセスを設定する場合は、追加のバックアップが必要ないこともあります。

5. アーカイブファイルシステムを変換する場合は、アクティブなアーカイブおよびステージングジョブを終了し、新しいアクティビティを停止してから続行してください。「[アーカイブおよびステージングプロセスのアイドル状態](#)」および「[アーカイブおよびステージングプロセスの停止](#)」を参照してください。
6. ファイルシステムをマウント解除します。コマンド *umount family-set-name* を使用します。ここで *family-set-name* は、*/etc/opt/SUNWsamfs/mcf* ファイル内で共有ファイルシステムに対して指定された名前です。

Oracle HSM ファイルシステムのマウントおよびアンマウントの詳細については、*mount_samfs* のマニュアルページを参照してください。この例では、*hsmfs1* ファイルシステムをアンマウントします。

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

7. Oracle HSM 共有ファイルシステムを非共有ファイルシステムに変換します。コマンド *samfsck -F -U file-system-name* を使用します。ここで *file-system-name* は、*/etc/opt/SUNWsamfs/mcf* ファイル内で共有ファイルシステムに対して指定された名前です。

この例では、*hsmfs1* というファイルシステムを変換します。

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

8. テキストエディタで */etc/opt/SUNWsamfs/mcf* ファイルを開き、該当するファイルシステムの行を探します。

この例では、*vi* エディタを使用します。

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

9. *mcf* ファイルで、ファイルシステムエントリの最終列にある「Additional Parameters」フィールドから、*shared* パラメータを削除します。その後、ファイルを保存してエディタを閉じます。

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

10. テキストエディタで */etc/vfstab* ファイルを開き、該当するファイルシステムの行を探します。

この例では、*vi* エディタを使用します。

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

11. */etc/vfstab* ファイルで、ファイルシステムエントリの最終列にある「Mount Options」フィールドから、*shared* マウントオプションを削除します。その後、ファイルを保存してエディタを閉じます。

この例では、*vi* エディタを使用します。

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

12. ファイル */etc/opt/SUNWsamfs/hosts.file-system-name* を削除します。
13. *mcf* ファイルを再度読み取り、ソフトウェア自体を適宜再構成するように Oracle HSM software に指示します。

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


14. ファイルシステムをマウントします。

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

15. ここで停止します。

第4章 ファイルおよびディレクトリの管理

この章では、次のトピックについて説明します。

- Oracle HSM ファイル属性の設定
- 拡張ファイル属性の使用
- 大容量ファイルの格納
- Linear Tape File System (LTFS) ボリュームの操作
- SMB/CIFS 共有でのディレクトリおよびファイルの管理
- アクセス制御リストの管理。

Oracle HSM ファイル属性の設定

使い慣れたインタフェースである標準の UNIX ファイルシステムを使用してユーザーと対話できることは、Oracle Hierarchical Storage Manager and StorageTek QFS Software の重要な利点です。ほとんどのユーザーは、相違を知る必要もありません。ただし、Oracle HSM ファイルシステムは、必要に応じて、非常に多くの機能を上級ユーザーに提供できます。Oracle HSM のファイル属性を使用すると、ユーザーは個々のファイルおよびディレクトリを操作するためのファイルシステムの動作を最適化できます。そのワークロードとデータの特徴を理解するユーザーは、ファイルごとにパフォーマンスを大幅に向上させることができます。たとえば、ユーザーは、指定されたファイルまたはディレクトリ内のデータの特徴に基づいて、直接入出力またはバッファ入出力を指定できます。大容量ファイルをより連続的に書き込めるようにファイルシステム領域を事前に割り当てることができ、特定のファイルまたはディレクトリの書き込み時に使用されるストライプ幅を指定できます。

`setfa` コマンドは、新規ファイルとディレクトリおよび既存のファイルとディレクトリの両方についてこれらのファイル属性を設定します。このコマンドは、存在しない指定のファイルまたはディレクトリを作成します。ディレクトリに適用すると、指定された属性をディレクトリ内のすべてのファイルとサブディレクトリに対して設定します。あとで作成されるファイルとディレクトリはこれらの属性を継承します。

次に、基本的なタスクの概要について説明します (追加情報については、*setfa* のマニュアルページを参照してください)。

- デフォルトのファイル属性値の復元
- ファイルシステム領域の事前割り当て
- ファイルにラウンドロビン式またはストライプ化割り当てを指定
- 指定したストライプグループデバイスでのファイルストレージの割り当て。

デフォルトのファイル属性値の復元

1. ファイルシステムのホストにログインします。

```
user@solaris:~#
```

2. ファイルのデフォルトの属性値をリセットするには、コマンド *setfa -d file* を使用します。ここで *file* は、ファイルのパスと名前です。

次の例では、ファイル */samfs1/data/2014/03/series3.15* のデフォルトをリセットします。

```
user@solaris:~# setfa -d /samfs1/data/2014/03/series3.15
```

3. ディレクトリとそのすべての内容のデフォルトの属性値を再帰的にリセットするには、コマンド *setfa -r directory* を使用します。ここで *directory* は、ディレクトリのパスと名前です。

次の例では、サブディレクトリ */samfs1/data/2014/02* のデフォルトをリセットします。

```
user@solaris:~# setfa -r /samfs1/data/2014/02/
```

4. ここで停止します。

ファイルシステム領域の事前割り当て

ファイルの容量を事前に割り当てると、ファイルの書き込み時にファイル全体を連続して書き込むための十分な領域が確保されます。大容量ファイルの書き込みおよび読み取りを連続するブロックで行うと、より小さくて散在したデータのブロックのシークとバッファリングに関連するオーバーヘッドが減ることで、効率と全体的

なパフォーマンスが向上します。そのため、事前割り当ては、予測可能な数の大きいデータのブロックを書き込む場合に最適です。事前に割り当てられた未使用の容量は、ファイルを閉じたときにファイルの一部であり続け、ファイル全体を削除するまでほかの使用には解放できません。

1. ファイルシステムのホストにログインします。

```
user@solaris:~#
```

2. すでにデータを含んでいる既存のファイルへの書き込みのために領域を事前に割り当てる必要がある場合、コマンド `setfa -L number-bytes file` を使用します。ここで `number-bytes` は、整数、または整数に `k` (K バイト)、`m` (M バイト)、または `g` (G バイト) を付けたもので、`file` はファイルの名前です。

コマンド `setfa -L` では、標準の割り当てが使用され、ストライプ化がサポートされています。事前に割り当てられたファイルは、事前に割り当てられたサイズよりも大きくなる可能性があります。次の例では、既存のファイル `tests/series119b` のために 121M バイトを事前に割り当てます。

```
user@solaris:~# setfa -L 121m tests/series119b
```

3. ストレージブロックの割り当てのない新しいファイルを書き込むための領域を事前に割り当てる必要がある場合は、コマンド `setfa -l number-bytes file` を使用します。ここでは:
 - `l` は小文字の「L」です。
 - `number-bytes` は整数、または整数にキロバイトを表す `k`、メガバイトを表す `m`、またはギガバイトを表す `g` を付けたものです。
 - `file` はファイルの名前です。

コマンド `setfa -l` は、指定されたバイト数を事前に割り当てます。結果のファイルは、事前に割り当てられたサイズに固定されており、事前に割り当てられたサイズより大きくなることも小さくなることもできません。次の例では、ファイル `data/2014/a3168445` を作成し、その内容のために 2G バイトの容量を事前に割り当てます。

```
user@solaris:~# setfa -l 2g data/2014/a3168445
```

4. ここで停止します。

ファイルにラウンドロビン式またはストライプ化割り当てを指定

デフォルトでは、Oracle HSM ファイルシステムは、マウント時にファイルシステムに指定された割り当て方式を使用します。ただし、ユーザーは、指定したディレクトリまたはファイルに指定したストライプ幅を使用して、優先される割り当て方式としてラウンドロビン式またはストライプ化を指定できます。

1. ファイルシステムのホストにログインします。

```
user@solaris:~#
```

2. ラウンドロビン式割り当てを指定するには、ストライプ幅 0 (ゼロ) を指定します。コマンド `setfa -s 0 directory-or-file` を使用します。ここで `directory-or-file` は、指定した割り当て方式を使用して書き込むディレクトリまたはファイルの名前です。

ストライプ幅 0 (ゼロ) は、ストライプ化されていないラウンドロビン式割り当てを指定します。ファイルシステムは、次に使用可能なデバイスでファイルの書き込みを開始します。ファイルが完成するかデバイスの容量がなくなるまで、連続したディスク割り当て単位 (DAU) を同じデバイス上のファイルに書き込みます。デバイスで容量がなくなった場合、ファイルシステムは次に使用可能なデバイスに移動し、ディスク割り当て単位の書き込みを続行します。ファイルが完成するまでプロセスは繰り返されます。次の例では、`data/field-reports` ディレクトリに書き込まれるすべてのファイルにラウンドロビン式割り当てを指定します。

```
user@solaris:~# setfa -s 0 data/field-reports
```

3. ストライプ化割り当てを指定するには、ストライプ幅を指定します。コマンド `setfa -s stripe-width directory-or-file` を使用します。ここで `stripe-width` は、範囲 [1-255] 内の整数で、`directory-or-file` は、指定した割り当て方式を使用して書き込むディレクトリまたはファイルの名前です。

範囲 [1-255] 内のストライプ幅は、ストライプ化割り当てを指定します。ファイルが完成するまで、ファイルシステムは、ストライプ幅で指定された数のディスク割り当て単位 (DAU) を複数のデバイスに並行して書き込みます。次の例で

は、ディレクトリに書き込まれるすべてのファイルにストライプ幅 1 を使用してストライプ化割り当てを指定するため、すべてのファイルのファイル割り当てが `data/field-reports` ディレクトリの `data/2014/` に書き込まれ、ファイルシステムは、そのファイルが完成するまで、使用可能なそれぞれのデバイスに 1 つのディスク割り当て単位を書き込みます。

```
user@solaris:~# setfa -s 1 data/2014/
```

4. ここで停止します。

指定したストライプグループデバイスでのファイルストレージの割り当て

ユーザーは、ラウンドロビン式またはストライプ化割り当てを開始すべきストライプグループデバイスを指定できます。Oracle HSM ストライプグループは、複数の物理ボリュームにわたりデータをストライプ化する論理ボリュームです。ラウンドロビン式ファイル割り当てが有効になっているときは、ファイル全体が指定されたストライプグループに書き込まれます。ストライプ化割り当てが有効になっているときは、指定されたストライプグループで最初の割り当てが行われます。

1. ファイルシステムのホストにログインします。

```
user@solaris:~#
```

2. ファイル全体を特定のストライプグループに書き込むには、ラウンドロビン式割り当てを使用します。コマンド `setfa -s 0 -gstripe-group-number` を使用します。ここで `stripe-group-number` は、指定したストライプグループを識別する範囲 `[0-127]` 内の整数です。

この例では、ファイル `reports/site51` を書き込むときに、ストライプグループ 0 で始まるラウンドロビン式割り当てを指定します。

```
user@solaris:~# setfa -s 0 -g0 reports/site51
```

3. 指定したストライプグループから始まる多数のストライプグループにわたりファイルをストライプ化するには、ストライプ化割り当てを使用します。コマンド `setfa -s stripe-width -gstripe-group-number` を使用します。ここで `stripe-width` は、ディスク割り当て単位の数を指定する範囲 `[1-255]` 内の整

数で、*stripe-group-number* は、指定したストライプグループを識別する範囲 [0-127] 内の整数です。

この例では、ファイル *assessments/site52* のストライプ化割り当てを指定します。グループごとに、ストライプグループ 21 で始まる 3 つのディスク割り当て単位を指定します。

```
user@solaris:~# setfa -s 3 -g21 assessments/site52
```

4. ここで停止します。

拡張ファイル属性の使用

その他の Solaris および Linux ファイルシステムと同様に、Oracle HSM ファイルシステムでは拡張ファイル属性がサポートされます。拡張属性は、ファイルシステム自体によってではなくユーザーまたはアプリケーションによってファイルに関連付けられる、任意のメタデータを保持します。拡張属性は、ファイルのダイジェスト、作成者とソースアプリケーションの名前、およびテキストファイルで使用される文字エンコーディングを保持するために使用されてきました。

リリース 6.1 以降の Oracle HSM では、464 文字以下を含む小さい拡張属性ファイルが、データパーティション内のブロックを使用する代わりにメタデータパーティション内の拡張 i ノードに格納されます。拡張属性が使用され、ファイルシステムのメタデータがより高速なデバイス (フラッシュストレージなど) に格納される場合に、この新しいアプローチによってファイルシステムのパフォーマンスが大幅に向上します。

拡張ファイル属性は、新しいファイルシステムを作成するか、または古いファイルシステムを回復ポイント (*samfsdump*) ファイルから復元するたびに、自動的に有効になります。拡張属性の使用に関する詳細は、Solaris *fsattr(5)* および Linux *xattr(7)* のマニュアルページを参照してください。

大容量ファイルの格納

Oracle HSM ファイルシステムは、非常に大きいファイルの操作に特に適しています。このセクションでは、次のトピックを扱います。

- [非常に大きいファイルでのディスクキャッシュの管理](#)
- [ファイルのセグメント化](#)

- 大規模なデータセットにリムーバブルメディアファイルを使用

非常に大きいファイルでのディスクキャッシュの管理

非常に大きなファイルを操作するときは、使用可能なディスクキャッシュのサイズに特に注意してください。ディスクキャッシュより大きいファイルを書き込む場合、非アーカイブファイルシステムは *ENOSPC* エラーを返すのに対し、アーカイブファイルシステムは、使用可能になることはない領域を待機するだけであるため、アプリケーションがブロックします。

Oracle HSM では、ディスクキャッシュのサイズを増やすための2つの代替方法が用意されています。

- ユーザーが、任意の時点で大きいファイルの一部のみをディスクにステージングできるように、ファイルをセグメント化します
- ユーザーがデータをディスクにステージングしないように、大規模なデータセットにリムーバブルメディアファイルを使用します。

ファイルのセグメント化

ファイルに対して Oracle HSM セグメント化属性を設定すると、ファイルシステムは、指定されたサイズのセグメントにファイルを分割し、いつでも現在必要なセグメントのみがディスクに存在するようにアクセス要求を管理します。ファイルの残りはリムーバブルメディアにあります。

大容量ファイルのセグメント化には多数の利点があります。

- ユーザーは、使用可能なディスクキャッシュより大きいファイルを作成してアクセスできます。

いつでもセグメントのみがキャッシュ内に存在するため、ディスクキャッシュに納まるセグメントサイズを選択するだけで済みます。完成したファイルは、メディアに格納できる任意のサイズまで大きくなることができます。

- ユーザーは、ディスクキャッシュから解放された大容量ファイルにより迅速にアクセスできます。大容量ファイルの一部をディスクにステージングする作業は、ファイル全体がステージングされるのを待機するよりはるかに高速です。
- ファイルがセグメント化されていると、各ファイルの変更された部分のみが再アーカイブされるため、アーカイブ処理の速度と効率が向上する可能性があります。

- 複数のドライブにマウントされたリムーバブルメディアボリュームにわたりファイルをストライプ化できます。その後、アーカイブ操作とステージング操作を並行して実行できるため、パフォーマンスが向上します。

次の2つの制限があります。

- 共有ファイルシステムではファイルをセグメント化できません。
- Solaris のメモリーマッピング関数である `mmap()` は、セグメント化ファイル内のバイトをプロセスのアドレススペースにマップできないため、バイナリ実行可能ファイルはセグメントできません。

セグメント化ファイルを作成するには、次のように進めます。

ファイルのセグメント化

1. ファイルシステムのホストにログインします。

```
user@solaris:~#
```

2. セグメント化する必要があるファイルを選択するか、必要に応じて作成します。
3. 単一のファイルをセグメント化するには、コマンド `segment [-s stage_ahead] -l segment_size file-path-name` を使用します。ここでは:
 - `stage_ahead` (オプション) は、特定のセグメントへのアクセス時に読み取る連続した追加のセグメントの数を指定する整数です。適切な値を選択すると、システムページキャッシュの使用率が向上する可能性があるため、入出力のパフォーマンスが改善します。デフォルトは `0` (無効) です。
 - `segment_size` は、各セグメントのサイズをまとめて指定する整数と単位です。サポートされる単位は `k` (K バイト)、`m` (M バイト)、および `g` (G バイト) です。最小サイズは 1M バイト (`1m` または `1024k`) です。
 - `file-path-name` は、ファイルのパスとファイル名です。

詳細は、`segment` のマニュアルページを参照してください。次の例では、1.5M バイト (`1536k`) のセグメントサイズを使用してファイル `201401.dat` をセグメント化します。

```
user@solaris:~# segment -l 1536k 201401.dat
```

- ディレクトリとそのすべてのサブディレクトリ内のファイルを再帰的にセグメント化するには、コマンド `segment [-s stage_ahead] -l segment_size -r directory-path-name` を使用します。ここで `directory-path-name` は、開始ディレクトリのパスと名前です。

次の例では、1M バイト (`1m`) のセグメントサイズを使用して `/hsm/hsmfs1/data` ディレクトリとそのサブディレクトリ内のファイルをすべてセグメント化します。

```
user@solaris:~# segment -l 1m -r /hsm/hsmfs1/data
```

- ここで停止します。

複数のボリュームにわたるセグメント化ファイルのストライプ化

複数のドライブを指定するアーカイブセットにセグメント化ファイルを割り当てることで、セグメント化ファイルをストライプ化入出力用に構成します。次のように進めます。

- ホストに `root` としてログインします。

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

- ファイル `/etc/opt/SUNWsamfs/archiver.cmd` をテキストエディタで開きます。

次の例では、`vi` エディタを使用してファイルを開きます。

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

- 複数のドライブにわたりセグメント化ファイルをストライプ化するには、セグメント化ファイルが含まれている各アーカイブセットのコピーごとに少なくとも2つのドライブを使用することを指定します。`archiver.cmd` ファイルで、`params` セクションを見つけます。各コピーのパラメータに `-drives number` パラメータが含まれていることを確認します。ここで `number` は2以上です。必要な変更を行なって、ファイルを保存してエディタを閉じます。

次の例では、*archiver.cmd* ファイルは、構成されているすべてのアーカイブセットの3つすべてのコピーに2つのドライブを指定します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
# Configuration file for Oracle HSM archiving file systems ...
...
#-----
# Copy Parameters
params
allsets -sort path -offline_copy stageahead -reserve set
allsets.1 -startage 10m -drives 2
allsets.2 -startage 24h -drives 2
allsets.3 -startage 48h -drives 2
endparams
...
:wq
root@solaris:~#
```

4. *archiver.cmd* ファイルでエラーを調べます。コマンド *archiver -lv* を使用します。

archiver -lv コマンドは、*archiver.cmd* ファイルを画面に出力し、エラーが見つからない場合は構成レポートを生成します。それ以外の場合、エラーを記録して停止します。

```
root@solaris:~# archiver -lv
Reading '/etc/opt/SUNWsamfs/archiver.cmd'.
...
Total space available: 300T
root@solaris:~#
```

5. *archiver.cmd* ファイルを再度読み取り、それに従って Oracle HSM ソフトウェア自体を再構成するようにこのソフトウェアに指示します。*/opt/SUNWsamfs/sbin/samd config* コマンドを使用します。

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

6. ここで停止します。

大規模なデータセットにリムーバブルメディアファイルを使用

Oracle HSM のリムーバブルメディアファイル全体がリムーバブルメディア上にあるため、ファイルシステムディスクキャッシュのスペースを占有することはありません。ファイルシステムは、リムーバブルメディアファイルをメモリーに直接読み取ります。そのため、ストレージメディアでは、ファイルのサイズの制限はまったくありません。単一のメディアカートリッジの容量を超えるリムーバブルファイルは、複数カートリッジのボリュームオーバーフローファイルになる可能性があります。ファイルシステムは、連続してデータを読み取ってメディアに書き込みます。

あらゆる点で、リムーバブルメディアファイルは一般的な UNIX ファイルに似ています。権限、ユーザー名、グループ名、およびファイルサイズがあります。ユーザーまたはアプリケーションがリムーバブルメディアファイルを要求すると、システムは対応するボリュームを自動的にマウントして、データがディスク上にある場合と同じようにユーザーはメモリーからデータにアクセスします。ただし、リムーバブルメディアファイルは、2つの主な点でほかの Oracle HSM ファイルとは異なります。リムーバブルメディアファイルは、Oracle Hierarchical Storage Manager software によってアーカイブされることはなく、NFS ではサポートされません。

Oracle Hierarchical Storage Manager software は、リムーバブルメディアファイルを管理しません。ファイルはアーカイブされることも解放されることもなく、ファイルが含まれているメディアはリサイクルされません。このため、アーカイブ処理以外の目的でリムーバブルメディアを使用する必要があるときに、リムーバブルメディアファイルが役に立ちます。これらのファイルは、Oracle HSM 構成ファイルとメタデータダンプファイルをバックアップするリムーバブル障害回復ボリュームを作成するために理想的です。また、ボリュームを読み取り専用でロードして、ファイルをリムーバブルメディアファイルとしてメモリーに読み込むことによって、外部ボリューム (ほかのアプリケーションによって作成されたボリューム) からデータを読み取ることもできます。

リムーバブルメディアファイルは解放できず、関連付けられたボリュームはリサイクルできないため、通常リムーバブルメディアファイルはアーカイブコピーと混在させるのではなく、専用のボリュームに隔離するべきです。

リムーバブルメディアファイルまたはボリュームオーバーフローファイルの作成

1. ファイルシステムのホストにログインします。

```
user@solaris:~#
```

2. リムーバブルメディアファイルの Oracle HSM ファイルシステム、パス、およびファイル名を選択します。

リムーバブルメディアファイルを作成したら、ファイルシステムが、リムーバブルメディアのデータを使用してこのパスとファイル名の要求に対応します。

3. リムーバブルメディアファイルを作成します。コマンド `request -m media-type -v volume-specifier data-file` を使用します。ここで `mediatype` は、[付録 A「装置タイプの用語集」](#)に一覧表示されている 2 文字のメディアタイプコードの 1 つ、`data-file` は、リムーバブルメディアファイル用に選択したパスと名前、`volume-specifier` は、次のいずれかです。
 - ボリュームシリアル番号またはボリュームシリアル番号のスラッシュ区切りのリスト

最初の例では、LTO (*li*) ボリューム `VOL080` で `file1` を作成します。

```
user@solaris:~# request -m li -v VOL080 /hsm/hsmfs1/data/file1
```

2 番目の例では、LTO (*li*) ボリューム `VOL080`、`VOL082`、および `VOL098` で `file2` を作成します。

```
user@solaris:~# request -m li -v VOL081/VOL082/VOL098 /hsm/hsmfs1/data/file2
```

- `-l volume-list-file`。ここで `volume-list-file` は、各行に単一のボリュームシリアル番号と、オプションで、スペースと、指定したボリュームでの開始位置を指定する 10 進数または 16 進数 (16 進数の前に `0x` を付けます) が一覧表示されているファイルのパスと名前です。

次の例では、`vi` エディタを使用して、ファイル `vsnsfile3` に一覧表示されている LTO (*li*) ボリュームで `file3` を作成します。

```
user@solaris:~# vi vsnsfile3
VOL180
VOL181
VOL182
:wq
user@solaris:~# request -m li -v -l vsnsfile3 /hsm/hsmfs1/data/file3
```

4. ここで停止します。

リムーバブルメディアファイルとしての外部テープボリュームの読み取り

1. ファイルシステムのホストにログインします。

```
user@solaris:~#
```

2. 外部テープがバーコード付きであること、書き込み保護されていること、読み取り専用で開いていること、および θ に位置付けられていることを確認します。
3. リムーバブルメディアファイルの Oracle HSM ファイルシステム、パス、およびファイル名を選択します。

リムーバブルメディアファイルを作成したら、ファイルシステムが外部テープのデータを使用して、このパスとファイル名の要求に対応します。

4. `-N` (外部メディア) オプションを使用してリムーバブルメディアファイルを作成します。コマンド `request -m media-type -N -v volume-serial-number data-file` を使用します。ここでは:
 - `mediatype` は、[付録A「装置タイプの用語集」](#)に一覧表示されている2文字のメディアタイプコードの1つです。
 - `volume-serial-number` は、外部テープのボリュームシリアル番号です。
 - `data-file` は、リムーバブルメディアファイルのパスと名前です。

次の例では、外部 LTO (*li*) ボリューム *FOR991* のリムーバブルメディアファイルを作成します。

```
user@solaris:~# request -m li -N -v FOR991 /hsm/hsmfs1/foreignfile
```

5. ここで停止します。

Linear Tape File System (LTFS) ボリュームの操作

Linear Tape File System は、順次アクセステープメディア上のデータをファイルシステムに編成する自己記述型のテープ形式であり、ランダムアクセスディスク上に存在する場合と同じように、ファイルにアクセスできます。Oracle HSM は LTFS の拡張サポートを提供します。このソフトウェアでは、Oracle HSM ファイルシステムで LTFS ファイルを使用でき、LTFS メディアの作成、アクセス、および管理を行うためのツールが提供されます。

このセクションでは、次のトピックについて説明します。

- [ライブラリへの LTFS メディアのインポート](#)
- [LTFS ディレクトリおよびファイルの Oracle HSM ファイルシステムへの接続](#)
- [Oracle HSM ソフトウェアを使用した LTFS メディアへのアクセス](#)
- [Oracle HSM ソフトウェアを使用した LTFS メディアの管理](#)

ライブラリへの LTFS メディアのインポート

Oracle HSM ソフトウェアは、LTFS メディアを自動的に認識します。そのため、ほかのメディアの場合と同様に、`samimport` コマンドを使用して LTFS ボリュームをインポートできます。追加情報については、「[リムーバブルメディアのインポートおよびエクスポート](#)」および `samimport` のマニュアルページを参照してください。

LTFS ディレクトリおよびファイルの Oracle HSM ファイルシステムへの接続

Oracle HSM software は、Linear Tape File System (LTFS) ディレクトリおよびファイルを Oracle HSM ファイルシステムに接続して、Oracle HSM ファイルの場合と同じようにアクセスおよび管理できます。ソフトウェアは、LTFS メタデータを LTFS ボリュームから Oracle HSM ファイルシステム内の空のディレクトリにコピーします。このメタデータを使用して、Oracle HSM は、アーカイブされた Oracle HSM ファイルと同じように LTFS メディアとファイルを管理します。ユーザーが一度にすべての LTFS ファイルにアクセスしたときか、LTFS メタデータが適用されるとすぐに、LTFS ファイルは使用のために LTFS メディアから Oracle HSM ディスクキャッシュにステージングされます。Oracle HSM ファイルシステムのアーカイブ処理および領域管理ポリシーは、Oracle HSM ファイルの場合と同じように適用されます。

このセクションでは、次のタスクについて説明します。

- 要求時の LTFS ファイルのアクセス可能化
- ディスクキャッシュで LTFS ファイルを即時にアクセス可能化

要求時の LTFS ファイルのアクセス可能化

LTFS ファイルを Oracle HSM ファイルシステムに接続すると、Oracle HSM software は、LTFS ボリュームから Oracle HSM ファイルシステム内の指定されたディレクトリにファイルシステムメタデータをコピーします。その後、ファイルは、ユーザーによってアクセスされるとディスクキャッシュにステージングされます。LTFS ファイルを接続するには、次のように進めます。

1. ファイルシステムのホストにログインします。

```
user@solaris:~#
```

2. LTFS ファイルをホストする Oracle HSM ファイルシステムで、LTFS メタデータを格納するディレクトリを作成します。

次の例では、ファイルシステムマウントポイント `/hsm/hsmfs1` の下にディレクトリ `ltfs1/` を作成します。

```
user@solaris:~# mkdir /hsm/hsmfs1/ltfs1
user@solaris:~#
```

3. LTFS ファイルを Oracle HSM ファイルシステムに接続します。コマンド `samltfs attach LTFS-media-type.LTFS-volume-serial-number SAMQFS-directory` を使用します。ここでは:

- `LTFS-media-type` は、LTFS データを保持するメディアタイプの 2 文字のメディアタイプコードです (付録A「装置タイプの用語集」を参照)。
- `LTFS-volume-serial-number` は、LTFS ボリュームの 6 文字の英数字のボリュームシリアル番号です。
- 指定するメディアタイプとボリュームシリアル番号は、カタログが LTFS ボリュームとして一覧表示するボリュームを識別します。

Oracle HSM カタログでは、LTFS メディアはラベル付きではなく、`non-SAM` および `tfs` とマークされています。

- *SAMQFS-directory* は、LTFS メタデータを保持するディレクトリのパスと名前です。

次の例では、LTO (*Li*) ボリューム *TFS233* を接続します。

```
user@solaris:~# samltfs attach li.TFS233 /hsm/hsmfs1/ltfs1
user@solaris:~#
```

4. ここで停止します。

ディスクキャッシュで LTFS ファイルを即時にアクセス可能化

LTFS ファイルを Oracle HSM ファイルシステムに取り込むと、Oracle HSM software は、LTFS ボリュームから Oracle HSM ファイルシステム内の指定されたディレクトリにファイルシステムメタデータをコピーし、すべてのファイルをすぐにディスクキャッシュにステージングします。LTFS ファイルを取り込むには、次のように進めます。

1. ファイルシステムのホストにログインします。

```
user@solaris:~#
```

2. LTFS ファイルをホストする Oracle HSM ファイルシステムで、LTFS メタデータを格納するディレクトリを作成します。

次の例では、ファイルシステムマウントポイント */hsm/hsmfs1* の下にディレクトリ *ltfs2/* を作成します。

```
user@solaris:~# mkdir /hsm/hsmfs1/ltfs2
user@solaris:~#
```

3. LTFS ファイルを Oracle HSM ファイルシステムに取り込みます。コマンド *samltfs ingest LTFS-media-type.LTFS-volume-serial-number SAMQFS-directory* を使用します。ここでは:

- *LTFS-media-type* は、LTFS データを保持するメディアタイプの 2 文字のメディアタイプコードです (付録A「装置タイプの用語集」を参照)。
- *LTFS-volume-serial-number* は、LTFS ボリュームの 6 文字の英数字のボリュームシリアル番号です。

- 指定するメディアタイプとボリュームシリアル番号は、カタログが LTFS ボリュームとして一覧表示するボリュームを識別します。

Oracle HSM カタログでは、LTFS メディアはラベル付きではなく、*non-SAM* および *tfs* とマークされています。

- *SAMQFS-directory* は、LTFS メタデータを保持するディレクトリのパスと名前です。

次の例では、LTO (*Li*) ボリューム *TFS234* を取り込みます。

```
user@solaris:~# samltfs ingest li.TFS234 /hsm/hsmfs1/ltfs2
user@solaris:~#
```

4. ここで停止します。

Oracle HSM ソフトウェアを使用した LTFS メディアへのアクセス

Oracle HSM ソフトウェアは、LTFS メディアのロードとアンロード、および Oracle HSM *defaultfs.conf* ファイルで指定された LTFS マウントポイントを使用したホストでの LTFS ファイルシステムのマウントまたはマウント解除を行うこともできます。

- テープドライブへの LTFS ボリュームのロードおよび LTFS ファイルシステムのマウント
- LTFS ファイルシステムのアンマウントおよびテープドライブからのボリュームのアンロード
- LTFS 構成およびステータス情報の表示。

テープドライブへの LTFS ボリュームのロードおよび LTFS ファイルシステムのマウント

1. ファイルシステムのホストにログインします。

```
user@solaris:~#
```

2. LTFS ボリュームをテープドライブにロードして、*defaultfs.conf* ファイルで指定されたマウントポイントにファイルシステムをマウントします。コマンド

`samltps load LTFS-media-type.LTFS-volume-serial-number` を使用します。
ここでは:

- `LTFS-media-type` は、LTFS データを保持するメディアタイプの 2 文字のメディアタイプコードです (付録A「装置タイプの用語集」を参照)。
- `LTFS-volume-serial-number` は、LTFS ボリュームの 6 文字の英数字のボリュームシリアル番号です。
- 指定するメディアタイプとボリュームシリアル番号は、カタログが LTFS ボリュームとして一覧表示するボリュームを識別します。

Oracle HSM カタログでは、LTFS メディアはラベル付きではなく、*non-SAM* および *tfs* とマークされています。

次の例では、LTO (*li*) ボリューム *TFS434* をロードして、`defaults.conf` ファイル `/mnt/lufs` で指定されたディレクトリにマウントします。

```
user@solaris:~# samltps load li.TFS234
```

3. ここで停止します。

LTFS ファイルシステムのアンマウントおよびテープドライブからのボリュームのアンロード

1. ファイルシステムのホストにログインします。

```
user@solaris:~#
```

2. LTFS ファイルシステムをアンマウントして、対応するボリュームをテープドライブからアンロードします。コマンド `samltps unload LTFS-media-type.LTFS-volume-serial-number` を使用します。ここでは:
 - `LTFS-media-type` は、LTFS データを保持するメディアタイプの 2 文字のメディアタイプコードです (付録A「装置タイプの用語集」を参照)。
 - `LTFS-volume-serial-number` は、LTFS ボリュームの 6 文字の英数字のボリュームシリアル番号です。
 - 指定するメディアタイプとボリュームシリアル番号は、カタログが LTFS ボリュームとして一覧表示する LTFS ボリュームを識別します。

Oracle HSM カタログでは、LTFS メディアはラベル付きではなく、*non-SAM* および *tfs* とマークされています。

次の例では、LTFS ファイルシステムをアンマウントして、LTO (*Li*) ボリューム *TFS435* をアンロードします。

```
user@solaris:~# samltfs unload li.TFS435
```

3. ここで停止します。

Oracle HSM ソフトウェアを使用した LTFS メディアの管理

Oracle HSM ソフトウェアは、LTFS メディアの作成、消去、および検証を行うために必要な基本的なツールを提供します。

- ボリュームを LTFS ファイルシステムとしてフォーマット
- LTFS データの消去、およびボリュームからの LTFS フォーマットとパーティションの削除
- LTFS ファイルシステムの整合性の検査

ボリュームを LTFS ファイルシステムとしてフォーマット

1. ファイルシステムのホストにログインします。

```
user@solaris:~#
```

2. LTFS ファイルシステムのリムーバブルメディアボリュームをパーティションに分割してフォーマットします。コマンド `samltfs mkltfs media-type.volume-serial-number` を使用します。ここでは:

- *media-type* は、LTFS 互換タイプのメディアの 2 文字のメディアタイプコードです (付録A「装置タイプの用語集」を参照)。
- *volume-serial-number* は、ボリュームの 6 文字の英数字のボリュームシリアル番号です。

次の例では、LTO (*Li*) ボリューム *VOL234* をパーティションに分割して、LTFS ボリュームとしてフォーマットします。

```
user@solaris:~# samltfs mkltfs li.VOL234
```

3. ここで停止します。

LTFS データの消去、およびボリュームからの LTFS フォーマットとパーティションの削除

1. ファイルシステムのホストにログインします。

```
user@solaris:~#
```

2. LTFS ボリュームを消去して、一般的な使用のために復元します。コマンド `samltps unltfs media-type.volume-serial-number` を使用します。ここでは:
 - `media-type` は、LTFS 互換タイプのメディアの 2 文字のメディアタイプコードです (付録A「装置タイプの用語集」を参照)。
 - `volume-serial-number` は、ボリュームの 6 文字の英数字のボリュームシリアル番号です。

次の例では、LTFS ファイルシステムデータとメタデータを消去して、LTO (*Li*) ボリューム *VOL234* でパーティションを削除します。

```
user@solaris:~# samltfs unltfs li.VOL234
```

3. ここで停止します。

LTFS ファイルシステムの整合性の検査

1. ファイルシステムのホストにログインします。

```
user@solaris:~#
```

2. LTFS ファイルシステムの整合性を検査します。コマンド `samltps ltfscck LTFS-media-type.LTFS-volume-serial-number` を使用します。ここでは:
 - `LTFS-media-type` は、LTFS データを保持するメディアタイプの 2 文字のメディアタイプコードです (付録A「装置タイプの用語集」を参照)。
 - `LTFS-volume-serial-number` は、LTFS ボリュームの 6 文字の英数字のボリュームシリアル番号です。

- 指定するメディアタイプとボリュームシリアル番号は、カタログが LTFS ボリュームとして一覧表示する LTFS ボリュームを識別します。

Oracle HSM カタログでは、LTFS メディアはラベル付きではなく、*non-SAM* および *tfs* とマークされています。

次の例では、LTO (*li*) ボリューム *VOL234* 上で LTFS ファイルシステムを検査します。

```
user@solaris:~# samltfs ltfscck li.VOL234
```

3. ここで停止します。

LTFS 構成およびステータス情報の表示

LTFS の構成およびステータスを表示するには、コマンド `samltfs status` を使用します。

```
user@solaris:~# samltfs status
```

SMB/CIFS 共有でのディレクトリおよびファイルの管理

このセクションでは、次のトピックについて説明します。

- [SMB/CIFS 共有でのシステム属性の管理](#)
- [アクセス制御リストの管理](#)。

SMB/CIFS 共有でのシステム属性の管理

システム属性は、Microsoft Windows ファイルシステムが解釈できる UNIX 以外のメタデータに Oracle HSM ファイルを関連付けることによって、SMB/CIFS ファイル共有をサポートします。このセクションでは、まず Oracle HSM によってサポートされるシステム属性の概要について説明します。その後、次のタスクの基本的な手順について説明します。

- [システム属性の表示](#)
- [システム属性の変更](#)。

Oracle HSM でサポートされるシステム属性

システム属性は、属性 *name* (値は *true*) または名前の否定 *noname* (値は *false*) によって表されるブール (*true* または *false*) 値です。Oracle HSM は、SMB/CIFS ファイル共有のサポートのために次のシステム属性を提供します。

- *appendonly* は、ユーザーがファイルへのデータの追加のみを行えることを意味します。*noappendonly* は、この制限は無効であることを意味します。
- *archive* は、ファイルが最後にコピーまたはバックアップされてから変更されたことを意味します。*noarchive* は、ファイルが最後にコピーまたはバックアップされてから変更されていないことを意味します。Oracle HSM では現在この属性は使用されません。
- *hidden* は、ファイルがデフォルトでファイルリストに表示されないことを意味します。*nohidden* は、ファイルがデフォルトで表示されることを意味します。
- *immutable* は、ディレクトリまたはファイルとその内容を変更または削除できないことを意味します。*noimmutable* は、ディレクトリまたはファイルを変更または削除できることを意味します。
- *nodump* は、ファイルをバックアップできないことを意味します。*nonodump* は、ファイルをバックアップできることを意味します。Oracle Solaris ではこの属性は使用されません。
- *nounlink* は、ファイルまたはディレクトリとその内容を削除または名前変更できないことを意味します。*nonounlink* は、ファイルまたはディレクトリとその内容を削除または名前変更できることを意味します。
- *offline* は、ファイルが Oracle HSM ファイルシステムから解放されていることを意味します。Microsoft Windows システムではファイルはプレビューされません。*nooffline* は、ファイルがオンラインで、Oracle HSM ファイルシステムから解放されていないことを意味します。
- *readonly* は、ファイルを削除または変更できないことを意味します。*noreadonly* は、ファイルを削除または変更できることを意味します。属性は、ディレクトリへの適用時には無視されます。
- *sparse* は、保管されているファイルにゼロ以外のデータのみが含まれていて、ファイルへのアクセス時、またはスパースファイルをサポートしないファイルシステムへのファイルのコピー時にファイルシステムによって復元された範囲にゼロが減らされていることを意味します。*nosparse* は、ファイルがスパースではないことを意味します。

- *system* は、ファイルは Microsoft Windows オペレーティングシステムに不可欠であり、決して変更および削除することはできず、デフォルトでリストに表示すべきではないことを意味します。*nosystem* は、ファイルがシステムファイルではないことを意味します。

システム属性の表示

Oracle HSM ファイルのシステム属性を表示するには、Solaris コマンド `ls -lv file` を使用します。ここで *file* は、ファイルのパスと名前です。

次の例では、ファイル `/hsm/hsmfs1/documents/master-plan.odt` のシステム属性を一覧表示します。

```
user@solaris:~# ls -lv /hsm/hsmfs1/documents/master-plan.odt
-rw-r--r--  1 root root  40560 Mar 4 15:52 /hsm/hsmfs1/documents/master-plan.odt
{archive,nohidden,noreadonly,nosystem,noappendonly,nonodump,noimmutable,nonounlink,
 nooffline,nosparsed}
user@solaris:~#
```

システム属性の変更

ファイルのシステム属性値を指定の値に変更するには、Solaris コマンド `chmod S+v{attributes}` を使用します。ここで *attributes* は、Oracle HSM でサポートされるシステム属性のコンマ区切りリストです。

構文と使用可能なオプションの包括的な説明については、*chmod* のマニュアルページを参照してください。次の例では、アーカイブ属性を *noarchive* (false) から *archive* (true) に変更します。

```
root@solaris:~# ls -lv /hsm/hsmfs1/documents/master-plan.odt
-r-xr-xr-x  1 root root  40561 Mar 4 15:52 /hsm/hsmfs1/documents/master-plan.odt
{noarchive,nohidden,readonly,nosystem,noappendonly,nonodump,noimmutable,
 nonounlink,offline,nosparsed}
root@solaris:~# chmod S+v{archive} /hsm/hsmfs1/documents/master-plan.odt
root@solaris:~# ls -lv /hsm/hsmfs1/documents/master-plan.odt
-r-xr-xr-x  1 root root  40561 Mar 4 15:52 /hsm/hsmfs1/documents/master-plan.odt
{archive,nohidden,readonly,nosystem,noappendonly,nonodump,noimmutable,
 nonounlink,offline,nosparsed}
```

アクセス制御リストの管理

アクセス制御リスト (ACL) は、ファイルまたはディレクトリのアクセス権を定義する表です。表内の各レコードまたはアクセス制御エントリ (ACE) は、特定のユーザー、グループ、またはユーザーまたはグループのクラスのアクセス権を定義します。デフォルトでは、Oracle HSM リリース 6.1 で作成する新しいファイルシステム

は、ネットワークファイルシステム (NFS) バージョン 4 および Solaris 11 で導入されたアクセス制御リスト (ACL) 実装を使用します。

Solaris ACL の管理、構文、および使用の包括的な説明は、このドキュメントの範囲外です。詳細は、docs.oracle.com にある *Oracle Solaris 11.1 Information Library* で入手可能なボリューム『*Oracle Solaris 11.1 の管理: ZFS ファイルシステム*』の章「ACL および属性を使用した Oracle Solaris ZFS ファイルの保護」を参照してください。また、Solaris の *ls* と *chmod* のマニュアルページも参照してください。

第5章 ライブラリ、メディア、およびドライブの管理

この章では、次のトピックを扱います。

- [自動メディアライブラリの管理](#)
- [ドライブの管理](#)
- [リムーバブルメディアの管理](#)。

自動メディアライブラリの管理

このセクションでは、ライブラリの保守と管理に関する基本的なタスクについて説明します。

- [ライブラリのオンライン化およびオフライン化](#)
- [リムーバブルメディアのインポートおよびエクスポート](#)
- [ライブラリカタログの保持](#)
- [ドライブをライブラリに取り付ける順序の判別](#)。

ライブラリのオンライン化およびオフライン化

- [ライブラリのオフライン化](#)
- [ライブラリのオンライン化](#)。

ライブラリのオフライン化

Oracle HSM の操作を 1 つのライブラリでのみ停止する必要がある場合、またはライブラリの電源を切断する必要がある場合は、後述するように、まずライブラリをオフラインにします。

1. ファイルシステムホストに *root* としてログインします。

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

2. アクティブなアーカイブ処理とステージングジョブを完了し、新規ジョブが開始されないようにします。「[アーカイブおよびステージングプロセスのアイドル状態](#)」を参照してください。
3. ドライブとライブラリのアクティビティを停止します。「[アーカイブおよびステージングプロセスの停止](#)」を参照してください。
4. ライブラリをオフラインにします。コマンド `samcmd off library-equipment-number` を使用します。ここで `library-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイル内のライブラリに割り当てられた装置の順序番号です。

ライブラリを `off` 状態にすると、入出力操作が停止し、ライブラリが Oracle HSM software の制御を受けなくなります。電源が切断されていないすべてのドライブは、`on` 状態のままとなります。次の例では、ライブラリ `800` をオフラインにして、`samcmd c` を使用して結果を確認します。

```
root@solaris:~# samcmd off 800
root@solaris:~# samcmd c
Device configuration samcmd      5.4 14:34:04 Mar  7 2014
samcmd on hsmfs1host
Device configuration:
ty  eq state  device_name                fs  family_set
sn  800 off    /dev/scsi/changer/c1t2d0   800 lib800
li  801 on     /dev/rmt/0cbn              800 lib800
li  802 on     /dev/rmt/1cbn              800 lib800
li  803 on     /dev/rmt/2cbn              800 lib800
li  804 on     /dev/rmt/3cbn              800 lib800
hy  900 on     historian                    900
root@solaris:~#
```

5. 準備ができたなら、ライブラリをオンラインにします。

ライブラリのオンライン化

1. ファイルシステムホストに `root` としてログインします。

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

2. ライブラリをオンラインにします。コマンド `samcmd on library-equipment-number` を使用します。ここで `library-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイル内のライブラリに割り当てられた装置の順序番号です。

ライブラリがオンラインになります。Oracle HSM software はデバイスの状態を問い合わせ、必要に応じてカタログを更新します。次の例では、ライブラリ `800` をオンラインにして、`samcmd c` を使用して結果を確認します。

```
root@solaris:~# samcmd on 800
root@solaris:~# samcmd c
Device configuration samcmd      5.4 15:04:14 Mar  7 2014
samcmd on hsmfs1host
Device configuration:
  ty  eq state  device_name                fs  family_set
sn  800 on    /dev/scsi/changer/c1t2d0    800 lib800
li  801 on    /dev/rmt/0cbn               800 lib800
li  802 on    /dev/rmt/1cbn               800 lib800
li  803 on    /dev/rmt/2cbn               800 lib800
li  804 on    /dev/rmt/3cbn               800 lib800
hy  900 on    historian                    900
root@solaris:~#
```

3. ここで停止します。

リムーバブルメディアのインポートおよびエクスポート

多くの自動ライブラリには、物理的にライブラリに入ることなく、メディアカートリッジを追加または削除できるローディングベイが含まれています。ベンダーによっては、これはメールボックス、メールスロット、メディアアクセスポート (MAP)、またはカートリッジアクセスポート (CAP) と呼ばれることがあります。このタイプのライブラリでは、Oracle HSM コマンドを使用して次のタスクを実行できます。

- [リムーバブルメディアカートリッジのインポート](#)
- [リムーバブルメディアカートリッジのエクスポート](#)。

ライブラリにメールボックスが含まれていない場合、ライブラリベンダーのドキュメントおよびローカルサイトのポリシーでライブラリメディアの追加と削除に関する手順を確認してください。変更後にライブラリが再初期化され、その内容を監査すると、Oracle HSM software はライブラリとヒストリアンカタログを自動的に更新します。

リムーバブルメディアカートリッジのインポート

Oracle HSM software の起動時にライブラリメールボックスにメディアカートリッジが含まれている場合、ソフトウェアはこれらのメディアカートリッジをライブラリに自動的にロードします。ソフトウェアの実行後には、次の手順を使用して、いつでもメディアをメールボックスからインポートできます。

1. ライブラリベンダーの指示に従ってメディアカートリッジをメールボックスに入れます。
2. ファイルシステムホストに *root* としてログインします。

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

3. カートリッジを自動ライブラリにインポートします。コマンド *samimport library-equipment-number* を使用します。ここで *library-equipment-number* は、*/etc/opt/SUNwsamfs/mcf* ファイル内でライブラリに指定された装置の順序番号です。

Oracle HSM software は、ストレージスロットにメディアを割り当て、それらの場所をカタログ化します。次の例では、メディアをライブラリ *800* にインポートします。

```
root@solaris:~# samimport 800
```

4. ここで停止します。

リムーバブルメディアカートリッジのエクスポート

1. ファイルシステムホストに *root* としてログインします。

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

2. 必要に応じて、エクスポート前にカートリッジのカタログレコードに情報メモを追加します。コマンド `chmed -I "note" identifier` を使用します。ここで `note` は最大 128 文字の文字列で、`identifier` は次のいずれかです。
 - `mediatype.volume-serial-number`。ここで `mediatype` は、[付録A「装置タイプの用語集」](#) に一覧表示されている 2 文字のメディアタイプコードの 1 つで、`volume-serial-number` は、ライブラリ内のボリュームを一意に識別する 6 文字の英数字の文字列です。
 - `library-equipment-number:slot`。ここで `library-equipment-number` は、`/etc/opt/SUNwsamfs/mcf` ファイル内で自動テープライブラリに指定される装置の順序番号で、`slot` は、カートリッジがあるライブラリ内のスロットアドレスです。

ボリュームのエクスポート後に、メモはヒストリアンカタログ内に保持されます。次の例では、LTO (*li*) カートリッジ `VOL054` のカタログエントリにメモを追加します。

```
root@solaris:~# chmed -I "To vault 20150411" li.VOL054
```

3. 指定したストレージスロットからメールボックスにカートリッジを移動するには、コマンド `samexport library-equipment-number:slot` を使用します。ここで `library-equipment-number` は、`/etc/opt/SUNwsamfs/mcf` ファイル内で自動テープライブラリに指定される装置の順序番号で、`slot` は、カートリッジがあるライブラリ内のスロットアドレスです。

次の例では、ライブラリ `800` のスロット `11` 内にある磁気テープカートリッジをエクスポートします。

```
root@solaris:~# samexport 800:11
```

4. 指定したカートリッジをメールボックスに移動するには、コマンド `samexport mediatype.volume-serial-number` を使用します。ここで `mediatype` は、[付録A「装置タイプの用語集」](#) にリストされている 2 文字のメディアタイプコードの 1 つで、`volume-serial-number` は、ライブラリ内のボリュームを一意に識別する英数字の文字列です。

Oracle HSM software は、Oracle HSM [ヒストリアン](#)によって保持されるカタログにカートリッジを追加します。次の例では、LTO (*Li*) テープカートリッジ *VOL109* をエクスポートします。

```
root@solaris:~# samexport li.VOL109
```

5. ライブラリベンダーの指示に従ってメディアカートリッジをメールボックスから削除します。
6. ここで停止します。

ライブラリカタログの保持

Oracle Hierarchical Storage Manager ライブラリカタログは、自動ライブラリとその内容のソフトウェアでの内部表現です。自動ライブラリが直接接続されている場合、Oracle HSM software はライブラリとその内容を完全に制御できます。そのため、ライブラリカタログのエントリは、物理ライブラリ内のスロットの1対1の表現になります。自動ライブラリがネットワーク接続されている場合、Oracle HSM は、ライブラリソフトウェアが仮想ライブラリまたはライブラリパーティションの形式で使用可能にするライブラリの部分にのみアクセスします。そのため、Oracle HSM ライブラリカタログのエントリは、ライブラリの一部のコンテンツのみを反映します。

このセクションでは、次のタスクについて説明します。

- [ライブラリカタログの表示](#)
- [ライブラリスロットの内容の監査](#)
- [直接接続自動ライブラリ全体の監査](#)
- [カタログからのメディアエラーのクリア](#)。

ライブラリカタログの表示

1. もっとも一般的に使用されるライブラリカタログ情報を表示するには、コマンド `samcmd v library-equipment-number` を使用します。ここで `library-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルが自動テープライブラリに割り当てる装置の順序番号です。

次の例では、ライブラリ *800* のカタログを表示します


```

root@solaris:~# samcmd v 800
Robot catalog samcmd      5.4      16:45:25 Mar 14 2014
samcmd on samqfshost                      count 32
Robot VSN catalog by slot      : eq 800
slot      access time count use  flags      ty vsn
   0      2014/03/14 11:23 875  0%  -il-o-b----- li VOL001
   1      2014/03/13 17:54 866  0%  -il-o-b----- li VOL002
   2      2014/03/14 11:26   3  0%  -il-o-b----- li VOL003
   3      2014/03/14 10:33   3  0%  -il-o-b----- li VOL004
   4      2014/03/14 11:34   5  0%  -il-o-b----- li VOL005
   5      2014/03/14 11:32   2  0%  -ilEo-b----f  li VOL006 MEDIA ERROR
   6      2014/03/13 18:07   2  0%  -il-o-b----- li VOL007
   7      2014/03/13 18:07   1  0%  -il-o-b----- li VOL008
   8      2014/03/13 18:07   1  0%  -il-o-b----- li VOL009
...
  18      2014/03/13 18:16   1  0%  -il-o-b----- li VOL019
  19      none                50  0%  -il-oCb----- li CLN020

```

2. `samcmd v` の表示を使用してボリュームのステータスを判別するには、`flags` 列内のエントリを調べて、次の一覧を確認します。
- *A* は、スロットで監査が必要なことを意味します。
 - *C* は、スロットにクリーニングカートリッジが含まれていることを意味します。
 - *E* は、ボリュームが不良であるか、クリーニングメディアが期限切れであることを意味します。
 - *L* は、ボリュームが Linear Tape File System (LTFS) ボリュームであることを意味します。
 - *N* は、ボリュームが外部メディアである (Oracle HSM 形式ではない) ことを意味します。
 - *R* は、ボリュームが読み取り専用であることを意味します (ソフトウェアフラグ)。
 - *U* は、ボリュームが使用不可であることを意味します。
 - *W* は、ボリュームが物理的に書き込み保護されていることを意味します。

- *x* は、スロットがエクスポートスロットであることを意味します。
 - *b* は、ボリュームにバーコードがあることを意味します。
 - *c* は、ボリュームのリサイクルがスケジュールされていることを意味します。
 - *f* は、ボリュームに空きがないかボリュームが壊れていることがアーカイバで検出されたことを意味します。
 - *d* は、ボリュームに重複するボリュームシリアル番号 (VSN) があることを意味します。
 - *l* は、ボリュームにラベルが付けられていることを意味します。
 - *o* は、スロットが占有されていることを意味します。
 - *p* は、ボリュームに優先度の高いボリュームがあることを意味します。
 - *-* は、対応するフラグが設定されていないことを意味します。
3. `samcmd v` の表示を使用して、ボリュームに使用されるメディアのタイプを識別するには、`ty` 列を調べて、[付録A「装置タイプの用語集」](#) 内または `mcf` のマニュアルページに表示されるコードを検索します。
 4. カタログ内のすべての情報を一覧表示するには、コマンド `dump_cat catalog-path-name` を使用します。ここで `catalog-path-name` は、`/etc/opt/SUNWsamfs/mcf` ファイルで指定したカタログファイルのパスとファイル名です。

次の例では、カタログファイル `catalog/800_cat` をダンプします。

```
root@solaris:~# dump_cat catalog/800_cat
# audit_time Wed Dec 31 17:00:00 1969
# version 530 count 32 mediatype
#Index VSN    Barcode  Type PTOC  Access Capacity ...  LVTime LVPos
#
  0     S00001 S00001L4 li    0x747   875   512000 ...    0   0x3
  1     S00002 S00002L4 li    0x5db   866   512000 ...    0   0x3
 13     S00014 S00014L4 li      0     4   512000 ...    0   0
 17     S00018 S00018L4 li      0     1   512000 ...    0   0
 18     S00003 S00003L4 li      0     3   512000 ...    0   0
```

5. ここで停止します。

ライブラリスロットの内容の監査

リムーバブルメディアボリュームについて報告された残りの容量でライブラリカタログを更新するには、ライブラリスロットを監査します。コマンド `auditslot` を使用します。

1. ファイルシステムホストに `root` としてログインします。

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

2. 指定したテープボリュームを監査するには、EOD (データの終わり) にスキップして、空き領域を更新し、コマンド `auditslot -e library-equipment-number:slot` を使用します。ここで `library-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルがライブラリに割り当てる装置の順序番号で、`slot` は、ライブラリ内でのカートリッジの場所です。

`auditslot` コマンドは、ボリュームを含むカートリッジをロードし、ラベルを読み取り、スロットのライブラリカタログエントリを更新します。開始後は EOD へのスキップを中断できません。状況によっては、完了までに数時間かかる場合があります。次の例では、テープライブラリ `800` 内のスロット `11` を監査します。

```
root@solaris:~# auditslot -e 800:11
root@solaris:~#
```

3. 指定した光磁気ボリュームを監査するには、コマンド `auditslot library-equipment-number:slot[:side]` を使用します。ここで `library-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルがライブラリに割り当てる装置の順序番号で、`slot` は、ライブラリ内でのカートリッジの場所で、`side` (オプション) は、両面光磁気ディスクの指定したサイドです。

次の例では、光磁気ライブラリ `700` のスロット `21` 内にあるボリュームのサイド `1` を監査します。

```
root@solaris:~# auditslot 800:21:1
root@solaris:~#
```

4. ここで停止します。

直接接続自動ライブラリ全体の監査

完全監査では、各カートリッジをドライブにロードし、ラベルを読み取り、ライブラリカタログを更新します。ライブラリの監査は、次のような状況で行います。

- Oracle HSM コマンドを使用しないで、自動ライブラリ内でカートリッジを移動したあと
- ライブラリカタログが信頼できない可能性があるとき (たとえば、停電後)
- メールボックスのない自動ライブラリにおいてカートリッジの追加、削除、または移動を行なったとき。

完全監査を実行するには、コマンド `samcmd audit library-equipment-number` を使用します。ここで `library-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルがライブラリに割り当てる装置の順序番号です。

完全監査は、メディアを含むスロットの数によって時間がかかる場合があります。

次の例では、テープライブラリ `800` を監査します。

```
root@solaris:~# audit 800
root@solaris:~#
```

カタログからのメディアエラーのクリア

リムーバブルメディアカートリッジの使用で問題が発生すると、Oracle HSM は対応するカタログエントリ上でエラーフラグを設定します。メディアが摩耗している、損傷している、またはクリーニングメディアのケースが期限切れである可能性があります。このような場合は、メディアを再使用しないようにしてください。ただし、メディアへのアクセスに関する問題はドライブの障害によっても生じることがあり、その場合メディアは支障なく再利用できます。後者の場合、カートリッジのエラーフラグをクリアする必要があります。

エラーフラグをクリアする前に、必ず問題の本質を把握してください。エラーフラグは Oracle HSM の操作とデータのセキュリティに不可欠です。実際にカートリッジに障害がある場合、このフラグをクリアすることはできません。

確信がある場合、エラーをクリアしてカートリッジの使用を試みることができます。次のように進めます。

1. ファイルシステムホストに *root* としてログインします。

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

2. リムーバブルメディアボリュームのステータスを確認します。コマンド *samcmd r* を使用します。

この例では、*samcmd r* コマンドにより、ドライブ *801* が LTO (*Li*) ボリューム *VOL004* にエラーフラグを設定したことを示しています。

```
root@solaris:~# samcmd r
Removable media status: all          samcmd 5.4          17:40:11 Mar 13 2014
ty  eq  status      act  use  state  vsn
li  801 -E-----r    0   0%  notrdy VOL004 MEDIA ERROR
      MEDIA ERROR
li  802 -----p    0   0%  notrdy
      empty
li  803 -----p    0   0%  notrdy
      empty
li  804 -----p    0   0%  notrdy
      empty
root@solaris:~#
```

3. エラーフラグを設定したドライブが疑わしい場合には、カートリッジをアンロードし、エラーフラグをクリアします。コマンド *samcmd unload drive-number* を使用します。ここで *drive-number* は、*/etc/opt/SUNWsamfs/mcf* ファイル内でドライブに指定された装置の順序番号です。

次の例では、ドライブ *801* をアンロードします。

```
root@solaris:~# samcmd unload 801
```

4. 指定したボリュームシリアル番号およびメディアタイプのメディアエラーフラグをクリアするには、コマンド *chmed -E media-type.volume-serial-number* を使用します。ここで *mediatype* は、付録A「装置タイプの用語集」に一覧表示されている2文字のメディアタイプコードの1つで、*volume-serial-number* は、ライブラリ内のボリュームを一意に識別する英数字の文字列です。

次の例では、LTO (*Li*) ボリューム *VOL004* 上でエラーフラグをクリアします。

```
root@solaris:~# chmed -E li.VOL004
      3:0 li VOL004      A1l---b-----      2.3T      2.3T      0              0 800 4 0 //
root@solaris:~#
```

5. 指定されたライブラリスロットにあるカートリッジのメディアエラーフラグをクリアするには、コマンド `chmed -E library-equipment-number:slot[:disk-side]` を使用します。ここで、*library-equipment-number* は `/etc/opt/SUNWsamfs/mcf` ファイルが自動テープライブラリに割り当てる装置の順序番号で、*slot* はターゲットボリュームがあるライブラリ内のスロットアドレスで、オプションの *disk-side* 値 (1 または 2) は両面光磁気ディスクのいずれかの面を指定します。

次の例では、ライブラリ *800* のスロット *31* 内にあるカートリッジのエラーフラグをクリアします。

```
root@solaris:~# chmed -E 800:31
```

6. ライブラリカタログを更新して、変更を反映します。コマンド `auditslot -e library-equipment-number:slot[:disk-side]` を使用します。

次の例では、ライブラリ *800* のスロット *31* を監査して、カタログを更新します。

```
root@solaris:~# auditslot -e 800:31
root@solaris:~#
```

7. 別のドライブにカートリッジをマウントして、エラーが再発するかどうかを確認します。コマンド `samcmd load media-type.volume-serial-number` を使用します。ここで *mediatype* は、付録A「装置タイプの用語集」に一覧表示されている2文字のメディアタイプコードの1つで、*volume-serial-number* は、ライブラリ内のボリュームを一意に識別する英数字の文字列です。

```
root@solaris:~# samcmd load li.VOL004
root@solaris:~#
```

- リムーバブルメディアボリュームのステータスを再度確認します。コマンド `samcmd r` を使用します。

```
root@solaris:~# samcmd r
Removable media status: all          samcmd 5.4          17:42:10 Mar 13 2014
ty  eq  status      act  use  state  vsn
li  801  -----p    0   0%  notrdy
      empty
li  802  --l-----r    0   0%  ready  VOL004
      idle
li  803  -----p    0   0%  notrdy
      empty
li  804  -----p    0   0%  notrdy
      empty
root@solaris:~#
```

- 新しいドライブでエラーが再発しない場合、カートリッジはおそらく正常です。
- エラーが再発する場合、リムーバブルメディアボリュームを廃棄することを検討してください。
- ここで停止します。

ヒストリアンカタログの管理

Oracle Hierarchical Storage Manager ヒストリアンは、カタログを持つが装置を持たない疑似ライブラリです。ヒストリアンは、Oracle HSM によって直接制御されなくなったボリュームをカタログします。そのため、ライブラリからエクスポートされてサイト外でのストレージのために送信されたボリュームや、スタンドアロンドライブに手動でロードされたボリュームのレコードを保持します。ライブラリからボリュームをエクスポートすると、Oracle HSM はヒストリアンカタログを自動的に更新します。ただし、レコードを追加または削除して、メモを添付することによって、手動での記録の保持にヒストリアンを使用することもできます。通常、ヒストリアンとの対話は、物理メディアライブラリとの対話と同程度です。

このセクションでは、次のタスクの概要について説明します。

- [ヒストリアンカタログの表示](#)
- [ヒストリアンカタログへのエントリの追加](#)

- [ヒストリアンカタログからのエントリの削除](#)。

ヒストリアンカタログの表示

ヒストリアンカタログは、物理ライブラリとまったく同じように表示されます。コマンド `samcmd v historian-equipment-number` を使用します。ここで `historian-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルがヒストリアンに割り当てる装置の順序番号です。

次の例では、装置の順序番号が `900` のヒストリアンのカタログを表示します。

```
root@solaris:~# samcmd v 900
Robot catalog samcmd      5.4      16:45:25 Mar 14 2014
samcmd on samqfshost      count 32
Robot VSN catalog by slot : eq 900
slot      access time count use  flags      ty vsn
  0      2014/03/14 11:23 875  0%  -il-o-b-----  li EXT001
  1      2014/03/13 17:54 866  0%  -il-o-b-----  li EXT002
```

ヒストリアンカタログへのエントリの追加

ヒストリアンカタログにエントリを追加するには、次の手順を実行します。

1. 指定したボリュームシリアル番号のヒストリアンカタログにエントリを追加するには、コマンド `samimport -v volume-serial-number -m mediatype historian-equipment-number` を使用します。ここでは:
 - `volume-serial-number` は、カタログ内のボリュームを一意に識別する英数字の文字列です。
 - `mediatype` は、[付録A「装置タイプの用語集」](#) に一覧表示されている 2 文字のメディアタイプコードの 1 つです。
 - `historian-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルがヒストリアンに割り当てる装置の順序番号です。

次の例では、LTO (`li`) ボリューム `EXT003` のレコードをヒストリアン `900` のカタログに追加します。

```
root@solaris:~# samimport -v EXT003 -m li 900
]
```

2. 指定したバーコードのヒストリアンカタログにエントリを追加するには、コマンド `samimport -b barcode -m mediatype historian-equipment-number` を使用します。ここで `barcode` は、対応する物理カートリッジに付けられたバーコードです。

次の例では、バーコード *EXT003L4* が付いた LTO (*Li*) ボリュームのレコードをヒストリアン *900* のカタログに追加します。

```
root@solaris:~# samimport -b EXT003L4 -m li 900
```

3. ここで停止します。

ヒストリアンカタログからのエントリの削除

ヒストリアンカタログからエントリを削除するには、コマンド *samexport historian-equipment-number:slot* を使用します。ここで *historian-equipment-number* は、*/etc/opt/SUNWsamfs/mcf* ファイルがヒストリアンに割り当てる装置の順序番号で、*slot* は、レコードのヒストリアンスロットアドレスです。

次の例では、ヒストリアン *900* のカタログのスロット *1* 内にあるボリューム *EXT002* のレコードを削除します。

```
root@solaris:~# samcmd v 900
Robot catalog samcmd      5.4      16:45:25 Mar 14 2014
samcmd on samqfshost      count 32
Robot VSN catalog by slot : eq 900
slot      access time count use flags      ty vsn
  0      2014/03/14 11:23 875 0% -il-o-b----- li EXT001
  1      2014/03/13 17:54 866 0% -il-o-b----- li EXT002
  2      2014/03/13 17:57 866 0% -il-o-b----- li EXT003
root@solaris:~# samexport 900:1
```

ヒストリアン情報の更新

ヒストリアンカタログエントリの情報フィールドを更新して、エクスポートされるボリュームの処理またはステータスに対する変更を示すことができます。コマンド *chmed -I "note" identifier* を使用します。ここで *note* は最大 128 文字の文字列で、*identifier* は次のいずれかです。

- *mediatype.volume-serial-number*。ここで *mediatype* は、[付録A「装置タイプの用語集」](#) に一覧表示されている 2 文字のメディアタイプコードの 1 つで、*volume-serial-number* は、ライブラリ内のボリュームを一意に識別する英数字の文字列です。または、次のコマンドを使用します
- *library-equipment-number:slot*。ここで *library-equipment-number* は、*/etc/opt/SUNWsamfs/mcf* ファイルが自動テープライブラリに割り当てる装置の順序番号で、*slot* は、カートリッジがあるライブラリ内のスロットアドレスです。

次の例では、LTO (*Li*) カートリッジ *VOL06E* がボールドから回収され、正常に検証されて、ボールドに返されたことを示します。

```
root@solaris:~# chmed -I "validated and revaulted 20150310" li.VOL06A
```

ドライブをライブラリに取り付ける順序の判別

自動ライブラリに複数のドライブが含まれている場合は、*mcf* ファイル内のドライブの順序が、ライブラリコントローラに表示されるドライブの順序と同じである必要があります。この順序は、ホストで表示され、ホストの */var/adm/messages* ファイルで報告されるデバイスの順序とは異なる場合があります。そのため、Oracle Hierarchical Storage Manager メタデータサーバーとデータムーバーホストの構成、ライブラリの変更、またはライブラリの構成の変更を行うときは常に、次に一覧表示するタスクを実行してドライブの順序を確認する必要があります。

- [ライブラリと Solaris ホストのドライブ情報の収集](#)
- 使用している装置に応じて、[直接接続ライブラリ内のドライブを Solaris デバイス名にマップするか](#)、[ACSL S 接続ライブラリ内のドライブを Solaris デバイス名にマップ](#)します。

ライブラリと Solaris ホストのドライブ情報の収集

1. ライブラリのドキュメントを参照してください。ドライブとターゲットの識別方法を確認してください。ローカルオペレータパネルがある場合、これを使用してドライブの順序を判別する方法を参照してください。
2. ライブラリにローカルオペレータパネルがマウントされている場合、これを使用して、ドライブをコントローラに接続する順序を判別します。各ドライブの SCSI ターゲット ID または World Wide Name を判別します。
3. Solaris ホストに *root* としてログインします。

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

4. */dev/rmt/* 内に Solaris 論理デバイス名を一覧表示して、出力をテキストファイルにリダイレクトします。

次の例では、*/dev/rmt/* のリストを *root* ユーザーのホームディレクトリ内のファイル *device-mappings.txt* にリダイレクトします。

```
root@solaris:~# ls -l /dev/rmt/ > /root/device-mappings.txt
```

- 次に、使用している装置 (直接接続テープライブラリまたは ACSLS 接続ライブラリ) に固有の手順を使用して、ドライブを Solaris デバイス名にマップします。

直接接続ライブラリ内のドライブを Solaris デバイス名にマップ

`/dev/rmt/` 内に一覧表示されている Solaris 論理ドライブ名ごと、およびライブラリが Oracle HSM サーバーホストに割り当てるドライブごとに、次の手順を実行します。

- まだ Oracle HSM Solaris ホストにログインしていない場合、`root` としてログインします。

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

- ドライブが使用されないように、実行中のすべてのアーカイブ処理を停止します。「[アーカイブおよびステージングプロセスのアイドル状態](#)」および「[アーカイブおよびステージングプロセスの停止](#)」を参照してください。
- テキストエディタで、「[ライブラリと Solaris ホストのドライブ情報の収集](#)」の手順で作成したデバイスマッピングファイルを開きます。単純な表にファイルを整理して、変更内容を保存します。

後続の手順でこの情報を参照する必要があります。次の例では、`vi` エディタを使用して、権限、所有権、日付属性を `/dev/rmt/` リストから削除して、ライブラリデバイス情報の見出しと領域を追加しました。

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

:w
```

- ライブラリで、すべてのドライブが空になっていることを確認します。
- Solaris 論理デバイス名にまだマップしていないライブラリ内の最初のドライブにテープをロードします。

次の例のために、LTO4 テープを HP Ultrium LTO4 テープドライブにロードします。

- ドライブをテープライブラリにマップする場合は、テープをマウントするドライブに対応する、Solaris `/dev/rmt/` エントリを識別します。ドライブを特定するまで、コマンド `mt -f /dev/rmt/number status` を実行します。ここで `number` は、`/dev/rmt/` 内のドライブを識別します。

次の例では、`/dev/rmt/0` にあるドライブは空ですが、`/dev/rmt/1` にあるドライブにはテープが保持されています。そのため、ライブラリがドライブ 1 として識別するドライブは、Solaris `/dev/rmt/1` に対応します。

```
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
```

- 前の手順で作成したデバイスマッピングファイルで、テープが保持されている Solaris デバイスのエントリを見つけて、表示されるスペースにライブラリのデバイス ID を入力します。次に、ファイルを保存します。

次の例では、`/dev/rmt/1` の行の `LIBRARY DEVICE NUMBER` フィールドに `1` を入力します。

```
root@solaris:~# vi /root/device-mappings.txt
LIBRARY SOLARIS          SOLARIS
DEVICE LOGICAL          PHYSICAL
NUMBER DEVICE          DEVICE
-----
          /dev/rmt/0cbn -> ../../devices/pci@8.../st@w500104f00093c438,0:cbn
1      /dev/rmt/1cbn -> ../../devices/pci@8.../st@w500104f0008120fe,0:cbn
```

```

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

:w

```

8. テープをアンロードします。
9. すべてのデバイスを Solaris 論理デバイス名にマップするエントリがデバイスマッピングファイルに保持されるまで、この手順を繰り返します。その後、ファイルを保存してエディタを閉じます。

```

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

:wq
root@solaris:~#

```

10. ここで停止します。マッピングファイルをあとで使用するために保持します。

ACSL5 接続ライブラリ内のドライブを Solaris デバイス名にマップ

1. まだ Oracle HSM Solaris ホストにログインしていない場合、*root* としてログインします。

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

2. ドライブが使用されないように、実行中のすべてのアーカイブ処理を停止します。「[アーカイブおよびステージングプロセスのアイドル状態](#)」および「[アーカイブおよびステージングプロセスの停止](#)」を参照してください。
3. テキストエディタで、「[ライブラリと Solaris ホストのドライブ情報の収集](#)」の手順で作成したデバイスマッピングファイルを開きます。単純な表にファイルを整理します。

後続の手順でこの情報を参照する必要があります。次の例では、`vi` エディタを使用して、権限、所有権、日付属性を `/dev/rmt/` リストから削除して、ライブラリデバイス情報の見出しと領域を追加します。

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

4. `/dev/rmt/` に一覧表示されている論理デバイス名ごとに、コマンド `luxadm display /dev/rmt/number` を使用してシリアル番号を表示します。ここで `number` は、`/dev/rmt/` 内のドライブを識別します。

次の例では、デバイス `/dev/rmt/0` のシリアル番号 `HU92K00200` を取得します。

```
root@solaris:~# luxadm display /dev/rmt/0
DEVICE PROPERTIES for tape: /dev/rmt/0
Vendor: HP
Product ID: Ultrium 4-SCSI
Revision: G25W
Serial Num: HU92K00200
...
Path status: Ready
root@solaris:~#
```

5. 次に、テキストエディタを使用して、`device-mappings.txt` ファイルの対応する行に各デバイスのシリアル番号を入力します。

次の例では、デバイス `/dev/rmt/0` のシリアル番号を `vi` エディタを使用して `device-mappings.txt` ファイルに記録します。

```
root@solaris:~# vi /root/device-mappings.txt
SOLARIS LOGICAL DEVICE  DEVICE SERIAL NUMBER  ACSLS DEVICE ADDRESS
```

```

-----
/dev/rmt/0          HU92K00200
/dev/rmt/1
/dev/rmt/2
/dev/rmt/3

```

6. `/dev/rmt/` にマップされているデバイスのシリアル番号ごとに、対応する ACSLS ドライブアドレスを取得します。ACSLs コマンド `display drive * -f serial_num` を使用します。

次の例では、デバイス `HU92K00200 (/dev/rmt/0)`、`HU92K00208 (/dev/rmt/1)`、`HU92K00339 (/dev/rmt/2)`、`HU92K00289 (/dev/rmt/3)` の ACSLS アドレスを取得します。

```

ACSSA> display drive * -f serial_num
2014-03-29 10:49:12 Display Drive
Acs  Lsm  Panel  Drive  Serial_num
0    2    10    16    331002031352
0    2    10    17    HU92K00200
0    2    10    18    HU92K00208
0    3    10    10    HU92K00339
0    3    10    11    HU92K00189
0    3    10    12    HU92K00289
root@solaris:~#

```

7. テキストエディタを使用して、`device-mappings.txt` ファイルの対応する行に各シリアル番号に対する ACSLS アドレスを入力します。ファイルを保存して、エディタを閉じます。

次の例では、`vi` エディタを使用して `device-mappings.txt` ファイルに情報を記録します。

```

root@solaris:~# vi /root/device-mappings.txt
SOLARIS LOGICAL DEVICE DEVICE SERIAL NUMBER ACSLS DEVICE ADDRESS
-----
/dev/rmt/0          HU92K00200          (acs=0, lsm=2, panel=10, drive=17)
/dev/rmt/1          HU92K00208          (acs=0, lsm=2, panel=10, drive=18)

```

```
/dev/rmt/2          HU92K00339          (acs=0, lsm=2, panel=10, drive=10)
/dev/rmt/3          HU92K00289          (acs=0, lsm=2, panel=10, drive=12)
:wq
root@solaris:~#
```

8. ここで停止します。マッピングファイルをあとで使用するために保持します。

ドライブの管理

次のものを含む、さまざまなドライブ管理タスクを Oracle HSM インタフェースから処理できます。

- [ドライブのロードおよびアンロード](#)
- [テープドライブのクリーニング](#)
- [暗号化機能のあるドライブの使用](#)
- [ドライブの問題の処理](#)。

ドライブのロードおよびアンロード

リムーバブルメディアが自動ライブラリ内に保管されているときは、ファイルシステムのアーカイブ処理とステージング処理プロセスによって、必要に応じてカートリッジがドライブに自動的にロードされます。ただし、リムーバブルメディアファイルの管理、Oracle HSM 構成のバックアップ、またはファイルシステムの回復時に、要求に応じてカートリッジをロードすることもできます。このセクションでは、次のトピックを扱います。

- [自動ライブラリに取り付けられたドライブのロードおよびアンロード](#)
- [スタンドアロンドライブの手動でのロードおよびアンロード](#)
- [ボリュームを手動でロードする必要があるときにオペレータに通知](#)。

自動ライブラリに取り付けられたドライブのロードおよびアンロード

- [指定したライブラリ位置からのドライブのロード](#)
- [指定したメディアタイプおよびボリュームシリアル番号のドライブのロード](#)
- [ライブラリで指定したドライブのアンロード](#)。

指定したライブラリ位置からのドライブのロード

コマンド `samcmd load library-equipment-number:slot[:disk-side]` を使用します。ここで `library-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルが自動テープライブラリに割り当てる装置の順序番号で、`slot` は、ターゲットボリュームがあるライブラリ内のスロットアドレスで、オプションの `disk-side` 値 (1 または 2 のいずれか) は、両面光磁気ディスクのいずれかのサイドを指定します。

カートリッジは、ライブラリ内で次に利用可能なドライブにロードされます。次の例では、ライブラリ `800` のスロット `11` 内にある磁気テープカートリッジをロードします。

```
root@solaris:~# samcmd load 800:11
```

指定したメディアタイプおよびボリュームシリアル番号のドライブのロード

コマンド `samcmd load mediatype.volume-serial-number` を使用します。ここで `mediatype` は、付録A「装置タイプの用語集」に一覧表示されている2文字のメディアタイプコードの1つで、`volume-serial-number` は、ライブラリ内のボリュームを一意に識別する英数字の文字列です。

カートリッジは、ライブラリ内で次に利用可能なドライブにロードされます。次の例では、LTO (`li`) テープカートリッジ `VOL109` をロードします。

```
root@solaris:~# samcmd load li.VOL109
```

ライブラリで指定したドライブのアンロード

コマンド `samcmd unload drive-equipment-number` を使用します。ここで `drive-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルがドライブに割り当てる装置の順序番号です。

ドライブが `unavail` の場合でもカートリッジはアンロードされます。次の例では、ドライブ `801` をアンロードします。

```
root@solaris:~# samcmd unload 801]
```

スタンドアロンドライブの手動でのロードおよびアンロード

Oracle HSM software は、スタンドアロンのリムーバブルメディアドライブを、独自のカタログを持つ小さい単一スロットのライブラリと同様に扱います。

スタンドアロンドライブへのカートリッジのロード

スタンドアロンドライブをロードするには、製造元の指示に従ってカートリッジをドライブに入れます。Oracle HSM システムは、カートリッジがロードされたことを認識し、ラベルを読み取り、ドライブのカタログを更新します。

スタンドアロンドライブからのカートリッジのアンロード

スタンドアロンドライブをアンロードするには、次の手順を実行します。

1. ドライブをアイドル状態にします。コマンド `samcmd idle drive-equipment-number` を使用します。ここで `drive-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルがドライブに割り当てる装置の順序番号です。

ドライブがアイドル状態になると、Oracle HSM ソフトウェアは、ドライブを使用する現在のアーカイブ処理プロセスをすべて終了しますが、新しいプロセスは開始しません。

```
root@solaris:~# samcmd idle 801
```

2. Oracle HSM がドライブを終了して `off` にするまで待ちます。

ドライブのステータスを確認するには、コマンド `samcmd r` を使用します。

3. ベンダーの指示に従ってカートリッジを削除します。
4. ここで停止します。

ボリュームを手動でロードする必要があるときにオペレータに通知

スタンドアロンドライブを使用する場合、またはボルトや、ライブラリの外部にあるほかの場所に必要なカートリッジを保管する場合、オペレータが非常駐カートリッジをロードする必要があるときに、Oracle HSM software は指定したアドレスに電子メールを送信できます。この機能を有効にするには、次の手順に従います。

ロード通知の有効化

1. ファイルシステムホストに `root` としてログインします。

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

2. ファイル `load_notify.sh` をディレクトリ `/opt/SUNWsamfs/examples/` からディレクトリ `/etc/opt/SUNWsamfs/scripts/` にコピーします。

次のコマンドは 1 行で入力し、改行はバックスラッシュ文字でエスケープします。

```
root@solaris:~# cp /opt/SUNWsamfs/examples/load_notify.sh /  
/etc/opt/SUNWsamfs/scripts/  
root@solaris:~#
```

3. ファイル `/etc/opt/SUNWsamfs/defaults.conf` をテキストエディタで開きます。ディレクティブ `exported_media` を検索します。必要に応じて行のコメントを解除するか行を追加して、値を `exported_media=available` に設定します。

この例では、`vi` エディタを使用します。

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

4. ファイル `/etc/opt/SUNWsamfs/defaults.conf` で、ディレクティブ `attended` を検索します。必要に応じて、行のコメントを解除するか行を追加します。値を `attended=yes` に設定します。ファイルを保存して、エディタを閉じます。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf  
# These are the defaults. ...  
# These are the defaults. ...  
exported_media=available  
attended=yes  
:wq  
root@solaris:~#
```

5. ファイル `/etc/opt/SUNWsamfs/scripts/load_notify.sh` をテキストエディタで開きます。通知の電子メールのデフォルトの受信者である `root` を検索します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/scripts/load_notify.sh
#       Notify operator to load volume.
...
# Change the email address on the following line to send email to
# the appropriate recipient.
/bin/ppriv -s I=basic -e /usr/bin/mailx -s "SAM-FS needs VSN $5" root <<EOF
...
```

6. ファイル `/etc/opt/SUNWsamfs/scripts/load_notify.sh` で、通知の電子メールの受信者をデフォルトの `root` から、非常駐ボリュームの担当オペレータの電子メールアドレスに変更します。ファイルを保存して、エディタを閉じます。

次の例では、受信者を `tapetech` に変更します。

```
#       Notify operator to load volume.
...
/bin/ppriv -s I=basic -e /usr/bin/mailx -s "SAM-FS needs VSN $5" tapetech <<EOF
...
:wq
root@solaris:~#
```

7. Oracle HSM ソフトウェアを再初期化します。 `sam-fsd` コマンドを使用します。

`sam-fsd` は、Oracle HSM 構成ファイルを読み取る初期化コマンドです。エラーを検出すると停止します。

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

8. `mcf` ファイルを再度読み取り、それに従ってファイルシステムとハードウェアを再構成するよう Oracle HSM ソフトウェアに指示します。コマンド `samd config` を使用します。

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

9. ここで停止します。

テープドライブのクリーニング

最新の Oracle StorageTek T10000D および Linear Tape Open (LTO) テープドライブは、自己モニターして、必要に応じてクリーニングを要求します。Oracle Hierarchical Storage Manager software はこれらの要求を受け入れ、必要に応じてクリーニングカートリッジを自動的にロードします。そのため、ほとんどの場合、ライブラリに十分なクリーニングカートリッジが含まれていること、および Oracle HSM がこれらのカートリッジを見つけることができることのみを保証する必要があります。

ドライブが要求したクリーニングが実行できない場合は、手動でクリーニングを開始できます。ただし、ほとんどの製造元では、ドライブからの要求がないときの定期的なクリーニングは行わないよう強く推奨しています。クリーニングカートリッジは研磨式です。使いすぎるとドライブおよびメディアが損傷する可能性があります。そのため、注意して、製造元の推奨事項に従ってください。

このセクションの残りの部分では、次のタスクの手順について説明します。

- [十分なクリーニングカートリッジの提供](#)
- [自動テープドライブクリーニングの有効化 \(推奨\)](#)
- [テープドライブの手動によるクリーニング](#)。

十分なクリーニングカートリッジの提供

1. ファイルシステムホストに *root* としてログインします。

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

2. 自動クリーニングを構成する計画がある場合 (推奨)、およびライブラリに複数のドライブがある場合、ライブラリ内のテープを一覧表示するファイルシステムカタログごとに必ず少なくとも2つのクリーニングカートリッジを用意してください。

ドライブでクリーニングが必要なときにクリーニングカートリッジが使用不可な場合、クリーニングが完了するまで、Oracle HSM ソフトウェアはドライブ状態を *down* に設定します。

3. ライブラリメールスロット (カートリッジアクセスポートとも呼ばれる) にクリーニングカートリッジを入れます。

4. クリーニングカートリッジを自動ライブラリにインポートします。コマンド `samimport library-equipment-number` を使用します。ここで `library-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルがライブラリに割り当てる装置の順序番号です。

次の例では、クリーニングカートリッジをライブラリ 800 のメールスロットに入れて、ライブラリにインポートします。

```
root@solaris:~# samimport 800
```

5. クリーニングカートリッジラベルが `CLEAN` であるか、文字 `CLN` で始まる場合、ここで停止します。

Oracle HSM ソフトウェアは、クリーニングカートリッジを認識して、メールボックスからストレージスロットに移動します。Oracle HSM は、ライブラリカタログを更新し、クリーニングメディアフラグを設定して、アクセス数をメディアタイプで推奨される最大クリーニング数に設定します (カートリッジを使用してドライブをクリーニングするたびに、この数は減ります)。

6. カートリッジにラベルが付いていない場合、クリーニングメディアとしてのフラグを付けます。コマンド `chmed +C library-equipment-number:slot` を使用します。ここで `library-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルがライブラリに割り当てる装置の順序番号で、`slot` は、ライブラリ内でのクリーニングカートリッジの場所です。

次の例では、ライブラリ 800 のスロット 31 内のカートリッジで `c` (クリーニングメディア) フラグを設定します。

```
root@solaris:~# chmed +C 800:31
```

7. アクセス数を、メディアタイプ用に推奨されたクリーニングの最大数に設定します。コマンド `chmed -count cleanings library-equipment-number:slot` を使用します。ここでは:
 - `cleanings` は、製造元が推奨するカートリッジあたりのクリーニングの最大回数です。
 - `library-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルがライブラリに割り当てる装置の順序番号です。

- `slot` は、ライブラリ内のクリーニングカートリッジの場所です。

カートリッジを使用してドライブをクリーニングするたびに、クリーニング数は減ります。次の例では、この数を、ライブラリ `800` で使用される LTO (タイプ `1i`) クリーニングカートリッジで推奨される最大クリーニング数である `50` に設定します。

```
root@solaris:~# chmed -count 50 800:31
```

8. 次に、自動テープドライブクリーニングを有効にする (推奨) か、ここで停止します。

自動テープドライブクリーニングの有効化 (推奨)

1. ファイルシステムホストに `root` としてログインします。

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

2. 使用を希望する自動クリーニング機能がライブラリに含まれている場合、ライブラリの製造元の推奨事項に従ってこの機能を構成します。ここで停止します。

次に、ドライブがクリーニングを要求したときに、ライブラリは必要なクリーニングメディアを自動的に提供します。

3. 使用を希望しない自動クリーニング機能がライブラリに含まれている場合、製造元の推奨事項に従ってこの機能を無効にします。
4. テキストエディタでファイル `/etc/opt/SUNWsamfs/defaults.conf` を開き、Oracle HSM の自動クリーニングを有効にします。行 `tapeclean = all autoclean on logsense on` を追加します。その後、ファイルを保存してエディタを閉じます。

この例では、`vi` エディタを使用します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults. ...
#sef = all on once
...
tapeclean = all autoclean on logsense on
:wq
```

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

5. Oracle HSM ソフトウェアを再初期化します。 *sam-fsd* コマンドを使用します。

sam-fsd は、Oracle HSM 構成ファイルを読み取る初期化コマンドです。エラーを検出すると停止します。

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

6. *mcf* ファイルを再度読み取り、それに従ってファイルシステムとハードウェアを再構成するよう Oracle HSM ソフトウェアに指示します。コマンド *samd config* を使用します。

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

7. ここで停止します。

テープドライブの手動によるクリーニング

1. ドライブの製造元のガイドラインで手動によるクリーニングについて確認してから、続行してください。

注意してください。過度に頻繁なクリーニングは、ドライブが破損する一般的な原因です。現在、多くの製造元が定期的またはスケジュールされたクリーニングをやめることを強く推奨しています。そのため、ドライブのクリーニングがいつ必要かを把握してください。

2. ドライブでクリーニングが必要なことが示されているかどうかデバイスログをモニターします。 *drive-equipment-number* ごとに1つのログがディレクトリー */var/opt/SUNWsamfs/devlog/* 内にあります。ここで *drive-equipment-number* は、 */etc/opt/SUNWsamfs/mcf* ファイルがドライブに割り当てる装置の順序番号です。
3. システムログファイル */var/adm/messages* にデバイスエラーがないかモニターします。
4. テープドライブをクリーニングします。コマンド *cleandrive drive-equipment-number* を使用します。

次の例では、ドライブ *802* をクリーニングします。


```
root@solaris:~# cleandrive 802
```

5. ここで停止します。

暗号化機能のあるドライブの使用

暗号化機能のあるドライブにファイルをアーカイブする場合、アーカイブ操作の計画時に次の点を考慮してください。

- 暗号化機能のあるドライブと暗号化機能のないドライブを、ライブラリ内に混在させないでください。
- ドライブの暗号化機能を有効にすると、暗号化を無効にできなくなります。
- 1つのテープボリューム上に、暗号化したファイルと暗号化していないファイルを混在させないでください。
- 暗号化機能が有効なドライブで、暗号化されていないデータを含むテープボリュームに暗号化されたファイルは追加できません。
- 暗号化機能が有効なドライブで、暗号化されていないデータを読み取ることはできません。

詳細は、ドライブおよび暗号化鍵管理システムのドキュメントを参照してください。

ドライブの問題の処理

通常、ベンダーの推奨事項に従ってドライブの問題を処理します。ただし、ドライブの保守、トラブルシューティング、または修復を始める前に、次のタスクの一方または両方を実行する必要がある場合があります。

- [保守または修復のためにドライブをオフラインにする](#)
- [ドライブの問題のあとでメディアをライブラリに戻す](#)

保守または修復のためにドライブをオフラインにする

1. ファイルシステムホストに `root` としてログインします。

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

2. 「[アーカイブおよびステージングプロセスのアイドル状態](#)」の説明に従って、アーカイブおよびステージングプロセスを休止します。

3. アーカイブおよびステージングプロセスを停止し、ドライブをオフラインにします。「[アーカイブおよびステージングプロセスの停止](#)」の手順を使用します。
4. ベンダー指定の保守、診断、または修復手順を実行します。

たとえば、スタック状態のカートリッジの削除を試行する前に、必ずベンダーの推奨事項を確認してください。スタック状態のカートリッジを不適切に削除すると、カートリッジとドライブが破損する可能性があります。

5. ドライブが再び操作可能になったら、ライブラリおよびドライブをオンラインにし、アーカイブおよびステージングプロセスを再開します。「[アーカイブおよびステージングプロセスの再起動](#)」の手順を使用します。
6. ここで停止します。

ドライブの問題のあとでメディアをライブラリに戻す

ドライブにマウントされたメディアでドライブの問題が発生した場合、修復プロセスの一環としてメディアを手動で削除する必要があります。これによって、カタログが不整合な状態になる可能性があります。そのため、次の適切な手順に従います。

自動監査を実行していないライブラリにメディアを戻す

修復後、ライブラリとドライブがオンラインに戻ったときに、自動監査を実行しないライブラリにメディアを戻す場合、次の手順を実行します。

1. カートリッジをそのカートリッジのストレージスロットに手動で戻します。

この場合、カタログは更新されず、ライブラリのコンテンツ間でカートリッジが一覧表示されたままになります。そのため、前に占有していたのと同じスロットにカートリッジを戻すことで不一致を修正します。

2. スロットが再度占有されていることを示すために、Oracle HSM カタログを更新します。コマンド `chmed library-equipment-number:slot` を使用します。ここで `slot` は、ライブラリ内でのスロットのアドレスです。

次の例では、ライブラリ `800` のスロット `42` のステータスを更新します。

```
root@solaris:~# chmed +o 800:42
root@solaris:~#
```

3. ここで停止します。

自動監査後のライブラリへのメディアの返却

修復後、ライブラリとドライブがオンラインに戻ったときに、ライブラリが自動監査を実行する場合、次の手順を実行します。

1. カートリッジをライブラリメールスロットに入れます。
2. カートリッジをライブラリにインポートします。コマンド `samimport library-equipment-number` を使用します。

この場合、監査によってカタログが調整され、ライブラリ内のカートリッジは一覧表示されなくなります。そのため、カートリッジをインポートすると、ライブラリと Oracle HSM カatalogの両方に追加されます。次の例では、カートリッジをライブラリ 800 のメールスロットに入れて、ライブラリにインポートしました。

```
root@solaris:~# samimport 800
```

3. ここで停止します。

リムーバブルメディアの管理

このセクションでは、次のトピックを扱います。

- [リムーバブルメディアのラベル付け](#)
- [データの整合性の維持](#)

リムーバブルメディアのラベル付け

注意:

カートリッジのラベル付けまたは再ラベル付けを行うと、カートリッジのデータが永続的にアクセス不能になります。カートリッジに再ラベル付けするのは、カートリッジに保管されているデータが不要であることを確認した場合のみにしてください。

ラベル付けプロセスによって、記録メディアに識別情報が書き込まれ、使用のために初期化されます (詳細は、ANSI X3.27-1987 で情報交換のための磁気テープのファイル構造およびラベル付けについて参照してください)。

メディアのラベル付けが必要な場合には、次から適切な手順を選択します。

- [バーコードからのラベルの生成](#)

- 新しいテープのラベル付けまたは既存のテープの再ラベル付け
- 新しい光磁気ディスクのラベル付けまたは既存の光磁気ディスクの再ラベル付け。

バーコードからのラベルの生成

カートリッジのバーコードから生成したボリュームシリアル番号 (VSN) の付いた、ラベルのない書き込み可能なカートリッジに自動的にラベルを付けるには、次の手順を実行します。

1. すべてのバーコードが読み取り可能であることを確認します。
2. ファイルシステムホストに *root* としてログインします。

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

3. */etc/opt/SUNWsamfs/defaults.conf* ファイルをテキストエディタで開きます。

この例では、ファイルの表示に *vi* エディタを使用します。

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

4. 対応するバーコードの最初の 6 文字からボリュームシリアル番号 (VSN) を生成する必要がある場合、まず、Oracle HSM がデフォルト値である *barcodes* に設定されているかどうかを確認します。 *defaults.conf* ファイルで、 *labels* ディレクティブの行が存在する場合はこれを見つけます。 *labels* ディレクティブが *barcodes* に設定されているか、コメントアウトされているか、ファイル内に存在しない場合は、Oracle HSM がデフォルト値である *barcodes* に設定されています。

この例では、 *defaults.conf* ファイルに行 *#labels = barcodes* が含まれます。

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

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

5. 対応するバーコードの最初の 6 文字からボリュームシリアル番号 (VSN) を生成する必要があり、Oracle HSM がデフォルト値に設定されている場合は、何も変更せずに *defaults.conf* ファイルを閉じます。ここで停止します。

labels ディレクティブが *barcodes* に設定されている場合、ソフトウェアが自動的に、対応するバーコードの最初の 6 文字から必要なボリュームシリアル番号 (VSN) を生成します。この例では、Oracle HSM はデフォルト設定を使用しています。そのため、ファイルを保存せずに *vi* エディタを閉じます。

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

6. それ以外の場合で、対応するバーコードの最初の 6 文字からボリュームシリアル番号 (VSN) を生成する必要がある場合は、*labels = barcodes* を入力するか、デフォルト以外のディレクティブをコメントアウトするか、単にデフォルト以外のディレクティブを削除します。次に、ファイルを保存し、エディタを閉じます。

この例では、ディレクティブがデフォルト値以外の *barcodes_low* に設定されています。そのため、このデフォルト以外の行をコメントアウトします。行 *labels = barcodes* を挿入します。修正したファイルを保存して、エディタを終了します。

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

7. カートリッジのバーコードの最後の 6 文字からボリュームシリアル番号 (VSN) を生成する必要がある場合、`labels` パラメータの値を `barcodes_low` に設定します。ファイルを保存して、エディタを閉じます。

この例では、行 `labels = barcodes_low` を挿入し、ファイルを保存し、エディタを閉じます。

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

8. `defaults.conf` ファイルを編集した場合、`sam-fsd` コマンドを実行します。

`sam-fsd` は、Oracle HSM 構成ファイルを読み取る初期化コマンドです。エラーを検出すると停止します。

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

9. `defaults.conf` ファイルを編集した場合、`mcf` ファイルを再度読み取り、それに従って Oracle HSM software 自体を再構成するようにこのソフトウェアに指示します。コマンド `samd config` を使用します。

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

10. ここで停止します。

新しいテープのラベル付けまたは既存のテープの再ラベル付け

注意:

カートリッジのラベル付けまたは再ラベル付けを行うと、カートリッジのデータが永続的にアクセス不能になります。カートリッジに再ラベル付けするのは、カートリッジに保管されているデータが不要であることを確認した場合のみにしてください。

1. ファイルシステムホストに `root` としてログインします。

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

2. ドライブにすでにロードされている新しいテープにラベルを付けるには、コマンド `tplabel -new volume-serial-number drive-equipment-number` を使用します。ここでは:

- `volume-serial-number` は必要とされるボリュームシリアル番号です。
- `drive-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルがドライブに割り当てる装置の順序番号です。

次の例では、ボリュームシリアル番号 `VOL600` をドライブ `803` 内の新しいテープカートリッジに割り当てます。

```
root@solaris:~# tplabel -new -vsn VOL600 803
root@solaris:~#
```

3. 自動メディアライブラリにある新しいテープにラベルを付けるには、コマンド `tplabel -new volume-serial-number library-equipment-number:slot` を使用します。ここでは:

- `volume-serial-number` は必要とされるボリュームシリアル番号です。
- `library-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルがドライブに割り当てる装置の順序番号です。
- `slot` は、ライブラリ内のカートリッジの場所です。

次の例では、ボリュームシリアル番号 `VOL601` をライブラリ `800` のスロット `19` 内の新しいテープカートリッジに割り当てます。

```
root@solaris:~# tplabel -new -vsn VOL601 800:19
root@solaris:~#
```

4. ドライブにロードされているテープに再度ラベルを付けるには、コマンド `tplabel -old old-volume-serial-number -new new-volume-serial-number drive-equipment-number` を使用します。ここでは:

- `volume-serial-number` は必要とされるボリュームシリアル番号です。
- `drive-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルがドライブに割り当てる装置の順序番号です。

この例では、ドライブ *804* 内のテープカートリッジを再初期化し、古いボリュームシリアル番号 *AZ0001* を新しいボリュームシリアル番号 *VOL120* に置き換えます。

```
root@solaris:~# tplabel -old AZ0001 -vsn VOL120 804
root@solaris:~#
```

5. テープライブラリにあるテープに再度ラベルを付けるには、コマンド `tplabel -old old-volume-serial-number -new new-volume-serial-number library-equipment-number:slot` を使用します。ここでは:
 - `volume-serial-number` は必要とされるボリュームシリアル番号です。
 - `library-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルがドライブに割り当てる装置の順序番号です。
 - `slot` は、ライブラリ内のカートリッジの場所です。

必要に応じて、既存のボリュームシリアル番号を再利用できます。この例では、ライブラリ *800* のスロット *23* 内のテープカートリッジに、既存のボリュームシリアル番号 *VOL121* で再度ラベルを付けてこれを再初期化します。

```
root@solaris:~# tplabel -old VOL601 -vsn VOL601 800:23
root@solaris:~#
```

6. ここで停止します。

新しい光磁気ディスクのラベル付けまたは既存の光磁気ディスクの再ラベル付け

1. ファイルシステムホストに `root` としてログインします。

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

2. ドライブにロードされた新しい光磁気カートリッジにラベルを付けるには、コマンド `odlabel -new volume-serial-number drive-equipment-number[:side]` を使用します。ここでは:
 - `volume-serial-number` は必要とされるボリュームシリアル番号です。

- *drive-equipment-number* は、`/etc/opt/SUNWsamfs/mcf` ファイルがドライブに割り当てる装置の順序番号です。
- *side* (オプション) は両面ディスクの指定された面です。

次の例では、ボリュームシリアル番号 *OD1700* をドライブ *701* 内の新しい片面光磁気カートリッジに割り当てます。

```
root@solaris:~# odlabel -new -vsn OD1700 701
root@solaris:~#
```

3. 自動メディアライブラリにある新しい光磁気カートリッジにラベルを付けるには、コマンド `odlabel -new volume-serial-number library-equipment-number:slot[:side]` を使用します。ここでは:

- *volume-serial-number* は必要とされるボリュームシリアル番号です。
- *library-equipment-number* は、`/etc/opt/SUNWsamfs/mcf` ファイルがドライブに割り当てる装置の順序番号です。
- *slot* はライブラリ内のカートリッジの場所で、*side* (オプション) は両面ディスクの指定された面です。

次の例では、ボリュームシリアル番号 *OD1701* をライブラリ *700* のスロット *42* 内にある新しい両面光磁気カートリッジのサイド *2* に割り当てます。

```
root@solaris:~# odlabel -new -vsn OD1701 700:42:2
root@solaris:~#
```

4. ドライブにロードされている光磁気カートリッジに再度ラベルを付けるには、コマンド `odlabel -old old-volume-serial-number -new new-volume-serial-number drive-equipment-number[:side]` を使用します。ここでは:

- *volume-serial-number* は必要とされるボリュームシリアル番号です。
- *drive-equipment-number* は、`/etc/opt/SUNWsamfs/mcf` ファイルがドライブに割り当てる装置の順序番号です。
- *side* (オプション) は両面ディスクの指定された面です。

この例では、ドライブ *702* 内の光磁気カートリッジを再初期化し、古いボリュームシリアル番号 *OD1120* を新しいボリュームシリアル番号 *OD1120* に置き換えます。

```
root@solaris:~# odlabel -old OD0001 -vsn OD1120 702
root@solaris:~#
```

5. 自動メディアライブラリにある既存の光磁気カートリッジに再度ラベルを付けるには、コマンド `odlabel -old volume-serial-number library-equipment-number:slot[:side]` を使用します。ここでは:
 - `volume-serial-number` は必要とされるボリュームシリアル番号です。
 - `library-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルがライブラリに割り当てる装置の順序番号です。
 - `side` (オプション) は両面ディスクの指定された面です。

必要に応じて、既存のボリュームシリアル番号を再利用できます。この例では、ライブラリ 700 のスロット 23 内の光磁気カートリッジに、既存のボリュームシリアル番号 `OD1121` で再度ラベルを付けてこれを再初期化します。

```
root@solaris:~# odlabel -old OD1121 -vsn OD1121 800:23
root@solaris:~#
```

6. ここで停止します。

データの整合性の維持

Oracle Hierarchical Storage Manager ソフトウェアは、リムーバブルテープメディア上に格納されているデータファイルの整合性を維持するために、オンデマンドツールと自動ツールの両方を提供します。このセクションでは、次のトピックについて説明します。

- [Data Integrity Validation \(DIV\) 設定およびステータスの表示](#)
- [特定のテープボリュームの整合性のチェック](#)
- [自動整合性検証のモニタリング](#)

Data Integrity Validation (DIV) 設定およびステータスの表示

このセクションでは、次のタスクを扱います。

- [DIV 設定の表示](#)
- [アーカイブファイルの書き込みステータス後の検証のモニター](#)

- デバイスの書き込みステータス後の検証のモニター。

DIV 設定の表示

Data Integrity Validation (DIV) 設定を表示するには、コマンド `samcmd L` を使用して、出力を Solaris `grep` コマンドおよび正規表現 `div` にパイプします。

次の例では、DIV は `OFF` です。

```
root@solaris:~# samcmd L | grep div
div                OFF
root@solaris:~#
```

アーカイブファイルの書き込みステータス後の検証のモニター

アーカイブ処理中のアーカイブファイルの検証ステータスをモニターするには、`samu` インタフェースを使用します。コマンド `samu -d a` を使用します。

```
root@solaris:~# samu -d a
Archiver status          samu 5.4          22:22:31 Mar 4 2014
sam-archiverd: Archiving files
sam-arfind: hsmfs1 mounted at /hsm/hsmfs1
Files waiting to start   12,576 schedule   26,695 archiving   13,120
...
sam-arcopy: qfs.arset1.2.344 ti.TKC960
Verifying archive file at position 1175
```

デバイスの書き込みステータス後の検証のモニター

アーカイブ処理中のデバイスの検証ステータスをモニターするには、`samu` インタフェースを使用します。コマンド `samu -d s` を使用します。

```
root@solaris:~# samu -d s
Device status          samu 5.4          22:27:53 Mar 4 2014
ty    eq state  device_name          fs  status
sn    800 on    /dev/scsi/changer/c1t2d0  800 n-----r
ti    801 on    /dev/rmt/0cbn         800 -----p
...
hy    805 on    historian              805 -----
ti    91 on    /dev/rmt/4cbn         90  -l----oVr
Verify averaging 240.9 MB/s
```

特定のテープボリュームの整合性のチェック

特定のテープボリュームのデータ整合性を検証する必要がある場合は、Oracle HSM `tpverify` コマンドを使用します。`tpverify` コマンドでは、Oracle T10000C/D、LTO、およびその他の一般的に使用されるメディアがサポートされま

す。T10000C/D メディアは、Oracle Data Integrity Validation を使用して検証されます。その他の形式は、幅広くサポートされる `SCSI verify(6)` コマンドを使用して検査されます。

次のセクションでは、`tpverify` のいくつかの使用法の概要について説明します。構文およびオプションの詳細は、`tpverify` のマニュアルページを参照してください。

- ライブラリ位置で指定されたテープ上のデータの検証
- メディアタイプおよびボリュームシリアル番号で指定されたテープ上のデータの検証
- 指定したドライブを使用したテープ上のデータの検証
- テープの先頭からデータの検証を再開
- T10000C/D テープ上にあるすべてのブロックの ECC の検証
- T10000C/D テープ上にあるすべてのブロックの ECC および DIV チェックサムの検証
- T10000C/D テープのメディア情報領域 (MIR) の再構築
- 指定したテープのデータ検証のキャンセル
- テープの DIV ステータスおよび検証の進行状況の表示

ライブラリ位置で指定されたテープ上のデータの検証

コマンド `tpverify library-equipment-number:slot` を使用します。ここで `library-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルが自動テープライブラリに割り当てる装置の順序番号で、`slot` は、ターゲットボリュームがあるライブラリ内のスロットアドレスです。

`tpverify` コマンドは、ライブラリメディアカタログを調べることで検証された最後のテープ位置を見つけます。次に、最初に使用可能なドライブにテープをロードし、デフォルトの方式 (T10000C/D メディアの場合は `tpverify` 標準方式、その他のメディアの場合は `SCSI verify(6)`) を使用して、最後に停止したポイントから検証を開始します。標準方式は、速度に合わせて最適化されており、Oracle HSM メディアの端、先頭、終わり、および最初の 1,000 ブロックを検証します。

次の例では、標準方式を使用して、ライブラリ `800` 上のスロット `9` に保管されている T10000D テープを検証します。

```
root@solaris:~# tpverify 800:9
```

メディアタイプおよびボリュームシリアル番号で指定されたテープ上のデータの検証

コマンド `tpverify mediatype.volume-serial-number` を使用します。ここで `mediatype` は、付録A「装置タイプの用語集」に一覧表示されている2文字のメディアタイプコードの1つで、`volume-serial-number` は、ライブラリ内のボリュームを一意に識別する英数字の文字列です。

`tpverify` コマンドは、ライブラリメディアカタログを調べることで検証された最後のテープ位置を見つけます。次に、最初に使用可能なドライブにテープをロードし、デフォルトの方式 (T10000C/D メディアの場合は `tpverify` 標準方式、その他のメディアの場合は SCSI `verify(6)`) を使用して、最後に停止したポイントから検証を開始します。

次の例では、SCSI `verify(6)` コマンドを使用して LTO (**li**) ボリューム **VOL006** を検証します。

```
root@solaris:~# tpverify li.VOL006
```

指定したドライブを使用したテープ上のデータの検証

コマンド `tpverify library-equipment-number:slot device-equipment-number` を使用します。ここでは:

- `library-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルが自動テープライブラリに割り当てる装置の順序番号です。
- `slot` は、ターゲットボリュームがあるライブラリ内のスロットアドレスです。
- `device-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルがドライブに割り当てる装置の順序番号です。

次の例では、ドライブ `803` を使用してライブラリ `800` 上のスロット `17` に保管されている T10000D テープを検証します。

```
root@solaris:~# tpverify 800:17 803
```

テープの先頭からデータの検証を再開

コマンド `tpverify -a library-equipment-number:slot` または `tpverify -a mediatype.volume-serial-number` を使用します。ここでは:

- `library-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルが自動テープライブラリに割り当てる装置の順序番号です。
- `slot` は、ターゲットボリュームがあるライブラリ内のスロットアドレスです。
- `mediatype` は、付録A「装置タイプの用語集」に一覧表示されている2文字のメディアタイプコードの1つです。
- `volume-serial-number` は、ライブラリ内のボリュームを一意に識別する英数字の文字列です。

`-a` オプションは、デフォルトの動作をオーバーライドし、メディアの先頭から検証を開始して、メディアカタログ内に記録された位置は無視します。

次の例では、テープの先頭から LTO (*li*) ボリューム `VOL016` を検証します。

```
root@solaris:~# tpverify -a li.VOL016
```

T10000C/D テープ上にあるすべてのブロックの ECC の検証

コマンド `tpverify -C library-equipment-number:slot` または `tpverify -C mediatype.volume-serial-number` を使用します。ここでは:

- `library-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルが自動テープライブラリに割り当てる装置の順序番号です。
- `slot` は、ターゲットボリュームがあるライブラリ内のスロットアドレスです。
- `mediatype` は、付録A「装置タイプの用語集」に一覧表示されている2文字のメディアタイプコードの1つです。
- `volume-serial-number` は、ライブラリ内のボリュームを一意に識別する英数字の文字列です。

`tpverify` コマンドは、ライブラリメディアカタログを調べることで検証された最後のテープ位置を見つけます。次に、`-C` オプションで指定された完全方式を使用して、最後に停止したポイントから検証を開始します。完全方式は標準方式よりも完全ですが、大幅に遅くなる可能性があります。メディア上のすべてのブロックでエラー訂正コード (ECC) を検査します。

次の例では、完全方式を使用して T10000D (*ti*) ボリューム `VOL516` を検証します。

```
root@solaris:~# tpverify -C ti.VOL516
```

T10000C/D テープ上にあるすべてのブロックの ECC および DIV チェックサムの検証

コマンド `tpverify -P library-equipment-number:slot` または `tpverify -P mediatype.volume-serial-number` を使用します。ここでは:

- `library-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルが自動テープライブラリに割り当てる装置の順序番号です。
- `slot` は、ターゲットボリュームがあるライブラリ内のスロットアドレスです。
- `mediatype` は、付録A「装置タイプの用語集」に一覧表示されている 2 文字のメディアタイプコードの 1 つです。
- `volume-serial-number` は、ライブラリ内のボリュームを一意に識別する英数字の文字列です。

`tpverify` コマンドは、ライブラリメディアカタログを調べることで検証された最後のテープ位置を見つけます。次に、`-P` オプションで指定された完全プラス方式を使用して、最後に停止したポイントから検証を開始します。完全プラス方式はきわめて完全ですが、ほかの方法よりも処理速度が遅くなります。これは、メディア上のすべてのブロックでエラー訂正コード (ECC) および Data Integrity Validation チェックサムを確認します。

次の例では、完全プラス方式を使用して T10000D (`ti`) ボリューム `VOL521` を検証します。

```
root@solaris:~# tpverify -P ti.VOL521
```

T10000C/D テープのメディア情報領域 (MIR) の再構築

コマンド `tpverify -M library-equipment-number:slot` または `tpverify -M mediatype.volume-serial-number` を使用します。ここでは:

- `library-equipment-number` は、`/etc/opt/SUNWsamfs/mcf` ファイルが自動テープライブラリに割り当てる装置の順序番号です。
- `slot` は、ターゲットボリュームがあるライブラリ内のスロットアドレスです。
- `mediatype` は、付録A「装置タイプの用語集」に一覧表示されている 2 文字のメディアタイプコードの 1 つです。

- *volume-serial-number* は、ライブラリ内のボリュームを一意に識別する英数字の文字列です。

テープがメディアカタログで不良とマークされている場合でも、*tpverify* コマンドは、Oracle StorageTek テープカートリッジで欠落または破損しているメディア情報領域 (MIR) を再構築します。再構築は、MIR の破損が検出された際に自動的に指定されます。

次の例では、MIR 再構築方式を使用して T1000D (*ti*) ボリューム *VOL523* を検証します。

```
root@solaris:~# tpverify -M ti.VOL523
```

指定したテープのデータ検証のキャンセル

コマンド *tpverify -c library-equipment-number:slot* または *tpverify -c mediatype.volume-serial-number* を使用します。ここでは:

- *library-equipment-number* は、*/etc/opt/SUNWsamfs/mcf* ファイルが自動テープライブラリに割り当てる装置の順序番号です。
- *slot* は、ターゲットボリュームがあるライブラリ内のスロットアドレスです。
- *mediatype* は、[付録A「装置タイプの用語集」](#) に一覧表示されている 2 文字のメディアタイプコードの 1 つです。
- *volume-serial-number* は、ライブラリ内のボリュームを一意に識別する英数字の文字列です。

tpverify -c コマンドは、現在の検証操作をキャンセルして、テープで最後に検証した位置をメディアカタログに記録します。そのため、検証ジョブを停止して、ドライブまたはボリュームをアーカイブ処理またはステージング処理のために解放し、あとで同じポイントで検証を再開できます。

次の例では、T1000D (*ti*) ボリューム *VOL533* の検証をキャンセルします。

```
root@solaris:~# tpverify -c ti.VOL533
```

テープの DIV ステータスおよび検証の進行状況の表示

コマンド *itemize -2 library-equipment-number* を使用します。*library-equipment-number* は、*/etc/opt/SUNWsamfs/mcf* ファイルが自動テープライブラリに割り当てる装置の順序番号です。

`itemize -2` コマンドは、指定されたライブラリでメディアをカタログし、各ボリュームの DIV ステータスと検証の進行状況を一覧表示します。

次の例では、装置の順序番号が `800` のライブラリ内のボリュームの検証ステータスを表示します。 `lvtime` (最後に検証された時間) フィールドには、 `tpverify` がテープの完全な検証を最後に完了した時間が表示されます。 `div` の `status` フィールド値は、テープが DIV 対応であることを示すのに対し、値 `none` は対応していないことを示します。 `lvpos` (最後に検証された位置) フィールドは、 `tpverify` が最後にキャンセルされた場所および次の実行時に開始する場所を示します。

```
root@solaris:~# itemize -2 800
Robot VSN catalog: eq: 800          count: 60
slot  access_time count use ty vsn
      lvtime      status          lvpos
  0    Apr  2 16:34   6  0% ti VOL519
      Apr  2 09:23   div          0
  1    Apr  2 16:17  28 29% ti VOL510
      Apr  2 16:17   div          0x9bb9
  2    none          0   0% ti VOL511
      none          none          0
...
root@solaris:~#
```

自動整合性検証のモニタリング

定期的なメディア検証は、 `tpverify` コマンドの自動化された形式です。このセクションでは、場合によっては必要になることがある保守タスクの手順について説明します。これらのタスクには次が含まれます。

- **verifyd.cmd** 構成ファイルの表示および検証
- **verifyd.cmd** 構成ファイルのリロード
- 定期的なメディア検証のテープの不具合データベースに一覧表示されたすべての不具合の表示
- 特定のボリュームに対して一覧表示された不具合の表示
- 定期的なメディア検証のテープの不具合データベースに一覧表示された不具合のクリア。

定期的なメディア検証を構成する手順については、お客様向けドキュメントライブラリ (<http://docs.oracle.com/en/storage/#sw>) の『Oracle Hierarchical Storage Manager and StorageTek QFS インストールおよび構成ガイド』を参照してください。

verifyd.cmd 構成ファイルの表示および検証

任意の時点で *verifyd.cmd* ファイルを表示するには、または編集後にファイルを検証するには、コマンド *tpverify -x* を使用します。

tpverify -x コマンドは、*/etc/opt/SUNWsamfs/verifyd.cmd* ファイルを検査して、エラーを呼び出すかファイルの内容を表示します。

```
root@solaris:~# tpverify -x
Reading '/etc/opt/SUNWsamfs/verifyd.cmd'.
PMV: off
  Run-time:
  Start Time: 2200
End Time: 0500
PMV Scan: all
PMV Method: Standard
STA Scan: off
Action: none
PMV VSNS: all
PMV Policy:
  Last Verified Age: 6m
root@solaris:~#
```

verifyd.cmd 構成ファイルのリロード

検証プロセスを停止せずに *verifyd.cmd* ファイルをリロードするには、コマンド *tpverify -r* を使用します。

```
root@solaris:~# tpverify -r
root@solaris:~#
```

定期的なメディア検証のテープの不具合データベースに一覧表示されたすべての不具合の表示

定期的なメディア検証によって特定され、テープの不具合データベースに保管されたすべての不具合を一覧表示するには、コマンド *tpverify -l* を使用します。

次の例では、データベースに不具合はありません。

```
root@solaris:~# tpverify -l
No defects found.
root@solaris:~#
```

特定のボリュームに対して一覧表示された不具合の表示

特定のボリュームで識別されたすべての不具合を一覧表示するには、コマンド *tpverify -l mediatype.volume-serial-number* を使用します。ここでは:

- *mediatype* (オプション) は、付録A「装置タイプの用語集」に一覧表示されている2文字のメディアタイプコードの1つです。
- *volume-serial-number* は、ライブラリ内のボリュームを一意に識別する英数字の文字列です。

次の例では、LTO (*ti*) ボリューム *VOL514* 用のデータベースに一覧表示された不具合はありません。

```
root@solaris:~# tpverify -l ti.VOL514
No defects found.
root@solaris:~#
```

定期的なメディア検証のテープの不具合データベースに一覧表示された不具合のクリア

定期的なメディア検証によって特定されたすべての不具合をテープの不具合データベースから削除するには、コマンド *tpverify -d* を使用します。

特定のボリュームで一覧表示されたすべての不具合を削除するには、コマンド *tpverify -d mediatype.volume-serial-number* を使用します。ここでは:

- *mediatype* (オプション) は、付録A「装置タイプの用語集」に一覧表示されている2文字のメディアタイプコードの1つです。
- *volume-serial-number* は、ライブラリ内のボリュームを一意に識別する英数字の文字列です。

```
root@solaris:~# tpverify -d
root@solaris:~# tpverify -d ti.VOL514
root@solaris:~#
```


第6章 デジタル保存のためのアーカイブの管理

ここまでこのドキュメントでは、ユーザーとアプリケーションが定期的にファイルの作成、変更、および削除を行う通常の UNIX ファイルシステムとしての Oracle Hierarchical Storage Manager and StorageTek QFS Software ソリューションの管理について説明しました。主に高度に統合されたバックアップサービスとしての役割を果たすアーカイブとともに、ディスクキャッシュに焦点を当てました。この章では、長期間のデータ保存のためのリポジトリおよび管理ソリューションとしてのアーカイブに再度焦点を当てます。以前に説明した管理の原則と手法が引き続き関係します。ただし現在では、ディスクキャッシュは主に、取り込み後に削除や変更が可能ではないアーカイブにファイルを取り込む手段としての役割を果たします。

正確な要件は条件によって異なります。法で定められた期間業務記録または医療記録を保持するリポジトリは、定期的な記録の破棄が必要になる場合があります。ただし、科学データ、歴史的記録や家系の記録、デジタル音楽、映画、またはテレビ番組を格納するアーカイブは、事実上永続的な内容の格納が必要になる場合があります。このため、Oracle HSM では、複数の方法でデジタル保存をサポートします。

- メッセージダイジェスト (チェックサム) を使用すると、損害、データ破損、およびファイルに対する未承認の変更を検出できるため、ハードウェアの問題を修正して、異常なファイルを、アーカイブの別の場所に格納されている正常なコピーと置き換えることができます。
- ファイルの fixity 属性は、修正されたファイルを変更できるのがスーパーユーザーだけになるようにメッセージダイジェストを処理します。Oracle HSM は、固定ファイルをステージングまたはアーカイブするたびに、このファイルが変更されていないことを証明するために、fixity 属性とともに格納されたチェックサムを再検証します。
- Oracle HSM Write Once Read Many (WORM) ファイルシステムを使用すると、ファイルを読み取り専用にして、指定された期間の保持を適用できます。これらのファイルシステムは、スーパーユーザーがファイルや、前述した fixity 属性などのファイル属性を変更できないように構成できます。

この章では、長期間のストレージソリューションのベースとなる基本的な Oracle HSM データ保護対策を簡単に見直すことから始めます (「[保存用ファイルシステムの構成](#)」)。その後、特にデータの保存に対処するタスクについて説明します。

- [メッセージダイジェスト \(チェックサム\) の使用](#)
- [ファイルを不変にする](#)
- [WORM ファイルシステムの使用](#)

保存用ファイルシステムの構成

すべての保存ソリューションが、正常な冗長性の高いファイルシステムから始まります。そのため、付属の『Oracle Hierarchical Storage Manager and StorageTek QFS インストールおよび構成ガイド』の実装の章をまだ確認していない場合は、この章を確認します。冗長なサーバー、ネットワーク接続、およびストレージデバイスを提供することで、アーカイブへのアクセスを保護します。それぞれが独立したメディアに格納された各ファイルの追加のコピーを少なくとも 2 つ構成することで、ファイルデータを保護します。ほとんどの場合、1 つのコピーをディスクまたはソリッドステートストレージデバイスにアーカイブして、2 つのコピーをテープメディアにアーカイブすることが理想的です。可能な場合は、Oracle HSM のデータ整合性の検証機能を実装することで、テープブロックが正しく読み書きされていることを確認します。定期的にダンプファイルを生成して、アーカイブログを定期的にバックアップすることで、ファイルシステムのメタデータを保護します。

メッセージダイジェスト (チェックサム) の使用

メッセージダイジェスト (チェックサム) を使用すると、保存担当者は、段階的な劣化、ハードウェアまたはオペレータのエラー、または内容に対する故意の承認されていない変更を示す可能性がある変更についてアーカイブ済みファイルをテストできます。メッセージダイジェストは単に、一方向の暗号化ハッシュ関数によって生成されたファイルの内容の数学上のサマリーです。暗号化ハッシュ関数は、その入力データでの変更に対して非常に敏感です。入力のわずかな変更であっても、出力が大きく変更されます。そのためメッセージダイジェストは、ファイルの破損と承認されていない変更を検出するうえで理想的です。ファイルのダイジェストを再計算して、格納されているダイジェスト値と結果の値を比較することで、ファイルが変更されたかどうかを示されます。

Oracle Hierarchical Storage Manager ファイルシステムは、次のいずれかの暗号化ハッシュ関数を使用して、メッセージダイジェストの取り込み、作成、格納、および検証を行うことができます。

- 暗号化関数の Secure Hash Algorithm ファミリの 160 ビットメンバーである SHA1

Secure Hash Algorithms は、連邦情報処理標準 (*Federal Information Processing Standard, FIPS*) の資料 180-4、米国国立標準技術研究所 (2012 年) で定義されています。Oracle HSM ではデフォルトで SHA1 が使用されます。
- Secure Hash Algorithm ファミリの 256 ビットメンバーである SHA256
- Secure Hash Algorithm ファミリの 384 ビットメンバーである SHA384
- Secure Hash Algorithm ファミリの 512 ビットメンバーである SHA512。
- Request for Comment (RFC) 1321 で Internet Engineering Task Force (IETF) によって定義された 128 ビットのメッセージダイジェスト関数である MD5
- 現在では主に古い Storage Archive Manager 実装との下位互換性のために役に立つ、独自の 128 ビットの Oracle HSM 関数。

ユーザーは、ファイルがリポジトリに取り込まれたときに既存のダイジェスト値を指定することも、即時に、またはファイルが最初にアーカイブされたときに、ファイルシステムにダイジェスト値を計算させることもできます。Oracle HSM ファイルシステムは、特殊なファイル属性を使用して、ファイルシステムメタデータとともにダイジェスト値を格納します。属性の設定後、対応するファイルが再アーカイブされるたびに、またオプションでファイルがアーカイブメディアからディスクキャッシュにステージングされるたびに、ファイルシステムはダイジェストを再計算し、格納されている値と照合して検証します。

ただし、Oracle HSM メディアの移行機能は、チェックサムを再計算せずにファイルを新しいメディアにコピーします (メディアの移行については、[8章「新しいストレージメディアへの移行」](#)を参照)。そのために、ファイルが正しくコピーされなかった場合、ファイルが再ステージングされて検証されるまで、破損が検出されないリスクがわずかに発生します。データ整合性検証 (DIV) を使用することで、このリスクが最小限に抑えられます (詳細は、『Oracle Hierarchical Storage Manager and StorageTek QFS インストールおよび構成ガイド』を参照)。

メッセージダイジェストの使用を開始する前に、最初に「[ファイルシステムホストのパフォーマンスが適切であることの確認](#)」を読むようにしてください。その後、

ダイジェストの指定、生成、および検証の手順について次のセクションを参照できます。

- [メッセージダイジェストの指定およびファイルの検証の有効化](#)
- [メッセージダイジェストの生成およびファイルの検証の有効化](#)
- [メッセージダイジェストの生成およびディレクトリ内にある各ファイルの検証の有効化](#)
- [ステージング中のファイルのメッセージダイジェストの検証](#)
- [ファイルがアーカイブされる前のメッセージダイジェストと検証属性の変更](#)

ファイルシステムホストのパフォーマンスが適切であることの確認

メッセージダイジェストを大いに利用する予定がある場合、適切なパフォーマンスを実現するために十分な計算リソースがファイルシステムホストにあることを確認してください。ほとんどの最新のプラットフォームでは、中央のプロセッササイクルを消費することなく、特殊な計算を効率的に実行できる専用の暗号化ハードウェアが組み込まれています。可能な場合は、これらの機能を必ず活用してください。

潜在的なファイルシステムホストの機能をチェックするには、次のように進めます。

1. ファイルシステムホストに *root* としてログインします。

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

2. ホストオペレーティングシステムが Solaris 11.1 以上であることを確認します。コマンド *uname -v* を使用します。

以前のバージョンのオペレーティングシステムでは、ハッシュ関数のハードウェアアクセラレーションはサポートされません。この例では、ホストオペレーティングシステムは Solaris 11.2 です。

```
root@solaris:~# uname -v
11.2
root@solaris:~#
```


- 命令セットアーキテクチャーを表示します。コマンドプロンプトで、コマンド `isainfo -v` を入力します。

```
root@solaris:~# isainfo -v
```

- Solaris 11 ホストが Oracle Sun SPARC T3 以上のシステムである場合、`isainfo -v` コマンドの出力では、`sha512`、`sha256`、`sha1`、および `md5` の暗号化アルゴリズムをサポートする命令セットが一覧表示されます。

この例では、Sun SPARC T4-2 ホストは、SHA1、SHA2、および MD5 アルゴリズムファミリのハードウェアアクセラレーションを提供します。

```
[Sun_SPARC_T4-2]root@solaris:~# isainfo -v
64-bit sparcv9 applications
    crc32c cbcond pause mont mpmul sha512 sha256 sha1 md5 camellia kasumi
    des aes ima hpc vis3 fmaf asi_blk_init vis2 vis popc
root@solaris:~#
```

- Solaris ホストが x86/64 システムである場合、`isainfo -v` コマンドの出力に `ssse3` (Supplemental Streaming SIMD Extensions 3) 命令セットが含まれていれば、SHA-1 ハードウェアアクセラレーションがサポートされます。

この例では、Sun X3-2 ホストで SHA-1 ダイジェストのハードウェアアクセラレーションがサポートされます。

```
[Sun_X3-2]root@solaris:~# isainfo -v
64-bit amd64 applications
    avx xsave pclmulqdq aes sse4.2 sse4.1 ssse3 popcnt tscp ahf cx16 sse3
    sse2 sse fxsr mmx cmov amd_sysc cx8 tsc fpu
root@solaris:~#
```

メッセージダイジェストの指定およびファイルの検証の有効化

メッセージダイジェストにすでに関連付けられたファイルをアーカイブする場合は、次のように進めます。

- ファイルシステムホストに `root` としてログインします。

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

2. コマンドプロンプトで、コマンド `ssum -a algorithm -h digest -G [-u]filename` を入力します。ここでは:
- `-a algorithm` は、指定されたメッセージダイジェストと照合してファイルを検証するときにファイルシステムが使用するべき暗号化ハッシュ関数を特定します。
 - `-h digest` は、ファイルシステムがファイルを検証するために使用するべきメッセージダイジェストを特定します。
 - `-G` は即時の検証を指定します。ファイルシステムは、`hash` ファイル属性を指定されたメッセージダイジェストの値に設定して、ファイルのメッセージダイジェストを個別に計算し、格納されている値と結果を比較します。指定されたダイジェストと計算したダイジェストが一致する場合、ファイルシステムはファイルの `validated` 属性を設定します。その後、ファイルが再アーカイブされるたびに有効性が再チェックされるように、`generate` 属性を設定します。
 - `-u` は `use` ファイル属性を設定します (オプション)。ファイルがステージングされるたびに、ファイルシステムはダイジェストを再計算し、`hash` 属性に格納されている値と照合して結果を検証します。
 - `filename` は、ファイルのパスと名前です。

この例では、SHA256 ダイジェストを指定して、ファイル `data10` のダイジェスト値を即時に再計算して、指定された値と照合して検証するようにファイルシステムに指示します。コマンド `s1s -D -h data10` を使用してファイル属性をチェックすると、`generate` および `validated` ファイル属性が設定され、`algorithm` 属性が `SHA-256` に設定され、ダイジェスト値が計算されて `hash` 属性に格納されていることがわかります。

```
root@solaris:~# ssum -h f03ce01b3828...f7459503007e -a sha256 -g data10
root@solaris:~# s1s -D -h data10
data10:
mode: -rw-r--r--   links: 1  owner: root      group: root
length: 14975  admin id: 0  inode: 90217.1
project: user.root(1)
access: Jul 16 16:14  modification: Jul 16 16:14
changed: Jul 16 16:15  attributes: Jul 16 16:14
```

```
creation: Jul 16 16:14 residence: Jul 16 16:14
checksum: generate validated algorithm: SHA-256
hash: f03ce01b3828...f7459503007e
root@solaris:~#
```

- 必要に応じて、通常どおりの方法でファイルを編集します。

この例では、*data10m* という名前のファイルが最後にアーカイブされてから、このファイルを変更しました。*sls -D -h* コマンドは、どちらのコピーでも最新の変更が反映されていないために *s* (無効) フラグが両方のコピーで設定されていることを示しています。*Solaris digest* コマンドを使用して、変更したファイルの SHA-256 ダイジェスト値をチェックすると、ファイルの *hash* 属性でも古いダイジェスト値が格納されることがわかります。

```
root@solaris:~# sls -D -h data10m
data10m:
mode: -rw-r--r-- links: 1 owner: root group: root
length: 14983 admin id: 0 inode: 90307.1
project: user.root(1)
copy 1: s----- Jul 17 16:47 dd.1 dk diskarchive f221
copy 2: s----- Jul 20 11:31 a8d.1 li VOL002
access: Jul 20 11:32 modification: Jul 20 11:31
changed: Jul 17 16:37 attributes: Jul 17 16:36
creation: Jul 17 16:36 residence: Jul 17 16:36
checksum: generate algorithm: SHA-256
hash: f03ce01b3828...f7459503007e
root@solaris:~# digest -a sha256 data10m
56c55bb421cc...71ac2ac0b7b0
root@solaris:~#
```

- 必要に応じて、再アーカイブ前に、変更したファイルのダイジェスト属性を変更できます。

この例では、ダイジェストアルゴリズムを SHA256 から SHA1 に変更して、即時に有効にします。

```
root@solaris:~# ssum -a sha1 -G data10m
```

```
root@solaris:~# sls -D -h data10m
data10m:
  mode: -rw-r--r--   links: 1  owner: root      group: root
  length: 14983  admin id: 0  inode: 90307.1
  project: user.root(1)
  release -a;
  copy 1: S----- Jul 20 13:00      e0.1   dk diskarchive f224
  copy 2: S----- Jul 20 13:05      a93.1  li VOL002
  access: Jul 20 16:39  modification: Jul 20 16:39
  changed: Jul 17 16:37  attributes: Jul 17 16:36
  creation: Jul 17 16:36  residence: Jul 20 16:29
  checksum: generate validated  algorithm: SHA-1
  hash: 92003525f0f8...53e29d0718c8
root@solaris:~#
```

5. それ以外の場合、変更したファイルがファイルシステムでアーカイブされ、ダイジェスト関連属性が自動的に更新されるのを待ちます。

変更したファイルがアーカイブされたら、ファイルシステムはダイジェスト値を再計算して、新しい値を *hash* 属性に格納し、古いバージョンのファイルのアーカイブ済みコピーで *S* (無効) フラグを設定します。この例では、ダイジェスト属性を変更せずにファイル *data10m* を編集しました。アーカイバは、スケジュールどおりにディスク上に新しい *copy 1* を作成して、*hash* 属性を更新しました。変更されていないファイルのコピーは、アーカイバが *copy 2* を作成する時間になるまで、*S* (無効) フラグが付けられたままテープ上に残ります。

```
root@solaris:~# sls -D -h data10m
data10m:
  mode: -rw-r--r--   links: 1  owner: root      group: root
  length: 14983  admin id: 0  inode: 90307.1
  project: user.root(1)
  copy 1: ----- Jul 17 16:47      dd.1   dk diskarchive f221
  copy 2: S----- Jul 20 11:31      a8d.1  li VOL002
  access: Jul 20 11:32  modification: Jul 20 11:31
  changed: Jul 17 16:37  attributes: Jul 17 16:36
  creation: Jul 17 16:36  residence: Jul 17 16:36
```

```
checksum: generate algorithm: SHA-256
```

```
hash: 56c55bb421cc...71ac2ac0b7b0
```

メッセージダイジェストの生成およびファイルの検証の有効化

ファイルのダイジェストを生成して、ファイル検証を有効にするには、次のように進めます。

1. ファイルシステムホストに *root* としてログインします。

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

2. コマンドプロンプトで、コマンド `ssum -a algorithm -g|G [-u] filename` を入力します。ここでは:

- `-a algorithm` は、ファイルのメッセージダイジェストの生成時にファイルシステムが使用する暗号化ハッシュ関数を指定します。
- `-g` は、ファイルの *generate* ファイル属性を設定します。はじめてファイルがアーカイブされると、ファイルシステムはメッセージダイジェストを計算します。ファイルが再アーカイブされるたびに、ファイルシステムはダイジェストを再計算し、格納されている値と照合して結果を検証します。
- `-G` は、ファイルの *generate* および *validate* ファイル属性を設定します。ファイルシステムは即時にメッセージダイジェストを計算して、結果を *hash* 属性に格納します。ファイルがアーカイブされるたびに、ファイルシステムはダイジェストを再計算し、格納されている値と照合して結果を検証します。
- `-u` は *use* ファイル属性を設定します (オプション)。ファイルがステージングされるたびに、ファイルシステムはダイジェストを再計算し、*hash* 属性に格納されている値と照合して結果を検証します。
- `filename` は、ファイルのパスと名前です。

この例では、アーカイブ前に、SHA256 アルゴリズムを使用してファイル *data11* のダイジェストを計算するようにファイルシステムに指示します。コマンド `sfs -D -h data10` を使用してファイル属性をチェックすると、ファイルごとに *generate* ファイル属性が設定され、*algorithm* 属性が *SHA-256* に設定されていることがわかります。ファイルはまだアーカイブされていないため、ダイジェスト値はまだ計算されておらず、*hash* 属性に格納されていません。

```
root@solaris:~# ssum -a sha256 -g data11
```

```

root@solaris:~# sls -D -h data11
data11:
  mode: -rw-r--r--   links: 1 owner: root      group: root
  length: 14975 admin id: 0 inode: 90218.1
  project: user.root(1)
  access: Jul 16 16:14 modification: Jul 16 16:14
  changed: Jul 16 16:22 attributes: Jul 16 16:14
  creation: Jul 16 16:14 residence: Jul 16 16:14
  checksum: generate algorithm: SHA-256
  hash:
root@solaris:~#

```

3. 必要に応じて、通常どおりの方法でファイルを編集します。

この例では、*data11m* という名前のファイルが最後にアーカイブされてから、このファイルを変更しました。*sls -D -h* コマンドは、どちらのコピーでも最新の変更が反映されていないために *s* (無効) フラグが両方のコピーで設定されていることを示しています。*Solaris digest* コマンドを使用して、変更したファイルの SHA-256 ダイジェスト値をチェックすると、ファイルの *hash* 属性でも古いダイジェスト値が格納されることがわかります。

```

root@solaris:~# sls -D -h data11m
data11m:
  mode: -rw-r--r--   links: 1 owner: root      group: root
  length: 14983 admin id: 0 inode: 90307.1
  project: user.root(1)
  copy 1: s----- Jul 17 16:47      dd.1   dk diskarchive f221
  copy 2: s----- Jul 20 11:31      a8d.1  li VOL002
  access: Jul 20 11:32 modification: Jul 20 11:31
  changed: Jul 17 16:37 attributes: Jul 17 16:36
  creation: Jul 17 16:36 residence: Jul 17 16:36
  checksum: generate algorithm: SHA-256
  hash: f03ce01b3828...f7459503007e
root@solaris:~# digest -a sha256 data11m
56c55bb421cc...71ac2ac0b7b0
root@solaris:~#

```

4. 必要に応じて、再アーカイブ前に、変更したファイルのダイジェスト属性を変更できます。

この例では、ダイジェストアルゴリズムを SHA256 から SHA1 に変更して、即時に有効にします。

```
root@solaris:~# ssum -a sha1 -G data11m
root@solaris:~# sls -D -h data11m
data11m:
  mode: -rw-r--r--   links: 1  owner: root      group: root
  length: 14983  admin id: 0  inode: 90307.1
  project: user.root(1)
  release -a;
  copy 1: S----- Jul 20 13:00      e0.1   dk diskarchive f224
  copy 2: S----- Jul 20 13:05      a93.1  li V0L002
  access: Jul 20 16:39  modification: Jul 20 16:39
  changed: Jul 17 16:37  attributes: Jul 17 16:36
  creation: Jul 17 16:36  residence: Jul 20 16:29
  checksum: generate validated  algorithm: SHA-1
  hash: 92003525f0f8...53e29d0718c8
root@solaris:~#
```

5. それ以外の場合、変更したファイルがファイルシステムでアーカイブされ、ダイジェスト関連属性が自動的に更新されるのを待ちます。

変更したファイルがアーカイブされたら、ファイルシステムはダイジェスト値を再計算して、新しい値を *hash* 属性に格納し、古いバージョンのファイルのアーカイブ済みコピーで *S* (無効) フラグを設定します。

この例では、ダイジェスト属性を変更せずにファイル *data11m* を編集しました。アーカイバは、スケジュールどおりにディスク上に新しい *copy 1* を作成して、*hash* 属性を更新しました。変更されていないファイルのコピーは、アーカイバが *copy 2* を作成する時間になるまで、*S* (無効) フラグが付けられたままテープ上に残ります。

```
root@solaris:~# sls -D -h data11m
```

```
mdata11:
  mode: -rw-r--r--   links:   1  owner: root      group: root
  length:    14983  admin id:    0  inode:   90307.1
  project: user.root(1)
  copy 1: ----- Jul 17 16:47      dd.1   dk diskarchive f221
  copy 2: s----- Jul 20 11:31      a8d.1  li VOL002
  access:      Jul 20 11:32  modification: Jul 20 11:31
  changed:     Jul 17 16:37  attributes:    Jul 17 16:36
  creation:    Jul 17 16:36  residence:     Jul 17 16:36
  checksum: generate algorithm: SHA-256
  hash: 56c55bb421cc...71ac2ac0b7b0
```

メッセージダイジェストの生成およびディレクトリ内にある各ファイルの検証の有効化

ディレクトリ内の各ファイルのダイジェストを再帰的に生成して、検証属性を設定するには、次のように進めます。

1. ファイルシステムホストに *root* としてログインします。

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

2. コマンドプロンプトで、コマンド `ssum -a algorithm -g|G [-u] -r directoryname` を入力します。ここでは:

- `-a algorithm` は、メッセージダイジェストの生成時にファイルシステムが使用する暗号化ハッシュ関数を指定します。
- `-g` は、各ファイルの *generate* ファイル属性を設定します。はじめてファイルがアーカイブされると、ファイルシステムはそのファイルのメッセージダイジェストを計算します。ファイルが再アーカイブされるたびに、ファイルシステムはダイジェストを再計算し、格納されている値と照合して結果を検証します。
- `-G` は、各ファイルの *generate* および *validate* ファイル属性を設定します。ファイルシステムは即時にメッセージダイジェストを計算して、結果を *hash* 属性に格納します。ファイルがアーカイブされるたびに、ファイルシステムはダイジェストを再計算し、格納されている値と照合して結果を検証します。

- `-u` は `use` ファイル属性を設定します (オプション)。ファイルがステージングされるたびに、ファイルシステムはダイジェストを再計算し、格納されている値と照合して結果を検証します。
- `-r` は、指定されたディレクトリ内のすべてのファイルにコマンドを再帰的に適用します。
- `directoryname` は、ディレクトリのパスと名前です。

最初の例では、アーカイブ前に、SHA256 アルゴリズムを使用してディレクトリ `datasetA` 内のファイルのダイジェストを計算するようにファイルシステムに指示します。コマンド `sls -D -h datasetA` を使用してファイル属性をチェックすると、ファイルごとに `generate` ファイル属性が設定され、`algorithm` 属性が `SHA-256` に設定されていることがわかります。ファイルはまだアーカイブされていないため、ダイジェスト値はまだ計算されておらず、`hash` 属性に格納されていません。

```
root@solaris:~# ssum -a sha256 -g -r datasetA
root@solaris:~# sls -D -h datasetA
datasetA/pdata0:
  mode: -rw-r--r--   links:   1  owner: root      group: root
  length:   14983  admin id:    0  inode:   90232.1
  project: user.root(1)
  access:      Jul 16 16:47  modification: Jul 16 16:47
  changed:     Jul 16 16:47  attributes:    Jul 16 16:47
  creation:    Jul 16 16:47  residence:     Jul 16 16:47
  checksum: generate  algorithm: SHA-256
  hash:
...
datasetA/pdata20:
  mode: -rw-r--r--   links:   1  owner: root      group: root
  length:   14983  admin id:    0  inode:   90234.1
  project: user.root(1)
  access:      Jul 16 16:47  modification: Jul 16 16:47
  changed:     Jul 16 16:47  attributes:    Jul 16 16:47
  creation:    Jul 16 16:47  residence:     Jul 16 16:47
  checksum: generate  algorithm: SHA-256
  hash:
```

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

2 番目の例では、アーカイブ前に、SHA256 アルゴリズムを使用してディレクトリ *datasetB* 内のファイルのダイジェストを即時に計算するようにファイルシステムに指示します。コマンド `sls -D -h datasetB` を使用してファイル属性をチェックすると、ファイルごとに *generate* および *validated* ファイル属性が設定され、*algorithm* 属性が *SHA-256* に設定され、ダイジェスト値が計算されて *hash* 属性に格納されていることがわかります。

```
root@solaris:~# ssum -a sha256 -G -r datasetB
root@solaris:~# sls -D -h datasetB
datasetB/qdata0:
  mode: -rw-r--r--   links: 1  owner: root      group: root
  length: 14983  admin id: 0  inode: 90232.1
  project: user.root(1)
  access: Jul 16 16:47  modification: Jul 16 16:47
  changed: Jul 16 16:47  attributes: Jul 16 16:47
  creation: Jul 16 16:47  residence: Jul 16 16:47
  checksum: generate validated  algorithm: SHA-256
  hash: 4d2800eb82b3...520341edde95
...
datasetB/qdata12:
  mode: -rw-r--r--   links: 1  owner: root      group: root
  length: 14983  admin id: 0  inode: 90234.1
  project: user.root(1)
  access: Jul 16 16:47  modification: Jul 16 16:47
  changed: Jul 16 16:47  attributes: Jul 16 16:47
  creation: Jul 16 16:47  residence: Jul 16 16:47
  checksum: generate validated  algorithm: SHA-256
  hash: 5b057f1b7b48...88c590d47dec
...
root@solaris:~#
```

ステージング中のファイルのメッセージダイジェストの検証

必要に応じて、ファイルを使用するためにディスクキャッシュにステージングする前に、このファイルを検証できます。次のように進めます。

1. ファイルシステムホストに *root* としてログインします。

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

2. コマンドプロンプトで、コマンド `ssum -u [-a algorithm [-h digest]] -g|G filename` を入力します。ここでは:

- `-u` は、*use* ファイル属性を設定することでステージング前の検証を指定します。ファイルで *use* 属性を設定すると、ファイルシステムは、メッセージダイジェストを生成して、ファイルの *hash* 属性に格納されている値と照合して結果を正常に検証するまで、ファイルをアーカイブメディアからディスクキャッシュにコピーしません。
- `-a algorithm`、`-h digest`、および `-g|G` は、必要な *algorithm*、*hash*、および *generate* 属性が以前に設定されていない場合に、これらの属性をファイルで設定するオプションパラメータです。
- *filename* は、ファイルのパスと名前です。

この例では、ファイル *data102* の検証をすでに有効にしました。コマンド `sls -D -h data102` が示すように、*generate* および *validated* ファイル属性が設定され、*algorithm* 属性が *SHA-256* に設定され、ダイジェスト値が計算されて *hash* 属性に格納されています。

```
root@solaris:~# ssum -a sha256 -F data102
root@solaris:~# sls -D -h data102
data102:
  mode: -rw-r--r--   links:   1  owner: root      group: root
  length:   14979  admin id:    0  inode:   90264.1
  project: user.root(1)
  access:      Jul 16 17:34  modification: Jul 16 17:34
  changed:     Jul 16 17:34  attributes:    Jul 16 17:34
  creation:    Jul 16 17:34  residence:     Jul 16 17:34
  checksum: generate validated  algorithm: SHA-256
  hash: baae932ce1cf...93166a2e36b5
```

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

そのため、ステージング前にファイルシステムがファイルを必ず検証するように、*use* 属性を設定できます。コマンド `sls -D -h data102` は、*use* 属性が次のように設定されたことを示します。

```
root@solaris:~# ssum -u data102
root@solaris:~# sls -D -h data102
data102:
  mode: -rw-r--r--   links: 1  owner: root      group: root
  length: 14979  admin id: 0  inode: 90264.1
  project: user.root(1)
  access: Jul 16 17:34  modification: Jul 16 17:34
  changed: Jul 16 17:34  attributes: Jul 16 17:34
  creation: Jul 16 17:34  residence: Jul 16 17:34
  checksum: generate use validated  algorithm: SHA-256
  hash: baae932ce1cf...93166a2e36b5
root@solaris:~#
```

ファイルがアーカイブされる前のメッセージダイジェストと検証属性の変更

ファイルが不変にされておらず、まだアーカイブされていない場合、次の手順を使用して、メッセージダイジェストと検証属性を変更できます。

1. ファイルシステムホストに *root* としてログインします。

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

2. 必要に応じて、ダイジェストアルゴリズムを変更します。コマンドプロンプトで、コマンド `ssum -a newalgorithm filename` を入力します。ここでは:
 - `-a newalgorithm` は、以前に指定されたダイジェストアルゴリズムを置き換える暗号化ハッシュ関数を指定します。
 - `filename` は、ファイルのパスと名前です。

この例の保存ポリシーでは、耐競合性の高い SHA256 関数が必要です。しかし、コマンド `sls -D -h` が示すように、ファイル `data319` のダイジェスト属性の設

定時に、誤って SHA1 アルゴリズムを指定しました。ファイルはまだアーカイブされていないため、アルゴリズムを正常に SHA256 に変更できます。

```
root@solaris:~# sls -D -h data319
data319:
  mode: -rw-r--r--   links: 1  owner: root      group: root
  length: 14983  admin id: 0  inode: 90301.1
  project: user.root(1)
  access: Jul 17 15:27  modification: Jul 17 15:27
  changed: Jul 17 15:28  attributes: Jul 17 15:27
  creation: Jul 17 15:27  residence: Jul 17 15:27
  checksum: generate  algorithm: SHA-1
  hash:
root@solaris:~# ssum -a sha256 data319
root@solaris:~# sls -D -h data319
data319:
  mode: -rw-r--r--   links: 1  owner: root      group: root
  length: 14983  admin id: 0  inode: 90301.1
  project: user.root(1)
  access: Jul 17 15:27  modification: Jul 17 15:27
  changed: Jul 17 15:28  attributes: Jul 17 15:27
  creation: Jul 17 15:27  residence: Jul 17 15:27
  checksum: generate  algorithm: SHA-256
  hash:
root@solaris:~#
```

- 必要に応じて、ダイジェスト属性をクリアして、デフォルトのファイル設定を復元します。コマンドプロンプトで、コマンド `ssum -d filename` を入力します。ここでは:
 - `-d` は、ファイルのダイジェスト属性をそのデフォルト値にリセットします。
 - `filename` は、ファイルのパスと名前です。

この例では、ファイル `data44` のメッセージダイジェストと検証を構成するつもりではありませんでした。しかし、コマンド `sls -D -h` が示すように、誤ってこれを行いました。ファイルはまだアーカイブされていないため、アーカイブおよびステージング中にダイジェスト検証を制御する属性である `generate` および

use を正常にクリアできます。*validated*、*algorithm*、および *hash* 属性内のデータは残りますが、ファイルシステムの動作には影響しません。

```
root@solaris:~# sls -D -h data44
data44:
  mode: -rw-r--r--   links: 1  owner: root      group: root
  length: 14983  admin id: 0  inode: 90292.1
  project: user.root(1)
  access: Jul 17 14:58  modification: Jul 17 14:57
  changed: Jul 17 14:58  attributes: Jul 17 14:57
  creation: Jul 17 14:57  residence: Jul 17 14:57
  checksum: generate use validated  algorithm: SHA-256
  hash: 3b4b15f8f69c...bae62c7e7568
root@solaris:~# ssum -d data44
root@solaris:~# sls -D -h data44
data44:
  mode: -rw-r--r--   links: 1  owner: root      group: root
  length: 14983  admin id: 0  inode: 90292.1
  project: user.root(1)
  access: Jul 17 14:58  modification: Jul 17 14:57
  changed: Jul 17 14:58  attributes: Jul 17 14:57
  creation: Jul 17 14:57  residence: Jul 17 14:57
  checksum: validated  algorithm: SHA-256
  hash: 3b4b15f8f69c...bae62c7e7568
root@solaris:~#
```

4. 必要に応じて、ファイルをアーカイブする前に、必要なメッセージダイジェストと検証属性をすべてリセットします。コマンドプロンプトで、適切なオプションとファイル名を指定してコマンド *ssum* を入力します。

この例では、ファイル *qndat44* でメッセージダイジェストを再度有効にして、アーカイブ前にダイジェストを検証します。ただし、ステージング前の検証は必要ありません。そのため、*generate* 属性は復元します、*use* 属性は復元しません。

```
root@solaris:~# ssum -g data44
root@solaris:~# sls -D -h data44
```

```
data44:
  mode: -rw-r--r--   links:   1  owner: root       group: root
  length:   14983  admin id:   0  inode:   90292.1
  project: user.root(1)
  access:      Jul 17 14:58  modification: Jul 17 14:57
  changed:     Jul 17 14:58  attributes:   Jul 17 14:57
  creation:    Jul 17 14:57  residence:    Jul 17 14:57
  checksum: generate validated  algorithm: SHA-256
  hash: 3b4b15f8f69c...bae62c7e7568
root@solaris:~#
```

ファイルを変えないにする

保存の要件では頻繁に、ファイルの変換性を保証するメカニズムが必要です。アーカイブでは、変更を防止して、そのような変更が発生していないことを証明する必要があります。変換性を提供するには、Oracle HSM アーカイブファイルシステムは、前述したメッセージダイジェストとダイジェスト関連のファイル属性を、ファイルを変えないにする追加の属性と結合します。ファイルを変えないにしたあと、スーパーユーザー権限を持つユーザーのみがそのステータスを変更できます。変換性を厳密な Write Once Read Many (WORM) ファイルシステムと結合した場合、スーパーユーザーも変更を行うことができなくなります (詳細は、[「WORM ファイルシステムの理解」](#)を参照)。

次のいずれかの方法で、ファイルを変えないにできます。

- [メッセージダイジェストを指定してファイルを変えないにする](#)
- [メッセージダイジェストを生成してファイルを変えないにする](#)

メッセージダイジェストを指定してファイルを変えないにする

アーカイブへの取り込み後にファイルが変更されていないことを保証する必要がある場合、次のように進めます。

1. ファイルシステムホストに `root` としてログインします。

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

2. コマンドプロンプトで、コマンド `ssum -a algorithm [-h digest] -F filename` を入力します。ここでは:

- `-a algorithm` は、指定されたメッセージダイジェストと照合してファイルを検証するときにファイルシステムが使用するべき暗号化ハッシュ関数を特定します。
- `-h digest` は、ファイルシステムがファイルを検証するために使用するべきメッセージダイジェストを特定します。
- `-F` は、即時の検証と不変性を指定し、`fixity`、`generate`、`validated`、および `use` ファイル属性を設定します。ファイルシステムは即時にメッセージダイジェストを計算して検証します。ファイルをステージングまたはアーカイブすると、ファイルシステムはメッセージダイジェストを再計算して再検証します。
- `filename` は、ファイルのパスと名前です。

この例では、SHA256 ダイジェストを指定して、ダイジェストを再計算するようにファイルシステムに指示し、ファイル `data20` の値を検証して、ファイルを不変にします。コマンド `sls -D -h data10` を使用してファイル属性をチェックすると、ファイルごとに `fixity`、`generate`、`use`、および `validated` ファイル属性が設定され、`algorithm` 属性が `SHA-256` に設定され、ダイジェスト値が計算されて `hash` 属性に格納されていることがわかります。

```
root@solaris:~# ssum -h bfaefde932cf...d450892eda63 -a sha256 -F data20
root@solaris:~# sls -D -h data20
data20:
  mode: -rw-r--r--   links:   1  owner: root           group: root
  length:      14979  admin id:      0  inode:   90264.1
  project: user.root(1)
  access:      Jul 16 17:34  modification: Jul 16 17:34
  changed:     Jul 16 17:34  attributes:    Jul 16 17:34
  creation:    Jul 16 17:34  residence:     Jul 16 17:34
  checksum: fixity generate use validated  algorithm: SHA-256
  hash: bfaefde932cf...d450892eda63
root@solaris:~#
```

メッセージダイジェストを生成してファイルを不変にする

関連付けられたメッセージダイジェストがすでにあるファイルをアーカイブするときに、アーカイブへの取り込み後にファイルが変更されていないことを保証する必要がある場合、次のように進めます。

1. ファイルシステムホストに *root* としてログインします。

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

2. コマンドプロンプトで、コマンド `ssum -a algorithm [-h digest] -F filename` を入力します。ここでは:

- `-a algorithm` は、`-h digest` パラメータで指定されたダイジェストを生成するために使用された暗号化ハッシュ関数を特定します。
- `-F` は、`fixity`、`generate`、`validated`、および `use` ファイル属性を設定します。ファイルシステムは即時にメッセージダイジェストを計算して検証します。ファイルをステージングまたはアーカイブすると、ファイルシステムはメッセージダイジェストを再計算して再検証します。
- `filename` は、ファイルのパスと名前です。

この例では、SHA256 ダイジェストを計算して、ファイル `data200` の値を検証し、ファイルを変換するようにファイルシステムに指示します。コマンド `sls -D -h data10` を使用してファイル属性をチェックすると、ファイルごとに `fixity`、`generate`、`validated`、および `use` ファイル属性が設定され、`algorithm` 属性が `SHA-256` に設定され、ダイジェスト値が計算されて `hash` 属性に格納されていることがわかります。

```
root@solaris:~# ssum -a sha256 -F data200
```

```
root@solaris:~# sls -D -h data200
```

```
data200:
```

```
mode: -rw-r--r--   links:   1  owner: root           group: root
```

```
length:   14979  admin id:    0  inode:   90264.1
```

```
project: user.root(1)
```

```
access:      Jul 16 17:34  modification: Jul 16 17:34
```

```
changed:     Jul 16 17:34  attributes:    Jul 16 17:34
```

```
creation:    Jul 16 17:34  residence:     Jul 16 17:34
```

```
checksum: fixity generate use validated  algorithm: SHA-256
```

```
hash: efde93cc12cf...d496602e36dd
```

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

ファイルのダイジェストと fixity 属性のチェック

1 つ以上のファイルのメッセージダイジェストと fixity 属性を表示するには、Oracle HSM のディレクトリ一覧表示コマンド `s1s` を使用します。次のように進めます。

メッセージダイジェストと検証属性の一覧表示

1. ファイルシステムホストに `root` としてログインします。

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

2. コマンドプロンプトで、コマンド `s1s -D -h filename` を入力します。ここでは:

- `-D` は、ファイル属性の詳細表示を指定します。
- `-h` は、表示にハッシュ (ダイジェスト) 値を含めます。
- `filename` は、パスと名前でも 1 つ以上のファイルを識別します。

この例では、表示の `checksum` および `hash` フィールドに一覧表示されているファイル `data02` のファイルダイジェスト属性を表示します。

```
root@solaris:~# s1s -D -h data02
data02:
mode: -rw-r--r--   links:   1  owner: root           group: root
length:      14975  admin id:      0  inode:   90217.1
project: user.root(1)
access:      Jul 16 16:14  modification: Jul 16 16:14
changed:     Jul 16 16:15  attributes:    Jul 16 16:14
creation:    Jul 16 16:14  residence:     Jul 16 16:14
checksum: generate use validated  algorithm: SHA-256
hash: f03ce01b3828...f7459503007e
root@solaris:~#
```

- `hash` 属性は、ファイルのメッセージダイジェスト `f03ce01b3828...f7459503007e` を格納します。
- `algorithm` 属性は、`SHA-256` 暗号化ハッシュ関数によって格納済みのメッセージダイジェストが生成されたことを示しています。

- *generate* 属性は、ファイルがアーカイブされるたびに、ファイルシステムはメッセージダイジェストを個別に再計算して、格納されている値と照合して検証することを示しています。
- *use* 属性は、ファイルがステージングされるたびに、ファイルシステムはメッセージダイジェストを個別に再計算して、格納されている値と照合して検証することを示しています。
- *validated* 属性は、個別に計算されたメッセージダイジェストが、最後のチェック時に、*hash* 属性に格納されている値と一致したことを示しています。
- ファイルが不変にされている場合、*fixity* 属性が表示されます。

WORM ファイルシステムの使用

法律上またはアーカイブに関する考慮事項で必要な場合、Write-Once Read-Many (WORM) ディレクトリとファイルをサポートするように構成された Oracle HSM ファイルシステムでこれらのディレクトリとファイルを作成できます。このセクションでは、WORM ファイルシステムと、次を含む、WORM ファイルとディレクトリの操作時に実行する必要がある特定のタスクの理解に焦点を当てます。

- [ディレクトリの WORM 対応化](#)
- [ファイルの WORM 保護のアクティブ化](#)
- [WORM ファイルの検索および一覧表示](#)

ファイルシステムの WORM サポートの有効化については、『Oracle Hierarchical Storage Manager and StorageTek QFS インストールおよび構成ガイド』を参照してください。

WORM ファイルシステムの理解

Write Once Read Many (WORM) ファイルシステムは、指定された保持期間ユーザーがファイルを読み取り専用にできるようにすることでデータを保護します。WORM 対応の Oracle HSM ファイルシステムでは、デフォルトおよびカスタマイズ可能なファイル保持期間、データとパスの不変性、および WORM 設定のサブディレクトリの継承がサポートされます。

ファイルシステムの構成方法に応じて、次の 2 つの Oracle HSM WORM モードのうちの 1 つを使用します。

- 標準のコンプライアンスモード (デフォルト)
- エミュレーションモード

標準の WORM モードでマウントされたファイルシステムでは、ユーザーは、コマンド `chmod 4000 path_name` を実行して、ディレクトリを WORM 対応にしてファイルの読み取り専用保持期間を開始します。`path_name` は、ファイルまたはディレクトリのパスと名前です。これによって、UNIX `setuid` (実行時にユーザー ID を設定) アクセス権が設定されます。`execute` アクセス権も持つファイルに対して `setuid` アクセス権を設定することはセキュリティリスクとなるため、標準の WORM モードでは、実行可能ではないファイルのみを読み取り専用にできます。

WORM エミュレーションモードでマウントされたファイルシステムでは、ユーザーは、コマンド `chmod 555 path_name` を実行して、ディレクトリを WORM 対応にしてファイルの読み取り専用保持期間を開始します。`path_name` は、書き込み可能ファイルまたはディレクトリのパスと名前です。エミュレーションモードでは `setuid` アクセス権は必要ないため、実行可能ファイルを読み取り専用にして、保持期間を割り当てることができます。

標準モードとエミュレーションモードはどちらも、厳密な WORM 実装と、制限の少ないライト実装を備えています。厳密な実装とライト実装のどちらでも、ファイルまたはディレクトリに対する保持が起動されたあとは、データまたはパスに対する変更は許可されません。どちらも、デフォルトの保持期間を 43,200 分 (30 日) に設定します。ただし、ライト実装では、`root` ユーザーへの制限が一部緩和されます。

厳密な実装では、だれも指定された保持期間を短縮したり、保持期間の終了前にファイルまたはディレクトリを削除したりできません。また、`sammkfs` を使用して、現在保持されているファイルとディレクトリが格納されているボリュームを削除することもできません。そのため、厳密な実装は、要求の厳しい法律上、規制上のコンプライアンス、および保存の要件を満たすのに適しています。

ライト実装では、`root` ユーザーは、保持期間の短縮、ファイルとディレクトリの削除、`sammkfs` コマンドを使用したボリュームの削除を行うことができます。これは、日常的なデータ損失に対する高レベルの保護を提供し、ファイルシステムとストレージリソースを管理する際に柔軟性を向上させます。ただし、スーパーユーザーにこの程度の制御を許可するファイルシステムは、法律上および規制上のコンプライアンス要件を満たしていない可能性があります。

WORM ファイルへのハードリンクとソフトリンクの両方を作成できます。WORM 対応ディレクトリ内にあるファイルとのハードリンクのみを作成できます。作成されたハードリンクは、元のファイルと同じ WORM 特性を持ちます。ソフトリンクも作成できますが、ソフトリンクでは WORM 機能を使用できません。WORM ファイルへのソフトリンクは、Oracle HSM ファイルシステム内の任意のディレクトリに作成できます。

WORM ファイルシステムの作成および構成の詳細は、お客様向けドキュメントライブラリの『*Oracle Hierarchical Storage Manager and StorageTek QFS インストールおよび構成ガイド*』を参照してください。

ディレクトリの WORM 対応化

ディレクトリを WORM 対応にする場合、WORM ファイルのサポートを追加しますが、それ以外の場合はディレクトリの特性は変更しないでください。ユーザーは、引き続き WORM 対応ディレクトリ内でファイルを作成および編集でき、WORM ファイルが含まれていない WORM 対応ディレクトリは削除可能です。ディレクトリを WORM 対応にするには、次のように進めます。

1. ファイルシステムサーバーにログインします。

```
user@solaris:~#
```

2. ディレクトリがすでに WORM 対応になっているかどうかを確認します。コマンド `s1s -Dd directory` を使用します。ここで `directory` は、ディレクトリのパスと名前です。コマンドの出力で属性 `worm-capable` を探します。

あるユーザーがディレクトリを WORM 対応にすると、現在および将来の子ディレクトリはすべて WORM 機能を継承するため、通常ディレクトリは WORM 対応になります (コマンドの詳細は、`s1s` マニュアルページを参照)。最初の例では、ターゲットディレクトリ `/hsm/hsmfs1/records` がすでに WORM 対応であることがわかります。

```
user@solaris:~# s1s -Dd /hsm/hsmfs1/records/2013/
/hsm/hsmfs1/records/2013:
  mode: drwxr-xr-x   links:   2  owner: root       group: root
  length:      4096  admin id:   0  inode:   1048.1
  project: user.root(1)
```

```
access:      Mar  3 12:15  modification: Mar  3 12:15
changed:     Mar  3 12:15  attributes:    Mar  3 12:15
creation:    Mar  3 12:15  residence:     Mar  3 12:15
worm-capable      retention-period: 0y, 30d, 0h, 0m
```

ただし、2 番目の例では、ターゲットディレクトリ `/hsm/hsmfs1/documents` は WORM 対応ではないことがわかります。

```
user@solaris:~# s1s -Dd /hsm/hsmfs1/documents
/hsm/hsmfs1/documents
mode: drwxr-xr-x  links:  2  owner: root      group: root
length:      4096  admin id:    0  inode:    1049.1
project: user.root(1)
access:      Mar  3 12:28  modification: Mar  3 12:28
changed:     Mar  3 12:28  attributes:    Mar  3 12:28
creation:    Mar  3 12:28  residence:     Mar  3 12:28
```

3. ディレクトリが WORM 対応ではない場合、およびファイルシステムが `worm_capable` または `worm_lite` マウントオプションを使用してマウントされた場合、Solaris コマンド `chmod 4000 directory-name` を使用して WORM サポートを有効にします。`directory-name` は、WORM ファイルを保持するディレクトリのパスと名前です。

コマンド `chmod 4000` によって、ディレクトリで `setuid` (実行時にユーザー ID を設定) 属性が設定され、標準の WORM サポートが有効になります。この例では、ディレクトリ `/hsm/hsmfs1/documents` を WORM 対応にして、`s1s -Dd` を使用して結果をチェックします。操作が成功すると、ディレクトリが WORM 対応になります。

```
user@solaris:~# chmod 4000 /hsm/hsmfs1/documents
user@solaris:~# s1s -Dd /hsm/hsmfs1/documents
/hsm/hsmfs1/documents
mode: drwxr-xr-x  links:  2  owner: root      group: root
length:      4096  admin id:    0  inode:    1049.1
project: user.root(1)
access:      Mar  3 12:28  modification: Mar  3 12:28
```

```

changed:      Mar  3 12:28  attributes:   Mar  3 12:28
creation:     Mar  3 12:28  residence:    Mar  3 12:28
worm-capable          retention-period: 0y, 30d, 0h, 0m

```

4. ディレクトリが WORM 対応ではない場合、およびファイルシステムが *worm_emul* または *emul_lite* マウントオプションを使用してマウントされた場合、Solaris コマンド *chmod 555 directory-name* を使用して WORM サポートを有効にします。*directory-name* は、WORM ファイルを保持するディレクトリのパスと名前です。

コマンド *chmod 555* によって、ディレクトリの書き込みアクセス権が削除され、WORM エミュレーションサポートが有効になります。この例では、ディレクトリ */hsm/hsmfs1/documents* を WORM 対応にして、コマンド *sls -Dd* を使用して結果をチェックします。操作が成功すると、ディレクトリが WORM 対応になります。

```

user@solaris:~# chmod 555 /hsm/hsmfs1/documents
user@solaris:~# sls -Dd /hsm/hsmfs1/documents
/hsm/hsmfs1/documents
mode: drwxr-xr-x  links:  2  owner: root      group: root
length:      4096  admin id:    0  inode:   1049.1
project: user.root(1)
access:      Mar  3 12:28  modification: Mar  3 12:28
changed:     Mar  3 12:28  attributes:   Mar  3 12:28
creation:    Mar  3 12:28  residence:    Mar  3 12:28
worm-capable          retention-period: 0y, 30d, 0h, 0m

```

ファイルの WORM 保護のアクティブ化

WORM 対応ディレクトリ内のファイルで WORM 保護をアクティブにすると、ファイルシステムでは、保持期間が期限切れになるまで、ファイルデータまたはそのデータへのパスに対する変更が許可されなくなります。そのため、注意して使用する必要があります。WORM 保護をアクティブにするには、次のように進めます。

1. ファイルシステムサーバーにログインします。

```

user@solaris:~#

```

2. ファイルシステムのデフォルト以外にしばらくの期間ファイルを保持する必要がある場合、ファイルのアクセス時間を変更して、必要な保持時間を指定します。Solaris コマンド `touch -a -texpiration-date path-name` を使用します。ここでは:

- `expiration-date` は、4 桁の年、2 桁の月、2 桁の日、2 桁の時間、2 桁の分、オプションで 2 桁の秒で構成される数字の文字列です。
- `path-name` は、ファイルのパスと名前です。

`touch` などの Oracle Solaris UNIX ユーティリティーは、2038 年 1 月 18 日午後 10:14 を超えて保持期間を延長できません。これらのユーティリティーは、符号付き 32 ビット数値を使用して、1970 年 1 月 1 日から秒単位で時間を表します。そのため、この期限を超えてファイルを保持する必要がある場合、デフォルトの保持期間を使用してください。

この例では、4 年後の 2019 年 10 月 4 日午前 11:59 に期限切れになるようファイルの保持期間を設定します。

```
user@solaris:~# touch -a -t201910141159 /hsm/hsmfs1/plans/master.odt
```

3. ファイルシステムが `worm_capable` または `worm_lite` マウントオプションを使用してマウントされた場合、Solaris コマンド `chmod 4000 path-name` を使用して WORM 保護をアクティブにします。`path-name` は、ファイルのパスと名前です。

`chmod 4000` コマンドによって、指定されたファイルで `setuid` (実行時にユーザー ID を設定) 属性が設定されます。実行可能ファイルに対してこの属性を設定することはセキュアではありません。そのため、ファイルシステムが `worm_capable` または `worm_lite` マウントオプションを使用してマウントされた場合、UNIX `execute` アクセス権を持つファイルでは WORM 保護を設定できません。

この例では、ファイル `master.odt` の WORM 保護をアクティブにします。`sls -D` を使用して結果をチェックして、`retention` 属性が現在は `active` に設定され、`retention-period` が 4 年に設定されていることを確認します。

```
user@solaris:~# chmod 4000 /hsm/hsmfs1/plans/master.odt
user@solaris:~# sls -Dd /hsm/hsmfs1/plans/master.odt
```



```

/hsm/hsmfs1/plans/master.odt:
 mode: -r-xr-xr-x  links: 1  owner: root      group: root
 length:      104  admin id: 0  inode: 1051.1
 project: user.root(1)
 access:      Mar  4 2018  modification: Mar  3 13:14
 changed:     Mar  3 13:16  retention-end: Apr  2 14:16 2014
 creation:    Mar  3 13:16  residence:     Mar  3 13:16
 retention:   active      retention-period: 4y, 0d, 0h, 0m

```

4. ファイルシステムが *worm_emul* または *emul_lite* マウントオプションを使用してマウントされた場合、Solaris コマンド *chmod 555 path-name* を使用して WORM 保護をアクティブにします。 *path-name* は、ファイルのパスと名前です。

コマンド *chmod 555* によって、ディレクトリの書き込みアクセス権が削除されます。そのため、必要に応じて実行可能ファイルを WORM 保護できます。次の例では、ファイル *master-plan.odt* の WORM 保持をアクティブにします。 *sls -D* を使用して結果をチェックして、 *retention* 属性が現在は *active* に設定され、 *retention-period* が 4 年に設定されていることを確認します。

```

user@solaris:~# chmod 555 /hsm/hsmfs1/plans/master.odt
user@solaris:~# sls -Dd /hsm/hsmfs1/plans/master.odt
/hsm/hsmfs1/plans/master.odt:
 mode: -r-xr-xr-x  links: 1  owner: root      group: root
 length:      104  admin id: 0  inode: 1051.1
 project: user.root(1)
 access:      Mar  4 2018  modification: Mar  3 13:14
 changed:     Mar  3 13:16  retention-end: Apr  2 14:16 2014
 creation:    Mar  3 13:16  residence:     Mar  3 13:16
 retention:   active      retention-period: 4y, 0d, 0h, 0m

```

WORM ファイルの検索および一覧表示

指定した検索基準を満たす WORM ファイルを検索して一覧表示するには、 *sfind* コマンドを使用します。次のように進めます。

1. ファイルシステムサーバーにログインします。

```
user@solaris:~#
```

2. WORM 保護されていて、アクティブに保持されているファイルを一覧表示するには、コマンド `sfind starting-directory -ractive` を使用します。`starting-directory` は、一覧表示プロセスを開始するディレクトリのパスと名前です。

```
user@solaris:~# sfind /hsm/hsmfs1/ -ractive
/hsm/hsmfs1/documents/2013/master-plan.odt
/hsm/hsmfs1/documents/2013/schedule.ods
/samma1/records/2013/progress/report01.odt
/samma1/records/2013/progress/report02.odt
/samma1/records/2013/progress/report03.odt ...
user@solaris:~#
```

3. 保持期間が期限切れになった WORM 保護ファイルを一覧表示するには、コマンド `sfind starting-directory -rover` を使用します。`starting-directory` は、一覧表示プロセスを開始するディレクトリのパスと名前です。

```
user@solaris:~# sfind /hsm/hsmfs1/ -rover
/samma1/documents/2007/master-plan.odt
/samma1/documents/2007/schedule.ods
user@solaris:~#
```

4. 指定されたの日付と時間のあとで保持期間が期限切れになる WORM 保護ファイルを一覧表示するには、コマンド `sfind starting-directory -rafter expiration-date` を使用します。ここでは:
 - `starting-directory` は、一覧表示プロセスを開始するディレクトリのパスと名前です。
 - `expiration-date` は、4 桁の年、2 桁の月、2 桁の日、2 桁の時間、2 桁の分、オプションで 2 桁の秒で構成される数字の文字列です。

次の例では、2015 年 1 月 1 日午前 12:01 のあとで保持期間が期限切れになるファイルを一覧表示します。

```
user@solaris:~# sfind /hsm/hsmfs1/ -rafter 201501010001
```

```
/hsm/hsmfs1/documents/2013/master-plan.odt
user@solaris:~#
```

5. 少なくとも指定した期間ファイルシステム内に残る必要がある WORM 保護ファイルを一覧表示するには、コマンド `sfind starting-directory -rremain time-remaining` を使用します。ここでは:
 - `starting-directory` は、検索を開始するディレクトリツリー内の場所です。
 - `time-remaining` は、時間の単位 *y* (年)、*d* (日)、*h* (時)、*m* (分) とペアになった、負ではない整数の文字列です。

この例では、少なくともさらに3年間保持される、ディレクトリ `/hsm/hsmfs1/` 内のファイルをすべて検索します。

```
user@solaris:~# sfind /hsm/hsmfs1/ -rremain 3y
/hsm/hsmfs1/documents/2013/master-plan.odt
user@solaris:~#
```

6. 指定した期間より長くファイルシステム内に残る必要がある WORM 保護ファイルを一覧表示するには、コマンド `sfind starting-directory -rlonger time` を使用します。ここでは:
 - `starting-directory` は、検索を開始するディレクトリツリー内の場所です。
 - `time-remaining` は、時間の単位 *y* (年)、*d* (日)、*h* (時)、*m* (分) とペアになった、負ではない整数の文字列です。

この例では、3年と90日を超えて保持される、ディレクトリ `/hsm/hsmfs1/` 内のファイルをすべて検索します。

```
user@solaris:~# sfind /hsm/hsmfs1/ -rremain 3y90d
/hsm/hsmfs1/documents/2013/master-plan.odt
user@solaris:~#
```

7. 永続的にファイルシステムに残る必要がある WORM 保護ファイルを一覧表示するには、コマンド `sfind starting-directory -rpermanent` を使用します。

この例では、ディレクトリ `/hsm/hsmfs1/` 内のどのファイルも永続的には保持されないことがわかります。

```
user@solaris:~# sfind /hsm/hsmfs1/ -rpermanent  
user@solaris:~#
```

第7章 構成およびファイルシステムのバックアップ

Oracle Hierarchical Storage Manager and StorageTek QFS Software をインストールおよび構成したときに、回復ポイントファイルとアーカイブログのコピーを格納するためのセキュアな場所を作成しました。また、回復ポイントの作成、ログのバックアップ、およびシステム構成の保護を自動的に行うプロセスを構成しました。これらの手順では、ファイルシステムの中心的な保護機能が提供されます。ただし、ときどきスケジュール外の保護対策が必要となる場合もあります。

- 物理インフラストラクチャーまたはデータセンター設備への重要な変更など、混乱を生じる可能性のある、予測されるイベントの前に、Oracle HSM の構成およびファイルシステムをバックアップします。
- 現在の構成が保護されるように、ソフトウェア、オペレーティングシステム、またはホストプラットフォームのアップグレードまたは再構成のあとに、Oracle HSM の構成およびファイルシステムをバックアップします。
- Oracle HSM サポートサービスを受ける前に、必要な構成およびステータス情報を収集します。

この章では、必要に応じて構成およびファイルシステムの回復ファイルを収集、作成、および格納するための手順について概要を示します。主要な3つのセクションで構成されます。

- [ファイルシステムのバックアップ](#)
- [Oracle HSM 構成のバックアップ](#)
- [samexplorer を使用した構成および診断情報の収集](#).

この章では、すべてのタスクでコマンド行インターフェースを使用します。Oracle HSM Manager のグラフィカルユーザーインターフェースを使用する場合は、オンラインヘルプで詳細な手順を参照してください。

ファイルシステムのバックアップ

このセクションでは、Oracle HSM ファイルシステムの保護を簡単に見直すことから始めます (「回復ポイントおよびアーカイブログの理解」)。続いて、次のタスクを実行するための手順を示します。

- 要求に応じた回復ポイントの作成
- アーカイブログのバックアップ
- **samexplorer** を使用した構成および診断情報の収集
- Oracle HSM 構成の手動バックアップ。

回復ポイントおよびアーカイブログの理解

ファイルシステムを保護するには、2つのことを実行する必要があります。

- データが保持されているファイルを保護する必要があります。
- データを使用、整理、検索、アクセス、および管理できるように、ファイルシステム自体を保護する必要があります。

Oracle HSM アーカイブファイルシステムでは、ファイルデータはアーカイバによって自動的に保護されます。変更されたファイルは、テープなどのアーカイブストレージメディアに自動的にコピーされます。ただし、ファイルしかバックアップしていないときに、ディスクデバイスまたは RAID グループに回復不能な障害が発生した場合は、データは保持されますが使用することは難しくなります。代替のファイルシステムの作成、各ファイルの特定、新しいファイルシステム内の適切な場所の決定、そのファイルの取り込み、およびそのファイルとユーザー、アプリケーション、その他のファイルとの間の失われた関係の再作成を行う必要があります。このような回復は、最善の状況でも、面倒で時間のかかるプロセスとなります。

したがって、すばやく効率的に回復するには、ファイルおよびアーカイブコピーを使用可能にするファイルシステムのメタデータを積極的に保護する必要があります。リムーバルメディア上でアーカイブされたコピーに、ディレクトリパス、iノード、アクセス制御、シンボリックリンク、およびポインタをバックアップする必要があります。

Oracle HSM ファイルシステムのメタデータを保護するには、回復ポイントをスケジュールし、アーカイブログを保存します。回復ポイントは、Oracle HSM ファイルシステムのメタデータのポイントインタイムバックアップコピーを格納する圧縮

ファイルです。データの損失 (ユーザーファイルの誤った削除から、ファイルシステム全体の壊滅的な損失まで) が発生した場合は、ファイルまたはファイルシステムが元の状態のままである最新の回復ポイントを見つければ、即座にファイルまたはファイルシステムの既知の良好な最新状態まで回復できます。次に、その時点で記録されたメタデータを復元し、メタデータに示されているファイルをアーカイブメディアからディスクキャッシュにステージングするか、または可能であれば、ユーザーおよびアプリケーションがファイルにアクセスするときに必要に応じてファイルシステムでファイルをステージングするようにします。

ポイントインタイムバックアップコピーと同様に、回復ポイントが障害が発生した時点のファイルシステムの状態の完全なレコードであることは、ほとんどありません。必然的に、1つの回復ポイントが完成してから、次の回復ポイントが作成されるまで、少なくとも数個のファイルが作成および変更されます。ファイルシステムが使用されていないときに回復ポイントを頻繁に作成するようにスケジュールすれば、この問題を最小限にできます。ただし、現実には、ファイルシステムは使用するために存在するため、スケジュールリングには妥協が必要です。

このため、アーカイバログファイルのポイントインタイムコピーを保存する必要もあります。それぞれのデータファイルがアーカイブされると、ログファイルには、アーカイブメディアのボリュームシリアル番号、アーカイブセットとコピー番号、メディアでのアーカイブ (*tar*) ファイルの位置、および *tar* ファイル内でのデータファイルのパスと名前が記録されます。この情報があれば、Solaris または Oracle HSM *tar* ユーティリティーを使用して、失われたファイルを回復ポイントから回復できます。ただし、この情報は変動します。大部分のシステムログと同様に、アーカイバログは急速に増加するため、頻繁に上書きされてしまいます。定期的にコピーして回復ポイントを補完していなければ、必要なときにログ情報がないことになります。

このセクションの残りの部分では、要求に応じて回復ポイントとログのコピーを作成する手順について説明します。次のサブセクションが含まれます。

- [要求に応じた回復ポイントの作成](#)
- [アーカイバログのバックアップ](#)。

要求に応じた回復ポイントの作成

通常のスケジュール外の時点で、アーカイブシステムファイルからメタデータを取得する必要がある場合があります。破損の可能性があることが予測されるシステム

や機能を保守する際には常に、前後に回復ポイントを作成し、ファイルシステムが確実に保護されるようにします。

オンデマンドでスケジュールの回復ポイントの作成を開始するには、次の手順を実行します。

1. Oracle HSM サーバーホストに *root* としてログインします。

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

2. 回復ポイントが格納される独立した場所を選択します。

この例では、最初にファイルシステムを構成したときに回復ポイント用に作成したディレクトリの下に、サブディレクトリ *unscheduled/* を作成します。*/zfs1* ファイルシステムはリモートの場所に配置され、Oracle HSM ファイルシステムと共通のコンポーネントは持っていません。

```
root@solaris:~# mkdir /zfs1/samqfs_recovery/unscheduled
root@solaris:~#
```

3. ファイルシステムのルートディレクトリに移動します。

この例では、マウントポイントディレクトリ */samqfs* に移動します。

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

4. データがリムーバブルメディアにコピーされるアーカイブファイルシステムをバックアップする場合は、メタデータのみをバックアップします。コマンド *samfsdump -f recovery-point* を使用します。ここで *recovery-point* は、最終的な回復ポイントファイルのパスおよびファイル名です。

詳細については、*samfsdump* のマニュアルページを参照してください。この例では、スケジュールされた保守関連の停電の前に、*samqfs* ファイルシステム用のスケジュール外の回復ポイントを作成します。ディレクトリ */zfs1/samqfs_recovery/unscheduled/* に回復ポイントファイル *20150315pre-outage* を作成します (次の2つ目のコマンドは1行で入力し、改行はバックスラッシュ文字でエスケープします)。


```

root@solaris:~# cd /samqfs
root@solaris:~# samfsdump -f /
/zfs1/samqfs_recovery/unscheduled/20150315pre-outage
root@solaris:~#

```

5. データがリムーバブルメディアにコピーされないスタンドアロンファイルシステムをバックアップする場合は、メタデータとデータの両方をバックアップします。コマンド `samfsdump -U -f recovery-point` を使用します。ここで `recovery-point` は、最終的な回復ポイントファイルのパスおよびファイル名です。

データとメタデータが含まれる回復ポイントファイルは、非常に大きくなる可能性があることに注意してください。詳細については、`samfsdump` のマニュアルページを参照してください。この例では、`samqfs` ファイルシステム用のスケジュール外の回復ポイントを作成します。リモートディレクトリ `/zfs1/samqfs_recovery/unscheduled/` に回復ポイントファイル `20150315pre-outage` を作成します (次の2つ目のコマンドは1行で入力し、改行はバックスラッシュ文字でエスケープします)。

```

root@solaris:~# cd /samqfs
root@solaris:~# samfsdump -f -U /
/zfs1/samqfs_recovery/unscheduled/20150315pre-outage
root@solaris:~#

```

6. アーカイブファイルシステムをバックアップする場合は、アーカイバログをバックアップします。
7. それ以外の場合は、状況に応じて、`samexplorer` の実行や、Oracle HSM 構成の手動バックアップが必要となることもあります。

アーカイバログのバックアップ

回復ポイントファイルに、ファイルシステムの復元時に必要となる情報のほぼすべてが含まれている場合は、回復ポイントが作成されたあとに作成または変更されたファイルのメタデータは保持されません。アーカイバログにはアーカイブされているすべてのファイルおよびカートリッジ上の格納場所が一覧表示されるため、アーカイバログを使用すれば、回復ポイントの作成後にアーカイブされたファイルを

回復できます。そのため、可能であれば、スケジュール外の回復ポイントを作成する際は必ず、アーカイバログファイルのスケジュール外のコピーを作成してください。次のように進めます。

1. Oracle HSM サーバーホストに *root* としてログインします。

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

2. バックアップされたアーカイバログが格納される独立した場所を選択します。

この例では、上記で作成した対応するスケジュール外の回復ポイントと同じディレクトリに、ログを格納することに決定します。/zfs1 ファイルシステムはリモートの場所に配置され、Oracle HSM ファイルシステムと共通のコンポーネントは持っていません。

```
root@solaris:~# ls /zfs1/samqfs_recovery/unscheduled
20150315pre-outage
root@solaris:~#
```

3. 選択した場所に現在のアーカイバログをコピーし、一意の名前を付けます。コマンド `cp /var/adm/samqfs.archive.log path/"date +%y%m%d";` を使用します。ここで *path* は選択した場所へのパスです。

次のコマンドは 1 行で入力し、改行はバックスラッシュ文字でエスケープします。

```
root@solaris:~# cp /var/adm/samqfs.archive.log /
/zfs1/samqfs_recovery/unscheduled/20150315pre-outage/"date +%y%m%d".archive.log
root@solaris:~#
```

4. 状況に応じて、*samexplorer* の実行や、Oracle HSM 構成の手動バックアップが必要となることもあります。

Oracle HSM 構成のバックアップ

Oracle HSM 構成を変更する際は必ず、変更した構成ファイルおよび関連情報をすべてバックアップして、投資を保護してください。次のタスクを実行します。

- [Oracle HSM 構成の手動バックアップ](#)

- [samexplorer](#) の実行.

Oracle HSM 構成の手動バックアップ

完全な冗長化のために、ソフトウェア、オペレーティングシステム、またはホストを大幅に変更する際は必ず、構成ファイルのローカルコピーを作成してください。次のように進めます。

1. ファイルシステムホストに *root* としてログインします。

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

2. バックアップ構成情報が保持されているサブディレクトリに、Oracle HSM 構成の手動バックアップコピー用のサブディレクトリを作成します。コマンド `mkdir mount-point/path` を使用します。ここで *mount-point* は選択した独立ファイルシステム用のマウントポイントディレクトリ、*path* は選択したディレクトリのパスと名前です。

この例では、アーカイブファイルシステム */samqfs* 用の回復ポイントを構成します。そのため、ディレクトリ */zfs1/sam_config/samconfig* を作成しました。

```
root@solaris:~# mkdir /zfs1/sam_config/samconfig
```

3. Oracle HSM 構成の手動バックアップコピーが保持されているサブディレクトリに、現在の Oracle HSM 構成用のサブディレクトリを作成します。コマンド `mkdir mount-point/path/subdirectory` を使用します。ここで *mount-point* は選択した独立ファイルシステム用のマウントポイント、*path/subdirectory* は選択したディレクトリのパスと名前です。

この例では、この目的のために初期構成時に作成したディレクトリ */zfs1/sam_config/samconfig* に、サブディレクトリを作成します。日付を使用してサブディレクトリの名前を付けます。

```
root@solaris:~# mkdir /zfs1/sam_config/samconfig/20150315
```

4. 構成ファイルを別のファイルシステムにコピーします。

```
/etc/opt/SUNWsamfs/
```

```
mcf
archiver.cmd
defaults.conf
diskvols.conf
hosts.family-set-name
hosts.family-set-name.local
preview.cmd
recycler.cmd
releaser.cmd
rft.cmd
samfs.cmd
stager.cmd
inquiry.conf
samremote          # SAM-Remote server configuration file
family-set-name    # SAM-Remote client configuration file
network-attached-library # Parameters file
scripts/*          # Back up all locally modified files
/var/opt/SUNWsamfs/
```

5. すべてのライブラリカタログデータ (ヒストリアンで保持されているデータを含む) をバックアップします。カタログごとに、コマンド `/opt/SUNWsamfs/sbin/dump_cat -V catalog-file` を使用します。ここで、*catalog-file* はカタログファイルのパスと名前です。出力を新しい場所にある *dump-file* にリダイレクトします。

この例では、*library1* のカタログデータを、NFS でマウントされた個別のファイルシステム *zfs1* 上のディレクトリにあるファイル *library1cat.dump* にダンプします (次のコマンドは 1 行で入力し、改行はバックスラッシュ文字でエスケープします)。

```
root@solaris:~# dump_cat -V /var/opt/SUNWsamfs/catalog/library1cat > /zfs1/sam
_config/20150315/catalogs/library1cat.dump
```

6. Oracle HSM のインストールおよび構成中に変更したシステム構成ファイルをコピーします。これには、次が含まれる可能性があります。

```
/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. Oracle HSM 構成の一部として作成したカスタムシェルスクリプトおよび *crontab* エントリを選択したサブディレクトリにコピーします。

たとえば、回復ポイントの作成を管理するために *crontab* エントリを作成した場合は、ここでコピーを保存します。

8. Oracle HSM 構成の一部として作成したカスタムシェルスクリプトおよび *crontab* エントリを選択したサブディレクトリにコピーします。

たとえば、回復ポイントの作成を管理するために *crontab* エントリを作成した場合は、ここでコピーを保存します。

9. 現在インストールされているソフトウェア (Oracle HSM、Solaris、Solaris Cluster (該当する場合) を含む) のリビジョンレベルを記録し、選択したサブディレクトリ内の *readme* ファイルに情報のコピーを保存します。
10. 必要になったときにソフトウェアをすばやく復元できるように、選択したサブディレクトリに、ダウンロードした Oracle HSM Oracle HSM、Solaris、および Solaris Cluster パッケージのコピーを保存します。
11. 次に、*samexplorer* を実行します。

samexplorer を使用した構成および診断情報の収集

samexplorer は、Oracle HSM ソフトウェアおよびファイルシステムに関する包括的な構成およびステータス情報を取得およびレポートする診断ツールです。Oracle HSM 構成を変更する際は必ず、*samexplorer* を実行して、構成ファイルのバック

アップコピーとともに結果レポートを格納するようにしてください。また、問題のトラブルシューティングを行うときや、Oracle HSM サポートサービスの担当者から実行するよう要求されたときにも、*samexplorer* を実行するようにしてください。次のように進めます。

samexplorer の実行

1. ファイルシステムホストに *root* としてログインします。
2. バックアップ構成情報が保持されているディレクトリに、*samexplorer* レポート用のサブディレクトリを作成します。コマンド *mkdir mount-point/path* を使用します。ここで *mount-point* は選択した独立ファイルシステム用のマウントポイントディレクトリ、*path* は選択したディレクトリのパスと名前です。

この例では、この目的のために初期構成時に作成したディレクトリ */zfs1/sam_config/* に、新しいサブディレクトリを作成します。新しいサブディレクトリに *explorer/* という名前を付けます。

```
root@solaris:~# mkdir /zfs1/sam_config/explorer
```

3. 選択したディレクトリに、*samexplorer* レポートを作成します。コマンド *samexplorer path/hostname.YYYYMMDD.hhmmz.tar.gz* を使用します。ここで *path* は選択したディレクトリへのパス、*hostname* は Oracle HSM ファイルシステムホストの名前、*YYYYMMDD.hhmmz* は日付とタイムスタンプです。

デフォルトのファイル名は */tmp/*

SAMreport.hostname.YYYYMMDD.hhmmz.tar.gz です。この例では、ホスト *samhost1* のレポートを、2015 年 3 月 15 日 (山地標準時午後 4:59) の日付で作成します (次のコマンドは 1 行で入力し、改行はバックslash文字でエスケープします)。

```
root@solaris:~# samexplorer /  
/zfs1/sam_config/explorer/samhost1.20150315.1659MST.tar.gz
```

```
Report name:    /zfs1/sam_config/explorer/samhost1.20150315.1659MST.tar.gz  
Lines per file: 1000  
Output format: tar.gz (default) Use -u for unarchived/uncompressed.
```

```
Please wait.....
```

Please wait.....

Please wait.....

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

/zfs1/sam_config/explorer/samhost1.20150315.1659MST.tar.gz

4. ファイルシステムを大幅に再構成する際は必ず、この手順を繰り返してください。
5. ここで停止します。Oracle HSM 構成がバックアップされています。

第8章 新しいストレージメディアへの移行

テープは、長期間にわたるデータのストレージおよび保存に適した、信頼性の高い安定したメディアです。ただし、その寿命には限りがあります。通常の機械的処理(マウント、テンション調整、および読み取り/書き込み)による摩耗や傷が、時間の経過とともに蓄積します。より性能の高い新しいドライブが使用可能になると、古いドライブはサポートが困難になり、互換性のあるメディアも高価で希少になります。したがって、ある時点で、新しいメディアにアーカイブを転送する必要があります。

Oracle HSM 階層型ファイルシステムでは、古いメディアを新しいメディアに置き換えることは複雑なプロセスです。テープメディアはファイルシステムの不可欠な部分です。ファイルシステムのメタデータは、各ファイルのデータの複数のコピーに関する場所を記録します(一部がディスク上にあり、一部がテープ上にあるなど)。したがって、テープをコピーする際は、ファイルシステムのiノードを更新して、移行されたファイルのコピーの新しい場所を反映させる必要があります。同時に、アーカイブ処理やステージング処理などの通常のファイルシステム操作がコピーの作成によって妨害されないように、メディアおよびドライブを管理する必要があります。

Oracle Hierarchical Storage Manager では、メディア移行の複雑さに対処するために2つの方法が用意されています。要件に応じて、それぞれに利点があります。

Oracle HSM 6.1 で導入されたメディア移行機能は、あるライブラリドライブにマウントされたメディアのボリューム全体を、別のドライブにマウントされたメディアにコピーし、その過程でファイルシステムのメタデータを更新します(手動でロードされたドライブはサポートされません)。これにより、システムのオーバーヘッドと管理者の作業負荷が最小限に抑えられます。ボリュームは、アーカイブ処理またはステージング処理のためにドライブが必要でないときに、バックグラウンドでコピーされます。使用するドライブの数、および1日のうちで移行を実行してもよい時間を指定できます。または、ドライブがアイドル状態になるたびに Oracle HSM

でボリュームを移行するようにもできます。ドライブまたはボリュームがアーカイブ処理またはステージングジョブに必要になると、メディア移行プロセスは優先度の高い操作に譲ります。StorageTek T1000D (またはそれ以降) 移行先ドライブが正しく構成されて使用可能になっている場合は、T10000 拡張コピー機能および Oracle HSM *xcopy* オプションを使用して移行できます。要求が行われたあと、ドライブ自体がコピーを処理し、サーバーリソースは使用されません。それ以外の場合は、Oracle HSM のサーバーコピーオプションを使用してサーバーの負荷を最小限に抑えることができます。その後、ファイルシステムサーバーは、構成可能な入出力バッファを介してドライブからドライブにボリュームをコピーします。

通常のアーカイブ処理でアーカイブファイルを 1 回に 1 つずつ処理することによっても、データを移行できます。システムを構成して、ファイルを古いメディアからディスクキャッシュにステージングしたあとで新しいメディアに再度アーカイブするようにできます。このファイルごとのアプローチでは、ファイルのグループ化および分散の方法を詳細に制御できます。ただし、より多くの管理が必要になります。ディスクキャッシュおよびドライブリソースの割り当てはユーザー自身で行うため、通常ファイルシステム操作への干渉を最小限に抑える必要がある場合は、慎重に計画する必要があります。

このドキュメントの残りの部分では、プロセスを順に説明します。

- [移行の準備](#)
- [移行方式の選択](#)
- [ボリューム全体の移行またはファイルのステージングおよび交換用メディアへの再アーカイブによるデータの移行。](#)

移行の準備

先に進む前に、次の手順を実行します。

- [ファイルシステムが常にバックアップされていることの確認](#)
- [必要なメディアの用意。](#)

ファイルシステムが常にバックアップされていることの確認

メディアの移行を開始する前に、通常は Oracle HSM アーカイブデータを保護する回復メカニズムが、切り替え中および切り替え後も有効に保たれていることを確認します。壊滅的なハードウェア障害とユーザーエラーの可能性は、移行操作中に大き

くも小さくもなりません。そのため、通常と同様に、現存の `samfsdump` 回復ポイントファイルからファイルやファイルシステム全体を回復できることを確認する必要があります。

移行中および移行後しばらくは、新しい移行先ボリュームではなく移行元のテープボリュームを参照する回復ポイントファイルが回復に利用されます。重大なハードウェア障害によってファイルシステムが使用不可になり、これらの古いテープが使用できない場合は、回復できなくなります。

したがって、少なくとも、現在のファイルシステムを新しいメディアから復元するために十分な新しい回復ポイントが作成されるまで、古いテープを保持することを計画してください。特定の時点のファイルを復元できる必要がある場合は、無期限ではなくても、より長い期間にわたって古いメディアを保持する必要があることがあります。理想的には、古いボリュームはすぐにアクセス可能なライブラリで維持すべきです。

必要なメディアの用意

移行先のライブラリに、移行されたファイルを保持するために十分なメディアが含まれていることを確認します。「[新しいテープのラベル付けまたは既存のテープの再ラベル付け](#)」の説明に従って、すべてのボリュームに正しくラベルが付けられていることを確認します。ボリュームにラベルが付いていない場合、移行は失敗します。

移行方式の選択

移行方式は、アーカイブの状態およびユーザーとアプリケーションの要件に応じて選択します。次の手順を使用して決定してください。

ニーズを満たす最適な移行方式の選択

1. 移行中にアーカイブを操作可能に保つかどうかを決定します。

アーカイブを休止し、すべてのリソースを移行のためだけに使用すると、タスクを簡素化し、より速く完了できます。ただし、アーカイブがアクティブに使用されている場合、これはほとんど実用になりません。

2. ボリューム全体ではなくアーカイブファイルのグループを選択的に移行する必要がある場合や、アーカイブファイルのグループ間の指定された関係を保持する必

要がある場合は、ステージングと再アーカイブの方式を使用します。「[ファイルのステージングおよび交換用メディアへの再アーカイブ](#)」に進みます。

3. 単に古いボリュームを新しいメディアにコピーする必要がある場合や、ファイルシステム操作への移行の影響を最小限に抑える必要がある場合は、ボリューム移行の方式を使用します。
4. 移行先ドライブとして使用できるファイバチャネル Oracle StorageTek T10000D (またはそれ以降) ドライブがない場合や、ソースとターゲットのテープのブロックサイズが同じでない場合は、サーバーコピー方式を使用します。

このモードでは、Oracle HSM ソフトウェアは、移行元ドライブからファイルシステムサーバー上の構成可能な入出力バッファに、有効なアーカイブファイルだけをコピーします。移行元と移行先のブロックサイズが異なっている場合、移行先のブロックサイズの方が大きければ、ソフトウェアは自動的に調整を実行します。次に、ソフトウェアはバッファから移行先ドライブにテープブロックを送信します。

5. ファイバチャネル StorageTek T10000D (またはそれ以降) 移行先ドライブがある場合、両方のドライブで現行のファームウェアが実行されている場合、ソースとターゲットのテープのブロックサイズが同じである場合、および移行元と移行先のドライブが同じストレージエリアネットワーク (SAN) スイッチを介して接続されている場合は、Oracle HSM *xcopy* オプションを使用します。

xcopy を指定すると、ファイルシステムサーバーからドライブに SCSI コピー要求が送信され、T10000D ドライブはコピー元テープからコピー先テープに、最初の有効なアーカイブファイルから始めて 1 ブロックずつコピーします。何らかの理由で *xcopy* 操作が失敗した場合、移行ソフトウェアは自動的にサーバーコピー方式に切り替わります。*xcopy* 方式では、サーバーのオーバーヘッドが最小限に抑えられ、パフォーマンスが最大になります。

ドライブおよびファームウェアの要件の詳細については、リリースノート (ダウンロード ZIP ファイル内の *README.txt*、またはファイルシステムサーバー上の */opt/SUNWsamfs/doc/README.txt*) を参照してください。

6. 移行元ボリュームに含まれている期限切れのファイルが比較的少ない場合は、*xcopy* オプションを *eod* (データの終わり) モードで使用します。

このモードの場合、T10000 ドライブは、テープ上で最初の有効なファイルから EOD (データの終わり) マークまでの間にあるすべてのアーカイブファイルをコ

ピーします。これらのファイルの一部が期限切れになっている場合、それらは有効なファイルを使用して移行先ボリュームにコピーされます。

7. 移行元ボリュームに含まれている期限切れのファイルが多い場合は、*xcopy* オプションを *repack* モードで使用します。

このモードの場合、T10000 ドライブは、期限切れではないアーカイブファイルだけを移行先ボリュームにコピーします。

8. [ファイルのステージングおよび交換用メディアへの再アーカイブ](#)に進みます。

ボリューム全体の移行

サーバーコピー方式または直接コピー方式を選択し、*migrationd.cmd* ファイルを作成して移行を構成します。次のタスクを実行します。

- [migrationd.cmd ファイルの作成](#)
- [アクティブな移行ジョブの確認](#)
- [ボリュームの移行](#)

migrationd.cmd ファイルの作成

1. Oracle HSM メタデータサーバーに *root* としてログインします。

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

2. */etc/opt/SUNWsamfs/migrationd.cmd* ファイルをテキストエディタで開きます。

この例では、*vi* エディタで新しいファイルを開き、最初のコメントを追加します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
# /etc/opt/SUNWsamfs/migrationd.cmd
# A configuration file for migrating data from old tape volumes to replacements
```

3. 少数のボリュームを移行する場合は、移行元ボリューム、移行先ボリューム、および移行の方向をそれぞれ指定します。移行元ボリュームごとに、*migrate = from source to destination* という形式の行を入力します。ここでは:

- *media_type* は、移行元のメディアの種類を識別する 2 文字のコードです (詳細は、付録A「装置タイプの用語集」を参照)。
- *VSN* は、ライブラリ内のテープボリウムを識別する一意のボリウムシリアル番号です。

この例では、古い LTO (*li*) テープ *VOL305* から新しい Oracle StorageTek T10000 (*ti*) テープカートリッジ *VOL820* にデータを移行します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
# /etc/opt/SUNWsamfs/migrationd.cmd
# A configuration file for migrating data from old tape volumes to replacements
# Migrate a single volume.
migrate = from li VOL305 to ti VOL820
```

4. 多数のボリウムを移行する必要がある場合は、移行元および移行先のボリウムのメディアプールを定義します。 *vsnpool = poolname library equipment _number media_type VSNlist* という形式の行を入力して各プールを定義します。ここでは:
 - *name* は、プールを一意に識別します。
 - *equipment_number* は、移行元ボリウムを保持しているライブラリに *mcf* ファイルで割り当てられている元の番号です。
 - *media_type* は、移行元のメディアの種類を識別する 2 文字のコードです (詳細は、付録A「装置タイプの用語集」を参照)。
 - *VSNlist* は、リテラル *VSN* や *VSN* のグループと範囲を識別する正規表現を、スペースで区切ったリストです。

この例では、古い LTO4 (*li*) テープボリウムから新しい LTO6 (*ti*) テープカートリッジにデータを移行します。移行するライブラリ *20* 内の LTO4 ボリウムを表す、移行元プール *pool11* の行を追加します。これらには、範囲 *VOL000* から *VOL299* までの *VSN* を持つボリウムと、2つの個別ボリウム *VOL300* および *VOL304* が含まれます。次に、ライブラリ *30* 内の LTO6 ボリウムの範囲を表す、移行先プール *pool12* の行を追加します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
# /etc/opt/SUNWsamfs/migrationd.cmd
# A configuration file for migrating data from old tape volumes to replacements
```

```
# pool1 contains the source volumes
vsnpool = pool1 library 20 li ^VOL[0-2][0-9][0-9] VOL300 VOL304
# pool2 contains the destination volumes
vsnpool = pool2 library 30 li ^VOL50[0-9]
```

5. 移行元および移行先のメディアプールを定義したら、移行の方向を指定します。*migrate = from sourcepool to destinationpool* という形式の行を入力します。ここでは:

- *sourcepool* は、移行されるデータが含まれているメディアプールです。
- *destinationpool* は、移行されるデータを受け取るメディアプールです

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
vsnpool = pool1 library 20 li ^VOL[0-2][0-9][0-9] VOL300 VOL304
# pool2 contains the destination volumes
vsnpool = pool2 library 30 ti ^VOL50[0-9]
# Migrate data from tapes in pool1 to tapes in pool2.
migrate = from pool1 to pool2
```

6. サーバーコピー移行方式だけを使用する予定である場合は、*xcopy* 機能を無効にします。*xcopy = off* という形式の行を入力します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
# Disable xcopy and the StorageTek T10000 Extended Copy feature.
xcopy = off
```

7. StorageTek T10000 拡張コピー機能だけを使用し、この機能をサポートするドライブが使用できないときにはデータを移行しない予定であれば、*xcopy* 移行のみを有効にします。*xcopy = only* という形式の行を入力します。

この例では、*xcopy* だけを有効にします。移行元または移行先のドライブが拡張コピー機能をサポートしていない場合、移行ソフトウェアは自動的に移行を取り消します。

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

```

...
# Enable xcopy, StorageTek T10000D Extended Copy feature.
# If the source or destination is not xcopy capable, cancel migration.
xcopy = only

```

8. 可能な場合には StorageTek T10000 拡張コピー機能を活用する予定であれば、*xcopy* 移行方式を有効にします。*xcopy = on* という形式の行を入力します。

この例では、互換性のあるドライブが移行期間中常に使用できるとは限らない場合でも *xcopy* を有効にします。移行元または移行先のドライブが拡張コピー機能をサポートしていない場合、移行ソフトウェアは自動的にサーバーコピーモードに切り替わります。

```

root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
# Enable xcopy, StorageTek T10000D Extended Copy feature.
# If the source or destination is not xcopy capable, automatically switch
# to the server buffer copy.
xcopy = on

```

9. 期限切れのファイルが少数含まれているテープボリュームを *xcopy* 方式を使用して移行する場合は、*xcopy* を *eod* (データの終わり) モードで実行するように設定します。*xcopy_eod = on* という形式の行を入力します。

```

root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
xcopy = on
xcopy_eod = on

```

10. 期限切れのファイルが多数含まれているテープボリュームを *xcopy* 方式を使用して移行する場合は、*xcopy* を *repack* モードで実行するように設定します。*xcopy_eod = off* という形式の行を入力します。

```

root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
xcopy = on
xcopy_eod = off

```


11. より優先度の高いアーカイブまたはステージングタスクによって中断されるまでに `xcopy` でコピーする必要があるデータの最小量を指定します。 `xcopy_minsize = amountunits` という形式の行を入力します。ここでは:

- `amount` は整数です。
- `units` は、キロバイトの場合は `k`、メガバイトの場合は `M`、ギガバイトの場合は `G`、テラバイトの場合は `T`、ペタバイトの場合は `P`、エクサバイトの場合は `E` です。

この値により、T10000 ドライブの効率的な使用とほかのタスクへの可用性の間で妥協点が定義されます。値が大きいほど、ドライブへのデータの書き込み効率が高くなります。値が小さいほど、アーカイブ処理およびステージング処理へのドライブの可用性が高くなります。この例では、最小のコピーサイズを 30G バイトに設定します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
xcopy_eod = on
# xcopy can be interrupted after 30GB copied.
xcopy_minsize = 30G
```

12. 1 日のうちで移行ジョブの実行を許可する期間を定義します。 `runtime = window` という形式の行を入力します。ここで、 `window` は次のいずれかの値です。

- `always` は、アーカイブ処理またはステージング処理のためにドライブとメディアが必要でないときは常に、移行デーモンによるデータの移行を許可します。移行デーモンは、使用しているドライブまたはメディアがアーカイブ処理またはステージング処理のために必要になると、それらを譲ります。
- `start_time end_time`。ここで、 `start_time` と `end_time` はそれぞれ、許可する期間の開始時間と終了時間を、24 時間制の時間と分で表現したものです (`HHMM`)。

`samcmd`、`migstart`、`migidle`、または `migstop` コマンドを発行して、いつでもこのディレクティブをオーバーライドできます。

ボリュームとドライブのどちらかがステージャーまたはアーカイバで必要になると、移行サービスはボリュームとドライブを放棄します。そのため、アーカイブ

処理またはステージング処理で(たとえば、ピーク時に)問題が発生しないかぎり、デフォルト値 *always* を受け入れるようにしてください。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
xcopy_minsize = 30G
# Run all of the time. Migration daemon will yield VSNs and drives when
# resources are wanted by the SAM-QFS archiver and stager.
run_time = always
```

13. ログディレクトリを指定することによってロギングを有効にします。 *logdir = path* という形式の行を入力します。ここで、 *path* はディレクトリパスとディレクトリ名です。

ディレクトリが定義されている場合、移行デーモンは、各移行元ボリュームから移行する各アーカイブファイルの移行先をログに記録します。移行元ボリュームごとに独自のログファイルがあり、 *media_type.VSN* という名前が付けられます。ここでは:

- *media_type* は、移行元メディアの種類を識別する 2 文字のコードです (詳細は、付録A「装置タイプの用語集」を参照)。
- *VSN* は、移行元ボリュームを識別する一意のボリュームシリアル番号です。

そのため、たとえば、VSN が *VOL300* である移行元ボリュームのログファイルの名前は *li.VOL300* になります。

アーカイバログと同様に、これらの移行ログは障害回復の際に非常に役立つ場合があります (詳細は、「回復ポイントおよびアーカイブログの理解」および『Oracle Hierarchical Storage Manager and StorageTek QFS Software ファイルシステム回復ガイド』を参照)。そのため、可能な場合は常にログディレクトリを指定してください。 */var/adm/* など、Oracle HSM ソフトウェアまたはハードウェアの障害によって影響を受けない場所を選択します。この例では、ディレクトリ */var/adm/hsm_migration_logs* を指定します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
run_time = always
# Log directory for the migration logs.
```

```
logdir = /var/adm/hsm_migration_logs
```

14. 移行 *i* ノードデータベースのホームディレクトリを指定します。 *dbdir = path* という形式の行を入力します。ここで、 *path* はディレクトリの絶対パス名です。

移行元ボリュームごとに *i* ノードデータベースが作成され、移行期間中は保持されます。移行元ボリューム上で見つかったアーカイブコピーごとに、224 バイトのデータベースレコードが1つ作成されます。そのため、移行元メディアに格納される可能性のある最大数のコピーに対応できる、十分なディスク領域がある場所を選択する必要があります。たとえば、Oracle StorageTek T10000D ボリュームごとに、最大 8,200,104,892 個のアーカイブコピーが保持される可能性があります。したがって、特定の時点で移行する T10000D ボリュームごとに、約 1.67 テラバイトのデータベース領域が必要になります (詳細は、*migration.cmd(1m)* のマニュアルページを参照)。

データベースのデフォルトの場所は */var/opt/SUNWsamfs/sammig/db* です。この例では、デフォルトのディレクトリを指定します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
logdir = /var/adm/hsm_migration_logs
# database home directory
dbdir = /var/opt/SUNWsamfs/sammig/db
```

15. 移行先デバイスの移行バッファースイズを設定します。 *buffsize = media_type blocks* という形式の行を入力します。ここでは:

- *media_type* は、移行元のメディアの種類を識別する 2 文字のコードです (詳細は、[付録A「装置タイプの用語集」](#)を参照)。
- *size* は *[2-8192]* の範囲の整数です。この整数値は、バッファが保持できるべきテープブロックの数を指定します。デフォルトは *64* です。

この例では、デフォルト数の Oracle StorageTek T10000 テープブロックを保持するために十分な領域を割り当てます。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
# database home directory
```

```
dbdir = /var/opt/SUNWsamfs/sammig/db
# allocate buffer space for 64 T10000D tape blocks
bufsize = ti 64
```

16. 1 ライブラリ当たりの移行に使用できるドライブの最大数を指定します。*max_drives = library-list* という形式の行を入力します。ここでは:

- *library-list* はライブラリエントリをスペースで区切ったリストで、各エントリの形式は *library equipment-number device-count* です。
- *equipment-number* は、*mcf* ファイルでライブラリに割り当てられている装置の順序番号です。
- *device-count* は、指定されたライブラリで使用できるドライブの数です。デフォルトでは、*device-count* はライブラリ内のドライブの数に設定されます。

ボリュームとドライブのどちらかがステージャーまたはアーカイバで必要になると、移行サービスはボリュームとドライブを放棄します。そのため、アーカイブ処理またはステージング処理で問題が発生しないかぎり、デフォルト設定を受け入れ、任意の空きドライブを移行に使用できるようにしてください。この例では、実際にドライブの使用を制限する必要があることがわかっています。そのため、移行用のドライブとしてライブラリ 20 の 8 台、ライブラリ 30 の 6 台、およびライブラリ 40 の 2 台を割り当てます。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
dbdir = /var/opt/SUNWsamfs/sammig/db
# allocate buffer space for 64 T10000D tape blocks
bufsize = ti 64
# For migration, use 8 drives in library 20, 6 in 30, and 2 in 40
max_drives = library 20 8 library 30 6 library 40 2
```

17. 同時に実行できる、移行に関連したコピー操作の最大数を指定します。*max_copy = processes* という形式の行を入力します。ここで、*processes* は整数です。

デフォルトは最大値です。これは、*mcf* ファイルに記載されたすべてのライブラリに構成されているドライブの数を 2 で割った値に等しくなります。この例では、最大 8 つの同時コピー処理を許可します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
bufsize = ti 64
# For migration, use 8 drives in library 20, 6 in 30, and 2 in 40
max_drives = library 20 8 library 30 6 library 40 2
# Up to 8 sam-migcopy process can be run simultaneously.
max_copy = 8
```

18. 同時に実行できる、移行に関連したテープスキャン操作の最大数を指定します。 *max_scan = processes* という形式の行を入力します。ここで、*processes* は整数です。

移行元 VSN 上のアーカイブコピーを特定するために、*sam-migrationd* デーモンは、*mcf* に構成されているすべてのファイルシステムをスキャンし、ディスクキャッシュからすべての *i* ノードを読み取り、各 *i* ノード内の *vsn* フィールドと移行元ボリュームのボリュームシリアル番号 (VSN) を比較します。この処理によってファイルシステムのメタデータのアクティビティが増加するため、ファイルシステムのパフォーマンスに悪影響が生じる可能性があります。

ファイルシステムのパフォーマンスと移行速度の間で最適なバランスが得られる値を選択してください。または、デフォルトの 4 を受け入れると、ほとんどの使用状況に対応できます。移行速度を最大にするためにファイルシステムを休止する場合は、*max_scan* を 0 に設定して、すべての移行元ボリュームが一度にスキャンされるようにします。この例では、通常のファイルシステム操作に影響を与えることなく最大 8 つの同時スキャン処理を許可できることが、経験からわかっています。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
bufsize = ti 64
# For migration, use 8 drives in library 20, 6 in 30, and 2 in 40
max_drives = library 20 8 library 30 6 library 40 2
```

```
# Run up to 8 sam-migcopy processes simultaneously.
max_copy = 8
# Scan up to 8 VSNS simultaneously.
max_scan = 8
```

19. ファイルを保存して、エディタを閉じます。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/migrationd.cmd
...
max_copy = 8
# Scan up to 8 VSNS simultaneously.
max_scan = 8
:wq
root@solaris:~#
```

アクティブな移行ジョブの確認

このセクションの手順では、シェルコマンドプロンプトから `samcmd` コマンドを使用してコマンドを入力する方法について説明します。ただし、すべてのコマンドは、`:command` という形式で `samu` インタフェースからも入力できます。ここで、`command` はコマンドの名前です。

1. 現在、Oracle HSM メタデータサーバーに `root` としてログインしていない場合は、ここでログインします。

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

2. 以前の移行が現在アクティブになっていたり不完全であったりしないこと確認します。最初に、現在の移行ステータスを確認します。コマンド `samcmd x` を使用します。

別の移行コピージョブが進行中の場合、このコマンドは、そのコピージョブの移行元および移行先のボリューム (メディアタイプと VSN)、コピーモード、完了率、および現在のステータスを表示します。

```
root@solaris:~# samcmd x
Migration status   samcmd   version HH:MM:SS month day year
samcmd on hsm61s01
```

```
Status: Stop: Waiting for :migstart
source  dest    cmod perc status
li VOL004 li VOL042 - 60% Copy idled
```

ほかの移行コピージョブが実行されていない場合、このコマンドでジョブは表示されません。

```
root@solaris:~# samcmd x
Migration status      samu      ver  time date
Source Vsns - wait:  0 fsscan: 0 copy: 0 update ino: 0 log: 0 done:  0
Status: Idle: Waiting for :migstart
source dest cmod perc status
```

- 次に、現在の移行元 (S) および移行先 (D) ボリュームのステータスを確認します。コマンド `samcmd y` を使用します。

最初の例では、唯一の移行元および移行先のボリュームにジョブの *end time* が `10/16 12:14` と表示されています。移行元ボリュームのコピーは完了しています。したがって、現在実行中のジョブはありません。

```
root@solaris:~# samcmd y
Migration vsn list   samcmd  version HH:MM:SS month day year
Status: Run Vsns:2 src:1 dest:1 maxcopy:2
ord m ty vsn      start time end time  status  Inodes done/tot  bytes
  0 S li VOLa01  10/16 12:12 10/16 12:14 complete 35023/35023 550.00M
  0 D li VOLa80  10/16 12:12 10/16 12:14 avail          550.00M
```

2 番目の例では、移行元および移行先のボリュームにジョブの *end time* が *none* と表示されています。移行元ボリュームは移行先ボリュームに引き続きコピーされています。したがって、移行ジョブがまだ実行されています。

```
root@solaris:~# samcmd y
Migration vsn list   samcmd  version HH:MM:SS month day year
Status: Run Vsns:2 src:1 dest:1 maxcopy:2
ord m ty vsn      start time end time  status  Inodes done/tot  bytes
  0 S li VOLa02  10/16 12:12 none      copy    0/35023 164.50M
```

```
0 D li VOLa81 10/16 12:12 none copy 148.75M
```

4. 最後に、ライブラリカタログ内のボリウムのリストを確認します。コマンド `samcmd v` を使用します。出力で次のフラグを探します。
 - *R* は、ボリウムが読み取り専用であることを意味します。移行が開始されると、移行元ボリウムは読み取り専用としてマークされます。
 - *S* (移行元) は、データが引き続きこのボリウムからコピーされていることを意味します。
 - *D* (移行先) は、データが引き続きこのボリウムにコピーされていることを意味します。
 - *m* は、移行元ボリウムの移行が完了したことを意味します。
 - *e* は、エラーのため移行元ボリウムが移行に失敗したことを意味します。

この例では、ボリウム `VOLa01` は `VOLa80` に正常に移行しました。ボリウム `VOLa02` は `VOLa81` に引き続き移行しています。移行に失敗しました。

```
root@solaris:~# samcmd v
Robot catalog samcmd version HH:MM:SS month day year
Robot VSN catalog by slot      : eq 800
slot          access time count use flags          ty vsn
count 64
  0    2015/06/29 17:00    1 95% -il---b--Rm-  li VOLa01
  1    2015/07/02 17:43    2 89% -il-o-b--RS-  li VOLa02
  2    2015/07/02 18:31    2 89% -il-o-b--Re-  li VOLa03
  ...
 51    2015/10/16 15:18    2 82% -il-o-b----- li VOLa80
 52    2015/10/16 15:25    2 84% -il-o-b---D-  li VOLa81
```

5. ジョブが実行されている場合は、それらが完了するまで待機します。
6. それ以外の場合は、すでに進行中の移行が存在しないことを確認したら、[ボリウムの移行](#)を実行します。

ボリウムの移行

このセクションの手順では、シェルコマンドプロンプトから `samcmd` コマンドを使用してコマンドを入力する方法について説明します。ただし、すべてのコマン

ドは、`:command` という形式で *samu* インタフェースからも入力できます。ここで、*command* はコマンドの名前です。

1. 現在、Oracle HSM メタデータサーバーに *root* としてログインしていない場合は、ここでログインします。

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

2. 移行元ファイルシステムがマウントされていることを確認します。
3. *migrationd.cmd* ファイルをアクティブにします。コマンド *samcmd migconfig* を使用します。

構成が成功した場合は、「*Configuring migration*」というメッセージが表示され、詳細の参照先のログファイルが示されます。

```
root@solaris:~# samcmd migconfig
samcmd: migconfig: Configuring migration (see /var/opt/SUNWsamfs/sammig/logfile)
root@solaris:~#
```

それ以外の場合、このコマンドはエラーで停止します。この構成コマンドを発行する前に移行プロセスを停止しなかったか、テープボリュームがまだ移行を待機しているときに移行を停止しました。

```
root@solaris:~# samcmd migconfig
samcmd: migconfig: Can't configure migration, migration status is not stop, or
migration job is pending
root@solaris:~#
```

4. 移行の構成を表示します。コマンド *samcmd y* を使用します。

構成が成功した場合は、指定されたボリュームが一覧表示され、移行元ボリュームのステータスは *sched_wait* (スケジュール済み、待機中) になり、移行先ボリュームのステータスは *avail* (使用可能) になります。この例では、構成が成功しました。

```
root@solaris:~# samcmd y
Migration vsn list    samcmd  version HH:MM:SS month day year
```

```
samcmd on hsm61sol
Status: Stop: Waiting for :migstart Vsns:2 src:1 dest:1 maxcopy:2
ord m ty vsn start time end time status Inodes done/tot bytes
 0 S li VOL001 none none sched_wait 0/0 0
 0 D li VOL501 none none avail 0
```

- 構成が成功した場合は、移行を開始します。コマンド `samcmd migstart` を使用します。

```
root@solaris:~# samcmd migstart
samcmd: migstart: State changed to start
root@solaris:~#
```

- 移行のステータスを確認します。コマンド `samcmd x` および `samcmd y` を使用します。

この例では、移行が開始しようとしています。この「*Migration status*」画面は、ジョブステータスが現在 *Run* であること、1 件のコピーが *s* (サーバー) コピーモード (*cmode*) で進行中であること、コピーが 0% 完了していること、0 個の *i* ノードが更新されたこと、および、移行元ボリュームが引き続き *Loading* 中であることを示しています。

```
root@solaris:~# samcmd x
Migration status samcmd version HH:MM:SS month day year
Source Vsns - wait: 0 fsscan: 0 copy: 1 update ino: 0 log: 0 done: 0
Status: Run
source dest cmod perc status
li VOL001 li VOL501 s 0% Loading li.VOL001
```

この「*Migration vsn list*」画面は、2 つのボリュームが現在処理されていることを示しています (移行元 1 つと移行先 1 つ)。両方のボリュームのステータスは現在 *copy* で、これは移行元が移行先にコピーされていることを示しています。この時点で、移行元から移行先に 0 バイトがコピーされており、35023 個の *i* ノードはどれも更新されていません。

```
root@solaris:~# samcmd y
```

```
Migration vsn list  samcmd  version HH:MM:SS month day year
Status: Run  Vsns:2 src:1 dest:1 maxcopy:2
ord m ty vsn      start time  end time  status  Inodes done/tot  bytes
  0 S li VOL001  10/16 12:12 none    copy           0/35023    0
  0 D li VOL501  10/16 12:12 none    copy           0/35023    0
```

7. 移行のステータスを定期的に再確認します。ここでもコマンド `samcmd x` および `samcmd y` を使用します。

この例では、「*Migration status*」画面は、コピーが現在 23% 完了していること、および、560 (0x00000230) テープブロックが移行元から読み取られたことを示しています。

```
root@solaris:~# samcmd x
Migration status  samcmd  version HH:MM:SS month day year
Source Vsns - wait:  0 fsscan: 0 copy: 1 update  ino: 0 log: 0 done:  0
Status: Run
source  dest      cmod perc status
li VOL001 li VOL501 s      24% 0x00000230 blocks read
```

この「*Migration vsn list*」画面は、164.50 メガバイトが移行元ボリユームから読み取られたこと、および、148.75 メガバイトが移行先ボリユームに書き込まれたことを示しています。

```
root@solaris:~# samcmd y
Migration vsn list  samcmd  version HH:MM:SS month day year
Status: Run  Vsns:2 src:1 dest:1 maxcopy:2
ord m ty vsn      start time  end time  status  Inodes done/tot  bytes
  0 S li VOL001  10/16 12:12 none    copy           0/35023  164.50M
  0 D li VOL501  10/16 12:12 none    copy           0/35023  148.75M
```

8. 移行が完了したら、終了ステータスを確認します。コマンド `samcmd x` および `samcmd y` を使用し、移行ログファイルを調べます。

この例では、「*Migration status*」画面に移行元および移行先のボリユームは表示されなくなり、1 件のコピーが完了したことが示されています。移行ステータス

タスはまだ *Run* で、*migidle* または *migstop* コマンドを入力するまで変化しません。

```
root@solaris:~# samcmd x
Migration status   samcmd  version HH:MM:SS month day year
Source Vsns - wait: 0 fsscan: 0 copy: 0 update ino: 0 log: 0 done: 1
Status: Run
source  dest   cmod perc status
```

この「*Migration vsn list*」画面は、550.00 メガバイトが移行元ボリュームから読み取られたこと、および、550.50 メガバイトが移行先ボリュームに書き込まれたことを示しています。35023 個の i ノードすべてが、移行されたアーカイブコピーの新しい場所を反映するように更新されています。

```
root@solaris:~# samcmd y
Migration vsn list   samcmd  version HH:MM:SS month day year
Status: Run Vsns:2 src:1 dest:1 maxcopy:2
ord m ty vsn      start time end   time status  Inodes done/tot   bytes
  0 S li VOL001  10/16 12:12 10/16 12:14 complete 35023/35023 550.00M
  0 D li VOL012  10/16 12:12 10/16 12:14 avail          550.00M
```

移行デーモンのログファイルには、移行の各段階が一覧表示され、最後にサマリーが示されます。この例では、Solaris の *tail* コマンドを使用して、最新のエントリを表示します。

```
root@solaris:~# tail /var/opt/SUNWsamfs/sammig/logfile
date time Info: Schedule: Create VsnList file.
date time Info: Schedule: VsnList file created, source: 1, destination: 1.
date time Info: Schedule: Migration status changed to Start.
date time Info: 'li.VOL001' Filesystem scan: Started
date time Info: 'li.VOL001' Filesystem scan: Completed, total copy bytes: 517.2M,
  inodes: 35023, multi vsn copy: 0, removable-media file: 0, obsolete copy: 0
date time Info: 'li.VOL001' Copy: Started, pid: 2459 destination 'li.VOL012'
date time Info: 'li.VOL001' Copy: Mode - server copy
date time Info: 'li.VOL001' Copy: Server copy started from position 0x4.
```

```

date time Info: 'li.VOL001' Copy: Tar header check started from position 0x4.
date time Info: 'li.VOL001' Copy: Tar header check succeeded, 5 inodes checked, 0
tar header error found.
date time Info: 'li.VOL001' Copy: Completed, pid: 2459, exit status: 12, signal: 0
date time Info: 'li.VOL001' Update inode: Started, source position: 0
date time Info: 'li.VOL001' Update inode: Completed.
date time Info: 'li.VOL001' Log: Started, source position: 0
date time Info: 'li.VOL001' Log: Completed.
date time Summ: 'li.VOL001'
date time Summ: 'li.VOL001' ===== Summary =====
date time Summ: 'li.VOL001' Status:      Complete
date time Summ: 'li.VOL001' Copy mode:  Server copy
date time Summ: 'li.VOL001' Start at:   date time
date time Summ: 'li.VOL001' End at:     date time
date time Summ: 'li.VOL001' Bytes:      550.00M
date time Summ: 'li.VOL001' Archive copies:                35023
date time Summ: 'li.VOL001' Read error copies:              0
date time Summ: 'li.VOL001' Multi vsn copies:                0
date time Summ: 'li.VOL001' Removable-Media file:           0
date time Summ: 'li.VOL001' ---Dest---  ---Bytes---  ---Copies---
date time Summ: 'li.VOL001' li VOL501      550.00M      35023
root@solaris:~#

```

- 最後に、ボリウム移行のログをセキュアな場所に必ずコピーします。

これらのログには、各移行元ボリウムから移行された各アーカイブファイルコピーについて、移行先ボリウムおよび開始位置が記録されます。この情報は、ファイルまたはファイルシステムを回復する必要がある場合に重要です。したがって、Oracle では、これらのファイルのバックアップコピーを回復ポイントおよびアーカイバログファイルとともに保存することを強く推奨しています。7章「構成およびファイルシステムのバックアップ」および『Oracle Hierarchical Storage Manager and StorageTek QFS インストールおよび構成ガイド』の対応する章の説明に従ってください。

移行デーモンは、ユーザーが `migrationd.cmd` ファイルで指定したディレクトリに、移行ログファイルを作成します。移行されたボリュームごとに、`media_type.VSN` という名前のファイルが作成されます。ここでは:

- `media_type` は、移行元メディアの種類を識別する 2 文字のコードです (詳細は、付録A「装置タイプの用語集」を参照)。
- `VSN` は、移行元ボリュームを識別する一意のボリュームシリアル番号です。

この例では、指定されたログディレクトリ `/var/adm/hsm_migration_logs/` にあるボリュームログを、NFS マウントされたリモートファイルシステム上にある、ファイルシステムの回復リソースを保存するディレクトリにコピーします。

```
root@solaris:~# ls /var/adm/hsm_migration_logs/
li.VOL001 li.VOL002 li.VOL003 li.VOL004 li.VOL005 li.VOL006 ... ti.801 ...
root@solaris:~# cp /var/adm/hsm_migration_logs/*.* /zfs/recover/hsmfs1/2015mig/
```

10. すべてのファイルが再度アーカイブされたら、要件に従ってテープを処分します (「移行後の古いメディアの処分」を参照)。
11. ここで停止します。移行が完了しました。

ファイルのステージングおよび交換用メディアへの再アーカイブ

ステージングと再アーカイブの方式を使用してアーカイブファイルを古いメディアから新しいメディアに移行するには、通常ファイルシステム操作を妨害することなく、移行するファイルを特定し、それをディスクキャッシュにステージングし、新しいメディアに書き込む必要があります。この章では、プロセス内の次のステージについて説明します。

- [使用可能なリソースの評価](#)
- [新しいメディアを使用するためのアーカイブ処理の構成](#)
- [交換用メディアへのデータの移行](#)

使用可能なリソースの評価

ステージングと再アーカイブ処理の詳細は、使用可能なディスクストレージ容量と使用可能なリムーバブルメディアドライブ数の 2 つの要素によって大きく異なります。メディアの移行中、Oracle HSM ステージャーは、古いメディアフォーマットを読み取ることができるドライブに古いリムーバブルボリュームをロードし、アー

カイクされたファイルをディスクキャッシュに復元します。そのあと、Oracle HSM アーカイバが、新しいメディアフォーマットを書き込むことができるドライブを使用して、新しいリムーバブルボリュームにファイルを再度アーカイブします。そのため、特定のテープボリューム上にあるすべてのファイルを一度にディスクにステージングして、すぐに新しいメディアにアーカイブする方法が理想的です。

これを行うには、移行中に大容量のリソースを割り当てる必要があります。

- テープ全体の容量と同等のディスク領域
- 古いテープ形式を読み込むドライブの排他的使用
- 新しい形式を書き込むドライブの排他的使用。

移行が完了するまでファイルシステムを休止できる場合は、上記のことは問題になりません。ただし、進行中のファイルシステムおよびアーカイブ操作を必要以上に妨害することなく、本稼動設定でデータを移行するには、検討が必要となります。ディスク領域またはテープドライブ数が不足している場合は、移行用に合理的に確保しておくことができるリソースを確認してから、移行プロセスを調整する必要があります。そのため、次のように進めます。

1. 通常のファイルシステム操作を妨害せずに、移行用に使用できるディスクキャッシュの量を評価します。
2. 移行専用で使用できるテープドライブの数を評価します。

使用可能なテープドライブの数が限定されている場合は、移行プロセスで通常の操作が妨害されないように、ステージングおよびアーカイブのプロセスを調整することを計画します。

3. 上記の見積もりに基づいて、ステージングおよびアーカイブのパラメータを決定します。使用可能なディスク領域に常時保持される移行ファイルの最大数、およびファイルをキャッシュから新しいメディアに移動できる最大速度を決定します。
4. リソースの見積もりを行なったら、移行後の古いメディアの処分を計画します。

新しいメディアを使用するためのアーカイブ処理の構成

新しいメディアを `archiver.cmd` ファイルに追加し、アーカイブのコピーディレクティブを変更して、1つのコピーは常に新しいメディアを使用して行われるようにします。

1. テキストエディタで `/etc/opt/SUNWsamfs/archiver.cmd` ファイルを開きます。

アーカイブポリシーには、2つのコピーが指定されています。どちらのコピーも、交換対象のメディアタイプに書き込まれます。この例では、`vi` エディタでファイルを開きます。DLT カートリッジ (タイプ `lt`) を交換します。

```
root@solaris: vi /etc/opt/SUNWsamfs/archiver.cmd
# =====
# /etc/opt/SUNWsamfs/archiver.cmd
# -----
...
# -----
# VSN Directives
vsns
allfiles.1 lt .*
allfiles.2 lt .*
endvsns
```

2. コピー 2 に対するディレクティブで、指定されたメディアタイプを新しいメディアの識別子に変更し、ファイルを保存して、テキストエディタを閉じます。

この例では、データを古い DLT テープから新しい LTO カートリッジに移行します。したがって、コピー 2 で、古いメディアタイプ `lt` (DLT) を `li` (LTO) に変更します。

```
root@solaris: vi /etc/opt/SUNWsamfs/archiver.cmd
# =====
# /etc/opt/SUNWsamfs/archiver.cmd
# -----
...
# -----
# VSN Directives
vsns
allfiles.1 lt .*
allfiles.2 li .*
endvsns
```



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

3. *archiver.cmd* ファイルに構文エラーがないかどうかを確認します。コマンド *archiver -lv* を実行して、エラーが見つからなくなるまでエラーを修正します。

archiver -lv コマンドは、ファイルを 1 行ずつ出力します。エラーが発生すると、エラーが発生したところで実行が停止します。

```
root@solaris:~# archiver -lv
Reading '/etc/opt/SUNWsamfs/archiver.cmd'.
1: # =====
2: # /etc/opt/SUNWsamfs/archiver.cmd
3: # -----
4: # Global Directives
5: logfile = /var/opt/SUNWsamfs/archiver.log
6: # -----
7: # File System Directives:
8: fs = samqfsm
9: all .
10: 1 5m ...
root@solaris:~#
```

4. 変更した *archiver.cmd* ファイルにエラーがなくなったら、コマンド *samd config* を使用して現在の構成にファイルをロードします。

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

5. 次に、カートリッジからカートリッジにデータを移行します。

交換用メディアへのデータの移行

ステージングとアーカイブによるデータ移行方式では、*sfind* (GNU *find* コマンドの Oracle HSM 拡張機能) を使用します。*sfind* コマンドは、指定されたテープポ

リユーム上でファイルを検索し、見つかったすべてのファイルに対して、`stage` および `rearchive` コマンドを起動する際に使用されます。

`sfind`、`stage`、または `rearchive` コマンドを使い慣れていない場合は、それぞれのマニュアルページを確認するようにしてください。次に、移行する必要があるデータを保持しているテープカートリッジごとに、次のように進めます。

別のカートリッジへのデータの移行

1. ファイルシステムホストに `root` としてログインします。

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

2. 移行するファイルを保持しているファイルシステムのマウントポイントディレクトリに移動します。

この例では、`/hsm/hsmfs1` でマウントされた `hsmfs1` ファイルシステムに格納されているファイルのアーカイブコピーを移行します。

```
root@solaris:~# cd /hsm/hsmfs1
root@solaris:~#
```

3. テープボリュームを選択します。

メディアタイプからメディアタイプにデータを移行する場合は、一度に1つのボリュームを操作します。次の例では、ボリュームシリアル番号 `VOL008` を操作します。

4. まず、選択したボリュームに、損傷して正常にステージングできないファイルがないかどうかを検索します。Oracle HSM コマンド `sfind . -vsn volume-serial-number -damaged` を使用します。ここで `volume-serial-number` は、ライブラリ内のボリュームを一意に識別する英数字の文字列です。

この例では、現在の作業ディレクトリ (`.`) から検索を開始します。`-vsn` パラメータは、現在のテープ `VOL008` 上で見つかったファイルに検索を限定します。`-damaged` フラグは、正常にステージングできないファイルに検索を限定します。

```
root@solaris:~# sfind . -vsn VOL008 -damaged
```

5. *sfind* による損傷したファイルの検索で結果が返されたら、ファイルの修正を試行します。コマンド *undamage -m media-type -vsn volume-serial-number file* を使用します。ここでは:
- *media-type* は、付録A「装置タイプの用語集」に一覧表示されている2文字のメディアタイプコードの1つです。
 - *volume-serial-number* は、ボリュームを一意に識別する英数字の文字列です。
 - *file* は、破損したファイルのパスおよび名前です。

一時的な入出力エラーにより、コピーに破損のマークが付くことがあります。この状態は、Oracle HSM *undamage* コマンドでクリアします。この例では、アーカイブファイルコピー */hsm/hsmfs1/data0008/20131025DAT* が破損していると報告されます。そのため、この破損を修復し、破損ファイルの検索を再試行します。

```
root@solaris:~# sfind . -vsn VOL008 -damaged
/hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# undamage -m lt -vsn VOL008 /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind . -vsn VOL008 -damaged
```

6. *sfind* コマンドにより再度このファイルが破損されていると一覧表示された場合、そのコピーは使用できません。アーカイブに、このファイルの破損していない別のコピーがないか探します。使用可能なコピーを一覧表示するには、コマンド *sls -D file* を使用します。ここで *file* は、ファイルのパスおよび名前です。検出されたコピーのステータスをチェックするには、コマンド *sfind file -vsn volume-serial-number* を使用します。

この例では、*undamage* コマンドはコピーを修正できませんでした。そのため、*sls* を使用して、ファイル */hsm/hsmfs1/data0008/20131025DAT* のすべてのコピーを一覧表示します。

```
root@solaris:~# undamage -m lt -vsn VOL008 /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind . -vsn VOL008 -damaged
/hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sls -D /hsm/hsmfs1/data0008/20131025DAT
20131025DAT:
```

```

mode: -rw-r--r--  links:   1  owner: root      group: other
      length:    319279  admin id:    7  inode: 1407.5
      project: system(0)
      offline; archdone; stage -n;
      copy 1: ---- May 21 07:12    1e4b1.1    lt VOL008
      copy 2: ---- May 21 10:29    109c6.1    lt VOL022
...

```

テープボリューム *VOL022* は、ファイルの 2 番目のコピーを保持しています。そのため、*sfind* で 2 番目のコピーをチェックします。

```

root@solaris:~# sfind /hsm/hsmfs1/data0008/20131025DAT -vsn VOL022 -damaged

```

7. コピーを使用できず、破損していないファイルのコピーが 1 つ存在する場合は、そのファイルを再アーカイブします。アーカイブに 2 つの良好なコピーが保持されたら、次に、破損したコピーをアーカイブ解除します。

この例では、ボリューム *VOL008* 上のファイル */hsm/hsmfs1/data0008/20131025DAT* のコピー 1 が使用不能ですが、*sfind* コマンドでコピー 2 の破損が見つかりませんでした。そのため、*archive* コマンドに *-c* オプションを指定して発行し、有効なコピー 1 を作成してから、ボリューム *VOL008* 上の破損したコピーをアーカイブ解除します。

```

root@solaris:~# sfind /hsm/hsmfs1/data0008/20131025DAT -vsn VOL022 -damaged
root@solaris:~# archive -c 1 /hsm/hsmfs1/data0008/20131025DAT
...
root@solaris:~# unarchive -m lt -vsn VOL008 /hsm/hsmfs1/data0008/20131025DAT

```

8. 使用可能なコピーが存在しない場合は、キャッシュ内にファイルが存在するかどうかを確認します。コマンド *sfind . -vsn volume-serial-number -online* を使用します。

この例では、ボリューム *VOL008* のコピー 1 とボリューム *VOL022* のコピー 2 がどちらも破損していて使用できません。そのため、ディスクキャッシュで、ファイルがオンラインで使用可能かどうかを確認します。

```
root@solaris:~# undamage -m lt -vsn VOL008 /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind . -vsn VOL008 -damaged
/hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# undamage -m lt -vsn VOL022 /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind /hsm/hsmfs1/data0008/20131025DAT -vsn VOL022 -damaged
/hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind /hsm/hsmfs1/data0008/20131025DAT -online
```

9. 使用可能なコピーが存在しないが、キャッシュ内にファイルが存在する場合は、そのファイルをアーカイブします。アーカイブに2つの良好なコピーが保持されたら、次に、破損したコピーをアーカイブ解除します。

この例では、ボリューム *VOL008* のコピー 1 およびボリューム *VOL022* のコピー 2 のいずれも使用できないため、*archive* コマンドを発行して2つの有効なコピーを作成してから、ボリューム *VOL008* 上の破損したコピーをアーカイブ解除します。

```
root@solaris:~# undamage -m lt -vsn VOL008 /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind . -vsn VOL008 -damaged
/hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# undamage -m lt -vsn VOL022 /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind /hsm/hsmfs1/data0008/20131025DAT -vsn VOL022 -damaged
/hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind /hsm/hsmfs1/data0008/20131025DAT -online
/hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# archive /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# unarchive -m lt -vsn VOL008 /hsm/hsmfs1/data0008/20131025DAT
```

10. 使用可能なコピーが存在せず、ディスクキャッシュ内にファイルが存在しない場合は、データが失われた可能性が高くなります。データがクリティカルである場合は、データ回復の専門家がいる会社に連絡して支援を求めてください。それ以外の場合、破損したコピーをアーカイブ解除します。

この例では、ボリューム *VOL008* のコピー 1 とボリューム *VOL022* のコピー 2 がどちらも使用できません。*sfind* コマンドは、ディスクキャッシュ内でファイル

を見つけられませんでした。データは重要ではありません。そのため、ボリューム *VOL008* 上の破損したコピーをアーカイブ解除します。

```
root@solaris:~# undamage -m lt -vsn VOL008 /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind . -vsn VOL008 -damaged
/hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# undamage -m lt -vsn VOL022 /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind /hsm/hsmfs1/data0008/20131025DAT -vsn VOL022 -damaged
/hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# sfind /hsm/hsmfs1/data0008/20131025DAT -online
root@solaris:~# archive /hsm/hsmfs1/data0008/20131025DAT
root@solaris:~# unarchive -m lt -vsn VOL008 /hsm/hsmfs1/data0008/20131025DAT
```

11. *sfind* による損傷したファイルの検索で結果が返されない場合は、ファイルを現在のテープからディスクキャッシュにステージングします。コマンド *sfind . -vsn volume-serial-number -offline -exec stage {}/;* を使用します

-vsn パラメータは、現在のテープで見つかったファイルに検索を限定します (必ず一度に1つのテープのデータを移行します)。

-offline パラメータは、データが上書きされないように、キャッシュ内にまだ存在しないファイルに *sfind* の出力をさらに限定します。

-exec stage {}/; 引数は、*sfind* が返す各パスとファイル名を取り、それを Oracle HSM *stage* コマンドの引数として使用します。次に、*stage* コマンドが、指定したファイルをディスクキャッシュに復元します。対象のすべてのファイルがステージングされるまでプロセスは繰り返されます。

この例では、*sfind -vsn VOL008 -damaged* コマンドが出力を返しません。そのため、*sfind* を使用して、*VOL008* 上で見つかったファイルのうち、キャッシュ内にないすべてのファイルをステージングします。

```
root@solaris:~# sfind . -vsn VOL008 -damaged
root@solaris:~# sfind . -vsn VOL008 -offline -exec stage {}/;
```

12. ファイルがテープからステージングされたら、ファイルを選択的に再度アーカイブします。コマンド *sfind . -vsn volume-serial-number -online -exec*

`rearch -r -m media-type {}/;` を使用します。ここで `media-type` は、移行元メディアのタイプです。

`-vsn` パラメータは、現在のテープでも見つかったファイルに検索を限定します (必ず一度に1つのテープのデータを移行します)。

`-online` パラメータは、データが上書きされないように、キャッシュ内に存在するファイルに `sfind` の出力をさらに限定します。

`-exec rearch -r -m media-type {}/;` 引数は、`sfind` が返す各パスとファイル名を取り、それを Oracle HSM `rearch -r -m media-type` コマンドの引数として使用します。`-r` 引数は、サブディレクトリを介してプロセスを再帰的に実行します。`-m` 引数は、ソースメディア上にあるファイルのみを再アーカイブします。

この例では、`-vsn` パラメータの値は `VOL008`、`-m` パラメータの値は DLT メディアを表す `lt` に指定されています。

```
root@solaris:~# sfind . -vsn VOL008 -online -exec rearch -r -m lt {}/;
```

13. `sfind` による検索でファイルが見つからなくなるまで、上記の手順を繰り返します。
14. すべてのファイルが再度アーカイブされたら、計画通りにテープを処分します (「[移行後の古いメディアの処分](#)」を参照)。
15. すべての古いメディアから新しいメディアにデータが移行されるまで、この手順を繰り返します。

移行後の古いメディアの処分

移行が完了しても、古いメディアの価値がすべて失われるわけではありません。したがって、それらの処分方法を慎重に検討してください。

- 少なくとも、新しい回復ポイントファイルが十分に蓄積され、新しい交換用メディアのみを使用してファイルシステム内のすべてのファイルを回復できるようになるまで、古いメディアを保持してください。
- ストレージ領域に余裕がある場合は、古いメディアを無期限に保持します。互換性のあるドライブが使用可能なかぎり、古いメディアはバックアップと回復の貴重なリソースになり得ます。

- ライブラリ容量に余裕がない場合は、古いメディアをエクスポートし、オフサイトのストレージに保持します。
- 古いメディアが再利用可能な場合は、そこに含まれているデータが有用でなくなっていることが確実であれば、古いボリュームのラベルを付け直します。たとえば、古い Oracle StorageTek T10000C ドライブ用のメディアのラベルを付け直し、新しい T10000D ドライブで使用することもできます。
- それ以外の場合は、古いボリューム上のデータにもメディアにも価値が残っていないければ、ボリュームをライブラリからエクスポートし、適切に処分します。

付録A 装置タイプの用語集

マスター構成ファイル (*mcf*) の *Equipment Type* フィールドの値は、デバイスおよび Oracle Hierarchical Storage Manager and StorageTek QFS Software 内のデバイス構成を特定します。装置タイプは2文字のコードで表されます。この用語集に記載されているコードは、サンプルを使用する際や既存の *mcf* (詳細は、*mcf(4)* のマニュアルページを参照してください) の内容を解釈する際のクイックリファレンスとして使用できます。

便宜上、コードを3つのセクションに分けてアルファベット順に記載しています。

- 推奨される装置およびメディアのタイプ
- その他の装置タイプとメディアタイプ

推奨される装置およびメディアのタイプ

このセクションでは、通常必要になる装置コードについて説明します。汎用装置コード (*rb*、*tp*、*od*) を表すコード、およびネットワーク接続ライブラリインタフェースと Oracle HSM 履歴を表すコードです。

汎用装置コード *rb*、*tp*、および *od* は、SCSI 接続のライブラリ、テープドライブ、および光学ディスクデバイス全般に使用できる推奨の装置タイプです。汎用装置タイプを指定すると、Oracle HSM が SCSI ベンダーコードに基づいて正しいタイプを自動的に設定します。

gXXX

XXX は [0-127] の範囲の整数で、*ma* ディスクキャッシュファミリセットの一部であるディスクデバイスのストライプ化グループです。

hy

オプションの Oracle HSM 履歴仮想ライブラリです。メディアカタログが格納されますが、ハードウェアへの関連付けはありません。エクスポートしたメディアの追跡に使用します。

ma

1つ以上の専用の *mm* ディスクデバイス上のファイルシステムメタデータが保持される高性能の QFS ファイルシステム。ファイルデータは別の *md*、*mr*、または *gXXX* データデバイスに格納されます。

md

ma ファイルシステムのファイルデータや *ms* ファイルシステムのデータおよびメタデータを格納するディスクデバイス。 *md* デバイスはファイルデータを、小さい 4K バイトのディスク割り当て単位 (DAU)、および大きい 16-、32-、または 64K バイトの DAU で格納します。 DAU のデフォルトは 64K バイトです。

mm

高パフォーマンス *ma* ファイルシステムのファイルシステムメタデータを格納するディスクデバイス。

mr

ma ファイルシステムのファイルデータを格納するディスクデバイス。 *mr* デバイスはファイルデータを、8-65528K バイトの範囲の 8K バイトの倍数で自由に調整できる大きなディスク割り当て単位 (DAU) で格納します。 DAU のデフォルトは 64K バイトです。

ms

Oracle HSM ファイルシステムで、ファイルシステムのメタデータをファイルデータを格納しているのと同じデバイスに格納します。

od

SCSI 接続光学ディスク。 Oracle HSM は、SCSI ベンダーコードを使用して適切な装置タイプを自動的に設定します。

rb

SCSI 接続テープライブラリ。 Oracle HSM は、SCSI ベンダーコードを使用して適切な装置タイプを自動的に設定します。

rd

SAM-Remote 疑似デバイス。 マスター構成ファイル (*mcf*) では、対応する「*Equipment Identifier*」フィールドに、疑似デバイスへのパス (*/dev/samrd/rd2* など) が含まれている必要があります。 対応する「*Family Set*」フィールドには、SAM-Remote サーバーのホスト名が含まれている必要があります。

sc

SAM-Remote クライアントシステム。 マスター構成ファイル (*mcf*) では、対応する「*Equipment Identifier*」フィールドに、クライアントの SAM-Remote クライアント構成ファイルへのパスが含まれている必要があります。 対応する「*Family Set*」フィールドには、サーバーのファミリーセット名が含まれている必要があります。「*Additional Parameters*」フィールドには、クライアントのライブラリカタログファイルへのフルパスが含まれている必要があります。

sk

ネットワーク接続ライブラリとの Oracle StorageTek ACSLS インタフェース。 マスター構成ファイル (*mcf*) 内の対応する「*Equipment Identifier*」フィールドに、ACSLS インタフェースのパラメータファイルへのパスが含まれている必要があります。 詳細は、*stk(7)* のマニュアルページを参照してください。

ss

SAM-Remote サーバー。 マスター構成ファイル (*mcf*) 内の対応する *Equipment Identifier* フィールドに、SAM-Remote サーバー構成ファイルへのパスが含

まれている必要があります。対応する *Family Set* フィールドにはサーバーのファミリーセット名が含まれている必要があります、これは、クライアント上の *mcf* の *Family Set* フィールドで使用されている名前と一致している必要があります。

tp

SCSI 接続テープドライブ。Oracle HSM は、SCSI ベンダーコードを使用して適切な装置タイプを自動的に設定します。ただし、*li* や *ti* などの具体的な装置コードを使用する場合は、一貫してそうする必要があります。たとえば、*mcf* ファイルで *li* (LTO) テープ装置を指定する場合、同じ装置を *archiver.cmd* ファイルで *tp* 装置として参照することはできません。

その他の装置タイプとメディアタイプ

このセクションに表示されている装置タイプもサポートされます。

ほとんどの場合、汎用装置タイプ *rb*、*tp*、および *od* を使用して、SCSI 接続のライブラリ、テープドライブ、および光ディスクデバイスを識別することをお勧めしています。汎用装置タイプは、ハードウェアを SCSI ベンダー ID を使用して動的に特定するよう Oracle HSM に指示します。次のタイプコードは、あるメディアタイプから別のメディアタイプに移行する際に不可欠であり、管理のために役に立つことがあります。ただし、たとえばこれらをマスター構成ファイル (*mcf*) で使用すると、ある時点で実際のハードウェアと一致しない可能性がある静的な装置構成がハードコーディングされます。

ac

Sun 1800、3500、または L11000 テープライブラリ。

at

Sony AIT-4 または AIT-5 テープドライブ。

cy

Cygnnet 光学ディスクライブラリ。

d3

StorageTek D3 テープドライブ。

dm

Sony DMF ライブラリ。

ds

DocuStore または Plasmon 光学ディスクライブラリ。

dt

DAT 4-mm テープドライブ。

e8

Exabyte X80 ライブラリ。

fd

Fujitsu M8100 128 トラックテープドライブ。

h4

HP SL48 または SL24 ライブラリ。

hc

Hewlett Packard L9-/L20-/L60 シリーズライブラリ。

i7

IBM 3570 テープドライブ。

ic

IBM 3570 メディアチェンジャー。

il

IBM 3584 テープライブラリ。

li

LTO-3 以降のテープドライブ。

lt

Digital Linear Tape (DLT)、Super DLT、DLT-S4 テープドライブ。

me

Metrum ライブラリ。

mf

IBM マルチファンクション光学ドライブ。

mo

5.25 インチ消去可能光学ドライブ。

o2

12 インチ WORM ドライブ。

ov

Overland Data Inc. Neo シリーズテープライブラリ。

pd

Plasmon D シリーズ DVD-RAM ライブラリ。

q8

Qualstar 42xx、62xx、または 82xx ライブラリ。

s3

StorageTek SL3000 ライブラリ。

s9

Oracle StorageTek 97xx シリーズライブラリ。

se

StorageTek 9490 テープドライブ。

sf

StorageTek T9940 テープドライブ。

sg

StorageTek 9840C 以降のテープドライブ。

sl

Spectra Logic または Qualstar テープライブラリ。

st

StorageTek 3480 テープドライブ。

ti

StorageTek T10000 (Titanium) テープドライブ。

vt

Metrum VHS (RSP-2150) テープドライブ。

wo

5.25 インチ光学 WORM ドライブ。

xt

Exabyte (850x) 8-mm テープドライブ。

付録B メディアステータスフラグ

メディアフラグには次の意味があります。

- *A* は、スロットで監査が必要なことを意味します。
- *C* は、スロットにクリーニングカートリッジが含まれていることを意味します。
- *D* は、ボリュームがメディアの移行先であることを意味します。
- *E* は、ボリュームが不良であるか、クリーニングメディアが期限切れであることを意味します。
- *L* は、ボリュームが Linear Tape File System (LTFS) ボリュームであることを意味します。
- *N* は、ボリュームが Oracle HSM 形式ではないことを意味します。
- *R* は、ボリュームが読み取り専用であることを意味します (ソフトウェアフラグ)。
- *S* は、ボリュームがメディアの移行元であることを意味します。
- *U* は、ボリュームが使用不可であることを意味します。
- *W* は、ボリュームが物理的に書き込み保護されていることを意味します。
- *X* は、スロットがエクスポートスロットであることを意味します。
- *b* は、ボリュームにバーコードがあることを意味します。
- *c* は、ボリュームのリサイクルがスケジュールされていることを意味します
- *f* は、ボリュームに空きがないかボリュームが壊れていることがアーカイバで検出されたことを意味します。
- *d* は、ボリュームに重複するボリュームシリアル番号 (VSN) があることを意味します
- *l* は、ボリュームにラベルが付けられていることを意味します。
- *o* は、スロットが占有されていることを意味します。
- *p* は、ボリュームに優先度の高いボリュームがあることを意味します。
- *-* は、表示で使用されているときは、対応するフラグが設定されていないことを意味します。

たとえば、`samcmd v` は、カタログされた各ボリュームのメディアフラグを含むカタログ情報を一覧表示します。

```
root@solaris:~# samcmd v 800
Robot catalog samcmd      6.1      16:45:25 Feb 14 2016
samcmd on samqfshost      count 32
Robot VSN catalog by slot : eq 800
slot      access time count use  flags      ty vsn
  0      2014/03/14 11:23 875  0%  -il-o-b----- li VOL001
  1      2014/03/13 17:54 866  0%  -il-o-b----- li VOL002
  2      2014/03/14 11:26   3  0%  -il-o-b----- li VOL003
  3      2014/03/14 10:33   3  0%  -il-o-b----- li VOL004
  4      2014/03/14 11:34   5  0%  -il-o-b----- li VOL005
  5      2014/03/14 11:32   2  0%  -ilEo-b----f  li VOL006 MEDIA ERROR
  6      2014/03/13 18:07   2  0%  -il-o-b----- li VOL007
  7      2014/03/13 18:07   1  0%  -il-o-b----- li VOL008
 10      2014/03/13 18:10   2  0%  -il-o-b----- li VOL011
```


付録C 共有ファイルシステムでのマウントオプション

Oracle Hierarchical Storage Manager and StorageTek QFS Software 共有ファイルシステムは、いくつかのマウントオプションを使用してマウントできます。この章では、これらのオプションのいくつかについて、その役割のコンテキスト内で説明します。

共有ファイルシステムのマウントオプション

ほとんどのマウントオプションは、`mount` コマンドを使用するか、`/etc/vfstab` ファイルに入力するか、または `samfs.cmd` ファイルに入力することによって指定できます。たとえば、次の `/etc/vfstab` ファイルには、共有ファイルシステムのためのマウントオプションが含まれています。

```
sharefs - /sfs samfs - no shared,mh_write
```

いくつかのマウントオプションは、`samu` オペレータユーティリティを使用して動的に変更できます。これらのオプションの詳細は、『Oracle Hierarchical Storage Manager and StorageTek QFS `samu` コマンドリファレンス』を参照してください。

これらのマウントオプションの詳細は、`mount_samfs` のマニュアルページを参照してください。

bg: バックグラウンドでのマウント

`bg` マウントオプションは、最初のマウント操作が失敗した場合は、それ以降のマウントの試行をバックグラウンドで実行することを指定します。デフォルトでは、`bg` は有効ではなく、マウント試行はフォアグラウンドで継続されます。

retry: ファイルシステムのマウントの再試行

`retry` マウントオプションは、システムでファイルシステムのマウントを試行する回数を指定します。デフォルトは 10000 です。

shared: Oracle HSM 共有ファイルシステムの宣言

shared マウントオプションは、ファイルシステムを Oracle HSM 共有ファイルシステムにすることを宣言します。ファイルシステムが Oracle HSM 共有ファイルシステムとしてマウントされるようにするには、このオプションを */etc/vfstab* ファイル内に指定する必要があります。このオプションが *samfs.cmd* ファイル内、または *mount* コマンド上に存在してもエラー条件は発生しませんが、ファイルシステムは共有ファイルシステムとしてマウントされません。

minallocsz および maxallocsz: 割り当てサイズの調整

mount コマンドの *minallocsz* および *maxallocsz* オプションは、容量を K バイト単位で指定します。これらのオプションは、最小のブロック割り当てサイズを設定します。ファイルサイズが大きくなる場合は、追加リースが認められると、メタデータサーバーによってブロックが割り当てられます。この割り当ての初期サイズを指定するには、*-o minallocsz=n* を使用します。メタデータサーバーは、アプリケーションのアクセスパターンに応じてブロック割り当てのサイズを *-o maxallocsz=n* の設定まで増やすことができますが、この値を超えることはできません。

mount コマンド行のこれらの *mount* オプションは、*/etc/vfstab* ファイルまたは *samfs.cmd* ファイル内に指定できます。

rdlease、wrlease、および aplease: Oracle HSM 共有ファイルシステムでのリースの使用

デフォルトでは、ホストがファイルを共有するときに、入出力 *lease* をそれ自体およびそのクライアントに発行すると、Oracle HSM メタデータサーバーでファイルシステムの整合性が保持されます。リースによって、指定された期間内でファイルを操作するためのアクセス権が共有ホストに付与されます。*read lease* を使用すると、ホストがファイルデータを読み取ります。*write lease* を使用すると、ホストが既存のファイルデータを上書きします。*append lease* を使用すると、ホストがファイルの末尾に追加データを書き込みます。メタデータサーバーは、必要に応じてリースを更新できます。

したがって、Oracle HSM 共有ファイルシステムに対する読み取りおよび書き込みは、データに対する POSIX の動作に類似しています。ただし、メタデータに対しては、アクセス時間が変化しても、ほかのホストにはすぐにわからないことがあります。

す。ファイルへの変更は、書き込みリースの最後にディスクにプッシュされます。読み取りリースが取得されると、新しく書き込まれたデータを表示できるように、システムは期限切れのキャッシュページをすべて無効にします。

次のマウントオプションは、リースの期間を設定します。

- `-o rdlease=number-seconds` は、読み取りリースの最大時間 (秒単位) を指定します。
- `-o wrlease=number-seconds` は、書き込みリースの最大時間 (秒単位) を指定します。
- `-o aplease=number-seconds` は、追加リースの最長時間 (秒単位) を指定します。

これら3つのいずれの場合も、`number-seconds` は [15-600] の範囲の整数です。各リースのデフォルトの時間は 30 秒です。リースが有効な場合、ファイルを切り捨てることはできません。これらのリースの設定の詳細は、`mount_samfs` のマニュアルページを参照してください。

現在のメタデータサーバーが停止したためにメタデータサーバーを変更する場合は、リース時間を切り替え時間に加える必要があります。これは、代替メタデータサーバーが制御を引き継ぐには、その前にすべてのリースが期限切れになっていることが必要であるためです。

リース時間を短く設定しておくこと、リースが期限切れになるごとに更新する必要があるため、クライアントホストとメタデータサーバーの間のトラフィックが増加します。

mh_write: マルチホスト読み取りと書き込みの有効化

`mh_write` オプションでは、複数ホストから同一ファイルへの書き込みアクセスが制御されます。メタデータサーバーホスト上でマウントオプションとして `mh_write` が指定されている場合は、Oracle HSM 共有ファイルシステムにより、複数のホストから同じファイルへの同時の読み取りと書き込みが可能になります。メタデータサーバーホストで `mh_write` を指定しないと、同時にファイルに書き込みができるホストは1つだけになります。

デフォルトでは、`mh_write` は無効になっており、`wrlease` マウントオプションの期間中ファイルに書き込めるのは1つのホストだけです。`mh_write` オプションが有効

になった状態で Oracle HSM 共有ファイルシステムがメタデータサーバー上にマウントされている場合は、複数のホストから同じファイルへの同時の読み取りと書き込みを実行できます。

メタデータサーバー上で *mh_write* が有効になっている場合は、Oracle HSM で次のことがサポートされます。

- 複数の読み取りホストとページ入出力
- 複数の読み取りホストおよび書き込みホストと、書き込みがあった場合にのみ直接入出力
- 1つの追加ホスト(その他のホストは読み取りまたは書き込みを行う)と、書き込みがあった場合にのみ直接入出力。

mh_write オプションを使用してファイルシステムをマウントしても、ロック動作は変わりません。ファイルロックは、*mh_write* が有効かどうかには関係なく同じ動作を行います。ただし、その他の点では、動作は一定ではない可能性があります。読み取りと書き込みが同時にあった場合、Oracle HSM 共有ファイルシステムは、ファイルへのすべてのホストアクセスに直接入出力を使用します。そのため、ページ整合入出力がその他のホストにただちに表示されるはずですが、ただし、非ページ整合入出力では期限切れのデータが表示されたり、場合によってはファイルに書き込まれたりしますが、これは、このような状況を防止している通常のリースメカニズムが無効になるためです。

このため、複数のホストが同じファイルに同時に書き込む必要がある場合、およびホストされているアプリケーションがページ整合入出力を実行し、競合する書き込みを調整する場合にかぎり、*mh_write* オプションを指定してください。それ以外の場合は、データの不一致が発生する可能性があります。*mh_write* を付けて *flock()* を使用してホスト間を調整しても、整合性は保証されません。詳細については、*mount_samfs* のマニュアルページを参照してください。

min_pool: 並行スレッドの最小数の設定

min_pool マウントオプションは、Oracle HSM 共有ファイルシステムの並行スレッドの最小数を設定します。Oracle Solaris システムでのデフォルト設定は *min_pool=64* です。この設定は、Oracle Solaris ではスレッドプール内にアクティブスレッドが少なくとも 64 個存在することを示します。共有ファイルシステムのアクティビティに応じて、*min_pool* の設定を [8-2048] の範囲の任意の値に調整できます。

`min_pool` マウントオプションは `samfs.cmd` ファイルに設定してください。これを `/etc/vfstab` ファイル内か、またはコマンド行で設定した場合は無視されます。

meta_timeo: キャッシュされた属性の保持

`meta_timeo` マウントオプションは、システムがメタデータ情報に対するチェックを待つ間隔の長さを決定します。デフォルトでは、システムはメタデータ情報を 3 秒ごとにリフレッシュします。たとえば、新しく作成されたファイルがいくつか含まれている共有ファイルシステムで `ls` コマンドを入力すると、3 秒が経過するまですべてのファイルに関する情報が返されない可能性があります。このオプションの構文は `meta_timeo=seconds` で、`seconds` は `[0-60]` の範囲の整数です。

stripe: ストライプ化割り当ての指定

デフォルトでは、共有ファイルシステム内のデータファイルは、ラウンドロビン式ファイル割り当て方式を使用して割り当てられます。ファイルデータが複数のディスクにわたってストライプ化されるように指定するには、メタデータホストとすべての潜在的なメタデータホスト上で `stripe` マウントオプションを指定できます。デフォルトでは、非共有ファイルシステムのファイルデータは、ストライプ化方式で割り当てられることに注意してください。

ラウンドロビン式割り当てでは、ファイルは、各スライスまたはストライプ化グループ上にラウンドロビン式で作成されます。1 つのファイルの最大のパフォーマンスは、スライスまたはストライプ化グループの速度になります。ファイル割り当て方式の詳細は、『Oracle Hierarchical Storage Manager and StorageTek QFS インストールおよび構成ガイド』(Oracle HSM お客様向けドキュメントライブラリ、docs.oracle.com/en/storage) を参照してください。

sync_meta: メタデータが書き込まれる頻度の指定

`sync_meta` オプションを `sync_meta=1` または `sync_meta=0` に設定できます。

デフォルト設定は `sync_meta=1` です。これは、メタデータが変更されるたびに Oracle HSM 共有ファイルシステムがファイルのメタデータをディスクに書き込むことを示します。この設定によってデータのパフォーマンスが低下しますが、データの整合性は保証されます。メタデータサーバーを変更する場合は、この設定が有効である必要があります。

`sync_meta=0` を設定した場合、Oracle HSM 共有ファイルシステムは、メタデータをバッファーに書き込んでからディスクに書き込みます。この遅延書き込みによって

より高いパフォーマンスが実現されますが、マシンの予定外の停止が発生したあとのデータの整合性は低下します。

worm_capable および **def_retention: WORM** 機能の有効化

worm_capable マウントオプションにより、ファイルシステムでは WORM ファイルがサポートされます。*def_retention* マウントオプションは、*def_retention=MyNdOhPm* の形式を使用して、デフォルトの保存期間を設定します。

この形式では、*M*、*N*、*O*、および *P* は負でない整数であり、*y*、*d*、*h*、および *m* は、それぞれ年、日、時、分を表します。これらの単位を任意に組み合わせることができます。たとえば *1y5d4h3m* は 1 年、5 日、4 時間、3 分、*30d8h* は 30 日と 8 時間、*300m* は 300 分をそれぞれ表します。この形式は、保存時間を分単位で指定していた旧バージョンのソフトウェアと下位互換性があります。

詳細は、『Oracle Hierarchical Storage Manager and StorageTek QFS インストールおよび構成ガイド』(Oracle HSM お客様向けドキュメントライブラリ、docs.oracle.com/en/storage) を参照してください。

付録D アーカイブのための構成ディレクティブ

この付録では、Oracle Hierarchical Storage Manager ファイルシステムを構成するディレクティブと、関連するソフトウェア操作の一覧を表示します。それぞれのディレクティブは、1つ以上のコンマ区切りフィールドで構成された1つのテキスト行です。関連するディレクティブは、一緒に Oracle HSM コマンド (*.cmd*) ファイルに格納されます。

この付録の残りには、3つの主要な種類のディレクティブの概要が記載されています。

- [アーカイブディレクティブ](#)
- [ステージングディレクティブ](#)
- [プレビュー要求ディレクティブ](#)

追加情報については、Oracle HSM のマニュアルページを参照してください。

Oracle HSM コマンドファイルは、ここで説明されているようにコマンド行から構成することも、Oracle HSM Manager software を使用して構成することもできます。Oracle HSM Manager の詳細は、オンラインヘルプを参照してください。

アーカイブディレクティブ

このセクションには、*archiver.cmd* ファイルを構成するアーカイブディレクティブの使用法に関する情報が記載されています。アーカイブディレクティブは、ファイルのコピー、使用されるメディア、およびアーカイブソフトウェアの全体的な動作を制御するアーカイブセットを定義します。

アーカイブディレクティブには4つの基本タイプがあります。

- [グローバルアーカイブディレクティブ](#)
- [ファイルシステムディレクティブ](#)
- [コピーパラメータ](#)

- **ボリュームシリアル番号 (VSN) 関連付けディレクティブ**

グローバルディレクティブとファイルシステムディレクティブの両方が、ファイルのアーカイブ方法を制御します。ただし、アーカイバは、グローバルディレクティブを評価する前にファイルシステム固有のディレクティブを評価します。そのため、ファイルシステムディレクティブは、競合が存在する場合にグローバルディレクティブをオーバーライドします。同様に、ファイルシステムディレクティブ内では、最初に表示されるディレクティブによって後続の競合するディレクティブがオーバーライドされます。

グローバルアーカイブディレクティブ

グローバルディレクティブはアーカイバ全体の動作を制御し、構成済みのすべてのファイルシステムに合わせて動作を最適化することを可能にします。グローバルディレクティブは、単一のキーワードまたは等号 (=) のあとに追加データフィールドを伴うキーワードで構成されます。グローバルディレクティブは、`archiver.cmd` ファイルを開始し、最初のファイルシステムディレクティブで終了します。

archivemeta: メタデータをアーカイブするかどうかの制御

`archivemeta` 指示は、ファイルシステムメタデータをアーカイブするかどうかを制御します。ファイルシステム内でファイルの移動やディレクトリ構造の変更が頻繁に行われる場合は、ファイルシステムメタデータをアーカイブします。ただし、ディレクトリ構造が適度に安定している場合は、メタデータのアーカイブを無効にして、リムーバブルメディアドライブが行うアクションを減らすことができます。デフォルトでは、メタデータはアーカイブされません。

このディレクティブの形式は、次のとおりです。

```
archivemeta=state
```

`state` には、`on` または `off` を指定します。デフォルトは `off` です。

メタデータのアーカイブ処理は、Version 1 と Version 2 のどちらのスーパーブロックを使用するかによって、次のように異なります。

- **Version 1** ファイルシステムの場合、アーカイバはディレクトリ、リムーバブルメディアファイル、セグメント索引 i ノード、シンボリックリンクをメタデータとしてアーカイブします。

- Version 2 ファイルシステムの場合、アーカイバはディレクトリおよびセグメントインデックス i ノードをメタデータとしてアーカイブします。リムーバブルメディアファイルおよびシンボリックリンクは、データブロックではなく i ノードに格納されます。これらはアーカイブされません。シンボリックリンクは、データとしてアーカイブされます。

archmax: アーカイブファイルサイズの制御

archmax ディレクティブは、アーカイブ (.tar) ファイルの最大サイズを指定します。*target-size* 値に達すると、アーカイブファイルにそれ以上のユーザーファイルが追加されることはありません。複数のサイズの大きいユーザーファイルが、1 つのアーカイブファイルに書き込まれます。

デフォルト値を変更するには、次のディレクティブを使用します。

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

ここで *media* は、[付録A「装置タイプの用語集」](#) および *mcf* のマニュアルページで定義されているメディアタイプの 1 つ、*target-size* は、アーカイブファイルの最大サイズです。この値はメディアによって異なります。デフォルトでは、光ディスクに書き込まれるアーカイブファイルは最大 5M バイトです。テープの場合の最大アーカイブファイルのデフォルトサイズは、512M バイトです。

アーカイブファイルのサイズとして、大きいサイズを設定した場合も、小さいサイズを設定した場合も、それぞれに利点と欠点があります。たとえば、テープにアーカイブする場合、*archmax* を大きいサイズに設定すると、テープドライブの停止と開始の頻度が下がります。しかし、大きいアーカイブファイルを書き込むと、テープの終わりが早すぎて、大量のテープが無駄になる可能性があります。ベストプラクティスとしては、メディア容量の 5% を超える値に *archmax* 指示を設定しないでください。

archmax 指示は、個々のアーカイブセットに対して設定することもできます。

bufsize: アーカイババッファサイズの設定

デフォルトでは、アーカイブ対象ファイルは、メモリーバッファを使用してアーカイブメディアにコピーされます。*bufsize* ディレクティブを使用すると、デフォルト値以外のバッファサイズを指定したり、オプションでバッファをロックし

たりできます。状況によっては、これらのアクションにより、パフォーマンスを向上させることができます。さまざまな *number-blocks* 値を試すことができます。このディレクティブの形式は、次のとおりです。

```
bufsize=media number-blocks [lock]
```

ここでは:

- *media* は、[付録A「装置タイプの用語集」](#)と *mcf* のマニュアルページで定義されているメディアタイプの1つです
- *number-blocks* は、[2-1024] の範囲の数値です。デフォルトは4です。この値にメディアタイプの *dev_blksize* 値が乗算され、その結果がバッファサイズとして使用されます。*dev_blksize* 値は *defaults.conf* ファイルで指定されます。詳細は、*defaults.conf* のマニュアルページを参照してください。
- *lock* は、アーカイブコピーの作成時にアーカイバがロックバッファを使用できるかどうかを指示します。

lock が指定されている場合、*sam-arcopy* の動作中は、アーカイバがメモリー内のアーカイブバッファにファイルロックを設定します。この動作により、入出力リクエストごとにバッファをロックまたはロック解除することに伴うオーバーヘッドが回避され、その結果システムのCPU時間を短縮できます。

lock 引数は、大容量メモリーを備えた大型システムだけで指定する必要があります。十分なメモリーがないと、メモリー不足状態となります。*lock* 引数が有益なのは、アーカイブ対象のファイルに対して直接入出力が使用可能となっている場合のみです。デフォルトでは、*lock* は指定されておらず、アーカイブ用を含むすべての直接入出力バッファに、ファイルシステムによってロックが設定されています。

-bufsize および *-lock* アーカイブセットコピーパラメータを使用すると、アーカイブセットごとにバッファサイズとロックを指定できます。詳細は、[「アーカイブのコピーディレクティブ」](#)を参照してください。

drives: アーカイブに使用するドライブ数の制御

デフォルトの場合、アーカイバはアーカイブ用自動ライブラリにあるすべてのドライブを使用します。使用するドライブ数を制限するには、*drives* ディレクティブを使用します。このディレクティブの形式は、次のとおりです。

```
drives=media-library count
```

ここで *media-library* は、*mcf* ファイルで定義された自動ライブラリのファミリーセット名、*count* は、アーカイブで使用できるドライブの数です。

この目的でアーカイブセットコピーパラメータ *-drivemax*、*-drivemin*、および *-drives* を使用することもできます。詳細は、「[アーカイブのコピーディレクティブ](#)」を参照してください。

examine: アーカイブスキャンの制御

examine ディレクティブは、アーカイバがアーカイブの準備ができていないファイルを特定するために使用する *method* を設定します。

```
examine=method
```

method は次のいずれかのディレクティブです。

- デフォルトである *noscan* は継続アーカイブを指定します。最初のスキャンのあと、内容が変更されてアーカイブが必要なときにのみ、ディレクトリがスキャンされます。ディレクトリおよび *i* ノード情報はスキャンされません。このアーカイブ方法は、特にファイル数が 1,000,000 を超えるファイルシステムで、スキャンアーカイブよりも高いパフォーマンスが得られます。
- *scan* は、スキャンアーカイブを指定します。ファイルシステムディレクトリがはじめてスキャンされた後で、常に *i* ノードがスキャンされます。
- *scandirs* はスキャンアーカイブを指定します。ディレクトリは常にスキャンされます。*i* ノード情報はスキャンされません。

アーカイバは、*no_archive* 属性が設定されたディレクトリをスキャンしません。そのため、変更されていないファイルが含まれるディレクトリに対してこの属性を設定することで、スキャン時間を短縮できます。

- *scaninodes* は、スキャンアーカイブを指定します。*i* ノードは常にスキャンされます。ディレクトリ情報はスキャンされません。

interval: アーカイブ間隔の指定

アーカイバは、マウントされているすべてのアーカイブ対応ファイルシステムのステータスを定期的にチェックします。タイミングは、各ファイルシステムでのス

キャン操作間の時間であるアーカイブ間隔によって制御されます。アーカイブ間隔を変更するには、*interval* ディレクティブを使用します。

継続アーカイブが設定されておらず、*startage*、*startsize*、または *startcount* のどのパラメータも指定されていない場合にのみ、*interval* ディレクティブは完全スキャンを開始します。継続アーカイブが設定されている (*examine=noscan*) 場合、*interval* ディレクティブはデフォルトの *startage* 値として機能します。このディレクティブの形式は、次のとおりです。

```
interval=time
```

time には、ファイルシステムのスキャンを行う時間間隔を指定します。デフォルトでは、*time* は秒単位と見なされ、値は **600** (10 分) です。別の時間単位 (分や時など) も指定できます。

アーカイバは、*samu* ユーティリティーの *arrun* コマンドを受信すると、すべてのファイルシステムのスキャンをすぐに開始します。*archiver.cmd* ファイルで *examine=scan* ディレクティブも指定されている場合は、*arrun* または *arscan* が実行されたあとで、スキャンが実行されます。

ファイルシステムに *hwm_archive* マウントオプションが設定されている場合は、アーカイブ間隔を自動的に短縮できます。ファイルシステムの利用率が高位境界値を超えると、アーカイバはスキャンを開始します。*high=percent* マウントオプションは、高位境界値をファイルシステムに設定します。

アーカイブ間隔の指定方法の詳細は、*archiver.cmd* および *mount_samfs* のマニュアルページを参照してください。

logfile: アーカイバログファイルの指定

アーカイバは、アーカイブ、再アーカイブ、またはアーカイブ解除された各ファイルに関する情報を含むログファイルを出力できます。ログファイルは、アーカイブアクションを連続的に記録したものです。デフォルトでは、アーカイバログファイルは有効になりません。ログファイルを指定するには、*logfile* ディレクティブを使用します。このディレクティブの形式は、次のとおりです。

```
logfile=pathname
```

pathname には、ログファイルの絶対パスとファイル名を指定します。*logfile* 指示は、個々のファイルシステムに対して設定することもできます。

アーカイバログファイルは、破損したり失われたりしたファイルシステムを回復するために不可欠であり、モニタリングと分析に役立つことがあります。そのため、アーカイバログを有効にして、バックアップしてください。詳細は、『Oracle Hierarchical Storage Manager and StorageTek QFS インストールおよび構成ガイド』を参照してください。

notify: イベント通知スクリプトの名前変更

notify 指示は、アーカイバのイベント通知スクリプトファイルの名前を設定します。このディレクティブの形式は、次のとおりです。

```
notify=filename
```

filename に、アーカイバのイベント通知スクリプトを含むファイルの名前、またはフルパスを指定します。デフォルトのファイル名は */etc/opt/SUNWsamfs/scripts/archiver.sh* です。

アーカイバはこのスクリプトを実行して、さまざまなイベントをサイト固有の方法で処理します。このスクリプトは、第1引数のキーワード *emerg*、*alert*、*crit*、*err*、*warning*、*notice*、*info*、*debug* のいずれかで呼び出されます。

そのほかの引数については、デフォルトのスクリプトで説明されています。詳細は、*archiver.sh* のマニュアルページを参照してください。

ovflmin: ボリュームオーバーフローの制御

ボリュームオーバーフローが有効になっていると、アーカイバは複数のボリュームにまたがるアーカイブファイルを作成できます。ファイルサイズが指定された最小サイズを超えると、アーカイバはこのファイルの残りの部分を同じタイプの別のボリュームに書き込みます。各ボリュームに書き込まれたファイル部分のことを、「セクション」と呼びます。*sls* コマンドは、アーカイブコピーの一覧を表示して、各ボリュームにあるファイルの各セクションを示します。

アーカイバは、*ovflmin* 指示によってボリュームオーバーフローを制御します。デフォルトでは、ボリュームオーバーフローは使用不可となっています。ボリュームオーバーフローを有効にするには、*archiver.cmd* ファイルで *ovflmin* ディレクティブを使用します。このディレクティブの形式は、次のとおりです。

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

ここで *media* は付録A「装置タイプの用語集」および *mcf* のマニュアルページで定義されているメディアタイプの1つで、*minimum-file-size* はボリュームオーバーフローをトリガーする最小ファイルのサイズです。*ovflmin* 指示は、個々のアーカイブセットに対して設定することもできます。

ボリュームオーバーフローは、及ぼす影響を検討したうえで慎重に使用してください。複数のボリュームをまたぐファイルの場合は、障害回復とリサイクルが非常に難しくなります。ボリュームオーバーフローファイルでは、チェックサムは生成されません。チェックサムの使用方法の詳細は、*ssum* のマニュアルページを参照してください。

scanlist_squash: スキャンリストの連結の制御

scanlist_squash パラメータは、スキャンリストの連結を制御します。デフォルトの設定は無効 (*off*) です。このパラメータはグローバルに使用することも、特定のファイルシステム用に使用することもできます。

on にすると、アーカイバが共通の親ディレクトリから下に再帰的にスキャンできるように、このディレクティブはディレクトリツリー内のサブディレクトリのスキャンリストを統合します。多数のファイルとサブディレクトリがファイルシステム内で変更されている場合、スキャンリストの統合によって、アーカイブパフォーマンスが大幅に低下することがあります。

setarchdone: archdone フラグ設定の制御

setarchdone グローバルディレクティブは、アーカイブされることがないファイルで *archdone* フラグが設定されるかどうかを制御します。このディレクティブの形式は、次のとおりです。

```
setarchdone=state
```

state は *on* または *off* のいずれかです。*examine* ディレクティブが *scandirs* または *noscan* に設定されている場合、デフォルトは *off* です。

archdone フラグは、フラグ付きのファイルを無視するようにアーカイブ処理に指示します。通常、ファイルの指定されたコピーがすべて作成されたら、ファイルがあとで変更されるまで、またはファイルがあとで変更されないかぎり、後続のアーカ

イブ操作でファイルがスキップされるように、アーカイブ処理は *archdone* フラグを設定します。

ただし、*setarchdone* が *on* に設定されている場合、アーカイブ処理は、アーカイブ基準を満たさないためにアーカイブされることがない、アーカイブされていないファイルを特定してフラグを付けます。これによって将来のアーカイブのオーバーヘッドを削減できますが、ファイルの評価によってオーバーヘッドが即時に増加して、パフォーマンスに悪影響を与える可能性があります。

wait: アーカイバ起動の遅延

wait ディレクティブを使用すると、アーカイバは、*samcmd* コマンド、*samu* インタフェース、または Oracle HSM Manager からの開始シグナルを待機します。このディレクティブの形式は、次のとおりです。

```
wait
```

デフォルトでは、*sam-fsd* 初期化コマンドの実行時にアーカイバは自動的に開始します。

wait 指示は、個々のファイルシステムに対して設定することもできます。

ファイルシステムディレクティブ

ファイルシステムディレクティブは、特定のファイルシステムのアーカイブ動作を定義します。

- **fs**: ファイルシステムの指定
- **copy-number [archive-age]**: ファイルシステムメタデータの複数コピーの指定
- ファイルシステムディレクティブとしての **interval**、**logfile**、**scanlist**

fs: ファイルシステムの指定

各 *fs=file-system-name* ディレクティブでは、指定されたファイルシステム *file-system-name* にのみ適用される一連のアーカイブディレクティブが導入されます。このディレクティブの形式は、次のとおりです。

```
fs=file-system-name
```

file-system-name は、*mcf* ファイルで定義されているファイルシステム名です。

fs= ディレクティブのあとに配置された汎用ディレクティブとアーカイブセット関連付けディレクティブは、指定したファイルシステムにのみ適用されます。

copy-number [archive-age]: ファイルシステムメタデータの複数コピーの指定

ファイルシステムメタデータには、ファイルシステムにおけるパス名が含まれます。メタデータの複数のコピーが必要な場合は、*archiver.cmd* ファイルの *fs=* ディレクティブの直後にコピー定義を配置します。

```
copy-number [archive-age]
```

時間は、整数と時間の単位の 1 つ以上の組み合わせで表されます。単位には、*s* (秒)、*m* (分)、*h* (時)、*d* (日)、*w* (週)、および *y* (年) が含まれます。ディレクトリが頻繁に変更される場合に、複数のメタデータコピーを指定すると、これによってファイルシステムがメタデータテープボリュームをマウントする頻度が高くなる可能性があります。そのため、デフォルトでは、Oracle HSM は、メタデータの単一のコピーのみを作成します。

この例では、*fs=samma1* ファイルシステムについてメタデータのコピー 1 が 4 時間 (4h) 後に、コピー 2 が 12 時間 (12h) 後に作成されます。

```
# General Directives
archivemeta = off
examine = noscan
# Archive Set Assignments
fs = samma1
1 4h
2 12h
```

ファイルシステムディレクティブとしての interval、logfile、scanlist

いくつかの指示は、すべてのファイルシステムを対象とするグローバル指示として指定することも、1 つのファイルシステムだけを対象とする指示として指定することもできます。これらの指示については、次のセクションで説明されています。

- **interval:** アーカイブ間隔の指定
- **logfile:** アーカイバログファイルの指定

- **scanlist_squash**: スキャンリストの連結の制御
- **wait**: アーカイバ起動の遅延

archive-set-name: アーカイブセット割り当てディレクティブ

アーカイブセット割り当てディレクティブは、同時にアーカイブされるファイルを指定します。次で説明する幅広い選択基準を使用すると、非常に細かくファイルを指定できます。ただし、やむをえない場合を除いて、使用は避けてください。通常は、可能なかぎり包括的なアーカイブセットを最小数だけ構成するようにしてください。アーカイブセットでは、アーカイブメディアのセットが排他的に使用されません。そのため、過度に制限された割り当て基準によって個別に定義されたアーカイブセットが多数あると、メディアの利用率が低くなり、システムのオーバーヘッドが高くなり、パフォーマンスが低下します。極端なケースでは、ライブラリ内に十分な容量が残っているにもかかわらず、使用可能なメディアの不足が原因でジョブが失敗する可能性があります。

アーカイブセット割り当て指示の形式は、次のとおりです。

```
archive-set-name path [-access interval [-nftv]] [-after date-time] [-minsize size] [-maxsize size] [-user username] [-group groupname] [-name regex]
```

ここでは:

- **archive-set-name** は、管理者によって定義されたアーカイブセットの名前です。

名前には、最大 29 文字の大文字と小文字 [A-Za-z]、数字 [0-9]、および下線 () を任意の組み合わせで含めることができますが、先頭文字は文字である必要があります。空白文字を含むその他の文字は含めることができず、独自のアーカイブセットには Oracle HSM の特殊なアーカイブセット *no_archive* と *all* の名前は使用できません。

- **path** は、ファイルシステム内でアーカイブ処理を開始するサブディレクトリのマウントポイントを基準とする相対パスを指定します。開始ディレクトリとそのサブディレクトリ内のすべてのファイルがアーカイブされます。ファイルシステム内のすべてのファイルを含めるには、ドット (.) 文字を使用します。パスの先頭にスラッシュ (/) を使用することはできません。
- **-access** は、**interval** によって指定された期間アクセスされていないファイルを再アーカイブします。**interval** は整数

で、*s* (秒)、*m* (分)、*h* (時)、*d* (日)、*w* (週)、および *y* (年) のいずれかの単位があとに付きます。

このパラメータを使用すると、あまり使用されないファイルの再アーカイブを、高コストのメディアから低コストのメディアにスケジュールできます。ソフトウェアは、ファイルのアクセス時間と変更時間の妥当性検査を行い、これらの時間がファイルの作成時間以降であり、さらにファイルの検証時間以前であることを確認します。*-nftv* (ファイル時間の検証なし) パラメータは、この検証を無効にします。

- *-after* は、*date-time* のあとで作成または変更されたファイルのみをアーカイブします。*date-time* は、*YYYY-MM-DD [hh:mm:ss] [Z]* 形式の式であり、*YYYY*、*MM*、*DD*、*hh*、*mm*、および *ss* は、それぞれ年、月、日、時、分、および秒を表す整数です。オプションの *Z* パラメータは、タイムゾーンを協定世界時 (UTC) に設定します。デフォルトは *00:00:00* と現地時間です。
- *-minsize* および *-maxsize* は、指定された *size* より大きい小さいファイルのみをアーカイブします。*size* は整数で、*b* (バイト)、*k* (K バイト)、*M* (M バイト)、*G* (G バイト)、*T* (T バイト)、*P* (P バイト)、*E* (E バイト) のいずれかの単位があとに付きます。
- *-user username* および *-group groupname* は、指定されたユーザーまたはグループ (あるいはその両方) に属するファイルのみをアーカイブします。
- *-name* は、正規表現 *regex* で定義されたパターンに一致するパスおよびファイル名を含むすべてのファイルをアーカイブします。

アーカイブのコピーディレクティブ

アーカイバは、デフォルトでアーカイブセット内のファイルのアーカイブ経過時間が4分であるときに、それらのファイルに対してアーカイブのコピーを1つ書き込みます。デフォルトの動作を変更するには、アーカイブのコピーディレクティブを使用します。アーカイブのコピーディレクティブは、関連するアーカイブセット割り当てディレクティブの直あとに配置する必要があります。

アーカイブのコピーディレクティブは、1、2、3、4 のいずれかの *copy-number* 値から始まります。この数字のあとに、そのコピーのアーカイブ特性を指定する1つまたは複数の引数が続きます。アーカイブのコピーディレクティブの形式は、次のとおりです。

```
copy-number [archive-age] [-release [attribute] [-norelease][-stage[attribute] [unarchive-age]
```

ここでは:

- オプションの *archive-age* パラメータは、新規または変更済みのファイルがアーカイブ対象になる前に、ディスクキャッシュ内に存在しなければならない時間です。整数と時間単位の1つ以上の組み合わせで *archive-age* を指定します。単位には *s* (秒)、*m* (分)、*h* (時間)、*d* (日)、*w* (週)、および *y* (年) が含まれます。デフォルトは *4m* (4分) です。
- オプションの *-release* パラメータは、アーカイブコピーが作成された直後に、ファイルで使用されているディスク領域が解放されるように Oracle HSM リリーサソフトウェアをクリアします。オプションの解放属性は、*-a*、*-n*、または *-d* です。*-a* (結合ステージング) 属性は、アーカイブセットから解放されたファイルのいずれかにアクセスしたときに、ソフトウェアがこれらのファイルをすべてステージングするように要求します。*-n* 属性は、ソフトウェアがアーカイブメディアからファイルを直接読み取り、ステージングしないように要求します。*-d* 属性は、デフォルトのステージング動作をリセットします。
- オプションの *-norelease* パラメータは、*-norelease* マークが付けられたコピーがすべて作成されるまで、ファイルで使用されているディスク領域を解放するように Oracle HSM リリーサソフトウェアをクリアしません。
- *-release -norelease* は同時に使用され、*-release -norelease* フラグが付けられたすべてのコピーが作成された直後に、Oracle HSM ソフトウェアがファイルで使用しているディスク領域を解放するように要求します。Oracle HSM は、リリーサプロセスの実行を待機しません。
- オプションの *-stage* パラメータ。オプションの解放属性は *-a*、*-c copy-number*、*-f*、*-I*、*-i input_file*、*-w*、*-n*、*-p*、*-V*、*-x*、*-r*、*-d* です。ここでは:

-a は、アーカイブセットのファイルのいずれかにアクセスしたときに、これらのファイルをすべてステージングするように要求します。

-c copy-number は、ソフトウェアが指定したコピー番号からステージングするように要求します。

-n は、ソフトウェアがアーカイブメディアからファイルを直接読み取り、ステージングしないように要求します。

-w は、各ファイルが正常にステージングされるまでソフトウェアが待機してから、続行するように要求します (*-d* または *-n* では無効です)。

`-d` は、デフォルトのステージング動作をリセットします。

- `unarchive-age` パラメータは、ファイルのアーカイブコピー再利用のため、メディア上の空き領域にアーカイブ解除する前に、アーカイブ内に存在しなければならない時間を指定します。時間は、整数と時間単位を1つ以上組み合わせて表現されます。単位には `s` (秒)、`m` (分)、`h` (時間)、`d` (日)、`w` (週)、および `y` (年) が含まれます。

下記の例には、アーカイブセット `allsamma1` の2つのコピーディレクティブが含まれています。1つ目のディレクティブは、アーカイブ経過時間が5分 (`5m`) に達するまで、コピー 1 を解放しません。2つ目のディレクティブは、アーカイブ経過時間が1時間 (`1h`) に達するまでコピー 2 を解放せず、アーカイブ解除期間の7年6か月 (`7y6m`) に達するとコピー 2 をアーカイブ解除します。

```
# Archive Set Assignments
fs = samma1
logfile = /var/adm/samma1.archive.log
allsamma1 .
    1 -norelease 5m
    2 -norelease 1h 7y6m
```

コピーパラメータ

コピーパラメータは、アーカイブセットで指定されたコピーの作成方法を定義します。`archiver.cmd` ファイルのアーカイブセットのコピーパラメータセクションは、`params` ディレクティブで始まり `endparams` ディレクティブで終わります。

```
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
```

各コピーパラメータの形式は、次のとおりです。

```
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]
```

ここでは:

- `archive-set-name` は、ファイルシステムディレクティブでアーカイブセット割り当てディレクティブによって定義されているアーカイブセットの名前か、指定されたコピーパラメータを定義済みのすべてのアーカイブセットに適用する特殊なディレクティブ `allsets` です。個々のアーカイブセットのパラメータを指定する前に、最初に `allsets` のパラメータを設定します。そうしないと、個々のアーカイブセットのパラメータによって `allsets` の指定がオーバーライドされ、その目的が果たされなくなります。
- `.copy-number` は、指定されたコピーパラメータの適用を、`copy-number` で指定されたアーカイブコピーに制限します。ここで `copy-number` は、[1-4] の範囲の整数であり、オプションの `R` は、パラメータの適用を再アーカイブのコピーに制限します。
- `-startage time` は、最初のファイルがアーカイブ要求に追加されたときから、アーカイブ処理が実際に開始されたときまでの間の間隔を指定します。`time` に整数と時間単位の 1 つ以上の組み合わせを指定します。単位には `s` (秒)、`m` (分)、`h` (時)、`d` (日)、`w` (週)、および `y` (年) が含まれます。デフォルトは `2h` (2 時間) です。
- `-startcount count` は、アーカイブ要求内のファイルの最小数を指定します。アーカイブ処理を待機しているファイルの数がこのしきい値に達すると、アーカイブ処理が開始されます。デフォルトでは、`count` は設定されません。
- `-startsize size` は、アーカイブ要求の最小サイズをバイト単位で指定します。アーカイブ処理を待機しているファイルの合計サイズがこのしきい値に達すると、アーカイブ処理が開始されます。デフォルトでは、`size` は設定されません。
- `-archmax` は、アーカイブファイルのサイズを `maximum-size` までに制限します。ここで `maximum-size` は、メディアによって異なります。磁気テープの場合、デフォルトの最大アーカイブファイルサイズは 512M バイトです。光ディスクに書き込まれるアーカイブファイルは、最大 5M バイトです。

同じ名前のグローバルアーカイブディレクティブについては、「[archmax: アーカイブファイルサイズの制御](#)」を参照してください。

- `-bufsize=media-type number-blocks` は、アーカイブメディアに書き出されるときにアーカイブファイルを保持するバッファのサイズを `number-blocks*dev_blksize` に設定します。ここで `number-blocks` は、バッファされるテープブロックの数を表す [2-32] の範囲の整数、`dev_blksize` は、`defaults.conf` ファイルでメディアタイプに指定されたブロックサイズです。デフォルトは 4 です。

- `-drivemax` は、1 台のドライブを使用してアーカイブされるデータの量を `maximum-size` メガバイト以下に制限します。ここで `maximum-size` は整数です。デフォルトでは、`maximum-size` は指定されていません。

`-drives` パラメータを使用して複数のドライブが指定されている場合、1 台の任意のドライブに書き込まれるデータの量を制限すると、ワークロードを平準化して、ドライブ全体の利用率を改善できます。

- `-drivemin minimum-size` は、1 台のドライブを使用してアーカイブされるデータの量を `minimum-size` メガバイト以上に制限します。ここで `minimum-size` は整数です。デフォルトは、`-archmax` 値 (指定されている場合)、または `defaults.conf` ファイルに一覧表示されたメディアタイプの値です。

ドライブに書き込まれるデータの量に下限を設定すると、ドライブの利用率と効率性を改善できます。`minimum-size` は、転送時間がメディアのロード、位置設定、およびアンロードに必要な時間を大幅に上回るよう十分な長さに設定してください。`-drivemin` を指定すると、データ転送が十分に長い場合にのみ複数のドライブが使用されます。

- `-drives number` は、アーカイブ処理で使用されるドライブ数を `number` までに制限します。ここで `number` は整数です。デフォルトは **1** です。

アーカイブセットに含まれるファイルのサイズが大きい、またはファイルの数が多の場合、ドライブの最大数を大きく設定すると、パフォーマンスを改善できます。使用可能なドライブの動作速度が異なる場合は、複数のドライブを指定することで、このようなばらつきを平準化して、アーカイブ処理の効率性を高めることもできます。

- `-fillvsns` は、小さいアーカイブファイルを使用して、より完全にアーカイブメディアボリュームを満杯にするようにアーカイブ処理プロセスに強制します。

デフォルトでは、アーカイバはアーカイブコピー内のすべてのファイルを保持するのに十分な領域のあるボリュームを選択します。このため、アーカイブファイルが大きくなり、多数のカートリッジにある残りの容量に収まらなくなります。その結果、メディア全体が十分に利用されなくなります。この問題は `-fillvsns` パラメータで対処できますが、その代償としてメディアのマウント、位置設定操作、およびアンマウントが多くなり、これらのすべてによってアーカイブ処理およびステージング処理のパフォーマンスが低下します。

- `-lock` は、直接入出力を使用してアーカイブコピーを作成するときに、ロックバッファを使用するように指示します。ロックバッファを使用することに

よってバッファのページングが回避され、直接入出力のパフォーマンスが改善されます。

使用可能なメモリーが制限されているシステム上で `-lock` パラメータを指定すると、メモリー不足の状態になる可能性があります。デフォルトでは、ロックバッファは指定されていないため、ファイルシステムがアーカイブバッファの制御を保持します。

- `-offline_copy_method` は、すでにファイルがディスクキャッシュから解放されているときに、アーカイブコピーを作成する方法を指定します。 `method` には `direct`、`stageahead`、`stageall`、または `none` を指定できます。

単一のアーカイブコピーが作成された直後にファイルが解放される可能性があるため、残りのコピーはオフラインのコピーから作成する必要があります。 `-offline_copy` メソッドを指定すると、使用可能にできるドライブの数とディスクキャッシュの空き領域に合わせて、コピープロセスを調整できます。

`direct` は、2台のドライブを使用して、オフラインボリュームからアーカイブボリュームに直接ファイルをコピーします。適切なバッファ領域を確保するには、このメソッドを使用するときに、`stage_n_window` マウントオプションで設定された値を大きくします。

`stageahead` は、アーカイブファイルをコピー先に書き込む間に、次のアーカイブファイルをステージングします。

`stageall` は、1台のドライブを使用して、アーカイブ処理の前にすべてのファイルをディスクキャッシュにステージングします。このメソッドを使用する場合は、ディスクキャッシュがファイルを保持できるだけの大きさであることを確認してください。

`none` (デフォルト) は、アーカイブボリュームにコピーする前に、必要に応じてファイルをディスクキャッシュにステージングします。

- `-sort` は、アーカイブ処理の前に、`criterion` でファイルをソートします。ここで `criterion` は、`age`、`priority`、`size`、または `none` です。

`age` は、変更時間 (最古から最新まで) でのソートを指定します。

`path` (デフォルト) は、フルパス名でのソートを指定します。これにより、同じディレクトリに存在するファイルがアーカイブメディア上にまとめられます。

priority は、アーカイブ処理のソートを優先順位高から低の順に指定します。

size は、ファイルをファイルサイズ最小から最大の順でソートします。

none は、ソートを指定せず、ファイルシステムで発生した順序でファイルをアーカイブします。

- *-rsort criterion* は、*-sort* と同様に、ただし逆順でファイルを *criterion* でソートします。
- *-recycle_dataquantity size* は、リサイクラが再アーカイブ対象としてスケジュールするデータの量を *size* バイトに制限します。ここで *size* は整数です。

リサイクラは、有効なアーカイブファイルのアーカイブボリュームを空にする必要があるときに、再アーカイブ処理をスケジュールします。リサイクル対象として選択したボリュームの実際の数、*-recycle_vsncount* パラメータによって異なる可能性もあることに注意してください。デフォルトは **1073741824** (1G バイト) です。

- *-recycle_hwm percent* は、リムーバブルメディアのリサイクル処理が開始されるメディアの最大利用率 (高位境界値または *hwm*) を設定します。ディスクメディアの場合、このパラメータは無視されます (下記の *-recycle_minobs* を参照)。デフォルトは **95** です。
- *-recycle_ignore* では、リサイクルプロセスの通常実行は許可されますが、アーカイブセット内では実際にメディアのリサイクルは実行できません。テスト用に使用されます。
- *-recycle_mailaddr mail-address* は、リサイクラの情報メッセージを *mail-address* に転送します。デフォルトでは、メールは送信されません。
- *-recycle_mingain* は、リサイクル対象のボリュームの選択を、少なくとも指定した *percentage* の空き領域が増加する数に制限します。デフォルトは **50** です。
- *-recycle_vsncount* は、リサイクラが再アーカイブ処理をスケジュールするボリュームの数を *count* に制限します。リサイクル対象として選択したボリュームの実際の数、*-recycle_dataquantity* パラメータによって異なる可能性もあります。ディスクメディアの場合、このパラメータは無視されます。デフォルトは **1** です。
- *-recycle_minobs* は、有効なファイルの再アーカイブ処理および元の *tar* ファイルの最終的な削除をトリガーする、ディスク常駐のアーカイブファイル内にある古いファイルの *percentage* を設定します。リムーバブルメディアの場合、この

パラメータは無視されます (上記の `-recycle_hwm` を参照)。デフォルトは `50` です。

- `-unarchage` は、アーカイブ解除期間を計算するための参照時間を `time_ref` に設定します。ここで `time_ref` は、ファイルアクセス時間を表す `access` (デフォルト) または変更時間を表す `modify` です。
- `-tapenonstop` は、リムーバブルメディアファイルを閉じることなく、アーカイブファイルの最後に単一のテープマークとファイルの終わり (EOF) ラベルを書き込みます。これによって、複数のアーカイブファイルの転送が高速化されますが、アーカイブセット全体がテープに書き込まれるまでテープカートリッジをアンロードできなくなります。デフォルトでは、Oracle HSM ソフトウェアは、アーカイブファイルの最後にあるファイルの終わりラベルのあとに 2 つの追加のテープマークを書き込むことで、テープファイルを閉じます。
- `-reserve keyword` は、指定されたアーカイブセットを排他的に使用するためにリムーバブルメディアボリュームを予約します。アーカイブセットのファイルを保持するために最初にボリュームを使用すると、ソフトウェアは、指定された 1 つ以上のキーワード `fs`、`set` と、`dir` (ディレクトリ)、`user`、または `group` のいずれかまたはこれらの両方に基づいて、一意の予約名をボリュームに割り当てます。

`fs` は、ファイルシステム名を予約名に含めます。 `arset.1 -reserve fs`。

`set` は、アーカイブセット割り当てディレクティブからのアーカイブセット名を予約名に含めます (`all -reserve set`)。

`dir` は、アーカイブセット割り当てディレクティブで指定されたディレクトリパスの最初の 31 文字を予約名に含めます。

`user` は、アーカイブファイルに関連付けられたユーザー名を含めます。 `arset.1 -reserve user`。

`group` は、アーカイブファイルに関連付けられたグループ名を含めます。 `arset .1 -reserve group`。

状況によっては、セット別にボリュームを予約すると便利ことがあります。ただし、ソフトウェアでメディアを選択する場合よりも本質的に非効率です。ボリュームを予約する場合は、システムは、より頻繁にカートリッジのマウント、アンマウント、および配置を行う必要があり、オーバーヘッドが増加してパ

パフォーマンスが低下します。予約スキームの制約が大きいと、使用可能なメディアが十分に利用されず、極端な例では使用可能なメディアがないためにアーカイブ処理が失敗することがあります。

- `-priority multiplier ranking` は、前述の `sort priority` パラメータとともに使用したときのファイルのアーカイブの優先順位を変更します。`ranking` は、 $[(-3.4000000000E+38) - 3.4000000000E+38] (-3.402823466 \times 10^{38} - 3.402823466 \times 10^{38})$ の範囲の実数で、`multiplier` は、次のリストから選択した相対的な `ranking` を変更するアーカイブ特性です。`age`、`archive_immediate`、`archive_overflow`、`archive_loaded`、`copies`、`copy1`、`copy2`、`copy3`、`copy4`、`offline`、`queuwait`、`re-archive`、`reqrelease`、`size`、`stage_loaded`、および `stage_overflow`。

優先順位の詳細は、`archiver` および `archiver.cmd` のマニュアルページを参照してください。

ボリュームシリアル番号 (VSN) プールディレクティブ

`archiver.cmd` ファイルの VSN プールセクションにより、ボリュームシリアル番号 (VSN) 関連付けディレクティブで 1 つの単位として指定できるアーカイブメディアボリュームの名前付きのコレクションが定義されます。

このセクションは `vsnpools` ディレクティブで始まり、`endvsnpools` ディレクティブまたは `archiver.cmd` ファイルの最後で終わります。VSN プール定義の構文は次のとおりです。

```
vsn-pool-name media-type volume-specification
```

ここでは:

- `vsn-pool-name` は、プールに割り当てる名前です。
- `media-type` は、[付録A「装置タイプの用語集」](#) および `mcf` のマニュアルページで一覧表示されている 2 文字の Oracle HSM メディアタイプ識別子です。
- `volume-specification` は、ボリュームシリアル番号に一致する 1 つ以上の正規表現のスペース区切りリストです。正規表現構文の詳細は、Solaris `regcmp` のマニュアルページを参照してください。

この例では、4 つの VSN プール (`users_pool`、`data_pool`、`proj_pool`、および `scratch_pool`) を定義します。スクラッチプールは、VSN 関連付け内の一部のボリュームを使い切ったとき、または別の VSN プールが空の状態になったときに使用

されるボリュームセットです。指定した3つのプールのいずれかがボリューム不足になった場合、アーカイバはスクラッチプール VSN を選択します。

```
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
```

ボリュームシリアル番号 (VSN) 関連付けディレクティブ

archiver.cmd ファイルの VSN 関連付けセクションでは、アーカイブセットにアーカイブメディアボリュームを割り当てます。このセクションは *vsns* ディレクティブで始まり、*endvsns* ディレクティブで終わります。

ボリューム割り当てディレクティブの形式は、次のとおりです。

```
archive-set-name.copy-number [media-type volume-specification] [-pool vsn-pool-name]
```

ここでは:

- *archive-set-name* は、アーカイブセット割り当てディレクティブが指定のボリュームに関連付けるアーカイブセットに割り当てた名前です。
- *copy-number* は、アーカイブセット割り当てディレクティブが指定のボリュームに関連付けるコピーに割り当てた番号です。これは、[1-4] の範囲の整数です。
- *media-type* は、[付録A「装置タイプの用語集」](#) および *mcf* のマニュアルページで一覧表示されている2文字の Oracle HSM メディアタイプ識別子です。
- *volume-specification* は、ボリュームシリアル番号に一致する1つ以上の正規表現のスペース区切りリストです。正規表現構文の詳細は、Solaris *regcmp* のマニュアルページを参照してください。
- *-pool vsn-pool-name* は、事前に指定された、単位として指定できるアーカイブメディアボリュームの名前付きコレクションです。ボリュームシリアル番号 (VSN) プールディレクティブを参照してください。

この例では、メディアを2行の VSN 指定に関連付けるさまざまな方法を示します。

```
vsns
archiveset.1 lt VSN001 VSN002 VSN003 VSN004 VSN005
archiveset.2 lt VSN0[6-9] VSN10
archiveset.3 -pool data_pool
```

endvsns

ステージングディレクティブ

書き込みは、ニアラインまたはオフラインの記憶装置からオンライン記憶装置に、ファイルデータをコピーして戻すことです。

ステージャーは、*samd* デーモンが実行されたときに起動します。ステージャーのデフォルトの動作は次のとおりです。

- ステージャーは、ライブラリ内のすべてのドライブを使用しようとする。
- 書き込みバッファサイズはメディアタイプ別に決定され、書き込みバッファはロックされない。
- ログファイルへの書き込みは行われぬ。
- 一度にアクティブであることが可能な書き込み要求は、最大 1000 個。

/etc/opt/SUNWsamfs/stager.cmd ファイルにディレクティブを挿入すると、ステージャーの動作をサイトに合わせてカスタマイズできます。

ファイルが *-n (never stage)* オプションでアーカイブされていなければ、アプリケーションでオフラインファイルが必要なときに、そのアーカイブコピーがディスクキャッシュにステージングされます。アプリケーションがファイルをすぐに利用できるようにするため、読み取り操作が書き込み操作のすぐあとを追跡するので、ファイル全体が書き込まれる前にアクセスを開始できます。

書き込みエラーとしては、メディアエラー、メディアを利用できない、自動ライブラリを利用できない、などがあります。ステージングエラーが返された場合、Oracle HSM ソフトウェアは次に使用可能なファイルのコピーを検索しようとして、そのようなコピーが存在し、アーカイブコピーのメディアを読み取るために使用できるデバイスがある場合)。

stager.cmd ファイル

stager.cmd ファイルには、デフォルト動作をオーバーライドするための指示を指定できます。ステージャーを構成して、ファイルをただちに書き込んだり、ファイルをまったく書き込まなかったり、部分的に書き込んだり、ほかの書き込みアクションを指定したりできます。たとえば、非書き込み属性を使用すると、ファイルをオンラインで書き込まずにアーカイブメディアから直接データにアクセスできる

ため、大きいファイルから小さいレコードにアクセスするアプリケーションに有益です。

このセクションでは、ステージャー指示について説明します。ステージャーディレクティブの詳細は、*stager.cmd* のマニュアルページを参照してください。Oracle HSM Manager software を使用している場合は、「ファイルシステムの概要」ページまたは「ファイルシステムの詳細」ページからステージングを制御できます。ファイルシステムをブラウズし、個々のファイルのステータスを表示できます。また、フィルタを使用して特定のファイルを表示し、書き込むファイルを選択することができます。書き込み元のコピーを選択することも、システムにコピーを選択させることもできます。

この例は、指定可能なディレクティブをすべて設定したあとの *stager.cmd* ファイルを示しています。

```
drives=dog 1
bufsize=od 8 lock
logfile=/var/adm/stage.log
maxactive=500
```

drives: ステージングに使用するドライブ数の指定

デフォルトでは、ステージャーはファイルの書き込みを行うときに利用可能なすべてのドライブを使用します。ステージャーによってすべてのドライブが使用中の状態のままになると、アーカイバのアクティビティに支障を来す恐れがあります。*drives* 指示は、ステージャーが利用できるドライブの数を指定します。このディレクティブの形式は、次のとおりです。

```
drives=library count
```

ここでは:

- *library* は、*mcf* ファイルに表示されるライブラリのファミリセット名です。
- *count* は、使用されるドライブの最大数です。デフォルトでは、このライブラリ用として *mcf* ファイルに構成されているドライブ数。

この例では、*dog* ファミリセットのライブラリにある 1 台のドライブだけをファイルのステージングに使用するよう指定します。

```
drives = dog 1
```

bufsize: ステージングバッファースizeの設定

デフォルトでは、書き込み対象ファイルは、アーカイブメディアからディスクキャッシュに復元される前に、メモリーバッファに読み取られます。*bufsize* ディレクティブを使用して、バッファースizeを指定したり、オプションでバッファをロックしたりできます。これらの操作により、パフォーマンスを向上させることができます。さまざまな *number-blocks* 値を試すことができます。このディレクティブの形式は、次のとおりです。

```
bufsize= media-type number-blocks [lock]
```

ここでは:

- *media-type* は、[付録A「装置タイプの用語集」](#) および *mcf* のマニュアルページで一覧表示されている 2 文字の Oracle HSM メディアタイプ識別子です。
- *number-blocks* は、[2-8192] の範囲の整数です。この値は *defaults.conf* ファイルに指定された *media-type_blksize* 値で乗算されます。*number-blocks* の値が高ければ高いほど、多くのメモリーが使用されます。デフォルトは 16 です。
- *lock* は、各ステージング操作の期間内にロックバッファを使用するように指示します。これにより、入出力要求ごとのステージングバッファのロックまたはロック解除に関連するオーバーヘッドが回避され、パフォーマンスが改善されます。使用可能メモリーが制限されたシステム上で *lock* パラメータを指定すると、メモリー不足の状態になる可能性があります。デフォルトでは、ロックバッファは指定されていないため、ファイルシステムがアーカイブバッファの制御を保持します。

lock 引数が有効であるのは、ステージング対象のファイルで直接入出力が有効になっている場合のみです。直接入出力の有効化についての詳細は、*setfa*、*sam_setfa*、および *mount_samfs* のマニュアルページを参照してください。

logfile: ステージングログファイルの指定

Oracle HSM ソフトウェアがファイルステージングイベント情報を収集し、それをログファイルに書き込むように要求できます。デフォルトでは、ログファイルへの書き込みは行われません。*logfile* ディレクティブは、ステージャーがログ情報を書き込むことができるログファイルを指定します。ステージャーは、書き込みを行なったファイルごとに 1 つまたは複数の行をログファイルに書き込みます。この 1

行には、ファイル名、書き込みを行なった日時、ボリュームシリアル番号 (VSN) などが含まれます。このディレクティブの形式は、次のとおりです。

```
logfile=filename [event-list]
```

ここで *filename* は、ログファイルのフルパス名、*event-list* は、ログが記録されるイベントタイプのスペース区切りリストです。

- *all* は、すべてのステージングイベントのログを記録します。
- *start* は、ファイルのステージングが開始されたときにログを記録します。
- *finish* (デフォルト) は、ファイルのステージングが終了したときにログを記録します。
- *cancel* (デフォルト) は、オペレータがステージングを取り消したときにログを記録します。
- *error* (デフォルト) は、ステージングエラーのログを記録します。

次のディレクティブは、`/var/adm/` ディレクトリにステージングログを作成します。

```
logfile=/var/adm/stage.log
```

ステージャーログエントリの形式は、次のとおりです。

```
status date time media-  
type volume position.offset inode filesize filename copy user group requestor equipment-  
number validation
```

ここでは:

- *status* は、開始を表す *S*、取り消しを表す *C*、エラーを表す *E*、完了を表す *F* です。
- *date* は、`yyyy/mm/dd` 形式の日付です。ここで *yyyy* は年を表す 4 桁の数字、*mm* は月を表す 2 桁の数字、*dd* は日を表す 2 桁の数字です。
- *time* は、`hh:mm:ss` 形式の時間です。ここで *hh*、*mm*、および *ss* は、それぞれ時間、分、秒を表す 2 桁の数字です。
- *media-type* は、[付録A「装置タイプの用語集」](#) および *mcf* のマニュアルページで一覧表示されている 2 文字の Oracle HSM メディアタイプ識別子です。
- *volume* は、ステージング対象のファイルを保持するメディアのボリュームシリアル番号 (VSN) です。

- *position.offset* は、ボリューム上のアーカイブ (*tar*) ファイルの先頭位置を表す 16 進数と、アーカイブファイルの先頭から相対的なステージング済みファイルのオフセットを表す 16 進数をドットで区切ったペアです。
- *inode* は、ステージング済みファイルの i ノード番号と生成番号をドットで区切ったものです。
- *filesize* は、ステージング済みファイルのサイズです。
- *filename* は、ステージング済みファイルの名前です。
- *copy* は、ステージング済みファイルを含むコピーのアーカイブコピー番号です。
- *user* は、ファイルを所有するユーザーです。
- *group* は、ファイルを所有するグループです。
- *requestor* は、ファイルを要求したグループです。
- *equipment-number* は、*mcf* ファイルに定義された、ファイルのステージング元であるドライブの装置番号です。
- *validation* は、ステージング済みファイルが検証済み (V) または未検証 (-) のどちらであるかを示します。

次の例に、一般的なステージャーログの一部を示します。

```
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: ステージング要求数の指定

maxactive ディレクティブでは、一度にアクティブにできる書き込みリクエストの数を指定できます。このディレクティブの形式は、次のとおりです。

```
maxactive=number
```

ここで *number* は、[1-500000] の範囲の整数です。デフォルトは 4000 です。

この例は、キューに同時に存在できるステージング要求が 500 個までであることを指定しています。

```
maxactive=500
```


copyse1: ステージング時のコピー選択順序の指定

ステージングディレクティブ *copyse1* は、ファイルシステムごとにステージャーのコピー選択順序を設定します。

```
copyse1=selection-order
```

selection-order は、最初から最後の順でのコピー番号のコロン区切りリストです。デフォルトの選択順序は *1:2:3:4* です。

詳細は、*stager.cmd* のマニュアルページを参照してください。この例は、ファイルシステム *samfs1* および *samfs2* のデフォルト以外のコピー選択順序を設定する *stager.cmd* ファイルを示しています。

```
logfile = /var/opt/SUNWsamfs/log/stager
drives = hp30 1
fs = samfs1
copyse1 = 4:3:2:1
fs = samfs2
copyse1 = 3:1:4:2
```

プレビュー要求ディレクティブ

Oracle HSM プロセスで、現在ドライブにロードされていないリムーバブルメディアボリュームが要求されると、その要求がプレビューキューに追加されます。キューに入れられた要求は、デフォルトでは先入れ先出し (FIFO) 順で処理されます。ただし、ファイル */etc/opt/SUNWsamfs/preview.cmd* を編集することにより、デフォルトの動作をオーバーライドできます。Oracle HSM ライブラリ制御デーモン (*sam-amlid*) は停止するまで、これらのディレクティブを使用の開始時に読み取ります。キューの優先順位を動的に変更することはできません。

ディレクティブには次の2つのタイプがあります。

- グローバルディレクティブはファイルの先頭に置かれ、すべてのファイルシステムに適用されます。
- ファイルシステムディレクティブは、*fs=directive* 形式を取り、個々のファイルシステムに固有です

次のセクションでは、プレビューキューを制御するように *preview.cmd* ファイルを編集する方法について説明します。

- [グローバルディレクティブ](#)
- [グローバルディレクティブまたはファイルシステム固有のディレクティブ](#)
- [preview.cmd](#) ファイルのサンプル

グローバルディレクティブ

純粋なグローバルディレクティブは、次のとおりです。

- **vsn_priority**: ボリューム優先順位の調整
- **age_priority**: キュー内で待機する時間に応じた優先順位の調整

vsn_priority: ボリューム優先順位の調整

`vsn_priority` ディレクティブは、優先順位の高いボリュームとしてフラグが付けられたボリューム (VSN) の優先順位を、指定された値だけ高くします。ディレクティブの形式は、次のとおりです。

```
vsn_priority=value
```

ここで `value` は実数です。デフォルトは `1000.0` です。

次のコマンドを使用して、ボリュームに高い優先順位のフラグを設定します。

```
chmed +p media-type.volume-serial-number
```

ここで `media-type` は、[付録A「装置タイプの用語集」](#) および `mcf` のマニュアルページに一覧表示されている 2 文字の Oracle HSM メディアタイプのいずれかです。`volume-serial-number` は、ライブラリ内にある優先順位の高いボリュームを一意に識別する英数字の文字列です。詳細な情報については、`chmed` のマニュアルページを参照してください。

age_priority: キュー内で待機する時間に応じた優先順位の調整

`age_priority` ディレクティブは、要求がキュー内に存在する時間に対して指定された相対的な優先順位を変更します。これにより、たとえば、優先順位が高く新しい要求より古い要求が無制限に優先されるようなことを回避できます。このディレクティブは、キュー内で待機する時間の相対的な重み付けを変更する乗数を指定します。形式は次のとおりです。

```
age_priority=weighting-factor
```

ここで *weighting-factor* は、*1.0* よりも大きい、よりも小さい、または等しい実数です。ここでは:

- *1.0* よりも大きい値を指定すると、集計優先順位を計算するときに、キュー内で待機する時間に指定された重み付けが大きくなります。
- *1.0* よりも小さい値を指定すると、合計優先順位を計算するときに、キュー内で待機する時間に指定された重み付けが小さくなります。
- *1.0* と等しい値を指定すると、キュー内で待機する時間に指定された相対的な重み付けは変更されません。

デフォルトは *1.0* です。

グローバルディレクティブまたはファイルシステム固有のディレクティブ

次のディレクティブは、グローバルに、またはファイルシステムごとに適用できます。

- ***hwm_priority***: ディスクキャッシュがほぼ満杯になったときの優先順位の調整
- ***hwm_priority***: ディスクキャッシュがほぼ空になったときの優先順位の調整
- ***lhwm_priority***: ディスクキャッシュが満杯になったときの優先順位の調整
- ***hlwm_priority***: ディスクキャッシュが空になったときの優先順位の調整

hwm_priority: ディスクキャッシュがほぼ満杯になったときの優先順位の調整

hwm_priority ディレクティブは、ファイルシステムの利用率が高位境界値 (*hwm*) を上回ったときに、アーカイブ要求とステージング要求に指定された相対的な重み付けを調整します。高位境界値は、リリーサプロセスが起動され、アーカイブメディア上のコピーを含むファイルで占有されたディスク領域の再利用が開始されるポイントです。このような状況で、アーカイブ処理用に指定された相対的な重み付けを大きくすると、解放プロセスはステージングされたアーカイブコピーおよび新しいファイル用に追加容量を解放できます。ディレクティブの形式は、次のとおりです。

```
hwm_priority=weighting-factor
```

ここで *weighting-factor* は実数です。デフォルトは *0.0* です。

lwm_priority: ディスクキャッシュがほぼ空になったときの優先順位の調整

lwm_priority ディレクティブは、ファイルシステムの利用率が低位境界値 (*lwm*) を下回ったときに、アーカイブ要求とステージング要求に指定された相対的な重み付けを調整します。低位境界値は、リリーサプロセスが停止するポイントです。このような状況で、アーカイブ処理用に指定された相対的な重み付けを小さくして、ステージング要求の優先順位を高くすると、ディスクキャッシュ内に配置されるファイルが増え、メディアマウントの要求が減り、ファイルシステムのパフォーマンスが向上します。ディレクティブの形式は、次のとおりです。

```
lwm_priority=weighting-factor
```

ここで *weighting-factor* は実数です。デフォルトは *0.0* です。

lhwm_priority: ディスクキャッシュが満杯になったときの優先順位の調整

lhwm_priority ディレクティブは、ディスクキャッシュが満杯に近づき、キャッシュの利用率が低位境界値と高位境界値 (*lwm* と *hwm*) の間であるときに、アーカイブ要求とステージング要求に指定された相対的な重み付けを調整します。このような状況で、アーカイブ処理用に指定された相対的な重み付けを大きくすると、解放プロセスはステージングされたアーカイブコピーおよび新しいファイル用に追加容量を解放できます。ディレクティブの形式は、次のとおりです。

```
lhwm_priority=weighting-factor
```

ここで *weighting-factor* は実数です。デフォルトは *0.0* です。

hlwm_priority: ディスクキャッシュが空になったときの優先順位の調整

hlwm_priority ディレクティブは、ディスクキャッシュが空に近づき、キャッシュの利用率が高位境界値と低位境界値 (*hwm* と *lwm*) の間であるときに、アーカイブ要求とステージング要求に指定された相対的な重み付けを調整します。このような状況で、アーカイブ処理用に指定された相対的な重み付けを小さくして、ステージング要求の優先順位を高くすると、ディスクキャッシュ内に配置されるファイルが増え、メディアマウントの要求が減り、ファイルシステムのパフォーマンスが向上します。ディレクティブの形式は、次のとおりです。

```
hlwm_priority=weighting-factor
```

ここで *weighting-factor* は実数です。デフォルトは *0.0* です。

preview.cmd ファイルのサンプル

指定されたメディアマウント要求の集計優先順位は、次の式に従って、すべての重み付け係数で設定された値を使用することで決定されます。

```
priority = vsn_priority + wm_priority + (age_priority * time-waiting-in-queue)
```

ここで *wm_priority* は、現在有効な境界値優先順位 (*hwm_priority*、*lwm_priority*、*hlwm_priority*、または *lhwm_priority*) であり、*time-waiting-in-queue* は、ボリューム要求がキューに入れられている秒数です。優先順位の計算の詳細な説明については、*preview.cmd* のマニュアルページの *PRIORITY CALCULATION* のセクションを参照してください。

データへのアクセスが非常に重要である場合やリムーバブルメディアドライブが不足している場合などの特殊な状況では、*preview.cmd* ファイルでディレクティブを使用すると、さらにファイルシステムアクティビティを操作上の要件および使用可能なリソースに適合させることができます。格納されたデータの整合性は、*preview.cmd* ファイルの設定による影響を受けないため、アーカイブ要求とステージング要求の適切なバランスが見つかるまで、自由に試すことができます。

次の理由の一方または両方のために、デフォルトの優先順位計算の調整が必要になることがあります。

- ユーザーおよびアプリケーションがファイルにアクセスしたときに使用できるように、アーカイブ要求の前にステージング要求が処理されるようにする。
- ファイルシステムが満杯に近づいたときに、アーカイブ要求にもっとも高い優先順位が与えられるようにする。

次の *preview.cmd* ファイルのサンプルは、上記に示した状況に対処します。

```
# 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
```

lwm_priority、*lhwm_priority*、および *hlwm_priority* の重み付けを負の値にすると、ディスクキャッシュ内の領域が使用可能であるときは常に、ステージング要求の優先順位がアーカイブ要求よりも高くなります。これにより、要求があったときは常にデータアクセスが可能になります。キュー内で複数の要求が 100 秒間待機していて、ファイルシステムが低位境界値を下回っている場合は、次のようになります。

- 優先ボリュームに対するアーカイブマウント要求の集計優先順位は
 $1000 + (-200) + (1 \times 100) = 900$
- 優先ボリュームに対するステージングマウント要求の集計優先順位は
 $1000 + 0 + (1 \times 100) = 1100$
- 非優先ボリュームに対するステージングマウント要求の集計優先順位は
 $0 + 0 + (1 \times 100) = 100$

ただし、ディスクキャッシュがほぼ空になったときは、アーカイブ要求を優先する必要があります。ファイルシステムが満杯になったときに、アーカイブされているファイルが非常に少ない場合は、アーカイブされたファイルのステージングや新しいファイルの取り込みに使用できる領域がありません。キュー内で複数の要求が 100 秒間待機していて、ファイルシステムが高位境界値を上回っている場合、次のようになります。

- 優先ボリュームに対するアーカイブマウント要求の集計優先順位は
 $1000 + 500 + (1 \times 100) = 1600$
- 優先ボリュームに対するステージングマウント要求の集計優先順位は
 $1000 + 0 + (1 \times 100) = 1100$
- 非優先ボリュームに対するステージングマウント要求の集計優先順位は
 $0 + 0 + (1 \times 100) = 100$

付録E 製品のアクセシビリティ機能

視力障がいや色覚異常など、視覚に障がいのある方は、コマンド行インタフェースをととして Oracle Hierarchical Storage Manager and StorageTek QFS Software (Oracle HSM) にアクセスできます。このテキストベースのインタフェースにはスクリーンリーダーとの互換性があり、すべての機能はキーボードを使用して制御します。

用語集

この用語集では、Oracle HSM ソフトウェアおよびファイルシステムに固有の用語に焦点を当てています。業界標準の定義については、Storage Networking Industry Association が保守している辞書 (<http://www.snia.org/education/dictionary/>) を参照してください。

アーカイバ	リムーバブルカートリッジへのファイルのコピーを自動制御するアーカイブプログラム。
アーカイブストレージ	アーカイブメディアに作成されるデータストレージ領域。
アーカイブセット	アーカイブセットは、アーカイブされるファイルのグループを識別し、ファイルは、サイズ、所有権、グループ、またはディレクトリの場所に関する共通の条件を共有します。アーカイブセットは、任意のファイルシステムグループ間で定義できます。
アーカイブメディア	アーカイブファイルの書き込み先であるメディア。アーカイブメディアには、リムーバブルなテープカートリッジまたは光磁気カートリッジと、アーカイブ処理用に構成されたディスクファイルシステムの両方が含まれます。
アドレス指定可能ストレージ	Oracle HSM のファイルシステムを通じてユーザーが参照する、オンライン、ニアライン、オフサイト、およびオフラインストレージを包含するストレージ領域。
イーサネット	パケット交換ローカルエリア網のテクノロジー。
オフサイトストレージ	サーバーから離れた遠隔地にあって災害時の障害回復に使用されるストレージ。
オフラインストレージ	ロード時にオペレータの介入を必要とするストレージ。
オンラインストレージ	いつでも利用可能なストレージ (ディスクキャッシュストレージなど)。
カートリッジ	データストレージメディア (磁気テープ、光学メディアなど) の容器。 ボリューム 、テープ、メディアと呼ぶこともあります。「 ボリュームシリアル番号 (VSN) 」を参照してください。
カーネル	基本的なオペレーティングシステム機能を提供するプログラム。UNIX カーネルは、プロセスの作成と管理を行い、ファイルシステムにアクセスする機能を提供し、一般的なセキュリティーを提供し、通信機能を用意します。
カタログ	自動ライブラリにあるリムーバブルメディアボリュームのレコード。1つの自動ライブラリにつき1つのカタログがあり、1つのサイトの自動ライブラリすべてにつき1つの履歴があります。ボリューム

	は、 ボリュームシリアル番号 (VSN) を使用して識別および追跡されます。
クライアントサーバー	あるサイトのプログラムが、別のサイトのプログラムに要求を送って応答を待つ、分散システムにおける対話モデル。要求側のプログラムをクライアントと呼びます。応答を行うプログラムをサーバーと呼びます。
グローバルディレクティブ	すべてのファイルシステムに適用され、最初の fs= 行の前に位置する、アーカイバーディレクティブとリリーサディレクティブ。
スーパーブロック	ファイルシステムの基本パラメータを定義する、ファイルシステム内のデータ構造。スーパーブロックは、ストレージファミリセット内のすべてのパーティションに書き込まれ、セットにおけるパーティションのメンバーシップを識別します。
ステー징	ニアラインまたはオンラインファイルをアーカイブストレージからオンラインストレージにコピーして戻すプロセス。
ストライプサイズ	割り当てられたディスク割り当て単位 (DAU) の数。書き込みがこの数に達すると、ストライプの次のデバイスへ移動します。 stripe=0 マウントオプションを使用した場合、ファイルシステムはストライプ化アクセスではなくラウンドロビン式アクセスを使用します。
ストライプ化	複数のファイルをインタレース方式で論理ディスクに同時に書き込むデータアクセス方法。Oracle HSM ファイルシステムには、ストライプグループを使用する「強いストライプ化」と、 stripe=x マウントパラメータを使用する「弱いストライプ化」の2種類のストライプ化があります。強いストライプ化はファイルシステムの設定時に使用可能にし、 mcf ファイルにストライプ化グループを定義する必要があります。弱いストライプ化は stripe=x マウントパラメータで使用可能にし、ファイルシステムごと、またはファイルごとに変更できます。 stripe=0 を設定すると、無効にできます。強いストライプ化と弱いストライプ化はどちらも、要素数が同じ複数のストライプ化グループでファイルシステムが構成されている場合に使用できます。「 ラウンドロビン 」も参照してください。
ストライプ化グループ	mcf ファイルで1つまたは複数の gXXX デバイスとして定義された、ファイルシステム内のデバイスのコレクション。複数のストライプ化グループは1つの論理デバイスとして扱われ、常にディスク割り当て単位 (DAU) と等しいサイズでストライプ化されます。
ストレージスロット	カートリッジがドライブで使用されていないときに保管される自動ライブラリ内の場所。
ストレージファミリセット	1つの論理デバイスにまとめられている、ディスクのセット。

タイマー	ユーザーが弱い制限値に達してから、このユーザーに強い制限値が課されるまでに経過する時間を追跡する割り当てソフトウェア。
ディスクキャッシュ	オンラインディスクキャッシュとアーカイブメディアとの間でデータファイルの作成と管理に使用する、ファイルシステムソフトウェアのディスクに格納されている部分。個々のディスクパーティションまたはディスク全体で、ディスクキャッシュとして使用できます。
ディスクのストライプ化	アクセスパフォーマンスの向上と全体的なストレージ領域の容量の増大を図るため、1つのファイルを複数のディスクに記録すること。「 ストライプ化 」も参照してください。
ディスクバッファ	SAM-Remote 構成において、クライアントからサーバーにデータをアーカイブするとき使用するサーバーシステム上のバッファ。
ディスク割り当て単位 (DAU)	Oracle HSM ファイルシステムにおいて、書き込まれるデータ量とは関係なく各入出力操作で消費される連続領域の最小量。つまり、ディスク割り当て単位によって、指定サイズのファイルを転送するときに必要な入出力操作の最小回数が決まります。これはディスクデバイスの ブロックサイズ の倍数にする必要があります。 ディスク割り当て単位は、選択された Oracle HSM デバイスタイプおよびユーザー要件によって異なります。 <i>md</i> デバイスタイプでは、デュアル割り当て単位が使用されます。DAU は、ファイルへの最初の 8 回の書き込みでは 4K バイト、後続の書き込みではユーザー指定の 16K、32K、または 64K バイトになるため、小さいファイルは相応の小さいブロックで書き込まれ、大きいファイルは大きいブロックで書き込まれます。 <i>mr</i> および ストライプ化グループ のデバイスタイプでは、 [8-65528]K バイトの範囲内で 8 の単位で調整可能な DAU が使用されます。そのため、ファイルは大きな均一ブロックで書き込まれることになり、大きな均一サイズのファイルのサイズにきわめて近くなります。
ディスク領域しきい値	管理者が定義した、ディスクキャッシュ利用率の最大レベルと最小レベル。リリーサは、これらの事前定義ディスク容量しきい値に基づいて、ディスクキャッシュ利用率を制御します。
ディレクトリ	ファイルシステム内のそのほかのファイルとディレクトリを指す、ファイルデータ構造。
データデバイス	ファイルシステムで、ファイルデータが格納されるデバイスまたはデバイスグループ。

デバイススキャナ	手動でマウントされたリムーバブルデバイスの有無を定期的にモニター監視し、ユーザーやほかのプロセスによって要求されることのある、マウント済みのカートリッジの存在を検出するソフトウェア。
デバイスロギング	Oracle HSM ファイルシステムをサポートするハードウェアデバイスの特定のエラー情報を提供する、構成可能な機能。
ドライブ	リムーバブルメディアボリューム間でデータを転送するためのメカニズム。
トランスポート	「 ロボット 」を参照してください。
ニアラインストレージ	アクセスする前に無人マウントが必要なりムーバブルメディアストレージ。通常、ニアラインストレージはオンラインストレージよりも安価ですが、アクセスに多少時間がかかります。
ネットワーク接続された自動ライブラリ	ベンダー提供のソフトウェアパッケージによって制御される、StorageTek、ADIC/Grau、IBM、Sony などの製品であるライブラリ。QFS のファイルシステムは、自動ライブラリ用に設計された Oracle HSM メディアチェンジャーデーモンを使用して、ベンダーソフトウェアと接続します。
パーティション	デバイスの一部または光磁気カートリッジの片面。
バックアップ	不注意によるファイルの消去を防ぐことを目的とした、ファイル群のスナップショット。バックアップには、ファイルの属性と関連データの両方が含まれます。
ヒストリアン	Oracle HSM ヒストリアンは、 <code>/etc/opt/SUNWsamfs/mcf</code> ファイルで定義されている自動メディアライブラリからエクスポートされたボリュームのカタログです。デフォルトでは、Oracle HSM ファイルシステムホストの <code>/var/opt/SUNWsamfs/catalog/historian</code> にあります。詳細については、Oracle HSM <i>historian</i> のマニュアルページを参照してください。
ファイバチャネル	デバイス間的高速シリアル通信を規定する ANSI 標準。ファイバチャネルは、SCSI-3 におけるバスアーキテクチャーの 1 つとして使用されます。
ファイルシステム	階層構造によるファイルとディレクトリの集まり。
ファイルシステム固有ディレクティブ	<code>archiver.cmd</code> ファイル内のグローバルディレクティブのあとのアーカイバディレクティブとリリーサディレクティブは特定のファイルシステム専用であり、 <code>fs =</code> から始まります。ファイルシステム固有ディレクティブは、次の fs = ディレクティブ行まで、またはファイルの終わりに到達するまで有効です。1 つのファイルシステムを対象

	<p>としたディレクトティブが複数存在する場合、ファイルシステム固有ディレクトティブがグローバルディレクトティブをオーバーライドします。</p>
ファミリーセット	<p>ディスクの集合や、自動ライブラリ内のドライブなど、独立した物理デバイスの論理的なグループ。ストレージファミリーセットも参照してください。</p>
ファミリーデバイスセット	<p>ファミリーセットを参照してください。</p>
ブロックサイズ	<p>ブロックデバイス (ハードディスク、磁気テープカートリッジなど) 上の最小のアドレスابلデータ単位のサイズ。ディスクデバイスでは、これはセクターサイズ (通常 512 バイト) と同等です。</p>
ブロック割り当てマップ	<p>ディスク上のストレージの利用可能な各ブロック。また、これらのブロックが使用中か空いているかを示す、ビットマップです。</p>
ホストファイル	<p>共有ファイルシステム内のすべてのホストの一覧からなるファイル。ファイルシステムを Oracle HSM 共有ファイルシステムとして初期化している場合、ファイルシステムが作成される前にホストファイル <code>/etc/opt/SUNWsamfs/hosts.fs-name</code> を作成する必要があります。<code>sammkfs</code> コマンドは、ファイルシステムを作成するときにホストファイルを使用します。<code>samsharefs</code> コマンドを使用すると、あとでホストファイルの内容を置換または更新できます。</p>
ボリューム	<ol style="list-style-type: none"> 1. ストレージメディア上のアクセス可能な単一の論理ストレージ領域で、通常はボリュームシリアル番号 (VSN) やボリュームラベルによって操作されます。ストレージディスクおよび磁気テープカートリッジは、1 つまたは複数のボリュームを保持できます。使用する場合、ボリュームはファイルシステムの指定されたマウントマウントポイントされます。 2. 単一の論理ボリュームを保持する磁気テープカートリッジ。 3. ランダムアクセスディスクデバイスのファイルシステム、ディレクトリ、またはファイルのことで、順次アクセスのリムーバブルメディアカートリッジ (テープなど) であるかのように構成および使用されます。
ボリュームオーバーフロー	<p>1 つのファイルを複数のボリュームにまたがらせる機能。ボリュームオーバーフローは、個々のカートリッジの容量を超える、非常に大きなファイルを使用するサイトで、便利に利用できます。</p>
ボリュームシリアル番号 (VSN)	<ol style="list-style-type: none"> 1. テープまたはディスクストレージボリュームに割り当てられたシリアル番号。ボリュームシリアル番号は、最大 6 文字の大文字英

数字で構成され、先頭は文字にする必要があります。また、テープライブラリやパーティションといった特定のコンテキストで、ボリュームを一意に特定する必要があります。ボリュームシリアル番号は、ボリュームラベルに書き込まれます。

2. 広義の具体的なストレージ**ボリューム** (特にリムーバブルメディア**カートリッジ**)。

マウントポイント	ファイルシステムがマウントされているディレクトリ。
ミラー書き込み	別々のディスク集合上で1つのファイルのコピーを2つ保管することによって、どちらかのディスクが故障してもデータを消失しないようにしてください。
メタデータ	データに関するデータ。メタデータは、ディスク上のファイルの正確なデータ位置を確認するために使用される索引情報です。ファイル、ディレクトリ、アクセス制御リスト、シンボリックリンク、リムーバブルメディア、セグメントに分割されたファイル、およびセグメントに分割されたファイルのインデックスに関する情報から構成されます。
メタデータデバイス	ファイルシステムのメタデータを保存するデバイス (ソリッドステートディスクやミラーデバイスなど)。ファイルデータとメタデータを別のデバイスに格納すると、パフォーマンスが向上します。メタデータデバイスは、 <i>mcf</i> ファイルにおいて、 <i>ma</i> ファイルシステム内の <i>mm</i> デバイスとして宣言されます。
メディア	テープまたは光磁気ディスクカートリッジ。
メディアリサイクリング	アクティブファイルのあまりないアーカイブメディアをリサイクルまたは再利用するプロセス。
ライブラリ	「 自動ライブラリ 」を参照してください。
ライブラリカタログ	「 カタログ 」を参照してください。
ラウンドロビン	個々のファイル全体を逐次的に論理ディスクに書き込むデータアクセス方法。1つのファイルがディスクに書き込まれるとき、そのファイル全体が第1論理ディスクに書き込まれます。そして、2つめのファイルはその次の論理ディスクに書き込まれる、というふうになります。各ファイルのサイズによって、入出力のサイズが決まります。「 ディスクのストライプ化 」および「 ストライプ化 」も参照してください。
リース	特定の期間中、ファイルを操作するアクセス権をクライアントホストに与える機能。メタデータサーバーは、各クライアントホストに対し

	てリースを発行します。ファイル操作を続行するため、必要に応じてリースが更新されます。
リサイクラ	期限切れアーカイブのコピーが格納されている空間またはカートリッジを回収する、Oracle HSM のユーティリティー。
リムーバブルメディアファイル	磁気テープや光磁気ディスクカートリッジなど、常駐場所であるリムーバブルメディアカートリッジから直接アクセスできる、特殊なタイプのユーザーファイル。アーカイブファイルデータや書き込みファイルデータの書き込みにも使用します。
リモート手続き呼び出し	「 RPC 」を参照してください。
リリーサ	アーカイブされたファイルを識別し、そのディスクキャッシュコピーを開放することで、利用可能なディスクキャッシュ空間を増やす Oracle HSM コンポーネント。リリーサは、オンラインディスクストレージの容量を、上限値と下限値に合わせて自動的に調整します。
ローカルファイルシステム	Solaris Cluster システムの 1 つのノードにインストールされたファイルシステム。ほかのノードからは、あまり利用されません。サーバーにインストールされたファイルシステムのことも指します。
ロボット	ストレージのスロットとドライブとの間でカートリッジを移動する 自動ライブラリ コンポーネント。 トランスポート とも呼ばれます。
解放優先順位	ファイルシステム内のファイルがアーカイブ後に解放される優先順位。解放優先順位は、ファイル属性のさまざまなウェイトを掛け合わせてから、その結果を合計することで計算されます。
回復ポイント	Oracle HSM ファイルシステムのメタデータについてポイントインタイムのバックアップコピーを格納する圧縮ファイル。 ユーザーファイルを不意に削除してしまった場合からファイルシステム全体が壊滅的に失われた場合に至るまで、データ損失時に管理者は、ファイルまたはファイルシステムが完全なままの時点の最新の回復ポイントを見つけるとほぼすぐに、ファイルまたはファイルシステムを最新の既知の良好な状態に回復できます。次に、管理者はその時点で記録されたメタデータを復元します。そして、メタデータに示されているファイルを管理者がアーカイブメディアからディスクキャッシュに書き込むか、または可能であれば、ファイルシステムがユーザーおよびアプリケーションがファイルにアクセスするときに必要な応じてファイルを書き込むようにします。
外部配列	ファイルに割り当てられた各データブロックのディスク上の位置を定義する、ファイルの i ノード内の配列。

割り当て	指定されたユーザー、グループ、または 管理セット ID が消費可能なストレージリソース量。 強い制限値 および 弱い制限値 を参照してください。
監査 (完全)	カートリッジをロードしてカートリッジの VSN を検証する処理。光磁気カートリッジの容量と領域に関する情報が確認され、自動ライブラリのカタログに入力されます。「 ボリュームシリアル番号 (VSN) 」を参照してください。
管理セット ID	共通の特性を共有するユーザーやグループについて、ストレージ管理者が定義したセット。通常、管理セットは、複数のグループからのユーザーが関与し、複数のファイルおよびディレクトリにまたがっているようなプロジェクトのストレージを管理するために作成されます。
間接ブロック	ストレージブロックのリストが入っているディスクブロック。ファイルシステムには、最大 3 レベルの間接ブロックがあります。第 1 レベルの間接ブロックには、データストレージに使用されるブロックのリストが入っています。第 2 レベルの間接ブロックには、第 1 レベルの間接ブロックのリストが入っています。第 3 レベルの間接ブロックには、第 2 レベルの間接ブロックのリストが入っています。
疑似デバイス	関連付けられているハードウェアがないソフトウェアのサブシステムまたはドライバ。
共有ホストファイル	共有ファイルシステムを作成する場合、システムはホストファイルからの情報をメタデータサーバー上の共有ホストファイルへコピーします。この情報は、 samsharefs -u コマンドを発行するときに更新します
強い制限値	割り当て において、指定されたユーザー、グループ、 管理セット ID などが消費可能なストレージリソースの最大の絶対量。「 弱い制限値 」を参照してください。
結合ステージング	グループのいずれかのメンバーに書き込まれるときに、関連ファイルのグループが書き込まれること。ファイルが同じディレクトリにあり、一緒に使用されることがよくある場合、ファイル所有者は Oracle HSM 結合書き込みファイル属性を設定することで、これらを関連付けることができます。その後、グループ内のいずれかのファイルがアプリケーションからアクセスされるときに、グループ内にオフラインのファイルがある場合、Oracle HSM は、グループ全体をアーカイブメディアからディスクキャッシュに書き込みます。これにより、すべての必要なファイルが同時に再度使用可能になります。
検索	ランダムアクセス入出力操作中にディスクデバイスの読み取り/書き込みヘッドをあるディスクの場所から別の場所に移動すること。

高位境界値

1. アーカイブファイルシステムにおいて、Oracle HSM ファイルシステムでリリーサプロセスを開始して、以前にアーカイブされたファイルをディスクから削除するときのディスクキャッシュ利用率(パーセント)。高位境界値が適切に構成されることで、ファイルシステムには新しいファイルや新しく書き込まれるファイル用に使用可能な領域が常に十分あります。詳細については、*sam-releaser* および *mount_samfs* のマニュアルページを参照してください。「**低位境界値**」と比較してください。
2. アーカイブファイルシステムの一部であるリムーバブルメディアライブラリにおいて、リサイクラプロセスを開始するときのメディアキャッシュ使用率(パーセント)。リサイクルすると、現在のデータのフルボリュームの一部が空になるため、新しいメディアと交換したりラベルを付け替えたりすることができます。

事前割り当て

ファイルの書き込みのために、ディスクキャッシュ上の連続した領域を予約するプロセス。事前割り当ては、サイズがゼロのファイルにのみ指定できます。詳細は、*setfa* のマニュアルページを参照してください。

自動ライブラリ

オペレータが処置を必要としない、リムーバブルメディアカートリッジを自動的にロードしたりロード解除したりするように設計された、ロボット制御の装置。自動ライブラリには、1つまたは複数のドライブと、ストレージスロットとドライブの間でカートリッジを移動するトランスポートメカニズムとが含まれています。

弱い制限値

割り当てにおいて、指定されたユーザー、グループ、**管理セット ID**などが無期限で書き込み可能なストレージ領域の最大量。ファイルは、強い制限値を上限として弱い制限値で許可された領域以上を使用できますが、これは割り当てで定義される短い**猶予期間**の間に限られます。「**強い制限値**」を参照してください。

正規表現

ほかの文字列(ファイル名、構成ファイルなど)検索、選択、および編集用に設計された標準化パターンマッチング言語による文字列。Oracle HSM ファイルシステム操作で使用される正規表現構文の詳細については、Oracle HSM Solaris *regex* および *regcmp* のマニュアルページを参照してください。

接続

信頼性の高いストリーム配信サービスを提供する、2つのプロトコルモジュール間のパス。TCP 接続は、1台のマシン上の TCP モジュールと他方上の TCP モジュールをつなぎます。

直接アクセス	ニアラインファイルをアーカイブメディアから直接アクセスすることができるのでディスクキャッシュに取り出す必要がないことを指定する、ファイル属性 (stage never)。
直接接続ライブラリ	SCSI インタフェースを使用してサーバーに直接接続された自動ライブラリ。SCSI 接続のライブラリは、Oracle HSM ソフトウェアによって直接制御されます。
直接入出力	大型ブロック整合逐次入出力に使用される属性の 1 つ。setfa コマンドの -D オプションは、直接入出力のオプションです。このオプションは、ファイルやディレクトリの直接入出力の属性を設定します。ディレクトリに対して設定した直接入出力の属性は、継承されます。
低位境界値	アーカイブファイルシステムにおいて、Oracle HSM ファイルシステムでリリーサプロセスを停止して、以前にアーカイブされたファイルをディスクから削除することを停止するときのディスクキャッシュ利用率 (パーセント)。低位境界値が適切に構成されることで、ファイルシステムでは最高のパフォーマンスを得られるようにできるだけ多くのファイルがキャッシュに保持される一方、新しいファイルや新しくステージングされるファイル用に使用可能な領域を確保します。詳細については、sam-releaser および mount_samfs のマニュアルページを参照してください。高位境界値と比較してください。
複数読み取りファイルシステム	複数のホストにマウント可能なファイルシステムを指定する、シングルライター、マルチリーダー機能。複数のホストがこのファイルシステムを読み込むことができますが、ファイルシステムへの書き込みを行えるのは 1 つのホストだけです。複数のリーダーは、mount コマンドの -o reader オプションによって指定します。シングルライターホストは、mount コマンドの -o writer オプションによって指定します。詳細については、mount_samfs のマニュアルページを参照してください。
名前空間	ファイルおよびその属性と格納場所を示す、ファイル群のメタデータ部分。
猶予期間	割り当てにおいて、ファイルシステムで特定のユーザー、グループ、管理セット ID などに属するファイルの合計サイズが割り当てで指定された弱い制限値を超えてもかまわない時間。
DAU	「ディスク割り当て単位 (DAU)」を参照してください。
FDDI	Fiber-Distributed Data Interface の略で、最大 200 km (124 マイル) まで延長可能な、ローカルエリアネットワークでのデータ転送規格。FDDI プロトコルは、トークンリングプロトコルが基礎になっています。

ftp	File Transfer Protocol の略で、2つのホスト間でファイルを転送するためのネットワークプロトコル。よりセキュアな代替方法については、「 sftp 」を参照してください。
i ノード	索引ノード。ファイルシステムがファイルを記述するとき使用するデータ構造です。i ノードは、名前以外のファイル属性をすべて記述します。ファイル属性には所有権、アクセス、アクセス権、サイズ、およびディスクシステム上におけるファイルの場所などが含まれます。
i ノードファイル	ファイルシステムに常駐しているすべてのファイルの i ノード構造を含む、ファイルシステム上の特殊ファイル (<i>.inodes</i>)。i ノードは長さ 512 バイトです。i ノードファイルは、ファイルシステムのファイルデータから分離されたメタデータファイルです。
LAN	ローカルエリアネットワーク。
LUN	論理ユニット番号 (Logical Unit Number)。
mcf	マスター構成ファイル (Master Configuration File)。ファイルシステム環境内のデバイス (トポロジ) 間の関係を定義する、初期化時に読み取られるファイル。
NFS	Network File System の略で、異機種システム混在ネットワーク上で、リモートファイルシステムへの透過アクセスを提供する、ネットワークファイルシステム。
NIS	Network Information Service の略で、ネットワーク上のシステムとユーザーに関する重要な情報を含む、分散ネットワークデータベース。NIS データベースは、マスターサーバーとすべてのスレーブサーバーに保存されます。
Oracle HSM	<ol style="list-style-type: none"> 1. Oracle Hierarchical Storage Manager の一般的な略語。 2. アーカイブ処理のために構成され、Oracle HSM ソフトウェアによって管理される QFS ファイルシステムを説明する形容詞。
QFS	単独で使用することも、Oracle Hierarchical Storage Manager によって制御されるアーカイブファイルシステムとして使用することもできる、高性能で大容量の UNIX ファイルシステムである Oracle HSM QFS Software 製品。
qfsdump	samfsdump (qfsdump) を参照してください。
qfsrestore	samfsrestore (qfsrestore) を参照してください。
RAID	Redundant Array of Independent Disks。複数の独立したディスクを使用してファイル保存の信頼性を保証するディスク技術です。1つのディ

	スクが故障してもデータを紛失することはなく、耐障害のディスク環境を提供できます。ディスクを個別で使用した場合より、スループットを向上できます。
RPC	リモート手続き呼び出し。カスタムネットワークデータサーバーの実装時に NFS が基盤として使用するデータ交換メカニズムです。
SAM	Oracle Hierarchical Storage Manager 製品の以前の名前である、Storage Archive Manager の一般的な略語。
SAM-QFS	<ol style="list-style-type: none"> 1. 旧バージョンの Oracle Hierarchical Storage Manager 製品の一般的な略称。 2. アーカイブ処理のために構成され、Oracle HSM ソフトウェアによって管理される QFS ファイルシステムを説明する形容詞。
SAM-Remote クライアント	多数の疑似デバイスが含まれ、独自のライブラリデバイスも持つことができる、クライアントデーモンを使用する Oracle HSM システム。クライアントは、SAM-Remote サーバーに依存して 1 つまたは複数のアーカイブのコピーに使用するアーカイブメディアを利用します。
SAM-Remote サーバー	全容量の Oracle HSM ストレージ管理サーバーと、SAM-Remote クライアントが共有するライブラリを定義する SAM-Remote サーバーデーモンの両方。
samfsdump (qfsdump)	制御構造ダンプを作成し、指定したファイル群に関する制御構造の情報をすべてコピーするプログラム。これは通常、ファイルデータのコピーは行いません。-U オプションが指定された場合、このコマンドはデータファイルのコピーも行います。Oracle Hierarchical Storage Manager パッケージがインストールされていない場合、このコマンドは <i>qfsdump</i> と呼ばれます。
samfsrestore (qfsrestore)	i ノードおよびディレクトリの情報を制御構造ダンプから復元するプログラム。 samfsdump (qfsdump) も参照してください。
SAN	ストレージエリアネットワーク。
SCSI	Small Computer System Interface。ディスク、テープドライブ、自動ライブラリなどの周辺デバイスに一般的に使用される電気通信仕様。
sftp	Secure File Transfer Protocol の略で、 ftp を基にした ssh のセキュアな実装。
Small Computer System Interface	「 SCSI 」を参照してください。
ssh	Secure Shell の略で、セキュアなりモートのコマンド行ログインおよびコマンド実行を可能にする暗号化ネットワークプロトコル。

Storage Archive Manager	Oracle Hierarchical Storage Manager 製品の以前の名前。
SUNW.qfs	Oracle HSM 共有ファイルシステムをサポートする Solaris Cluster リソースタイプ。 <i>SUNW.qfs</i> リソースタイプは、共有ファイルシステムのメタデータサーバー (MDS) 用のフェイルオーバーリソースを定義します
tar	テープアーカイブ。イメージのアーカイブに使用される、標準のファイルおよびデータ記録フォーマット。
TCP/IP	Transmission Control Protocol/Internet Protocol。ホストツーホストのアドレッシングとルーティング、パケット配信 (IP)、および信頼性の高いアプリケーションポイント間データ配信 (TCP) を行うインターネットプロトコルです。
vfstab ファイル	<i>vfstab</i> ファイルには、ファイルシステムのマウントオプションが含まれます。コマンド行で指定されたマウントオプションは、 <i>/etc/vfstab</i> ファイル内の指定をオーバーライドし、 <i>/etc/vfstab</i> ファイル内で指定されたマウントオプションは <i>samfs.cmd</i> ファイル内の指定をオーバーライドします。
WORM	Write-Once-Read-Many。書き込みできるのは1回だけで、読み込みは何度でも行えるという、メディアの記録方式です。

索引

た

ドキュメント
入手可能, 16

