



Sun StorageTek™ QFS ファイルシステム構成および 管理マニュアル

Version 4, Update 6

Sun Microsystems, Inc.
www.sun.com

Part No. 820-1716-10
2007年5月, Revision A

コメントの送付: <http://www.sun.com/hwdocs/feedback>

Copyright 2007 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) は、本書に記述されている技術に関する知的所有権を有しています。これら知的所有権には、<http://www.sun.com/patents> に掲載されているひとつまたは複数の米国特許、および米国ならびにその他の国におけるひとつまたは複数の特許または出願中の特許が含まれています。

本書およびそれに付随する製品は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社による事前の許可なく、本製品および本書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

本製品のフォント技術を含む第三者のソフトウェアは、著作権法により保護されており、提供者からライセンスを受けているものです。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。

本製品は、株式会社モリサワからライセンス供与されたリュウミン L-KL (Ryumin-Light) および中ゴシック BBB (GothicBBB-Medium) のフォント・データを含んでいます。

本製品に含まれる HG 明朝 L と HG ゴシック B は、株式会社リコーがリョービマジクス株式会社からライセンス供与されたタイプフェイスマスタをもとに作成されたものです。平成明朝体 W3 は、株式会社リコーが財団法人日本規格協会 文字フォント開発・普及センターからライセンス供与されたタイプフェイスマスタをもとに作成されたものです。また、HG 明朝 L と HG ゴシック B の補助漢字部分は、平成明朝体 W3 の補助漢字を使用しています。なお、フォントとして無断複製することは禁止されています。

Sun、Sun Microsystems、AnswerBook2、docs.sun.com、Java、Solstice DiskSuite、SunPlex、および Sun StorageTek は、米国およびその他の国における米国 Sun Microsystems 社の商標もしくは登録商標です。サンのロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャーに基づくものです。

Mozilla は、米国およびその他の国における Netscape Communications Corporation の商標および登録商標です。

OPENLOOK、OpenBoot、JLE は、サン・マイクロシステムズ株式会社の登録商標です。

ATOK は、株式会社ジャストシステムの登録商標です。ATOK8 は、株式会社ジャストシステムの著作物であり、ATOK8 にかかる著作権その他の権利は、すべて株式会社ジャストシステムに帰属します。ATOK Server/ATOK12 は、株式会社ジャストシステムの著作物であり、ATOK Server/ATOK12 にかかる著作権その他の権利は、株式会社ジャストシステムおよび各権利者に帰属します。

本書で参照されている製品やサービスに関しては、該当する会社または組織に直接お問い合わせください。

OPEN LOOK および Sun™ Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザーおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカル・ユーザー・インターフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

U.S. Government Rights—Commercial use. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

本書には、技術的な誤りまたは誤植のある可能性があります。また、本書に記載された情報には、定期的に変更が行われ、かかる変更は本書の最新版に反映されます。さらに、米国サンまたは日本サンは、本書に記載された製品またはプログラムを、予告なく改良または変更することがあります。

本製品が、外国為替および外国貿易管理法(外為法)に定められる戦略物資等(貨物または役務)に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典:	<i>Sun StorageTek QFS File System Configuration and Administration Guide</i> Part No: 819-7935-10 Revision A
-----	--



Adobe PostScript

目次

はじめに xxvii

1. ファイルシステムの概要 1

ファイルシステムの機能 1

ボリューム管理 2

ページ入出力と直接入出力のサポート 2

大容量 3

ファイルシステムの高速回復 3

メタデータストレージ 4

vnode インタフェース 4

共有ファイルシステムのサポート 4

Linux クライアントのサポート 5

追加のファイルシステム機能 6

設計の基本 7

i ノードファイルとファイルの特徴 7

ディスク割り当て単位の指定 8

DAU 設定とファイルシステムジオメトリ 8

ms および maSun StorageTek SAM ファイルシステム 9

デュアルおよびシングル割り当て方式 10

データ整合 11

データディスクでのストライプ幅	12
ファイル割り当て方式	15
メタデータ割り当て	15
ラウンドロビン式割り当て	16
ストライプ化割り当て	18
ストライプ化グループ	21
不一致のストライプ化グループ	24
不一致のストライプ化グループの例	25
論理ユニット番号 (LUN) による割り当て制御	29
2. ファイル システムの構成	31
File System Manager ソフトウェアの使用	31
▼ はじめて File System Manager を起動する	32
追加の管理者アカウントとユーザーアカウントの作成	32
▼ 追加のアカウントを作成する	33
権限レベルの割り当て	33
複数ユーザーで使用するアカウントの作成	34
▼ File System Manager でアクセスするサーバーを追加する	35
セッションタイムアウトを設定する	36
mcf ファイルの機能	36
「Equipment Identifier」フィールド	37
「Equipment Ordinal」フィールド	38
「Equipment Type」フィールド	38
「Family Set」フィールド	39
「Device State」フィールド	40
「Additional Parameters」フィールド	40
mcf ファイルの例	41
ファイル設定、オプション、指示の相互関係	42
ファイルシステムの初期化	43

構成の例 44

- ▼ Sun StorageTek QFS ラウンドロビン式ディスクの構成を作成する 45
- ▼ Sun StorageTek QFS ストライプ化ディスク構成を作成する 46
- ▼ Sun StorageTek QFS ストライプ化グループ構成を作成する 47

3. 操作の実行 51

ファイルとファイル属性の表示 51

ファイル属性とファイル状態 51

ファイル情報の表示 53

sls(1) 出力について 53

保存行について 55

システムに対する構成ファイルの変更の伝達 55

- ▼ mcf または defaults.conf の情報を Sun StorageTek QFS Sun Cluster 環境で変更する 56
- ▼ mcf または defaults.conf ファイルシステムの情報を SAM-QFS 環境で変更する 57
- ▼ mcf または defaults.conf リムーバブルメディアドライブ情報を変更する 58

共有ホストファイルの変更 59

- ▼ 新しいエントリを追加、または既存のエントリを変更する 59
- ▼ ホスト名の変更、エントリの並べ替え、エントリの挿入を行う 60

マウントパラメータの設定 62

mount(1M) コマンド 62

/etc/vfstab ファイル 63

samfs.cmd ファイル 64

ファイルシステムのマウント解除 65

- ▼ スタンドアロン QFS または SAM-QFS ファイルシステムをマウント解除する 65
- ▼ Sun StorageTek QFS 共有ファイルシステムをマウント解除する 66

ファイルシステムへのディスクキャッシュの追加 66

▼ ファイルシステムにディスクキャッシュを追加する	67
ファイルシステムの再作成	68
▼ ファイルシステムをバックアップして再作成する	68
4. Sun StorageTek QFS 共有ファイルシステムの構成	71
Sun StorageTek QFS 共有ファイルシステムのマウントとマウント解除	72
▼ 共有ファイルシステムをマウントする	72
▼ 共有ファイルシステムをマウント解除する	73
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	74
▼ メタデータサーバーで変換を行う	74
▼ 各クライアントで変換を行う	76
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	78
▼ 各クライアントで変換を行う	78
▼ サーバーで変換を行う	79
クライアントホストの追加または削除	80
▼ クライアントホストを追加する	81
▼ クライアントホストを削除する	86
Sun StorageTek QFS 共有環境での <code>mcf</code> ファイルの更新	88
ローカルホストの構成ファイルの作成	90
Sun StorageTek QFS 環境でのメタデータサーバーの変更	93
▼ 使用可能なメタデータサーバーを変更する	94
▼ 使用不可のメタデータサーバーを変更する	94
SAM-QFS 環境でのメタデータサーバーの変更	96
▼ SAM-QFS 環境でメタデータサーバーを変更する	96
Sun StorageTek QFS 共有ファイルシステムでのクライアントサーバー通信	98
5. ファイルシステム割り当ての管理	99
概要	99
割り当てのタイプ、割り当てファイル、割り当てレコード	100

弱い制限値と強い制限値	101
ディスクブロックとファイル割り当て	101
割り当ての有効化	102
割り当て設定のガイドライン	102
▼ 割り当てを使用するための新しいファイルシステムを構成する	103
▼ 割り当てを使用するための既存ファイルシステムを構成する	104
▼ ディレクトリとファイルへの管理セット ID を割り当てる	107
無限割り当てを設定する	107
▼ 無限割り当てを設定する	107
デフォルトの割り当て値を有効にする	108
▼ ユーザー、グループ、管理セットのデフォルト割り当て値を有効にする	108
割り当てを有効にする	109
▼ ユーザー、グループ、管理セットの割り当て値を有効にする	109
▼ 既存の割り当てファイルを使用してユーザー、グループ、管理セットの割り当て値を有効化または変更する	109
割り当ての検査	111
▼ 超過した割り当てを検査する	111
割り当ての変更と削除	113
▼ 猶予期間を変更する	114
猶予期間の期限を変更する	115
▼ 追加のファイルシステム資源の割り当てを禁止する	117
▼ ファイルシステム割り当てを削除する	119
▼ 割り当てを修正する	121
6. Sun Cluster 環境での Sun StorageTek QFS の構成	123
事前確認	124
制限事項	125
Sun Cluster システムと Sun StorageTek QFS ソフトウェアの相互作用	126
共有ファイルシステムでのデータアクセス	126

非共有ファイルシステムでのデータアクセス 127

Sun Cluster 用 Solaris Volume Manager に対する Sun StorageTek QFS のサポート 127

- ▼ Sun Cluster 用 Solaris Volume Manager でファイルシステムを構成する 128

構成例について 134

Sun Cluster 環境での Sun StorageTek QFS 共有ファイルシステムの構成 135

メタデータサーバーのリソースの検討事項 136

構成例 136

- ▼ Sun StorageTek QFS 共有ファイルシステムの作成を準備する 138
- ▼ Sun StorageTek QFS 共有ファイルシステムを作成し、Sun Cluster ノードを構成する 141
- ▼ 構成の妥当性を検査する 142
- ▼ Oracle Real Application Cluster 用 Sun Cluster データサービスを構成する 144

Sun Cluster 環境での非共有ファイルシステムの構成 146

例 1: raw グローバル装置上の HA-NFS 147

- ▼ 非共有 Sun StorageTek QFS ファイルシステムの作成を準備する 148
- ▼ Sun StorageTek QFS ファイルシステムを作成し、Sun Cluster ノードを構成する 149
- ▼ ネットワークネームサービスと IPMP 妥当性検査を構成する 151
- ▼ HA-NFS を構成し、高可用 Sun StorageTek QFS ファイルシステムを構成する 153

例 2: Solstice DiskSuite/Solaris ボリュームマネージャーで制御されるボリューム上の HA-NFS 155

- ▼ Solstice DiskSuite/Solaris ボリュームマネージャーソフトウェアを準備する 156
- ▼ Sun StorageTek QFS ファイルシステムを準備する 157
- ▼ Sun StorageTek QFS ファイルシステムを作成し、Sun Cluster ノードを構成する 159
- ▼ ネットワークネームサービスと IPMP 妥当性検査を構成する 161

- ▼ HA-NFS を構成し、高可用 Sun StorageTek QFS ファイルシステムを構成する 161

例 3: VxVM ボリューム上の HA-NFS 162

- ▼ VxVM ソフトウェアを構成する 163
- ▼ Sun StorageTek QFS ファイルシステムの作成を準備する 165
- ▼ Sun StorageTek QFS ファイルシステムを作成し、Sun Cluster ノードを構成する 166
- ▼ 構成の妥当性を検査する 167
- ▼ ネットワークネームサービスと IPMP 妥当性検査を構成する 167
- ▼ HA-NFS を構成し、高可用 Sun StorageTek QFS ファイルシステムを構成する 168

クラスタの外部にある共有クライアントの構成 168

構成の前提条件 168

Sun StorageTek QFS メタデータサーバー Sun Cluster ノード 168

Sun StorageTek QFS メタデータクライアントノード 169

Sun Cluster 装置の構成 169

クラスタの外部にクライアントを構成するための要件 169

ソフトウェアの最低限のリリースレベル 170

サポートされるハードウェアアーキテクチャー 170

ストレージの要件 170

構成の手順 170

- ▼ /etc/hosts ファイルを編集する 171
- ▼ メタデータサーバーネットワークを構成する 171
- ▼ メタデータクライアントネットワークを構成する 172
- ▼ Sun Cluster のインストールと構成を行う 172
- ▼ Sun StorageTek QFS メタデータサーバーを構成する 173
- ▼ Sun StorageTek QFS メタデータクライアントを構成する 177

Sun StorageTek QFS の構成の変更 180

- ▼ 共有ファイルシステムの構成を変更する 180

- ▼ raw グローバル装置を使用するファイルシステム上の HA-NFS を無効にする 181
- ▼ Solaris ボリュームマネージャーで制御されるボリュームを使用するファイルシステム上の HA-NFS を無効にする 182
- ▼ VxVM で制御されるボリュームを使用する Sun StorageTek QFS ファイルシステム上の HA-NFS を無効にする 184

Sun Cluster を使用した高可用性 Sun StorageTek SAM 構成 186

7. 高度な機能 187

デーモン、プロセス、およびトレースの使用 187

デーモンとプロセス 188

トレースファイル 189

トレースファイルの内容 189

トレースファイルの切り換え 190

トレース対象プロセスの判別 190

ファイル属性を設定するための `setfa(1)` コマンドの使用 192

ファイルやディレクトリのファイル属性の選択 192

ファイル領域の事前割り当て 192

ファイル割り当て方式とストライプ幅の選択 193

ストライプ化グループ装置の選択 194

WORM-FS ファイルシステムの構成 194

WORM-FS 機能の有効化 195

WORM 「lite」 オプション 197

WORM ファイルの作成 197

例 1. WORM トリガーは `chmod 4000` 198

例 2. WORM トリガーは `chmod 4000` 199

例 3. WORM トリガーは `chmod -w` 200

例 4. WORM トリガーは `chmod -w` 201

保存期間 202

デフォルト保存期間の設定 204

touch を使用した保存期間の設定	205
ファイルの保存期間の延長	207
sls を使用した WORM-FS ファイルの表示	207
sfind を使用した WORM-FS ファイルの検索	208
大容量ファイルの格納	209
複数読み取りファイルシステムの構成	210
異機種システム混在環境における SAN-QFS ファイルシステムの使用	212
事前確認	214
SAN-QFS ファイルシステムの使用開始	214
▼ メタデータコントローラで SAN-QFS ファイルシステムを有効にする	215
▼ クライアントで SAN-QFS ファイルシステムを有効にする	216
▼ クライアントで SANergy ソフトウェアをインストールする	216
SAN-QFS ファイルシステムのマウント解除	217
▼ SANergy クライアントの SAN-QFS ファイルシステムをマウント解除する	217
▼ メタデータコントローラの SAN-QFS ファイルシステムをマウント解除する	218
▼ Sun StorageTek QFS クライアントの SAN-QFS ファイルシステムをマウント解除する	218
▼ Sun StorageTek QFS サーバーの SAN-QFS ファイルシステムをマウント解除する	219
トラブルシューティング: SAN-QFS ファイルシステムのマウント解除と SANergy ファイルホールド	219
▼ SANergy ファイルホールドが存在する状態でファイルシステムをマウント解除する	219
SAN-QFS ファイルシステムにおけるブロック割り当て	220
SAN-QFS ファイルシステムにおけるファイルデータとファイル属性	220
samgrowfs(1M) による SAN-QFS ファイルシステムの拡張	220
SAN-QFS 共有ファイルシステムと Sun StorageTek QFS 共有ファイルシステムの比較	221

- 入出力タイプについて 221
 - ページ入出力 221
 - 直接入出力 222
 - 入出力切り換え 222
- 大容量ファイル転送パフォーマンスの向上 223
 - ▼ ファイル転送パフォーマンスを向上させる 223
- Qwrite 機能の有効化 226
- 書き込みスロットルの設定 227
- 遅延フラッシュ率の設定 228
- iノードの数とiノードハッシュテーブルの調整 229
 - ninodes パラメータ 229
 - nhino パラメータ 230
 - ninodes パラメータと nhino パラメータの設定が必要な場合 230
- A. トラブルシューティング Sun StorageTek QFS 231
 - ファイルシステムの完全性の確認とファイルシステムの修復 231
 - ▼ ファイルシステムを検査する 232
 - ▼ ファイルシステムを修復する 233
 - 共有ファイルシステムでの失敗またはハングアップした `sammkfs(1M)` コマンド
または `mount(1M)` コマンドのトラブルシューティング 233
 - `sammkfs(1M)` コマンドの失敗からの回復 234
 - ▼ `mcf` ファイルを確認し、`mcf` ファイルの変更をシステムに伝達する 234
 - `mount(1M)` コマンドの失敗からの回復 235
 - ▼ ファイルシステムがマウント可能かどうかを確認する 235
 - ▼ `samfsinfo(1M)` コマンドと `samsharefs(1M)` コマンドを使用する 237
 - ▼ `samfsconfig(1M)` コマンドを使用する 239
 - ハングアップした `mount(1M)` コマンドからの回復 240
 - ▼ ネットワーク接続を確認する 240

- ▼ クライアントがサーバーにアクセスできるかどうかを確認する 243
 - ▼ サーバーがクライアントにアクセスできるかどうかを確認する 245
 - ▼ sam-sharefsd トレースログを調査する 246
- Linux クライアントのトラブルシューティング 249
- トラブルシューティングツール 250
 - よくある質問 251
- B. Sun StorageTek QFS 共有ファイルシステムのマウントオプション 253
- バックグラウンドでのマウント: bg オプション 254
 - ファイルシステムマウントの再試行: retry オプション 254
 - Sun StorageTek QFS 共有ファイルシステムの宣言: shared オプション 254
 - 割り当てサイズの調整: minallocsz=*n* および maxallocsz=*n* オプション 255
 - Sun StorageTek QFS 共有ファイルシステムでのリースの使用: rdlease=*n*、
wrlease=*n*、および aplease=*n* オプション 255
 - 複数ホストの読み取りと書き込みの有効化: mh_write オプション 256
 - 並行スレッドの最小数の設定: min_pool=*n* オプション 258
 - キャッシュした属性の保持: meta_timeo=*n* オプション 259
 - ストライプ化割り当ての指定: stripe オプション 259
 - メタデータの書き込み頻度の指定: sync_meta=*n* オプション 259
 - WORM 機能の有効化: worm_capable および def_retention オプション 260
- C. samu(1M) オペレータユーティリティーの使用法 261
- 概要 261
 - ▼ samu(1M) を起動する 262
 - ▼ samu(1M) 画面を表示する 262
 - ▼ samu(1M) を停止する 263
 - samu(1M) とのやりとり 263
 - 装置の入力 263
 - オンラインヘルプの表示 264
 - ▼ 表示画面からオンラインヘルプにアクセスする 264

オペレータ表示	264
(a) - アーカイバの状態表示	265
ナビゲーション	265
表示例	266
フィールドの説明	267
(c) - デバイス構成表示	267
ナビゲーション	268
表示例	268
フィールドの説明	269
(c) - メモリー表示	269
表示例	270
(d) - デモントレースコントロールの表示	270
表示例	271
(D) - ディスクボリュームディクショナリ	272
表示例	272
フラグ	273
(f) - ファイルシステムの表示	273
表示例	273
フィールドの説明	275
(F) - 光磁気ディスクラベル表示	275
(h) - ヘルプ表示	276
ナビゲーション	276
表示例	276
(I) - iノード表示	277
ナビゲーション	278
表示例	278
(J) - プレビュー共用メモリー表示	279
ナビゲーション	279

表示例	279
(K) - カーネル統計情報表示	280
ナビゲーション	280
表示例	280
(l) - 使用率の表示	281
表示例	281
(L) - 共用メモリー表示	282
表示例	282
(m) - 外部ストレージの状態表示	283
表示例	283
フィールドの説明	284
(M) - 共用メモリー表示	284
ナビゲーション	285
表示例	285
(n) - 書き込みの状態表示	286
表示例	286
(N) - ファイルシステムパラメータ表示	287
ナビゲーション	287
表示例	287
(o) - 光磁気ディスクの状態表示	288
ナビゲーション	288
表示例	289
フィールドの説明	289
(p) - リムーバブルメディアの読み込み要求の表示	290
ナビゲーション	291
表示例	291
フィールドの説明	292
フラグ	292

- (P) - アクティブサービス表示 293
 - ナビゲーション 293
 - 表示例 293
- (r) - リムーバブルメディアの状態表示 293
 - 表示例 294
 - フィールドの説明 294
- (R) - Sun SAM-Remote 情報表示 295
- (s) - デバイスの状態表示 295
 - ナビゲーション 296
 - 表示例 296
 - フィールドの説明 297
- (S) - セクターデータ表示 297
 - ナビゲーション 297
- (t) - テープドライブの状態表示 298
 - ナビゲーション 298
 - 表示例 298
 - フィールドの説明 299
- (T) - SCSI センスデータ表示 299
 - ナビゲーション 300
- (u) - 書き込み待ち行列表示 300
 - ナビゲーション 300
 - 表示例 301
 - フィールドの説明 301
- (U) - 装置テーブル表示 302
 - ナビゲーション 302
 - 表示例 303
- (v) - 自動ライブラリカタログ表示 303
 - ナビゲーション 304

表示例	305
フィールドの説明	306
フラグ	306
(w) - 保留書き込み待ち行列の表示	307
ナビゲーション	308
表示例	308
フィールドの説明	309
オペレータ表示の状態コード	309
リムーバブルメディア装置表示の状態コード	309
ファイルシステム表示の状態コード	310
オペレータ表示のデバイスの状態	311
▼ ドライブの状態を down から on に変更する	313
▼ ドライブの状態を on から down に変更する	313
オペレータコマンド	314
デバイスコマンド	315
ファイルシステムコマンド: 入出力管理	315
:flush_behind <i>eq value</i> コマンド	315
:force_nfs_async <i>eq</i>	
コマンドと :noforce_nfs_async <i>eq</i> コマンド	316
:readahead <i>eq contig</i> コマンド	316
:sw_raid <i>eq</i> コマンドと :nosw_raid <i>eq</i> コマンド	317
:writebehind <i>eq contig</i> コマンド	317
:wr_throttle <i>eq value</i> コマンド	317
ファイルシステムコマンド: 直接入出力管理	318
:dio_rd_form_min <i>eq value</i>	
コマンドと :dio_wr_form_min <i>eq value</i> コマンド	318
:dio_rd_ill_min <i>eq value</i>	
コマンドと :dio_wr_ill_min <i>eq value</i> コマンド	318
:dio_rd_consec <i>eq value</i>	
コマンドと :dio_wr_consec <i>eq value</i> コマンド	319

:dio_szero *eq* コマンドと :nodio_szero *eq* コマンド 319
 :forcedirectio *eq*
 コマンドと :noforcedirectio *eq* コマンド 320
 ファイルシステムコマンド: Sun StorageTek QFS 共有ファイルシステム
 320
 :meta_timeo *eq interval* コマンド 320
 :mh_write *eq* コマンドと :nomh_write *eq* コマンド 320
 :minallocsz *eq value*
 コマンドと :maxallocsz *eq value* コマンド 321
 :rdlease *eq interval* コマンド、:wrlease *eq interval* コマンド、
 :aplease *eq interval* コマンド 321
 ファイルシステムコマンド: そのほか 321
 :abr *eq* コマンドと :noabr *eq* コマンド 321
 :dmr *eq* コマンドと :nodmr *eq* コマンド 322
 :invalid *eq interval* コマンド 322
 :mm_stripe *eq value* コマンド 322
 :qwrite *eq*
 コマンドと :noqwrite *eq* コマンド 322
 :refresh_at_eof *eq*
 コマンドと :norefresh_at_eof *eq* コマンド 323
 :suid *eq* コマンドと :nosuid *eq* コマンド 323
 :stripe *eq value* コマンド 323
 :sync_meta *eq value* コマンド 324
 :trace *eq* コマンドと :notrace *eq* コマンド 325
 そのほかのコマンド 325
 :clear *vsname [index]* コマンド 325
 :devlog *eq [option]* コマンド 325
 :diskvols *volume [+flag | -flag]* コマンド 326
 :dtrace コマンド 326
 :fs *fsname* コマンド 327
 :mount *mntpt* コマンド 327

:open *eq* コマンド 327
:read *addr* コマンド 327
:refresh *i* コマンド 328
:snap [*filename*] コマンド 328
:!*shell_command* コマンド 328

用語集 329

索引 343

目次

図 1-1	5 台の装置を使用した、ms ファイルシステムでのラウンドロビン割り当て	17
図 1-2	5 台の装置を使用した、ma ファイルシステムでのラウンドロビン割り当て	18
図 1-3	5 台の装置を使用した、ms ファイルシステムでのストライプ化	20
図 1-4	5 台の装置を使用した、ma ファイルシステムでのストライプ化	21
図 1-5	Sun StorageTek QFS のラウンドロビン式ストライプ化グループ	22
図 1-6	Sun StorageTek QFS のストライプ化グループ割り当て	24
図 1-7	ストライプ化割り当てにおいて不一致のストライプ化グループを使用する Sun StorageTek QFS ファイルシステム	26
図 4-1	ネットワークインタフェース	92
図 7-1	Sun StorageTek QFS ソフトウェアと SANergy ソフトウェアを使用した SAN-QFS ファイルシステム	213

表目次

表 1-1	製品の概要	1
表 1-2	.inode ファイルの内容	8
表 1-3	Sun StorageTek QFS 装置タイプの値および DAU のサイズ	11
表 1-4	ms ファイルシステムでのデフォルトのストライプ幅	12
表 1-5	デフォルトのストライプ幅	13
表 1-6	デフォルトの割り当て方式	15
表 1-7	メタデータ割り当て	15
表 1-8	ファイルシステム例の特徴	27
表 2-1	File System Manager アクセス権レベル	33
表 2-2	Sun StorageTek QFS または SAM-QFS の「Equipment Type」フィールド	39
表 3-1	ユーザー指定ファイル属性	52
表 3-2	sls(1) 出力の説明	53
表 4-1	ローカルホストの構成ファイルのフィールド	90
表 5-1	割り当てファイル名	100
表 7-1	デーモンとプロセス	188
表 7-2	ファイル割り当てとストライプ幅	193
表 7-3	SAN-QFS 共有ファイルシステムと Sun StorageTek QFS 共有ファイルシステム	221
表 A-1	エラー表示	231
表 A-2	/proc のファイル	250
表 B-1	リース関連の mount(1M) のオプション	256

表 B-2	mh_write オプションに基づくファイルアクセス	257
表 C-1	a 表示のコントロールキー	265
表 C-2	:a <i>filesystem</i> 表示のコントロールキー	266
表 C-3	samu(1M) a の表示フィールドの説明	267
表 C-4	c 表示のコントロールキー	268
表 C-5	samu(1M) c の表示フィールドの説明	269
表 C-6	samu(1M) D 表示の flags フィールド	273
表 C-7	samu(1M) f の表示フィールドの説明	275
表 C-8	h 表示のコントロールキー	276
表 C-9	I 表示のコントロールキー	278
表 C-10	J 表示のコントロールキー	279
表 C-11	p 表示のコントロールキー	280
表 C-12	samu(1M) m の表示フィールドの説明	284
表 C-13	M 表示のコントロールキー	285
表 C-14	N 表示のコントロールキー	287
表 C-15	o 表示のコントロールキー	288
表 C-16	samu(1M) o の表示フィールドの説明	289
表 C-17	p 表示のコントロールキー	291
表 C-18	samu(1M) p の表示フィールドの説明	292
表 C-19	samu(1M) p 表示の flags フィールド	292
表 C-20	P 表示のコントロールキー	293
表 C-21	samu(1M) r の表示フィールドの説明	294
表 C-22	s 表示のコントロールキー	296
表 C-23	samu(1M) s の表示フィールドの説明	297
表 C-24	S 表示のコントロールキー	297
表 C-25	t 表示のコントロールキー	298
表 C-26	samu(1M) t の表示フィールドの説明	299
表 C-27	T 表示のコントロールキー	300
表 C-28	u 表示のコントロールキー	300
表 C-29	samu(1M) u の表示フィールドの説明	301

表 C-30	U 表示のコントロールキー	302
表 C-31	v 表示のコントロールキー	304
表 C-32	samu(1M) v の表示フィールドの説明	306
表 C-33	samu(1M) v 表示の flags フィールド	306
表 C-34	w 表示のコントロールキー	308
表 C-35	samu(1M) w の表示フィールドの説明	309
表 C-36	リムーバブルメディア装置表示の状態コード	309
表 C-37	ファイルシステム表示の状態コード	311
表 C-38	オペレータ表示のデバイスの状態	311
表 C-39	デバイスコマンドのアクション	315
表 C-40	トレースコマンドの引数	326

はじめに

このマニュアル、『Sun StorageTek™ QFS ファイルシステム構成および管理マニュアル』では、Sun StorageTek QFS Version 4, Update 6 (4U6) に組み込まれているファイルシステムのソフトウェアについて説明します。

Sun StorageTek QFS 製品には、スタンドアロンのファイルシステムとしても、共有ファイルシステムとしても使用でき、また、Sun StorageTek SAM 製品に含まれているストレージ/アーカイブ管理機能と組み合わせて使用することもできるファイルシステムが組み込まれています。Sun StorageTek Storage Archive Manager (SAM) ソフトウェアと組み合わせて使用する場合は、SAM-QFS と呼ばれます。

Sun StorageTek QFS ソフトウェアパッケージは、次のオペレーティングシステム (OS) 環境で実行できます。

- SPARC® プラットフォーム版 Solaris™ 9 オペレーティングシステム (OS) 04/03 以降
- SPARC または x64 プラットフォーム版 Solaris 10 OS
- x86 (32 ビット) 版 Solaris 10 OS (Sun StorageTek QFS 共有クライアントのみ)
- x86/x64 プラットフォーム版 Red Hat Enterprise Linux 3.0 (UD-5 および UD-8) (Sun StorageTek QFS 共有クライアントのみ)
- x64 プラットフォーム版 Red Hat Enterprise Linux 4.0 (UD-2 および UD-4) (Sun StorageTek QFS 共有クライアントのみ)
- x64 プラットフォーム版 SuSE Linux Enterprise Server 8 (Service Pack 4) (Sun StorageTek QFS 共有クライアントのみ)
- x64 プラットフォーム版 SuSE Linux Enterprise Server 9 (Service Pack 2)、および SGI Altix Itanium システム (Sun StorageTek QFS 共有クライアントのみ)
- x64 プラットフォーム版 SuSE Linux Enterprise Server 10 (Sun StorageTek QFS 共有クライアントのみ)

注 - これは、SuSE Enterprise Linux 8 をサポートする最後のバージョンの Sun StorageTek QFS です。次のバージョンのソフトウェアでは、SuSE Enterprise Linux 8 はサポートされません。

このマニュアルは、Sun StorageTek QFS ファイルシステムのインストール、構成、および保守を担当しているシステム管理者向けに書かれています。対象読者であるシステム管理者は、インストール、構成、アカウントの作成、システムバックアップの実行、そのほか Solaris に関する基本的なシステム管理作業の実行をはじめとする、Solaris OS の作業手順に精通している必要があります。

マニュアルの構成

このマニュアルは次の章で構成されています。

- 第 1 章では、ソフトウェアの概要情報を提供しています。
- 第 2 章では、ファイルシステムの構成に関する情報を提供しています。
- 第 3 章では、ファイルシステムの初期化、サーバーの追加、ディスクキャッシュの追加、そのほかのシステム管理アクティビティーなど、さまざまな作業の実行方法を説明します。
- 第 4 章では、Sun StorageTek QFS 共有ファイルシステムの構成方法を説明します。
- 第 5 章では、ファイルシステム割り当ての使用方法を説明します。
- 第 6 章では、Sun™ Cluster 環境での Sun StorageTek QFS ソフトウェアの構成方法を説明します。
- 第 7 章では、マルチリーダーファイルシステムやパフォーマンス改善機能の使用方法など、そのほかの高度な事項について説明します。
- 付録 A では、Sun StorageTek QFS ソフトウェアのトラブルシューティングに関する情報を提供します。
- 付録 B では、Sun StorageTek QFS 共有ファイルシステムのマウントオプションの一覧を提供します。
- 付録 C では、samu(1M) オペレータユーティリティーの使用方法を説明します。

UNIX コマンド

このマニュアルには、システムの停止、システムの起動、およびデバイスの構成などに使用する基本的な UNIX[®] コマンドと操作手順に関する説明は含まれていない可能性があります。これらについては、次を参照してください。

- 使用しているシステムに付属のソフトウェアマニュアル
- 次の URL にある、Solaris オペレーティングシステムのマニュアル
<http://docs.sun.com>

シェルプロンプトについて

表 P-1 に、このマニュアルで使用するシェルプロンプトを示します。

表 P-1 シェルプロンプトについて

シェル	プロンプト
UNIX の C シェル	<i>machine_name%</i>
UNIX の Bourne シェルと Korn シェル	\$
スーパーユーザー (シェルの種類を問わない)	#

書体と記号について

表 P-2 に、このマニュアルで使用する表記規則を示します。

表 P-2 表記規則

書体または記号*	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例。	.login ファイルを編集します。 ls -a を実行します。 % You have mail.
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して表します。	% su Password:
<i>AaBbCc123</i>	コマンド行の可変部分。実際の名前や値と置き換えてください。	rm <i>filename</i> と入力します。
[]	コマンド構文で、角括弧は、引数が任意であることを表します。	scmadm [-d sec] [-r n[:n][,n]...] [-z]
{ arg arg }	コマンド構文で、中括弧および縦棒は、引数を 1 つ指定する必要があります。	sndradm -b { <i>phost</i> <i>shost</i> }
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	% grep `^#define \ XV_VERSION_STRING`
『 』	参照する書名を示します。	『Solaris ユーザーマニュアル』
「 」	参照する章、節、または、強調する語を示します。	第 6 章「データの管理」を参照。 この操作ができるのは「スーパーユーザー」だけです。

* 使用しているブラウザにより、これらの設定と異なって表示される場合があります。

関連マニュアル

このマニュアルは、Sun StorageTek QFS および Sun StorageTek SAM ソフトウェア製品の操作方法を説明するマニュアルセットの一部です。表 P-3 に、これらの製品の release 4U6 に関するマニュアルセット一式を示します。

表 P-3 Sun StorageTek の関連マニュアル

タイトル	Part No.
Sun StorageTek QFS インストールおよびアップグレードの手引き	820-1721-10
Sun StorageTek Storage Archive Manager (SAM) ファイルシステム構成および管理マニュアル	820-1741-108
Sun StorageTek Storage Archive Manager (SAM) アーカイブ構成および管理マニュアル	820-1726-10
Sun StorageTek Storage Archive Manager (SAM) インストールおよびアップグレードの手引き	820-1731-10
Sun StorageTek Storage Archive Manager (SAM) トラブルシューティングマニュアル	820-1736-10
Sun StorageTek QFS, Sun StorageTek Storage Archive Manager (SAM) ご使用にあたって Version 4, Update 6	820-1746-10
Sun StorageTek QFS Linux クライアントガイド	820-1751-10

Sun のオンラインマニュアル

Sun StorageTek QFS ソフトウェアのディストリビューションには、Sun のネットワークストレージ関連のドキュメント Web サイト、または docs.sun.com から表示できる PDF ファイルが含まれています。

docs.sun.com からマニュアルにアクセスする

このウェブサイトには、Solaris ほか、多数の Sun ソフトウェア製品のマニュアルが用意されています。

1. このウェブサイトには、次の URL からアクセスできます。

<http://docs.sun.com>

docs.sun.com ページが表示されます。

2. サーチボックスで Sun StorageTek QFS を検索し、目的の製品のマニュアルを見つけます。

Sun のネットワークストレージ関連のマニュアルのサイトにアクセスする

このウェブサイトには、ネットワークストレージ関連の製品のマニュアルが用意されています。

1. このウェブサイトには、次の URL からアクセスできます。

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

「Storage Software」 ページが表示されます。

2. 「Sun StorageTek QFS Software」のリンクをクリックします。

Sun 以外の Web サイト

このマニュアルで紹介する Sun 以外の Web サイトが使用可能かどうかについては、Sun は責任を負いません。このようなサイトやリソース上、またはこれらを経由して利用できるコンテンツ、広告、製品、またはその他の資料についても、Sun は保証しておらず、法的責任を負いません。また、このようなサイトやリソース上、またはこれらを経由して利用できるコンテンツ、商品、サービスの使用や、それらへの依存に関連して発生した実際の損害や損失、またはその申し立てについても、Sun は一切の責任を負いません。

Sun の技術サポート

このマニュアルには掲載されていない本製品に関する技術的なご質問は、次の Web サイトからお寄せください。

<http://www.sun.com/service/contacting>

ライセンス

Sun StorageTek QFS および Sun StorageTek SAM ソフトウェアのライセンスの入手については、ご購入先にお問い合わせください。

インストールのサポート

インストールと構成のサービスについては、Sun の Enterprise Services (1-800-USA4SUN) またはご購入先にお問い合わせください。

コメントをお寄せください

マニュアルの品質改善のため、お客様からのご意見およびご要望をお待ちしております。コメントは下記よりお送りください。

<http://docs.sun.com/app/docs/form/comments>

ご意見をお寄せいただく際には、下記のタイトルと Part No. を記載してください。

Sun StorageTek QFS ファイルシステム構成および管理マニュアル, 820-1716-10

第1章

ファイルシステムの概要

この章は、次の節で構成されています。

- 1 ページの「ファイルシステムの機能」
 - 7 ページの「設計の基本」
 - 15 ページの「ファイル割り当て方式」
-

ファイルシステムの機能

Sun StorageTek QFS ファイルシステムは構成可能なファイルシステムで、ユーザーに標準の UNIX ファイルシステム (UFS) インタフェースを提供します。表 1-1 に、このファイルシステムとストレージ/アーカイブ管理 (Sun StorageTek SAM) ソフトウェアとの使用または組み合わせの方法を示します。

表 1-1 製品の概要

製品	コンポーネント
Sun StorageTek QFS ファイルシステム	スタンドアロンのファイルシステム。
Sun StorageTek QFS 共有ファイルシステム	複数のホストシステムにマウントできる分散ファイルシステム。

表 1-1 製品の概要 (続き)

製品	コンポーネント
SAM ファイルシステム	Sun StorageTek SAM ソフトウェアとともに組み込まれるファイルシステム。Sun StorageTek QFS ファイルシステムの一部の機能は、このファイルシステムに組み込まれていません。
SAM-QFS	Sun StorageTek QFS ソフトウェアと Sun StorageTek SAM ソフトウェアを組み合わせると、Sun StorageTek QFS 製品の高度なファイルシステム機能と、Sun StorageTek SAM 製品のストレージ管理機能を同時に利用できます。この組み合わせは SAM-QFS と呼ばれています。 注: 特に注記のないかぎり、このマニュアルで示される Sun StorageTek QFS ファイルシステムに関する情報は、SAM-QFS の構成にも適用されます。

Sun StorageTek QFS ファイルシステムでは、ユーザープログラムまたは UNIX カーネルを変更する必要はありません。以降の項目では、Sun StorageTek QFS ファイルシステムの機能のいくつかを説明します。

ボリューム管理

Sun StorageTek QFS ファイルシステムは、ストライプ化とラウンドロビンの両方のディスクアクセスをサポートしています。マスター構成ファイル (mcf) とマウントパラメータを使用して、ボリューム管理機能を指定し、制御する装置間の関係をファイルシステムが認識できるようにします。これは、1 つの装置または装置の一部分だけしか指定できない多くの UNIX システムとは対照的です。Sun StorageTek QFS ファイルシステムは、追加のボリューム管理アプリケーションを必要としません。ただし、Sun StorageTek QFS 環境の装置でミラーリングを使用する場合は、論理ボリュームマネージャーなどの追加パッケージを入手する必要があります。

Sun StorageTek QFS に組み込まれたボリューム管理機能では、標準の Solaris OS のデバイスドライバインタフェースを使用して、入出力要求を下位の装置との間で受け渡します。Sun StorageTek QFS ソフトウェアは、各ファイルシステムが常駐するストレージをファミリーセットにグループ化します。

ページ入出力と直接入出力のサポート

Sun StorageTek QFS ファイルシステムは、2 つのタイプの入出力、すなわちページ入出力 (キャッシュ入出力またはバッファー入出力とも呼ばれる) と直接入出力をサポートします。これらの入出力タイプでは、次の処理が行われます。

- ページ入出力を使用する場合、ユーザーデータは仮想メモリーページにキャッシュされ、カーネルがデータをディスクに書き込みます。標準 Solaris OS インタフェースによって、ページ入出力が管理されます。これが、Sun StorageTek QFS のデフォルトの入出力です。
- 直接入出力を使用する場合、ユーザーデータがユーザーメモリーからディスクに直接書き込まれます。直接入出力を指定するには、Solaris OS の `directio(3C)` 関数呼び出しを使用するか、`setfa(1)` コマンドで `-D` オプションを使用します。大容量ブロックの境界割り当てされた逐次入出力では、直接入出力を使用することで、パフォーマンスが大幅に向上します。

大容量

Sun StorageTek QFS ソフトウェアがサポートするファイルの最大サイズは 2^{63} バイトです。このように大容量のファイルは、1つのファイルシステム内でも、多数のディスクまたは RAID デバイスにストライブ化できます。これは Sun StorageTek QFS ファイルシステムが 64 ビットアドレッシングを使用しているためであり、純粋な 64 ビットシステムではない標準の UNIX ファイルシステムとは対照的です。

構成できるファイルシステムの数、実質的には無制限です。ボリュームマネージャーを使用すると、各ファイルシステムは最大 252 のデバイスパーティション (通常はディスク) を含むことができます。各パーティションは最大 16T バイトのデータを格納できます。この構成により、実質的に無制限の記憶容量が提供されます。

Sun StorageTek QFS ファイルシステム内のファイル数には、事前定義された制限はありません。ファイルの情報を保持する `i` ノード領域は動的に割り当てられるため、ファイルの最大数は、使用可能なディスクストレージの容量によってのみ制限されます。`i` ノードは、マウントポイントの下の `.inodes` ファイルに記録されます。`.inodes` ファイルは、1 ファイルにつき 512 バイトの記憶領域が必要です。

Sun StorageTek QFS ファイルシステムでは、`i` ノードはメタデータ装置にあり、ファイルデータ装置とは切り離されています。実際には、メタデータ (`mm`) 装置のサイズによって Sun StorageTek QFS ファイルシステム内のファイル数が制限されますが、メタデータ装置を追加することで、ファイルの最大数を増やすことができます。ファイル数の物理的な上限は $2^{32}-1$ 個で、推奨する上限は 10^7 個です。

ファイルシステムの高速回復

ファイルシステムの重要な機能は、ファイルシステムの予定外の停止に対して迅速に回復する能力です。標準の UNIX ファイルシステムでは、システム障害のあとで、不一致を修正するために長時間のファイルシステムの検査 (`fsck(1M)`) が必要です。

Sun StorageTek QFS ファイルシステムでは、ファイルシステムが (`sync(1M)` を使用して) ディスクに書き込めなくなる障害が発生したあとでも、多くの場合、ファイルシステムの検査をする必要はありません。また、Sun StorageTek QFS ファイルシス

テムは、ジャーナルを使用しないで、システム障害から回復します。識別レコード、逐次書き込み、およびすべてのクリティカルな入出力操作に対するエラー検査を使用して、動的に回復します。システム障害後、テラバイト単位の Sun StorageTek QFS ファイルシステムであっても、すぐに再マウント可能です。

メタデータストレージ

ファイルシステムは、メタデータを使用してファイルとディレクトリの情報を参照します。通常、メタデータはファイルデータと同じデバイスに常駐します。ただし、Sun StorageTek QFS ファイルシステムは、オプションで、ファイルシステムのメタデータを別のデバイスに格納することでファイルデータから分離します。Sun StorageTek QFS ファイルシステムを使用すると、1 つ以上の個別のメタデータ装置を定義することができ、デバイスのヘッド移動や回転応答時間を減らし、RAID キャッシュの使用率を向上させたり、ファイルデータをミラー化せずにメタデータをミラー化することができます。

Sun StorageTek QFS ファイルシステムは、i ノードメタデータ情報を別ファイルに保存します。これによって、ファイル数やファイルシステム全体を動的に拡大することが可能になります。

vnode インタフェース

Sun StorageTek QFS ファイルシステムは、標準の Solaris オペレーティングシステム (OS) 仮想ファイルシステム (vfs/vnode) インタフェースを使用して実装されます。

vfs/vnode インタフェースを使用することで、このファイルシステムは標準の Solaris OS カーネルとともに動作するので、ファイル管理サポートのためにカーネルを変更する必要がありません。このように、ファイルシステムは、オペレーティングシステムの変化の影響を受けないので、通常は、オペレーティングシステムの更新時に大規模な回帰テストを行う必要がありません。

カーネルは、Sun StorageTek QFS ファイルシステムに常駐するものも含めて、ファイルに対するすべての要求を受け取ります。Sun StorageTek QFS のファイルと見なされるファイルの場合、カーネルは該当するファイルシステムに要求を渡して、処理できるようにします。Sun StorageTek QFS ファイルシステムは、`/etc/vfstab` ファイルおよび `mount(1M)` コマンドでタイプ `samfs` と識別されます。

共有ファイルシステムのサポート

Sun StorageTek QFS 共有ファイルシステムは、複数の Solaris OS ホストシステムにマウントできる分散ファイルシステムです。Sun StorageTek QFS 共有ファイルシステム環境では、1 つの Solaris OS ホストがファイルシステムのメタデータサーバーと

して動作し、そのほかのホストはクライアントとして構成できます。複数のホストを潜在的なメタデータサーバーとして構成できますが、同時にメタデータサーバーにできるホストは1つだけです。Sun StorageTek QFS 共有ファイルシステムのマウントポイント数に制限はありません。

Sun StorageTek QFS 共有ファイルシステムの利点は、ファイバチャネルディスクからホストにファイルデータが直接渡ることです。データは、ローカルバス入出力 (直接アクセス入出力とも呼ばれる) を介してやりとりされます。これは、ネットワーク上でデータを転送するネットワークファイルシステム (NFS) とは対照的です。

共有ファイルシステムは、Sun StorageTek QFS 共有ファイルシステムまたは SAM-QFS 共有ファイルシステムとして実装できます。ms ファイルシステムタイプ、または、ma ファイルシステムタイプを使用します。

Sun StorageTek QFS 共有ファイルシステムは次のものをサポートしていません。

- 次のファイルタイプ
 - b - ブロック特殊ファイル
 - c - キャラクタ特殊ファイル
 - p - FIFO (名前付きパイプ) 特殊ファイル
- セグメント化ファイル。セグメント化したファイル環境では、SAM-QFS 共有ファイルシステムは実装できません。
- 必須のロック。必須のロックを設定すると、EACCES エラーが戻されます。ただし、アドバイザリロックはサポートされています。アドバイザリロックの詳細は、fcntl(2) システムコールを参照してください。

共有ファイルシステムについての詳細は、第4章、71 ページの「Sun StorageTek QFS 共有ファイルシステムの構成」を参照してください。

Linux クライアントのサポート

共有ファイルシステム内では、Solaris クライアントだけでなく Linux クライアントにも Sun StorageTek QFS ソフトウェアをインストールできます。Sun StorageTek QFS Linux クライアントソフトウェアは、次の Linux ディストリビューションをサポートします。

- x86/x64 プラットフォーム版 Red Hat Enterprise Linux 3.0 (UD-5 および UD-8) (Sun StorageTek QFS 共有クライアントのみ)
- x64 プラットフォーム版 Red Hat Enterprise Linux 4.0 (UD-2 および UD-4) (Sun StorageTek QFS 共有クライアントのみ)
- x64 プラットフォーム版 SuSE Linux Enterprise Server 8 (Service Pack 4) (Sun StorageTek QFS 共有クライアントのみ)
- x64 プラットフォーム版 SuSE Linux Enterprise Server 9 (Service Pack 2)、および SGI Altix Itanium システム (Sun StorageTek QFS 共有クライアントのみ)

- x64 プラットフォーム版 SuSE Linux Enterprise Server 10 (Sun StorageTek QFS 共有クライアントのみ)

注 – これは、SuSE Enterprise Linux 8 をサポートする最後のバージョンの Sun StorageTek QFS です。次のバージョンのソフトウェアでは、SuSE Enterprise Linux 8 はサポートされません。

共有の Sun StorageTek QFS Solaris クライアントとは異なり、Linux クライアントはクライアントのみの動作に制限されます。このクライアントを潜在的なメタデータサーバーとして構成することはできません。Linux クライアントは Sun StorageTek SAM ソフトウェアとの対話をサポートしますが、stage、archive、release、および samu などの Sun StorageTek SAM コマンドはサポートしません。Sun StorageTek QFS ファイルシステム機能のみを備えています。

Sun StorageTek QFS Linux クライアントソフトウェアの詳細については、『Sun StorageTek QFS Linux クライアントガイド』を参照してください。

追加のファイルシステム機能

Sun StorageTek QFS ファイルシステムでは、次の追加機能もサポートされています。

- **ファイル領域の事前割り当て** – setfa(1) コマンドを使用して、逐次読み取り書き込みを高速に行えるように連続したディスク領域を事前割り当てできます。
- **アプリケーションプログラミングインタフェース (API) ルーチン** – API ルーチンを使用すると、連続したディスク領域の事前割り当てや特定のストライプ化グループへのアクセスなど、プログラムによってさまざまな特殊機能を実行できます。これらのルーチンの詳細については、intro_libsam(3) のマニュアルページを参照してください。
- **調整可能なディスク割り当て単位 (DAU)** – DAU はオンライン記憶領域の基本単位です。Sun StorageTek QFS ファイルシステムには、調整可能な DAU が組み込まれています。これは、物理ディスクストレージ装置を備えたファイルシステムを調整したり、読み取り/変更/書き込みの操作で発生するシステムオーバーヘッドを解消したりする場合に役に立ちます。DAU サイズは、4K バイトの倍数で調整可能です。詳細は、8 ページの「ディスク割り当て単位の指定」を参照してください。
- **複数のストライプ化グループのサポート** – 1 つのファイルシステムで複数の RAID 装置をサポートするため、Sun StorageTek QFS ソフトウェアはストライプ化グループの定義をサポートしています。ストライプ化グループのディスクブロック割り当てを最適化でき、その結果、ディスク上の割り当てマップを更新するオーバーヘッドが軽減されます。ユーザーは、API ルーチンによって、または setfa(1) コマンドを使用して、ファイルをストライプ化グループに割り当てることができます。

- **Sun Cluster システムの相互運用性** - Sun StorageTek QFS ファイルシステムは、ローカルファイルシステムおよび Sun Cluster 環境での高可用ファイルシステムとしてサポートされています。詳細は、123 ページの「Sun Cluster 環境での Sun StorageTek QFS の構成」を参照してください。

設計の基本

Sun StorageTek QFS ファイルシステムは、マルチスレッドの高度なストレージ管理システムです。このソフトウェアの機能を最大限に活用するには、可能であれば必ず複数のファイルシステムを作成してください。

Sun StorageTek QFS ファイルシステムは、ディレクトリ参照時にリニアな検索方法を使用し、ディレクトリの最初から最後まで検索します。ディレクトリ内のファイル数が増加するにつれ、ディレクトリ全体の検索時間も長くなります。ディレクトリに数千個のファイルがあると、検索時間が極度に長くなることがあります。検索時間は、ファイルシステムを復元する場合にも長くなります。パフォーマンスを向上させ、ファイルシステムのダンプや復元の時間を短縮するには、1 つのディレクトリ内のファイル数を 10,000 個未満にしてください。

ディレクトリ名参照キャッシュ (DNLC) 機能は、ファイルシステムのパフォーマンスを向上させます。このキャッシュには、パスが短い (30 文字以下) ファイルのディレクトリ参照情報が格納されるため、ディレクトリ参照を急いで行う必要がなくなります。DNLC 機能は、Solaris OS 9 以降のすべてのバージョンで使用できます。

次の項目では、ファイルシステムの設計に影響するいくつかの追加機能について説明しています。

- 7 ページの「i ノードファイルとファイルの特徴」
- 8 ページの「ディスク割り当て単位の指定」

i ノードファイルとファイルの特徴

ファイルシステムに格納されるファイルの種類は、ファイルシステムの設計に影響します。i ノードとは、ファイルまたはディレクトリの特徴を示す情報の 512 バイトのブロックです。この情報は、ファイルシステム内で動的に割り当てられます。

i ノードは、ファイルシステムのマウントポイントの下にある `.inodes` ファイルに格納されます。Sun StorageTek QFS の `.inodes` ファイルは、ファイルデータ装置とは別のメタデータ装置に置くことができます。

標準の Solaris OS の i ノードと同じく、Sun StorageTek QFS ファイルシステムの i ノードには、ファイルの POSIX 標準 i ノード時刻として、ファイルアクセス時刻、ファイル変更時刻、i ノードが変更された時刻が含まれています。Sun StorageTek

QFS ファイルシステムの i ノードには、それ以外にも、表 1-2 に示される時刻が含まれています。

表 1-2 .inode ファイルの内容

時刻	内容
access	ファイルが最後にアクセスされた時刻。POSIX 標準。
modification	ファイルが最後に変更された時刻。POSIX 標準。
changed	i ノード情報が最後に変更された時刻。POSIX 標準。
attributes	Sun StorageTek QFS ファイルシステム固有の属性が最後に変更された時刻。Sun による拡張機能。
creation	ファイルが作成された時刻。Sun による拡張機能。
residence	ファイルのオフラインとオンラインが切り換えられた時刻。Sun による拡張機能。

注 – WORM-FS (書き込み 1 回、読み取り複数回) パッケージがインストールされている場合、i ノードには**保存終了**日付も含まれています。詳細は、194 ページの「WORM-FS ファイルシステムの構成」を参照してください。

i ノードファイル情報の参照についての詳細は、51 ページの「ファイルとファイル属性の表示」を参照してください。

ディスク割り当て単位の指定

ディスク領域は、ディスク割り当て単位 (DAU) と呼ばれるオンラインディスク記憶領域の基本単位で割り当てられます。セクター、トラック、シリンダが物理ディスクジオメトリを表すのに対し、DAU はファイルシステムジオメトリを表します。適切な DAU サイズとストライプサイズを選択すると、パフォーマンスが向上し、磁気ディスクの使用率が最適化されます。DAU 設定は、ファイルが割り当てられるときに使用される連続領域の最小容量になります。

次の項では、DAU 設定とストライプ幅の構成方法について説明します。

DAU 設定とファイルシステムジオメトリ

Sun StorageTek QFS ファイルシステムは調整可能な DAU を使用します。この構成可能な DAU を使用することで、ファイルシステムを物理的なディスクストレージに合わせて調整できます。この機能によって、読み取り/変更/書き込みの操作で発生するシステムオーバーヘッドが最小限になるので、非常に大きなファイルを操作する

アプリケーションには特に有効です。読み取り/変更/書き込み操作の制御方法については、223 ページの「大容量ファイル転送パフォーマンスの向上」を参照してください。

各ファイルシステムは、マウントされてサーバー上でアクティブになっている複数のファイルシステムの 1 つである場合でも、それに固有の DAU 設定を使用できます。使用可能な DAU 設定は、使用しているファイルシステムのタイプによって異なります。DAU 設定は、ファイルシステムを作成するときに `sammkfs(1M)` コマンドによって決まります。動的に変更することはできません。

DAU 設定は、マスター構成 (`mcf`) ファイルで指定された装置およびファイルシステムの定義と組み合わせられます。`mcf` ファイルの詳細については、31 ページの「ファイルシステムの構成」を参照してください。

ms および maSun StorageTek SAM ファイルシステム

使用できるファイル割り当て方式として、ms ファイルシステムタイプと ma ファイルシステムタイプがあります。

単一のパーティションにインストールされる、単純な Sun StorageTek QFS Sun StorageTek SAM ファイルシステムの場合、ファイルシステムは `mcf` ファイル内の装置タイプの値 `ms` で定義されます。ms ファイルシステムでは、許容される装置タイプはタイプ `md` のみであり、メタデータとファイルデータはどちらも `md` 装置に書き込まれます。デフォルトでは、`md` 装置の DAU は 64K バイトです。

複数のパーティションにインストールされる、さらに複雑な Sun StorageTek QFS ファイルシステムは、`mcf` ファイル内の装置タイプの値 `ma` で定義されます。ma ファイルシステムでは、メタデータは `mm` 装置に書き込まれ、データは `md`、`mr`、または `gXXX` 装置に書き込むことができます。

ma ファイルシステムでは、次の装置を混在させることができます。

- mm 装置と mr 装置
- mm 装置と gXXX 装置
- mm 装置、mr 装置、および gXXX 装置
- mm 装置と md 装置

これらの装置タイプの詳細については、表 2-2 を参照してください。

デュアルおよびシングル割り当て方式

md 装置および mm 装置はデュアル割り当て方式を使用し、次のようになります。

- md データ装置では、小さな割り当ては 4K バイト、大きな割り当ては DAU になります。DAU のデフォルトは 64K バイトです。sammkfs(1M) コマンドの -a *allocation-unit* オプションを使用してファイルシステムを初期化するとき、このデフォルト値を無効にすることができます。DAU サイズには、16K、32K、または 64K バイトを指定できます。
- md 装置にファイルが作成されると、システムはファイルの最初の 8 つのアドレスを小さな割り当ての中に割り当てます。さらに領域が必要な場合、ファイルシステムは 1 つまたは複数の大きな割り当て (DAU) を使用してファイルを拡張します。その結果、大容量ファイルの入出力のパフォーマンスが向上する一方で、小さいファイルが多数できることから生じるディスクの断片化が最小限に抑えられます。

注 - ms ファイルシステムを使用するときは、ストライプの幅をゼロより大きく設定し、メタデータ情報がディスクをまたがってストライプ化されるようにしてください。ただし、ストライプの幅と DAU サイズを設定する前に、12 ページの「データディスクでのストライプ幅」を読んで理解しておくようにします。

- mm メタデータ装置では、小さな割り当ては 4K バイト、大きな割り当ては 16K バイトになります。デュアル割り当て方式は、ファイルシステムがメタデータをディスクに書き込むときの効率が上がり、ディスク断片化を最小限に抑えるのに有効です。

ファイルシステムに格納されるファイルデータの種類によっては、DAU サイズを大きくすることで、ファイルシステムのパフォーマンスが大幅に向上する場合があります。ファイルシステムのパフォーマンス調整についての詳細は、第 7 章 187 ページの「高度な機能」を参照してください。

ma Sun StorageTek QFS ファイルシステムのみが、シングル割り当て方式を使用する装置を使用できます。これらのファイルシステムは、次のように個別のメタデータ装置とデータデバイスで構成されます。

- メタデータ装置は、装置タイプ mm としてのみ定義可能です。
- データデバイスは、装置タイプ md、mr、または gXXX として定義できます。md 装置では、DAU のサイズが 16K バイト、32K バイト、または 64K バイトに制限されます。

mr および gXXX 装置は、シングル割り当て方式です。1 つのファイルシステム内で mr 装置と gXXX 装置が混在できますが、1 つのファイルシステム内で md 装置は、mr 装置または gXXX 装置と混在できません。

mr および gXXX データ装置を使用する Sun StorageTek QFS ファイルシステムの DAU サイズは構成可能です。データ装置で使用できる DAU サイズは、mcf ファイルで各データ装置に割り当てられた装置タイプによって異なります。表 1-3 に、これらの DAU サイズを示します。

表 1-3 Sun StorageTek QFS 装置タイプの値および DAU のサイズ

装置タイプ	DAU のサイズ
mr または gXXX	8K バイト単位でデフォルトのサイズを調整して、さまざまな DAU のサイズを指定できます。DAU サイズは、16K バイト ~ 65,528K バイト (64M バイト) の範囲で指定できます。デフォルトの DAU サイズは、mr の場合 64K バイト、gXXX の場合 256K バイトです。
md	この装置タイプでは、デュアル割り当て方式が使用されます。DAU のサイズは、16K、32K、または 64K バイトに構成できます。DAU のデフォルトサイズは 64K バイトです。 ma ファイルシステムの md 装置はデータの格納専用であり、メタデータの格納には使用されません。ms ファイルシステムの md 装置は、ファイルデータとメタデータの両方の格納に使用されます。

注 - このソフトウェアの version 3.5 を使用してファイルシステムを作成したか、またはこのソフトウェアの version 4 で sammkfs 互換性モードフラグを使用してファイルシステムを作成した場合、version 1 のスーパーブロックを使用している可能性があります。version 1 のスーパーブロックでは、mm 装置はデュアル割り当て方式を使用せず、mm 装置の割り当ては 16K バイトです。Sun StorageTek QFS ファイルシステムの md 装置を定義できるのは、version 2 のスーパーブロックのみです。version 1 のスーパーブロックを使用しているかどうか調べるには、samfsinfo(1M) コマンドを使用します。

データ整合

データ整合とは、RAID コントローラの割り当て単位とファイルシステムの割り当て単位を一致させることです。Sun StorageTek QFS ファイルシステムの最適な整合式は次のとおりです。

割り当て単位 = RAID ストライプ幅 × データディスク数

たとえば、RAID-5 ユニットに合計 9 個のディスクがあり、そのうち 1 つはパリティディスクで、データディスクの数は 8 個であるとしします。RAID のストライプ幅が 64K バイトの場合、最適な割り当て単位は 64×8 で 512K バイトです。

データファイルは、同じファイルシステム内にある各ストライプ化グループ (gXXX) またはデータディスク (mr または md) を通してストライプ化またはラウンドロビン処理されたものとして割り当てられます。

整合が取れていないと、読み取り/変更/書き込み操作が発生するためにパフォーマンスが損なわれます。

データディスクでのストライプ幅

ストライプ幅のデフォルト値は、Sun StorageTek QFS ms ファイルシステムと ma ファイルシステムとの間で異なります。ストライプ幅は、mount(1M) コマンドの -o stripe=*n* オプションによって指定されます。ストライプ幅を 0 に設定すると、ラウンドロビン式割り当てが使用されます。

次の項では、各種のファイルシステムでのストライプ幅について説明します。

ms ファイルシステムでのストライプ幅はマウント時に設定されます。表 1-4 に、デフォルトのストライプ幅を示します。

表 1-4 ms ファイルシステムでのデフォルトのストライプ幅

DAU	デフォルトのストライプ幅	ディスクに書き込まれるデータ量
16K バイト	8 DAU	128K バイト
32K バイト	4 DAU	128K バイト
64K バイト (デフォルト)	2 DAU	128K バイト

たとえば、sammkfs(1M) をデフォルト設定で実行すると、デフォルトの大きな DAU は 64K バイトになります。mount(1M) コマンドを実行するときにストライプ幅を指定しないと、デフォルトが使用され、ストライプ幅はマウント時に 2 に設定されます。

注 – ms ファイルシステムでは、メタデータ情報がディスクをまたがってストライプ化されるよう、ストライプ幅をゼロより大きく設定することが重要です。

表 1-4 の 1 列目の数値に 2 列目の数値を乗じると、結果は 128K バイトになることに注意してください。ディスクに書き込まれるデータ量が 128K バイト以上になると、Sun StorageTek QFS ファイルシステムはもっとも効率的に動作します。

ストライプ化グループを使用していない ma ファイルシステムでのストライプ幅

Sun StorageTek QFS ma ファイルシステムでは、マウント時に設定されるストライプ幅は、ストライプ化グループが構成されているかどうかによって異なります。ストライプ化グループとは、ストライプ化された装置のグループとしての集まりです。スト

ライプ化グループの詳細については、15 ページの「ファイル割り当て方式」を参照してください。この項では、ストライプ化グループなしで構成された Sun StorageTek QFS ファイルシステムのストライプ幅について説明します。

ストライプ化グループが構成されていない場合、ma ファイルシステムでの DAU とストライプ幅の関係は、ms ファイルシステムの場合の関係と類似したものになります。違っている点は、64K バイトを超える DAU が可能なことと、DAU を 8K バイトのブロック単位で構成できることです。DAU の最大サイズは 65,528K バイトです。

デフォルトでは、ストライプ幅が指定されていない場合、ディスクに書き込まれるデータ量は 128K バイトまたはその前後になります。書込み操作で 1 つの入出力要求あたり最低でも 1 つのストライプ全体を書き込む場合、Sun StorageTek QFS ファイルシステムの効率は最高になります。表 1-5 にデフォルトのストライプ幅を示します。

表 1-5 デフォルトのストライプ幅

DAU	デフォルトのストライプ幅	ディスクに書き込まれるデータ量
16K バイト	8 DAU	128K バイト
24K バイト	5 DAU	120K バイト
32K バイト	4 DAU	128K バイト
40K バイト	3 DAU	120K バイト
48K バイト	2 DAU	96K バイト
56K バイト	2 DAU	112K バイト
64K バイト (デフォルト)	2 DAU	128K バイト
72K バイト	1 DAU	72K バイト
128K バイト	1 DAU	128K バイト
128K バイトより大	1 DAU	DAU サイズ

ストライプ化グループを使用する ma ファイルシステムでのストライプ幅

ストライプ化グループが Sun StorageTek QFS ファイルシステムに構成されている場合、割り当てられる領域の最小容量は、DAU のサイズにストライプ化グループ内のデバイス数を乗じたものになります。ストライプ化グループを使用すると、割り当ての容量は非常に大きくなる可能性があります。

ストライプ化グループを使用すると、データは複数のディスク装置に一度に書き込まれ、それらのディスク装置は1台の装置のように作動します。ストライプ化グループでの割り当ては、DAUのサイズにストライプ化グループ内の要素数を乗じたものになります。

-o stripe=*n* マウントオプションによって、1つのストライプ化グループでの割り当て数が決まります。その数になると、別のストライプ化グループでの割り当てが行われます。ファイルシステムが -o stripe=0 を使用してマウントされた場合、割り当ては、常に1つのストライプ化グループを対象とします。

デフォルトの設定は -o stripe=0 で、ラウンドロビン割り当て方式を指定します。最小の設定は -o stripe=0 (ストライプ化無効) で、最大の設定は -o stripe=255 です。不一致のストライプ化グループが存在する場合は、システムが -o stripe=0 を設定します。この場合、ファイルが常駐できるストライプ化グループは1つだけです。

割り当て方式についての詳細は、15 ページの「ファイル割り当て方式」を参照してください。

メタデータディスクでのストライプ幅

mount_samfs(1M) コマンドで -o mm_stripe=*n* オプションを使用することで、メタデータディスク上のメタデータ情報をストライプ化できます。デフォルトのストライプ幅は -o mm_stripe=1 です。ファイルシステムが 16K バイトの DAU を1つメタデータディスクに書き込んでから、次のメタデータディスクに切り換わるように指定されます。メタデータディスクでは、小さな 4K バイトの DAU が使用されます。

デフォルトでは、複数のメタデータ装置がある場合、メタデータは、mount(1M) コマンドの -o mm_stripe=*n* オプションの指定に従って割り当てられます。最小の設定は -o mm_stripe=0 (ストライプ化無効) で、最大の設定は -o mm_stripe=255 です。

ファイル割り当て方式

Sun StorageTek QFS ソフトウェアでは、ラウンドロビン割り当て方式とストライプ化割り当て方式の両方を指定できます。表 1-6 に、デフォルトで使用されるファイル割り当て方式を示します。

表 1-6 デフォルトの割り当て方式

ファイルシステム	メタデータ	ファイルデータ
Sun StorageTek QFS	ストライプ化	ストライプ化
Sun StorageTek QFS (ストライプ化グループ)	ストライプ化	ラウンドロビン式
Sun StorageTek QFS 共有ファイルシステム	ストライプ化	ラウンドロビン式

この節の以降の項では、割り当てについて詳細に説明します。

メタデータ割り当て

メタデータ割り当ては、使用するファイルシステムのタイプによって異なります。

- ms ファイルシステムでは、メタデータは md 装置に割り当てられます。
- ma ファイルシステムでは、メタデータは mm 装置に割り当てられます。mm 装置にはファイルデータは割り当てられません。

i ノードは長さ 512 バイトです。ディレクトリの初期の長さは 4K バイトです。表 1-7 に、システムによるメタデータの割り当てを示します。

表 1-7 メタデータ割り当て

メタデータタイプ	ma ファイルシステム用の割り当ての増分値	ms ファイルシステム用の割り当ての増分値
i ノード (.inodes ファイル)	16K バイト DAU	16、32、または 64K バイト (DAU)
間接ブロック	16K バイト DAU	16、32、または 64K バイト (DAU)
ディレクトリ	4K バイトブロックおよび 16K バイト DAU	4K バイト、最大で合計 32K バイト、それを超えた場合は DAU サイズ

ラウンドロビン式割り当て

ラウンドロビン式割り当て方式では、ファミリーセット内の連続している各装置に、一度に1つのデータファイルが書き込まれます。ラウンドロビン式割り当てでは、複数データストリームの場合に役立ちます。このような環境では、全体のパフォーマンスがストライプ化のパフォーマンスを上回るためです。

ラウンドロビンディスク割り当てを使用すると、1つのファイルを1つの論理ディスクに書き込むことができます。その次のファイルはその次の論理ディスクに書き込まれ、それ以後も同様です。書き込まれたファイル数がファミリーセットに定義された装置数と同じになると、ファイルシステムは、選択されている最初の装置から再度開始します。ファイルが物理デバイスのサイズを超えると、ファイルの前半が最初の装置に書き込まれ、使用可能な記憶領域がある次の装置にファイルの残りが書き込まれます。書き込むファイルのサイズによって、入出力サイズが決まります。

ラウンドロビン式割り当ては、`/etc/fstab` ファイルに `stripe=0` と入力することで明示的に指定できます。

次の図は、`ms` および `ma` ファイルシステムでのラウンドロビン式割り当てを示しています。これらの図では、ファイル1がディスク1、ファイル2がディスク2、ファイル3がディスク3のように対応して書き込まれます。ファイル6が作成されるとディスク1に書き込まれ、ラウンドロビン式割り当て方式が再開します。

ms ファイルシステム

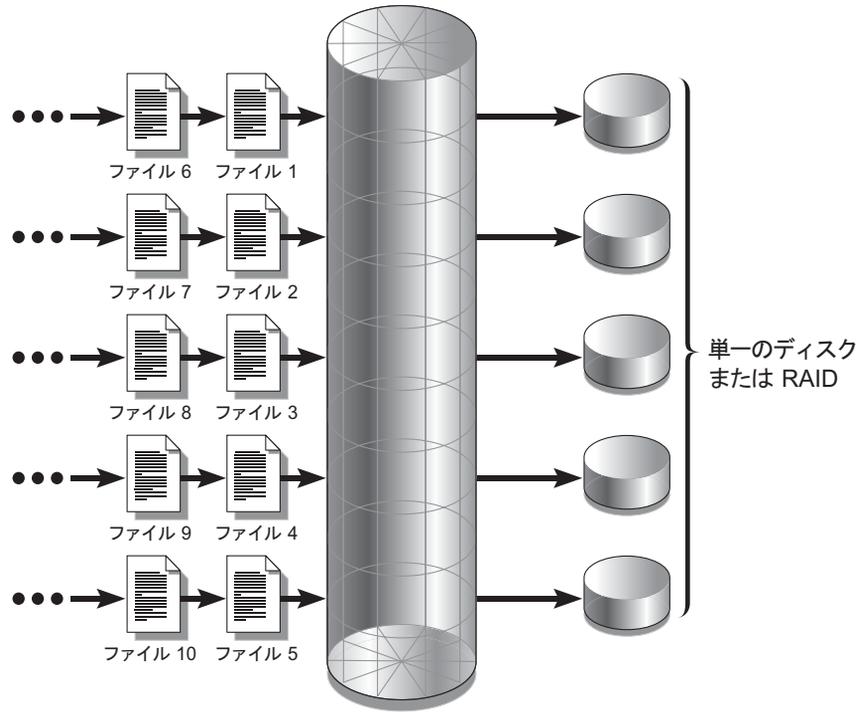


図 1-1 5 台の装置を使用した、ms ファイルシステムでのラウンドロビン割り当て

ma ファイルシステム

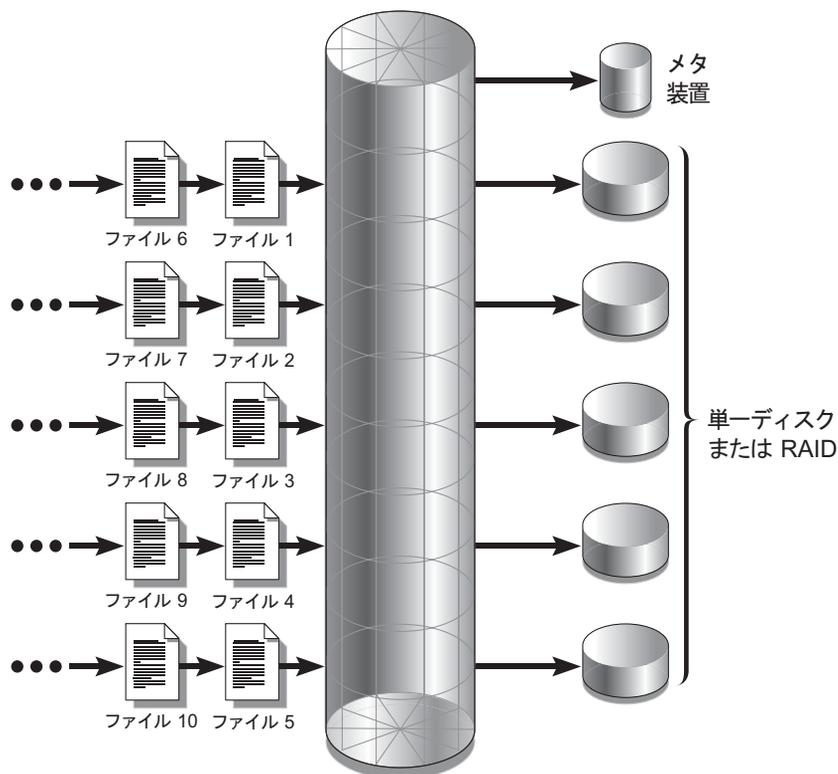


図 1-2 5 台の装置を使用した、ma ファイルシステムでのラウンドロビン割り当て

ストライプ化割り当て

デフォルトでは、Sun StorageTek QFS ファイルシステムはストライプ化割り当て方式を使用して、ファイルシステムのファミリーセット内のすべてのデバイスにデータを分散します。ストライプ化とは、複数の装置にインタレース形式で並行してファイルを書き込む方式です。

ストライプ化が使用されるのは、1つのファイルに対するパフォーマンスに、すべての装置を合計したパフォーマンスが要求される場合です。ストライプ化装置を使用するファイルシステムは、逐次形式ではなくインタレース形式でブロックをアドレス指定します。ストライプ化によって、複数の入出力ストリームが複数のディスクに1つのファイルを同時に書き込むことができるので、通常はパフォーマンスが向上します。DAU とストライプ幅によって、入出力転送のサイズが決まります。

ストライプ化を使用するファイルシステムでは、ファイル 1 がディスク 1、ディスク 2、ディスク 3、ディスク 4、ディスク 5 に書き込まれます。ファイル 2 もディスク 1 ~ ディスク 5 に書き込まれます。DAU にストライプ幅を乗じた値によって、各ディスクに書き込まれるデータ容量 (ブロック単位) が決まります。

Sun StorageTek QFS ファイルシステムが md 装置へのファイルの書き込みを開始するとき、ファイルを 4K バイトの小さな DAU に収容しようとします。ファイルが、割り当て済みの最初の 8 個の小さな DAU (32K バイト) に収まらない場合、ファイルシステムは、そのファイルの残りを 1 つ以上の大きな DAU に書き込みます。

Sun StorageTek QFS ファイルシステムが mr 装置にファイルを書き込むとき、最初に 1 つの DAU に、次にほかの DAU に、次にまたほかの DAU に、という具合に書き込んでいきます。mr 装置の DAU サイズは 1 つだけです。

複数のアクティブなファイルがある場合に、ラウンドロビン割り当てではなくストライプ化割り当てを使用すると、ディスクヘッドの移動が大幅に増加します。入出力が複数ファイルに同時に発生する場合は、ラウンドロビン式割り当てを使用します。

次の図は、ストライプ化割り当てを使用した ms および ma ファイルシステムを示しています。これらの図では、ファイルのうち、 $DAU \times$ ストライプ幅で計算されるバイト数のデータが、ディスク 1 に書き込まれます。ファイルのうち、次の $DAU \times$ ストライプ幅で計算されるバイト数のデータが、ディスク 2 に書き込まれます。以降同様に続きます。ストライプの順序は、ファイルに対して FIFO (先入れ先出し) になります。ストライプ化によって、入出力の負荷がすべてのディスクに分散されます。

ms ファイルシステム

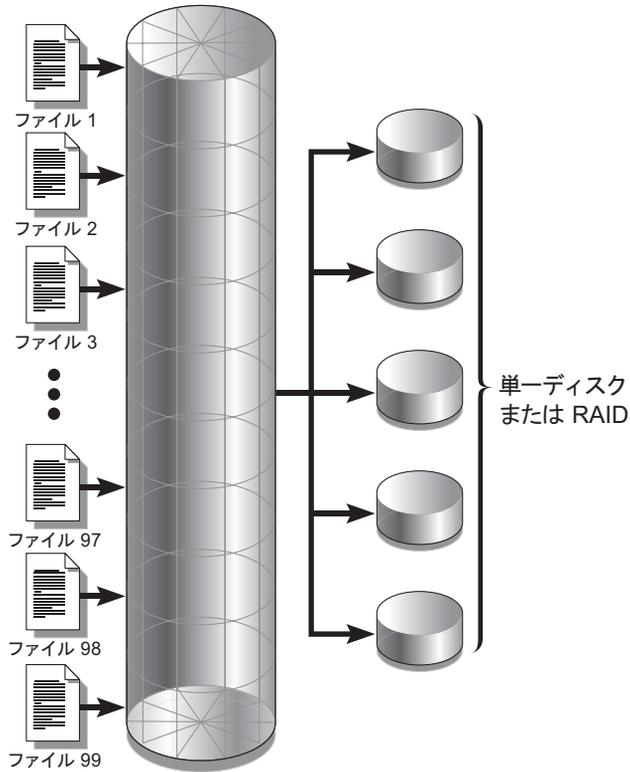


図 1-3 5 台の装置を使用した、ms ファイルシステムでのストライプ化

ma ファイルシステム

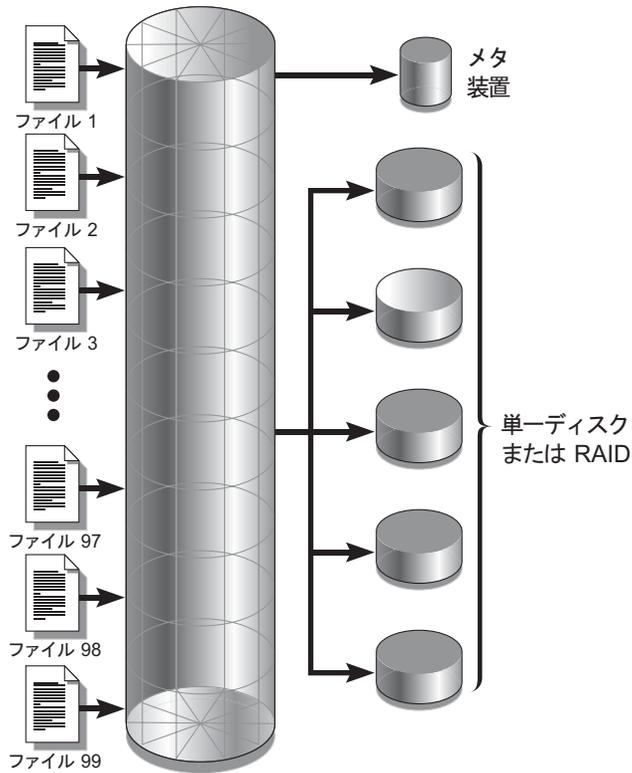


図 1-4 5 台の装置を使用した、ma ファイルシステムでのストライプ化

ストライプ化グループ

ストライプ化グループとは、非常に大規模な入出力要件とテラバイト単位のディスクキャッシュを持つファイルシステムのために設計された、Sun StorageTek QFS の割り当て方式です。ストライプ化グループを使用すると、複数の物理ディスクを考慮した装置タイプ値を指定できます。複数のストライプ化グループの装置タイプのエントリを、単一の Sun StorageTek QFS ファイルシステムで構成できます。ストライプ化グループを使用すると、非常に大規模な RAID 構成の場合に、ビットマップ領域とシステムの更新時間を節約できます。

ストライプ化グループは、Sun StorageTek QFS ファイルシステム内の装置のコレクションです。mcF ファイルで gXXX 装置として定義されていれば、ストライプ化グループを使用して、複数の装置との間で 1 つのファイルの書き込みや読み取りを行うことができます。1 ファイルシステム内に最大 128 のストライプ化グループを指定できます。

図 1-5 は、ストライプ化グループとラウンドロビン式割り当てを使用する Sun StorageTek QFS ma ファイルシステムを示しています。図 1-5 で、qfs1 ファイルシステムに書き込まれるファイルはグループ g0、g1、および g2 の間でラウンドロビン割り当てされます。各グループは、2 つの物理 RAID デバイスで構成されています。

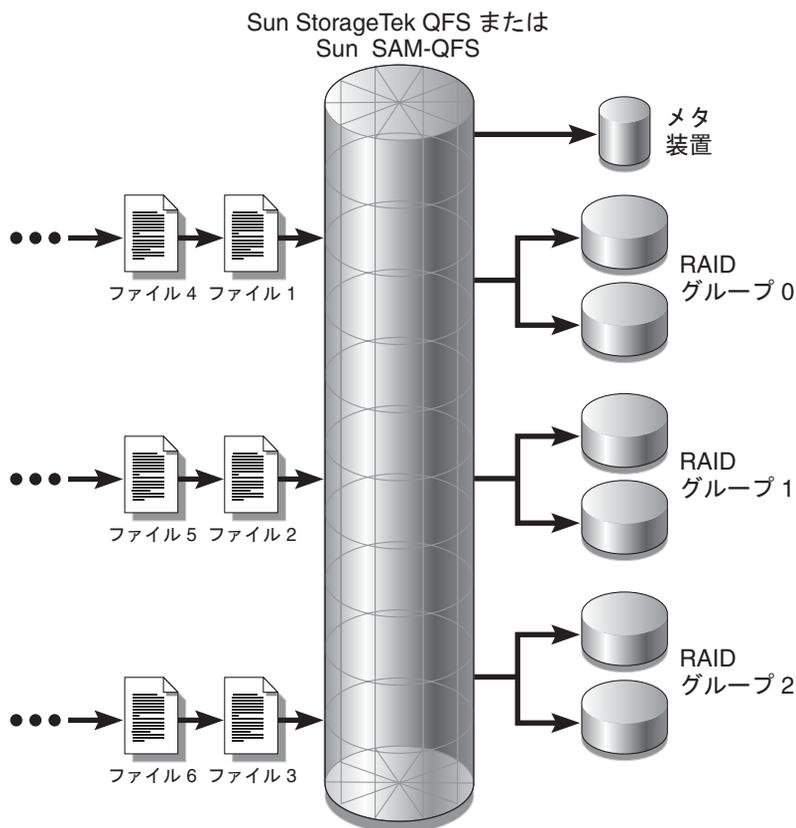


図 1-5 Sun StorageTek QFS のラウンドロビン式ストライプ化グループ

図 1-5 の構成で、`/etc/vfstab` のマウントポイントオプションは `stripe=0` に設定されています。コード例 1-1 に、これらのストライプ化グループを宣言した `mcf` ファイルを示します。

コード例 1-1 ストライプ化グループを示す `mcf` ファイルの例

# Equipment	Eq	Eq	Fam	Dev	Additional
# Identifier	Ord	Type	Set	State	Parameters
#					
qfs1	10	ma	qfs1		
/dev/dsk/c0t1d0s6	11	mm	qfs1	-	
/dev/dsk/c1t1d0s2	12	g0	qfs1	-	
/dev/dsk/c2t1d0s2	13	g0	qfs1	-	
/dev/dsk/c3t1d0s2	14	g1	qfs1	-	
/dev/dsk/c4t1d0s2	15	g1	qfs1	-	
/dev/dsk/c5t1d0s2	16	g2	qfs1	-	
/dev/dsk/c6t1d0s2	17	g2	qfs1	-	

図 1-6 は、ストライプ化グループとストライプ化割り当てを使用する Sun StorageTek QFS `ma` ファイルシステムを示しています。qfs1 ファイルシステムに書き込まれるファイルはグループ `g0`、`g1`、および `g2` にストライプ化されます。各グループには、4 つの物理 RAID デバイスがあります。`/etc/vfstab` のマウントポイントオプションは、`stripe=1` 以上に設定されています。

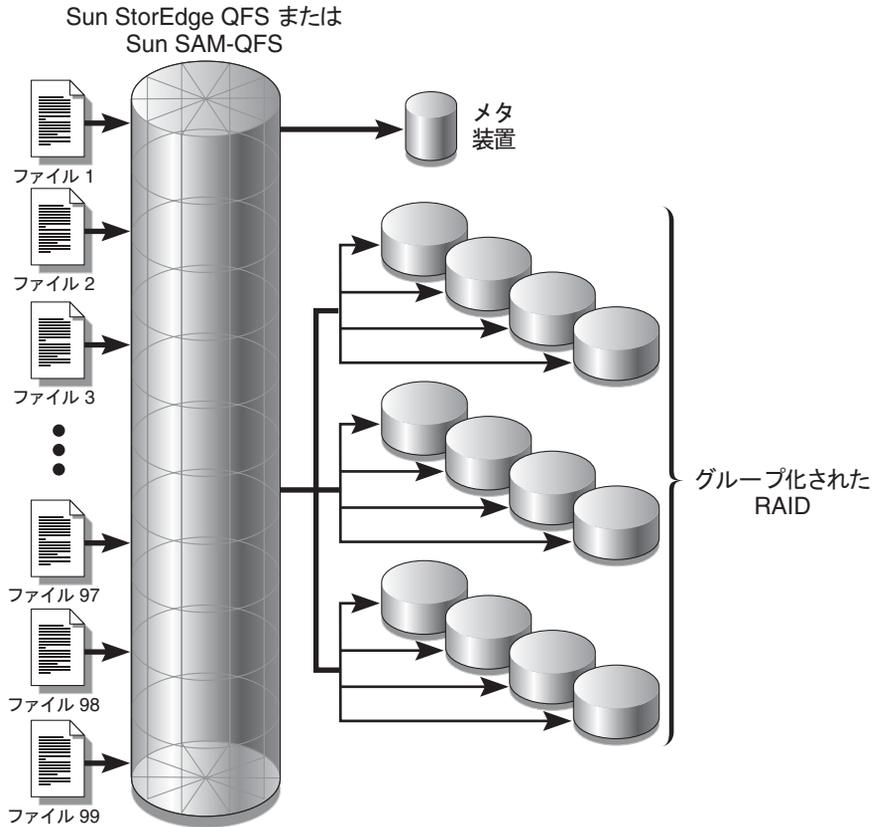


図 1-6 Sun StorageTek QFS のストライプ化グループ割り当て

不一致のストライプ化グループ

不一致のストライプ化グループ、つまり各グループ内の装置の数が異なっている複数のストライプ化グループを使用してファイルシステムを構成することは可能です。

Sun StorageTek QFS ファイルシステムは不一致のストライプ化グループをサポートしますが、不一致のグループでのストライプ化はサポートしません。不一致のストライプ化グループから構成されるファイルシステムは、ラウンドロビン式ファイルシステムとしてマウントされます。

注 - 不一致のストライプ化グループが含まれているファイルシステムでは、1つのファイルが複数のストライプ化グループにまたがることはできません。そのファイルが格納されたストライプ化グループがいっぱいになると、ファイルを拡張できなくなります。不一致のストライプ化グループがある場合は、`setfa(1)` コマンドの `-g` オプションを使用して、ファイルを目的のグループに格納します。詳細は、`setfa(1)` のマニュアルページを参照してください。

ストライプ化グループをどこまでいっぱいにするかを決定するには、`samu(1M)` オペレータユーティリティーを使用し、`m` 画面にアクセスして外部ストレージの状態を表示します。

次の例に、不一致のストライプ化グループを使用して、さまざまな種類のファイルを格納するためのファイルシステムの設定方法を示します。

不一致のストライプ化グループの例

自分のサイトに映像と音声の両方のデータが格納されたファイルシステムを作成する必要がある場合を想定します。

映像ファイルは、容量が非常に大きく、音声ファイルに比べて高いパフォーマンスを必要とします。ストライプ化グループを使用すると大容量ファイルのパフォーマンスが最大限になるため、映像ファイルは大容量のストライプ化グループのあるファイルシステムに格納することにします。

音声ファイルは、映像ファイルよりも小容量で、必要なパフォーマンスも低くなります。音声ファイルは、小容量のストライプ化グループに格納することにします。このように1つのファイルシステムで、映像ファイルと音声ファイルの両方をサポートできます。

図 1-7 は、必要なファイルシステムを示しています。これは、ストライプ化割り当てで不一致のストライプ化グループを使用する `ma` ファイルシステムです。

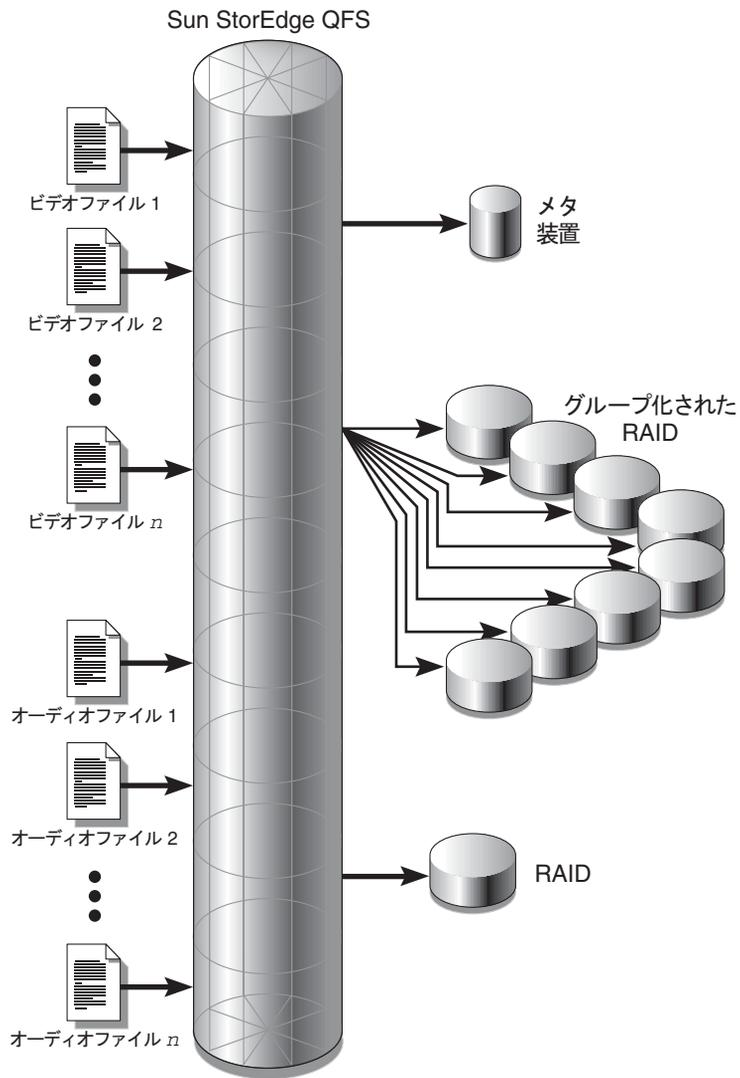


図 1-7 ストライプ化割り当てにおいて不一致のストライプ化グループを使用する Sun StorageTek QFS ファイルシステム

表 1-8 に、このファイルシステム例の特徴を示します。

表 1-8 ファイルシステム例の特徴

特徴	説明
ファイルシステム名	avfs
ストライプ化グループ数	2 つ。映像ファイルのグループが g0。音声ファイルのグループが g1。
ストライプ幅	0
DAU	128K バイト
g0 のディスク数	8
g0 の最大ブロックサイズ	8 ディスク × 128K バイトの DAU = 1024K バイト 1 ブロックの書き込みで書き込まれるデータ容量。各ディスクが 128K バイトのデータを受け取るため、すべてのディスクに同時に書き込まれる容量の合計は 1024K バイト。
g1 のディスク数	1
g1 の最大ブロックサイズ	1 ディスク × 128K バイトの DAU = 128K バイト

次の行を /etc/vfstab ファイルに追加すると、環境によって avfs ファイルシステムが認識されます。

```
avfs - /avfs samfs - no stripe=0
```

/etc/vfstab ファイルでは、stripe=0 はラウンドロビン式ファイルシステムを指定するために使用されます。0 よりも大きな値は不一致のストライプ化グループではサポートされないため、これが使用されます。

コード例 1-2 に、ファイルシステム avfs の mcf ファイルを示します。

コード例 1-2 ファイルシステム avfs の mcf ファイル

```
# Equipment      Eq  Eq  Fam Dev  Additional
# Identifier      Ord Type Set  State Parameters
#
avfs              100 ma  avfs
/dev/dsk/c00t1d0s6 101 mm  avfs -
#
/dev/dsk/c01t0d0s6 102 g0  avfs -
/dev/dsk/c02t0d0s6 103 g0  avfs -
/dev/dsk/c03t0d0s6 104 g0  avfs -
/dev/dsk/c04t0d0s6 105 g0  avfs -
```

コード例 1-2 ファイルシステム avfs の mcf ファイル (続き)

```
/dev/dsk/c05t0d0s6 106 g0 avfs -  
/dev/dsk/c06t0d0s6 107 g0 avfs -  
/dev/dsk/c07t0d0s6 108 g0 avfs -  
/dev/dsk/c08t0d0s6 109 g0 avfs -  
#  
/dev/dsk/c09t1d0s6 110 g1 avfs -
```

このファイルシステムの mcf ファイルの準備ができれば、コード例 1-3 に示す `sammkfs(1M)` および `mount(1M)` コマンドを入力して avfs ファイルシステムを作成およびマウントします。

コード例 1-3 ファイルシステム avfs を作成およびマウントするコマンド

```
# sammkfs -a 128 avfs  
# mount avfs
```

ファイルシステムのマウント後、コード例 1-4 に示すコマンドを使用して 2 種類のファイル用に 2 つのディレクトリを作成します。

コード例 1-4 ファイルシステム avfs にディレクトリを作成するコマンド

```
# cd /avfs  
# mkdir video  
# mkdir audio
```

ディレクトリを作成したら、コード例 1-5 に示す `setfa(1)` コマンドを使用して、大容量ストライプ化グループを映像に、小容量ストライプ化グループを音声に割り当てることができます。これらのディレクトリに作成されるファイルは、属性が継承されるため、対応するストライプ化グループに割り当てられます。

コード例 1-5 ファイル属性を設定するコマンド

```
# setfa -g0 video  
# setfa -g1 audio
```

`sammkfs(1M)` コマンドについては、`sammkfs(1M)` のマニュアルページを参照してください。`mount(1M)` コマンドの詳細については、`mount_samfs(1M)` のマニュアルページを参照してください。`setfa(1)` コマンドの詳細については、`setfa(1)` のマニュアルページを参照してください。

論理ユニット番号 (LUN) による割り当て制御

必要に応じて、`nalloc` コマンドを使用して特定の Sun StorageTek QFS データパーティションへの割り当てを無効にできます。このコマンドは、その装置への以後の割り当てを禁止します。この機能は、現時点ではデータパーティションにのみ使用でき、メタデータパーティションには使用できません。

パーティションへの割り当ては、`alloc` コマンドまたは `on` コマンドのいずれかで再起動できます。

パーティションの割り当て状態 (`allocflag`) は、次に起動したときも持続します。

`nalloc` コマンドおよび `alloc` コマンドは `samu` インタフェースで使用でき、`samu on` コマンドでも割り当てをオンに設定できます。`samu` 画面には、無効にされたパーティションの `nalloc` の状態が表示されます。`samtrace` および `samfsinfo` 出力にも、割り当て状態が含まれます。

`samu` インタフェースの詳細については、261 ページの「`samu(1M)` オペレータユーティリティの使用法」を参照してください。

第2章

ファイル システムの構成

インストールおよび構成操作についての詳細は、『Sun StorageTek QFS インストールおよびアップグレードの手引き』を参照してください。この章では、Sun StorageTek QFS 環境で使用されるファイルシステムの構成について追加の情報を提供します。この章の内容は次のとおりです。

- 31 ページの「File System Manager ソフトウェアの使用」
- 36 ページの「mcf ファイルの機能」
- 43 ページの「ファイルシステムの初期化」
- 44 ページの「構成の例」

File System Manager ソフトウェアの使用

File System Manager ソフトウェアは、中央の位置からネットワーク内の複数のファイルシステムを構成、制御、保護、および監視することができるブラウザインタフェースツールです。この中央の位置にアクセスするには、ネットワーク内の任意のホスト上で Web ブラウザを使用できます。

このソフトウェアの目的は、ファイルシステムに関連したもっとも一般的な作業を、コマンド行インタフェース (CLI) を使用するよりも簡単に実行できるようにすることです。File System Manager ソフトウェアのインストール方法は、『Sun StorageTek QFS インストールおよびアップグレードの手引き』を参照してください。

デフォルトでは、File System Manager はそれがインストールされているサーバーを管理するよう設定されます。File System Manager を使用して、Sun StorageTek QFS ソフトウェアを実行しているほかのサーバーを管理することもできますが、File System Manager からのアクセスを行えるように、最初にそれらの追加サーバーを構成する必要があります。管理対象サーバーを追加する方法は、35 ページの「File System Manager でアクセスするサーバーを追加する」を参照してください。

▼ はじめて File System Manager を起動する

CLI コマンドではなく、File System Manager を起動して使用し、ファイルシステムの管理および構成を行う場合は、次の手順に従います。

1. File System Manager ソフトウェアがインストールされているサーバー、またはそのサーバーにネットワークアクセス可能な任意のコンピュータにログインします。
2. 以前のバージョンのソフトウェアからアップグレードした場合は、Web ブラウザを起動し、ブラウザのキャッシュをクリアします。
3. Web ブラウザから File System Manager ソフトウェアを起動します。

```
https://hostname:6789
```

hostname には、File System Manager ソフトウェアがインストールされているホストの名前を入力します。ホスト名のほかにドメイン名を指定する必要がある場合は、*hostname* を *hostname.domainname* の形式で指定します。この URL は、先頭が `http` ではなく `https` であることに注意してください。

Sun Java™ Web Console のログインページが表示されます。

4. ユーザー名のプロンプトで、`root` または別の有効なユーザー名を入力します。

注 – 旧バージョンから File System Manager ソフトウェアをアップグレードした場合は、`samadmin` ユーザーアカウントも使用できます。File System Manager の全操作に対する完全なアクセス権限を取得するために、ユーザー名に `samadmin`、パスワードに `samadmin` のパスワードを入力してもかまいません。

5. パスワードのプロンプトで、パスワードを入力します。
6. 「ログイン」をクリックします。
7. 「ストレージ」セクションで「File System Manager」をクリックします。

これで、File System Manager にログインしました。

追加の管理者アカウントとユーザーアカウントの作成

File System Manager を初期構成したあと、いつでも追加の管理者アカウントとゲストアカウントを作成できます。ゲストアカウントは、管理ステーションのローカルアカウントです。

File System Manager ソフトウェアを削除した場合は、手動で作成した追加アカウントは削除されません。次のいずれかまたは両方の手順を使用して、手動で追加したアカウントを管理する必要があります。

▼ 追加のアカウントを作成する

1. ブラウザインタフェースの外で、管理ステーションサーバーに `root` でログインします。
2. `useradd` および `passwd` コマンドを使用して、各ユーザーを追加します。
たとえば、アカウント名が `bobsmith` であるユーザーを追加する場合、次のように入力します。

```
# /usr/sbin/useradd bobsmith
```

```
# /usr/bin/passwd bobsmith
```

このようにして追加したユーザーアカウントには、File System Manager 機能に対する読み取り専用表示権限が付与されます。別の権限を付与するには、次の節 33 ページの「権限レベルの割り当て」を参照してください。

権限レベルの割り当て

ユーザーに対して、File System Manager 機能のすべて、または一部のアクセス権を割り当てることができます。次の表に、File System Manager のユーザーに割り当てることができる、5 つの権限レベルを示します。

表 2-1 File System Manager アクセス権レベル

管理権限レベル	説明
<code>com.sun.netstorage.fsmgr.config</code>	ユーザーには無制限のアクセス権があります。
<code>com.sun.netstorage.fsmgr.operator.media</code>	ユーザーは、ライブラリの追加と削除、スタンダードアロンドライブの追加と削除、ボリュームシリアル名 (VSN) の予約、VSN のインポート、VSN の読み込みと読み込み解除、VSN のエクスポートなどが可能です。
<code>com.sun.netstorage.fsmgr.operator.sam.control</code>	ユーザーは、アーカイブ操作を起動、停止、または休止することができます。
<code>com.sun.netstorage.fsmgr.operator.file</code>	ユーザーは、ファイルシステムへの書き込み処理の起動または停止と、ファイルシステムの復元が可能です。
<code>com.sun.netstorage.fsmgr.operator.filesystem</code>	ファイルシステムのマウントやマウント解除、マウントオプションの編集、ファイルシステムのチェック (<code>fsck</code>) が可能です。

すべてまたは一部の権限をユーザーに指定するには、`/etc/user_attr` ファイルに次の行を追加します。

```
account-name:::auths=privilege-level
```

`account-name` はユーザーのアカウント名、`privilege-level` はユーザーに割り当てる権限のレベルです。

たとえば、ユーザー `bobsmith` にすべての権限 (権限レベル `com.sun.netstorage.fsmgr.config`) を割り当てる場合は、`/etc/user_attr` ファイルに次の行を追加します。

```
bobsmith:::auths=com.sun.netstorage.fsmgr.config
```

`bobsmith` に対して、ファイルシステムの書き込みと復元をする権限 (権限レベル `com.sun.netstorage.fsmgr.operator.file`) と、VSN のエクスポートやインポート、VSNの割り当てを行う権限 (権限レベル `com.sun.netstorage.operator.media`) を割り当てる場合は、`/etc/user_attr` ファイルに次の行を追加します。

```
bobsmith:::auths=com.sun.netstorage.fsmgr.operator.file,  
com.sun.netstorage.fsmgr.operator.media
```

複数ユーザーで使用するアカウントの作成

複数のユーザーで使用できる汎用的な File System Manager アカウントを作成することができます。また、その一部のユーザーだけがアクセスできる権限を持つ役割を作成することもできます。

1. `useradd` コマンドと、`passwd` コマンドを使用して、アカウントを追加します。

たとえば、複数のユーザーで使用するアカウント `guest` を追加する場合、次のように入力します。

```
# /usr/sbin/useradd guest  
# /usr/bin/passwd guest
```

2. `roleadd` コマンドと、`passwd` コマンドを使用して、役割を追加します。

`guest` アカウントに、特別な権限を持つ `admin` という役割を作成するためには、次のように入力します。

```
# /usr/sbin/roleadd admin  
# /usr/bin/passwd admin
```

3. /etc/user_attr ファイルで権限レベルを指定します。

admin の役割に対して、ファイルシステムの復元や書き込みを行う権限を割り当てるには、/etc/user_attr ファイルに次の行を追加します。

```
admin:::auths=com.sun.netstorage.fsmgr.operator.file
guest:::type=normal;roles=admin
```

この例では、ユーザーが guest でログインした場合に、File System Manager は、「No Role」または「Admin」のどちらかを選択するようユーザーに促します。ユーザーが「Admin」の役割のパスワードを知っている場合、「Admin」を選択してパスワードを入力すると、ファイルシステムの復元や書き込みをする権限を持ちます。ほかのすべてのユーザーは、必ず「No Role」を選択して、読み取り専用の権限を持つようにします。

同じ権限レベルを持つ複数のユーザーがソフトウェアに同時にログインできるため、あるユーザーの変更を、ほかのユーザーの変更で上書きしてしまう危険性があります。これを防ぐために、どのユーザーが変更できるようにするか、ほかのユーザーにはどのように変更を通知するかについて、方針を決めておくことを推奨します。

▼ File System Manager でアクセスするサーバーを追加する

デフォルトでは、File System Manager はそれがインストールされているサーバーを管理するよう設定されています。File System Manager を使用して、Sun StorageTek QFS ソフトウェアを実行しているほかのサーバーを管理することもできますが、File System Manager からのアクセスを行えるように、最初にそれらの追加サーバーを構成する必要があります。

1. ブラウザインタフェースの外部で、Telnet ユーティリティを使用し、追加するサーバーに接続します。root としてログインします。
2. fsmadm(1M) add コマンドを使用し、管理ステーション (File System Manager ソフトウェアがインストールされているシステム) を、このサーバーをリモート管理できるホストのリストに追加します。

このコマンドでリストに追加したホストのみが、サーバーをリモート管理できます。

例:

```
# fsmadm add management_station.sample.com
```

3. 管理ステーションの追加が成功したことを確認するには、fsmadm(1M) の list コマンドを使用し、出力されたリストにその管理ステーションが含まれていることを確認します。

4. File System Manager のブラウザインタフェースに管理者ユーザーとしてログインします。
5. 「サーバー」 ページから「追加」 をクリックします。
「サーバーの追加」 ウィンドウが表示されます。
6. 「サーバー名かIPアドレス」 フィールドに、新しいサーバーの名前または IP アドレスを入力します。
7. 「了解」 をクリックします。

セッションタイムアウトを設定する

Java Web Console フレームワークのセッションタイムアウトは、デフォルトで 15 分です。File System Manager のインストールプログラムは、このセッションタイムアウトを 60 分に変更します。セッションタイムアウト値を別の値に変更することはできますが、安全性確保のため、60 分以下の値を設定することを推奨します。

セッションタイムアウト値を変更するには、管理ステーションで次のコマンドを入力します。

```
/opt/SUNWfsmgr/bin/fsmgr session <timeout-in-minutes>
```

たとえば、タイムアウト値を 45 分に変更する場合、次のように入力します。

```
/opt/SUNWfsmgr/bin/fsmgr session 45
```

mcf ファイルの機能

/etc/opt/SUNWsamfs/mcf にあるマスター構成ファイル (mcf) は、Sun StorageTek QFS ソフトウェアによって管理または使用されるすべての装置について説明しています。システム構成時にこの ASCII ファイルを作成するときに、各装置の属性を宣言し、各ファイルシステム内の装置をファミリーセットにグループ化します。

mcf ファイルには、RAID およびディスク装置を識別してファイルシステムに編成するために、これらのファイルシステムが必要とする情報が含まれています。また、ファイルシステムに組み込まれる各自動ライブラリまたは装置のエントリも含まれます。mcf ファイルのサンプルは、/opt/SUNWsamfs/examples/mcf にあります。

mcf ファイルは、コード例 2-1 で示されるように、6 つの列つまりフィールドに分割された指定コードの行で構成されています。

コード例 2-1 mcf ファイルのフィールド

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

mcf ファイルにデータを入力するときは、次の規則に従います。

- ファイルのフィールドの間には、空白文字またはタブ文字を入力します。
- mcf ファイルにはコメント行を組み込むことができます。コメント行は先頭にシャープ (#) を付けます。
- 一部のフィールドは省略可能です。オプションのフィールドに含まれる情報に意味がないことを示すには、ハイフン (-) を使用します。

mcf ファイルの書き込みの詳細については、mcf(4) のマニュアルページを参照してください。File System Manager を使用して、mcf ファイルを自動的に作成することもできます。File System Manager のインストールについては、『Sun StorageTek QFS インストールおよびアップグレードの手引き』を参照してください。File System Manager の使用方法については、オンラインヘルプを参照してください。

以降の項では、mcf ファイルの各フィールドについて説明します。

- 37 ページの「Equipment Identifier」フィールド
- 38 ページの「Equipment Ordinal」フィールド
- 38 ページの「Equipment Type」フィールド
- 39 ページの「Family Set」フィールド
- 40 ページの「Device State」フィールド
- 40 ページの「Additional Parameters」フィールド

「Equipment Identifier」フィールド

「Equipment Identifier」フィールドは必須です。「Equipment Identifier」フィールドを使用して、次の種類の情報を指定します。

- ファイルシステム名。このフィールドでファイルシステム名を指定する場合、その名前は「Family Set」フィールド内の名前と同じものにする必要があります。mcf ファイルの後続の行で、ファイルシステムに含まれるすべてのディスクまたは装置を定義する必要があります。mcf ファイルでは、複数のファイルシステムを宣言できます。通常、mcf ファイルの最初のデータ行では最初のファイルシステムを宣言し、後続の行ではそのファイルシステムに組み込む装置を指定します。mcf ファイルでそのほかのファイルシステムを宣言するときは、読みやすいように空白のコメント行を前に付けることもできます。ファイルシステム名の先頭には、英字を使用する必要があり、英字、数字、下線記号 (_) だけが使用できます。

- `nodev` キーワード。キーワード `nodev` は、`mcf` ファイルの常駐するシステムが、Solaris ホスト上の Sun StorageTek QFS 共有システムでクライアントホストとして使用されることを示します。ファイルシステムが Sun Cluster 環境にある場合は、このキーワードを使用しないでください。メタデータサーバーに常駐する 1 つまたは複数のメタデータ装置に対してのみ、このフィールドにこのキーワードを「Equipment Identifier」として指定できます。Sun StorageTek QFS 共有ファイルシステムのメンバーに対する `mcf` ファイルの作成の詳細は、『Sun StorageTek QFS インストールおよびアップグレードの手引き』を参照してください。
- ディスクパーティションまたはスライスの説明。このフィールドに `/dev/` エントリがある場合は、ディスクパーティションまたはスライスが指定されます。
- 自動ライブラリまたは光磁気ディスクドライブの説明。`/dev/samst` エントリは、自動ライブラリまたは光ドライブを特定します。ネットワーク接続の自動ライブラリを構成する場合、詳細については『Sun StorageTek Storage Archive Manager (SAM) インストールおよびアップグレードの手引き』を参照してください。
- テープドライブの説明。このエントリは、次の 2 つの形式のいずれかにすることができます。
 - `/dev/rmt` エントリ。
 - `/dev/rmt` リンクと同じファイルを指示するシンボリックリンクへのパス。この方法でテープドライブを指定する場合は、必ず、ファイルシステムをマウントする前にリンクを作成してください。

「Equipment Identifier」フィールドにファイルシステムの名前を指定する場合、その名前は 31 文字に制限されます。そのほかのすべての内容では、このフィールドは 127 文字に制限されます。

「Equipment Ordinal」フィールド

`mcf` ファイルの各行の、「Equipment Ordinal (*eq*)」フィールドには、定義しているファイルシステム構成要素または装置の数値識別子を指定する必要があります。1 ~ 65534 の一意の整数を指定します。このフィールドは必須です。

「Equipment Type」フィールド

「Equipment Type」フィールドには、2 文字、3 文字、または 4 文字のコードを入力します。このフィールドは必須です。

Sun StorageTek QFS または SAM-QFS ファイルシステムの「Equipment Type」フィールドには、表 2-2 に示されているいずれかの値を指定できます。

表 2-2 Sun StorageTek QFS または SAM-QFS の「Equipment Type」フィールド

「Equipment Type」 フィールドの内容	意味
ma	メタデータを別の装置 (mm 装置) に格納する Sun StorageTek QFS ファイルシステムを定義します。
ms	データとメタデータを同じ装置に格納する Sun StorageTek QFS ファイルシステムを定義します。
md	ファイルデータを格納するため、ストライプ装置またはラウンドロビン式の装置を定義。
mm	i ノードおよびそのほかのデータ以外の情報を格納するため、メタデータ装置を定義。複数のメタデータ装置を指定できます。Sun StorageTek QFS ma ファイルシステム上のメタデータ (i ノード、ディレクトリ、割り当てマップなど) は、メタデータ装置に格納され、ファイルデータ装置とは分離されます。デフォルトでは、複数のメタデータ装置がある場合、メタデータはラウンドロビン式割り当てで割り当てられます。
mr	ラウンドロビン式の装置またはストライプ化データ装置を定義します。
gXXX	ストライプ化グループデータ装置を定義します。ストライプ化グループは、先頭の g のあとに数値を付けて指定。この番号は 0 ~ 127 の整数である必要があります。たとえば、g12 です。 ストライプ化グループのすべてのメンバーは、タイプとサイズが同じである必要があります。1 つのファイルシステム内の異なるストライプ化グループは、同じ数のメンバーを持つ必要はありません。1 つのファイルシステムに、md、mr、および gXXX 装置を混合させることはできません。 データは、グループ間でストライプ化 (すべてのグループの装置数が同じ場合) またはラウンドロビン処理できます。デフォルトはラウンドロビンです。

ファイルシステムの装置タイプに加え、自動ライブラリやその他の装置を指定するには、ほかのコードが使用されます。特定の装置タイプの詳細は、mcf(4) のマニュアルページを参照してください。

「Family Set」フィールド

「Family Set」フィールドには、装置のグループの名前を指定します。このフィールドは必須です。

ファミリーセット名は、先頭に英字を使用する必要があり、英字、数字、下線記号 (_) だけを使用できます。

ファイルシステム内のディスク装置を定義する行では、すべて同じファミリーセット名が指定される必要があります。このソフトウェアは、ファミリーセット名を使用して、装置をファイルシステムとしてグループ化します。 `sammkfs(1M)` コマンドが実行されると、ファミリーセット名がファイルシステム内のすべての装置に物理的に記録されます。 `samfsck(1M)` コマンドで `-F` オプションと `-R` オプションを同時に使用することで、この名前を変更できます。 `sammkfs(1M)` コマンドについては、 `sammkfs(1M)` のマニュアルページを参照してください。 `samfsck(1M)` コマンドの詳細は、 `samfsck(1M)` のマニュアルページを参照してください。

自動ライブラリ内の装置を定義する行および自動ライブラリに関連するドライブ内の装置を定義する行では、同じファミリーセット名が指定される必要があります。

スタンドアロンの手動で読み込んだリムーバブルメディア装置では、このフィールドはダッシュ (-) にすることができます。

ファミリーセットの最初の装置の直前に、識別子 `#family-set-name` を入れることで、特定のファミリーセットに関連するコメントを作成できます。識別子のコメント行と、そのファミリーセット最後のデバイス行との間に追加されるすべてのコメントは、そのファミリーセットと関連付けられます。その後、 `File System Manager` ソフトウェアでファミリーセットが削除された場合、そのファミリーセットに関連するコメントも `mcf` から削除されます。

「Device State」フィールド

「Device State」フィールドでは、ファイルシステムを初期化したときの装置の状態を指定します。装置の有効な状態は `on` と `off` です。このフィールドはオプションです。値を指定しない場合は、このフィールドが省略されていることを示すためにダッシュ文字 (-) を挿入します。

「Additional Parameters」フィールド

SAM-QFS ファイルシステムでは、「Additional Parameters」フィールドは省略可能なので、空白のままにすることができます。デフォルトでは、ライブラリカタログファイルは `/var/opt/SUNWsamfs/catalog/family-set-name` に書き込まれます。このフィールドは、ライブラリカタログファイルの代替パスを指定するために使用されます。

Sun StorageTek QFS 共有ファイルシステムでは、このフィールドにはキーワード `shared` を指定してください。

Sun StorageTek QFS 非共有ファイルシステムでは、ダッシュ (-) を入力するか、このフィールドを空白のままにします。

mcf ファイルの例

ファイルシステムの構成はそれぞれ固有です。システムの条件や実際のハードウェアはサイトごとに異なります。次のコード例で、mcf ファイルの例を示します。

コード例 2-2 に、2 つのストライプ化グループがある Sun StorageTek QFS ファイルシステムの mcf ファイルを示します。

コード例 2-2 ストライプ化グループを示す mcf ファイルの例

```
# Sun StorageTek QFS file system configuration
#
# Equipment          Eq   Eq   Fam.  Dev.  Additional
# Identifier         Ord  Type Set   State Parameters
#-----
qfs1                 10   ma   qfs1  -
/dev/dsk/c2t1d0s7   11   mm   qfs1  -
/dev/dsk/c3t0d0s6   12   g0   qfs1  -
/dev/dsk/c3t0d1s6   13   g0   qfs1  -
/dev/dsk/c4t0d0s6   14   g1   qfs1  -
/dev/dsk/c4t0d1s6   15   g1   qfs1  -
```

コード例 2-3 に、3 つの Sun StorageTek QFS ファイルシステムの mcf ファイルを示します。

コード例 2-3 3 つの SAM-QFS ファイルシステムを示す mcf ファイルの例

```
# SAM-QFS file system configuration example
#
# Equipment          Eq   Eq   Fam.  Dev.  Additional
# Identifier         Ord  Type Set   State Parameters
#-----
qfs1                 10   ma   qfs1  -
/dev/dsk/c1t13d0s6  11   mm   qfs1  -
/dev/dsk/c1t12d0s6  12   mr   qfs1  -
#
qfs2                 20   ma   qfs2  -
/dev/dsk/c1t5d0s6   21   mm   qfs2  -
/dev/dsk/c5t1d0s6   22   mr   qfs2  -
#
qfs3                 30   ma   qfs3  -
/dev/dsk/c7t1d0s3   31   mm   qfs3  -
/dev/dsk/c6t1d0s6   32   mr   qfs3  -
/dev/dsk/c6t1d0s3   33   mr   qfs3  -
/dev/dsk/c5t1d0s3   34   mr   qfs3  -
```

コード例 2-4 に、md 装置を使用する 1 つの SAM-QFS ファイルシステムの mcf ファイルを示します。この mcf ファイルではテーブライブラリも定義されています。

コード例 2-4 1 つのファイルシステムと 1 つのライブラリを示す mcf ファイルの例

# Equipment	Eq	Eq	Fam.	Dev.	Additional
# Identifier	Ord	Type	Set	State	Parameters
#-----	---	--	-----	-----	-----
samfs1	10	ma	samfs1	-	
/dev/dsk/c1t2d0s6	11	mm	samfs1	-	
/dev/dsk/c1t3d0s6	12	md	samfs1	-	
/dev/dsk/c1t4d0s6	13	md	samfs1	-	
/dev/dsk/c1t5d0s6	14	md	samfs1	-	
# scalar 1000 and	12	AIT	tape	drives	
/dev/samst/c5t0u0	30	rb	robot1	-	
/dev/rmt/4cbn	101	tp	robot1	on	
/dev/rmt/5cbn	102	tp	robot1	on	
/dev/rmt/6cbn	103	tp	robot1	on	
/dev/rmt/7cbn	104	tp	robot1	off	
/dev/rmt/10cbn	105	tp	robot1	on	
/dev/rmt/11cbn	106	tp	robot1	on	
/dev/rmt/3cbn	107	tp	robot1	on	
/dev/rmt/2cbn	108	tp	robot1	on	
/dev/rmt/1cbn	109	tp	robot1	on	
/dev/rmt/0cbn	110	tp	robot1	on	
/dev/rmt/9cbn	111	tp	robot1	on	
/dev/rmt/8cbn	112	tp	robot1	on	

ファイルシステム構成のそのほかの例は、『Sun StorageTek QFS インストールおよびアップグレードの手引き』を参照してください。

ファイル設定、オプション、指示の相互関係

各ファイルシステムは mcf ファイルで定義されますが、ファイルシステムの動作は、デフォルトのシステム設定、/etc/vfstab ファイルの設定、samfs.cmd ファイルの設定、mount(1M) コマンドのオプションの相互関係によって決まります。

ストライプ幅などのいくつかのマウントオプションは、複数の場所で指定できます。このとき、設定方法によっては別の方法での設定が無効になります。

マウントオプションを指定するさまざまな方法については、62 ページの「マウントパラメータの設定」を参照してください。

ファイルシステムの初期化

ファイルシステムを新規作成する場合、または古いファイルシステムや破損したファイルシステムを交換する場合は、`sammkfs(1M)` コマンドを使用してそのファイルシステムを初期化する必要があります。

`sammkfs(1M)` コマンドによって新しいファイルシステムを構築します。
`-a allocation-unit` オプションを使用すると DAU 設定を指定できます。

Sun StorageTek QFS ソフトウェアの version 4U2 以降のリリースは、2 種類のスーパーブロックの設計をサポートします。コード例 2-5 の `samfsinfo(1M)` コマンドの出力は、`samfs1` ファイルシステムで version 2 のスーパーブロックが使用されていることを示しています。

コード例 2-5 `samfsinfo(1M)` コマンドの例

```
# samfsinfo samfs1
name:      samfs1          version:      2
time:      Wed Feb 21 13:32:18 1996
count:     1
capacity:  001240a0        DAU:      16
space:     000d8ea0
ord eq capacity space device
  0 10 001240a0 000d8ea0 /dev/dsk/c1t1d0s0
```

次に示すこれらのスーパーブロックの操作や機能の違いに注意してください。

- 4U0 より前のリリースは、version 1 のスーパーブロック設計のみをサポートしています。
- 4U0 以降のリリースは、version 2 のスーパーブロックをサポートしています。4U0 のソフトウェアをアップグレードとしてインストールした場合、version 2 のスーパーブロックに依存する機能を使用するには、あらかじめ release 4U0 以降の `sammkfs(1M)` コマンドを使用して、既存のファイルシステムを再初期化しておく必要があります。アクセス制御リスト (ACL) や Sun StorageTek QFS 共有ファイルシステムなどの特定の機能は、version 2 のスーパーブロックだけでサポートされます。ファイルシステムの再初期化については、ソフトウェアのインストールアップグレードプロセスで手順を説明しますが、ソフトウェアをインストールしたあとであればいつでも実行できます。



注意 – version 2 のスーパーブロックを使用するファイルシステムは、4U0 より前のリリースに戻すことができません。4U6 release のソフトウェアを使用して、version 1 のスーパーブロックを作成することはできません。

version 2 のスーパーブロックを必要とする機能や、`sammkfs(1M)` コマンドを使用した version 2 のスーパーブロックの作成についての詳細は、『Sun StorageTek QFS インストールおよびアップグレードの手引き』を参照してください。

次の例では、`sammkfs(1M)` コマンドのもっとも単純な形式を示します。引数はファイルシステム名だけです。

```
# sammkfs samqfs1
```

このコマンドは、スタンドアロンの Sun StorageTek QFS または SAM-QFS ファイルシステムのための version 2 のスーパーブロックを構築します。

`sammkfs(1M)` コマンド、そのオプション、version 1 および version 2 のスーパーブロックについての詳細は、`sammkfs(1M)` のマニュアルページを参照してください。`sammkfs(1M)` コマンドを使用して、共有 Sun StorageTek QFS ファイルシステムを初期化する方法については、『Sun StorageTek QFS インストールおよびアップグレードの手引き』を参照してください。

構成の例

この節では、構成サンプルを示し、サーバーで `mcf` ファイルを設定する際のさまざまな手順や考慮事項について説明します。次の手順について説明します。

- 45 ページの「Sun StorageTek QFS ラウンドロビン式ディスクの構成を作成する」
- 46 ページの「Sun StorageTek QFS ストライプ化ディスク構成を作成する」
- 47 ページの「Sun StorageTek QFS ストライプ化グループ構成を作成する」

SAM-QFS のすべての構成サンプルでは、自動ライブラリやその他の装置も定義し、最初からディスクキャッシュのサイズを超えてファイルシステムを拡張していることに注意してください。リムーバブルメディア装置の構成は、1 つの例だけに示しています。リムーバブルメディア装置の構成については、『Sun StorageTek Storage Archive Manager (SAM) インストールおよびアップグレードの手引き』を参照してください。

構成サンプルでは、ファイルシステムがシステムに読み込まれていることと、すべてのファイルシステムがマウントを解除されていることが前提になっています。

▼ Sun StorageTek QFS ラウンドロビン式ディスクの構成を作成する

この構成例は、応答時間の遅いディスクにメタデータを分離する Sun StorageTek QFS ファイルシステムのもので、ラウンドロビン式割り当てが 4 つのパーティションで使用されます。各ディスクは別のコントローラにあります。

この手順では、次のように仮定しています。

- メタデータ装置は、コントローラ 5 (装置順序 11 として指定された装置の LUN 0) で使用される単一パーティション (s6) です。
- データ装置は、4 つのコントローラに接続された 4 つのディスクで構成されます。

1. エディタを使用して、コード例 2-6 に示すように、mcf ファイルを作成します。

コード例 2-6 Sun StorageTek QFS ラウンドロビン式 mcf ファイルの例

```
# Sun StorageTek QFS disk cache configuration
# Round-robin mcf example
# Equipment      Eq  Eq    Fam.  Dev    Additional
# Identifier     Ord Type  Set   State  Parameters
#-----
qfs1             1   ma   qfs1
/dev/dsk/c5t0d0s6 11  mm   qfs1  on
/dev/dsk/c1t1d0s6 12  mr   qfs1  on
/dev/dsk/c2t1d0s6 13  mr   qfs1  on
/dev/dsk/c3t1d0s6 14  mr   qfs1  on
/dev/dsk/c4t1d0s6 15  mr   qfs1  on
```

2. `mkdir(1)` コマンドを使用して、`/qfs1` ファイルシステムの `/qfs` マウントポイントを作成します。

```
# mkdir /qfs
```

3. `sammkfs(1M)` コマンドを使用して、ファイルシステムを初期化します。
次の例では、デフォルトの 64K バイトの DAU を使用します。

```
# sammkfs qfs1
```

4. エディタを使用して、/etc/vfstab ファイルを変更します。

mr データ装置を含む Sun StorageTek QFS ファイルシステムは、デフォルトでストライプ化割り当てを使用します。そのため、ラウンドロビン式割り当てを使用する場合は stripe=0 を設定する必要があります。ファイルシステムに対してラウンドロビンを明示的に設定するには、次のように stripe=0 と設定します。

```
qfs1 - /qfs samfs - yes stripe=0
```

5. mount(1M) コマンドを使用して、ファイルシステムをマウントします。

```
# mount /qfs
```

▼ Sun StorageTek QFS ストライプ化ディスク構成を作成する

この構成例では、デフォルトでファイルデータが 4 つのデータパーティションにストライプ化されています。

この手順では、次のように仮定しています。

- メタデータ装置は、コントローラ 0 (LUN 1) で使用される単一パーティション (s6) です。メタデータは装置番号 11 だけに書き込まれます。
- データ装置は、4 つのコントローラに接続された 4 つのディスクで構成されます。各ディスクは別のコントローラにあります。

1. エディタを使用して、コード例 2-7 に示すように、mcf ファイルを作成します。

コード例 2-7 Sun StorageTek QFS ストライプ化ディスク mcf ファイルの例

```
# Sun StorageTek QFS disk cache configuration
# Striped Disk mcf example
# Equipment      Eq  Eq  Fam.  Dev.  Additional
# Identifier     Ord Type Set  State Parameters
#-----
qfs1             10  ma  qfs1
/dev/dsk/c0t1d0s6 11  mm  qfs1  on
/dev/dsk/c1t1d0s6 12  mr  qfs1  on
/dev/dsk/c2t1d0s6 13  mr  qfs1  on
/dev/dsk/c3t1d0s6 14  mr  qfs1  on
/dev/dsk/c4t1d0s6 15  mr  qfs1  on
```

2. `mkdir(1)` コマンドを使用して、`/qfs1` ファイルシステムの `/qfs` マウントポイントを作成します。

```
# mkdir /qfs
```

3. `sammkfs(1M)` コマンドを使用して、ファイルシステムを初期化します。
デフォルトの DAU は 64K バイトですが、次の例では DAU のサイズを 128K バイトに設定しています。

```
# sammkfs -a 128 qfs1
```

この構成では、このファイルシステムに書き込まれるすべてのファイルは、128K バイトとしてすべての装置上にストライプ化されます。

4. エディタを使用して、`/etc/vfstab` ファイルを変更します。

Sun StorageTek QFS ファイルシステムは、デフォルトでストライプ化割り当てを使用します。この例では、デフォルトの `stripe=1` をストライプ幅に設定しています。次の設定では、4 つのすべての `mr` 装置すべてに、1 DAU のストライプ幅でデータがストライプ化されます。

```
qfs1 - /qfs samfs - yes stripe=1
```

5. `mount(1M)` コマンドを使用して、ファイルシステムをマウントします。

```
# mount /qfs
```

▼ Sun StorageTek QFS ストライプ化グループ構成を作成する

ストライプ化グループを使用すると、大容量ファイルに対応できるように RAID 装置をグループ化できます。DAU は、ビットマップで 1 ビットとして表されます。ストライプ化グループに n 個の装置がある場合、 n に DAU を乗じた値が最小割り当てになります。ビットマップの 1 ビットだけを使用して $n \times$ DAU が表されます。

ストライプ化グループで割り当てられる最小ディスク容量は、次のとおりです。

割り当てられる最小ディスク領域 = DAU × グループ内のディスクの数



注意 – データの 1 バイトの書き込みは、ストライプ化グループに割り当てられた最小ディスク容量全体を満たします。ストライプ化グループの使用は、特定のアプリケーションに限られます。ファイルシステムでストライプ化グループを使用する場合は、その影響を理解していることが重要です。

ストライプ幅の合計に装置数を乗じた値よりも小さなファイル (この例では 128K バイト x 4 ディスク = 512K バイト未満のサイズのファイル) でも、512K バイトのディスク領域が使用されます。512K バイトを超えるファイルには、必要に応じて合計 512K バイト単位で領域が割り当てられます。

ストライプ化グループの装置は、同じサイズである必要があります。ストライプ化グループのサイズを増やして装置を追加することは不可能です。ただし、`samgrowfs(1M)` コマンドを使用して、ストライプ化グループを追加することは可能です。このコマンドの詳細は、`samgrowfs(1M)` のマニュアルページを参照してください。

この構成例は、応答時間の遅いディスクにメタデータを分離する Sun StorageTek QFS ファイルシステムのもので、2 つのストライプ化グループが 4 つのドライブに設定されています。

この手順では、次のように仮定しています。

- メタデータ装置は、コントローラ 0 (LUN 1) で使用される単一パーティション (s6) である。
- データ装置は、4 つのコントローラに接続している 4 つのディスク (2 つの同じディスクから構成される 2 つのグループ) で構成されます。各ディスクは、別の LUN にあります。パーティション 6 がディスク全体を占有する仮定され、ディスク全体がデータ格納のために使用されます。

1. エディタを使用して、コード例 2-8 に示すように、`mcf` ファイルを作成します。

コード例 2-8 Sun StorageTek QFS ストライプ化グループ `mcf` ファイルの例

```
# Sun StorageTek QFS disk cache configuration
# Striped Groups mcf example
# Equipment      Eq   Eq   Fam.  Dev.  Additional
# Identifier      Ord  Type Set   State Parameters
#-----
qfs1              10   ma  qfs1
/dev/dsk/c0t1d0s6 11   mm  qfs1  on
/dev/dsk/c1t1d0s6 12   g0  qfs1  on
/dev/dsk/c2t1d0s6 13   g0  qfs1  on
/dev/dsk/c3t1d0s6 14   g1  qfs1  on
/dev/dsk/c4t1d0s6 15   g1  qfs1  on
```

2. `mkdir(1)` コマンドを使用して、`/qfs1` ファイルシステムの `/qfs` マウントポイントを作成します。

```
# mkdir /qfs
```

3. `sammkfs(1M)` コマンドを使用して、ファイルシステムを初期化します。
次の例では、DAU サイズが 128K バイトに設定されます。

```
# sammkfs -a 128 qfs1
```

4. エディタを使用して、`/etc/vfstab` ファイルを変更します。

次の例では、デフォルトの `stripe=0` が設定されます。この場合、基本的に、ストライプ化グループ `g0` からストライプ化グループ `g1` までのラウンドロビン式割り当てが指定されます。

```
qfs1 - /qfs samfs - yes stripe=0
```

この `/etc/vfstab` ファイルでは、`stripe=` オプションを使用してストライプ幅が設定されます。この例では、`g0` と `g1` の 2 つのストライプ化グループがあります。`stripe=0` と指定されているので、ファイルは 2 つのストライプ化グループにラウンドロビン方式で書き込まれます。

注 – ストライプ化グループを作成したあとで構成を変更するには、`sammkfs(1M)` コマンドを再度実行する必要があります。

5. `mount(1M)` コマンドを使用して、ファイルシステムをマウントします。

```
# mount /qfs
```


第3章

操作の実行

この章では、ファイルシステムの操作に関連する項目を示します。この章の内容は次のとおりです。

- 51 ページの「ファイルとファイル属性の表示」
- 55 ページの「システムに対する構成ファイルの変更の伝達」
- 62 ページの「マウントパラメータの設定」
- 65 ページの「ファイルシステムのマウント解除」
- 66 ページの「ファイルシステムへのディスクキャッシュの追加」
- 68 ページの「ファイルシステムの再作成」

ファイルとファイル属性の表示

Sun StorageTek QFS ファイルシステムに固有の属性としては、ユーザー設定と一般的なファイル状態の両方があります。ここでは、これらの特性について説明し、`sls` コマンドを使用して表示する方法について説明します。

ファイル属性とファイル状態

ファイルのユーザー指定の属性とシステム指定の状態が、ファイルの `i` ノードに格納されます。`sls(1) -D` コマンドを使用すると、これらの `i` ノード属性を表示できます。`sls(1)` のオプションの詳細は、`sls(1)` のマニュアルページを参照してください。

ユーザーは、次のコマンドを指定して属性を設定できます。

- `archive(1)`
- `ssum(1)`

- release(1)
- segment(1)
- setfa (1)
- stage(1)

また、次のアプリケーションプログラミングインタフェース (API) ルーチンを指定すると、アプリケーションによって属性を設定できます。

- sam_archive(3)
- sam_release(3)
- sam_segment(3)
- sam_setfa(3)
- sam_ssum(3)
- sam_stage(3)

表 3-1 に、i ノードに含まれるユーザー指定の属性を示します。

表 3-1 ユーザー指定ファイル属性

ファイル属性	説明
setfa -D	ファイルが直接入出力の対象としてマークされます。
setfa -gn	ファイルが、ストライプ化グループ <i>n</i> に割り当てられるようにマークされます。
setfa -sm	ファイルが、ストライプ幅 <i>m</i> で割り当てられるようにマークされます。

注 – Sun StorageTek SAM のアーカイブ機能に固有の、ユーザー指定のファイル属性が多数あります。詳細については、『Sun StorageTek Storage Archive Manager (SAM) ファイルシステム構成および管理マニュアル』を参照してください。

表 3-1 に示す属性は、ファイルとディレクトリの両方に設定できます。ディレクトリの属性を設定すると、そのあとでそのディレクトリに作成されるファイルは、すべてのディレクトリ属性を継承します。親ディレクトリに属性が適用される前に作成されたファイルは、ディレクトリの属性を継承しません。

WORM-FS パッケージをインストールした場合は、ファイルに WORM (Write Once Read Many) 属性を適用したり、ファイルの保存期間を設定したりすることもできます。詳細は、194 ページの「WORM-FS ファイルシステムの構成」を参照してください。

ファイル情報の表示

Sun StorageTek QFS の `s1s(1)` コマンドは、標準の UNIX の `ls(1)` コマンドを拡張したもので、ファイルに関して詳細な情報を表示します。コード例 3-1 に、ファイル `hgc2` の `i` ノード情報を表示した `s1s(1)` コマンドの出力の詳細を示します。

コード例 3-1 SAM-QFS 環境での `s1s(1)` の出力

```
# s1s -D hgc2
hgc2:
mode: -rw-r--r--  links: 1  owner: root      group: other
length: 14971  admin id: 0  inode: 30.5
archdone;
segments 3, offline 0, archdone 3, damaged 0;
copy 1: ---- Jun 13 17:14      2239a.48  lt MFJ192
copy 2: ---- Jun 13 17:15      9e37.48  lt AA0006
access: Jun 13 17:08  modification: Jun 13 17:08
changed: Jun 13 17:08  attributes: Jun 13 17:10
creation: Jun 13 17:08  residence: Jun 13 17:08
```

`s1s(1)` 出力について

表 3-2 に、コード例 3-1 に示した `s1s(1)` 出力の各行の意味を示します。

注 – アーカイブに関する行は、Sun StorageTek SAM 環境の `s1s(1)` 出力にのみ表示されます。

表 3-2 `s1s(1)` 出力の説明

行番号	タグ	内容
1	<code>mode:</code>	ファイルのモードと権限、ファイルへのハードリンク数、ファイルの所有者、および所有者が属するグループ。
2	<code>length:</code>	ファイルサイズ (バイト数)、ファイルの管理 ID 番号、およびファイルの <code>i</code> ノード番号。 デフォルトでは、管理 ID 番号は 0。この番号が 0 よりも大きい場合は、ファイルやブロックを数えるための、ファイルのアカウントカテゴリを示します。この番号は、ファイルシステム割り当てがこのファイルシステムで有効になっていない場合でも、0 より大きい値に設定できます。ファイルシステム割り当ての詳細については、99 ページの「ファイルシステム割り当ての管理」を参照してください。 <code>i</code> ノード番号には 2 つの部分があり、 <code>i</code> ノード番号、ピリオド (.)、 <code>i</code> ノード生成番号の順に構成されます。

表 3-2 sls(1) 出力の説明 (続き)

行番号	タグ	内容
3	archdone;	ファイル固有のファイル属性。この行の詳細は、sls(1) のマニュアルページを参照してください。
4	segments	<p>セグメントインデックス情報。この行は、ファイルがセグメントインデックスでない場合には表示されません。この行の一般的な書式は次のとおりです。</p> <p>segments <i>n</i>, offline <i>o</i>, archdone <i>a</i>, damaged <i>d</i>;</p> <ul style="list-style-type: none"> • segments <i>n</i> は、このファイルのデータセグメントの総数を示します。この例では 3 です。 • offline <i>o</i> は、オフラインのデータセグメントの数を示します。この例では、オフラインセグメントがありません。 • archdone <i>a</i> は、アーカイブ条件の満たされたセグメントの数を示します。この例では 3 です。 • damaged <i>d</i> は、破壊されたセグメントの数を示します。この例では、破壊されたセグメントがありません。
5, 6	copy 1: , copy 2:	<p>アーカイブコピー行です。sls(1) コマンドは、各アーカイブまたは期限切れのアーカイブコピーのアーカイブコピー行を 1 行表示します。</p> <p>この行の 4 つの位置は次のものを示しています。</p> <p>1 – エントリが期限切れかアクティブか。</p> <ul style="list-style-type: none"> • s は、アーカイブコピーが期限切れであることを示します。つまり、ファイルが変更されたため、このアーカイブのコピーは以前のファイルの状態です。 • u は、コピーがアーカイブ解除されたことを示します。アーカイブ解除は、ファイルまたはディレクトリのアーカイブエントリが削除されるプロセスです。 • ハイフン (-) は、アーカイブのコピーがアクティブで有効であることを示します。 <p>2 – アーカイブのコピーが再アーカイブされるかどうか。</p> <ul style="list-style-type: none"> • r は、アーカイブのコピーの再アーカイブのスケジュールがアーカイバによって設定されていることを示します。 • ハイフン (-) は、アーカイブのコピーがアーカイバによって再アーカイブされないことを示します。 <p>3 – 未使用。</p> <p>4 – コピーが破壊されているかどうか。</p> <ul style="list-style-type: none"> • D は、アーカイブのコピーが破壊されていることを示します。破壊されたアーカイブのコピーは、書き込みの対象外です。 • ハイフン (-) は、アーカイブコピーが破壊されていないことを示します。このアーカイブのコピーは書き込みの対象外です。

表 3-2 sls(1) 出力の説明 (続き)

行番号	タグ	内容
		<p>アーカイブコピーの残りの行の書式は次のとおりです。</p> <ul style="list-style-type: none"> • アーカイブのコピーがアーカイブメディアに書き込まれた日付と時刻。 • 小数点 (.) で区切られた 2 つの 16 進数。最初の 16 進数 (2239a) は、カートリッジ上のアーカイブファイルの開始位置を示します。2 番目の 16 進数 (48) は、アーカイブファイル内の、このコピーのファイルバイトオフセット (512 で割った値) です。 • アーカイブコピーが格納された、メディアタイプとボリュームシリアル名 (VSN)。
7	access:	ファイルが最後にアクセスおよび変更された時刻。
8	changed:	ファイルの内容とファイルの属性が最後に変更されてからの時間。
9	creation:	ファイルが作成され、ファイルシステムに格納された時刻。

保存行について

オプションの WORM-FS パッケージを使用している場合は、sls(1) の出力に保存行も表示されます。保存行の書式は次のとおりです。

```
retention: active retention-period: 3y 0d 0h 0m
```

これは、このファイルに保存期間が設定されているかを示し、設定されている場合はその長さを示します。retention-end の日付は保存期間が期限切れになる日付を示しています。WORM-FS 機能の使用方法については、194 ページの「WORM-FS ファイルシステムの構成」を参照してください。

システムに対する構成ファイルの変更の伝達

ここでは、システムから構成ファイルの変更を伝達する手順について説明します。この手順では、次のファイルの変更を伝達します。

- mcf(4)
- defaults.conf

- archiver.cmd (SAM-QFS ファイルシステムのみ)
- stager.cmd (SAM-QFS ファイルシステムのみ)
- 共有ホストファイル (Sun StorageTek QFS 共有および SAM-QFS 共有ファイルシステムのみ)

これらの手順は、次の場合に実行する必要があります。

- 情報の追加、削除、または訂正のために、これらのファイルを更新する場合。
- Sun StorageTek QFS ファイルシステムがすでに動作しているときに、Sun StorageTek SAM archiver.cmd、defaults.conf、または stager.cmd のファイルを作成または更新する場合。

次の各項目では、これらの手順について説明します。

- 56 ページの「mcf または defaults.conf の情報を Sun StorageTek QFS Sun Cluster 環境で変更する」
- 57 ページの「mcf または defaults.conf ファイルシステムの情報を SAM-QFS 環境で変更する」
- 58 ページの「mcf または defaults.conf リムーバブルメディアドライブ情報を変更する」
- 59 ページの「共有ホストファイルの変更」
- 59 ページの「新しいエントリを追加、または既存のエントリを変更する」
- 60 ページの「ホスト名の変更、エントリの並べ替え、エントリの挿入を行う」

▼ mcf または defaults.conf の情報を Sun StorageTek QFS Sun Cluster 環境で変更する

Sun Cluster 環境で高可用性のために構成される共有ファイルシステムの mcf または defaults.conf 情報を変更するには、Sun Cluster システム内のすべての参加ノードでこの手順を実行します。

1. vi(1) などのエディタを使用してファイルを編集し、ファイルシステム情報を変更します。
2. mcf ファイルを変更する場合は、sam-fsd(1M) コマンドを使用して mcf ファイルに誤りがないか調べます。

```
# sam-fsd
```

このコマンドの出力にエラーがある場合は、次の操作に進む前にエラーを修正します。

3. `samd(1M) config` コマンドを使用して、`mcf(4)` または `defaults.conf` ファイルの変更を伝達します。

```
# samd config
```

これらのファイルの詳細は、`defaults.conf(4)` または `mcf(4)` のマニュアルページを参照してください。

▼ `mcf` または `defaults.conf` ファイルシステムの情報を SAM-QFS 環境で変更する

1. `vi(1)` などのエディタを使用してファイルを編集し、ファイルシステム情報を変更します。
2. `mcf` ファイルを変更する場合は、`sam-fsd(1M)` コマンドを使用して `mcf` ファイルに誤りがないか調べます。

```
# sam-fsd
```

このコマンドの出力にエラーがある場合は、次の操作に進む前にエラーを修正します。

3. 1 つ以上のファイルシステムに関連する情報を削除または変更する場合は、`samcmd(1M) aridle` コマンドを実行し、`mcf` ファイルで定義された影響を受ける各ファイルシステムのアーカイバを休止状態にします。

このコマンドは、次の形式で使用します。

```
samcmd aridle fs.fsname
```

fsname には、ファイルシステムの名前を指定します。

4. 1 つ以上のドライブに関連する情報を削除または変更する場合は、`samcmd(1M) idle` コマンドを実行し、`mcf` ファイル内の影響を受ける各ドライブに割り当てられた装置番号を休止状態にします。

このコマンドは、次の形式で使用します。

```
samcmd idle eq
```

eq には、ドライブの装置番号を指定します。

5. `umount(1M)` コマンドを使用して、変更の影響を受ける各ファイルシステムをマウント解除します。

ファイルシステムのマウント解除方法については、65 ページの「ファイルシステムのマウント解除」を参照してください。

6. `samcmd(1M) config` コマンドを使用して、変更を伝達します。

```
# samd config
```

7. `mount(1M)` コマンドを使用して、マウント解除したファイルシステムを再マウントします。

これらのファイルの詳細は、`defaults.conf(4)` または `mcf(4)` のマニュアルページを参照してください。

▼ `mcf` または `defaults.conf` リムーバブルメディアドライブ情報を変更する

1. ファイルを編集して、リムーバブルメディアドライブの情報を変更します。
2. `mcf` ファイルを変更する場合は、`sam-fsd(1M)` コマンドを使用して `mcf` ファイルに誤りがないか調べます。

```
# sam-fsd
```

このコマンドの出力にエラーがある場合は、次の操作に進む前にエラーを修正します。

3. 1 つ以上のファイルシステムに関連する情報を削除または変更する場合は、`samcmd(1M) aridle` コマンドを実行し、`mcf` ファイルで定義された影響を受ける各ファイルシステムのアーカイバを休止状態にします。

このコマンドは、次の形式で使用します。

```
samcmd aridle fs.fsname
```

fsname には、ファイルシステムの名前を指定します。

- 1 つ以上のドライブに関連する情報を削除または変更する場合は、`samcmd(1M)` `idle` コマンドを、`mcf` ファイル内の影響を受ける各ドライブに割り当てられた装置番号について実行します。

このコマンドは、次の形式で使います。

```
samcmd idle eq
```

`eq` には、ドライブの装置番号を指定します。

- `samd(1M)` `stop` コマンドを使用して、すべてのリムーバブルメディアの動作を停止します。

```
# samd stop
```

- `samd(1M)` `config` コマンドを使用して、変更を伝達し、システムを再起動します。

```
# samd config
```

- `samd(1M)` `start` コマンドを使用して、すべてのリムーバブルメディアの動作を再開します。

```
# samd start
```

これらのファイルの詳細は、`defaults.conf(4)` または `mcf(4)` のマニュアルページを参照してください。

共有ホストファイルの変更

ファイルシステムをマウント解除することなく、共有ホストファイルに新しいホストエントリを追加したり、既存のエントリを変更したりできます。ホスト名の変更、エントリの並べ替え、またはエントリの挿入を行うには、その前にファイルシステムをマウント解除する必要があります。共有ホストファイルを変更するには、次の手順を使います。

▼ 新しいエントリを追加、または既存のエントリを変更する

共有ホストファイルの最後に新しいホストエントリを追加したり、共有ホストファイルの既存のエントリの列 2 ~ 5 を変更したりするには、この手順を使います。

1. メタデータサーバーとして機能しているホストがわからない場合は、`samsharefs(1M) family-set-name` コマンドを実行してメタデータサーバー名を表示します。

このコマンドは、ファイルシステムを構成した任意のホストから実行します。

例:

```
# samsharefs sharefs1
```

2. メタデータサーバーで、共有ホストファイルを一時作業ファイルに保存します。

例:

```
# samsharefs sharefs1 > /tmp/file
```

3. (省略可能) 共有ホストファイルのコピーを保存します。

例:

```
# cp /tmp/file /var/opt/SUNWsamfs/hosts.date
```

4. `vi(1)` などのエディタを使用して、一時作業ファイルを編集します。

マウントされたファイルシステムでは、ファイルの末尾に新しいホストエントリを追加し、既存のエントリの列 2 ~ 5 を変更できます。

5. 一時作業ファイルを保存し、閉じます。

6. 一時作業ファイルを `SUNWsamfs` ディレクトリにコピーします。

例:

```
# cp /tmp/file /var/opt/SUNWsamfs/hosts.sharefs1
```

7. 新しい共有ホストファイルをファイルシステムに適用します。

例:

```
# samsharefs -u sharefs1
```

▼ ホスト名の変更、エントリの並び替え、エントリの挿入を行う

共有ホストファイルでホスト名の変更、エントリの並び替え、またはエントリの挿入を行うには、ファイルシステムをマウント解除する必要があります

1. メタデータサーバーとして機能しているホストがわからない場合は、`samsharefs(1M) -R family-set-name` コマンドを実行してメタデータサーバー名を表示します。

このコマンドは、ファイルシステムを構成した任意のホストから実行します。

例:

```
# samsharefs -R sharefs1
```

2. 各参加クライアントでファイルシステムをマウント解除してから、メタデータサーバーをマウント解除します。
3. メタデータサーバーで、共有ホストファイルを一時作業ファイルに保存します。

例:

```
# samsharefs -R sharefs1 > /tmp/file
```

4. (省略可能) 共有ホストファイルのコピーを保存します。

例:

```
# cp /tmp/file /var/opt/SUNWsamfs/hosts.date
```

5. `vi(1)` などのエディタを使用して、一時作業ファイルを編集します。
6. 共有ホストファイルを保存し、閉じます。
7. 新しい共有ホストファイルを `SUNWsamfs` ディレクトリにコピーします。

例:

```
# cp /tmp/file /var/opt/SUNWsamfs/hosts.sharefs1
```

8. 新しい共有ホストファイルをファイルシステムに適用します。

例:

```
# samsharefs -uR sharefs1
```

9. メタデータサーバーでファイルシステムをマウントしてから、クライアントでファイルシステムをマウントします。

マウントパラメータの設定

Solaris OS の `mount(1M)` コマンドを使用して、Sun StorageTek QFS ファイルシステムをマウントできます。

マウントパラメータを使用して、ファイルシステムの特性を操作します。マウントパラメータの指定方法はいくつかあります。最上位階層の方法は、下位の階層の方法よりも優先されます。マウントオプションは次の方法で指定できます。方法は、階層が上のものから順に示してあります。

- `mount(1M)` コマンドのコマンド行オプション。Solaris OS `mount(1M)` コマンド行で指定したオプションによって、`/etc/vfstab` ファイルで指定したそのほかのオプション、`samfs.cmd` ファイルで指定した指示、システムのデフォルト設定が無効になります。
- `/etc/vfstab` ファイルの設定。
- `samfs.cmd` ファイルでの指示。
- システムデフォルト。デフォルトのシステム設定は、Solaris OS にすでに定義されている構成可能な設定。`samfs.cmd` ファイル、`/etc/vfstab` ファイル、および `mount(1M)` コマンドの指定で、システム設定を無効にできます。

また、`samu(1M)` オペレータユーティリティーまたは `samcmd(1M)` コマンドを使用して、マウントオプションを指定することもできます。この方法を使用して有効または無効にしたマウントオプションでは、ファイルシステムがマウント解除されるまで、その状態は継続します。

次の項では、マウントオプションの指定方法について説明します。マウントオプションの指定方法の詳細については、付録 B 253 ページの「Sun StorageTek QFS 共有ファイルシステムのマウントオプション」を参照してください。『Sun StorageTek QFS インストールおよびアップグレードの手引き』にもファイルシステムのマウント方法についての説明があります。

`mount(1M)` コマンド

Sun Solaris OS の `mount(1M)` コマンドを使用すると、ファイルシステムをマウントし、新たに設定を指定して `/etc/vfstab` ファイルや `/etc/opt/SUNWSamfs/samfs.cmd` ファイルに指定した設定を無効にできます。たとえば、ストライプ幅、先読み、後書き、ディスクキャッシュ利用率の最高境界値と最低境界値を指定できます。

`samfs.cmd` ファイルと組み合わせて `mount(1M)` コマンドを使用する方法の 1 つとしては、主に `samfs.cmd` ファイルでマウントオプションを指定して、システムの設定を試したり調整したりするときに `mount(1M)` コマンドのオプションを使用できます。

たとえば、次のコマンドは `setuid` 実行を禁止し、`qwrite` を使用可能にした状態で、ファイルシステム `qfs1` を `/work` にマウントします。`qfs1` ファイルシステム名は、装置 ID です。これは、`mcf` ファイルで、このファイルシステムの「Equipment Identifier」フィールドにも指定されています。複数のマウントオプションを指定するには、コンマで各オプションを区切ります。

```
# mount -o nosuid,qwrite qfs1 /work
```

Sun StorageTek QFS または SAM-QFS 共有ファイルシステムをマウントする場合は、はじめにメタデータサーバーでファイルシステムをマウントしてから、各参加クライアントホストでファイルシステムをマウントしてください。`mount` コマンドには `shared` オプションを付けます。このコマンドはメタデータサーバーと参加ホストで同じにする必要があることに注意してください。

`mount(1M)` コマンドの詳細については、`mount_samfs(1M)` のマニュアルページを参照してください。

/etc/vfstab ファイル

`mcf` ファイルで定義された各 Sun StorageTek QFS ファイルシステムは、Solaris OS のシステムファイル `/etc/vfstab` 内に対応する行が必要です。これは、ファイルシステムをマウントするために必要です。

`/etc/vfstab` ファイルのファイルシステム行の例を次に示します。

```
qfs1 - /qfs samfs - yes stripe=0
```

左から順に、各フィールドは次の内容を示しています。

- ファイルシステムのファミリーセット名。
- `samfsck(1M)` に対するファイルシステム。
- マウントポイント。
- ファイルシステムタイプ。これは、Sun StorageTek QFS ファイルシステムの場合でも、常に `samfs` です。
- `samfsck(1M)` パス。
- ブートオプションでのマウント。
- 空白を含まず、コンマで区切られたマウントパラメータ。

`/etc/vfstab` ファイルのフィールドは、空白文字またはタブ文字で区切る必要があります。

マウントパラメータのフィールドには、`mount_samfs(1M)` のマニュアルページに `-o` オプションの引数として記載されているマウントパラメータをどれでも指定できます。これらのパラメータは、`samfs.cmd` ファイルの指示行として、または `mount(1M)` コマンドの `-o` オプションの引数として指定できるパラメータとほぼ同じです。`samfs.cmd` ファイルの場合には、さまざまな入出力の設定、先読み、後書き、ストライプ幅、ストレージ/アーカイブ管理のさまざまな設定、`Qwrite`、およびそのほかの機能についての指定を組み込むことができます。

使用可能なマウントパラメータの詳細は、`mount_samfs(1M)` のマニュアルページを参照してください。`/etc/vfstab` ファイルの変更の詳細は、`vfstab(4)` のマニュアルページを参照してください。

samfs.cmd ファイル

`/etc/opt/SUNWsamfs/samfs.cmd` ファイルを使用すると、すべての Sun StorageTek QFS ファイルシステムに対してマウントパラメータを指定できます。このファイルは、複数のファイルシステムすべてに、同じマウントパラメータを指定して構成する場合に役立ちます。

このファイルを使用すると、すべてのマウントパラメータを 1 つの場所に読みやすい書式で定義できます。このファイルの先頭部分で指定される指示はグローバル指示で、すべての Sun StorageTek QFS ファイルシステムに適用されます。このファイルの次の部分には、個々のファイルシステムに適用する固有のパラメータを指定できます。共通のパラメータを一度に 1 つの場所に指定できる点が、このファイルと `/etc/vfstab` ファイルとの違いです。`/etc/vfstab` ファイルでは、各ファイルシステムのすべてのマウントパラメータを指定する必要があります。

`samfs.cmd` ファイルに指定できるマウントパラメータは、`/etc/vfstab` ファイルや、`mount(1M)` コマンドの `-o` オプションの引数として指定できるパラメータとほとんど同じです。指定できる使用可能なマウントパラメータは、入出力の設定、先読み、後書き、ストライプ幅、ストレージ/アーカイブ管理のさまざまな設定、`WORM-FS`、`Qwrite`、およびそのほかの機能に関連するものです。このファイルに指定できるマウントパラメータの詳細は、`samfs.cmd(4)` のマニュアルページを参照してください。

`samfs.cmd` ファイルでは、1 行に 1 つの指示を指定します。ファイルにはコメントを指定することもできます。ハッシュ記号 (`#`) を先頭に付けてください。ハッシュ記号の右側の文字は、コメントとして扱われます。

すべてのファイルシステムにグローバルに適用する指示は、すべての `fs =` 行よりも前に記述します。特定のファイルシステム固有の指示は、すべてのグローバル指示よりもあとの `fs =` で始まる行に記述します。特定のファイルシステム固有の指示によって、グローバル指示は無効になります。

コード例 3-2 は、すべてのファイルシステムにディスクキャッシュ利用率の最低境界値と最高境界値が設定され、2つのファイルシステムの個別のパラメータが指定された、`samfs.cmd` ファイルの例を示しています。

コード例 3-2 `samfs.cmd` ファイルの例

```
low = 50
high = 75
fs = samfs1
    high = 65
    writebehind = 512
    readahead = 1024
fs = samfs5
    partial = 64
```

`samfs.cmd` ファイルの指示が使用され、デフォルトのシステム設定が無効になります。ただし、このファイルの指示は `mount(1M)` コマンドの引数によって無効になります。また、`/etc/vfstab` ファイルのエントリによっても、`samfs.cmd` ファイルに指定された指示は無効になります。

`mount(1M)` コマンドについては、`mount_samfs(1M)` のマニュアルページを参照してください。`samfs.cmd` ファイルに入力できる指示の詳細は、`samfs.cmd(4)` のマニュアルページを参照してください。

ファイルシステムのマウント解除

Solaris OS の `umount(1M)` コマンドを使用して、Sun StorageTek QFS ファイルシステムをマウント解除できます。

▼ スタンドアロン QFS または SAM-QFS ファイルシステムをマウント解除する

- `umount(1M)` コマンドを使用して、ファイルシステムをマウント解除します。

```
# umount /samqfs
```

ファイルシステムをマウント解除する前に、NFS で使用するために共有していたファイルシステムを `unshare(1M)` コマンドで非共有にできます。非共有にしたあと、マウント解除コマンドを再度発行してみてください。このコマンドが失敗し、

ファイルシステムのマウント解除が必要な場合は、`umount(1M)` コマンドに `-f` オプションを付けます。`-f` オプションを使用すると、ファイルシステムのマウント解除が強制されます。

マウント解除手順の詳細については、『Sun StorageTek QFS インストールおよびアップグレードの手引き』を参照してください。

▼ Sun StorageTek QFS 共有ファイルシステムをマウント解除する

73 ページの「共有ファイルシステムをマウント解除する」の説明に従います。

ファイルシステムへのディスクキャッシュの追加

ファイルシステムのディスクキャッシュを増やすには、ディスクパーティションまたはディスクドライブを追加してから、`mcf` ファイルを更新し、`samgrowfs(1M)` コマンドを使用してファイルシステムを拡張します。ファイルシステムを再初期化または復元する必要はありません。

`mcf` ファイルを変更するときは、次の点に注意してください。

- 1 ファイルシステムには最大 252 個のディスクパーティションを構成できます。
- Sun StorageTek QFS ファイルシステムのサイズを増やすには、1 つ以上の新しいメタデータパーティションを追加する必要があります。メタデータパーティションでは、装置タイプ値 `mm` が必要です。
- メタデータまたはデータ用に新しいパーティションを追加する場合は、`mcf` ファイルで、既存のディスクパーティションの次に追加します。
- `mcf` ファイルでは、装置 ID の名前を変更しないでください。`mcf` ファイル内の名前がスーパーブロック内の名前と一致しないと、ファイルシステムをマウントできなくなります。または、次のメッセージが `/var/adm/messages` に記録されます。

```
WARNING SAM-FS superblock equipment identifier <id>s on eq <eq>
does not match <id> in mcf
```

▼ ファイルシステムにディスクキャッシュを追加する

1. `umount(1M)` コマンドを使用して、拡張するファイルシステムをマウント解除します。

ファイルシステムが共有されている場合は、すべてのクライアントホストでファイルシステムをマウント解除してから、メタデータサーバーでファイルシステムをマウント解除します。次に、メタデータサーバーで、この手順の残りの操作を実行できます。

ファイルシステムのマウント解除の詳細は、65 ページの「ファイルシステムのマウント解除」を参照してください。

2. この手順でファイルシステムの名前を変更する場合は、`-R` オプションと `-F` オプションを指定して `samfsck(1M)` コマンドを実行し、名前を変更します。

このコマンドの詳細は、`samfsck(1M)` のマニュアルページを参照してください。

3. `/etc/opt/SUNWsamfs/mcf` ファイルを編集し、ディスクキャッシュを追加します。

4. `sam-fsd(1M)` 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ます。

```
# sam-fsd
```

このコマンドの出力にエラーがある場合は、次の操作に進む前にエラーを修正します。

5. `samd(1M) config` コマンドを実行して、`mcf` ファイルの変更をシステムに伝達します。

```
# samd config
```

詳細は、`samd(1M)` のマニュアルページを参照してください。

6. 拡張されたファイルシステムで、`samgrowfs(1M)` コマンドを実行します。

たとえば、次のコマンドを入力して、ファイルシステム `samfs1` を拡張します。

```
# samgrowfs samfs1
```

ファイルシステムの名前を変更した場合は、新しい名前を使用して `samgrowfs(1M)` コマンドを実行します。このコマンドの詳細は、`samgrowfs(1M)` のマニュアルページを参照してください。

7. ファイルシステムをマウントします。

Sun StorageTek QFS ファイルシステムのマウントについては、`mount_samfs(1M)` のマニュアルページを参照してください。

8. ファイルシステムが Sun StorageTek QFS 共有ファイルシステムの場合は、メタデータサーバーの `mcf` ファイルと一致するよう、参加している各クライアントホストの `mcf` ファイルを編集します。

ファイルシステムの再作成

次のいずれかを実行するためには、ファイルシステムを再作成する必要があります。

- ディスクまたはパーティションの変更
- ディスクまたはパーティションの追加
- ディスクまたはパーティションの削除

ここでは、この手順について説明します。

▼ ファイルシステムをバックアップして再作成する

1. サイトでカスタマイズしたすべてのシステムファイルと構成ファイルのバックアップを取ります。

使用しているソフトウェアに応じて、バックアップするファイルには `mcf`、`archiver.cmd`、`defaults.conf`、`samfs.cmd`、または `inquiry.conf` が含まれます。Sun StorageTek QFS 環境のすべてのファイルシステムについて、これらのファイルのバックアップを取ります。また、`/etc/opt/SUNWsamfs` ディレクトリのファイル、`/var/opt/SUNWsamfs` ディレクトリのファイル、および共有ホストファイルのバックアップコピーがあることを確認してください。

2. 変更する各ファイルシステムのバックアップを取ったことを確認します。

ファイルシステムは、サイトの方針に従って定期的にバックアップを取る必要があります。ファイルシステムにすでに存在するバックアップファイルに問題がない場合は、ここで再度バックアップを取る必要はありません。ただし、最新のダンプファイルの作成後に作成された情報を保持するためにファイルシステムのバックアップを取る必要がある場合は、ここでバックアップを取ってください。`qfsdump` コマンドを実行してダンプファイルを作成する方法については、『Sun StorageTek QFS インストールおよびアップグレードの手引き』を参照してください。

3. ファイルシステムをマウント解除します。

方法については、65 ページの「ファイルシステムのマウント解除」を参照してください。

4. この手順でファイルシステムの名前を変更する場合は、`-R` オプションと `-F` オプションを指定して `samfsck(1M)` コマンドを実行します。
詳細については、`samfsck(1M)` のマニュアルページを参照してください。
5. `/etc/opt/SUNWsamfs/mcf` ファイルを編集し、パーティションを追加、変更、または削除します。
詳細は、66 ページの「ファイルシステムへのディスクキャッシュの追加」を参照してください。
6. `sam-fsd(1M)` 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ます。

```
# sam-fsd
```

このコマンドの出力で `mcf` ファイルのエラーがある場合は、次の操作に進む前にエラーを修正します。

7. `samd(1M) config` コマンドを実行して、`mcf` ファイルの変更をシステムに伝達します。

```
# samd config
```

詳細は、`samd(1M)` のマニュアルページを参照してください。

8. `sammkfs(1M)` コマンドを実行して、ファイルシステムを再作成します。
たとえば、次のコマンドでは `samfs10` が作成されます。

```
# sammkfs samfs10
```

9. `mount(1M)` コマンドを実行して、ファイルシステムをマウントします。
Sun StorageTek QFS ファイルシステムのマウントについては、`mount_samfs(1M)` のマニュアルページを参照してください。
10. `cd(1)` コマンドを実行して、ファイルシステムのマウントポイントに移動します。
11. `qfsrestore(1M)` コマンドまたは File System Manager を使用して、各ファイルを復元します。
所有していたダンプファイル、または手順 1 で作成したダンプファイルから復元します。
詳細については、`qfsdump(1M)` のマニュアルページ、または File System Manager のオンラインヘルプを参照してください。

第4章

Sun StorageTek QFS 共有ファイルシステムの構成

この章では、Sun StorageTek QFS 共有ファイルシステムの構成方法および保守方法について説明します。この章の内容は次のとおりです。

- 72 ページの「Sun StorageTek QFS 共有ファイルシステムのマウントとマウント解除」
- 74 ページの「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」
- 78 ページの「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」
- 80 ページの「クライアントホストの追加または削除」
- 93 ページの「Sun StorageTek QFS 環境でのメタデータサーバーの変更」
- 98 ページの「Sun StorageTek QFS 共有ファイルシステムでのクライアントサーバー通信」

Sun StorageTek QFS 共有ファイルシステムのマウントとマウント解除

Sun StorageTek QFS 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` ファイルを作成して、すべてのホストにコピーできます。

Sun StorageTek QFS 共有ファイルシステムのマウントの詳細については、253 ページの「Sun StorageTek QFS 共有ファイルシステムのマウントオプション」および `mount_samfs(1M)` のマニュアルページを参照してください。ファイルシステムのマウントおよびマウント解除の詳細については、第 3 章、51 ページの「操作の実行」を参照してください。

▼ 共有ファイルシステムをマウントする

1. メタデータサーバーおよびすべてのクライアントホストで、スーパーユーザーになります。
2. `mount(1M)` コマンドを使用して、メタデータサーバーをマウントします。
ファイルシステムは、いずれかのクライアントホストにマウントする前に、メタデータサーバーにマウントします。
3. `mount(1M)` コマンドを使用して、クライアントホストをマウントします。
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。
`mount(1M)` コマンドについては、`mount(1M)` 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をマウント解除する

1. `umount(1M)` コマンドを使用して、すべての参加クライアントホストのファイルシステムをマウント解除します。

例:

```
client# umount /samqfs
```

必要に応じて、`umount(1M)` コマンドの `-f` オプションを使用します。`-f` オプションを使用すると、ファイルシステムのマウント解除が強制されます。

注 – ファイルシステムがメタデータサーバーにマウントされていないと、共有クライアントの強制マウント解除は完了しない場合があります。

2. メタデータサーバーでファイルシステムをマウント解除します。

```
metaserver# umount /samqfs
```

マウント解除するとき、ファイルシステムにいくつかの条件が存在することがあり、そのために `umount(1M)` コマンドの再実行が必要になる場合があります。それでもファイルシステムがマウント解除されない場合は、`unshare(1M)`、`fuser(1M)`、または別のコマンドを `umount(1M)` コマンドと組み合わせて使用してください。マウント解除手順の詳細については、`umount(1M)` のマニュアルページおよび『Sun StorageTek QFS インストールおよびアップグレードの手引き』を参照してください。

`-o await_clients #` フラグを `umount` コマンドと一緒に使用することもできます。これにより、マウント解除プロセスは、指定された秒数 (#) だけ、クライアントがマウント解除するのを待ちます。待機期間が終わると、またはすべてのクライアントがマウント解除を完了すると同時に、マウント解除が進行します。この引数を非共有ファイルシステムに指定した場合、またはホストが共有ファイルシステムのメタデータサーバーでない場合、オプションは無視されます。

このフラグは、`-f` オプションと一緒に使用することもできます。その場合、ソフトウェアはマウント解除を強制する前に、指定された時間だけ待機します。

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

Sun StorageTek QFS 共有ファイルシステムの初期インストールおよび構成を行うには、『Sun StorageTek QFS インストールおよびアップグレードの手引き』の説明に従ってください。この章の多くの例では、このマニュアルで述べられているホスト名と構成情報を使用しています。

非共有 Sun StorageTek QFS ファイルシステムを Sun StorageTek QFS 共有ファイルシステムへ変換するには、先にメタデータサーバーで変換を行ってから、各クライアントで変換する必要があります。ここでは、この手順について説明します。

▼ メタデータサーバーで変換を行う

この手順を実行するためには root の権限が必要です。

1. 主メタデータサーバーとして使用するシステムに、スーパーユーザーでログインします。
2. サイトでカスタマイズしたすべてのシステムファイルと構成ファイルのバックアップを取ります。

ソフトウェアによって異なりますが、これらのファイルには、mcf、archiver.cmd、defaults.conf、samfs.cmd、または inquiry.conf が含まれる場合があります。すべてのファイルシステムのこれらのファイルのバックアップを取ります。また、/etc/opt/SUNWsamfs ディレクトリのファイル、/var/opt/SUNWsamfs ディレクトリのファイルのバックアップコピーがあることを確認してください。

3. 変更する各ファイルシステムのバックアップを取ったことを確認します。
ファイルシステムは、サイトの方針に従って定期的にバックアップを取る必要があります。ファイルシステムにすでに存在するバックアップファイルに問題がない場合は、ここで再度バックアップを取る必要はありません。
4. umount(1M) コマンドを使用して、ファイルシステムをマウント解除します。
方法については、65 ページの「ファイルシステムのマウント解除」を参照してください。

5. `samfsck(1M) -S -F family-set-name` コマンドを使用して、ファイルシステムを Sun StorageTek QFS 共有ファイルシステムに変換します。

`family-set-name` には、新しい Sun StorageTek QFS 共有ファイルシステムに変換するファイルシステムのファミリセット名を指定します。

例:

```
# samfsck -S -F sharefs1
```

6. `/etc/opt/SUNWsamfs/mcf` ファイルを編集して、ファイルシステムの「Additional Parameters」フィールドに `shared` というキーワードを追加します。

例:

コード例 4-1 共有ファイルシステム `sharefs1` の `mcf` ファイル

```
# Equipment                Eq  Eq   Family  Dev  Add
# Identifier                Ord Type Set      State Params
# -----                --- ---- -
sharefs1                    10  ma   sharefs1 on    shared
/dev/dsk/c2t50020F23000065EEd0s6 11  mm   sharefs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12  mr   sharefs1 on
/dev/dsk/c7t50020F2300006099d0s6 13  mr   sharefs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14  mr   sharefs1 on
```

7. `/etc/vfstab` ファイルを編集して、ファイルシステムの「Mount Parameters」フィールドに `shared` というキーワードを追加します。

例:

コード例 4-2 `/etc/vfstab` ファイルの例

```
# File /etc/vfstab
# FS name  FS to fsck  Mnt pt    FS type  fsck pass  Mt@boot  Mt params
sharefs1  -           /sharefs1 samfs    -         no        shared
```

8. /etc/opt/SUNWsamfs/hosts.*fsname* ホスト構成ファイルを作成します。

例:

コード例 4-3 Sun StorageTek QFS 共有ファイルシステムのホストファイル例

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IPServerNotServer
# Name AddressesPriorityUsedHost
# -----
titan titan-ge0 1-server
tethys tethys-ge02-server
```

ホスト構成ファイルの作成方法についての詳細は、『Sun StorageTek QFS インストールおよびアップグレードの手引き』を参照してください。

9. samsharefs(1M) `-u -R family-set-name` コマンドを実行して、ファイルシステムとホスト構成を初期化します。

例:

```
# samsharefs -u -R sharefs1
```

注 - このコマンドでエラーメッセージが表示されることがありますが、無視してかまいません。

10. samd(1M) `config` コマンドを実行します。

```
# samd config
```

これで、sam-fsd デーモンに構成変更を知らせます。

11. mount(1M) コマンドを実行して、ファイルシステムをマウントします。

▼ 各クライアントで変換を行う

1. mkdir(1) コマンドを使用して、ファイルシステムのマウントポイントを作成します。

例:

```
# mkdir /sharefs1
```

2. (省略可能) /etc/opt/SUNWsamfs/hosts.*file-system-name*.local ローカルホスト構成ファイルを作成します。

Sun StorageTek QFS 共有ホストシステムに複数のホストインタフェースがある場合は、この操作を実行するとよいでしょう。ローカルホストの構成ファイルでは、ファイルシステムのアクセス時にメタデータサーバーとクライアントホストで使用できるホストインタフェースを定義します。このファイルを使用して、環境内の共有ネットワークとプライベートネットワークにおけるファイルシステムのトラフィックを指定します。

コード例 4-4 に、ローカルホスト構成ファイルの例を示します。

コード例 4-4 ファイル hosts.sharefs1.local

```
# This is file /etc/opt/SUNWsamfs/hosts.sharefs1.local
# Host Name      Host Interfaces
# -----
titan            172.16.0.129
tethys          172.16.0.130
```

ローカルホストファイル作成の詳細については、90 ページの「ローカルホストの構成ファイルの作成」を参照してください。

3. 既存の Sun StorageTek QFS ファイルシステムから新しい Sun StorageTek QFS 共有ファイルシステムにファイルを移動する場合は、変更される各ファイルシステムのバックアップがあることを確認してください。

ファイルシステムは、サイトの方針に従って定期的にバックアップを取る必要があります。ファイルシステムにすでに存在するバックアップファイルに問題がない場合は、ここで再度バックアップを取る必要はありません。

4. umount(1M) コマンドを使用して、ファイルシステムをマウント解除します。

方法については、65 ページの「ファイルシステムのマウント解除」を参照してください。

5. /etc/vfstab ファイルを編集して、ファイルシステムの「Mount Parameters」フィールドに shared というキーワードを追加します。

例:

```
# File /etc/vfstab
# FS name    FS to fsck    Mnt pt      FS type    fsck pass    Mt@boot    Mt params
sharefs1    -            /sharefs1   samfs       -            no          shared
```

6. /etc/opt/SUNWsamfs/hosts.*fsname* ホスト構成ファイルを作成します。
コード例 4-5 に例を示します。

コード例 4-5 Sun StorageTek QFS 共有ファイルシステムのホストファイル例

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IPServerNotServer
# Name AddressesPriorityUsedHost
# -----
titan titan-ge0 1- server
tethys tethys-ge0 2- server
```

ホスト構成ファイルの作成方法についての詳細は、『Sun StorageTek QFS インストールおよびアップグレードの手引き』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

Sun StorageTek QFS 共有ファイルシステムを非共有 Sun StorageTek QFS ファイルシステムへ変換するには、先に各クライアントで変換を行ってから、メタデータサーバーで変換する必要があります。ここでは、この手順について説明します。

▼ 各クライアントで変換を行う

1. `umount(1M)` コマンドを使用して、ファイルシステムをマウント解除します。
方法については、65 ページの「ファイルシステムのマウント解除」を参照してください。
2. /etc/opt/SUNWsamfs/mcf ファイルからファイルシステムのエントリを削除します。
3. /etc/vfstab ファイルからファイルシステムのエントリを削除します。
4. `samd(1M) config` コマンドを実行します。

```
# samd config
```

これで、`sam-fsd` デーモンに構成変更を知らせます。

5. ファイルシステムのマウントポイントを削除します。

▼ サーバーで変換を行う

この手順を実行するためには root の権限が必要です。

1. スーパーユーザーとして、メタデータサーバーシステムにログインします。
2. サイトでカスタマイズしたすべてのシステムファイルと構成ファイルのバックアップを取ります。

ソフトウェアによって異なりますが、ファイルは、mcf(4)、archiver.cmd、defaults.conf、samfs.cmd、inquiry.conf などです。すべてのファイルシステムのこれらのファイルのバックアップを取ります。また、/etc/opt/SUNWsamfs ディレクトリのファイル、/var/opt/SUNWsamfs ディレクトリのファイルのバックアップコピーがあることを確認してください。

3. 既存の Sun StorageTek QFS 共有ファイルシステムから新しい Sun StorageTek QFS ファイルシステムにファイルを移動する場合は、変更される各ファイルシステムのバックアップがあることを確認してください。

ファイルシステムは、サイトの方針に従って定期的にバックアップを取る必要があります。これについては、インストール手順の最後の操作で説明しています。ファイルシステムにすでに存在するバックアップファイルに問題がない場合は、ここで再度バックアップを取る必要はありません。

4. `umount(1M)` コマンドを使用して、ファイルシステムをマウント解除します。
方法については、65 ページの「ファイルシステムのマウント解除」を参照してください。
5. `samfsck(1M) -F -U file-system-name` を実行して、Sun StorageTek QFS 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。

file-system-name には、新しい非共有ファイルシステムに変換する Sun StorageTek QFS 共有ファイルシステムの名前を指定します。

例:

```
# samfsck -F -U samfs1
```

6. /etc/opt/SUNWsamfs/mcf ファイルを編集して、ファイルシステムの「Additional Parameters」フィールドから shared というキーワードを削除します。

例:

```
# Equipment          Eq Eq  Family  Dev  Add
# Identifier         Ord Type Set    State Params
# -----
samfs1              10 ma   samfs1  on
/dev/dsk/c2t50020F23000065EE0s6 11 mm   samfs1  on
/dev/dsk/c7t50020F2300005D22d0s6 12 mr   samfs1  on
/dev/dsk/c7t50020F2300006099d0s6 13 mr   samfs1  on
/dev/dsk/c7t50020F230000651Cd0s6 14 mr   samfs1  on
```

7. /etc/vfstab ファイルを編集して、ファイルシステムの「Mount Parameters」フィールドから shared というキーワードを削除します。

例:

```
# File /etc/vfstab
# FS name  FS to fsck  Mnt pt  FS type  fsck pass  Mt@boot  Mt params
samfs1    -           /samfs1  samfs    -          no
```

8. /etc/opt/SUNWsamfs/hosts.*file-system-name* 構成ファイルを削除します。
9. samd(1M) config コマンドを実行します。

```
# samd config
```

これで、sam-fsd デーモンに構成変更を知らせます。

10. mount(1M) コマンドを実行して、ファイルシステムをマウントします。

クライアントホストの追加または削除

次の項では、Sun StorageTek QFS 共有ファイルシステムのクライアントホストシステムを追加および削除する方法について説明します。

- 81 ページの「クライアントホストを追加する」
- 86 ページの「クライアントホストを削除する」
- 88 ページの「Sun StorageTek QFS 共有環境での mcf ファイルの更新」
- 90 ページの「ローカルホストの構成ファイルの作成」

▼ クライアントホストを追加する

すべての参加ホストでファイルシステムの構成とマウントが終了すると、Sun StorageTek QFS 共有ファイルシステムにクライアントホストを追加できます。Sun Cluster 環境のノードとなっているクライアントホストを追加する場合は、そのノードをクラスタの既存のリソースグループに追加する必要があります。詳細については、『Sun Cluster System Administration Guide for Solaris OS』を参照してください。

1. メタデータサーバーでスーパーユーザーになります。
2. `samsharefs(1M)` コマンドを使用して、現在の Sun StorageTek QFS 共有ファイルシステムの情報を取り出し、編集可能なファイルに書き込みます。
 - Sun StorageTek QFS 共有ファイルシステムがマウントされている場合は、現在のメタデータサーバーで `samsharefs(1M)` コマンドを実行します。

例:

```
# samsharefs sharefs1 > /etc/opt/SUNWsamfs/hosts.sharefs1
```

- Sun StorageTek QFS 共有ファイルシステムがマウント解除されている場合は、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のいずれかから、`-R` オプションを付けて `samsharefs(1M)` コマンドを実行します。

例:

```
# samsharefs -R sharefs1 > /etc/opt/SUNWsamfs/hosts.sharefs1
```

`samsharefs(1M)` コマンドは、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として構成されているクライアントホストでのみ実行できます。詳細は、`samsharefs(1M)` のマニュアルページを参照してください。

3. vi(1) などのエディタを使用して、Sun StorageTek QFS 共有ファイルシステムの情報ファイルを開きます。

コード例 4-6 にこの手順を示します。

コード例 4-6 編集前の hosts.sharefs1

```
# vi /etc/opt/SUNWsamfs/hosts.sharefs1
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IPServerNotServer
# Name AddressesPriorityUsedHost
# -----
titan 172.16.0.129 1-server
tethys 172.16.0.130 2-
mimas mimas --
dione dione --
```

4. エディタを使用して、新しいクライアントホストのための行を追加します。

コード例 4-7 は、helene の行を最終行として追加したあとのファイルを示しています。

コード例 4-7 編集後の hosts.sharefs1

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IPServerNotServer
# Name AddressesPriorityUsedHost
# -----
titan 172.16.0.129 1-server
tethys 172.16.0.130 2-
mimas mimas - -
dione dione - -
helene helene - -
```

5. samsharefs(1M) コマンドを使用して、バイナリファイルの現在の情報を更新します。

このコマンドで使用するオプションやこのコマンドを実行するシステムは、Sun StorageTek QFS 共有ファイルシステムがマウントされているかどうかによって次のように異なります。

- ファイルシステムがマウントされている場合は、現在のメタデータサーバーで samsharefs(1M) -u コマンドを実行します。

例:

```
# samsharefs -u sharefs1
```

- ファイルシステムがマウント解除されている場合は、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のどれかから、`samsharefs(1M) -R -u` コマンドを実行します。

例:

```
# samsharefs -R -u sharefs1
```

クライアントホスト `helene` が認識されるようになりました。

6. スーパーユーザーで、追加するクライアントホストにログインします。
7. `format(1M)` コマンドを使用して、クライアントホストディスクの存在を確認します。
8. クライアントホストの `mcf` ファイルを更新します。

ホストシステムから共有ファイルシステムへのアクセス、または共有ファイルシステムのマウントを実行できるようにするには、前もってその共有ファイルシステムを `mcf` ファイルの中で定義してください。Sun StorageTek QFS 共有ファイルシステム内のすべてのクライアントホストと一致するよう、`mcf` ファイルを更新してください。ファイルシステムとディスクの宣言情報では、ファミリセット名、装置番号、および装置タイプのデータがメタデータサーバー上の構成と同じである必要があります。クライアントホスト上の `mcf` ファイルにも `shared` キーワードを組み込んでください。ただし、コントローラの割り当てがホストごとに異なることがあるため、装置名を変更できます。

`mcf` ファイルの編集方法については、88 ページの「Sun StorageTek QFS 共有環境での `mcf` ファイルの更新」を参照してください。

9. メタデータサーバーホストで `samd(1M) config` コマンドを実行します。

```
# samd config
```

これによって、`sam-fsd` デーモンに構成変更を知らせます。

10. (省略可能) 新しいクライアントホストにローカルホスト構成ファイルを作成します。

Sun StorageTek QFS 共有ホストシステムに複数のホストインタフェースがある場合は、この手順を実行するとよいでしょう。ローカルホストの構成ファイルでは、ファイルシステムのアクセス時にメタデータサーバーとクライアントホストで使用できるホストインタフェースを定義します。このファイルを使用して、環境内の共有ネットワークとプライベートネットワークにおけるファイルシステムのトラフィックを指定します。

ローカルホストファイル作成の詳細は、90 ページの「ローカルホストの構成ファイルの作成」を参照してください。

11. クライアントホストで **samd(1M) config** コマンドを実行します。

```
# samd config
```

これで、sam-fsd デーモンに構成変更を知らせます。

12. このファイルシステムで、sam-sharefsd デーモンが動作していることを確認します。

それには、コード例 4-8 に示す ps(1) および grep(1) コマンドを使用します。

コード例 4-8 ps(1) コマンドの出力

```
# ps -ef | grep sam-sharefsd  
root 26167 26158 0 18:35:20 ? 0:00 sam-sharefsd sharefs1  
root 27808 27018 0 10:48:46 pts/21 0:00 grep sam-sharefsd
```

コード例 4-8 は、sharefs1 ファイルシステムで sam-sharefsd デーモンがアクティブなことを示しています。システムに戻された出力で、sam-sharefsd デーモンが Sun StorageTek QFS 共有ファイルシステムでアクティブでないと示された場合は、233 ページの「共有ファイルシステムでの失敗またはハングアップした sammkfs(1M) コマンドまたは mount(1M) コマンドのトラブルシューティング」で説明するいくつかの診断手順を実行します。

13. 新しい Sun StorageTek QFS 共有ファイルシステムにあらかじめマウントポイントがない場合は、mkdir(1) コマンドを使用してマウントポイントのディレクトリを作成します。

例:

```
# mkdir /sharefs1
```

14. chmod(1M) コマンドを実行して、マウントポイントのアクセス権セットを 755 に設定します。

例:

```
# chmod 755 /sharefs1
```

アクセス権は、すべての参加ホストで同一である必要があります。マウント後にファイルシステムを使用するには、ユーザーがマウントポイントに対して実行権を持つ必要があるため、初期のアクセス権セットとして 755 を推奨します。ファイルシステムをマウントすると、root ディレクトリのアクセス権によって、この設定は無効になります。

15. /etc/vfstab ファイルを変更します。

/etc/vfstab ファイルに、Sun StorageTek QFS 共有ファイルシステムのエントリを指定してください。「Mount Parameters」フィールドに shared を指定します。さらに、次のいずれかを実行します。

- 起動時にこのシステムがマウントされないようにする場合は、Mt@boot フィールドに no と入力します。
- 起動時に Sun StorageTek QFS 共有ファイルシステムを自動的にマウントする場合は、次のことを行います。
 - Mt@boot フィールドに yes と入力する。
 - Mt params フィールドに bg マウントオプションを追加する。bg マウントオプションを使用すると、メタデータサーバーが応答しない場合に、ファイルシステムがバックグラウンドでマウントされます。

コード例 4-9 に、Mt params フィールドの shared および bg エントリを示します。

コード例 4-9 /etc/vfstab ファイルの例

```
# File /etc/vfstab
# FS name  FS to fsck  Mnt pt    FS type   fsck  Mt@boot  Mt params
#                                     pass
sharefs1  -              /sharefs1 samfs     -      yes      shared,bg
```

16. メタデータサーバーで df(1M) コマンドを実行して、ファイルシステムがメタデータサーバーにマウントされていることを確認します。

例:

```
# df -k
```

ファイルシステムが、表示されたリストに含まれているはずです。

17. クライアントホストから mount(1M) コマンドを実行して、Sun StorageTek QFS 共有ファイルシステムをマウントします。

例:

```
# mount /sharefs1
```

Sun StorageTek QFS 共有ファイルシステムのマウントの詳細については、253 ページの「Sun StorageTek QFS 共有ファイルシステムのマウントオプション」または mount_samfs(1M) のマニュアルページを参照してください。

▼ クライアントホストを削除する

1. メタデータサーバーおよびすべてのクライアントホストで、スーパーユーザーになります。

注 - `samsharefs(1M)` コマンドを使用すると、メタデータサーバーまたはクライアントホストに実際にログインしていることを確認できます。

2. `umount(1M)` コマンドを使用して、Sun StorageTek QFS 共有ファイルシステムがマウントされた各クライアントホスト上で Sun StorageTek QFS 共有ファイルシステムをマウント解除します。

例:

```
client# umount sharefs1
```

3. `umount(1M)` コマンドを使用して、Sun StorageTek QFS 共有ファイルシステムをメタデータサーバーからマウント解除します。

例:

```
metaserver# umount sharefs1
```

4. まだログインしていない場合は、Sun StorageTek QFS 共有ファイルシステムのメタデータサーバーに、スーパーユーザーでログインします。

5. `samsharefs(1M)` コマンドを使用して、現在の構成情報を取得します。

次の例のコマンドによって、現在の構成情報がファイル `/etc/opt/SUNWsamfs/hosts.sharefs1` に書き込まれます。

```
# samsharefs -R sharefs1 > /etc/opt/SUNWsamfs/hosts.sharefs1
```

6. vi(1) などのエディタを使用して、Sun StorageTek QFS 共有ファイルシステムの情報ファイルを開きます。

コード例 4-10 に、クライアントホストを削除前のファイルを示します。

コード例 4-10 クライアントホストを削除する前の hosts.sharefs1

```
# vi /etc/opt/SUNWsamfs/hosts.sharefs1
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IPServerNotServer
# Name AddressesPriorityUsedHost
# -----
titan 172.16.0.129 1-server
tethys 172.16.0.130 2-
mimas mimas - -
dione dione - -
helene helene --
```

7. エディタを使用して、サポートされなくなったクライアントホストを削除します。

コード例 4-11 は、helene の行を削除したあとのファイルを示しています。

コード例 4-11 クライアントホストを削除したあとの hosts.sharefs1

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IPServerNotServer
# Name AddressesPriorityUsedHost
# -----
titan 172.16.0.129 1-server
tethys 172.16.0.130 2-
mimas mimas - -
dione dione - -
```

8. samsharefs(1M) -R -u コマンドを使用して、現在のホストの情報を更新します。

例:

```
# samsharefs -R -u sharefs1
```

ホスト helene が削除されました。

9. samsharefs(1M) -R コマンドを使用して、現在の構成を表示します。

例:

```
# samsharefs -R sharefs1
```

10. `mount(1M)` コマンドを使用して、Sun StorageTek QFS 共有ファイルシステムをメタデータサーバーにマウントし、次にファイルシステム内の各クライアントホストにマウントします。

`mount(1M)` コマンドについては、`mount_samfs(1M)` のマニュアルページを参照してください。

Sun StorageTek QFS 共有環境での `mcf` ファイルの更新

`samfsconfig(1M)` コマンドによって構成情報が生成されます。この構成情報は、Sun StorageTek QFS 共有ファイルシステムに組み込む装置を指定するときに役立ちます。そのあと、この情報を使用して各クライアントホストの `mcf` ファイルを更新できます。

各クライアントホストごとに `samfsconfig(1M)` コマンドを個別に入力します。コントローラの番号は各クライアントホストによって割り当てられるため、メタデータサーバーと異なるコントローラの番号になる場合があることに注意してください。

注 – Sun StorageTek QFS 共有ファイルシステムのマウント後にメタデータサーバーの `mcf` ファイルを更新する場合は、その共有ファイルシステムにアクセス可能なすべてのホストの `mcf` ファイルを必ず更新してください。

例 1: コード例 4-12 では、`samfsconfig(1M)` コマンドを使用して、クライアント `tethys` のファミリーセット `sharefs1` についてのデバイス情報を検出する方法を示します。`tethys` は潜在的なメタデータサーバーのため、共有ファイルシステム内のもう 1 つのメタデータサーバーである `titan` と同じメタデータディスクに接続されています。

コード例 4-12 `tethys` に対する `samfsconfig(1M)` コマンドの例

```
tethys# samfsconfig /dev/dsk/*
#
# Family Set 'sharefs1' Created Wed Jun 27 19:33:50 2003
#
sharefs1                10 ma sharefs1 on shared
/dev/dsk/c2t50020F23000065EE0s6 11 mm sharefs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12 mr sharefs1 on
/dev/dsk/c7t50020F2300006099d0s6 13 mr sharefs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14 mr sharefs1 on
```

クライアントホスト `tethys` の `mcf` ファイルに、`samfsconfig(1M)` コマンドの出力の最後の 5 行をコピーします。次の内容を確認します。

- 各 Device State フィールドが `on` に設定されていること。

- shared キーワードが、ファイルシステム名の Additional Parameters フィールドに指定されていること。

コード例 4-13 に、編集後の mcf ファイルを示します。

コード例 4-13 sharefs1 のクライアントホスト tethys の mcf ファイル

```
# Equipment                               Eq Eq   Family   Dev   Add
# Identifier                               Ord Type Set     State Params
# -----
sharefs1                                  10 ma   sharefs1 on    shared
/dev/dsk/c2t50020F23000065EE0s6         11 mm   sharefs1 on
/dev/dsk/c7t50020F2300005D22d0s6         12 mr   sharefs1 on
/dev/dsk/c7t50020F2300006099d0s6         13 mr   sharefs1 on
/dev/dsk/c7t50020F230000651Cd0s6         14 mr   sharefs1 on
```

例 2: コード例 4-14 では、samfsconfig(1M) コマンドを使用して、クライアントホスト mimas のファミリセット sharefs1 についてのデバイス情報を検出する方法を示します。この例で、mimas はメタデータサーバーにできず、メタデータディスクに接続していません。

コード例 4-14 mimas に対する samfsconfig(1M) コマンドの例

```
mimas# samfsconfig /dev/dsk/*
#
# Family Set 'sharefs1' Created Wed Jun 27 19:33:50 2001
#
# Missing slices
# Ordinal 0
# /dev/dsk/c1t50020F2300005D22d0s6      12    mr   sharefs1   on
# /dev/dsk/c1t50020F2300006099d0s6      13    mr   sharefs1   on
# /dev/dsk/c1t50020F230000651Cd0s6      14    mr   sharefs1   on
```

mimas に対する samfsconfig(1M) コマンドの出力では、メタデータディスクに相当する Ordinal 0 がないことに注意してください。存在しない装置に対して、samfsconfig(1M) プロセスではファイルシステムのその要素はコメント化され、ファイルシステムのファミリセット宣言の行は省略されます。mcf ファイルを次のように編集してください。

- クライアントホスト mimas の mcf ファイルに、sharefs1 で始まるファイルシステムのファミリセット宣言の行を作成します。ファイルシステムのファミリセット宣言の行の Additional Parameters フィールドに shared キーワードを入力する。
- 存在しない装置番号のエントリごとに 1 つまたは複数の nodev 行を作成します。これらの行では、アクセスできない装置の「Equipment Identifier」フィールドに nodev キーワードを指定する必要があります。
- 各 Device State フィールドが on に設定されていることを確認する

■ デバイス行のコメントを解除する

コード例 4-15 に、mimas の編集後の mcf ファイルを示します。

コード例 4-15 クライアントホスト mimas の mcf ファイル

```
# The mcf File For mimas
# Equipment                      Eq  Eq  Family  Device Addl
# Identifier                      Ord Type Set      State  Params
-----
sharefs1                          10  ma   sharefs1 on      shared
nodev                              11  mm   sharefs1 on
/dev/dsk/c1t50020F2300005D22d0s6  12  mr   sharefs1 on
/dev/dsk/c1t50020F2300006099d0s6  13  mr   sharefs1 on
/dev/dsk/c1t50020F230000651Cd0s6  14  mr   sharefs1 on
```

ローカルホストの構成ファイルの作成

ローカルホストの構成ファイルは次の場所に常駐する必要があります。

```
/etc/opt/SUNWsamfs/hosts.family-set-name.local
```

ローカルホストの構成ファイルにはコメントを指定できます。コメント行は先頭にハッシュ記号 (#) を付ける必要があります。ハッシュ記号より右側の文字は無視されます。

表 4-1 に、ローカルホストの構成ファイルのフィールドを示します。

表 4-1 ローカルホストの構成ファイルのフィールド

フィールド	内容
Host Name	このフィールドには、Sun StorageTek QFS 共有ファイルシステムに含まれる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の名前 (英数字) を指定します。

表 4-1 ローカルホストの構成ファイルのフィールド (続き)

フィールド	内容
Host Interfaces	<p>このフィールドには、ホストインタフェースアドレスをコンマで区切って指定します。このフィールドは、<code>ifconfig(1M) -a</code> コマンドの出力から作成できます。次のどれかの方法で個々のインタフェースを指定できます。</p> <ul style="list-style-type: none"> ドット付き 10 進数 (dotted-decimal) の IP アドレス形式 IP version 6 の 16 進数のアドレス形式 ローカルのドメイン名サービス (DNS) が特定のホストインタフェースに対してに解決処理するシンボリック名 <p>各ホストは、ホストが指定のホストインタフェースに接続をしようとするかどうかを、このフィールドを使用して決定します。システムはアドレスを左から右の順に評価し、リスト内の最初に応答したアドレスを使用して接続されます (このアドレスは共有ホストファイルにも含まれます)。</p>

Sun StorageTek QFS 共有ファイルシステムでは、各クライアントホストは、メタデータサーバー IP アドレスのリストをメタデータサーバーホストから取得します。

メタデータサーバーとクライアントホストは、メタデータサーバー上の `/etc/opt/SUNWsamfs/hosts.fsname` ファイルと各クライアントホスト (存在する場合) 上の `hosts.fsname.local` ファイルの両方を使用して、ファイルシステムにアクセスするときに使用するホストインタフェースを判別します。このプロセスは次のとおりです (「ネットワーククライアント」という意味の「クライアント」が、クライアントホストとメタデータサーバーホストの両方を指すために使用されることに注意してください)。

- クライアントは、ファイルシステムのディスク上のホストファイルからメタデータサーバーホストの IP インタフェースのリストを取得します。
このファイルを確認するには、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 `samsharefs(1M)` コマンドを実行します。
- クライアントは `hosts.fsname.local` ファイルに対する自分のファイルを検索します。
- 検索結果によって、次のどれかの処理を実行します。
 - `hosts.fsname.local` ファイルが存在しない場合、クライアントは成功するまで、システムホスト構成ファイルの各アドレスに接続をしようとします。
 - `hosts.fsname.local` ファイルが存在する場合、クライアントは次のことを行います。
 - メタデータサーバーの `/etc/opt/SUNWsamfs/hosts.fsname` ファイルと、`hosts.fsname.local` ファイルの両方で、メタデータサーバーのアドレスリストを比較します。
 - 両方のファイルに含まれるアドレスのリストを作成し、成功するまでこれらのアドレスに接続をしようとします。アドレスの順序が 2 つのファイルで異なる場合、クライアントは `hosts.fsname.local` ファイルの順序を使用します。

例: コード例 4-16 は 4 つのホストを示すホストファイルの例です。

コード例 4-16 Sun StorageTek QFS 共有ファイルシステムのホストファイル例

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IP Server NotServer
# Name Addresses Priority Used Host
# -----
titan 172.16.0.129 1-server
tethys 172.16.0.130 2-
mimas mimas - -
dione dione - -
```

図 4-1 に、これらのシステムのインタフェースを示します。

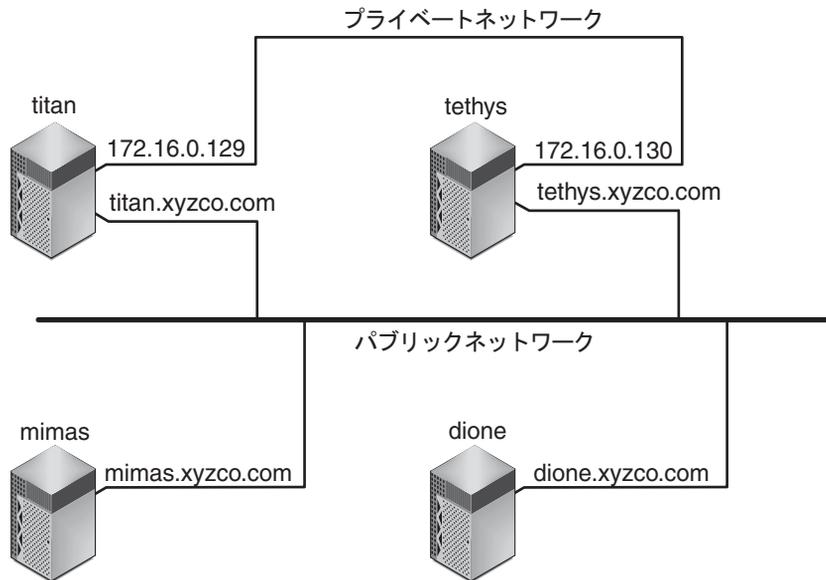


図 4-1 ネットワークインタフェース

システム titan および tethys は、インタフェース 172.16.0.129 および 172.16.0.130 でプライベートネットワーク接続を共有します。titan および tethys が常にプライベートネットワーク接続で通信できるようにするために、シス

システム管理者は、各システムに /etc/opt/SUNWsamfs/hosts.sharefs1.local の同一コピーを作成しています。コード例 4-17 は、これらのファイルの内容を示しています。

コード例 4-17 titan と tethys の hosts.sharefs1.local ファイル

```
# This is file /etc/opt/SUNWsamfs/hosts.sharefs1.local
# Host Name      Host Interfaces
# -----      -
titan           172.16.0.129
tethys          172.16.0.130
```

システム mimas および dione は、プライベートネットワーク上にはありません。titan と tethys の共有インタフェースで titan と tethys に常に接続されることを保証するため、システム管理者は /etc/opt/SUNWsamfs/hosts.sharefs1.local と同一のコピーを mimas と dione に作成します。コード例 4-18 に、このファイルの情報を示します。

コード例 4-18 mimas と dione の hosts.sharefs1.local ファイル

```
# This is file /etc/opt/SUNWsamfs/hosts.sharefs1.local
# Host Name      Host Interfaces
# -----      -
titan           titan
tethys          tethys
```

Sun StorageTek QFS 環境でのメタデータサーバーの変更

この節では、Sun Cluster ソフトウェアなどのソフトウェアパッケージの自動 Membership Services 機能を使用しないで、Sun StorageTek QFS 共有ファイルシステムでメタデータサーバーとして機能するホストを変更する方法について説明します。

メタデータサーバーシステムは、次の状況のときに手動で変更できます。

- メタデータサーバーが使用不可になった。
- 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を変更したい。

メタデータサーバーを正常に変更するには、既存のメタデータサーバーのマウントポイントと、すべての潜在的なメタデータサーバーが等しい必要があります。

変更の実行時に既存のメタデータサーバーが使用可能かどうかによって、次のどちらかの手順を選択します。

- 94 ページの「使用可能なメタデータサーバーを変更する」
- 94 ページの「使用不可のメタデータサーバーを変更する」

▼ 使用可能なメタデータサーバーを変更する

- 既存のメタデータサーバーで `samsharefs(1M) -s` コマンドを実行して、新しいメタデータサーバーを宣言します。

例:

```
titan# samsharefs -s tethys sharefs1
```

注 – Sun SAM-QFS 環境では、このコマンドを実行する前に、メタデータサーバー上でのすべてのアーカイブ動作を停止するようにします。

▼ 使用不可のメタデータサーバーを変更する

共有ファイルシステムのメタデータサーバーに障害が発生した場合、メタデータサーバーの変更は、必ずメタデータサーバーを再起動したあとで行う方が安全です。そうでない場合は、再起動前にサーバーが入出力が発生し得ないことを確認します。次の方法はいずれも、サーバーを停止する目的では使用しないでください。ファイルシステムが破壊される可能性があります。

- L1-A キーシーケンスの実行
- 別のホストへの強制的なフェイルオーバー
- それまでのメタデータサーバーへの `go` (続行) コマンドの実行、ダンプファイルの要求、または `sync` コマンドの実行

同様に、メタデータサーバーでパニックが発生し、カーネルが `adb(1)` になった場合は、メタデータサーバーを変更してからサーバーで `a :c` (続行) コマンドを実行するのを避けてください。この手順を行うと、それまでのメタデータサーバーが無効なバッファーを、新しいアクティブなファイルシステムに書き込んでしまいます。

メタデータサーバーを変更するには、次の手順を使用します。

1. 再起動しないと既存のメタデータサーバーが起動できないことを確認します。

特に、サーバーの電源が切断されている、再起動されている、停止されている、あるいはメタデータディスクから切断されていることを確認します。目的は、以前のメタデータサーバーを停止して、すべてのバッファをフラッシュまたは破棄する (または、書き込みできないようにする) ことです。

コード例 4-19 は、kadb プロンプトでのキーシーケンスを示しています。

コード例 4-19 メタデータサーバーが再起動できないようにするための kadb プロンプトでのキーシーケンス

```
kadb[1]: sync # Forces a dump
kadb[1]: $q # Exits the debugger for prom
```

コード例 4-20 は、PROM プロンプトでのキーシーケンスを示しています。

コード例 4-20 メタデータサーバーが再起動できないようにするための PROM プロンプトでのキーシーケンス

```
{0} > sync            # Forces the buffers out
{0} > boot args        # Discards buffers
```

args には、*-r* や *-v* など *boot(1M)* コマンドの引数を指定します。詳細は、*boot(1M)* のマニュアルページを参照してください。

2. 新しい潜在的なメタデータサーバーで、少なくとも最大リース時間だけ待機してから、*samsharefs(1M)* コマンドを実行します。

例:

```
# samsharefs -R -s tethys sharefs1
```

すべてのクライアントリースが期限切れになってから *samsharefs(1M)* コマンドを実行するよう、待機します。リース時間が期限切れかどうかわからない場合は、*samu(1M) N* の表示を使用します。*samu(1M)* については、261 ページの「*samu(1M)* オペレータユーティリティの使用法」を参照してください。リースとその期間の詳細については、255 ページの「Sun StorageTek QFS 共有ファイルシステムでのリースの使用: *rdlease=n*, *wrlease=n*, および *aplease=n* オプション」を参照してください。



注意 – マウント済みファイルシステムで *samsharefs(1M)* コマンドの *-R* オプションを使用してメタデータサーバーホストを変更する場合、まず、アクティブなメタデータサーバーを停止して無効にしてから切断する必要があります。このようにしないと、ファイルシステムが破壊されることがあります。

3. (省略可能) ファイルシステムをマウント解除します。

この手順は、ファイルシステムの検査を実行する場合にのみ実行します。

66 ページの「Sun StorageTek QFS 共有ファイルシステムをマウント解除する」の手順を使用します。

4. (省略可能) samfsck(1M) コマンドを実行して、ファイルシステムの検査を実行します。

Sun StorageTek QFS 共有ファイルシステムのメタデータサーバーに障害が発生した場合は、サーバーを再起動し、すべてのクライアントでファイルシステムをマウント解除してから、`samfsck(1M)` を実行する必要があります。サーバーとクライアントは、ファイルの長さが変化する前にブロックを事前割り当てします。`samfsck(1M)` コマンドは余分なブロックの割り当てられたファイルを整理しますが、このような余分なブロックにデータが含まれていることがあります。このような整理済みのファイルがクライアントからのサイズ更新を待っている場合、クライアントが続行しようとするすると、ファイルからこれらのブロックが失われます。その結果、ファイルからデータが失われ、失われたデータはゼロとして読み込まれます。

SAM-QFS 環境でのメタデータサーバーの変更

この節では、Sun Cluster ソフトウェアなどのソフトウェアパッケージの自動 Membership Services 機能を使用しないで、SAM-QFS 共有ファイルシステムでメタデータサーバーとして機能するホストを変更する方法について説明します。

メタデータサーバーシステムは、次の状況のときに手動で変更できます。

- メタデータサーバーが使用不可になった。
- 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を変更したい。

メタデータサーバーを正常に変更するには、既存のメタデータサーバーのマウントポイントと、すべての潜在的なメタデータサーバーが等しい必要があります。

▼ SAM-QFS 環境でメタデータサーバーを変更する

Sun StorageTek SAM は、常に 1 台のホスト上だけで実行することができます。この手順では、転送時に両方のシステムが起動していることを前提にしています。例では、Sun StorageTek SAM アーカイブ機能を、ホスト A からホスト B に移します。

この手順を実行する前に、ホスト B がホスト A からのロボットカタログにアクセスできることを確認します。archiver.cmd ファイル、mcf ファイル、stager.cmd ファイル、およびその他の構成ファイルは、ホスト A 上のものと同じにしてください。

1. 次の手順に従い、ホスト A の Sun StorageTek SAM アーカイブプロセスを休止状態にします。
 - a. `samcmd aridle` と `samcmd stidle` コマンドを実行して、ホスト A でのアーカイブと書き込み処理を停止させます。

これらのコマンドは、実行中のアーカイブと書き込み処理が完了するまで待ちますが、新たな処理を開始させません。
 - b. ホスト A のすべてのテープドライブを休止状態にします。

`samcmd eq idle` コマンドで休止状態にできます。`eq` は、ドライブの装置番号です。このコマンドは、実行中の入出力が完了したあと、ドライブをオフの状態にします。
 - c. アーカイバとステージャーが休止状態になり、テープドライブがすべてオフの状態になったら、`samd stop` コマンドを実行してロボットとテープに関連するすべてのデーモンを停止します。
 - d. リサイクラを実行する `cron` ジョブがある場合は、`cron` タブからこの記述を削除して、リサイクラが現在実行中でないことを確認します。

この時点で、Sun StorageTek SAM は停止しており、ホスト B へのファイルシステムのフェイルオーバーを実行できます。

2. ホスト B で `samd config` コマンドを実行して、ホスト B の Sun StorageTek SAM を起動します。

ホスト B の起動によって、`sam-fsd` とそのサブプロセス (アーカイバやステージャーなど) は再構成され、構成ファイルの再読み込みを行います。また、`sam-amld` とテープライブラリ関連のデーモンも起動します。このとき、書き込みを待っているすべての Sun StorageTek QFS 共有クライアントアプリケーションは、書き込み要求を再発行する必要があります。

これまでの手順で、ホスト B は、すべての Sun StorageTek QFS ファイルシステムの Sun StorageTek SAM サーバーおよびメタデータサーバーとして完全に機能しています。

Sun StorageTek QFS 共有ファイルシステムでのクライアントサーバー通信

Sun StorageTek QFS 共有ファイルシステムの動作は、割り込み可能なハード接続の動作です。各クライアントは、メタデータサーバーが使用不可な場合でも、サーバーとの通信を繰り返し試行します。メタデータサーバーが応答しない場合、ユーザーは Ctrl-C を押すことで、保留中のブロックされた入出力転送を終了できます。入出力試行が中断されると、クライアントは入出力が完了するまで継続します。

システムによって、状態を説明する次のメッセージが生成されます。

```
SAM-FS:Shared server is not responding.
```

クライアントの sam-sharefsd デーモンがアクティブでない場合、またはサーバーの sam-sharefsd デーモンがアクティブでない場合にもこのメッセージが生成されます。サーバーが応答するときに、次のメッセージが生成されます。

```
SAM-FS:Shared server is responding.
```

ファイルシステムがメタデータサーバーにマウントされていないが、クライアントにマウントされている場合、システムで次のメッセージが生成されます。

```
SAM-FS:Shared server is not mounted.
```

Sun StorageTek QFS 共有ファイルシステムがサーバーにマウントされると、次のメッセージが生成されます。

```
SAM-FS:Shared server is mounted.
```

メタデータサーバーはすべてのクライアントに代わってファイル名を検索するので、メタデータサーバーの Solaris ディレクトリ名参照キャッシュ (DNLC) がデフォルトサイズのままで、パフォーマンスが低下します。クライアントが多数のファイルを頻繁に開く場合にパフォーマンスを向上するには、このキャッシュをデフォルトの2倍または3倍にします。

手順については、『Solaris カーネルのチューンアップ・リファレンスマニュアル』に説明があります。ディレクトリ名参照キャッシュのサイズを制御するパラメータは ncssize です。

第5章

ファイルシステム割り当ての管理

この章では、ファイルシステム割り当てを有効にする方法、および管理方法について説明します。この章の内容は次のとおりです。

- 99 ページの「概要」
- 102 ページの「割り当ての有効化」
- 111 ページの「割り当ての検査」
- 113 ページの「割り当ての変更と削除」

概要

ファイルシステム割り当てによって、ファイルシステム内の特定のユーザー、ユーザーグループ、または管理セットと呼ばれるサイトで決められたユーザーグループが使用できるオンラインおよび総ディスク領域の容量を制御します。

割り当ては、各ユーザーが使用できる領域の容量や i ノード数を制限することで、ファイルシステムのサイズを制御するときに役立ちます。割り当てが特に有効なのは、ユーザーのホームディレクトリを含むファイルシステムの場合です。割り当てを有効にしてから、利用状況を監視して、ニーズの変化に応じて調整できます。

ファイルシステムは、データのブロックとファイルの i ノードをユーザーに提供します。各ファイルは 1 つの i ノードを使用し、ファイルデータはディスク割り当て単位 (DAU) で格納されます。DAU のサイズは、ファイルシステムの作成時に決まります。割り当ては、512 バイトの倍数でディスクに設定します。

この節では、割り当ての使用に関する情報を説明します。

- 100 ページの「割り当てのタイプ、割り当てファイル、割り当てレコード」
- 101 ページの「弱い制限値と強い制限値」
- 101 ページの「ディスクブロックとファイル割り当て」

割り当てのタイプ、割り当てファイル、割り当てレコード

割り当ては、ユーザー ID、グループ ID、または管理者のサイト固有のグループに対して設定できます。このサイト固有のグループ化は、「管理セット ID」といいます。たとえば、管理セット ID を使用して、ファイルシステムの割り当てを適用するプロジェクトに関わるユーザーの集合を識別できます。

quota マウントオプションが有効で、ファイルシステムのルートディレクトリに 1 つまたは複数の割り当てファイルが検出された場合に、割り当ては有効になります。quota マウントオプションはデフォルトで有効です。noquota を有効にしてファイルシステムをマウントした場合、割り当ては無効になります。マウントオプションの詳細は、mount_samfs(1M) のマニュアルページを参照してください。

各割り当てファイルには一連のレコードが含まれます。レコード 0 は、システム管理者の割り当ておよびリソース使用率のためのレコードです。システム管理者の割り当ては適用されませんが、割り当てファイルの後続レコードのテンプレートとして、システム管理者のレコードを含む任意のレコードを使用できます。詳細については、109 ページの「既存の割り当てファイルを使用してユーザー、グループ、管理セットの割り当て値を有効化または変更する」を参照してください。

レコード 1 は、割り当てファイルのタイプによって異なりますが、ユーザー 1、グループ 1、または管理セット ID 1 のための割り当てファイル内のレコードです。レコード 1 およびそれ以降のすべてのレコードを編集し、ユーザーごとに異なる割り当てを設定することができます。表 5-1 に、割り当てファイル名と、それによって /root で有効になる割り当てを示します。

表 5-1 割り当てファイル名

割り当てファイル名	割り当てタイプ
.quota_u	UID (システムユーザー ID)
.quota_g	GID (システムグループ ID)
.quota_a	AID (システム管理セット ID)

ユーザーに対してデフォルトの割り当てを設定するには、割り当てファイルのレコード 0 を編集し、レコード 0 の値をほかのすべてのユーザーの初期割り当て設定として使用できるようにします。デフォルトでは、ユーザー割り当てが特に設定されない場合は、レコード 0 の値が使用されます。

割り当てファイルには、1 つあたり 128 バイトの容量が必要です。初期の 0 割り当てファイルに必要なサイズを計算するには、次の式を使用してください。

$$(highest-ID + 1) \times 128 = x$$
$$x / 4096 = 0 \text{ 割り当てファイルのサイズ}$$

弱い制限値と強い制限値

弱い制限値と強い制限値を両方設定できます。強い制限値では、使用可能なシステム資源の容量を設定します。ユーザーはこの制限値を超えることはできません。弱い制限値では、一時的に強い制限値まで超過できるシステム資源使用量のレベルを設定します。弱い制限値は、強い制限値よりも高く設定しないでください。

ユーザーが強い制限値を超えて資源を割り当てようとすると、操作は異常終了します。その場合、操作に失敗して `EDQUOT` エラーが発生します。

ユーザーが弱い制限値を超えると、タイマーが開始され、猶予期間に入ります。タイマーが作動している間、ユーザーは弱い制限値を超えて操作できます。弱い制限値を下回ると、タイマーはリセットされます。猶予期間が終わってタイマーが停止したときに、ユーザーが弱い制限値を下回っていないと、弱い制限値が強い制限値として適用されます。

たとえば、ユーザーの弱い制限値が 10,000 ブロックで、強い制限値が 12,000 ブロックであると仮定します。ユーザーのブロック使用が 10,000 ブロックを超えて、タイマーが猶予期間を過ぎると、このユーザーは、使用量が 10,000 ブロックの弱い制限値を下回らないかぎり、そのファイルシステム上にそれ以上のディスクブロックを割り当てられなくなります。

システム管理者は、`samquota(1M)` コマンドを使用してタイマー値を確認できます。`squota(1)` コマンドは、ユーザー用の `samquota(1M)` コマンドです。`squota(1)` ユーザーコマンドには、ユーザーが自分の割り当てに関する情報を得るために指定できるオプションがあります。

ディスクブロックとファイル割り当て

ユーザーは、ブロックを使用しなくても、すべて空のファイルを作成することで、i ノード割り当てを超過する可能性があります。また、ユーザーは、ユーザー割り当てのすべてのデータブロックに相当する大容量のファイルを作成することで、1 つの i ノードしか使用しなくてもブロック割り当てを超過する可能性があります。

ファイルシステム割り当ては、ユーザーが割り当てることのできる 512 バイトのブロックの数で表されます。ただし、ディスク領域は、DAU の数でユーザーファイルに割り当てられます。DAU 設定は、`sammkfs(1M)` コマンドの `-a allocation-unit` オプションを使用して指定されます。ブロック割り当ては、ファイルシステムの DAU の倍数になるように設定するとよいでしょう。このように設定しないと、ユーザーが割り当てられる最大ブロック数は、もっとも近い DAU 数に切り捨てられます。ブロック割り当ての設定方法については、108 ページの「デフォルトの割り当て値を有効にする」を参照してください。

割り当ての有効化

ファイルシステムの編集、割り当てファイルの作成、さまざまな割り当てコマンドの入力などの処理を通じて、割り当てを有効にできます。

この節では、割り当てを使用するためのファイルシステムの構成方法や、割り当てを有効にする方法について詳しく説明します。

割り当て設定のガイドライン

割り当てを有効にする前に、各ユーザーに割り当てるディスク領域の容量と i ノード数を決める必要があります。ファイルシステムの合計領域を超過しないようにする場合は、合計サイズをユーザー数で分割します。たとえば、3 ユーザーが 100M バイトのスライスを共有し、ディスク領域のニーズが同等の場合は、各ユーザーに 33M バイトを割り当てることができます。すべてのユーザーが割り当てに達する可能性の低い環境では、加算したときにファイルシステムの合計サイズを上回るように個別の割り当てを設定することもできます。たとえば、3 ユーザーが 100M バイトのスライス共有する場合は、各ユーザーに 40M バイトを割り当てることができます。

次の書式で割り当てコマンドを使用して、割り当て情報を表示できます。

- `squota(1)` コマンドは一般ユーザー用です。ユーザーが、自分の割り当て情報をユーザー、グループ、または管理セットごとに検索できる
- `samquota(1M)` コマンドはシステム管理者用です。システム管理者が割り当て情報を検索したり、割り当てを設定したりできる。ユーザー、グループ、または管理セットに対して、それぞれ `-U`、`-G`、および `-A` オプションを使用します。それをコード例 5-1 に示します。

コード例 5-1 `samquota(1M)` を使用して情報を取り出す

```
# samquota -U Janet /mount-point #Prints a user quota
# samquota -G pubs /mount-point #Prints a group quota
# samquota -A 99 /mount-point #Prints an admin set quota
```

▼ 割り当てを使用するための新しいファイルシステムを構成する

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。割り当てを使用するため既存のファイルシステムを構成するには、104 ページの「割り当てを使用するための既存ファイルシステムを構成する」を参照してください。

この手順を始める前に、`samfs.cmd` ファイルまたは `/etc/vfstab` ファイルで `noquota` マウントオプションを指定していないことを確認してください。

1. スーパーユーザーになります。
2. ファイルシステムを作成します。
『Sun StorageTek QFS インストールおよびアップグレードの手引き』に概要を示した手順に従うか、44 ページの「構成の例」の例を使用して、`mcf` ファイルの作成、マウントポイントの作成、ファイルシステムの初期化などを行います。
3. `mount(1M)` コマンドを使用して、ファイルシステムをマウントします。

例:

```
# mount /qfs1
```

4. `dd(1M)` コマンドを使用して、割り当てファイルを作成します。
このコマンドの引数は、次に示すように作成する割り当てのタイプによって異なります。
 - 管理セットの割り当てを作成するには、次のコマンドを使用します。

```
# dd if=/dev/zero of=/qfs1/.quota_a bs=4096 count=1
```

- グループの割り当てを作成するには、次のコマンドを使用します。

```
# dd if=/dev/zero of=/qfs1/.quota_g bs=4096 count=1
```

- ユーザーの割り当てを作成するには、次のコマンドを使用します。

```
# dd if=/dev/zero of=/qfs1/.quota_u bs=4096 count=1
```

`dd(1M)` コマンドについては、`dd(1M)` のマニュアルページを参照してください。

5. `umount(1M)` コマンドを使用して、割り当てファイルを作成したファイルシステムをマウント解除します。

例:

```
# umount /qfs1
```

ファイルシステムは、マウント解除する必要があります。これにより、ファイルシステムを再マウントして、マウント時に割り当てファイルを読み込ませることができません。`umount(1M)` コマンドについては、`umount(1M)` のマニュアルページを参照してください。

6. `samfsck(1M)` コマンドを使用して、ファイルシステムの検査を実行します。

次の例で、`-F` オプションは割り当てファイルで使用中の値をリセットします。

```
# samfsck -F qfs1
```

7. `mount(1M)` コマンドを使用して、ファイルシステムを再マウントします。

割り当ては、`/root` ディレクトリに 1 つまたは複数の割り当てファイルが検出された場合に有効になります。

注 `/etc/vfstab` ファイルまたは `samfs.cmd` ファイルに `quota` マウントオプションを指定する必要はありません。デフォルトで、`quota` マウントオプションは、`mount(1M)` コマンドで有効になっています。割り当ては、割り当てファイルが検出されたときに自動的に有効になります。

`mount(1M)` コマンドの詳細については、`mount_samfs(1M)` のマニュアルページを参照してください。

8. `samquota(1M)` コマンドを使用して、ユーザー、グループ、または管理セットの割り当てを設定します。

この章のこのあとの項目では、この作業の手順と例を示します。`samquota(1M)` コマンドについては、`samquota(1M)` のマニュアルページ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を構成する

この手順は、すでにファイルが存在す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。割り当てを使用するための新しいファイルシステムを構成するには、103 ページの「割り当てを使用するための新しいファイルシステムを構成する」を参照してください。

この手順を始める前に、`samfs.cmd` ファイルまたは `/etc/vfstab` ファイルで `noquota` マウントオプションを指定していないことを確認してください。

1. `su(1)` コマンドを使用して、スーパーユーザーになります。
2. `mount(1M)` コマンドを使用して `/etc/mnttab` ファイルを調べ、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。

```
# mount
```

表示されたマウントリストにファイルシステムがあることを確認します。

3. `cd(1)` コマンドを使用して、割り当てを有効にするファイルシステムのルートディレクトリに移動します。

例:

```
# cd /oldfs1
```

4. `ls(1) -a` コマンドを使用してこのディレクトリのファイルリストを表示し、このファイルシステムに割り当てが存在しないことを確認します。

`.quota_u`、`.quota_g`、`.quota_a` のいずれかのファイルが存在する場合は、このファイルシステムの割り当てが有効になっています。

ファイルシステムでいずれかの割り当てタイプが設定されている場合は、ほかの割り当てタイプをあとから設定できます。新しいファイルを追加するときに、既存の割り当てファイルを変更しないように注意してください。

5. 実施するタイプの割り当て用の割り当てファイルが存在しない場合は、`dd(1M)` コマンドを使用して割り当てファイルを作成します。

適用する割り当てのタイプについて、既存の ID 番号でもっとも高い値を確認します。初期の 0 割り当てファイルは、これらの ID のレコードを保持できるような大きさにしてください。各割り当てファイルレコードには 128 バイトが必要です。

たとえば、管理セットの割り当てを有効にするときに、ファイルシステムを使用しているもっとも大きな管理セット ID が 1024 である場合は、次のように計算します。

$$(1024 + 1) \times 128 = 131200$$

$$131200 / 4096 = 32.031\dots$$

次のコマンドを使用します。

```
# dd if=/dev/zero of=/oldfs1/.quota_a bs=4096 count=33
```

`dd(1M)` コマンドについては、`dd(1M)` のマニュアルページを参照してください。

6. `umount(1M)` コマンドを使用して、割り当てファイルを作成したファイルシステムをマウント解除します。

例:

```
# umount /oldfs1
```

ファイルシステムは、マウント解除する必要があります。これにより、ファイルシステムを再マウントして、マウント時に割り当てファイルを読み込ませることができます。ファイルシステムのマウント解除の詳細は、65 ページの「ファイルシステムのマウント解除」を参照してください。

7. `samfsck(1M) -F` コマンドを使用して、ファイルシステムの検査を実行します。

このコマンドによって、現在の正しい使用量情報を反映するように、割り当てファイルに割り当てられたレコードが更新されます。

例:

```
# samfsck -F /oldfs1
```

8. `mount(1M)` コマンドを使用して、割り当てファイルを作成したファイルシステムを再マウントします。

割り当ては、`/root` ディレクトリに 1 つまたは複数の割り当てファイルが検出された場合に有効になります。

`/etc/vfstab` ファイルまたは `samfs.cmd` ファイルに `quota` マウントオプションを指定する必要はありません。デフォルトで、`quota` マウントオプションは、`mount(1M)` コマンドで有効になっています。割り当ては、割り当てファイルが検出されたときに自動的に有効になります。

注 - 割り当てファイルが存在し、割り当てを有効にしてファイルシステムがマウントされている場合、ブロックまたはファイルの割り当てや解放が行われると、割り当てレコードが実際の使用量と一致しくなくなります。割り当ての設定されたファイルシステムがマウントされ、割り当てが無効の状態で作動している場合は、`samfsck(1M) -F` コマンドを実行し、割り当てファイルの使用量カウントを更新してから、割り当てを有効にしてファイルシステムを再マウントしてください。

`mount(1M)` コマンドの詳細については、`mount_samfs(1M)` のマニュアルページを参照してください。

9. `samquota(1M)` コマンドを使用して、ユーザー、グループ、または管理セットの割り当てを設定します。

この章のこのあとの項目では、この作業の手順と例を示します。`samquota(1M)` コマンドについては、`samquota(1M)` のマニュアルページを参照してください。

▼ ディレクトリとファイルへの管理セット ID を割り当てる

1. `su(1)` コマンドを使用して、スーパーユーザーになります。
2. 管理 ID を設定します。

次のように、`samchaid(1M)` コマンドを使用して、ディレクトリまたはファイルの管理セット ID を変更します。

- ファイルまたはディレクトリの ID を設定するには、ディレクトリ名またはパスを指定します。

例:

```
# samchaid 100 admin.dir
```

- ディレクトリツリーの ID を設定するには、`-R` オプションと、必要であれば、`-h` オプションを使用します。`-R` オプションでは再帰的な操作が指定され、`-h` オプションではターゲットではなくリンクが変更されます。

例:

```
# samchaid -R -h 22 /qfs1/joe /qfs1/nancee
```

`samchaid(1M)` コマンドについては、`samchaid(1M)` のマニュアルページを参照してください。

無限割り当てを設定する

無限割り当ては、特別な割り当ての一種です。無限割り当てと設定されているユーザーは、使用可能なすべてのファイルシステム資源へのアクセスが常に許可されます。無限割り当ての値は、ユーザー、グループ、または管理セット ID の割り当てファイルの 0 レコードに設定できます。そのレコードの値は、新しいユーザー、グループ、または管理セット ID のデフォルト値として使用できます。

▼ 無限割り当てを設定する

- `samquota(1M)` コマンドを使用して、割り当てを 0 に設定します。

例:

```
# samquota -U fred -b 0:h -f 0:h /qfs1
```

samquota(1M) コマンドを使用して、強い制限値および弱い制限値をすべて 0 にすることで、特定のユーザー、グループ、管理セット ID の無限割り当てを設定できます。コード例 5-2 に、無限割り当ての設定方法を示します。

コード例 5-2 無限割り当てを設定する

```
# samquota -G sam -b 0:s,h -f 0:s,h /sam6
# samquota -G sam /sam6
```

			Online Limits			Total Limits		
Type	ID	In Use	Soft	Hard	In Use	Soft	Hard	
/sam6								
Files	group 101	339	0	0	339	0	0	
Blocks	group 101	248	0	0	2614	0	0	
Grace period			0s			0s		

---> Infinite quotas in effect.

デフォルトの割り当て値を有効にする

samquota(1M) コマンドを使用して、ユーザー、グループ、または管理セットのデフォルト割り当てを有効にすることができます。このためには、ユーザー、グループ、管理セットのデフォルトの割り当て値を 0 に設定します。

▼ ユーザー、グループ、管理セットのデフォルト割り当て値を有効にする

- samquota(1M) コマンドを使用します。

たとえば、次の samquota(1M) コマンドでは、すべての管理セット ID のデフォルト割り当てが設定されます。

```
# samquota -A 0 -b 12000:s -b 15000:h -b 12G:s:t -b 15G:h:t \
-f 1000:s -f 1200:h -t 1w /qfs1
```

このコマンドによって、すべてのユーザーの初期化されていない管理セットの割り当てが次のように設定されます。

- オンラインブロックの弱い制限値 (-b limit:s) は、12,000 ブロックに設定される。
- オンラインブロックの強い制限値 (-b limit:h) は、15,000 ブロックに設定される。
- 総ブロックの弱い制限値 (-b limit:s:t) は、12 ギガブロックに設定される。
- 総ブロックの強い制限値 (-b limit:h:t) は、15 ギガブロックに設定される。
- ファイルの弱い制限値 (-f limit:s) は、1000 ファイルに設定される。

- ファイルの強い制限値 (`-f limit:h`) は、1200 ファイルに設定される。
- 猶予期間 (`-t limit`) は 1 週間に設定される。

注 - 割り当てレコードがすでに存在する場合は、既存の値が引き続き有効になります。これは、管理グループにすでにブロックが割り当てられている場合などに発生します。

同様に、ユーザーまたはグループのデフォルト割り当ては、`-A 0` の代わりに `-U 0` または `-G 0` を指定して設定できます。

`samquota(1M)` コマンドについては、`samquota(1M)` のマニュアルページを参照してください。

割り当てを有効にする

`samquota(1M)` コマンドを使用して、特定のユーザー、グループ、または管理セットの割り当て値の組み合わせを有効にすることができます。

▼ ユーザー、グループ、管理セットの割り当て値を有効にする

- `samquota(1M)` コマンドを使用します。

コード例 5-3 に、ユーザー、グループ、および管理セットの制限値をそれぞれ有効にするコマンドを示します。

コード例 5-3 割り当てのコマンド

```
# samquota -U joe -b 15000:s -b 20000:h -b 12G:s:t -b 15G:h:t \  
-f 500:s -f 750:h -t 3d /qfs1  
# samquota -G proj -b 15000:s -b 20000:h -b 12G:s:t -b 15G:h:t \  
-f 500:s -f 750:h -t 3d /qfs1  
# samquota -A 7 -b 15000:s -b 20000:h -b 12G:s:t -b 15G:h:t \  
-f 500:s -f 750:h -t 3d /qfs1
```

`samquota(1M)` コマンドについては、`samquota(1M)` のマニュアルページを参照してください。

▼ 既存の割り当てファイルを使用してユーザー、グループ、管理セットの割り当て値を有効化または変更する

割り当てを設定したあとで、既存の割り当てファイルをテンプレートとして使用して、別のユーザー、グループ、または管理セットに対する割り当て値を作成できます。この手順は、次のとおりです。この手順を使用して、任意の割り当て設定を変更することもできます。

1. `samquota(1M)` コマンドを使用して割り当てファイルを検索し、出力を一時ファイルにリダイレクトします。

次の1つまたは複数の追加のオプションを付けて `-e` オプションを使用してください。 `-U userID`、`-G groupID`、または `-A adminsetID`。

コード例 5-4 に、テンプレートとして使用するために、`quota.group` 作成したり取り出したりする方法を示します。

注 – グループ割り当てエントリをテンプレートとして使用し、ユーザー割り当てエントリを作成できます。

コード例 5-4 ファイル `quota.group`

```
# samquota -G sam -e /sam6 > /tmp/quota.group
# cat /tmp/quota.group

# Type ID
#
#           Online Limits
#           soft          hard
# Files
# Blocks
# Grace Periods
#
samquota -G 101 \
-f      200:s:o -f      300:h:o          -f      200:s:t -f      300:h:t \
-b      40000:s:o -b     60000:h:o        -b 40000000:s:t -b 60000000:h:t \
-t      0s:o          -t      0s:t          /sam6
```

2. エディタを使用して、作成した一時ファイルを編集します。

コード例 5-5 は、手順 1 で作成したファイルを `vi(1)` エディタで開いた様子を示しています。Group ID 101 が 102 に変更されています。これは、グループ 101 の割り当てセットをグループ 102 にコピーするコマンドを生成する効果があります。

コード例 5-5 編集後のファイル `quota.group`

```
# Type ID
#
#           Online Limits
#           soft          hard
# Files
# Blocks
# Grace Periods
#
samquota -G 102 \
-f      200:s:o -f      300:h:o          -f      200:s:t -f      300:h:t \
-b      40000:s:o -b     60000:h:o        -b 40000000:s:t -b 60000000:h:t \
-t      1d:o          -t      1d:t          /sam6
```

3. ファイルを保存して、エディタを終了します。
4. エディタで加えた変更を適用するには、シェルを使用してファイルを実行します。
例:

```
# sh -x /tmp/quota.group
```

この例で、`-x` オプションは、実行するコマンドのエコーをシェルに指示しています。必要に応じて、`-x` オプションは省略できます。

割り当ての検査

ディスクおよび i ノード割り当てを有効にしたあとで、各割り当てを検査できます。`samquota(1M)` コマンドは、個々のユーザー、グループ、管理セットについての割り当てレポートを生成する管理者コマンドです。`squota(1)` コマンドは、ユーザーが自分の割り当てを検査するためのユーザーコマンドです。

▼ 超過した割り当てを検査する

1. スーパーユーザーになります。
2. `samquota(1M)` コマンドを使用して、マウントされたファイルシステムで有効な割り当てを表示します。
 - ユーザー割り当てを表示するには、次のコマンドを指定します。

```
# samquota -U userID [ file ]
```

`userID` には、割り当てを調べるユーザーのユーザー ID (数値) またはユーザー名を指定します。

`file` には、選択したユーザー、グループ、または管理セットに対して、特定のファイルシステムを指定します。`file` 引数には、ファイルシステム内の任意のファイル名も指定できます。通常は、`file` にはファイルシステムのルートディレクトリ名を指定します。

例 1: コード例 5-6 によって、サーバー上の `sam6` ファイルシステムのユーザー `hm1259` の割り当て統計が取り出され、このユーザーは割り当てを超過していないことを示す出力が表示されます。

コード例 5-6 ユーザー hm1259 の割り当て超過の検査

```
# samquota -U hm1259 /sam6
```

	Type	ID	In Use	Online Limits		In Use	Total Limits	
				Soft	Hard		Soft	Hard
/sam6								
Files	user	130959	13	100	200	13	100	200
Blocks	user	130959	152	200	3000	272	1000	3000
Grace period				0s			0s	

例 2: コード例 5-7 によって、すべてのマウントされた Sun StorageTek QFS ファイルシステムのユーザー memil の割り当て統計が取り出され、このユーザーが割り当てを超過していることを示す出力が表示されます。出力の Blocks 行の正符号 (+) に注意してください。ファイルに対する割り当てが弱い制限値を超過している場合は、正符号は Files 行にも表示されます。

コード例 5-7 ユーザー memil の割り当て超過の検査

```
# samquota -U memil
```

	Type	ID	In Use	Online Limits		In Use	Total Limits	
				Soft	Hard		Soft	Hard
/sam6								
Files	user	130967	4	500	750	4	500	750
Blocks	user	130967	41016+	40000	50000	41016	50000	50000
Grace period				1w			0s	
---> Warning: online soft limits to be enforced in 6d23h36m45s								
/sam7								
Files	user	130967	4	500	750	4	500	750
Blocks	user	130967	4106	40000	50000	4106	50000	50000
Grace period				1w			0s	

強い制限値を超過している場合、または弱い制限値を超過して猶予期間が過ぎた場合は、該当する In Use フィールドにアスタリスク記号 (*) がマークされます。割り当てレコードの割り当て値に一貫性がない (たとえば、弱い制限値が強い制限値よりも大きい場合) と判別されると、感嘆符 (!) がフィールドにマークされ、すべての割り当て操作が禁止されます。

- グループ割り当てを表示するには、次のコマンドを指定します。

```
# samquota -G groupID [ file ]
```

groupID には、割り当てを確認するユーザーグループのグループ ID (数値) またはグループ名を指定します。*file* には、選択したグループ用の特定のファイルシステムを指定します。*file* 引数には、ファイルシステム内の任意のファイル名も指定できます。通常は、*file* にはファイルシステムのルートディレクトリ名を指定します。

たとえば、次のコマンドでは、qfs3 ファイルシステムのグループ turtles のユーザー割り当てが取り出されます。

```
# samquota -G turtles /qfs3
```

- 管理セットの割り当てを表示するには、次のコマンドを指定します。

```
# samquota -A adminsetID [ file ]
```

adminsetID には、割り当てを調べるサイト固有管理者セットの管理セット ID (数値) を指定します。*file* には、選択した管理セット用の特定のファイルシステムを指定します。*file* 引数には、ファイルシステム内の任意のファイル名も指定できます。通常は、*file* にはファイルシステムのルートディレクトリ名を指定します。

たとえば、次のコマンドでは、すべてのマウントされた Sun StorageTek QFS ファイルシステムの管理セット 457 のユーザー割り当て統計が取り出されます。

```
# samquota -A 457
```

割り当ての変更と削除

割り当てを変更して、ユーザーに割り当てるディスク領域の容量や i ノード数を調整できます。また、ユーザーやファイルシステム全体から割り当てを削除することもできます。この節では、割り当ての変更や削除の方法について説明します。

- 114 ページの「猶予期間を変更する」
- 115 ページの「猶予期間の期限を変更する」
- 117 ページの「追加のファイルシステム資源の割り当てを禁止する」
- 119 ページの「ファイルシステム割り当てを削除する」
- 121 ページの「割り当てを修正する」

▼ 猶予期間を変更する

samquota(1M) コマンドを使用して、弱い制限値の猶予期間を変更できます。

1. samquota(1M) コマンドを使用して、ユーザー、グループ、または管理セットの割り当て統計を取得します。

方法については、111 ページの「超過した割り当てを検査する」を参照してください。

例: コード例 5-8 は、グループ sam に関する情報を取り出し、このグループが弱い制限値を超過していることを示します。

コード例 5-8 弱い制限値を超過する

```
# samquota -G sam /sam6
```

			Online Limits			Total Limits		
	Type	ID	In Use	Soft	Hard	In Use	Soft	Hard
/sam6								
Files	group	101	32	2000	2000	32	2000	2000
Blocks	group	101	41888*	40000	60000000	43208	60000000	60000000
Grace period				1w			1w	

---> Online soft limits under enforcement (since 30s ago)

2. samquota(1M) コマンドの出力を調べて、割り当て値をどのように変更するかを決めます。
3. samquota(1M) コマンドを使用して、弱い制限値の猶予期間を変更します。
コード例 5-9 に、使用する samquota(1M) コマンドのオプションを示します。

コード例 5-9 samquota(1M) を使用して弱い制限値の猶予期間を変更する

```
# samquota -U userID -t interval file  
# samquota -G groupID -t interval file  
# samquota -A adminID -t interval file
```

これらのコマンドの引数は次のとおりです。

- *userID* は、割り当てを変更するユーザーのユーザー ID (数値) またはユーザー名です。
- *groupID* は、割り当てを変更するユーザーグループのグループ ID (数値) またはグループ名です。
- *adminID* は、割り当てを変更するサイト固有の管理者セットの管理セット ID (数値) です。

- *interval* は猶予期間に使用する間隔です。間隔には長さを表す整数値を指定し、必要であれば単位乗数を指定します。デフォルトの単位乗数は *s* で、秒数を表します。また、*w* (週)、*d* (日)、*h* (時間)、または *m* (分) も指定できます。
 - *file* は、選択したユーザー、グループ、または管理セットに対する特定のファイルシステムです。*file* 引数には、ファイルシステム内の任意のファイル名も指定できます。通常、*file* にはファイルシステムのルートディレクトリ名を指定します
- たとえば、ユーザー *memil* の猶予期間を変更するとします。コード例 5-10 は、割り当てとその出力を確認する *samquota(1M)* コマンドを示しています。

コード例 5-10 猶予期間を変更する

```
# samquota -U memil /sam6
```

	Type	ID	In Use	Online Limits		In Use	Total Limits	
				Soft	Hard		Soft	Hard
/sam6								
Files	user	130967	4	500	750	4	500	750
Blocks	user	130967	41016+	40000	50000	41016	50000	50000
Grace period				3d			0s	
---> Warning: online soft limits to be enforced in 2d23h59m7s								

猶予期間を短縮するには、次のコマンドを入力します。

```
# samquota -U memil -t 1d /sam6
```

コード例 5-11 に、新しい割り当てを確認する *samquota(1M)* コマンドを示します。

コード例 5-11 新しい割り当てを確認する

```
# samquota -U memil /sam6
```

	Type	ID	In Use	Online Limits		In Use	Total Limits	
				Soft	Hard		Soft	Hard
/sam6								
Files	user	130967	4	500	750	4	500	750
Blocks	user	130967	41016+	40000	50000	41016	50000	50000
Grace period				1d			0s	
---> Warning: online soft limits to be enforced in 23h58m31s								

猶予期間の期限を変更する

ユーザーが弱い制限値を超過した場合は、猶予期間そのものを変更しても、すでに開始した猶予期間の有効期限タイマーは変更されません。猶予期間がすでに開始している場合は、*samquota(1M)* コマンドを使用すると、次のどれかの方法で猶予期間を変更できます。

- 猶予期間タイマーをクリアする – ユーザーが弱い制限値を超えたまま、次にファイルまたはブロックを割り当てると、猶予期間タイマーがリセットされて猶予期間が再開します。

コード例 5-12 は、タイマーをクリアし、グループ sam のユーザーが次に /sam6 でブロックまたはファイルを割り当てようとしたときにカウントが始まるようにするために使用するコマンドを示しています。

コード例 5-12 タイマーをクリアする

```
# samquota -G sam -x clear /sam6
Setting Grace Timer:  continue? y
# samquota -G sam /sam6
```

	Type	ID	In Use	Online Limits		In Use	Total Limits	
				Soft	Hard		Soft	Hard
/sam6								
Files	group	101	32	2000	2000	32	2000	2000
Blocks	group	101	41888+	40000	60000000	43208	60000000	60000000
Grace period				1w			1w	

---> Warning: online soft limits to be enforced in 6d23h59m56s

- 猶予期間タイマーをリセットする – 猶予期間がリセットされると、タイマーがリセットされ、猶予期間が再開します。

コード例 5-13 では、猶予期間をリセットします。

コード例 5-13 猶予期間タイマーをリセットする

```
# samquota -G sam -x reset /sam6
Setting Grace Timer:  continue? y
# samquota -G sam /sam6
```

	Type	ID	In Use	Online Limits		In Use	Total Limits	
				Soft	Hard		Soft	Hard
/sam6								
Files	group	101	32	2000	2000	32	2000	2000
Blocks	group	101	41888	40000	60000000	43208	60000000	60000000
Grace period				1w			1w	

---> Warning: online soft limits to be enforced in 6d23h59m52s

- 猶予期間に値を設定する – タイマーに値を設定すると、すぐにその値からカウントが始まります。値の制限はない。猶予期間を超える値も指定できる

コード例 5-14 では、非常に長い有効期限を設定します。

コード例 5-14 非常に長い猶予期間を設定する

```
# samquota -G sam -x 52w /sam6
Setting Grace Timer:  continue? y
# samquota -G sam /sam6
```

コード例 5-14 非常に長い猶予期間を設定する (続き)

```

Type      ID      In Use      Online Limits      Total Limits
Soft      Hard      In Use      Soft      Hard
/sam6
Files group 101      32      2000      2000      32      2000      2000
Blocks group 101      41888+  40000 60000000  43208 60000000 60000000
Grace period      1w      1w
---> Warning:  online soft limits to be enforced in 51w6d23h59m54s
```

- 猶予期間タイマーを終了する - タイマーがすぐに終了するように設定します。コード例 5-15 では、猶予期間を終了します。

コード例 5-15 猶予期間タイマーを終了する

```
# samquota -G sam -x expire /sam6
Setting Grace Timer:  continue? y
# samquota -G sam /sam6

Type      ID      In Use      Online Limits      Total Limits
Soft      Hard      In Use      Soft      Hard
/sam6
Files group 101      32      2000      2000      32      2000      2000
Blocks group 101      41888      40000 60000000  43208 60000000 60000000
Grace period      1w      1w
---> Online soft limits under enforcement (since 6s ago)
```

▼ 追加のファイルシステム資源の割り当てを禁止する

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資源の使用を禁止します。矛盾する割り当て値を作成して、ファイルシステム資源の割り当てを禁止できます。たとえば、ブロックまたはファイルの強い制限値が弱い制限値よりも小さい場合や、ユーザーの弱い制限値が強い制限値よりも大きい場合に、追加の割り当てを禁止できます。

ファイルシステムは、矛盾した割り当ての設定を特別な割り当てとして処理します。矛盾した割り当ての値は、ユーザー、グループ、または管理セット ID の割り当てファイルの 0 レコードに設定することができます。また、そのレコードの値を、新しいユーザー、グループ、または管理セット ID のデフォルト値にすることができます。

この手順では、ユーザー、グループ、または管理セットに対する追加のシステム資源の割り当てを禁止する方法を示します。

1. スーパーユーザーになります。
2. 現在の割り当て情報を取得して、保存して確認します。

コード例 5-16 は、`samquota(1M)` コマンドを使用して、グループ `sam` の現在のグループ割り当て情報を取り出し、バックアップファイルに書き込む方法を示しています。

コード例 5-16 グループ割り当て情報を取り出す

```
# samquota -G sam -e /sam6 | & tee restore.quota.sam

# Type ID
# Online Limits Total Limits
# soft hard soft hard
# Files
# Blocks
# Grace Periods
#
samquota -G 101 \
-f 2000:s:o -f 2000:h:o -f 2000:s:t -f 2000:h:t \
-b 40000:s:o -b 60000000:h:o -b 60000000:s:t -b 60000000:h:t \
-t 1w:o -t 1w:t \
-x 51w6d23h59m:o -x clear /sam6
```

ユーザーの割り当て情報を取得するには、`-G` オプションの代わりに `e -U userID` オプションを指定します。管理セットの割り当て情報を取得するには、`-G` オプションの代わりに `-A adminID` オプションを指定します。

3. `samquota(1M)` コマンドを使用して、弱い制限値を 0 以外の割り当てに設定し、強い制限値を 0 割り当てに設定します。

次のコマンドでは、グループ `sam` の割り当てが矛盾するように設定されます。

```
# samquota -G sam -f 1:s -f 0:h -b 1:s -b 0:h /sam6
```

ユーザーまたは管理セットの割り当てを矛盾させるには、`-G` オプションの代わりに、`-U userID` オプションまたは `-A adminID` オプションを指定します。

4. `samquota(1M)` コマンドを使用して、変更内容を確認します。

コード例 5-17 にこの例を示します。

コード例 5-17 変更した割り当てを確認する

```
# samquota -G sam /sam6

Type ID In Use Online Limits Total Limits
Soft Hard In Use Soft Hard
/sam6
```

コード例 5-17 変更した割り当てを確認する (続き)

```
Files group 101 32! 1 0 32! 1 0
Blocks group 101 41888! 1 0 43208! 1 0
Grace period 1w 1w
---> Quota values inconsistent; zero quotas in effect.
```

この出力では、0 割り当てが有効になっています。この出力では、割り当ての超過状態を示す感嘆符 (!) に注意してください。

5. sh(1) コマンドおよび samquota(1M) コマンドを使用して、ファイルとブロックの割り当てが禁止されていた以前の状態にグループの割り当てを戻してから、変更された割り当てを確認します。

コード例 5-18 にこれらのコマンドを示します。

コード例 5-18 グループ割り当てを復元する

```
# sh restore.quota.sam
Setting Grace Timer: continue? y
Setting Grace Timer: continue? y
# samquota -G sam /sam6

          Type      ID      In Use      Online Limits      Total Limits
          Soft      Hard      In Use      Soft      Hard      Soft      Hard
/sam6
Files group 101      32      2000      2000      32      2000      2000
Blocks group 101      41888+ 40000 60000000 43208 60000000 60000000
Grace period 1w      1w
---> Warning: online soft limits to be enforced in 6d23h59m54s
```

ユーザー割り当てについてこの操作を実行するには、-G オプションの代わりに -U *userID* オプションを指定します。管理セットの割り当てでこの操作を行うには、-G オプションの代わりに -A *adminID* オプションを指定します。

▼ ファイルシステム割り当てを削除する

ファイルシステム割り当てを削除または無効化するには、マウント処理で割り当てを無効にする必要があります。

1. su(1) コマンドを使用して、スーパーユーザーになります。
2. (省略可能) テキストエディタを使用して、/etc/vfstab ファイルまたは samfs.cmd ファイルに noquota マウントオプションを追加します。

または、mount コマンドを実行するときに noquota をオプションとしてあとから指定します。手順 4 を参照してください。

3. ファイルシステムがマウントされている場合は、`umount(1M)` コマンドを使用してファイルシステムをマウント解除します。

例:

```
# umount /myfs
```

ファイルシステムのマウント解除の詳細については、65 ページの「ファイルシステムのマウント解除」を参照してください。

4. `mount(1M)` コマンドを使用して、ファイルシステムを再マウントします。

手順 2 を実行しなかった場合は、`mount(1M)` コマンドに `noquota` オプションを付けます。

例:

```
# mount -o noquota /myfs
```

5. 次のいずれかを実行して、割り当てファイルを破棄します。

- あとで割り当て機能を用に戻す予定があるため、割り当てファイルを削除しない場合は、ファイルシステムをマウント解除し、ファイルシステムで `-F` オプションを指定して `samfsck(1M)` コマンドを実行し、`noquota` マウントオプションなしで再度ファイルシステムをマウントします。
- 割り当て機能を用に戻さない場合、または割り当てファイルに使用されている領域を再利用する場合は、`rm(1)` コマンドを使用して、`.quota_u` ファイル、`.quota_g` ファイル、`.quota_a` ファイルを削除します。

例:

```
# rm /myfs/.quota_[agu]
```

▼ 割り当てを修正する

1. スーパーユーザーになります。
2. ファイルシステムがマウントされている場合は、`umount(1M)` コマンドを使用してファイルシステムをマウント解除します。

例:

```
# umount /myfs
```

ファイルシステムのマウント解除の詳細については、65 ページの「ファイルシステムのマウント解除」を参照してください。

3. `samfsck(1M) -F` コマンドを使用して、ファイルシステムの検査を実行します。
`samfsck(1M)` コマンドによって、現在の正しい使用量情報を反映するように、割り当てファイルに指定されたレコードが更新されます。

例:

```
# samfsck -F myfs
```

4. `mount(1M)` コマンドを使用して、ファイルシステムを再マウントします。

例:

```
# mount /myfs
```


第6章

Sun Cluster 環境での Sun StorageTek QFS の構成

この章では、Sun Cluster 環境での Sun StorageTek QFS の動作について説明します。また、Sun Cluster 環境での Sun StorageTek QFS 共有ファイルシステム、および Sun Cluster 環境での非共有 Sun StorageTek QFS ファイルシステムの構成例も示します。

この章の内容は次のとおりです。

- 124 ページの「事前確認」
- 125 ページの「制限事項」
- 126 ページの「Sun Cluster システムと Sun StorageTek QFS ソフトウェアの相互作用」
- 127 ページの「Sun Cluster 用 Solaris Volume Manager に対する Sun StorageTek QFS のサポート」
- 134 ページの「構成例について」
- 135 ページの「Sun Cluster 環境での Sun StorageTek QFS 共有ファイルシステムの構成」
- 146 ページの「Sun Cluster 環境での非共有ファイルシステムの構成」
- 168 ページの「クラスタの外部にある共有クライアントの構成」
- 180 ページの「Sun StorageTek QFS の構成の変更」
- 186 ページの「Sun Cluster を使用した高可用性 Sun StorageTek SAM 構成」

事前確認

Sun StorageTek QFS ソフトウェアの version 4U2 以降では、Sun StorageTek QFS を Sun Cluster 環境にインストールして、高可用性を持つファイルシステムを構成できます。使用する構成方法は、ファイルシステムが共有か非共有かによって異なります。

この章は、Sun StorageTek QFS ソフトウェアと Sun Cluster 環境をすでに使用しているユーザーを対象に書かれています。また、次の両方の経験があることが前提になっています。

- Sun Cluster ソフトウェアで制御される、高可用性でスケーラブル、またはフェイルオーバーリソースとしてのファイルシステムの構成。
- Sun StorageTek QFS スタンドアロンファイルシステムおよび Sun StorageTek QFS 共有ファイルシステムの両方のインストールと構成。

この章をお読みになる前に、次のマニュアルをお読みになることを推奨します。

- 『Sun StorageTek QFS インストールおよびアップグレードの手引き』の次の章
 - 第2章。Sun Cluster 環境における Sun StorageTek QFS のハードウェアとソフトウェアの前提条件が書かれています。
 - 第4章。高可用性を得るための、Sun Cluster システムへの Sun StorageTek QFS ファイルシステムインストールの追加作業が説明されています。
- 次の Sun Cluster ソフトウェアマニュアル
 - 『Sun Cluster Concepts Guide for Solaris OS』。Sun Cluster 環境の基礎が書かれています。特に、次の事項に関する項目を確認してください。
 - ローカルディスク
 - グローバル装置
 - 装置 ID (DID)
 - ディスク装置グループ
 - ディスク装置グループのフェイルオーバー
 - ローカルおよびグローバル名前空間
 - クラスタファイルシステム
 - HAStoragePlus リソースタイプ
 - ボリュームマネージャー
 - 『Sun Cluster Software Installation Guide for Solaris OS』。Sun Cluster ソフトウェアのインストール手順が書かれています。

- 『Sun Cluster Data Services Planning and Administration Guide for Solaris OS』。Sun Cluster 環境にさまざまなデータサービスを取り入れるための計画方法について説明されています。
- 『Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS』。Oracle Real Application Cluster (RAC) 用 Sun Cluster データサービスで Sun StorageTek QFS 共有ファイルシステムを使用する方法が説明されています。

注 – File System Manager ソフトウェアは、Sun Cluster 環境でファイルシステムを制御する目的でも使用できます。File System Manager ソフトウェアは、クラスタノードを認識および識別します。サーバーを追加した場合は、ユーザーに対してクラスタノードを追加するよう自動的に促します。Sun Cluster を構成するノードに対して、アーカイブ非対応の高可用性 (HA) の共有またはスタンドアロン Sun StorageTek QFS ファイルシステムのどちらを作成するかを選択するオプションがあります。詳細な情報は、File System Manager のオンラインヘルプを参照してください。

制限事項

Sun Cluster 環境の Sun StorageTek QFS ソフトウェアには、次の制限があります。

- 共有ファイルシステムには、次の制限があります。
 - SUNW.qfs リゾースタイプとしてだけ構成できます。HAStoragePlus リゾースタイプとしては構成できません。
 - Oracle Real Application Cluster 用の Sun Cluster データサービスだけが使用する、スケラブルなファイルシステムとして構成できます。
- 非共有ファイルシステムには、次の制限があります。
 - HAStoragePlus リゾースタイプとしてだけ構成できます。SUNW.qfs リゾースタイプとしては構成できません。
 - Sun Cluster でサポートされるすべてのフェイルオーバーアプリケーションは、非共有 Sun StorageTek QFS ソフトウェアで高可用性ローカルファイルシステムとしてサポートされます。サポートは、特定のアプリケーションに限定されません。
 - 非共有 Sun StorageTek QFS ファイルシステムは、Sun Cluster システムの基礎となるネイティブファイルシステムにできません。

注 – フェイルバックは、SUNW.qfs エージェントの機能としてはサポートされていません。

注 – Sun StorageTek QFS ファイルシステムを Sun Cluster 環境にインストールすると、信頼性が向上し、予定されていない停止時間を減らすことができますが、予定されている停止時間はなくすることはできません。ファイルシステムの健全性を維持するために、Sun StorageTek QFS ソフトウェアを停止して samfsck プロセスを実行することが必要な場合があります。また、ソフトウェアのパッチまたは更新を適用するためには、シャットダウンも必要です。

Sun Cluster システムと Sun StorageTek QFS ソフトウェアの相互作用

共有ファイルシステムは Sun Cluster ディスク識別子 (DID) サポートを使用して、Oracle Real Application Cluster 用 Sun Cluster データサービスによるデータアクセスを可能にしています。非共有ファイルシステムはグローバル装置ボリュームサポートと、ボリュームマネージャーで制御されたボリュームサポートを使用して、Sun Cluster システムでサポートされたフェイルオーバーアプリケーションからのデータアクセスを可能にしています。

共有ファイルシステムでのデータアクセス

DID サポートにより、Sun Cluster システムで制御された各装置は、それがマルチパスかどうかにかかわらず一意のディスク ID を割り当てられます。一意の DID 装置ごとに、対応するグローバル装置があります。Sun StorageTek QFS 共有ファイルシステムは、DID 装置 (/dev/did/*) だけで構成される冗長ストレージに構成でき、DID 装置はホストバスアダプタ (HBA) を通じてその装置への直接接続を持つノードからのみアクセス可能です。

DID 装置に Sun StorageTek QFS 共有ファイルシステムを構成し、そのファイルシステムで使用するよう SUNW.qfs リソースタイプを構成すると、ファイルシステムの共有メタデータサーバーの可用性が高まります。Oracle Real Application Cluster 用 Sun Cluster データサービスは、そのファイルシステム内のデータにアクセスできます。また、Sun StorageTek QFS の Sun Cluster エージェントは、必要に応じてそのファイルシステム用のメタデータサーバーを自動的に再配置します。

注 – Sun StorageTek QFS ソフトウェアの version 4U6 以降では、Sun Cluster 環境内のクラスタの外部に共有クライアントを持つこともできます。構成の詳細手順については、168 ページの「クラスタの外部にある共有クライアントの構成」を参照してください。

非共有ファイルシステムでのデータアクセス

グローバル装置は、Sun Cluster システム内のすべてのノードから基礎となる DID 装置をアクセスするための Sun Cluster システムのメカニズムで、その DID 装置を格納しているノードが使用可能なことを前提にしています。グローバル装置、およびボリュームマネージャーで制御されたボリュームは、Sun Cluster システム内のすべてのノードからアクセス可能にできます。非共有 Sun StorageTek QFS ファイルシステムは、raw グローバル装置 (/dev/global/*) またはボリュームマネージャーで制御されたボリュームから構成される冗長ストレージ上に構成できます。

このようなグローバル装置またはボリュームマネージャーで制御された装置上に非共有ファイルシステムを構成し、そのファイルシステムで使用するように HAStoragePlus リソースタイプを構成すると、そのファイルシステムはほかのノードへのフェイルオーバー機能を持ち、可用性が高まります。

Sun Cluster 用 Solaris Volume Manager に対する Sun StorageTek QFS のサポ ー ト

Sun StorageTek QFS の 4U4 release では、Sun Cluster 用 Solaris Volume Manager のサポートが追加されました。これは Solaris 9 および Solaris 10 の OS リリースにバンドルされている Solaris™ Volume Manager を拡張したものです。Sun StorageTek QFS は、Solaris 10 上の Sun Cluster 用 Solaris Volume Manager のみをサポートしません。

Sun Cluster 用 Solaris Volume Manager に対する Sun StorageTek QFS のサポートを導入することで、共有 Sun StorageTek QFS のホストベースミラーリングを利用できるほか、Oracle RAC ベースアプリケーション用に Oracle が実装したアプリケーションバイナリリカバリ (ABR) およびダイレクトミラー読み込み (DMR) を利用できるようになりました。

Sun Cluster 用 Solaris Volume Manager を Sun StorageTek QFS とともに使用するためには、Sun Cluster ソフトウェアとそこに含まれている追加の単体ソフトウェアパッケージが必要です。

Sun Cluster 用 Solaris Volume Manager のサポートを追加することで、新たに 4 つのマウントオプションが導入されました。Sun Cluster 用 Solaris Volume Manager 上に Sun StorageTek QFS が構成されたことが検出された場合にのみ、これらのマウントオプションを使用できます。マウントオプションは次のとおりです。

- abr - アプリケーションバイナリリカバリを使用可能にします
- dmr - ダイレクトミラー読み込みを使用可能にします

- noabr - アプリケーションバイナリリカバリを使用不可にします
- nodmnr - ダイレクトミラー読み込みを使用不可にします

Sun Cluster 用 Solaris Volume Manager での Sun StorageTek QFS の使用例を、次に示します。

この例では、次の設定が既に行われていることを前提とします。

- Sun Cluster がインストールされていて、インストールが完了している状態です。
- 追加パッケージがインストールされていて、Sun Cluster Oracle RAC フレームワークを使用できる状態です。
- 追加の Sun Cluster SUNWscmd パッケージがインストールされていて、Sun Cluster の Solaris Volume Manager を使用できる状態です。
- Oracle RAC フレームワークリソースグループが作成されていて、オンラインの状態です。

この例では、3 つの共有 Sun StorageTek QFS ファイルシステムがあります。

- CRS - このファイルシステムは、Oracle 10G が Oracle RAC クラスタレディサービス (CRS) を構成するために必要です。詳細は、『Sun Cluster Concepts』を参照してください。
- Data - このファイルシステムは、Oracle システムファイル、Oracle 制御ファイル、Oracle データファイルなどの、Oracle 関連ファイルを格納するために使用されます。
- Redo - このファイルシステムは、Oracle REDO ログ、Oracle アーカイブファイル、フラッシュバックファイル、ログを格納するために使用されます。

▼ Sun Cluster 用 Solaris Volume Manager でファイルシステムを構成する

1. 各ノードで metadb を作成します。

例:

```
# metadb -a -f -c3 /dev/rdsk/c0t0d0s7
```

2. 1 つのノードで ディスクグループを作成します。

例:

```
# metaset -s datadg -M -a -h scNode-A scNode-B
```

3. あるノードのデバイスを取得するために、`scdidadm` を実行します。

例:

```
scNode-A # scdidadm -l
13 scNode-A: /dev/rdisk/c6t600C0FF00000000000332B62CF3A6B00d0 /dev/did/rdisk/d13
14 scNode-A: /dev/rdisk/c6t600C0FF00000000000876E950F1FD9600d0 /dev/did/rdisk/d14
15 scNode-A: /dev/rdisk/c6t600C0FF00000000000876E9124FAF9C00d0 /dev/did/rdisk/d15
16 scNode-A: /dev/rdisk/c6t600C0FF00000000000332B28488B5700d0 /dev/did/rdisk/d16
17 scNode-A: /dev/rdisk/c6t600C0FF0000000000086DB474EC5DE900d0 /dev/did/rdisk/d17
18 scNode-A: /dev/rdisk/c6t600C0FF00000000000876E975EDA6A000d0 /dev/did/rdisk/d18
19 scNode-A: /dev/rdisk/c6t600C0FF0000000000086DB47E331ACF00d0 /dev/did/rdisk/d19
20 scNode-A: /dev/rdisk/c6t600C0FF00000000000876E9780ECA8100d0 /dev/did/rdisk/d20
21 scNode-A: /dev/rdisk/c6t600C0FF000000000004CAD5B68A7A100d0 /dev/did/rdisk/d21
22 scNode-A: /dev/rdisk/c6t600C0FF0000000000086DB43CF85DA800d0 /dev/did/rdisk/d22
23 scNode-A: /dev/rdisk/c6t600C0FF000000000004CAD7CC3CDE500d0 /dev/did/rdisk/d23
24 scNode-A: /dev/rdisk/c6t600C0FF0000000000086DB4259B272300d0 /dev/did/rdisk/d24
25 scNode-A: /dev/rdisk/c6t600C0FF00000000000332B21D0B90000d0 /dev/did/rdisk/d25
26 scNode-A: /dev/rdisk/c6t600C0FF000000000004CAD139A855500d0 /dev/did/rdisk/d26
27 scNode-A: /dev/rdisk/c6t600C0FF00000000000332B057D2FF100d0 /dev/did/rdisk/d27
28 scNode-A: /dev/rdisk/c6t600C0FF000000000004CAD4C40941C00d0 /dev/did/rdisk/d28
```

ミラーリングの方式は次のとおりです。

```
21 <-> 13
14 <-> 17
23 <-> 16
15 <-> 19
```

4. 1 つのノードに対して、セットになるようにデバイスを追加します。

例:

```
# metaset -s datadg -a /dev/did/rdisk/d21 /dev/did/rdisk/d13
/dev/did/rdisk/d14 \
/dev/did/rdisk/d17 /dev/did/rdisk/d23 /dev/did/rdisk/d16
/dev/did/rdisk/d15 \
/dev/did/rdisk/d19
```

5. 1 つのノードにミラーを作成します。

例:

```
metainit -s datadg d10 1 1 /dev/did/dsk/d21s0
metainit -s datadg d11 1 1 /dev/did/dsk/d13s0
metainit -s datadg d1 -m d10
metattach -s datadg d11 d1

metainit -s datadg d20 1 1 /dev/did/dsk/d14s0
metainit -s datadg d21 1 1 /dev/did/dsk/d17s0
metainit -s datadg d2 -m d20
metattach -s datadg d21 d2

metainit -s datadg d30 1 1 /dev/did/dsk/d23s0
metainit -s datadg d31 1 1 /dev/did/dsk/d16s0
metainit -s datadg d3 -m d30
metattach -s datadg d31 d3

metainit -s datadg d40 1 1 /dev/did/dsk/d15s0
metainit -s datadg d41 1 1 /dev/did/dsk/d19s0
metainit -s datadg d4 -m d40
metattach -s datadg d41 d4

metainit -s datadg d51 -p d1 10m
metainit -s datadg d52 -p d1 200m
metainit -s datadg d53 -p d1 800m

metainit -s datadg d61 -p d2 10m
metainit -s datadg d62 -p d2 200m
metainit -s datadg d63 -p d2 800m

metainit -s datadg d71 -p d1 500m
metainit -s datadg d72 -p d1 65g

metainit -s datadg d81 -p d2 500m
metainit -s datadg d82 -p d2 65g
```

6. 各ノードで Sun StorageTek QFS のインストールを行います。

例:

```
pkgadd -d .SUNWqfsr SUNWqfsu
```

7. 各ノードで mcf ファイルを作成します。

例:

```
/etc/opt/SUNWsamfs/mcf file:

#
# File system Data
#
Data                2    ma   Data   on     shared
/dev/md/datadg/dsk/d53 20   mm   Data   on
/dev/md/datadg/dsk/d63 21   mm   Data   on
/dev/md/datadg/dsk/d3  22   mr   Data   on
/dev/md/datadg/dsk/d4  23   mr   Data   on
#

# File system Crs
#
Crs                  4    ma   Crs    on     shared
/dev/md/datadg/dsk/d51 40   mm   Crs    on
/dev/md/datadg/dsk/d61 41   mm   Crs    on
/dev/md/datadg/dsk/d52 42   mr   Crs    on
/dev/md/datadg/dsk/d62 43   mr   Crs    on
#

# File system Redo
#
Redo                 6    ma   Redo   on     shared
/dev/md/datadg/dsk/d71 60   mm   Redo   on
/dev/md/datadg/dsk/d81 61   mm   Redo   on
/dev/md/datadg/dsk/d72 62   mr   Redo   on
/dev/md/datadg/dsk/d82 63   mr   Redo   on
```

8. ファイルシステムの hosts ファイルを作成します。

例:

```
/etc/opt/SUNWsamfs/hosts.Data
/etc/opt/SUNWsamfs/hosts.Crs
/etc/opt/SUNWsamfs/hosts.Oracle

# scNode-A:root> /usr/cluster/bin/scconf -p | egrep "Cluster node
name:|Node private hostname:"
      Cluster node name:      scNode-A
      Node private hostname:  clusternode1-priv
      Cluster node name:      scNode-B
      Node private hostname:  clusternode2-priv

# Host          Host IP          Server  Not   MDS Server
# Name          Address          Priority Used   Host
#-----
scNode-A        clusternode1-priv  1      -     server
scNode-B        clusternode2-priv  2      -
```

9. /etc/opt/SUNWsamfs/samfs.cmd ファイルを作成します。

例:

```
fs = Data
    stripe=1
    sync_meta=1
    mh_write
    qwrite
    forcedirectio
    notrace
    rdlease=300
    wrlease=300
    aplease=300
```

```

fs = Crs
    stripe=1
    sync_meta=1
    mh_write
    qwrite
    forcedirectio
    notrace
    rdlease=300
    wrlease=300
    aplease=300

fs = Redo
    stripe=1
    sync_meta=1
    mh_write
    qwrite
    forcedirectio
    notrace
    rdlease=300
    wrlease=300
    aplease=300

```

10. Sun StorageTek QFS ファイルシステムを作成します。詳細については、『Sun StorageTek QFS インストールおよびアップグレードの手引き』を参照してください。

例:

```
/opt/SUNWsamfs/sbin/sammkfs -S <filesystem>
```

11. Sun Cluster のリソースグループを構成して、Sun StorageTek QFS メタデータサーバーのフェイルオーバーを管理します。

- a. /etc/vfstab マウントエントリを構築し、追加します。

例:

```

#
#
# RAC on shared QFS
Data - /cluster/Data samfs - no shared,notrace
Redo - /cluster/Redo samfs - no shared,notrace
Crs - /cluster/Crs samfs - no shared,notrace

```

- b. クラスタをまたがるファイルシステムを、各ノードにマウントします。

はじめに、共有 Sun StorageTek QFS ファイルシステムを現在のメタデータサーバーにマウントし、次にそのファイルシステムを各メタデータクライアントにマウントします。

この処理を確認するため、次のように入力します。

```
# df -h -F samfs
```

- c. Sun Cluster のリソースグループを作成して、メタデータサーバーを管理します。

QFS リソースタイプを登録します。

```
# scrgadm -a -t SUNW.qfs
```

リソースグループに Sun Cluster と共有 Sun StorageTek QFS メタデータノードを追加します。

```
# scrgadm -a -g sc-QFS-rg -h scNode-A,sc-Node-B -y  
RG_DEPENDENCIES="rac-framework-rg"
```

リソースグループに共有 Sun StorageTek QFS ファイルシステムリソースと SUNWqfs リソースタイプを追加します。

```
# scrgadm -a -g sc-QFS-rg -t SUNW.qfs -j sc-qfs-fs-rs -x QFSFileSystem=  
/cluster/Data, \  
/cluster/Redo,/cluster/Crs
```

リソースグループをオンラインにします。

```
# scswitch -Z -g sc-QFS-rg
```

これで、共有 Sun StorageTek QFS ファイルシステムを使用することができます。

構成例について

この章では、Sun Cluster システムでの Sun StorageTek QFS 共有ファイルシステム、および Sun Cluster システムでの非共有 Sun StorageTek QFS ファイルシステムの構成例を示します。すべての構成例は、次から構成されるプラットフォームに基づいています。

- 2 つのノードの SunPlex™ システム (scnode-A と scnode-B)
- 高可用性と冗長性の両方を得るためのマルチホストアクセス可能なディスク装置を含んだ DID 装置

この章のすべての例は、コード例 6-1 に基づいています。このコード例で、`scdidadm(1M)` コマンドは DID 装置を表示し、`-L` オプションは Sun Cluster システム内のすべてのノード上の DID 装置を含む、DID 装置のパスを表示します。

コード例 6-1 DID 装置とその DID 装置パスを表示するコマンド

```
# scdidadm -L
1  scnode-A:/dev/dsk/c0t0d0      /dev/did/dsk/d1
2  scnode-A:/dev/dsk/c0t1d0      /dev/did/dsk/d2
3  scnode-A:/dev/dsk/c0t6d0      /dev/did/dsk/d3
4  scnode-A:/dev/dsk/c6t1d0      /dev/did/dsk/d4
4  scnode-B:/dev/dsk/c7t1d0      /dev/did/dsk/d4
5  scnode-A:/dev/dsk/c6t2d0      /dev/did/dsk/d5
5  scnode-B:/dev/dsk/c7t2d0      /dev/did/dsk/d5
6  scnode-A:/dev/dsk/c6t3d0      /dev/did/dsk/d6
6  scnode-B:/dev/dsk/c7t3d0      /dev/did/dsk/d6
7  scnode-A:/dev/dsk/c6t4d0      /dev/did/dsk/d7
7  scnode-B:/dev/dsk/c7t4d0      /dev/did/dsk/d7
8  scnode-A:/dev/dsk/c6t5d0      /dev/did/dsk/d8
8  scnode-B:/dev/dsk/c7t5d0      /dev/did/dsk/d8
9  scnode-B:/dev/dsk/c0t6d0      /dev/did/dsk/d9
10 scnode-B:/dev/dsk/c1t0d0      /dev/did/dsk/d10
11 scnode-B:/dev/dsk/c1t1d0      /dev/did/dsk/d11
```

コード例 6-1 は、DID 装置 d4 ~ d8 が両方の Sun Cluster システム (scnode-A と scnode-B) からアクセス可能なことを示しています。Sun StorageTek QFS ファイルシステムのサイズ要件、および目的とする用途と構成に関する知識から、ファイルシステムへの最適な装置の割り当てを決定できます。Solaris の `format(1M)` コマンドを使用すると、各 DID 装置のサイズおよびパーティションレイアウトを決定し、必要に応じて各 DID 装置のパーティションをサイズ変更できます。使用可能な DID 装置に関して、サイズ要件に合わせて、複数の装置、およびファイルシステムを格納するための対応するパーティションを構成できます。

Sun Cluster 環境での Sun StorageTek QFS 共有ファイルシステムの構成

Sun StorageTek QFS 共有ファイルシステムを Sun Cluster 環境にインストールするときは、ファイルシステムのメタデータサーバーを `SUNW.qfs` リソースタイプで構成できます。これにより、メタデータサーバーの可用性が高まり、Sun StorageTek QFS 共有ファイルシステムを、Sun Cluster 環境内で構成されたすべてのノードからグローバルにアクセス可能にできます。

通常、Sun StorageTek QFS 共有ファイルシステムはスケーラブルなアプリケーションと関連付けられます。Sun StorageTek QFS 共有ファイルシステムは 1 つ以上の Sun Cluster ノードにマウントされ、そのノードでスケーラブルなアプリケーションが有効になっています。

Sun Cluster システム内のノードで障害が発生するか、リソースグループを切り替えると、メタデータサーバーのリソース (Sun StorageTek QFS Sun Cluster エージェント) はファイルシステムのメタデータサーバーを必要に応じて自動的に再配置します。これにより、共有ファイルシステムへのほかのノードのアクセスが影響を受けなくなります。

注 – Sun Cluster システムで制御された Sun StorageTek QFS 共有ファイルシステムのメタデータサーバーを手動で再配置するには、Sun Cluster の管理用コマンドを使用する必要があります。これらのコマンドについての詳細は、Sun Cluster のマニュアルを参照してください。

メタデータサーバーのリソースの検討事項

Sun Cluster システムの起動時、メタデータサーバーのリソースにより、そのリソースグループに含まれるすべてのノードにファイルシステムが確実にマウントされます。ただし、このようなノードへのファイルシステムのマウントは監視されません。そのため、ある種の故障が発生した場合、メタデータサーバーのリソースがオンライン状態であっても、ファイルシステムを一部のノードで使用できないことがあります。

Sun Cluster の管理用コマンドを使用してメタデータサーバーのリソースグループをオフラインにした場合、メタデータサーバーのリソースであるファイルシステムは、そのノードにマウントされたままになります。ファイルシステムを (ノードを停止する以外の方法で) マウント解除するには、Sun Cluster の管理用コマンドを使用して、メタデータサーバーのリソースグループを管理対象外の状態にする必要があります。

あとでファイルシステムを再マウントするには、リソースグループを管理対象の状態にしてから、オンライン状態にする必要があります。

構成例

ここでは、Oracle Real Application Cluster 用 Sun Cluster データサービスで raw DID 装置にインストールされた Sun StorageTek QFS 共有ファイルシステムの例を示します。Oracle Real Application Cluster 用 Sun Cluster データサービスで Sun StorageTek QFS 共有ファイルシステムを使用する方法についての詳細は、『Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS』を参照してください。

コード例 6-1 に示すように、DID 装置 d4 ~ d8 は可用性が高く、コントローラ制御のストレージに含まれています。Sun Cluster 環境に Sun StorageTek QFS 共有ファイルシステムを構成するには、コントローラ制御のストレージが、RAID-1 または RAID-5 を使用して装置の冗長性をサポートしている必要があります。

この例を単純にするため、2 つのファイルシステムを作成します。

- qfs1—Oracle Real Application Cluster の共有インストール、構成、およびログファイルに使用されます。
- qfs2—Oracle Real Application Cluster ソフトウェアで共有されるデータベースファイルに使用されます。

また、装置 d4 は Sun StorageTek QFS のメタデータに使用されます。この装置には 50G バイトのスライスが 2 つあります。それ以外の装置 d5 ~ d8 は Sun StorageTek QFS のファイルデータに使用されます。

この構成には、次の項で説明するように 5 つの手順があります。

1. Sun StorageTek QFS ファイルシステムの作成準備
2. ファイルシステムの作成と Sun Cluster ノードの構成
3. 構成の検証
4. ネットワークネームサービスの構成
5. Oracle Real Application Cluster 用 Sun Cluster データサービスの構成

▼ Sun StorageTek QFS 共有ファイルシステムの作成を準備する

1. Sun Cluster システムの 1 つのノードから、`format(1M)` ユーティリティーを使用して `/dev/did/dsk/d4` のパーティションをレイアウトします (コード例 6-2)。

この例で、このアクションはノード `scnode-A` から実行しています。

コード例 6-2 `/dev/did/dsk/d4` のパーティションのレイアウト

```
# format /dev/did/rdisk/d4s2
# format> partition
[ 出力省略 ]
# partition> print
Current partition table (unnamed):
Total disk cylinders available: 12800 + 2 (reserved cylinders)

Part      Tag      Flag      Cylinders      Size      Blocks
0         usr      wm        1 - 6400      50.00GB   (6400/0/0) 104857600
1         usr      wm      6401 - 12800   50.00GB   (6400/0/0) 104857600
2      backup  wu         0 - 12800     100.00GB  (6400/0/0) 209715200
3 unassigned wu          0              0          (0/0/0)      0
4 unassigned wu          0              0          (0/0/0)      0
5 unassigned wu          0              0          (0/0/0)      0
6 unassigned wu          0              0          (0/0/0)      0
7 unassigned wu          0              0          (0/0/0)      0
```

注：パーティション 2 (バックアップ) は使用されず、デフォルトで `format(1M)` によって作成されます。

パーティション (スライス) 0 は、ボリュームの VTOC (Volume Table of Contents) に含まれていなかったものが、50G バイトのパーティションとして構成されます。パーティション 1 は、パーティション 0 と同じサイズになるよう構成されます。

2. 同じノードで、`format(1M)` ユーティリティーを使用して `/dev/did/dsk/d5` のパーティションをレイアウトします (コード例 6-3)。

コード例 6-3 /dev/did/dsk/d5 のパーティションのレイアウト

```
# format /dev/did/rdisk/d5s2
# format> partition
[ 出力省略 ]
# partition> print
Current partition table (unnamed):
Total disk cylinders available: 34530 + 2 (reserved cylinders)
```

Part	Tag	Flag	Cylinders	Size	Blocks
0	usr	wm	1 - 34529	269.77GB	(34529/0/0) 565723136
1	usr	wm	0 - 0	0	(0/0/0)
2	backup	wu	0 - 34529	269.77GB	(34530/0/0) 565739520
3	unassigned	wu	0	0	(0/0/0) 0
4	unassigned	wu	0	0	(0/0/0) 0
5	unassigned	wu	0	0	(0/0/0) 0
6	unassigned	wu	0	0	(0/0/0) 0
7	unassigned	wu	0	0	(0/0/0) 0

注：パーティション 2（バックアップ）は使用されず、デフォルトで format(1M) によって作成されます。

3. 同じノードで、装置 d5 のパーティション構成を装置 d6 ~ d8 に複製します。
この例では、装置 d6 用のコマンドを示します。

```
# prtvtoc /dev/did/rdisk/d5s2 | fmthard -s - /dev/did/rdisk/d6s2
```

4. ファイルシステムの潜在的なホストとなるすべてのノードで、次の手順を実行します。
 - a. mcf ファイルに 2 つの新しい構成エントリ (qfs1 と qfs2) を追加して、6 つのパーティションが 2 つの Sun StorageTek QFS 共有ファイルシステムになるよう構成します (コード例 6-4)。

コード例 6-4 mcf ファイルへの構成エントリの追加

```
# cat >> /etc/opt/SUNWsamfs/mcf <<EOF
#
# Sun StorageTek QFS file system configurations
#
# Equipment      Equipment      Equipment      Family      Device      Additional
# Identifier      Ordinal        Type           Set          State        Parameters
# -----
qfs1              100            ma             qfs1         -            shared
/dev/did/dsk/d4s0 101            mm             qfs1         -
/dev/did/dsk/d5s0 102            mr             qfs1         -
/dev/did/dsk/d6s0 103            mr             qfs1         -
qfs2              200            ma             qfs2         -            shared
/dev/did/dsk/d4s1 201            mm             qfs2         -
/dev/did/dsk/d7s0 202            mr             qfs2         -
/dev/did/dsk/d8s0 203            mr             qfs2         -
EOF
```

mcf ファイルの詳細については、36 ページの「mcf ファイルの機能」または『Sun StorageTek QFS インストールおよびアップグレードの手引き』を参照してください。

- b. /etc/opt/SUNWsamfs/samfs.cmd ファイルを編集して、Oracle Real Application Cluster 用 Sun Cluster データサービスに必要なマウントオプションを追加します (コード例 6-5)。

コード例 6-5 samfs.cmd ファイルの例

```
fs = qfs2
stripe = 1
sync_meta = 1
mh_write
qwrite
forcedirectio
rdlease = 300
```

Oracle Real Application Cluster 用 Sun Cluster データサービスに必要なマウントオプションについての詳細は、『Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS』を参照してください。

c. 構成が正しいことを検証します。

この検証は、必ず各ノードで `mcf` ファイルおよび `samfs.cmd` ファイルを構成したあとに行なってください。

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

▼ Sun StorageTek QFS 共有ファイルシステムを作成し、Sun Cluster ノードを構成する

この手順は、作成する各ファイルシステムごとに実行してください。この例では、`qfs1` ファイルシステムの作成方法について説明します。

1. 次のコマンドを使用して、Sun Cluster のプライベートインターコネクト名を調べます。

```
# /usr/cluster/bin/scconf -p | egrep "Cluster node name:|Node private \  
hostname:"  
Cluster node name:                scnode-A  
Node private hostname:            clusternode1-priv  
Cluster node name:                scnode-B  
Node private hostname:            clusternode2-priv
```

2. ファイルシステムの潜在的なホストとなる各ノードで、次の手順を実行します。

- a. `samd(1M) config` コマンドを使用して、新しい Sun StorageTek QFS 構成があることを Sun StorageTek QFS デーモンに知らせます。

```
# samd config
```

- b. 手順 1 で調べた Sun Cluster システムのプライベートインターコネクト名を基に、ファイルシステム (`/etc/opt/SUNWsamfs/hosts.family-set-name`) 用の Sun StorageTek QFS 共有ホストファイルを作成します。
3. 一意の Sun StorageTek QFS 共有ファイルシステムのホスト構成ファイルを編集し、Sun Cluster システムのインターコネクト名を指定します (コード例 6-6)。

Sun Cluster ソフトウェアのフェイルオーバーおよびフェンシング動作として、Sun StorageTek QFS 共有ファイルシステムは Sun Cluster システムと同じインターコネクト名を使用する必要があります。

コード例 6-6 各ファイルシステムのホスト構成ファイルの編集

```
# cat > hosts.qfs1 <<EOF
# File /etc/opt/SUNWsamfs/hosts.qfs1
# Host      Host IP                      Server  Not  Server
# Name      Addresses                          Priority Used Host
# -----
scnode-A    clusternode1-priv                      1      -   server
scnode-B    clusternode2-priv                      2      -
EOF
```

4. Sun Cluster システムの 1 つのノードから、`sammkfs(1M) -S` コマンドを使用して Sun StorageTek QFS 共有ファイルシステムを作成します。

```
# sammkfs -S qfs1 < /dev/null
```

5. ファイルシステムの潜在的なホストとなる各ノードで、次の手順を実行します。

- a. `mkdir(1M)` コマンドを使用してファイルシステムのグローバルマウントポイントを作成し、`chmod(1M)` コマンドを使用して `root` をマウントポイントの所有者にします。また、`chown(1M)` コマンドを使用して、読み取り/書き込みアクセス権 (755) を持っている `other` がマウントポイントを使用できるようにします。

```
# mkdir /global/qfs1
# chmod 755 /global/qfs1
# chown root:other /global/qfs1
```

- b. `/etc/vfstab` ファイルに Sun StorageTek QFS 共有ファイルシステムのエントリを追加します。

```
# cat >> /etc/vfstab <<EOF
# device      device      mount      FS      fsck      mount      mount
# to mount    to fsck     point      type     pass     at boot    options
#
qfs1          -          /global/qfs1  samfs   -        no        shared
EOF
```

▼ 構成の妥当性を検査する

この手順は、作成した各ファイルシステムごとに実行してください。この例では、ファイルシステム `qfs1` の構成の検証方法について説明します。

1. どのノードがファイルシステムのメタデータサーバーとして機能しているか不明な場合は、`samsharefs(1M) -R` コマンドを使用します。

コード例 6-7 で、`qfs1` のメタデータサーバーは `scnode-A` です。

コード例 6-7 メタデータサーバーとなっているノードの確認

```
# samsharefs -R qfs1
#
# Host file for family set 'qfs1'
#
# Version: 4      Generation: 1      Count: 2
# Server = host 1/scnode-A, length = 165
#
scnode-A clusternode2-priv 1 - server
scnode-B clusternode2-priv 2 -
```

2. `mount(1M)` コマンドを使用して、ファイルシステムを最初にメタデータサーバーにマウントしてから、Sun Cluster システムの各ノードにマウントします。

注 – ファイルシステムをメタデータサーバーに最初にマウントすることが重要です。

```
# mount qfs1
# ls /global/qfs1
lost+found/
```

3. `samsharefs(1M) -s` コマンドを実行して Sun StorageTek QFS 共有ファイルシステムをノード間で変更し、任意のフェイルオーバーを検証します。

```
# samsharefs -s scnode-B qfs1
# ls /global/qfs1
lost+found/
# samsharefs -s scnode-A qfs1
# ls /global/qfs1
lost+found
```

4. 必要な Sun Cluster のリソースタイプがリソース構成に追加されているかどうか検証します。

```
# scrgadm -p | egrep "SUNW.qfs"
```

5. Sun Cluster のリソースタイプが見つからない場合は、`scrgadm(1M) -a -t` コマンドを使用してリソース構成に追加します。

```
# scrgadm -a -t SUNW.qfs
```

6. SUNW.qfs リソースタイプを登録および構成します。

```
# scrgadm -a -g qfs-rg -h scnode-A,scnode-B
# scrgadm -a -g qfs-rg -t SUNW.qfs -j qfs-res \
-x QFSFileSystem=/global/qfs1,/global/qfs2
```

7. `scswitch(1M) -Z -g` コマンドを使用して、リソースグループをオンラインにします。

```
# scswitch -Z -g qfs-rg
```

8. リソースグループが、構成されたすべてのノードで機能することを確認します。

```
# scswitch -z -g qfs-rg -h scnode-B
# scswitch -z -g qfs-rg -h scnode-A
```

▼ Oracle Real Application Cluster 用 Sun Cluster データサービスを構成する

ここでは、Sun StorageTek QFS 共有ファイルシステムで使用する Oracle Real Application Cluster 用のデータサービスを構成する方法の例を示します。詳細については、『Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS』を参照してください。

1. 『Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS』の説明に従って、データサービスをインストールします。
2. Sun StorageTek QFS 共有ファイルシステムをマウントします。
3. ファイルシステムに正しい所有権とアクセス権を設定し、Oracle データベースが正常に動作するようにします。

```
# chown oracle:dba /global/qfs1 /global/qfs2
# chmod 755 /global/qfs1 /global/qfs2
```

4. oracle ユーザーとして、Oracle Real Application Cluster のインストールとデータベースファイルに必要なサブディレクトリを作成します。

```
$ id
uid=120(oracle) gid=520(dba)
$ mkdir /global/qfs1/oracle_install
$ mkdir /global/qfs2/oracle_db
```

Oracle Real Application Cluster のインストールでは、Oracle の動作に使用される ORACLE_HOME 環境変数の値として /global/qfs1/oracle_install ディレクトリパスが使用されます。Oracle Real Application Cluster のデータベースファイルのパスは、接頭辞として /global/qfs2/oracle_db ディレクトリパスが付きます。

5. Oracle Real Application Cluster ソフトウェアをインストールします。

インストール時に、手順 4 で定義したインストール用パス (/global/qfs1/oracle_install) を指定します。

6. Oracle Real Application Cluster のデータベースを作成します。

データベース作成時に、データベースファイルを qfs2 共有ファイルシステムに配置することを指定します。

7. Oracle Real Application Cluster データベースインスタンスの起動と停止を自動化する場合は、リソースグループとリソースの必要な依存性が設定されていることを確認します。

詳細については、『Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS』を参照してください。

注 – Oracle Real Application Cluster データベースインスタンスの起動と停止を自動化する予定の場合は、Sun Cluster ソフトウェア version 3.1 9/04 または互換バージョンを使用する必要があります。

注 – Oracle RAC 10g 構成用に使用されている共有 Sun StorageTek QFS 構成では、クラスタレディサービス (CRS) の Oracle インストーラが root.sh を実行するようユーザーに促すとき、このコマンドは失敗する場合があります。また、OCR (Oracle Cluster Registry) ファイルが root.sh によって作成された場合は、CRS レジストリが不安定になることもあります。

回避策は、OCR ファイルが 700416 より大きくなるように事前割り当てをしておくことです。たとえば、root.sh を実行する前に、次のように、ユーザー oracle として 1M バイトのファイルを割り当てておきます。

```
$ dd if=/dev/zero of=<OCR file path> bs=1024k count=1M
```

Sun Cluster 環境での非共有ファイルシステムの構成

非共有 Sun StorageTek QFS ファイルシステムを Sun Cluster システムにインストールするときは、ファイルシステムを Sun Cluster の HAStoragePlus リソースタイプで高可用性 (HA) に構成します。通常、Sun Cluster システム上の非共有 Sun StorageTek QFS ファイルシステムは、高可用性ネットワークファイルサーバー (HA-NFS) や高可用性 ORACLE (HA-ORACLE) などの 1 つまたは複数のフェイルオーバーアプリケーションに関連付けられます。非共有 Sun StorageTek QFS ファイルシステムとフェイルオーバーアプリケーションはどちらも単一のリソースグループで有効となり、そのリソースグループは一度に 1 つの Sun Cluster ノードで有効になります。

非共有 Sun StorageTek QFS ファイルシステムは、一度に 1 つのノードにマウントされます。Sun Cluster の障害モニターでエラーが検出されるか、リソースグループを切り替えると、非共有 Sun StorageTek QFS ファイルシステムおよびそれに関連する HA アプリケーションは、リソースグループがどのように構成されていたかに応じて、別のノードにフェイルオーバーします。

Sun Cluster のグローバル装置グループ (/dev/global/*) に含まれるすべてのファイルシステムは、HAStoragePlus リソースタイプで使用できます。ファイルシステムが HAStoragePlus リソースタイプで構成されていると、Sun Cluster リソースグループの一部となり、Sun Cluster の Resource Group Manager (RGM) で制御されるファイルシステムは、リソースグループが有効なノードでローカルにマウントされます。RGM がリソースグループを切り替えるか、別の構成済み Sun Cluster ノードへフェイルオーバーすると、非共有 Sun StorageTek QFS ファイルシステムは現在のノードからマウント解除されて、新しいノードに再マウントされます。

各非共有 Sun StorageTek QFS ファイルシステムには、最低 2 つの raw ディスクパーティションまたはボリュームマネージャーで制御されるボリューム (Solstice DiskSuite™/Solaris ボリュームマネージャーまたは VERITAS Volume Manager) が必要で、1 つは Sun StorageTek QFS メタデータ (i ノード) 用、もう 1 つは Sun StorageTek QFS ファイルデータ用です。複数のデータパスで、複数のパーティションまたは複数のディスクにまたがるボリュームを作成すると、非共有 Sun StorageTek QFS ファイルシステムのパフォーマンスが向上します。メタデータおよびファイルデータのパーティションのサイズ設定についての詳細は、7 ページの「設計の基本」を参照してください。

ここでは、非共有 Sun StorageTek QFS ファイルシステムを使用する Sun Cluster システム構成の 3 つの例を示します。この例の中で、ファイルシステムは次の HA-NFS ファイルマウントポイントの組み合わせで構成されます。

- raw グローバル装置 (例 1)
- Solstice DiskSuite/Solaris ボリュームマネージャーで制御されるボリューム (例 2)

■ VERITAS Volume Manager (VxVM) で制御されるボリューム (例 3)

これらの構成を簡単にするため、各ファイルシステムの 10% が Sun StorageTek QFS のメタデータに使用され、残りの容量が Sun StorageTek QFS のファイルデータに使用されます。サイズ設定とディスクレイアウトの検討事項については、『Sun StorageTek QFS インストールおよびアップグレードの手引き』を参照してください。

例 1: raw グローバル装置上の HA-NFS

この例は、raw グローバル装置上の HA-NFS での非共有 Sun StorageTek QFS ファイルシステムの構成方法を示しています。この構成では、raw グローバル装置がコントローラ制御のストレージに含まれている必要があります。このコントローラ制御のストレージは、RAID-1 または RAID-5 を使用した装置の冗長性をサポートしている必要があります。

コード例 6-1 に示すように、この例で使用される DID 装置 d4 ~ d7 は可用性が高く、コントローラ制御のストレージに含まれています。HAStoragePlus リソースタイプにはグローバル装置を使用する必要があるため、各 DID 装置 (/dev/did/dsk/dx) は、/dev/global/dsk/dx という構文を使用してグローバル装置からアクセス可能です。

この例の主要な手順は次のとおりです。

1. 非共有ファイルシステムの作成準備
2. ファイルシステムの作成と Sun Cluster ノードの構成
3. ネットワークネームサービスと IPMP (IP Measurement Protocol) の検証テストの構成
4. HA-NFS の構成と高可用ファイルシステムの構成

▼ 非共有 Sun StorageTek QFS ファイルシステムの作成を準備する

1. format(1M) ユーティリティーを使用して /dev/global/dsk/d4 のパーティションをレイアウトします。

```
# format /dev/global/rdisk/d4s2
# format> partition
[ 出力省略 ]
# partition> print
Current partition table (original):
Total disk cylinders available: 34530 + 2 (reserved cylinders)
Part      Tag      Flag      Cylinders      Size      Blocks
0  unassigned  wm      1 - 3543      20.76GB  (3543/0/0)  43536384
1  unassigned  wm    3544 - 34529  181.56GB  (30986/0/0) 380755968
2  backup      wu      0 - 34529    202.32GB  (34530/0/0) 424304640
3  unassigned  wu      0              0          (0/0/0)      0
4  unassigned  wu      0              0          (0/0/0)      0
5  unassigned  wu      0              0          (0/0/0)      0
6  unassigned  wu      0              0          (0/0/0)      0
7  unassigned  wu      0              0          (0/0/0)      0
```

注：パーティション 2（バックアップ）は使用されず、デフォルトで format(1m) によって作成されます。

パーティション (スライス) 0 は、ボリュームの VTOC (Volume Table of Contents) に含まれていなかったものが、20G バイトのパーティションとして構成されます。残りの容量は、パーティション 1 として構成されます。

2. グローバル装置 d4 のパーティション構成をグローバル装置 d5 ~ d7 に複製します。

この例では、グローバル装置 d5 用のコマンドを示します。

```
# prtvtoc /dev/global/rdisk/d4s2 | fmthard \
-s - /dev/global/rdisk/d5s2
```

3. ファイルシステムの潜在的なホストとなるすべてのノードで、次の手順を実行します。

- a. mcf ファイルに新しいファイルシステムエントリを追加して、8 つのパーティション (4 つのグローバル装置でそれぞれ 2 つのパーティション) を Sun StorageTek QFS ファイルシステムに構成します。

```
# cat >> /etc/opt/SUNWsamfs/mcf <<EOF
#
# Sun StorageTek QFS file system configurations
#
# Equipment      Equipment      Equipment      Family      Device      Additional
# Identifier      Ordinal        Type           Set         State       Parameters
# -----
qfsnfs1          100            ma             qfsnfs1     on
/dev/global/dsk/d4s0 101            mm             qfsnfs1
/dev/global/dsk/d5s0 102            mm             qfsnfs1
/dev/global/dsk/d6s0 103            mm             qfsnfs1
/dev/global/dsk/d7s0 104            mm             qfsnfs1
/dev/global/dsk/d4s1 105            mr             qfsnfs1
/dev/global/dsk/d5s1 106            mr             qfsnfs1
/dev/global/dsk/d6s1 107            mr             qfsnfs1
/dev/global/dsk/d7s1 108            mr             qfsnfs1
EOF
```

mcf ファイルの詳細については、36 ページの「mcf ファイルの機能」を参照してください。

- b. mcf ファイルに追加した構成情報が正しいかどうか検証し、mcf ファイルにエラーが見つかった場合は修正してから先に進みます。
- Sun StorageTek QFS ファイルシステムを HAStoragePlus リソースタイプで構成する前に、この手順を完了しておくことが重要です。

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

▼ Sun StorageTek QFS ファイルシステムを作成し、Sun Cluster ノードを構成する

1. ファイルシステムの潜在的ホストとなる各ノードで、samd(1M) config コマンドを実行します。

このコマンドは、新しい Sun StorageTek QFS 構成ファイルがあることを Sun StorageTek QFS デーモンに知らせます。

```
# samd config
```

2. Sun Cluster システムの 1 つのノードから、`sammkfs(1M)` コマンドを使用してファイルシステムを作成します。

```
# sammkfs qfsnfs1 < /dev/null
```

3. ファイルシステムの潜在的なホストとなる各ノードで、次の手順を実行します。

- a. `mkdir(1M)` コマンドを使用してファイルシステムのグローバルマウントポイントを作成し、`chmod(1M)` コマンドを使用して `root` をマウントポイントの所有者にします。また、`chown(1M)` コマンドを使用して、読み取り/書き込みアクセス権 (755) を持っている `other` がマウントポイントを使用できるようにします。

```
# mkdir /global/qfsnfs1
# chmod 755 /global/qfsnfs1
# chown root:other /global/qfsnfs1
```

- b. `/etc/vfstab` ファイルに Sun StorageTek QFS ファイルシステムのエントリを追加します。

「mount options」フィールドに、`sync_meta=1` という値があることに注意してください。

```
# cat >> /etc/vfstab <<EOF
# device      device      mount      FS      fsck      mount      mount
# to mount    to fsck     point      type    pass     at boot    options
#
qfsnfs1      -          /global/qfsnfs1  samfs   2        no        sync_meta=1
EOF
```

- c. ファイルシステムをマウントおよびマウント解除して、構成を検証します。

```
# mount qfsnfs1
# ls /global/qfsnfs1
lost+found/
# umount qfsnfs1
```

4. `scrgadm(1M) -p | egrep` コマンドを使用して、必要な Sun Cluster リソースタイプがリソース構成に追加されていることを検証します。

```
# scrgadm -p | egrep "SUNW.HAStoragePlus|SUNW.LogicalHostname|SUNW.nfs"
```

5. 必要な Sun Cluster のリソースタイプが見つからない場合は、`scrgadm(1M) -a -t` コマンドを使用して構成に追加します。

```
# scrgadm -a -t SUNW.HAStoragePlus
# scrgadm -a -t SUNW.LogicalHostname
# scrgadm -a -t SUNW.nfs
```

▼ ネットワークネームサービスと IPMP 妥当性検査を構成する

ここでは、Sun Cluster ノード用のネットワークネームサービスと IPMP 妥当性検査の構成方法の例を示します。詳細は、『Sun Cluster Software Installation Guide for Solaris OS』、『Solaris のシステム管理 (IP サービス)』および『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』を参照してください。

1. `vi` などのテキストエディタを使用して、ノード名として Sun Cluster システムおよびファイルを参照するよう `/etc/nsswitch.conf` ファイルを編集します。
この手順は、NIS (Network Information Name サービス) サーバーを構成する前に実行します。

```

# cat /etc/nsswitch.conf
#
# /etc/nsswitch.nis:
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# uses NIS (YP) in conjunction with files.
#
# the following two lines obviate the "+" entry in /etc/passwd and /etc/group.
passwd:    files nis
group:     files nis

# Cluster s/w and local /etc/hosts file take precedence over NIS
hosts:     cluster files nis [NOTFOUND=return]
ipnodes:   files

# Uncomment the following line and comment out the above to resolve
# both IPv4 and IPv6 addresses from the ipnodes databases. Note that
# IPv4 addresses are searched in all of the ipnodes databases before
# searching the hosts databases. Before turning this option on, consult
# the Network Administration Guide for more details on using IPv6.
# ipnodes: nis [NOTFOUND=return] files

networks:  nis[NOTFOUND=return] files
protocols: nis [NOTFOUND=return] files
rpc:       nis[NOTFOUND=return] files
ethers:    nis[NOTFOUND=return] files
netmasks: nis[NOTFOUND=return] files
bootparams: nis[NOTFOUND=return] files
publickey: nis[NOTFOUND=return] files

netgroup:  nis

automount: files nis
aliases:   files nis
[ これ以降のファイル内容は省略 ]

```

2. /etc/nsswitch.conf ファイルに加えた変更が正しいことを確認します。

```

# grep '^hosts:' /etc/nsswitch.conf
hosts:    cluster files nis [NOTFOUND=return]
#

```

3. 使用可能なネットワークアダプタを使用して、IPMP 妥当性検査を設定します。
例としてアダプタ qfe2 と qfe3 を使用します。

a. 各アダプタの IPMP テストアドレスを静的に構成します。

```
# cat >> /etc/hosts << EOF
#
# Test addresses for scnode-A
#
192.168.2.2      \uname -n'-qfe2
192.168.2.3      \uname -n'-qfe2-test
192.168.3.2      \uname -n'-qfe3
192.168.3.3      \uname -n'-qfe3-test
#
# Test addresses for scnode-B
#
192.168.2.4      \uname -n'-qfe2
192.168.2.5      \uname -n'-qfe2-test
192.168.3.4      \uname -n'-qfe3
192.168.3.5      \uname -n'-qfe3-test
EOF
```

b. IPMP アダプタを動的に構成します。

```
# ifconfig qfe2 plumb \uname -n'-qfe2-test netmask + broadcast + deprecated \
-failover -standby group ipmp0 up
# ifconfig qfe2 addif \uname -n'-qfe2 up
# ifconfig qfe3 plumb \uname -n'-qfe3-test netmask + broadcast + deprecated \
-failover -standby group ipmp0 up
# ifconfig qfe3 addif \uname -n'-qfe3 up
```

c. 構成を確認します。

```
# cat > /etc/hostname.qfe2 << EOF
\uname -n'-qfe2-test netmask + broadcast + deprecated -failover -standby \
group ipmp0 up addif \uname -n'-qfe2 up
EOF

# cat > /etc/hostname.qfe3 << EOF
\uname -n'-qfe3-test netmask + broadcast + deprecated -failover -standby \
group ipmp0 up addif \uname -n'-qfe3 up
EOF
```

▼ HA-NFS を構成し、高可用 Sun StorageTek QFS ファイルシステムを構成する

ここでは、HA-NFS の構成方法の例を示します。HA-NFS についての詳細は、『Sun Cluster Data Service for Network File System (NFS) Guide for Solaris OS』および NFS のマニュアルを参照してください。

1. Sun StorageTek QFS ファイルシステム用の NFS 共有ポイントを作成します。

共有ポイントは、Sun StorageTek QFS ファイルシステムではなく、/global ファイルシステムに含まれている必要があります。

```
# mkdir -p /global/nfs/SUNW.nfs
# echo "share -F nfs -o rw /global/qfsnfs1" > \
/global/nfs/SUNW.nfs/dfstab.nfs1-res
```

2. NFS リソースグループを作成します。

```
# scrgadm -a -g nfs-rg -y PathPrefix=/global/nfs
```

3. サイトでのアドレスを使用して、/etc/hosts テーブルに NFS 論理ホストを追加します。

```
# cat >> /etc/hosts << EOF
#
# IP Addresses for LogicalHostnames
#
192.168.2.10      lh-qfs1
EOF
```

4. `scrgadm(1M) -a -L -g` コマンドを使用して、NFS リソースグループに論理ホストを追加します。

```
# scrgadm -a -L -g nfs-rg -l lh-nfs1
```

5. `scrgadm(1M) -c -g` コマンドを使用して、HAStoragePlus リソースタイプを構成します。

```
# scrgadm -c -g nfs-rg -h scnode-A,scnode-B
# scrgadm -a -g nfs-rg -j qfsnfs1-res -t SUNW.HAStoragePlus \
-x FilesystemMountPoints=/global/qfsnfs1 \
-x FilesystemCheckCommand=/bin/true
```

6. リソースグループをオンラインにします。

```
# scswitch -Z -g nfs-rg
```

7. NFS リソースタイプを構成し、HAStoragePlus リソースに依存関係を設定します。

```
# scrgadm -a -g nfs-rg -j nfs1-res -t SUNW.nfs -y \  
Resource_dependencies=qfsnfs1-res
```

8. NFS リソースグループをオンラインにします。

```
# scswitch -e -j nfs1-res
```

これで、NFS リソース /net/lh-nfs1/global/qfsnfs1 が完全に構成され、高可用になりました。

9. 高可用 NFS ファイルシステムを Sun StorageTek QFS ファイルシステム上で使用できるようにする前に、リソースグループがすべての構成済みノード間でエラーなく切り替えることができ、オンラインにしたり、オフラインにできることをテストして確認します。

```
# scswitch -z -g nfs-rg -h scnode-A  
# scswitch -z -g nfs-rg -h scnode-B  
# scswitch -F -g nfs-rg  
# scswitch -Z -g nfs-rg
```

例 2: Solstice DiskSuite/Solaris ボリュームマネージャーで制御されるボリューム上の HA-NFS

この例は、Solstice DiskSuite/Solaris ボリュームマネージャーソフトウェアで制御されるボリューム上の HA-NFS での、非共有 Sun StorageTek QFS ファイルシステムの構成方法を示しています。この構成では、RAID-1 または RAID-5 ボリュームを使用する冗長なコントローラ制御のストレージに、DID 装置を含めるかどうかを決定できます。通常、Solaris ボリュームマネージャーは基礎となるコントローラ制御のストレージが冗長でない場合だけ使用されます。

コード例 6-1 に示すように、この例で使用される DID 装置 d4 ~ d7 は可用性が高く、コントローラ制御のストレージに含まれています。Solaris ボリュームマネージャーでは、Solaris ボリュームマネージャーがボリュームを構成する raw 装置の割り当てに、DID 装置が使用されている必要があります。Solaris ボリュームマネージャーはグローバルにアクセス可能なディスクグループを作成し、それが HAStoragePlus リソースタイプで Sun StorageTek QFS ファイルシステムの作成に使用されます。

この例には次の手順があります。

1. Solstice DiskSuite/Solaris ボリュームマネージャーソフトウェアの準備
2. 非共有ファイルシステムの作成準備
3. ファイルシステムの作成と Sun Cluster ノードの構成
4. ネットワークネームサービスと IPMP 妥当性検査の構成
5. HA-NFS の構成と高可用ファイルシステムの構成

▼ Solstice DiskSuite/Solaris ボリュームマネージャーソフトウェアを準備する

1. Solaris ボリュームマネージャーのメタデータベース (metadb) が、Sun StorageTek QFS ファイルシステムの潜在的ホストとなる各ノード上ですでに構成されていることを確認します。

```
# metadb
```

	flags	first blk	block count	
a	m p luo	16	8192	/dev/dsk/c0t0d0s7
a	p luo	16	8192	/dev/dsk/c1t0d0s7
a	p luo	16	8192	/dev/dsk/c2t0d0s7

metadb(1M) コマンドでメタデータベース構成が返されなかった場合は、各ノードで、1 つまたは複数のローカルディスクに 3 つ以上のデータベース複製を作成します。各複製は、サイズを 16M バイト以上にする必要があります。メタデータベース複製の作成方法についての詳細は、『Sun Cluster Software Installation Guide for Solaris OS』を参照してください。

2. この Sun StorageTek QFS ファイルシステムのすべての Solaris ボリュームマネージャーボリュームを含むよう、HA-NFS ディスクグループを作成します。

```
# metaset -s nfsdg -a -h scnode-A scnode-B
```

3. DID 装置 d4 ~ d7 を、Solaris ボリュームマネージャーがボリュームを作成する raw 装置のプールに追加します。

```
# metaset -s nfsdg -a /dev/did/dsk/d4 /dev/did/dsk/d5 \  
/dev/did/dsk/d6 /dev/did/dsk/d7
```

▼ Sun StorageTek QFS ファイルシステムを準備する

1. `format(1M)` ユーティリティを使用して `/dev/global/dsk/d4` のパーティションをレイアウトします。

```
# format /dev/global/rdisk/d4s2
# format> partition
[ 出力省略 ]
# partition> print
Current partition table (original):
Total disk cylinders available: 34530 + 2 (reserved cylinders)
Part      Tag      Flag      Cylinders      Size      Blocks
0  unassigned  wm      1 - 3543      20.76GB  (3543/0/0)  43536384
1  unassigned  wm    3544 - 34529  181.56GB  (30986/0/0) 380755968
2  backup      wu      0 - 34529    202.32GB  (34530/0/0) 424304640
3  unassigned  wu      0              0          (0/0/0)      0
4  unassigned  wu      0              0          (0/0/0)      0
5  unassigned  wu      0              0          (0/0/0)      0
6  unassigned  wu      0              0          (0/0/0)      0
7  unassigned  wu      0              0          (0/0/0)      0
```

注：パーティション 2（バックアップ）は使用されず、デフォルトで `format(1m)` によって作成されます。

この例では、パーティション(スライス)0は、ボリュームの VTOC (Volume Table of Contents) に含まれていなかったものが、20G バイトのパーティションとして構成されたことを示しています。残りの容量は、パーティション 1 として構成されます。

2. DID 装置 `d4` のパーティション構成を DID 装置 `d5 ~ d7` に複製します。

この例では、装置 `d5` 用のコマンドを示します。

```
# prtvtoc /dev/global/rdisk/d4s2 | fmthard \
-s - /dev/global/rdisk/d5s2
```

3. 8 つのパーティション (4 つの DID 装置にそれぞれ 2 つのパーティション) で、2 つの RAID-1 (ミラー化) Sun StorageTek QFS メタデータボリュームおよび 2 つの RAID-5 (パリティストライプ化) Sun StorageTek QFS ファイルデータボリュームを構成します。

- a. これら 4 つのドライブのパーティション (スライス) 0 を 2 つの RAID-1 セットにします。

```
# metainit -s nfsdg -f d1 1 1 /dev/did/dsk/d4s0
# metainit -s nfsdg -f d2 1 1 /dev/did/dsk/d5s0
# metainit -s nfsdg d10 -m d1 d2
# metainit -s nfsdg -f d3 1 1 /dev/did/dsk/d6s0
# metainit -s nfsdg -f d4 1 1 /dev/did/dsk/d7s0
# metainit -s nfsdg d11 -m d3 d4
```

- b. これら 4 つのドライブのパーティション 1 を 2 つの RAID-5 セットにします。

```
# metainit -s nfsdg d20 -p /dev/did/dsk/d4s1 205848574b
# metainit -s nfsdg d21 -p /dev/did/dsk/d5s1 205848574b
# metainit -s nfsdg d22 -p /dev/did/dsk/d6s1 205848574b
# metainit -s nfsdg d23 -p /dev/did/dsk/d7s1 205848574b
# metainit -s nfsdg d30 -r d20 d21 d22 d23
```

- c. ファイルシステムの潜在的ホストとなる各ノードで、Sun StorageTek QFS ファイルシステムのエントリを mcf ファイルに追加します。

```
# cat >> /etc/opt/SUNWsamfs/mcf <<EOF

# Sun StorageTek QFS file system configurations
#
# Equipment          Equipment   Equipment   Family   Device   Additional
# Identifier          Ordinal    Type        Set      State    Parameters
# -----
qfsnfs1              100        ma          qfsnfs1  on
/dev/md/nfsdg/dsk/d10 101        mm          qfsnfs1
/dev/md/nfsdg/dsk/d11 102        mm          qfsnfs1
/dev/md/nfsdg/dsk/d30 103        mr          qfsnfs1
EOF
```

mcf ファイルの詳細については、36 ページの「mcf ファイルの機能」を参照してください。

4. 各ノードで mcf(4) の構成が正しいかどうか検証し、エラーが見つかった場合は mcf ファイルを修正してから先に進みます。

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

▼ Sun StorageTek QFS ファイルシステムを作成し、Sun Cluster ノードを構成する

1. ファイルシステムの潜在的ホストとなる各ノードで、`samd(1M) config` コマンドを使用します。

このコマンドは、新しい Sun StorageTek QFS 構成ファイルがあることを Sun StorageTek QFS デーモンに知らせます。

```
# samd config
```

2. Solaris ボリュームマネージャーの、ディスクグループの調停検出を有効にします。この機能は、Sun Cluster システムによるドライブエラーの検出に役立ちます。

```
# metaset -s nfsdg -a -m scnode-A  
# metaset -s nfsdg -a -m scnode-B
```

3. ファイルシステムの潜在的ホストとなる各ノードで、NFS ディスクグループがあることを確認します。

```
# metaset -s nfsdg -t
```

4. Sun Cluster システムの 1 つのノードから、`sammkfs(1M)` コマンドを使用してファイルシステムを作成します。

```
# sammkfs qfsnfs1 < /dev/null
```

5. ファイルシステムの潜在的なホストとなる各ノードで、次の手順を実行します。
 - a. `mkdir(1M)` コマンドを使用してファイルシステムのグローバルマウントポイントを作成し、`chmod(1M)` コマンドを使用して `root` をマウントポイントの所有者にします。また、`chown(1M)` コマンドを使用して、読み取り/書き込みアクセス権 (755) を持っている `other` がマウントポイントを使用できるようにします。

```
# mkdir /global/qfsnfs1  
# chmod 755 /global/qfsnfs1  
# chown root:other /global/qfsnfs1
```

- b. /etc/vfstab ファイルに Sun StorageTek QFS ファイルシステムのエントリを追加します。

「mount options」フィールドに、sync_meta=1 という値があることに注意してください。

```
# cat >> /etc/vfstab << EOF
# device      device      mount      FS      fsck      mount      mount
# to mount    to fsck     point      type    pass     at boot    options
#
qfsnfs1      -      /global/qfsnfs1  samfs   2        no        sync_meta=1
EOF
```

- c. ファイルシステムをマウントおよびマウント解除して、構成を検証します。

この手順は、ノード 1 つずつで実行してください。この例では、qfsnfs1 ファイルシステムが 1 つのノードにマウントおよびマウント解除されます。

```
# mount qfsnfs1
# ls /global/qfsnfs1
lost+found/
# umount qfsnfs1
```

注 - マウントポイントをテストするときは、metaset -r (開放) および -t (取得) コマンドを使用して、Sun Cluster ノード間で nfsdg ディスクグループを移動します。次に、samd(1M) config コマンドを使用して、構成の変更をデーモンに警告します。

6. scrgadm(1M) -p | egrep コマンドを使用して、必要な Sun Cluster リソースタイプがリソース構成に追加されていることを検証します。

```
# scrgadm -p | egrep "SUNW.HAStoragePlus|SUNW.LogicalHostname|SUNW.nfs"
```

7. 必要な Sun Cluster リソースタイプが見つからない場合は、次の 1 つまたは複数のコマンドで追加します。

```
# scrgadm -a -t SUNW.HAStoragePlus
# scrgadm -a -t SUNW.LogicalHostname
# scrgadm -a -t SUNW.nfs
```

▼ ネットワークネームサービスと IPMP 妥当性検査を構成する

ネットワークネームサービスと IPMP 妥当性検査を構成するには、151 ページの「ネットワークネームサービスと IPMP 妥当性検査を構成する」の説明に従います。

▼ HA-NFS を構成し、高可用 Sun StorageTek QFS ファイルシステムを構成する

ここでは、HA-NFS の構成方法の例を示します。HA-NFS についての詳細は、『Sun Cluster Data Service for Network File System (NFS) Guide for Solaris OS』および NFS のマニュアルを参照してください。

1. Sun StorageTek QFS ファイルシステム用の NFS 共有ポイントを作成します。

共有ポイントは、Sun StorageTek QFS ファイルシステムではなく、/global ファイルシステムに含まれている必要があります。

```
# mkdir -p /global/nfs/SUNW.nfs
# echo "share -F nfs -o rw /global/qfsnfs1" > \
/global/nfs/SUNW.nfs/dfstab.nfs1-res
```

2. NFS リソースグループを作成します。

```
# scrgadm -a -g nfs-rg -y PathPrefix=/global/nfs
```

3. NFS リソースグループに論理ホストを追加します。

```
# scrgadm -a -L -g nfs-rg -l lh-nfs1
```

4. HAStoragePlus リソースタイプを構成します。

```
# scrgadm -c -g nfs-rg -h scnode-A,scnode-B
# scrgadm -a -g nfs-rg -j qfsnfs1-res -t SUNW.HAStoragePlus \
-x FilesystemMountPoints=/global/qfsnfs1 \
-x FilesystemCheckCommand=/bin/true
```

5. リソースグループをオンラインにします。

```
# scswitch -Z -g nfs-rg
```

6. NFS リソースタイプを構成し、HAStoragePlus リソースに依存関係を設定します。

```
# scrgadm -a -g nfs-rg -j nfs1-res -t SUNW.nfs -y \  
Resource_dependencies=qfsnfs1-res
```

7. scswitch(1M) -e -j コマンドを使用して、NFS リソースをオンラインにします。

```
# scswitch -e -j nfs1-res
```

これで、NFS リソース /net/lh-nfs1/global/qfsnfs1 が完全に構成され、高可用になりました。

8. 高可用 NFS ファイルシステムを Sun StorageTek QFS ファイルシステム上で使用できるようにする前に、リソースグループがすべての構成済みノード間でエラーなく切り替えることができ、オンラインにしたり、オフラインにできることをテストして確認します。

```
# scswitch -z -g nfs-rg -h scnode-A  
# scswitch -z -g nfs-rg -h scnode-B  
# scswitch -F -g nfs-rg  
# scswitch -Z -g nfs-rg
```

例 3: VxVM ボリューム上の HA-NFS

この例は、VERITAS Volume Manager で制御されるボリューム (VxVM ボリューム) 上の HA-NFS での、非共有 Sun StorageTek QFS ファイルシステムの構成方法を示しています。この構成では、RAID-1 または RAID-5 を使用する冗長なコントローラ制御のストレージに、DID 装置を含めるかどうかを決定できます。通常、VxVM は基礎となるストレージが冗長でない場合だけ使用されます。

コード例 6-1 に示すように、この例で使用される DID 装置 d4 ~ d7 は可用性が高く、コントローラ制御のストレージに含まれています。VxVM では、VxVM がボリュームを構成する raw 装置の割り当てに、共有 DID 装置が使用されている必要があります。VxVM は、ディスクグループを Sun Cluster 装置グループとして登録することで、可用性の高いディスクグループを作成します。このようなディスクグループはグローバルにアクセスできませんが、フェイルオーバーが可能で、少なくとも 1 つのノードからアクセス可能です。ディスクグループは、HAStoragePlus リソースタイプで使用できます。

注 - VxVM は独立した追加パッケージになっているため、インストール、パッチ適用、およびライセンスが必要です。VxVM のインストールについての詳細は、VxVM Volume Manager のマニュアルを参照してください。

VxVM で Sun StorageTek QFS ソフトウェアを使用するには、次の VxVM パッケージをインストールする必要があります。

- VRTSvlic
- VRTSvm doc
- VRTSvmm an
- VRTSvmpr o
- VRTSvxvm
- VRTSob および VRTSobgui (オプションのブラウザインタフェースパッケージ)

この例には次の手順があります。

1. VxVM ソフトウェアの構成
2. 非共有ファイルシステムの作成準備
3. ファイルシステムの作成と Sun Cluster ノードの構成
4. 構成の検証
5. ネットワークネームサービスと IPMP 妥当性検査の構成
6. HA-NFS の構成と高可用ファイルシステムの構成

▼ VxVM ソフトウェアを構成する

この節では、VxVM ソフトウェアを Sun StorageTek QFS ソフトウェアと組み合わせて使用するための構成方法の例を示します。VxVM ソフトウェアについての詳細は、VxVM のマニュアルを参照してください。

1. VERITAS 用 DMP (動的マルチパス処理) の状態を確認します。

```
# vxddmpadm listctlr all
```

2. `scdidadm(1M)` ユーティリティを使用して、VxVM で使用する物理装置の HBA コントローラ番号を調べます。

次の例に示すように、マルチノードアクセス可能なストレージは、HBA コントローラ `c6` を使用する `scnode-A` と、コントローラ `c7` を使用する `scnode-B` が利用できます。

```
# scdidadm -L  
[ 出力の一部省略 ]  
4 scnode-A:/dev/dsk/c6t60020F20000037D13E26595500062F06d0 /dev/did/dsk/d4  
4 scnode-B:/dev/dsk/c7t60020F20000037D13E26595500062F06d0 /dev/did/dsk/d4
```

3. VxVM を使用して、使用可能なすべてのストレージをコントローラ `c6` 経由でアクセスできるように構成します。

```
# vxdmpadm getsubpaths ctrl=c6
```

4. このコントローラのすべての装置を、VxVM で制御します。

```
# vxdiskadd fabric_
```

5. ディスクグループを作成し、ボリュームを作成してから、新しいディスクグループを起動します。

```
# /usr/sbin/vxdg init qfs-dg qfs-dg00=disk0 \  
qfsdg01=disk1 qfsdg02=disk2 qfsdg03=disk3
```

6. 前に起動したディスクグループが、このシステム上で有効なことを確認します。

```
# vxdg import nfsdg  
# vxdg free
```

7. Sun StorageTek QFS メタデータ用の 2 つのミラー化ボリュームと、Sun StorageTek QFS ファイルデータボリューム用の 2 つのボリュームを構成します。

このミラー化動作はバックグラウンドプロセスとして実行され、完了までにある程度の時間を要します。

```
# vxassist -g nfsdg make m1 10607001b
# vxassist -g nfsdg mirror m1&
# vxassist -g nfsdg make m2 10607001b
# vxassist -g nfsdg mirror m2&
# vxassist -g nfsdg make m10 201529000b
# vxassist -g nfsdg mirror m10&
# vxassist -g nfsdg make m11 201529000b
# vxassist -g nfsdg mirror m11&
```

8. 前に作成した VxVM ディスクグループを、Sun Cluster 制御のディスクグループとして構成します。

```
# scconf -a -D type=vxvm,name=nfsdg,nodelist=scnode-A:scnode-B
```

▼ Sun StorageTek QFS ファイルシステムの作成を準備する

この手順は、ファイルシステムの潜在的なホストとなる各ノードで実行します。

1. mcf ファイルに Sun StorageTek QFS ファイルシステムのエントリを追加します。

コード例 6-8 mcf ファイルへのファイルシステムの追加

```
# cat >> /etc/opt/SUNWsamfs/mcf <<EOF
# Sun StorageTek QFS file system configurations
#
# Equipment      Equipment  Equipment  Family     Device     Additional
# Identifier      Ordinal    Type        Set        State      Parameters
# -----
qfsnfs1          100        ma          qfsnfs1    on
/dev/vx/dsk/nfsdg/m1    101        mm          qfsnfs1
/dev/vx/dsk/nfsdg/m2    102        mm          qfsnfs1
/dev/vx/dsk/nfsdg/m10   103        mr          qfsnfs1
/dev/vx/dsk/nfsdg/m11   104        mr          qfsnds1
EOF
```

mcf ファイルの詳細については、36 ページの「mcf ファイルの機能」を参照してください。

2. mcf(4) の構成が正しいかどうか検証し、エラーが見つかった場合は mcf ファイルを修正してから先に進みます。

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

▼ Sun StorageTek QFS ファイルシステムを作成し、Sun Cluster ノードを構成する

1. ファイルシステムの潜在的ホストとなる各ノードで、samd(1M) config コマンドを使用します。

このコマンドは、新しい Sun StorageTek QFS 構成ファイルがあることを Sun StorageTek QFS デーモンに知らせます。

```
# samd config
```

2. Sun Cluster システムの 1 つのノードから、sammkfs(1M) コマンドを使用してファイルシステムを作成します。

```
# sammkfs qfsnfs1 < /dev/null
```

3. ファイルシステムの潜在的なホストとなる各ノードで、次の手順を実行します。

- a. mkdir(1M) コマンドを使用してファイルシステムのグローバルマウントポイントを作成し、chmod(1M) コマンドを使用して root をマウントポイントの所有者にします。また、chown(1M) コマンドを使用して、読み取り/書き込みアクセス権 (755) を持っている other がマウントポイントを使用できるようにします。

```
# mkdir /global/qfsnfs1
# chmod 755 /global/qfsnfs1
# chown root:other /global/qfsnfs1
```

- b. /etc/vfstab ファイルに Sun StorageTek QFS ファイルシステムのエントリを追加します。

「mount options」フィールドには、sync_meta=1 という値があることに注意してください。

```
# cat >> /etc/vfstab << EOF
# device      device      mount      FS      fsck      mount      mount
# to mount    to fsck     point      type    pass     at boot    options
#
qfsnfs1      -          /global/qfsnfs1 samfs    2         no         sync_meta=1
EOF
```

▼ 構成の妥当性を検査する

1. ファイルシステムの潜在的なホストとなるすべてのノードが、正しく構成されているかどうか検査します。

それには、163 ページの「VxVM ソフトウェアを構成する」で作成したディスクグループをノードへ移動し、ファイルシステムをマウントしたり、マウント解除します。この検査は、ノード1つずつで実行してください。

```
# scswitch -z -D nfsdg -h scnode-B
# mount qfsnfs1
# ls /global/qfsnfs1
lost+found/
# umount qfsnfs1
```

2. 必要な Sun Cluster のリソースタイプがリソース構成に追加されていることを確認します。

```
# scrgadm -p | egrep "SUNW.HAStoragePlus|SUNW.LogicalHostname|SUNW.nfs"
```

3. 必要な Sun Cluster リソースタイプが見つからない場合は、次の1つまたは複数のコマンドで追加します。

```
# scrgadm -a -t SUNW.HAStoragePlus
# scrgadm -a -t SUNW.LogicalHostname
# scrgadm -a -t SUNW.nfs
```

▼ ネットワークネームサービスと IPMP 妥当性検査を構成する

ネットワークネームサービスと IPMP 妥当性検査を構成するには、151 ページの「ネットワークネームサービスと IPMP 妥当性検査を構成する」の説明に従います。

▼ HA-NFS を構成し、高可用 Sun StorageTek QFS ファイルシステムを構成する

高可用に HA-NFS とファイルシステムを構成するには、161 ページの「HA-NFS を構成し、高可用 Sun StorageTek QFS ファイルシステムを構成する」の説明に従ってください。

クラスタの外部にある共有クライアントの構成

Sun Cluster 環境を構成するとき、クラスタの外部に共有クライアントを置く場合は、次の構成を行います。

次の例は、2 ノードのメタデータサーバークラスタ構成に基づいています。

構成の前提条件

クラスタの外部に共有クライアントを設定するためには、次の項目を構成するか確認してください。

- 168 ページの「Sun StorageTek QFS メタデータサーバー Sun Cluster ノード」
- 169 ページの「Sun StorageTek QFS メタデータクライアントノード」
- 169 ページの「Sun Cluster 装置の構成」
- 169 ページの「クラスタの外部にクライアントを構成するための要件」
- 170 ページの「ソフトウェアの最低限のリリースレベル」
- 170 ページの「サポートされるハードウェアアーキテクチャー」
- 170 ページの「ストレージの要件」

Sun StorageTek QFS メタデータサーバー Sun Cluster ノード

Sun StorageTek QFS メタデータサーバー Sun Cluster ノードは、次の要件を満たしていなければなりません。

- mm 装置として識別された Sun StorageTek QFS メタデータ装置を、データ装置と同じ物理 LUN に常駐させることはできません。
- Sun Cluster の `scconf localonly` フラグを、Shared Sun StorageTek QFS 構成で mm 装置として定義されたメタデータ装置用に有効にすることはできません。

- mr 装置として識別された Sun StorageTek QFS データ装置を、メタデータ装置と同じ物理 LUN に常駐させることはできません。

Sun StorageTek QFS メタデータクライアントノード

Sun StorageTek QFS メタデータクライアントノードは、次の要件を満たしていなければなりません。

- mm 装置として識別された Sun StorageTek QFS メタデータ装置は、Shared Sun StorageTek QFS 構成で nodev 装置として識別される必要があります。
- Sun StorageTek QFS クライアント上で mr 装置として識別されたデータ装置には、クラスタノード上で使用される /dev/did/dsk 構文でなく、/dev/dsk 構文を使用してください。

Sun Cluster 装置の構成

すべてのデータ装置に、localonly フラグを設定してください。

/etc/opt/SUNWsamfs/mcf ファイルでデータ装置にローカルモードを設定すると、どの装置を Sun StorageTek QFS データ装置として使用するかを指定できます。

Sun Cluster 内で稼働しているノードで、root として次の操作を実行します。

```
scconf -r -D name=dsk/dX,nodelist=node2
```

```
scconf -c -D name=dsk/dX,localonly=true
```

クラスタの外部にクライアントを構成するための要件

Sun Cluster と Shared Sun StorageTek QFS クライアントの両方を含んだ構成は複雑であるため、Sun StorageTek QFS のメタデータトラフィック用に別個のプライベートネットワークが不可欠です。さらに、次のことも言えます。

- メタデータサーバー (MDS) 用に使用される Sun Cluster ノードは、IPMP グループを使用するようにします。
- Sun Cluster Sun StorageTek QFS リソースグループには、MDS とクラスタの外部に存在するクライアントとの間でメタデータ通信を行うための、論理ホスト名を設定するようにします。
- Sun Cluster Sun StorageTek QFS リソースグループは、論理ホスト名リソースと Sun StorageTek QFS ファイルシステムリソースの間の依存関係プロパティを使用するようにします。たとえば、Sun StorageTek QFS ファイルシステムリソースは、論理ホスト名リソースに依存するようになります。
- ma、mr、および gx 装置のみがサポートされます。
- この構成の場合、ソフトウェアボリュームマネージャーはサポートされません。

- ほかの Sun Cluster エージェントはサポートされません。

ソフトウェアの最低限のリリースレベル

最低でも、次のソフトウェアのリリースレベルが必要です。

- Sun Cluster 3.1 Update 3、Update 4、または Sun Cluster 3.2 (SPARC および AMD)
- Shared Sun StorageTek QFS 4.5、Patch 03 または Sun StorageTek QFS 4.6 (SUNWqfsr および、SUNWqfsu パッケージ)
- Solaris 9 Update 8 (SPARC および X86 クライアントのみ)、または Solaris 10 (SPARC および AMD) このほかのオペレーティングシステムはサポートされません。

サポートされるハードウェアアーキテクチャー

次のハードウェアアーキテクチャーがサポートされます。

注 – 混成アーキテクチャーはサポートされません。

- SPARC
- AMD64

ストレージの要件

共有ストレージ構成には、RAID5 をサポートしたハードウェアレベルのミラーリングを組み込む必要があります。サーバーとクライアントは Sun StorageTek Traffic Manager (MPxIO) 構成を使用するようにし、共有ストレージのみがサポートされません。

構成の手順

次に示す例では、3つの SPARC Sun Cluster ノードからなる構成を使用します。それらのノードは、次のように識別されます。

```
ctelab30    MDS #SPARC Sun Cluster Node
ctelab31    MDS #SPARC Sun Cluster Node
ctelab32    MDC #SPARC QFS Client Node
```

▼ /etc/hosts ファイルを編集する

- オペレーティングシステムをインストールしたあと、各ノードの /etc/hosts ファイルを編集して、ノードを準備します。

例:

```
### SC Cluster Nodes ###
129.152.4.57 ctelab30 # Cluster Node
129.152.4.58 ctelab31 # Cluster Node
129.152.4.59 ctelab32 # QFS Client Node

### SC Logical ###
192.168.4.100 sc-qfs1

### QFS NET ### ##
## ctelab30
192.168.4.20 ctelab30-4
192.168.4.160 ctelab30-qfe1-test
192.168.4.210 ctelab30-qfe2-test

## ctelab31
192.168.4.21 ctelab31-4
192.168.4.161 ctelab31-qfe1-test
192.168.4.211 ctelab31-qfe2-test

## ctelab32
192.168.4.22 ctelab32-qfs
```

▼ メタデータサーバーネットワークを構成する

次の例では、サーバーネットワークの設定プロセスを示します。これらの例では、次の設定を想定しています。

- ce0、ce1 は、Sun Cluster プライベートインターコネクタに使用される
- qfe0 は、パブリックネットワーク (129.152.4.0) に使用される
- qfe1、qfe2 は、Sun StorageTek QFS プライベートネットワークで使用される IPMP グループに使用される

この例では、/etc/hosts、/etc/netmasks、/etc/nsswitch.conf、/etc/hostname.qfe1、および /etc/hostname.qfe2 の各ファイルを、次のようにして各サーバークラスターノードで変更してください。

1. /etc/nsswitch.conf ファイルを確認します。

例:

```
hosts: cluster files dns nis
```

2. 次の情報を `/etc/netmasks` ファイルに付加します。

```
192.168.4.0 255.255.255.0
```

3. `/etc/hostname.qfe1` ファイルを編集して、次の情報を含めます。

```
ctelab30-4 netmask + broadcast + group qfs_ipmp1 up addif
ctelab30-qfe1-test deprecated -failover netmask + broadcast +
up
```

4. `/etc/hostname.qfe2` ファイルを編集して、次の情報を含めます。

```
ctelab30-qfe2-test netmask + broadcast + deprecated group
qfs_ipmp1 -failover standby up
```

▼ メタデータクライアントネットワークを構成する

次の例では、クライアントネットワークの設定プロセスを示します。これらの例では、次の設定を想定しています。

- `qfe0` は、パブリックネットワーク (129.152.4.0) に使用される
- `qfe1` は、Sun StorageTek QFS プライベートネットワークに使用される (192.168.4.0)

この例では、`/etc/hosts`、`/etc/netmasks`、`/etc/nsswitch.conf`、`/etc/hostname.qfe1`、および `/etc/hostname.qfe2` を、次のようにして各メタデータコントローラ (MDC) ノードで変更してください。

1. `/etc/nsswitch.conf` ファイルを調べ、次のように変更します。

```
hosts: files dns nis
```

2. 次の情報を `/etc/netmasks` ファイルに付加します。

```
192.168.4.0 255.255.255.0
```

3. `/etc/hostname.qfe1` ファイルを編集して、次の情報を含めます。

```
ctelab32-4
```

▼ Sun Cluster のインストールと構成を行う

オペレーティングシステムの準備を完了し、各ノードで MPxIO マルチパス化ソフトウェアを使用可能にしたあと、次のように Sun Cluster ソフトウェアのインストールと構成を行うことができます。

1. Sun Cluster のマニュアルに従って、Sun Cluster ソフトウェアをインストールします。

2. 定数デバイスとして使用する共有ストレージ装置を識別します。

例:

```
scdidadm -L
scconf -a -q globaldev=dx
scconf -c -q reset
```

▼ Sun StorageTek QFS メタデータサーバーを構成する

Sun Cluster ソフトウェアをインストールし、クラスタ構成を確認したあと、次のようにして Sun StorageTek QFS MDS のインストールと構成を行うことができます。

1. 『Sun StorageTek QFS インストールおよびアップグレードの手引き』の説明に従って、Sun StorageTek QFS ソフトウェアをインストールします。

例:

```
# pkgadd-d .SUNWqfsr SUNWqfsu
```

2. Sun Cluster コマンド `scdidadm -L` を使用して、Sun StorageTek QFS 構成に使用される装置を識別します。
3. `mcf` ファイルを編集して、ファイルシステムデバイスを反映させます。

例:

```
#
# File system Qfs1 #
Qfs1 2 maQfs1onshared
/dev/did/dsk/d7s0 20 mmQfs1on
/dev/did/dsk/d8s0 21 mmQfs1on
/dev/did/dsk/d16s0 22 mrQfs1on
/dev/did/dsk/d10s0 23 mrQfs1on
/dev/did/dsk/d13s0 24 mrQfs1on
#
# File system Qfs2 #
Qfs2 5 maQfs2onshared
/dev/did/dsk/d9s0 50 mmQfs2on
/dev/did/dsk/d11s0 51 mmQfs2on
/dev/did/dsk/d17s0 52 mrQfs2on
/dev/did/dsk/d12s0 53 mrQfs2on
/dev/did/dsk/d14s0 54 mrQfs2on
/dev/did/dsk/d15s0 55 mrQfs2on
/dev/did/dsk/d18s0 56 mrQfs2on
```

4. MDS Sun StorageTek QFS データ装置上でローカルモードを設定します。

たとえば、前述で定義された Qfs1 ファイルシステムの場合、mr 装置として定義された装置について、次のことが実行されます。

```
#/usr/cluster/bin/scconf -r -D name=dsk/d16,nodelist=ctelab31
#/usr/cluster/bin/scconf -c -D name=dsk/d16,localonly=true
#/usr/cluster/bin/scconf -r -D name=dsk/d10,nodelist=ctelab31
#/usr/cluster/bin/scconf -c -D name=dsk/d10,localonly=true
#/usr/cluster/bin/scconf -r -D name=dsk/d13,nodelist=ctelab31
#/usr/cluster/bin/scconf -c -D name=dsk/d13,localonly=true
```

5. /etc/opt/SUNWsamfs/defaults.conf ファイルを編集します。

例:

```
trace
all = on
sam-fsd.size = 10M
sam-sharefsd.size = 10M
endtrace
```

6. Sun StorageTek QFS ファイルシステムの hosts ファイルを構築します。

hosts ファイルについては、『Sun StorageTek QFS インストールおよびアップグレードの手引き』および 59 ページの「共有ホストファイルの変更」を参照してください。

注 – ここでは、クラスタの外部にある MDC と通信するので、ネットワークを通じた Sun StorageTek QFS メタデータトラフィックを確立する必要があります。MDC は Sun Cluster 構成のメンバーではないので、このトラフィックには論理ホストを使用します。この構成例では、sc-qfs1 がそのホスト名です。

MDS 上に共有ホストテーブルを構築するには、次の手順に従います。

a. Sun Cluster scconf コマンドを使用して、ホスト順序情報を取得します。

例:

```
# /usr/cluster/bin/scconf -p | egrep Cluster node name: |Node private
hostname: |Node ID:
```

b. `scconf` コマンドの出力を書き留めておきます。

例:

```
Cluster node name:  ctelab30
Node ID:  1
Node private hostname:  clusternode1-priv

Cluster node name:  ctelab31
Node ID:  2
Node private hostname:  clusternode2-priv
```

c. 共有 `hosts` ファイルを作成します。

たとえば、`/etc/opt/SUNWsamfs/hosts.Qfs1` ファイルには次の情報が含まれます。

```
#
# MDS
# Shared MDS Host file for family set 'Qfs1'
#
#
ctelab30 clusternode1-priv,sc-qfs11-server
ctelab31 clusternode2-priv,sc-qfs12-
ctelab32 ctelab32-4--
```

d. ローカル `hosts` ファイルを作成します。

たとえば、`/etc/opt/SUNWsamfs/hosts.Qfs1.local` ファイルには次の情報が含まれます。

```
#
# MDS
# Local MDS Host file for family set 'Qfs1'
ctelab30      clusternode1-priv
ctelab31      clusternode2-priv
```

7. `sammkfs` コマンドを使用して、ファイルシステムを作成します。

例:

```
# /opt/SUNWsamfs/sbin/sammkfs -S Qfs1
```

8. 各クラスターノード上にマウントポイントを準備します。

例:

```
# mkdir -p /cluster/qfs1 /cluster/qfs2
```

9. /etc/vfstab ファイルにファイルシステムのエントリを付加します。

例:

```
###
# QFS Filesystems
###
Qfs1 - /cluster/qfs1 samfs - no shared
Qfs2 - /cluster/qfs2 samfs - no shared
```

10. ファイルシステムをマウントします。

例:

```
# mount Qfs1, mount Qfs2 no each cluster node
```

11. Sun Cluster MDS リソースグループを作成します。

次の手順を実行して、Sun Cluster の下に MDS リソースグループを作成します。

a. QFS リソースタイプを追加します。

例:

```
# /usr/cluster/bin/scrgadm -a -t SUNW.qfs
```

b. MDS リソースグループを作成します。

例:

```
# /usr/cluster/bin/scrgadm -a -g sc-qfs-rg -h ctelab30,ctelab31
# /usr/cluster/bin/scrgadm -c -g sc-qfs-rg -y RG_description= Metadata Server
+ MDC Clients
```

c. リソースグループに論理ホスト名を追加します。

例:

```
# /usr/cluster/bin/scrgadm -a -L -g sc-qfs-rg -l sc-qfs1 -n
qfs_ipmpl@ctelab30,qfs_ipmpl@ctelab31
# /usr/cluster/bin/scrgadm -c -j sc-qfs1 -y RG_description= Logical Hostname
resource for sc-qfs1
```

d. MDS リソースグループに Sun StorageTek QFS ファイルシステムリソースを追加します。

例:

```
# /usr/cluster/bin/scrgadm -a -g sc-qfs-rg -t SUNW.qfs -j fs-qfs-rs -x \
# QFSFileSystem=/cluster/qfs1,/cluster/qfs2 -y Resource_dependencies=sc-qfs1
```

e. リソースグループをオンラインにします。

例:

```
# /usr/cluster/bin/scswitch -Z -g sc-qfs-rg
```

f. 状態を確認します。

例:

```
# /usr/cluster/bin/scswitch
```

▼ Sun StorageTek QFS メタデータクライアントを構成する

すべてのメタデータクライアントにオペレーティングシステムをインストールしたあと、Sun StorageTek QFS クライアントのインストールと構成に進むことができます。

それらの手順を実行する前に、MPxIO が有効になっていることと、各クライアントがすべてのディスク装置にアクセスできることを確認してください。

1. 『Sun StorageTek QFS インストールおよびアップグレードの手引き』の説明に従って、Sun StorageTek QFS ソフトウェアをインストールします。

例:

```
# pkgadd-d .SUNWqfsr SUNWqfsu
```

2. `format` コマンドを MDC に対して、また、Sun Cluster の `scdidadm -L` コマンドを MDS に対して使用し、Sun StorageTek QFS の構成に使用する装置を識別します。

3. メタデータクライアント上に mcf ファイルを構築します。

例:

```
#
# File system Qfs1
#
Qfs1      2  ma Qfs1 on shared
nodev    20  mm Qfs1 on
nodev    21  mm Qfs1 on
/dev/dsk/c6t600C0FF0000000000332B21D0B90000d0s0 22  mr Qfs1 on
/dev/dsk/c6t600C0FF0000000000876E9124FAF9C00d0s0 23  mr Qfs1 on
/dev/dsk/c6t600C0FF000000000004CAD7CC3CDE500d0s0 24  mr Qfs1 on
#
# File system Qfs2
# Qfs2      5  ma Qfs2 on shared
nodev    50  mm Qfs2 on
nodev    51  mm Qfs2 on
/dev/dsk/c6t600C0FF0000000000332B057D2FF100d0s0 52  mr Qfs2 on
/dev/dsk/c6t600C0FF0000000000876E975EDA6A000d0s0 53  mr Qfs2 on
/dev/dsk/c6t600C0FF0000000000876E9780ECA8100d0s0 54  mr Qfs2 on
/dev/dsk/c6t600C0FF000000000004CAD139A855500d0s0 55  mr Qfs2 on
/dev/dsk/c6t600C0FF000000000004CAD4C40941C00d0s0 56  mr Qfs2 on
```

4. /etc/opt/SUNWsamfs/defaults.conf ファイルを編集します。

例:

```
trace
all = on
sam-fsd.size = 10M
sam-sharefsd.size = 10M
endtrace
```

5. Sun StorageTek QFS ファイルシステムの hosts ファイルを構築します。

MDS hosts ファイルからの情報を使用し、次に示す例に従います。

注 – MDS と MDC の間のメタデータ通信では、クラスタのメンバーでないクライアントは論理ホストを通じて通信しなければなりません。

a. 共有ホストファイルを作成します。

たとえば、`/etc/opt/SUNWsamfs/hosts.Qfs1` ファイルには次の情報が含まれます。

```
#
# MDC
# Shared Client Host file for family set 'Qfs1'
ctelab30  sc-qfs1      1      -      server
ctelab31  sc-qfs1      2      -
ctelab32  ctelab32-4     -      -
```

b. ローカルホストファイルを作成します。

たとえば、`/etc/opt/SUNWsamfs/hosts.Qfs1.local` ファイルには次の情報が含まれます。

```
#
# MDC
# Local Client Host file for family set 'Qfs1'
ctelab30  sc-qfs1@ctelab32-4
ctelab31  sc-qfs1@ctelab32-4
```

注 - `/etc/opt/SUNWsamfs/hosts.QFS1.local` ファイルは、クライアントごとに異なります。この例では、クライアントはメタデータのトラフィックのために、`ctelab32-4` 上に構成されたインターフェースを使用してホスト `sc-qfs1` にバインドしています。

6. 各クラスターノード上にマウントポイントを作成します。

例:

```
# mkdir -p /cluster/qfs1 /cluster/qfs2
```

7. `/etc/vfstab` ファイルを編集します。

例:

```
###
# QFS Filesystems
###
Qfs1 - /cluster/qfs1 samfs - yes bg,shared
Qfs2 - /cluster/qfs2 samfs - yes bg,shared
```

8. ファイルシステムをマウントします。

例:

```
# mount Qfs1, mount Qfs2 no each MDC node
```

Sun StorageTek QFS の構成の変更

この節では、Sun StorageTek QFS 共有ファイルシステムまたは非共有ファイルシステムの構成を Sun Cluster 環境で変更、無効化、および削除する方法を示します。次の節で構成されています。

- 180 ページの「共有ファイルシステムの構成を変更する」
- 181 ページの「raw グローバル装置を使用するファイルシステム上の HA-NFS を無効にする」
- 182 ページの「Solaris ボリュームマネージャーで制御されるボリュームを使用するファイルシステム上の HA-NFS を無効にする」
- 184 ページの「VxVM で制御されるボリュームを使用する Sun StorageTek QFS ファイルシステム上の HA-NFS を無効にする」

▼ 共有ファイルシステムの構成を変更する

この手順例は、136 ページの「構成例」の例を基にしています。

1. 各ノードに `oracle` ユーザーとしてログインし、データベースインスタンスをシャットダウンして、リスナーを停止します。

```
$ sqlplus "/as sysdba"  
SQL > shutdown immediate  
SQL > exit  
$ lsnrctl stop listener
```

2. メタデータサーバーに `superuser` としてログインし、メタデータサーバーのリソースグループを管理対象外の状態にします。

```
# scswitch -F -g qfs-rg
# scswitch -u -g qfs-rg
```

この時点で、共有ファイルシステムはすべてのノードからマウント解除されます。これで、ファイルシステムの構成、マウントオプションなどを変更できるようになりました。必要であれば、ファイルシステムを再作成することもできます。再作成後にファイルシステムを再び使用するには、136 ページの「構成例」の説明に従ってください。

3. メタデータサーバーのリソースグループ構成または Sun StorageTek QFS ソフトウェアに変更を加える場合は、リソース、リソースグループ、およびリソースタイプを削除し、すべてが削除されたことを確認します。

たとえば、新しいパッケージへのアップグレードが必要になることがあります。

```
# scswitch -n -j qfs-res
# scswitch -r -j qfs-res
# scrgadm -r -g qfs-rg
# scrgadm -r -t SUNW.qfs
# scstat
```

この時点で、リソースグループを再作成して、別の名前、ノードリストなどを定義できます。必要であれば、Sun StorageTek QFS 共有ソフトウェアを削除またはアップグレードすることもできます。新しいソフトウェアのインストール後、メタデータリソースグループおよびリソースを再作成し、オンラインにできます。

▼ raw グローバル装置を使用するファイルシステム上の HA-NFS を無効にする

この一般的な手順例を使用すると、raw グローバル装置を使用している非共有 Sun StorageTek QFS ファイルシステム上の HA-NFS を無効にできます。この手順例は、147 ページの「例 1: raw グローバル装置上の HA-NFS」を基にしています。

1. `scswitch(1M) -F -g` コマンドを使用して、リソースグループをオフラインにします。

```
# scswitch -F -g nfs-rg
```

2. NFS、Sun StorageTek QFS、および LogicalHost の各リソースタイプを無効にします。

```
# scswitch -n -j nfs1-res
# scswitch -n -j qfsnfs1-res
# scswitch -n -j lh-nfs1
```

3. 以前に構成したリソースを削除します。

```
# scrgadm -r -j nfs1-res
# scrgadm -r -j qfsnfs1-res
# scrgadm -r -j lh-nfs1
```

4. 以前に構成したリソースグループを削除します。

```
# scrgadm -r -g nfs-rg
```

5. NFS 構成ディレクトリを空にします。

```
# rm -fr /global/nfs
```

6. 以前に追加され、不要になったリソースタイプがある場合は、それら使用していたリソースタイプを無効にします。

```
# scrgadm -r -t SUNW.HAStoragePlus
# scrgadm -r -t SUNW.LogicalHostname
# scrgadm -r -t SUNW.nfs
```

▼ Solaris ボリュームマネージャーで制御されるボリュームを使用するファイルシステム上の HA-NFS を無効にする

この一般的な手順例を使用すると、Solstice DiskSuite/Solaris ボリュームマネージャーで制御されるボリュームを使用している非共有 Sun StorageTek QFS ファイルシステム上の HA-NFS を無効にできます。この手順例は、155 ページの「例 2: Solstice DiskSuite/Solaris ボリュームマネージャーで制御されるボリューム上の HA-NFS」を基にしています。

1. リソースグループをオフラインにします。

```
# scswitch -F -g nfs-rg
```

2. NFS、Sun StorageTek QFS、および LogicalHost の各リソースタイプを無効にします。

```
# scswitch -n -j nfs1-res  
# scswitch -n -j qfsnfs1-res  
# scswitch -n -j lh-nfs1
```

3. 以前に構成したリソースを削除します。

```
# scrgadm -r -j nfs1-res  
# scrgadm -r -j qfsnfs1-res  
# scrgadm -r -j lh-nfs1
```

4. 以前に構成したリソースグループを削除します。

```
# scrgadm -r -g nfs-rg
```

5. NFS 構成ディレクトリを空にします。

```
# rm -fr /global/nfs
```

6. 以前に追加され、不要になったリソースタイプがある場合は、それら使用していたリソースタイプを無効にします。

```
# scrgadm -r -t SUNW.HAStoragePlus  
# scrgadm -r -t SUNW.LogicalHostname  
# scrgadm -r -t SUNW.nfs
```

7. RAID-5 および RAID-1 セットを削除します。

```
# metaclear -s nfsdg -f d30 d20 d21 d22 d23 d11 d1 d2 d3 d4
```

8. ドライブエラーの調停検出を削除します。

```
# metaset -s nfsdg -d -m scnode-A
# metaset -s nfsdg -d -m scnode-B
```

9. nfsdg ディスクグループから共有 DID 装置を削除します。

```
# metaset -s nfsdg -d -f /dev/did/dsk/d4 /dev/did/dsk/d5 \
/dev/did/dsk/d6 /dev/did/dsk/d7
```

10. Sun Cluster システム内のノード間で、ディスクグループ nfsdg の構成を削除します。

```
# metaset -s nfsdg -d -f -h scnode-A scnode-B
```

11. メタデータベースが不要になった場合は、それを削除します。

```
# metadb -d -f /dev/dsk/c0t0d0s7
# metadb -d -f /dev/dsk/c1t0d0s7
# metadb -d -f /dev/dsk/c2t0d0s7
```

▼ VxVM で制御されるボリュームを使用する Sun StorageTek QFS ファイルシステム上の HA-NFS を無効にする

この一般的な手順例を使用すると、VxVM で制御されるボリュームを使用している非共有 Sun StorageTek QFS ファイルシステム上の HA-NFS を無効にできます。この手順例は、162 ページの「例 3: VxVM ボリューム上の HA-NFS」を基にしています。

1. リソースグループをオフラインにします。

```
# scswitch -F -g nfs-rg
```

2. NFS、Sun StorageTek QFS、および LogicalHost の各リソースタイプを無効にします。

```
# scswitch -n -j nfs1-res
# scswitch -n -j qfsnfs1-res
# scswitch -n -j lh-nfs1
```

3. 以前に構成したリソースを削除します。

```
# scrgadm -r -j nfs1-res
# scrgadm -r -j qfsnfs1-res
# scrgadm -r -j lh-nfs1
```

4. 以前に構成したリソースグループを削除します。

```
# scrgadm -r -g nfs-rg
```

5. NFS 構成ディレクトリを空にします。

```
# rm -fr /global/nfs
```

6. 以前に追加され、不要になったリソースタイプがある場合は、それら使用していたリソースタイプを無効にします。

```
# scrgadm -r -t SUNW.HAStoragePlus
# scrgadm -r -t SUNW.LogicalHostname
# scrgadm -r -t SUNW.nfs
```

7. サブディスクを削除します。

```
# vxdg destroy nfsdg
```

8. VxVM 装置を削除します。

```
# vxdisk rm fabric_0 fabric_1 fabric_2 fabric_3 fabric_4
```

Sun Cluster を使用した高可用性 Sun StorageTek SAM 構成

Sun Cluster ソフトウェアを使用することにより、Sun StorageTek SAM を高可用性用に構成することもできます。Sun Cluster ソフトウェアは主ノードに障害が起きたとき、クラスタ内のほかのノードが自動的にアーカイブの作業負荷をホストできるようにすることにより、大幅に停止時間を減らし、生産性を高めることができます。

高可用性 SAM (HA-SAM) は Sun StorageTek QFS Sun Cluster エージェントに依存するので、この構成は、Sun StorageTek QFS Sun Cluster エージェントによってマウントおよび管理される共有 Sun StorageTek QFS ファイルシステムを使用してインストールしてください。

詳細については、『Sun StorageTek Storage Archive Manager (SAM) アーカイブ構成および管理マニュアル』を参照してください。

第7章

高度な機能

この章では、システムの基本的な管理や使用に含まれない高度な機能について説明します。この章の内容は次のとおりです。

- 187 ページの「デーモン、プロセス、およびトレースの使用」
- 192 ページの「ファイル属性を設定するための `setfa(1)` コマンドの使用」
- 194 ページの「WORM-FS ファイルシステムの構成」
- 209 ページの「大容量ファイルの格納」
- 210 ページの「複数読み取りファイルシステムの構成」
- 212 ページの「異機種システム混在環境における SAN-QFS ファイルシステムの使用」
- 221 ページの「入出力タイプについて」
- 223 ページの「大容量ファイル転送パフォーマンスの向上」
- 226 ページの「Qwrite 機能の有効化」
- 227 ページの「書き込みスロットルの設定」
- 228 ページの「遅延フラッシュ率の設定」
- 229 ページの「i ノードの数と i ノードハッシュテーブルの調整」

デーモン、プロセス、およびトレースの使用

システムのデーモンやプロセスについて理解していると、デバッグを行うときに役に立ちます。この節では、Sun StorageTek QFS のデーモンとプロセスについて説明します。また、デーモンのトレースについても説明します。

デーモンとプロセス

Sun StorageTek QFS のすべてのデーモンには、`sam-daemon_named` という形式で名前が付けられています。プロセスも同様の名前が付けられていますが、最後が小文字の `d` でないことだけが違います。

表 7-1 は、システムで実行可能なデーモンおよびプロセスの一部を示しています。これ以外にも、システムの稼働状態によって `sam-genericd` や `sam-catserverd` なども実行されています。

表 7-1 デーモンとプロセス

プロセス	説明
<code>sam-fsd</code>	マスターデーモン。
<code>sam-sharefsd</code>	Sun StorageTek QFS 共有ファイルシステムデーモンを呼び出します。
<code>sam-rpcd</code>	遠隔手続き呼び出し (RPC) のアプリケーションプログラミングインタフェース (API) サーバプロセスを制御します。

Sun StorageTek QFS ソフトウェアを実行すると、`init` により、`/etc/inittab` 処理の一部として `sam-fsd` デーモンが起動されます。デーモンは、`init` レベル 0、2、3、4、5、6 で起動され、障害発生時には自動的に再起動します。

Sun StorageTek QFS 共有ファイルシステムでは、`sam-fsd` デーモンが常にアクティブになっています。また、それぞれのマウントされた共有ファイルシステムでは、1 つの `sam-sharefsd` デーモンがアクティブになっています。

Sun StorageTek QFS 共有ファイルシステムがマウントされると、ソフトウェアは共有ファイルシステムデーモン (`sam-sharefsd`) を起動します。サーバホストとクライアントホストの通信には TCP ソケットが使用されます。メタデータサーバに接続するすべてのクライアントは、ホストファイルに対して妥当性検査が行われています。

注 – ホストファイルの詳細については、`hosts.fs(4)` のマニュアルページを参照してください。

メタデータサーバの `sam-sharedfsd` デーモンは、`sam-qfs` という名前の監視ソケットを開きます。Sun StorageTek QFS のインストール時、`/etc/services` ファイルに `sam-qfs` のエントリが自動的に追加されます。このエントリは削除しないでください。また、`/etc/inet/services` ファイルに定義される共有ファイルシステムのポート番号は 7105 です。このポートが、ほかのサービスと競合しないことを確認してください。

注 – Sun StorageTek QFS 4U2 リリース以前は、ファイルシステムごとに 1 つのポートが必要でした。これらのエントリーは、ファイルから削除できます。

メタデータ操作、ブロック割り当ておよび割り当て解除、およびレコードロックは、すべてメタデータサーバーファイルシステムで実行されます。sam-sharefsd デーモンには情報が保持されません。このため、このデーモンの停止や再起動によって、ファイルシステムの整合性の問題が発生することはありません。

トレースファイル

Sun StorageTek QFS のプロセスには、トレースファイルにメッセージを書き出すことができるものがあります。メッセージには、これらのデーモンが実行する処理の状態と進捗に関する情報が含まれています。メッセージは、主に Sun の担当者がパフォーマンスの改善や問題の診断のために使用します。メッセージの内容と書式は、リリースによって変更する可能性があります。

トレースファイルはデバッグ時に使用できます。デフォルトでは、トレースファイルが有効になりません。トレースファイルを有効にするには、defaults.conf ファイルを編集します。すべてのプロセスのトレースを有効にするか、個々のプロセスのトレースを有効にすることができます。トレース可能なプロセスについては、defaults.conf(4) のマニュアルページを参照してください。

デフォルトでは、トレースファイルは /var/opt/SUNWsamfs/trace ディレクトリに書き込まれます。ディレクトリ内のトレースファイルには、プロセスごとの名前が付きます (archiver、catserver、fsd、ftpd、recycler、sharefsd、および stager)。トレースファイルの名前を変更するには、defaults.conf 構成ファイルに指示を入力します。また、トレースファイルのサイズに制限を設けたり、トレースログを切り替えたりすることもできます。トレースファイルの各種設定については、defaults.conf(4) のマニュアルページを参照してください。

トレースファイルの内容

トレースファイルメッセージには、メッセージの時刻と発信元が含まれます。メッセージは、プロセスのイベントによって生成されます。イベントは、defaults.conf ファイルで指示を使用して選択します。

デフォルトのイベントは次のとおりです。

- カスタマー通知 syslog または notify ファイルメッセージ
- 重大でないプログラムエラー
- 重大な syslog メッセージ
- プロセスの開始と終了

- そのほかのイベント

また、次のイベントもトレースできます。

- メモリー割り当て
- プロセス間通信
- ファイルの処理
- オペレータメッセージ
- キューの内容の変更
- そのほかのイベント

デフォルトのメッセージの要素 (プログラム名、プロセス ID (PID)、および時刻) は、必ず含まれます。除外することはできません。オプションとして、次の要素をメッセージに含めることができます。

- 日付 (時刻は常に含まれます)
- 発信元のファイル名と行番号
- イベントの種類

トレースファイルの切り換え

トレースファイルが無制限に大きくならないように、`sam-fsd` デーモンはトレースファイルのサイズを監視し、定期的に次のスクリプトを実行します。

```
/opt/SUNWsamfs/sbin/trace_rotate
```

このスクリプトによって、トレースファイルは、順次番号の付いたコピーファイルに移動されます。スクリプトは、システムでの操作に合わせて変更できます。また、この機能は、`cron(1)` やそのほかのコマンドを使用して提供することもできます。

トレース対象プロセスの判別

現在トレースされているプロセスを判別するには、コマンド行に `sam-fsd(1M)` コマンドを入力します。コード例 7-1 に、このコマンドからの出力を示します。

コード例 7-1 sam-fsd コマンドの出力

```
# sam-fsd
Trace file controls:
sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
              cust err fatal misc proc date
              size 0 age 0
sam-archiverd /var/opt/SUNWsamfs/trace/sam-archiverd
              cust err fatal misc proc date
              size 0 age 0
sam-catserverd /var/opt/SUNWsamfs/trace/sam-catserverd
              cust err fatal misc proc date
              size 0 age 0
sam-fsd       /var/opt/SUNWsamfs/trace/sam-fsd
              cust err fatal misc proc date
              size 0 age 0
sam-rftd      /var/opt/SUNWsamfs/trace/sam-rftd
              cust err fatal misc proc date
              size 0 age 0
sam-recycler  /var/opt/SUNWsamfs/trace/sam-recycler
              cust err fatal misc proc date
              size 0 age 0
sam-sharefsd  /var/opt/SUNWsamfs/trace/sam-sharefsd
              cust err fatal misc proc date
              size 0 age 0
sam-stagerd   /var/opt/SUNWsamfs/trace/sam-stagerd
              cust err fatal misc proc date
              size 0 age 0
sam-serverd   /var/opt/SUNWsamfs/trace/sam-serverd
              cust err fatal misc proc date
              size 0 age 0
sam-clientd   /var/opt/SUNWsamfs/trace/sam-clientd
              cust err fatal misc proc date
              size 0 age 0
sam-mgmt      /var/opt/SUNWsamfs/trace/sam-mgmt
              cust err fatal misc proc date
              size 0 age 0
```

トレースファイル有効化の詳細は、defaults.conf(4)のマニュアルページおよびsam-fsd(1M)のマニュアルページを参照してください。

ファイル属性を設定するための setfa(1) コマンドの使用

Sun StorageTek QFS ファイルシステムでは、一般ユーザーがファイルやディレクトリのパフォーマンス属性を設定できます。これらのパフォーマンス属性は、アプリケーションでファイルごと、またはディレクトリごとに有効にできます。この節では、アプリケーションプログラマが、これらの機能を使用して、ファイルやディレクトリの属性を選択したり、ファイル領域を事前に割り当てたり、ファイルの割り当て方を指定したり、ディスクのストライプ幅を指定したりする方法について説明します。

次の項目で説明する機能の使用の詳細については、setfa(1) のマニュアルページを参照してください。

ファイルやディレクトリのファイル属性の選択

setfa(1) コマンドでは、新規ファイルまたは既存ファイルの属性が設定されます。ファイルがまだ存在していない場合は作成されます。

属性は、ファイルと同じくディレクトリにも設定できます。ディレクトリに対して setfa(1) を使用すると、そのディレクトリ内に作成されるファイルとディレクトリは、元のディレクトリに設定される属性を継承します。ファイルまたはディレクトリの属性をデフォルトにリセットするには、-d (デフォルト) オプションを使用します。-d オプションを使用すると、属性がまずデフォルトにリセットされます。そのあとで、ほかの属性が処理されます。

ファイル領域の事前割り当て

一般ユーザーがファイルの領域を事前に割り当てることができます。この領域はファイルに関連付けられるため、ファイルシステム内のほかのファイルは、そのファイルに割り当てられたディスクアドレスを使用できなくなります。事前割り当てにより、そのファイルの領域が必ずあることが保証され、ファイルシステムが満杯になってしまう状態を避けることができます。事前割り当ては、データが実際にディスクに書き込まれるときではなく、要求時に行われます。

ファイルの事前割り当てを行うと、領域が無駄になる可能性があることに注意してください。ファイルサイズが割り当て容量よりも小さい場合は、現在のファイルサイズから割り当て容量までについては、カーネルがファイルに領域を割り当てます。ファイルが閉じたときに、割り当て領域未満の領域は解放されません。

ファイル領域の事前割り当てを行うには、`setfa(1)` コマンドに `-L` または `-l` (小文字の `L`) オプションを指定します。どちらのオプションでも、引数にファイルの長さを指定できます。空でも、データが格納されていても、既存のファイルには `-L` オプションを使用します。データが入っていないファイルには `-l` オプションを使います。`-l` オプションを使うと、ファイルは事前割り当てされた領域を超えて大きくなることができません。

たとえば、`/qfs/file_alloc` という名前の 1G バイトのファイルを事前割り当てするには、次のように入力します。

```
# setfa -l 1g /qfs/file_alloc
```

ファイルの領域が事前割り当てされたあとで、ファイルのサイズを 0 に切り捨てるか、ファイルを削除すると、ファイルに割り当てた領域がすべて戻されます。ファイルの事前割り当て領域を一部だけファイルシステムに戻す方法はありません。また、`-l` オプションでファイルを事前割り当てすると、あとから事前割り当てサイズを超えてファイルを拡張する方法はありません。

ファイル割り当て方式とストライプ幅の選択

デフォルトでは、マウント時に指定した割り当て方式とストライプ幅がファイルで使用されます (`mount_samfs(1M)` のマニュアルページを参照してください)。ただし、エンドユーザーがファイルまたはディレクトリで異なる割り当て方式を使用することがあります。そのために、ユーザーは `-s` (ストライプ) オプション付きで `setfa(1)` コマンドを使用します。

割り当て方式にはラウンドロビン式とストライプ化があります。表 7-2 に示すように、`-s` オプションは割り当て方法とストライプ幅を指定します。

表 7-2 ファイル割り当てとストライプ幅

<code>-s</code> オプション	割り当て方式	ストライプ幅	説明
0	ラウンドロビン式	なし	装置の領域がなくなるまで、1 つの装置にファイルが割り当てられます。
1 ~ 255	ストライプ化	1 ~ 255 DAU DAU	ファイルは、1 ディスクあたり指定の DAU 数ずつ、すべてのディスク装置にストライプ化されます。

次の例は、ラウンドロビン割り当て方式を指定することによりファイルを明示的に作成する方法を示します。

```
# setfa -s 0 /qfs/100MB.rrobin
```

次の例は、64 DAU のストライプ幅でストライプ割り当て方式を指定して、ファイルを明示的に作成する方法を示しています (事前割り当ては行いません)。

```
# setfa -s 64 /qfs/file.stripe
```

ストライプ化グループ装置の選択

ストライプ化グループ装置は Sun StorageTek QFS ファイルシステムだけでサポートされています。

ユーザーは、特定のストライプ化グループからファイルの割り当てを開始するように指定できます。ファイル割り当て方式がラウンドロビン式の場合は、指定のストライプ化グループにファイルが割り当てられます。

コード例 7-2 の `setfa(1)` コマンドは、`file1` と `file2` を 2 つのストライプ化グループにそれぞれ分散することを指定しています。

コード例 7-2 2 つのストライプ化グループに分散させる `setfa(1)` コマンド

```
# setfa -g0 -s0 file1
# setfa -g1 -s0 file2
```

この機能は、`raw` 装置の速度と同等のパフォーマンスレベルを実現する必要があるアプリケーションで特に重要です。詳細については、`setfa(1)` のマニュアルページを参照してください。

WORM-FS ファイルシステムの構成

WORM (Write Once Read Many) 技術は、データの完全性の理由から、また、この技術を使用して格納されるファイルの法的有効性が広く認められたため、多くのアプリケーションで使用されています。Sun StorageTek QFS ソフトウェアの `release 4U3` からは、`SUNWsamfswm` というアドオンパッケージとして WORM-FS 機能を利用できるようになりました。ソフトウェアの `4U4 release` では、WORM-FS インタフェース

が修正され、新しい Sun StorageTek 5310 NAS (Network Attached Storage) アプライアンスと互換になりました。ssum を使用する以前の WORM-FS インタフェースはサポートされなくなりました。

注 – WORM-FS パッケージ (SUNwsamfswm) は Sun StorageTek QFS ソフトウェアパッケージに含まれていますが、pkgadd コマンドを使用して別個にインストールしてください。

WORM-FS 機能では、デフォルトおよびカスタマイズされたファイル保存期間、データおよびパスの不変性、WORM 設定のサブディレクトリへの継承が提供されます。

WORM-FS は、2 つのモードのいずれかで動作します。Sun 標準標準モード (以後、単に「標準モード」と呼ぶ) がデフォルトのモードです。Sun エミュレーション標準モード (以後、単に「エミュレーションモード」と呼ぶ) は、Sun StorageTek 5320 NAS (Network Attached Storage) アプライアンスのエミュレーションモードと互換になるように設計されており、Network Appliance によって定義されたインタフェースによく似ています。

標準モードとエミュレーションモードの違いの 1 つは、保存できるファイルの性質に関する制約です。具体的に言うと、標準モードでは、UNIX 実行可能アクセス権を持つファイルを保存できません。エミュレーションモードでは、そのような制約がありません。標準モードで制約が存在する理由は、NFS および FTP 用に定義された保存トリガーの性質にあります。それらのプロトコルでは、保存の要求は、ファイルに setuid モードを設定するよう指定することによって行われます。ファイルが保存されると、クライアントからは setuid モードビットが設定されて見えますが、考えられるセキュリティホール (root ユーザーの所有する実行可能ファイルを WORM にでき、したがって削除不可になる) は、実行可能ファイルに対する制約によって防止されます。この方法の利点は、システム上のどのファイルが本当に WORM 保護ファイルであるかを、ユーザーまたはアプリケーションがより簡単に判別できることです。

WORM-FS 機能の有効化

WORM-FS 機能を有効にするには、4 つのマウントオプションを使用できます。

- `worm_capable`
標準 WORM モード。WORM トリガーコマンド `chmod 4000 file-name/directory-name` を使用して、ファイルまたはディレクトリの WORM ビットを設定します。
- `worm_lite`
標準 WORM モードの一部の制約を緩和します。システム管理者は、ファイルを保存期間が期限切れになる前に削除したり、ファイルの保存期間を短縮したりできます。ファイルのデータとパスの完全性は変わりません。詳細は、197 ページの「WORM 「lite」 オプション」を参照してください。

■ worm_emul

Sun StorageTek 5320 NAS (Network Attached Storage) アプライアンスのエミュレーションモードとの互換性を提供するために設計された WORM エミュレーションモード。これは、標準 WORM 機能を異なる WORM トリガーによって提供します。WORM ビットは、ディレクトリまたはファイルを書き込み可能から読み取り専用に変更することによって設定されます。

■ emul_lite

Sun StorageTek 5320 NAS (Network Attached Storage) アプライアンスの「lite」バージョンとの互換性を提供するために設計された WORM エミュレーションモード。これは、標準 WORM lite 機能を異なる WORM トリガーによって提供します。WORM ビットは、ディレクトリまたはファイルを書き込み可能から読み取り専用に変更することによって設定されます。worm_lite オプションのように、管理者はファイルに対して特別な操作を行うことができます。詳細は、197 ページの「WORM 「lite」 オプション」を参照してください。

これら 4 つのマウントオプションには、多少の排他性があります。「lite」から標準 WORM モードへのアップグレードはできますが、標準 WORM モードからエミュレーションモードへの変更、またはエミュレーションモードから標準モードへの変更はできません。これらのオプションは、ファイルシステムをマウントするときにコマンド行で指定するか、/etc/vfstab 内に示すか、または /opt/SUNWsamfs/samfs.cmd 内で指定できます。マウントオプションの優先順位に関する通常の規則が適用されます。

WORM 属性はマウントテーブルに格納され、ファイルシステム内のどのディレクトリにも WORM ファイルを作成できます。

注 - /etc/vfstab 内で WORM マウントオプションを設定するには、システム管理者権限を持たなければなりません。

コード例 7-3 に WORM-FS マウントオプションの例を示します。/samfs1 にマウントされたファイルシステム samfs1 は WORM 機能があり、ファイルのデフォルト保存期間が 60 分に設定されています。

コード例 7-3 WORM-FS のマウントオプションの使用

```
# cat /etc/vfstab
#device device mount FS fsckmountmount
#to mount to fsck pointtypepassat bootoptions
#
fd      -   /dev/fdfd-no-
/proc   -   /procproc-no-
/dev/dsk/c0t0d0s1- -swap-no-
samfs1  -   /samfs1samfs=yes worm_capable,def_retention=60
swap    -   /tmptmpfs=yes-
```

ファイルシステムに WORM ファイルが 1 つ以上ある状態で、WORM-FS 機能を有効にすると、WORM 機能が反映されるようにファイルシステムのスーパーブロックが更新されます。それ以後、`sammkfs` によってファイルシステムを再構築することはできません。ただし、`worm_lite` または `emul_lite` マウントオプションを使用している場合は除きます。

WORM 「lite」 オプション

`worm_lite` および `emul_lite` マウントオプションは、変更された WORM 環境を作成し、WORM 対応ボリュームと保存されたファイルに対して実行できる操作の制限を緩和します。WORM lite オプションは、データ保存の保証は必要だが WORM ほど厳しい制約をシステムが受ける必要がない、マニュアル管理および保存ポリシーを持つ会社にとってのソリューションとなります。一部のデータ保存設定を変更し、場合によっては無効にするメカニズムが存在します。

より厳しい標準 WORM ポリシーにアップグレードする前に、WORM lite オプションを使用して、WORM システムおよびアプリケーションのテストと構成を行うこともできます。

WORM lite 環境の動作は、標準 WORM モードによく似ています。ファイルのデータとパスは変わりませんが、システム管理者は次の特別な操作を実行できます。

- WORM ファイルを保存時間が期限切れになる前に削除する
- WORM ファイルの保存時間を短縮する
- WORM lite 対応ボリュームを削除するか、`sammkfs` コマンドを使用して再構築する

WORM ファイルの作成

WORM マウントオプションを使用すると、ファイルシステムに WORM ファイルを格納できるようになりますが、WORM ファイルが自動的に作成されるわけではありません。WORM ファイルを作成するには、最初にディレクトリを WORM 対応にする必要があります。そのためには、通常のディレクトリを作成してから WORM トリガーコマンドを使用して、ディレクトリに WORM ビットを設定します。使用するマウントオプションに応じて、次の WORM トリガーコマンドを使用できます。

- `worm_capable` または `worm_lite` マウントオプションを使用する場合は、`chmod 4000 directory-name` を使用して WORM ビットを設定します。
- `worm_emul` または `emul_lite` マウントオプションを使用する場合は、ディレクトリの書き込み権を削除して WORM ビットを設定します。

ディレクトリに WORM ビットを設定したあと、そのディレクトリ内にファイルを作成し、適切な WORM トリガーを使用して、保存するファイルに WORM ビットを設定できます。WORM トリガーは、ファイルとディレクトリのどちらでも同じものです。

システム全体のデフォルトの保存値を使用して、4 つのマウントオプションのそれぞれに WORM トリガーを使用する例を次に示します。

例 1。WORM トリガーは chmod 4000

標準 WORM 機能を使用した WORM トリガーの単純な適用

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 -          /samfs1 samfs -          no          bg,worm_capable

[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod 4000 WORM
[root@ns-east-44]# sls -D

WORM:
mode: drwxr-xr-x  links:  2  owner: root      group: root
length:      4096  admin id:    0  inode:      1025.1
access:      Jan 30 15:50  modification: Jan 30 15:50
changed:     Jan 30 15:50  attributes:  Jan  1  1970
creation:    Jan 30 15:50  residence:   Jan 30 15:50
worm-capable      retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod 4000 test
[root@ns-east-44]# sls -D

test:
mode: -r-Sr--r--  links:  1  owner: root      group: root
length:           0  admin id:    0  inode:      1026.3
access:          Jan 30 15:51  modification: Jan 30 15:51
changed: Jan 30 15:51  retention-end: Mar  1  15:51 2007
creation:    Jan 30 15:51  residence:   Jan 30 15:51
retention:   active          retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
rm: test not removed: Read-only file system
[root@ns-east-44]# ls
test
```

例 2。WORM トリガーは chmod 4000

標準 WORM lite 機能を使用した WORM トリガーの単純な適用

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 -          /samfs1 samfs -          no          bg,worm_lite

[root@ns-east-44]# mount samfs1
[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod 4000 WORM
[root@ns-east-44]# sls -D

WORM:
  mode: drwxr-xr-x  links:  2  owner: root      group: root
  length:          4096  admin id:    0  inode:   1025.1
  access:          Jan 30 16:12  modification: Jan 30 16:12
  changed:        Jan 30 16:12  attributes:   Jan 1  1970
  creation:       Jan 30 16:12  residence:    Jan 30 16:12
worm-capable      retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod 4000 test
[root@ns-east-44]# sls -D

test:
  mode: -r-Sr--r--  links:  1  owner: root      group: root
  length:           0  admin id:    0  inode:   1026.1
  access:           Jan 30 16:13  modification: Jan 30 16:13
  changed: Jan 30 16:13  retention-end: Mar  1 16:13 2007
  creation: Jan 30 16:13  residence:    Jan 30 16:13
  retention: active      retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
[root@ns-east-44]# ls
[root@ns-east-44]#
```

例 3. WORM トリガーは `chmod -w`

WORM エミュレーションモードを使用した WORM トリガーの単純な適用

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 -          /samfs1 samfs  -          no          bg,worm_emul

[root@ns-east-44]# mount samfs1
[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod -w WORM
[root@ns-east-44]# sls -D

WORM:
  mode: drwxr-xr-x  links:  2  owner: root      group: root
  length: 4096  admin id:  0  inode:  1025.1
  access: Jan 30 16:26  modification: Jan 30 16:26
  changed: Jan 30 16:26  attributes: Jan 1 1970
  creation: Jan 30 16:26  residence: Jan 30 16:26
worm-capable      retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod -w test
[root@ns-east-44]# sls -D

test:
  mode: -r--r--r--  links:  1  owner: root      group: root
  length: 0  admin id:  0  inode:  1026.1
  access: Jan 30 16:26  modification: Jan 30 16:26
  changed: Jan 30 16:26  retention-end: Mar 1 16:26 2007
  creation: Jan 30 16:26  residence: Jan 30 16:26
  retention: active      retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
rm: test not removed: Read-only file system
[root@ns-east-44]# ls
test
```

例 4。WORM トリガーは `chmod -w`

WORM エミュレーション lite モードを使用した WORM トリガーの単純な適用

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 -          /samfs1 samfs -          no          bg,emul_lite

[root@ns-east-44]# mount samfs1
[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod -w WORM
[root@ns-east-44]# sls -D

WORM:
  mode: drwxr-xr-x  links:  2  owner: root      group: root
  length:          4096  admin id:  0  inode:  1025.1
  access:          Jan 30 16:36  modification: Jan 30 16:36
  changed:        Jan 30 16:36  attributes: Jan 1 1970
  creation:       Jan 30 16:36  residence:   Jan 30 16:36
worm-capable      retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod -w test
[root@ns-east-44]# sls -D

test:
  mode: -r--r--r--  links:  1  owner: root      group: root
  length:           0  admin id:  0  inode:  1026.1
  access:          Jan 30 16:36  modification: Jan 30 16:36
  changed: Jan 30 16:36  retention-end: Mar  1 16:36 2007
  creation:       Jan 30 16:36  residence:   Jan 30 16:36
  retention:      active      retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
[root@ns-east-44]# ls
[root@ns-east-44]#
```

注 – WORM トリガーの適用には注意が必要です。ファイルに WORM 機能が適用されると、ファイルデータとパスを変更できなくなります。この機能をファイルに適用すると、元に戻せません。さらに、ファイルに WORM トリガーを適用すると、そのボリュームも WORM ボリュームになり、変更できなくなります。そのボリュームに対しては、ボリューム管理または RAID インタフェースを使用して廃棄することしかできません。WORM 「lite」 オプションの 1 つを使用して作成したボリュームは、`sammkfs` を使用して再構築することもできます。

保存期間

WORM-FS 機能には、カスタマイズ可能なファイル保存期間も用意されています。ファイルに保存期間を割り当てると、指定した期間の間、そのファイルで WORM 機能が維持されます。

注 – Solaris または UNIX ユーティリティーを使用して保存期間を初期に割り当てるか延長した場合は、その期間を 2038 年 1 月 18 日のあとまで延長することはできません。これは、それらのユーティリティーが符号付き 32 ビットの数値を使用して、時間を秒単位で表しているためです。時間は、1970 年 1 月 1 日の時点から計測されます。この時点からの 2**31 秒は、2038 年 1 月 18 日の午後 10 時 14 分ころまでとなります。ただし、デフォルトの保存期間を使用して、この日付を超えることができます。204 ページの「デフォルト保存期間の設定」を参照してください。

ファイルに保存期間を設定するには、次のいずれかを行います。

- `touch` ユーティリティーを使用するか、`libc` のサブルーチン `utimes()` を使用するプログラムによって、ファイルのアクセス時刻を進めます。ファイルの保存期間は、分単位で格納されます。アクセス時刻を進めてから、適切な WORM トリガーを使用して WORM ビットを設定してください。
- デフォルトのファイル保存期間を使用します。これを行うには、適切な WORM トリガーを適用し、ファイルシステムがデフォルトの保存期間を適用できるようにします。詳細は、204 ページの「デフォルト保存期間の設定」を参照してください。

コード例 7-4 に、WORM 対応ディレクトリにファイルを作成し、そのファイルに (`chmod 4000` コマンドで) WORM トリガーを使用し、`sls` コマンドを使用してファイルの WORM 機能を表示する様子を示します。この例では、ファイルシステムのデフォルトの保存期間を使用しています (コード例 7-3 で設定した 60 分間)。

コード例 7-4 WORM 対応ディレクトリと WORM ファイルの作成

```
# cd WORM
# echo "This is a test file" >> test
# sfs -D
test:
    mode: -rw-r--r--  links: 1  owner: root group: other
    length: 20  admin id: 0  inode: 1027.1
    access: Oct 30 02:50  modification: Oct 30 02:50
    changed: Oct 30 02:50  attributes: Oct 30 02:50
    creation: Oct 30 02:50  residence: Oct 30 02:50

    checksum: gen  no_use  not_val  algo: 0

# chmod 4000 test
# sfs -D
test:
    mode: -r--r--r--  links: 1  owner: root group: other
    length: 20  admin id: 0  inode: 1027.1
    access: Oct 30 02:50  modification: Oct 30 02:50
    changed: Oct 30 02:50  retention-end: Oct 30 2005 03:50
    creation: Oct 30 02:50  residence: Oct 30 02:50
    retention: active retention-period: 0y, 0d, 1h, 0m
    checksum: gen  no_use  not_val  algo: 0
```

WORM-FS 機能を追加すると、Sun StorageTek QFS ファイルシステム内のファイルが 3 つの状態を持つようになります。

- 通常
- 保存
- 期限切れ

通常状態は、Sun StorageTek QFS ファイルシステム内の通常ファイルの状態を表しています。ファイルに WORM ビットが設定されると、保存 (*active*) 状態へ移行します。ファイルの保存期間を超過すると、期限切れ (*over*) 状態になります。

ファイルに保存期間が割り当てられ、そのファイルに WORM トリガーが適用されると、ファイルのパスとデータが変更禁止になります。保存期間が切れると状態は「期限切れ」に変化しますが、パスとデータは変更禁止のままです。

ファイルが期限切れ状態のときは、次の 2 つの操作のみが可能です。

- 保存期間の延長。WORM 「lite」 オプションを使用している場合以外、保存期間の短縮はできません。
- ファイルの削除

保存期間が切れるとファイルの状態が「active」に戻り、新しい終了日と期間がそれぞれ設定されます。

WORM-FS 機能と組み合わせて、ファイルへのハードリンクとソフトリンクの両方を使用できます。ハードリンクは、WORM 機能を持つディレクトリ内にあるファイルに対してのみ作成できます。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、Sun StorageTek QFS ファイルシステム内のすべてのディレクトリで作成できます。

WORM-FS 機能のもう 1 つの特性はディレクトリ継承です。WORM 属性を含んでいるディレクトリの下に作成された新しいディレクトリは、この属性を親から継承します。ディレクトリにデフォルトの保存期間が設定されている場合、その保存期間は新しいサブディレクトリにも継承されます。WORM ビットは、親ディレクトリが WORM 機能を持つすべてのファイルに設定できます。一般ユーザーは、自分が所有するか、通常の UNIX アクセス権を使用してアクセス可能なディレクトリおよびファイルに WORM 機能を設定できます。

注 – WORM 機能を持つディレクトリは、そこに WORM ファイルが含まれていない場合にかぎり、削除することができます。

デフォルト保存期間の設定

ファイルシステムのデフォルト保存期間は、`/etc/vfstab` ファイルのマウントオプションとして設定できます。

例:

```
samfs1 - /samfs1 samfs - no
bg,worm_emul,def_retention=1y60d
```

デフォルトの保存期間を設定する形式は、*MyNdOhPm* です。M、N、O、P は負以外の整数です。また、y、d、h、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 4U5 より前のソフトウェアバージョンと下位互換性があります。注意すべき重要な点は、期間の粒度は分単位ですが、期間の正確度は 1 日を基にしていることです。また、日、時、および分を処理する機能は、保存期間を決定するときに、うるう年を考慮しません。これらのいずれか(またはすべて)を使用してデフォルトの保存期間を設定するときは、このことを考慮してください。

デフォルトの保存期間を使用して、ファイルまたはディレクトリに 2038 年のあとまでの保存期間を設定することもできます。これを行うには、デフォルトの保存期間に 2038 を超える値を設定し、ファイルシステムをマウントします。そのあと、適切な

WORM トリガーを使用してデフォルトの保存期間を適用します。デフォルトの保存期間を使用して、ディレクトリとファイルに 2038 年のあとまでの保存期間を設定する例を次に示します。

コード例 7-5 2038 年のあとまでの保存期間の延長

```
[root@ns-east-44]# grep samfs1 /etc/vfstab
samfs1 -          /samfs1 samfs -          no
bg,worm_capable,def_retention=34y
[root@ns-east-44]# mount samfs1
[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod 4000 WORM
[root@ns-east-44]# sls -D
WORM:
mode: drwxr-xr-x  links:  2  owner: root      group: root
length: 4096  admin id:  0  inode: 1026.1
access: Feb 20 14:24  modification: Feb 20 14:24
changed: Feb 20 14:24  attributes: Jul 26 1970
creation: Feb 20 14:24  residence: Feb 20 14:24
worm-capable      retention-period: 34y, 0d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod 4000 test
[root@ns-east-44]# sls -D
test:
mode: -r-Sr--r--  links:  1  owner: root      group: root
length: 0  admin id:  0  inode: 1027.1
access: Feb 20 14:24  modification: Feb 20 14:25
changed: Feb 20 14:25  retention-end: Feb 20 14:25 2041
creation: Feb 20 14:24  residence: Feb 20 14:24
retention: active      retention-period: 34y, 0d, 0h, 0m
```

touch ユーティリティーを使用して、ディレクトリにデフォルトの保存期間を設定することも可能です。方法については、205 ページの「touch を使用した保存期間の設定」を参照してください。この保存期間はファイルシステムのデフォルト保存期間を上書きし、サブディレクトリがあれば、それにも継承されます。

touch を使用した保存期間の設定

touch ユーティリティーを使用して、ファイルまたはディレクトリの保存期間を設定または延長できます。また、touch を使用して、ディレクトリのデフォルト保存期間を短縮できます。ただし、ファイルに対して保存期間を短縮することはできません。

保存期間を設定するには、touch を使用してファイルまたはディレクトリのアクセス時刻を進めたあと、その時点で使用されている WORM モードに応じて、chmod コマンドを使用するか書き込み権を削除することにより、WORM トリガーを適用します。

コード例 7-6 に、touch ユーティリティを使用したファイルの保存期間の設定と、そのあとの WORM トリガーの適用を示します。

コード例 7-6 touch および chmod を使用した保存期間の設定

```
# touch -a -t200508181125 test
# sls -D
test:
  mode: -rw-r--r--  links: 1  owner: root      group: root
  length:          0  admin id: 0  inode: 1027.1
  access:          Aug 18 2005  modification: Aug 18 11:19
  changed:         Aug 18 11:19  attributes:   Aug 18 11:19
  creation:        Aug 18 11:19  residence:    Aug 18 11:19

# chmod 4000 test
# sls -D
test:
  mode: -r-Sr--r--  links: 1  owner: root      group: root
  length:          0  admin id: 0  inode: 1027.1
  access:          Aug 18 2005  modification: Aug 18 11:19
  changed:         Aug 18 11:19  retention-end: Aug 18 2005 11:25
  creation:        Aug 18 11:19  residence:    Aug 18 11:19
  retention:       active  retention-period: 0y, 0d, 0h, 6m
```

touch の -a オプションは、ファイルやディレクトリのアクセス時刻を変更するために使用します。-t オプションによって、アクセス時刻フィールドに使用する時刻が指定されます。時刻引数の形式は [[CC]YY]MMDDhhmm[.SS] で、次のようになります。

- [CC] — 年の上位 2 桁です。
- [YY] — 年の下位 2 桁です。
- MM — 月です (01 ~ 12)。
- DD — 日です (01 ~ 31)。
- hh — 時です (00 ~ 23)。
- mm — 分です (00 ~ 59)。
- [SS] — 秒です (00 ~ 61)。

CC、YY、および SS フィールドは省略可能です。CC および YY を指定しなかった場合、デフォルトは現在の年になります。これらのオプションについての詳細は、touch のマニュアルページを参照してください。

保存期間として永続保存を設定するには、アクセス時刻を設定可能な最大値である 203801182214.07 に設定します。

ファイルの保存期間の延長

コード例 7-7 に、touch を使用してファイルの保存期間を延長する例を示します。

コード例 7-7 touch を使用したファイルの保存期間の延長

```
# sls -D test
test:
  mode: -r-Sr--r--  links: 1  owner: root      group: root
  length:          0  admin id: 0  inode: 1029.1
  access:          Aug 18 11:35  modification: Aug 18 11:33
  changed:        Aug 18 11:33  retention-end: Aug 18 2005 11:35
  creation:       Aug 18 11:33  residence:     Aug 18 11:33
  retention:      over          retention-period: 0y, 0d, 0h, 2m
# touch -a -t200508181159 test
# sls -D
test:
  mode: -r-Sr--r--  links: 1  owner: root      group: root
  length:          0  admin id: 0  inode: 1029.1
  access:          Aug 18 11:35  modification: Aug 18 11:33
  changed:        Aug 18 11:33  retention-end: Aug 18 2005 11:59
  creation:       Aug 18 11:33  residence:     Aug 18 11:33
  retention:      active          retention-period: 0y, 0d, 0h, 26m
```

この例で、保存期間は WORM トリガーが最初に適用されたときから 26 分後の、2005 年 8 月 18 日 AM 11:59 に延長されました。

注 - touch を使用した保存期間の延長は、アクティブな WORM モードとは関係ありません。

sls を使用した WORM-FS ファイルの表示

sls コマンドを使用すると、WORM ファイル属性が表示されます。-D オプションでは、ディレクトリに WORM 機能があるかどうかが表示されます。このオプションをファイルに対して使用すると、コマンド行で指定された保存期間の開始、保存期間の終了、現在の保存状態、および期間が表示されます。

保存期間の開始は、ファイルの `modified time` フィールドに格納されます。保存期間の終了は、ファイルの `attribute time` フィールドに格納されます。この時刻はカレンダー日で表示されます。`sls` 出力のそのほかの行には、保存期間の状態と期間が表示されます。

コード例 7-8 に、`sls -D` によるファイルの保存状態の表示例を示します。

コード例 7-8 `sls` を使用したファイルの保存状態の確認

```
sls -D test
test:
  mode: -r-Sr--r--  links: 1  owner: root group: root
  length: 5  admin id: 0  inode: 1027.1
  access: Aug 18 2005  modification: Aug 18 11:19
  changed: Aug 18 11:19  retention-end: Aug 18 2005 11:25
  creation: Aug 18 11:19  residence: Aug 18 11:19
  retention: active  retention-period: 0y, 0d, 0h, 6m
```

この例で、`retention: active` と示されているとおり保存状態は `active` で、ファイルには `WORM` ビットが設定されています。保存期間は 2005 年 8 月 18 日 11:19 に始まり、2005 年 8 月 18 日 11:25 に終了します。保存期間は、0 年、0 日、0 時間、6 分と指定されています。

`sfind` を使用した `WORM-FS` ファイルの検索

`sfind` ユーティリティを使用すると、特定の保存期間が指定されたファイルを検索できます。これらのオプションについての詳細は、`sfind(1)` のマニュアルページを参照してください。次の選択肢があります。

- `-ractive` — 保存期間が有効なファイルを検索します。
- `-rover` — 保存期間が切れたファイルを検索します。
- `-rafter date` — 指定された日付に保存期間が終了するファイルを検索します。日付は `YYYYMMDDHHmm` で指定し、`YYYY` は年、`MM` は月、`DD` は日、`HH` は時、`mm` は分です。

たとえば、コード例 7-9 は 2004 年 12 月 24 日 15:00 に保存期間が切れるファイルを検索するコマンドを示しています。

コード例 7-9 `sfind` を使用した、特定の日付以後に期限切れになるすべての `WORM` ファイルの検索

```
# sfind -rafter 200412241500
```

- `-rremain time` - 保存期間の残りが指定した時間以上あるファイルを検索します。時間は `MyNdOhPm` で指定し、M、N、O、および P は任意の負以外の整数、y、d、h、および m はそれぞれ年、日、時、分です。
たとえば、コード例 7-10 は期限切れまでが 1 年 10 日 5 時間 10 分よりも長いファイルを検索するコマンドを示しています。

コード例 7-10 `sfind` を使用した、残りが指定した時間よりも長いすべての WORM ファイルの検索

```
# sfind -rremain 1y10d5h10m
```

- `-rlonger time` - 保存期間が指定した時間よりも長いファイルを検索します。時間は `MyNdOhPm` で指定し、M、N、O、および P は任意の負以外の整数、y、d、h、および m はそれぞれ年、日、時、分です。
たとえば、コード例 7-11 は保存期間が 10 日よりも長いファイルを検索するコマンドを示しています。

コード例 7-11 `sfind` を使用した、保存期間が指定した時間よりも長いすべての WORM ファイルの検索

```
# sfind -rlonger 10d
```

- `-rpermanent` - 保存期間が永続的なファイルを検索します。

大容量ファイルの格納

非常に大きな容量のファイル进行操作するときは、システムで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も大きなファイルを書き込もうとした場合は、使用しているファイルシステムの種類によって次のように動作が異なります。

- Sun StorageTek QFS ファイルシステムを使用している場合は、システムから ENOSPC エラーが戻されます。
- SAM-QFS ファイルシステムを使用している場合は、プログラムがブロックし、存在しない領域を待ちます。これは、利用可能なディスク領域が、要求を処理するためには不十分であるためです。

SAM-QFS の環境で操作していて、アプリケーションでディスクキャッシュよりも大きなファイルを書き込む必要がある場合は、`segment(1)` コマンドを使用してファイルをセグメント化できます。`segment(1)` コマンドについては、`segment(1)` のマニュアルページか、『Sun StorageTek Storage Archive Manager (SAM) アーカイブ構成および管理マニュアル』を参照してください。

複数読み取りファイルシステムの構成

複数読み取りファイルシステムは、単一の書き込みホストと複数の読み取りホストから構成されます。複数読み取りファイルシステムを有効にする `writer` オプションと `reader` オプションは、Sun StorageTek QFS ファイルシステムでのみ使用できます。これらのマウントオプションについては、この節および `mount_samfs(1M)` のマニュアルページで説明しています。

複数読み取りファイルシステムは、`mount(1M)` コマンドで `-o writer` オプションを指定することにより単一の書き込みホストにマウントできます。`writer` マウントオプションを指定されたホストシステムだけが、そのファイルシステムに書き込みを許可されるホストです。`writer` ホストシステムによって、ファイルシステムが更新されます。複数読み取りファイルシステム内の1つのホストだけが、`writer` マウントオプションを有効にしてファイルシステムをマウントします。`-o writer` を指定すると、ディレクトリは変更するごとにディスクに書き込まれ、ファイルは閉じられたときにディスクに書き込まれます。



注意 – 複数の書き込みホストに複数読み取りファイルシステムを同時にマウントすると、ファイルシステムが破壊されることがあります。このような状況が発生しないように、サイト管理者の責任において管理してください。

複数読み取りファイルシステムは、`mount(1M)` コマンドで `-o reader` オプションを指定することにより1つまたは複数の読み取りホストにマウントできます。複数読み取りファイルシステムを読み取り側としてマウントできるホストシステムの数に制限はありません。

複数読み取りファイルシステムと Sun StorageTek QFS 共有ファイルシステムの大きな違いは、複数読み取りホストはディスクからメタデータを読み取りますが、Sun StorageTek QFS 共有ファイルシステムのクライアントホストはネットワークを介してメタデータを読み取ることです。Sun StorageTek QFS 共有ファイルシステムでは、複数読み取りホストをサポートしています。この構成では、複数の共有ホストがデータを追加する一方で、複数の読み取りホストがデータを配布することができます。

注 – ファイルシステムを Sun StorageTek QFS 共有ファイルシステムとしてマウントする場合は、どのホストに対しても `writer` オプションを指定することはできません。ただし、`reader` オプションは指定できます。

Sun StorageTek QFS 共有ファイルシステムのクライアントホストを読み取り専用のホストにするには、Sun StorageTek QFS 共有ファイルシステムをそのホストにマウントするときに、`reader` マウントオプションを指定します。さらに、Sun StorageTek QFS 共有ファイルシステムで `reader` オプションを使用する場合は、`sync_meta` マウントオプションを 1 に設定します。Sun StorageTek QFS 共有ファイルシステムの詳細については、71 ページの「Sun StorageTek QFS 共有ファイルシステムの構成」を参照してください。マウントオプションの詳細は、`mount_samfs(1M)` のマニュアルページを参照してください。

複数読み込みファイルシステムのすべての読み取り側が、`ma` 装置を記述するデバイス定義にアクセスできることを確認してください。主メタデータサーバーのホストに常駐する `mcf` ファイルの行を、代替メタデータサーバーの `mcf` ファイルにコピーします。構成によっては、ディスクパーティションがすべてのホストで共通の現れ方をするとはいえないため、行をコピーしたあとにディスクコントローラの情報を更新する必要が生じる場合があります。

複数読み取りファイルシステム環境では、Sun StorageTek QFS ソフトウェアによって、同一ファイルシステムにアクセスするすべてのサーバーは常に現在の環境にアクセスできることが保証されます。書き込み側がファイルを閉じると、Sun StorageTek QFS ファイルシステムはそのファイルのすべての情報をディスクにすぐに書き込みます。ファイルが書き込み側によって閉じられると、`reader` ホストがファイルにアクセスできません。`refresh_at_eof` マウントオプションを指定すると、複数ファイル読み込みシステムのすべてのホストシステムがファイルシステムと同期されるようになります。

デフォルトの場合、`reader` ホスト上のファイルのメタデータ情報はファイルがアクセスされるたびに無効になり、更新されます。変更されたデータは無効になります。これには、`cat(1)`、`ls(1)`、`touch(1)`、`open(2)`、またはそのほかの方法によるアクセスがあった場合が含まれます。このように即時に更新が行われると、更新時のデータの正確さは保証されますが、パフォーマンスに影響してしまう可能性があります。サイトの必要に応じて、`mount(1M)` コマンドの `-o invalid=n` オプションを使って 0 秒から 60 秒までの更新間隔を設定できます。更新間隔に小さな値を設定すると、Sun StorageTek QFS ファイルシステムによるディレクトリやほかのメタデータ情報の読み取りは最後の更新から n 秒後に行われます。更新の数が多いとシステムのオーバーヘッドは増加しますが、 n が 0 以外の数値であると無効な情報が存在する可能性が出てきます。



注意 – reader ホストで読み取り操作のためにファイルを開いているとき、そのファイルが書き込み側によって削除されたり切り捨てられたりすることに対する保護はありません。読み取り側が誤って書き込み操作を行うことを防ぐには、アプリケーションロックなど、別途の仕組みを使用する必要があります。

異機種システム混在環境における SAN-QFS ファイルシステムの使用

SAN-QFS ファイルシステムでは、Sun StorageTek QFS システムに格納されたデータに、複数のユーザーがディスク最高速でアクセスできます。この機能の特に有効な用途としては、データベースやデータストリーム、Web ページサービスのほか、異機種システム混在環境で、高パフォーマンスの共有ディスクアクセスを要求するアプリケーションが挙げられます。

SAN-QFS ファイルシステムは、ストレージエリアネットワーク (SAN) においてファイバ接続装置と組み合わせて使用できます。SAN-QFS ファイルシステムで、Sun StorageTek QFS ソフトウェアや、Tivoli SANergy などのファイル共有ソフトウェアを使用すると、データへの高速アクセスが可能になります。SAN-QFS ファイルシステムを使用するには、(2.2.4 以上の) SANergy ソフトウェアと Sun StorageTek QFS ソフトウェアの両方が必要です。Sun StorageTek QFS ソフトウェアおよび SANergy ソフトウェアのサポート対象バージョンの詳細は、ご購入先にお問い合わせください。

注 – Solaris OS およびサポート対象の Linux オペレーティングシステムが含まれている環境の Solaris ホストでは、SAN-QFS ファイルシステムではなく Sun StorageTek QFS 共有ファイルシステムを使用します。

Sun StorageTek QFS 共有ファイルシステムの詳細については、71 ページの「Sun StorageTek QFS 共有ファイルシステムの構成」を参照してください。Sun StorageTek QFS 共有ファイルシステムと SAN-QFS ファイルシステムとの相違点については、221 ページの「SAN-QFS 共有ファイルシステムと Sun StorageTek QFS 共有ファイルシステムの比較」を参照してください。

図 7-1 は、Sun StorageTek QFS ソフトウェアと SANergy ソフトウェアの両方を使用する SAN-QFS ファイルシステムを示したもので、具体的には各クライアントとメタデータコントローラ (MDC) システムがローカルエリアネットワーク (LAN) を介してメタデータを管理する仕組みを表しています。ここではクライアントがストレージとの直接入出力を実行します。

Solaris OS のみを実行しているクライアントがすべて Sun StorageTek QFS ソフト

ウェアのホストとなっている一方で、Solaris 以外の OS を実行している各種クライアントがいずれも SANergy ソフトウェアと NFS ソフトウェアのホストとなっていることに注意してください。SAN-QFS ファイルシステムのメタデータサーバーは、Sun StorageTek QFS と SANergy の両ソフトウェアのホストとして機能するだけでなく、SANergy MDC としても機能します。

注 – SANergy ソフトウェアは x64 ハードウェアプラットフォームでサポートされません。

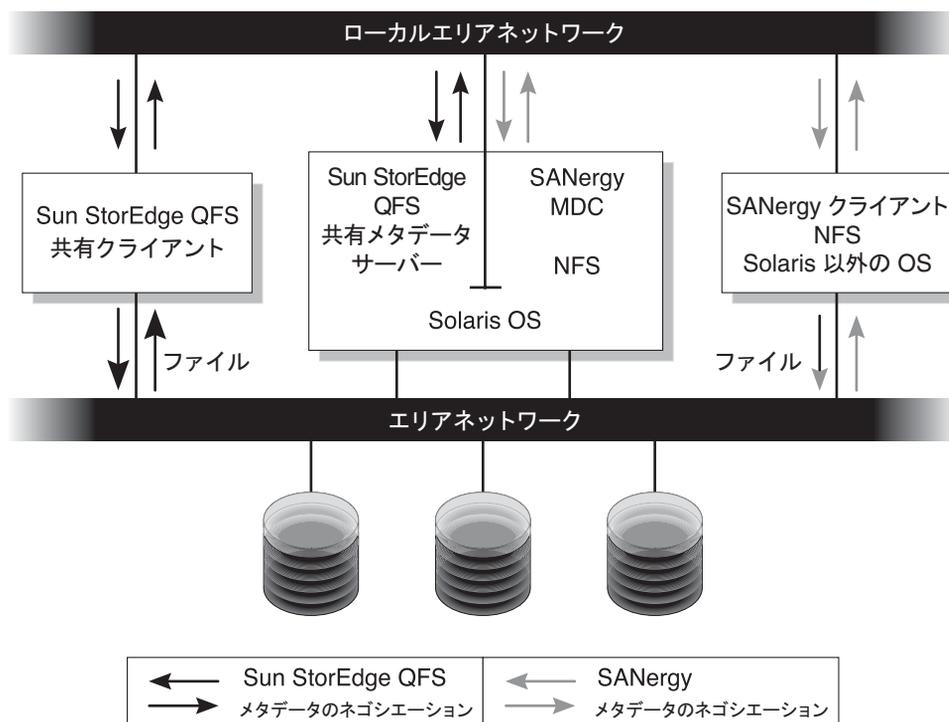


図 7-1 Sun StorageTek QFS ソフトウェアと SANergy ソフトウェアを使用した SAN-QFS ファイルシステム

ここでは、SAN-QFS ファイルシステムのそのほかの面について説明します。

- 214 ページの「事前確認」
- 214 ページの「SAN-QFS ファイルシステムの使用開始」
- 217 ページの「SAN-QFS ファイルシステムのマウント解除」
- 219 ページの「トラブルシューティング: SAN-QFS ファイルシステムのマウント解除と SANergy ファイルホールド」

- 220 ページの「SAN-QFS ファイルシステムにおけるブロック割り当て」
- 220 ページの「SAN-QFS ファイルシステムにおけるファイルデータとファイル属性」
- 220 ページの「samgrowfs(1M) による SAN-QFS ファイルシステムの拡張」
- 221 ページの「SAN-QFS 共有ファイルシステムと Sun StorageTek QFS 共有ファイルシステムの比較」

事前確認

SAN-QFS ファイルシステムを使用可能にする場合は、次の留意事項を念頭に置き、これに従って計画を立ててください。

- SAN-QFS ファイルシステムでの使用を前提に構成したディスクは、ボリュームマネージャーでは制御できません。
- SAN-QFS 環境で使用可能にするか、同環境に新たに含める Sun StorageTek QFS メタデータサーバーは、SANergy メタデータコントローラとして構成する必要があります。
- SAN-QFS ファイルシステムはセグメント化したファイルを認識しません。その結果、SAN-QFS 環境でセグメント化ファイルを使用すると予期しない動作になることがあります。
- Sun StorageTek QFS の mcf ファイルで ms または md 装置に分類される装置は、SAN-QFS ファイルシステムでサポートされません。

注 – このマニュアルは、Solaris 以外のクライアントを SANergy ソフトウェアと NFS ソフトウェアのホストにしてファイルシステム共有を行う構成を想定しているため、説明および事例がその想定に準じたものとなっています。Solaris 以外のクライアントを NFS ソフトウェアのホストにするのではなく、Samba ソフトウェアのホストにする場合は、Samba のマニュアルを参照してください。

SAN-QFS ファイルシステムの使用開始

ここでは、SAN-QFS ファイルシステムを使用可能にする方法について説明します。次に示す手順を記載の順序に従って実行してください。

- 215 ページの「メタデータコントローラで SAN-QFS ファイルシステムを有効にする」
- 216 ページの「クライアントで SAN-QFS ファイルシステムを有効にする」
- 216 ページの「クライアントで SANergy ソフトウェアをインストールする」

▼ メタデータコントローラで SAN-QFS ファイルシステムを有効にする

SAN-QFS ファイルシステムの使用環境では、SANergy メタデータコントローラ (MDC) としての機能を割り当てるホストシステムを 1 台選択しますが、ここが Sun StorageTek QFS ファイルシステムの常駐先となります。

1. スーパーユーザーとして、Sun StorageTek QFS ファイルシステムが常駐しているホストにログインします。
2. Sun StorageTek QFS ファイルシステムがテスト済みで、かつその機能が完全であることを確認します。
3. SANergy ソフトウェアのインストールと構成を行います。
詳細は、SANergy のマニュアルを参照してください。
4. `pkginfo(1)` コマンドを使用して、SANergy ソフトウェアのリリースレベルを確認します。

```
# pkginfo -l SANergy
```

5. 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。
`mount(1M)` コマンドを使用して、ファイルシステムをマウントするか、すでにマウント状態であることを確認します。
6. `share(1M)` コマンドを次の形式で使用して、クライアントホストへの NFS アクセスを有効にします。

```
MDC# share -F nfs -d qfs-file-system-name /mount-point
```

`qfs-file-system-name` には、`qfs1` などの Sun StorageTek QFS ファイルシステム名を指定します。`share(1M)` コマンドの詳細は、`share(1M)` または `share_nfs(1M)` のマニュアルページを参照してください。

`mount-point` には、`qfs-file-system-name` のマウントポイントを指定します。

7. Microsoft Windows クライアントに接続し、かつセキュリティー機能および名前空間機能を使用可能にする場合は、NFS ではなく Samba を構成します。

そのためには、`/etc/init.d/sanergy` ファイルに `SANERGY_SMBPATH` 環境変数を追加し、Samba 構成ファイルの場所を示すようにします。たとえば、Samba 構成ファイルの名前が `/etc/swf/smb.conf` の場合は、`/etc/init.d/sanergy` ファイルの先頭に次の行を追加する必要があります。

```
SANERGY_SMBPATH=/etc/sfw/smb.conf  
export SANERGY_SMBPATH
```

8. (省略可能) MDC のファイルシステムテーブル (/etc/dfs/dfstab) を編集して、起動時のアクセスを有効にします。

このアクセスを起動時に自動的に有効にする場合は、この操作を実行してください。

▼ クライアントで SAN-QFS ファイルシステムを有効にする

MDC で SAN-QFS ファイルシステムを有効にすると、クライアントホストでも有効にできるようになります。SAN-QFS ファイルシステムは、IRIX、Microsoft Windows、AIX、Linux の各ホストなどいくつかのクライアントホストをサポートしています。サポート対象クライアントの具体例については、ご購入先にお問い合わせください。

どのクライアントもそれぞれ異なる動作特性を有しています。ここでは、一般的な条件を前提として、クライアントで SAN-QFS ファイルシステムを有効にする操作について説明します。使用クライアントの具体的な事柄については、当該クライアントに付属のマニュアルを参照してください。

1. 各クライアントホストにログインします。
2. 各クライアントのシステムデフォルト表を編集し、SAN-QFS ファイルシステムを追加します。

Solaris OS の場合であれば、各クライアントの /etc/vfstab ファイルを編集し、次のように Sun StorageTek QFS ファイルシステムの名前を追加します。

```
server:/qfs1 - /qfs1 nfs - yes noac,hard,intr,timeo=1000
```

別のオペレーティングシステムプラットフォームでは、システムデフォルト表の常駐先が /etc/vfstab 以外のファイルになっている場合があります。Linux システムであれば、システムデフォルト表の常駐ファイルは /etc/fstab です。

/etc/vfstab ファイルの編集の詳細については、『Sun StorageTek QFS インストールおよびアップグレードの手引き』を参照してください。NFS クライアントの必須または推奨マウントオプションについては、SANergy のマニュアルを参照してください。

▼ クライアントで SANergy ソフトウェアをインストールする

クライアントホストでファイルシステムを有効にすると、そのクライアントに SANergy ソフトウェアをインストールできます。ここでは、一般的な条件の下で SANergy をインストールする手順について説明します。

1. SANergy ソフトウェアのインストールと構成を行います。
詳細は、SANergy のマニュアルを参照してください。

2. `mount` コマンドを使用して、ファイルシステムの NFS マウントを行います。

例:

```
# mount host:/mount-point/ local-mount-point
```

`host` には、MDC を指定します。

`mount-point` には、MDC における Sun StorageTek QFS ファイルシステムのマウントポイントを指定します。

`local-mount-point` には、SANergy クライアントにおけるマウントポイントを指定します。

3. SANergy を、その `fuse` コマンドを使用して融合します。

```
# fuse |mount-point
```

`mount-point` には、SANergy クライアントにおけるマウントポイントを指定します。

SAN-QFS ファイルシステムのマウント解除

ここでは、SANergy ソフトウェアを使用中の SAN-QFS ファイルシステムをマウント解除する方法について説明します。次に示す手順を記載の順序に従って実行してください。

- 217 ページの「SANergy クライアントの SAN-QFS ファイルシステムをマウント解除する」
- 218 ページの「メタデータコントローラの SAN-QFS ファイルシステムをマウント解除する」
- 218 ページの「Sun StorageTek QFS クライアントの SAN-QFS ファイルシステムをマウント解除する」
- 219 ページの「Sun StorageTek QFS サーバーの SAN-QFS ファイルシステムをマウント解除する」

▼ SANergy クライアントの SAN-QFS ファイルシステムをマウント解除する

SAN-QFS ファイルシステムをマウント解除する各クライアントホストに対して次の手順を実行します。

1. スーパーユーザー として、クライアントホストにログインします。

2. SAN-QFS ファイルシステムを、SANergy の `unfuse` コマンドを使用して、同ソフトウェアとの融合を解除します。

```
# unfuse |mount-point
```

mount-point には、SANergy クライアントにおけるマウントポイントを指定します。

3. `umount(1M)` コマンドを使用して、ファイルシステムを NFS からマウント解除します。

```
# umount host:/mount-point/ local-mount-point
```

host には、MDC を指定します。

mount-point には、MDC における Sun StorageTek QFS ファイルシステムのマウントポイントを指定します。

local-mount-point には、SANergy クライアントにおけるマウントポイントを指定します。

▼ メタデータコントローラの SAN-QFS ファイルシステムをマウント解除する

1. `superuser` として、MDC システムにログインします。
2. `unshare(1M)` コマンドを使用して、クライアントホストへの NFS アクセスを無効にします。

```
MDC# unshare qfs-file-system-name /mount-point
```

qfs-file-system-name には、`qfs1` などの Sun StorageTek QFS ファイルシステム名を指定します。`unshare(1M)` コマンドについては、`unshare(1M)` のマニュアルページを参照してください。

mount-point には、*qfs-file-system-name* のマウントポイントを指定します。

▼ Sun StorageTek QFS クライアントの SAN-QFS ファイルシステムをマウント解除する

該当する各クライアントホストに対して、次の手順を実行します。

1. スーパーユーザー として、Sun StorageTek QFS クライアントホストにログインします。

2. `umount(1M)` コマンドを使用して、ファイルシステムをマウント解除します。
例:

```
# umount /qfs1
```

▼ Sun StorageTek QFS サーバーの SAN-QFS ファイルシステムをマウント解除する

1. スーパーユーザーとして、Sun StorageTek QFS ファイルシステムが常駐しているホストシステムにログインします。
2. `umount(1M)` コマンドを使用して、ファイルシステムをマウント解除します。

トラブルシューティング: SAN-QFS ファイルシステムのマウント解除と SANergy ファイルホールド

SANergy ソフトウェアは、Sun StorageTek QFS ファイルにホールドを発行して暫定的に内部に保管し、緊急のアクセスに備えています。ホールドが有効な状態で SANergy に障害が発生した場合は、SAN-QFS ファイルシステムをマウント解除できません。SAN-QFS ファイルシステムをマウント解除できない場合は、`/var/adm/messages` ファイルの中に、SANergy ホールドが解除されていないことを示すコンソールメッセージがないかどうかを確認してください。

ホールドは SANergy ファイル共有機能で可能なかぎり消去しますが、緊急時や SANergy ファイル共有システムに障害がある場合は、次の手順に従って再起動を回避します。

▼ SANergy ファイルホールドが存在する状態でファイルシステムをマウント解除する

1. `unshare(1M)` コマンドを使用して、NFS アクセスを無効にします。
2. `samunhold(1M)` コマンドを使用して、SANergy ファイルシステムホールドを解除します。
このコマンドについては、`samunhold(1M)` のマニュアルページを参照してください。
3. `umount(1M)` コマンドを使用して、ファイルシステムをマウント解除します。

SAN-QFS ファイルシステムにおけるブロック割り当て

SANergy ソフトウェアでは、ブロック割り当ては適用されないため、ブロック割り当てを考慮せずにファイルを書き込むことができます。割り当ての詳細については、102 ページの「割り当ての有効化」を参照してください。

SAN-QFS ファイルシステムにおけるファイルデータとファイル属性

SANergy ソフトウェアは、NFS ソフトウェアを使用してメタデータ処理を行うため、ファイルデータおよびファイル属性には、NFS のクローズ / オープン一貫性モデルが適用されます。SANergy クライアントのファイルデータおよび属性では、ファイルを開く動作に関する POSIX 一貫性モデルがサポートされていません。

samgrowfs(1M) による SAN-QFS ファイルシステムの拡張

samgrowfs(1M) コマンドを使用すると、SAN-QFS ファイルシステムのサイズを拡大することができます。この作業を実行するには、66 ページの「ファイルシステムへのディスクキャッシュの追加」に説明された手順に従います。



注意 – 手順に従うときには、mcf ファイルの行ごとの装置の順序がファイルシステムのスーパーブロックに指定されている装置の順序と一致している必要があることに注意してください。

samgrowfs(1M) コマンドを実行すると、実行前に mcf ファイルにあった装置はスーパーブロック内の位置を保ちます。新しい装置は、そのあとのエントリに検出順に書き込まれます。

この新しい順序がスーパーブロックの順序と一致しないと、SAN-QFS ファイルシステムを拡張できません。

SAN-QFS 共有ファイルシステムと Sun StorageTek QFS 共有ファイルシステムの比較

SAN-QFS 共有ファイルシステムと Sun StorageTek QFS 共有ファイルシステムは、次のような類似点があります。

- ファイルの書き込みができる
- 主ファイルシステムホストがデータの書き込みを行わないことが望ましいデータの取り込み環境で役立つ
- ファイルへの書き込みで競合がある環境で有利

表 7-3 に、両ファイルシステムの相違点を示します。

表 7-3 SAN-QFS 共有ファイルシステムと Sun StorageTek QFS 共有ファイルシステム

SAN-QFS ファイルシステム	Sun StorageTek QFS 共有ファイルシステム
NFS プロトコルをメタデータに使用。	自然メタデータを使用。
異機種システム混在環境に適しています (つまり、Sun システム以外のホストがある場合)。	同機種だけの Solaris OS 環境に適しています。
複数の異機種ホストによるデータ書き込みが可能であることが要件の環境で役立ちます。	複数のホストが同時に同一ファイルに書き込む必要がある場合に適しています。

入出力タイプについて

Sun StorageTek QFS ファイルシステムは、ページ入出力、直接入出力、および双方の切り替えに対応しています。これらの入出力タイプについて次に説明します。

ページ入出力

ページ入出力を使用する場合、ユーザーデータは仮想メモリーページにキャッシュされ、カーネルがデータをディスクに書き込みます。標準 Solaris OS インタフェースによってページ入出力が管理されます。デフォルトではページ入出力 (バッファ入出力またはキャッシュ入出力とも呼ばれる) が選択されます。

直接入出力

直接入出力プロセスでは、データ転送がユーザーのバッファとディスクとの間で直接行われます。つまり、システムでかかる時間が非常に短くなります。パフォーマンスのために、直接入出力は、ブロックが境界割り当てされた大容量の逐次入出力の場合にだけ指定してください。

`setfa(1)` コマンドと `sam_setfa(3)` ライブラリルーチンの `-D` オプションを使用して、ファイルやディレクトリの直接入出力属性を設定します。ディレクトリに設定する場合、そのディレクトリ内に作成されたファイルおよびディレクトリは直接入出力属性を継承します。`-D` オプションを設定すると、ファイルは直接入出力を行います。

また、Sun Solaris OS の `directio(3C)` 関数呼び出しを使用して、ファイルで直接入出力を選択することもできます。関数呼び出しを使用して直接入出力を有効にする場合、設定が有効なのはファイルがアクティブな間だけです。

ファイルシステムについて直接入出力を有効にするには、次のどちらかを行います。

- `mount(1M)` コマンドで `-o forcedirectio` オプションを指定します。
- `forcedirectio` キーワードを `/etc/vfstab` ファイルのマウントオプション列に挿入するか、`samfs.cmd` ファイル内に指示として使用します。

詳細は、`setfa(1)`、`sam_setfa(3)`、`directio(3C)`、`samfs.cmd(4)`、`mount_samfs(1M)` のマニュアルページを参照してください。

入出力切り換え

デフォルトではページ入出力が優先され、入出力切り換えが使用不可になります。ただし、Sun StorageTek QFS ファイルシステムは自動入出力切り換えをサポートしています。このプロセスでは、サイトで定義された量のページ入出力が行われると、システムが自動的に直接入出力に切り替わります。

入出力切り換えは、入出力動作が頻繁に行われる場合に、ページキャッシュの使用量を減らすことを目的として行います。入出力切り換えを有効にするには、`samu(1M)` を使用するか、`dio_wr_consec` パラメータと `dio_rd_consec` パラメータを `samfs.cmd` ファイルで指示として使用するか、`mount(1M)` コマンドでオプションとして使用します。

これらのオプションの詳細は、`mount_samfs(1M)` または `samfs.cmd(4)` のマニュアルページを参照してください。

大容量ファイル転送パフォーマンスの向上

Sun StorageTek QFS ファイルシステムは、各種ファイルサイズを扱えるように調整されています。ファイルシステム設定を有効にして、大容量ファイルのディスクファイル転送のパフォーマンスを向上させることができます。

注 – 本稼働環境以外でパフォーマンス調整を試すことをお勧めします。変数調整を誤ると、システム全体に想定外の影響を与える可能性があります。

Sun™ Enterprise Services (SES) サポート契約がある場合は、SES に連絡の上でパフォーマンス調整パラメータを変更してください。

▼ ファイル転送パフォーマンスを向上させる

1. 装置の最大読み取り/書き込み指示を設定します。

デバイスドライバが 1 回に読み書きする最大バイト数は、Solaris の `/etc/system` ファイルの `maxphys` パラメータで制御します。`maxphys` パラメータのデフォルト値は Sun Solaris OS のバージョンによって異なりますが、通常は 128K バイト前後です。

次の行を `/etc/system` に追加し、`maxphys` を 8M バイトに設定します。

```
set maxphys = 0x800000
```

注 – `maxphys` の値は、2 の累乗に設定してください。

2. SCSI ディスクの最大転送パラメータを設定します。

sd ドライバが、/kernel/drv/sd.conf ファイルの sd_max_xfer_size 定義を参照することによって、特定のファイルの大容量転送が実現します。この定義がない場合は、sd デバイスドライバ定義 sd_max_xfer_size に定義されている値 (1024 × 1024 バイト) が使用されます。

大容量転送を有効にし、促進するには、/kernel/drv/sd.conf ファイルの最後に次の行を追加します。

```
sd_max_xfer_size=0x800000;
```

3. ファイバディスクの最大転送パラメータを設定します。

ssd ドライバが、/kernel/drv/ssd.conf ファイルの ssd_max_xfer_size 定義を参照することによって、指定ファイルの大容量転送が実現します。この定義がない場合は、ssd デバイスドライバ定義 ssd_max_xfer_size に定義されている値 (1024 × 1024 バイト) が使用されます。

/kernel/drv/ssd.conf ファイルの最後に次の行を追加します。

```
ssd_max_xfer_size=0x800000;
```

注 – Solaris 10 x86 プラットフォームでは、この変更は /kernel/drv/sd.conf ファイル内で行います。最大転送サイズが 8M バイトの場合は、次の行を追加します。
sd_max_xfer_size=0x800000

4. システムを再起動します。

5. writebehind パラメータを設定します。

この操作はページ入出力だけに影響します。

Sun StorageTek QFS ファイルシステムでページ入出力を実行するときファイルシステムが後書きするバイト数は、writebehind パラメータで指定します。writebehind の値を RAID の読み取り/変更/書き込み値の倍数にすると、パフォーマンスが向上します。

このパラメータはキロバイト単位で指定し、8K バイトの倍数に切り捨てられます。このパラメータを設定しても、直接入出力の実行時には無視されます。デフォルトの writebehind 値は 512K バイトです。この値は、大容量ブロックの逐次入出力に適しています。

writebehind サイズは、ハードウェアおよびソフトウェア RAID 5 両方の RAID -5 ストライプサイズの倍数に設定します。RAID -5 ストライプサイズは、構成されているストライプ幅にデータディスク数を乗じた値です。

たとえば、3つのデータディスクと1つのパリティディスク (3 + 1) から構成される、ストライプ幅 16K バイトの RAID -5 装置を構成していると仮定します。writebehind 値には 48K バイトや 96K バイトなどの 48 の倍数を指定し、読み取り/変更/書き込み RAID -5 パリティ生成のオーバーヘッドを回避する必要があります。

Sun StorageTek QFS ファイルシステムの場合、DAU (sammkfs(1M) -a コマンド) も RAID -5 ストライプサイズの倍数にする必要があります。このように割り当てることで、ブロックが連続して確保されます。

writebehind サイズをリセットしたあとで、システムパフォーマンスをテストする必要があります。次の例は、ディスク書き込みのタイミングのテストです。

```
# timex dd if=/dev/zero of=/sam/myfile bs=256k count=2048
```

writebehind パラメータを設定するには、マウントオプション、samfs.cmd ファイル、/etc/vfstab ファイル、samu(1M) ユーティリティのいずれかを使用します。マウントオプションでの指定の詳細については、mount_samfs(1M) のマニュアルページの -o writebehind=*n* オプションを参照してください。samfs.cmd ファイルでの指定の詳細は、samfs.cmd(4) のマニュアルページを参照してください。samu(1M) での指定の詳細は、samu(1M) のマニュアルページを参照してください。

6. readahead パラメータを設定します。

この操作はページ入出力だけに影響します。

Sun StorageTek QFS ファイルシステムでページ入出力を実行するときに、当該ファイルシステムが先読みするバイト数は、readahead パラメータで指定します。このパラメータはキロバイト単位で指定し、8K バイトの倍数に切り捨てられます。このパラメータを設定しても、直接入出力の実行時には無視されます。

readahead パラメータのサイズを増やすと、ある時点までは大容量ファイル転送のパフォーマンスが向上します。転送速度が最大になるように readahead のサイズをリセットしたあとで、システムのパフォーマンスをテストする必要があります。次の例は、ディスク読み取りのタイミングのテストです。

```
# timex dd if=/sam/myfile of=/dev/null bs=256k
```

環境でさまざまな readahead サイズをテストする必要があります。readahead パラメータはページ入出力の入出力パフォーマンスが向上するサイズに設定する必要がありますが、大きくしすぎるとパフォーマンスが低下します。readahead 値を設定

するときは、メモリー容量や並行ストリーム数を考慮することも重要です。設定した `readahead` 値にストリーム数を乗じるとメモリー容量を上回る場合は、ページスラッシュの原因になります。

デフォルトの `readahead` の値は 1024K バイトです。この値は、大容量ブロックの逐次入出力に適しています。小容量のランダム入出力アプリケーションでは、`readahead` は通常の要求サイズに設定します。データベースアプリケーションでは独自の先読みが行われるため、このようなアプリケーションについては `readahead` を 0 に設定します。

`readahead` 設定はマウントオプション、`samfs.cmd` ファイル、`/etc/vfstab` ファイル、または `samu(1M)` ユーティリティーのコマンドで有効にできます。マウントオプションによるこの設定の有効化の詳細については、`mount_samfs(1M)` のマニュアルページの `-o readahead=n` オプションを参照してください。`samfs.cmd` ファイルによるこの設定の有効化の詳細は、`samfs.cmd(4)` のマニュアルページを参照してください。`samu(1M)` によるこの設定の有効化の詳細は、`samu(1M)` のマニュアルページを参照してください。

7. ストライプ幅を設定します。

ファイルシステムのストライプ幅は、`mount(1M)` コマンドの `-o stripe=n` オプションで指定します。ストライプ幅は、ディスク割り当て単位 (DAU) のサイズに基づきます。 n 引数は、 $n \times$ DAU バイトが装置に書き込まれてから、書き込みが次の装置に切り換わることを指定します。DAU のサイズは、`sammkfs(1M)` コマンドの `-a` オプションを使用してファイルシステムを初期化すると、設定されます。

`-o stripe=0` と設定すると、ラウンドロビン式割り当てを使用してファイルシステムの装置にファイルが割り当てられます。この方法では、ある装置がいっぱいになるまで、各ファイルが完全にその装置に割り当てられます。ラウンドロビン式は、マルチストリーム環境に適した設定です。`-o stripe=n` を 0 よりも大きな整数に設定すると、ストライプ化方式を使用してファイルシステムにファイルが割り当てられません。適切な `-o stripe=n` の設定を判別するには、さまざまな設定を試してパフォーマンスの統計を確認してください。ストライプ化は、必要な帯域幅を備えた既製アプリケーションに適した設定です。

ストライプ幅は、`/etc/vfstab` ファイルまたは `samfs.cmd` ファイルでも設定できます。

`mount(1M)` コマンドの詳細については、`mount_samfs(1M)` のマニュアルページを参照してください。`samfs.cmd` ファイルの詳細は、`samfs.cmd(4)` のマニュアルページを参照してください。

Qwrite 機能の有効化

デフォルトの Sun StorageTek QFS ファイルシステムでは、1 ファイルの同時読み取り / 書き込みは無効になっています。これは、UNIX vnode インタフェース標準で定義されているモードです。排他的アクセスが与えられるのは 1 つの書き込み操作だ

けで、そのほかの書き込み側および読み取り側は待機する必要があります。Qwrite を使用すると、さまざまなスレッドから同一ファイルへの同時読み取りと書き込みが可能になります。

Qwrite 機能は、同一ファイルへの複数の同時トランザクションを可能にするためにデータベースアプリケーションで使用できます。通常、データベースアプリケーションは大容量ファイルを管理し、同一ファイルへの同時読み取りや書き込みを実行します。しかし、ファイルへのシステムコールごとに、カーネル内の読み取り/書き込みロックの取得と解放が繰り返されます。このロックによって、同一ファイルへのオーバーラップした (まったく同時の) 操作が禁止されます。アプリケーションそのものがファイルロックメカニズムを実装している場合は、カーネルのロックメカニズムが入出力を不必要に直列化してパフォーマンスを妨げます。

Qwrite は、`/etc/vfstab` ファイル、`samfs.cmd` ファイル、およびマウントオプションで有効にできます。`mount(1M)` コマンドで `-o qwrite` オプションを指定すると、ファイルシステムのロックメカニズムを迂回し (ネットワークファイルシステム (NFS) 経由でファイルシステムにアクセスするアプリケーションの場合を除く)、アプリケーションがデータアクセスを制御できるようになります。`qwrite` を指定すると、ファイルシステムは、さまざまなスレッドからの同一ファイルへの同時読み取りと書き込みを有効にします。このオプションは、複数の要求をドライブレベルでキューイングすることで入出力パフォーマンスを向上させます。

次の例では、`mount(1M)` コマンドを使用して、データベースファイルシステムで Qwrite を有効にしています。

```
# mount -F samfs -o qwrite /db
```

この機能の詳細は、`samfs.cmd(4)` のマニュアルページの `qwrite` 指示または `mount_samfs(1M)` のマニュアルページの `-o qwrite` オプションを参照してください。

書き込みスロットルの設定

`-o wr_throttle=n` オプションでは、1 ファイルに対する未処理の書き込みデータのキロバイト数が *n* に制限されます。デフォルトでは、Sun StorageTek QFS ファイルシステムでは `wr_throttle` を 16M バイトに設定します。

ファイルの未処理の書き込みデータが *n* バイトになると、そのファイルに書き込もうとするアプリケーションは中断し、十分なバイト数の入出力が完了するまでアプリケーションを再開できません。

サイトに数千のストリームがある場合は(ファイルシステムに数千の NFS 共有ワークステーションがアクセスするなど)、同時に大量のメモリーがディスクにフラッシュされることを防止するため、`-o wr_throttle=n` オプションを調整してください。一般的に、次の式に示すように `-o wr_throttle=n` オプションの $1024 \times n$ 引数にストリーム数を乗じると、ホストシステムの合計メモリーサイズから Solaris OS で必要なメモリーを引いた値よりも小さくなります。

$$\text{ストリーム数} \times n \times 1024 < \text{合計メモリー} - \text{Solaris OS で必要なメモリー}$$

既製アプリケーションでは、メモリーに保持されるページ数が増加するため、デフォルトの 16,384K バイトよりも大きな値を使用してもよいでしょう。

遅延フラッシュ率の設定

逐次書き込みページと書き込みのページの遅延フラッシュ率は、2つのマウントパラメータで制御します。`flush_behind` マウントパラメータと `stage_flush_behind` マウントパラメータは、`samfs.cmd` ファイル、`/etc/vfstab` ファイル、または `mount(1M)` コマンドに指定します。

`flush_behind=n` マウントパラメータでは、最大遅延フラッシュ値を設定します。変更されたページが逐次書き込みされる場合は、非同期でディスクに書き込まれ、Solaris™ ボリュームマネージャー (SVM) レイヤーがページを空にしておくのに役立ちます。この機能を有効にするには、`n` を 16 ~ 8192 の整数に設定します。デフォルトでは、`n` は 0 に設定されており、この機能は使用できません。`n` 引数は、キロバイト単位で指定します。

`stage_flush_behind=n` マウントパラメータでは、最大書き込み遅延フラッシュ値を設定します。書き込みのページが書き込まれるときは、非同期でディスクに書き込まれ、SVM レイヤーがページを空にしておくのに役立ちます。この機能を有効にするには、`n` を 16 ~ 8192 の整数に設定します。デフォルトでは、`n` は 0 に設定されており、この機能は使用できません。`n` 引数は、キロバイト単位で指定します。

これらのマウントパラメータの詳細は、`mount_samfs(1M)` または `samfs.cmd(4)` のマニュアルページを参照してください。

iノードの数とiノードハッシュテーブルの調整

Sun StorageTek QFS ファイルシステムでは、`/etc/system` ファイルに次の 2 つの調整可能なパラメータを設定できます。

- `ninodes`
- `nhino`

これらのパラメータについてデフォルト以外の設定を有効にするには、`/etc/system` ファイルを編集してからシステムを再起動します。

このあとの各項目で、これらのパラメータについて詳しく説明します。

ninodes パラメータ

`ninodes` パラメータは、デフォルトの i ノードの最大数を指定します。`ninodes` の値は、Sun StorageTek QFS ソフトウェアがアプリケーションで多くの i ノードを使用していないときでも自分に割り当てておく、内部的な i ノードの数を指定します。

このパラメータの `/etc/system` ファイルでの書式は、次のとおりです。

```
set samfs:ninodes = value
```

value の範囲は 16 ~ 2000000 です。`ninodes` のデフォルト値は次のいずれかになります。

- `ncsize` 設定に等しい値。`ncsize` パラメータは DLNC (Directory Name Look-up Cache) 内のエントリ数を設定する、Solaris の調整パラメータである。`ncsize` の詳細については、『Solaris カーネルのチェーンアップ・リファレンスマニュアル』を参照してください。
- 2000。`ncsize` の設定が 0 または範囲外の場合、ファイルシステムは `ninodes` を 2000 に設定します。

nhino パラメータ

nhino パラメータは、内部的な i ノードハッシュテーブルのサイズを指定します。

このパラメータの `/etc/system` ファイルでの書式は、次のとおりです。

```
set samfs:nhino = value
```

value の範囲は 1 ~ 1048756 で、2 の累乗 (0 を除く) にする必要があります。nhino のデフォルト値は次のどれかになります。

- `ninodes` の値に等しい値を 8 で割った結果に等しい値を、必要に応じて、もっとも近い 2 の累乗の値に切り上げます。たとえば、`/etc/system` に次の行があるとします。

```
set samfs:ninodes 8000
```

この例では、nhino が設定されていない場合、システムでは 1024 が入ると仮定します。この数値は 8000 を 8 で割り、もっとも近い 2 の累乗 (2^{10}) に切り上げたものです。

- 512. `ninodes` の設定が範囲外の場合、ファイルシステムは nhino を 512 に設定します。

ninodes パラメータと nhino パラメータの設定が必要な場合

(ディレクトリから i ノードを入手するか、NFS ファイルハンドルから i ノード番号を抽出してから) i ノードを番号で検索するとき、Sun StorageTek QFS ファイルシステムは内部の i ノードが入ったキャッシュを検索します。この処理を速めるため、システムは検査する i ノードの数を減らせるようにハッシュテーブルを保持していません。

大きなハッシュテーブルがあると、メモリー使用にそれほど負担をかけることなく、比較や検索の数を減らせます。nhino 値があまりに大きい場合、i ノードのリスト全体を走査する操作を行うときには時間がかかります (inode の同期やマウント解除)。大量のファイルを操作するサイトや大量の NFS I/O を行うサイトでは、これらのパラメータの値をデフォルト値より大きく設定したほうが便利です。

ファイル数の少ないファイルシステムしかないサイトの場合、これらのパラメータはデフォルト値より小さくしたほうが便利です。これには、たとえば 1 つだけでサイズの大きい tar(1) ファイルにほかのファイルシステムのバックアップを作成するような場合があります。

付録 A

トラブルシューティング Sun StorageTek QFS

この付録では、Sun StorageTek QFS ファイルシステムの問題のトラブルシューティングに使用可能なツールと手順について説明します。特に、次の項目があります。

- 231 ページの「ファイルシステムの完全性の確認とファイルシステムの修復」
- 233 ページの「共有ファイルシステムでの失敗またはハングアップした `sammkfs(1M)` コマンドまたは `mount(1M)` コマンドのトラブルシューティング」
- 249 ページの「Linux クライアントのトラブルシューティング」

ファイルシステムの完全性の確認とファイルシステムの修復

Sun StorageTek QFS ファイルシステムは、ファイルシステムの動作に不可欠な、ディレクトリ、間接ブロック、および `i` ノードのレコードに妥当性検査データを書き込みます。ディレクトリの検索時にファイルシステムが破損を検出した場合は、EDOM エラーが発生し、ディレクトリは処理されません。間接ブロックが有効ではない場合は、ENOCSEI エラーが発生し、ファイルは処理されません。表 A-1 は、これらのエラー表示の概要を示しています。

表 A-1 エラー表示

エラー	Solaris OS での意味	Sun StorageTek QFS での意味
EDOM	引数がドメイン外。	妥当性検査レコードの値が範囲外。
ENOCSEI	使用可能な CSI 構造がない。	構造間のリンクが無効。

さらに、i ノードの妥当性検査が行われ、ディレクトリと相互に検査されます。

次のファイルでエラー状況を監視します。

- 表 A-1 に示すエラーについて、`/etc/syslog.conf` で指定されたログファイル
- 装置エラーについて、`/var/adm/messages` ファイル

矛盾があった場合は、ファイルシステムをマウント解除し、`samfsck(1M)` コマンドを使用して調べる必要があります。

注 – `samfsck(1M)` コマンドは、マウントされたファイルシステムで実行できますが、その結果は信頼できません。このため、マウント解除したファイルシステムのみでこのコマンドを実行することをお勧めします。

▼ ファイルシステムを検査する

- `samfsck(1M)` コマンドを使用して、ファイルシステムの検査を実行します。
このコマンドは、次の形式で使用します。

```
samfsck -V family-set-name
```

family-set-name には、`mcf` ファイルに指定されているファイルシステムの名前を指定します。

`samfsck(1M)` の出力を画面とファイルの両方に送るには、次のように `tee(1)` コマンドと組み合わせて使用します。

- C シェルの場合

```
# samfsck -V family-set-name |& tee file
```

- Bourne シェルの場合

```
# samfsck -V family-set-name 2>&1 | tee file
```

`samfsck(1M)` によって戻される致命的でないエラーは、前に `NOTICE` と付いています。致命的でないエラーには、紛失ブロックやオーファンがあります。`NOTICE` エラーが戻されても、ファイルシステムの整合性は保たれます。これらの致命的でないエラーは、定期保守のための停止中に修復できます。

致命的なエラーは、前に ALERT が付いています。このようなエラーには、重複ブロック、無効なディレクトリ、無効な間接ブロックがあります。これらのエラーが発生した場合、ファイルシステムの整合性が失なわれています。ハードウェアの故障が原因でない ALERT エラーがある場合は、ご購入先にご連絡ください。

`samfsck(1M)` コマンドがファイルシステムの破損を検出し、ALERT メッセージを返した場合は、破損の原因を調べてください。ハードウェアの障害の場合は、ファイルシステムを修復する前にハードウェアを修理します。

`samfsck(1M)` コマンドと `tee(1)` コマンドの詳細は、`samfsck(1M)` と `tee(1)` のマニュアルページを参照してください。

▼ ファイルシステムを修復する

1. `umount(1M)` コマンドを使用して、ファイルシステムをマウント解除します。
ファイルシステムがマウントされていない場合は、`samfsck(1M)` コマンドを実行します。ファイルシステムのマウント解除については、65 ページの「ファイルシステムのマウント解除」を参照してください。
2. `samfsck(1M)` コマンドを使用して、ファイルシステムを修復します。共有ファイルシステムを修復する場合は、メタデータサーバーからこのコマンドを実行します。
次の書式で `samfsck(1M)` コマンドを実行して、ファイルシステムを修復できます。

```
# samfsck -F -V fsname
```

fsname には、`mcf` ファイルで指定されているファイルシステムの名前を指定します。

共有ファイルシステムでの失敗またはハングアップした `sammkfs(1M)` コマンドまたは `mount(1M)` コマンドのトラブルシューティング

この節では、`sammkfs(1M)` コマンドまたは `mount(1M)` コマンドが失敗したときや、`mount(1M)` コマンドが共有ファイルシステムでハングアップしたときの対処方法について説明します。

この節で説明する手順は、クライアントホストでもサーバーでも実行できます。メタデータサーバーだけで実行できるコマンドは、先頭に `server#` プロンプトが付いています。

sammkfs(1M) コマンドの失敗からの回復

sammkfs(1M) コマンドによって、エラーが戻されたり、予期しない一連の装置を初期化するというメッセージが戻される場合は、この項の手順を実行する必要があります。mcf ファイルを確認し、mcf ファイルの変更をシステムに伝達する手順も含まれています。

▼ mcf ファイルを確認し、mcf ファイルの変更をシステムに伝達する

1. `sam-fsd(1M)` コマンドを使用して、mcf ファイルを確認します。

例:

```
# sam-fsd
```

`sam-fsd(1M)` コマンドの出力を確認し、修正の必要なエラーがあるかどうか調べます。

2. `sam-fsd(1M)` コマンドの出力で、`/etc/opt/SUNWsamfs/mcf` ファイルにエラーがあることが示された場合は、mcf ファイルを編集して問題を解決します。
3. `sam-fsd(1M)` コマンドをもう一度使用して、mcf ファイルを確認します。
`sam-fsd(1M)` コマンドの出力で mcf ファイルが正しいと示されるまで、この手順の手順 1、手順 2、および手順 3 を繰り返します。
4. `samd(1M) config` コマンドを実行します。

これは、`sam-fsd` デーモンに構成の変更を知らせ、mcf ファイルの変更を伝達するために必要です。

例:

```
# samd config
```

mount(1M) コマンドの失敗からの回復

mount(1M) コマンドが失敗する理由はいくつかあります。この項では、マウントの問題を解決するために実行できるいくつかの処置について説明します。mount(1M) コマンドが失敗ではなくハングアップした場合は、240 ページの「ハングアップした mount(1M) コマンドからの回復」を参照してください。

失敗した mount(1M) の動作とその対処方法のいくつかを次に示します。

- mount(1M) コマンドが失敗し、クライアントで「Shared server is not mounted」というメッセージが表示された場合は、サーバーホストを判別し、ファイルシステムをメタデータサーバーにマウントします。
- mount コマンドが失敗し、ファイルシステムと mcf ファイルが一致しないというメッセージが表示された場合は、次のことを確認します。
 - mcf ファイルの構文が有効かどうか。詳細は、234 ページの「mcf ファイルを確認し、mcf ファイルの変更をシステムに伝達する」を参照してください。
 - 最近 mcf ファイルに対して加えた変更内容が有効であり、設定済みになっているかどうか。詳細は、234 ページの「mcf ファイルを確認し、mcf ファイルの変更をシステムに伝達する」を参照してください。
 - mcf ファイルが、クライアントでの違いに合わせて装置名やコントローラの番号を調整している、サーバーの mcf ファイルと一致しているかどうか。samfsconfig(1M) コマンドを使用すると、このような一部の問題を診断できます。samfsconfig(1M) コマンドの使用方法については、239 ページの「samfsconfig(1M) コマンドを使用する」を参照してください。
- そのほかの理由で mount(1M) コマンドが失敗した場合は、この項で説明する手順を使用して、mount(1M) コマンドが正常に作動するために機能している必要のあるシステムの設定を確認してください。手順は次のとおりです。
 - 235 ページの「ファイルシステムがマウント可能かどうかを確認する」
 - 237 ページの「samfsinfo(1M) コマンドと samsharefs(1M) コマンドを使用する」
 - 239 ページの「samfsconfig(1M) コマンドを使用する」

▼ ファイルシステムがマウント可能かどうかを確認する

この手順でエラーが見つからなかった場合は、237 ページの「samfsinfo(1M) コマンドと samsharefs(1M) コマンドを使用する」を行うと、ファイルシステムが作成され、共有ホストファイルが正しく初期化されているかどうかの確認に役立ちます。

次の手順で、mount(1M) コマンドが失敗したときに確認する項目を示します。

1. マウントポイントディレクトリが存在していることを確認します。

これには、複数の方法があります。たとえば、次の形式で `ls(1)` コマンドを実行します。

```
ls -ld mountpoint
```

`mountpoint` には、Sun StorageTek QFS 共有ファイルシステムのマウントポイントの名前を指定します。

`ls(1)` コマンドの出力を確認するときに、出力にディレクトリがアクセスモード 755 で表示されていることを確認します。つまり、`drwxr-xr-x` というコードが必要です。コード例 A-1 に出力例を示します。

コード例 A-1 アクセスモード値

```
# ls -ld /sharefs1
drwxr-xr-x  2 root      sys           512 Mar 19 10:46 /sharefs1
```

アクセスがこのレベルでない場合は、次の `chmod(1)` コマンドを入力します。

```
# chmod 755 mountpoint
```

`mountpoint` には、Sun StorageTek QFS 共有ファイルシステムのマウントポイントの名前を指定します。

2. `/etc/vfstab` ファイルに、ファイルシステムに対応するエントリがあることを確認します。

コード例 A-2 に、`sharefs1` という名前の共有ファイルシステムのエントリを示します。

コード例 A-2 `/etc/vfstab` ファイルの例

```
# File /etc/vfstab
# FS name  FS to fsck  Mnt pt FS type  fsck pass  Mt@boot  Mt params
sharefs1   -                /sharefs1 samfs -        yes      shared,bg
```

`shared` フラグが、`/etc/vfstab` ファイルの共有ファイルシステムのエントリの「Mount Parameters」フィールドにあることを確認します。

3. マウントポイントディレクトリが、NFS で使用するために共有されていないことを確認します。

マウントポイントが共有されている場合は、`unshare(1M)` コマンドを使用して共有を解除します。

例:

```
# unshare mountpoint
```

mountpoint には、Sun StorageTek QFS 共有ファイルシステムのマウントポイントの名前を指定します。

▼ samfsinfo(1M) コマンドと samsharefs(1M) コマンドを使用する

この手順では、これらのコマンドの出力の分析方法を示します。

1. サーバーで `samfsinfo(1M)` コマンドを入力します。

このコマンドは、次の形式で使用します。

```
samfsinfo filesystem
```

filesystem には、mcf ファイルで指定した Sun StorageTek QFS 共有ファイルシステムの名前を指定します。コード例 A-3 に、`samfsinfo(1M)` コマンドと出力を示します。

コード例 A-3 samfsinfo(1M) コマンドの例

```
titan-server# samfsinfo sharefs1
samfsinfo: filesystem sharefs1 is mounted.
name:      sharefs1      version:      2      shared
time:      Mon Apr 29 15:12:18 2002
count:     3
capacity:  10d84000      DAU:           64
space:     10180400
meta capacity: 009fe200      meta DAU:     16
meta space: 009f6c60
ord  eq  capacity  space  device
1   11  086c0000  080c39b0  /dev/dsk/c1t2100002037E9C296d0s6
2   12  086c4000  080bca50  /dev/dsk/c3t50020F2300005D22d0s6
3   13  086c4000  080a9650  /dev/dsk/c3t50020F2300006099d0s6
4   14  086c4000  08600000  /dev/dsk/c3t50020F230000651Cd0s6
```

コード例 A-3 の出力には、次の行に `shared` キーワードが示されています。

```
name:sharefs1      version:2      shared
```

次の行のあとに表示されるファイルシステムの装置、順序、装置番号のリストにも注意してください。

```
ord  eq  capacity      space  device
```

これらの番号が、ファイルシステムの `mcf(4)` エントリの装置と対応していることを確認してください。

2. サーバーで `samsharefs(1M)` コマンドを入力します。

このコマンドは、次の形式で使用します。

```
samsharefs -R filesystem
```

filesystem には、`mcf` ファイルで指定した Sun StorageTek QFS 共有ファイルシステムの名前を指定します。コード例 A-4 に、`samsharefs(1M)` コマンドと出力を示します。

コード例 A-4 `samsharefs(1M)` コマンドの例

```
titan-server# samsharefs -R sharefs1
#
# Host file for family set `sharefs1'
#
# Version: 3      Generation: 50      Count: 4
# Server = host 0/titan, length = 216
#
titan 173.26.2.129,titan.foo.com 1 - server
tethys 173.26.2.130,tethys.foo.com 2 -
dione dione.foo.com 0 -
mimas mimas.foo.com 0 -
```

`samfsinfo(1M)` コマンドまたは `samsharefs(1M)` コマンドの診断出力について、次の情報を考慮してください。

- どちらかのコマンドで診断メッセージまたはエラーメッセージが発行された場合は、エラーを解決してください。`samfsinfo(1M)` コマンドの出力に `shared` キーワードが含まれることを確認します。

- これらのコマンドは、ホストの mcf(4) エントリにファイルシステムに対する nodev 装置のない代替サーバーホストおよびクライアントホストで実行できません。

samfsinfo(1M) コマンドと samsharefs(1M) コマンドで異常が表示されない場合は、239 ページの「samfsconfig(1M) コマンドを使用する」を行います。

▼ samfsconfig(1M) コマンドを使用する

mcf ファイルにファイルシステムに対する nodev デバイスエントリがあるクライアントでは、ファイルシステム全体にアクセスできない場合や、共有ホストファイルに直接アクセスできない場合があります。samfsconfig(1M) コマンドを使用すると、共有ファイルシステムのデータパーティションがアクセス可能かどうかを判別できます。

- samfsconfig(1M) コマンドを実行します。

このコマンドは、次の形式で使います。

```
samfsconfig list-of-devices
```

list-of-devices には、mcf ファイルのファイルシステムエントリにあるデバイスリストを指定します。リストでは複数の装置を空白文字で区切ります。

例 1。コード例 A-5 に、mcf ファイルに nodev エントリがないホスト tethys の mcf ファイルを示します。samfsconfig(1M) コマンドを実行した様子も示してあります。

コード例 A-5 nodev エントリがない場合の samfsconfig(1M) コマンドの例

```
tethys# cat /etc/opt/SUNWsamfs/mcf
sharefs1          10 ma sharefs1 on shared
/dev/dsk/c1t2100002037E9C296d0s6 11 mm sharefs1 -
/dev/dsk/c3t50020F2300005D22d0s6 12 mr sharefs1 -
/dev/dsk/c3t50020F2300006099d0s6 13 mr sharefs1 -
/dev/dsk/c3t50020F230000651Cd0s6 14 mr sharefs1 -
tethys# samfsconfig /dev/dsk/c1t2100002037E9C296d0s6
/dev/dsk/c3t50020F2300005D22d0s6 /dev/dsk/c3t50020F2300006099d0s6
/dev/dsk/c3t50020F230000651Cd0s6
#
# Family Set 'sharefs1' Created Mon Apr 29 15:12:18 2002
#
sharefs1          10 ma sharefs1 - shared
/dev/dsk/c1t2100002037E9C296d0s6 11 mm sharefs1 -
/dev/dsk/c3t50020F2300005D22d0s6 12 mr sharefs1 -
/dev/dsk/c3t50020F2300006099d0s6 13 mr sharefs1 -
/dev/dsk/c3t50020F230000651Cd0s6 14 mr sharefs1 -
```

例 2。コード例 A-6 に、mcf ファイルに nodev エントリのあるホストで使用した samfsconfig(1M) コマンドを示します。

コード例 A-6 nodev エントリがある場合の samfsconfig(1M) コマンドの例

```
dione# cat /etc/opt/SUNWsamfs/mcf
sharefs1                10    ma    sharefs1  on  shared
nodev                   11    mm    sharefs1  -
/dev/dsk/c4t50020F23000055A8d0s3  12    mr    sharefs1  -
/dev/dsk/c4t50020F23000055A8d0s4  13    mr    sharefs1  -
/dev/dsk/c4t50020F23000055A8d0s5  14    mr    sharefs1  -
dione# samfsconfig /dev/dsk/c4t50020F23000055A8d0s3
/dev/dsk/c4t50020F23000055A8d0s4 /dev/dsk/c4t50020F23000055A8d0s5
# Family Set 'sharefs1' Created Mon Apr 29 15:12:18 2002
# Missing slices
# Ordinal 1
# /dev/dsk/c4t50020F23000055A8d0s3  12    mr    sharefs1  -
# /dev/dsk/c4t50020F23000055A8d0s4  13    mr    sharefs1  -
# /dev/dsk/c4t50020F23000055A8d0s5  14    mr    sharefs1  -
```

例 1 と例 2 では、ファイルシステム付属のメタデータ (mm) 装置以外に、ファイルシステムのすべてのスライスが出力に表示されていることを確認します。例 2 ではそのようになっています。

ハングアップした mount(1M) コマンドからの回復

mount(1M) コマンドがハングアップした場合は、この項の手順を実行してください。たとえば、接続エラーや「Server not responding」というメッセージで mount(1M) コマンドが失敗して、30 秒以内に解決できない場合は、mount(1M) コマンドがハングアップします。

ハングアップした mount(1M) コマンドのもっとも一般的な処置を最初に示します。その手順が機能しない場合は、そのあとの手順を実行してください。

▼ ネットワーク接続を確認する

netstat(1M) コマンドを使用して、sam-sharefsd デーモンのネットワーク接続が正しく構成されていることを確認します。

1. メタデータサーバーでスーパーユーザーになります。

4. samsharefs(1M) コマンドを入力して、ホストファイルを確認します。

コード例 A-11 は、samsharefs(1M) コマンドおよび正しい出力を示しています。

コード例 A-8 samsharefs(1M) -R コマンド

```
titan-server# samsharefs -R sharefs1
#
# Host file for family set 'sharefs1'
#
# Version: 3      Generation: 50      Count: 4
# Server = host 0/titan, length = 216
#
titan 173.26.2.129 1 - server
tethys 173.26.2.130 2 -
dione dione 0 -
mimas mimas 0 -
```

システムの出力で、次の点を確認します。

- 出力の 1 列目にホスト名があり、サーバーとして指定されている。
- 2 列目にホストの IP アドレスがある。複数の IP アドレスがある場合は、すべて正しいことを確認する。

5. サーバーで netstat(1M) コマンドを入力します。

コード例 A-9 は、サーバー titan で入力した netstat(1M) コマンドを示しています。

コード例 A-9 サーバーでの netstat(1M) の例

```
titan-server# netstat -a | grep sam-qfs
*.sam-qfs *.*          0      0 24576  0 LISTEN
*.sam-qfs *.*          0      0 24576  0 LISTEN
titan.32834 titan.sam-qfs 32768  0 32768  0 ESTABLISHED
titan.sam-qfs titan.32891 32768  0 32768  0 ESTABLISHED
titan.sam-qfs tethys.32884 24820  0 24820  0 ESTABLISHED
titan.sam-qfs dione.35299 24820  0 24820  0 ESTABLISHED
*.sam-qfs *.*          0      0 24576  0 LISTEN
```

サーバーの netstat(1M) コマンドの出力で、次の内容を確認します。

- 3 つの LISTEN エントリ。
- そのホストに対する 2 つの ESTABLISHED エントリ。
- sam-fsd デーモンが構成され、実行されている各クライアントごとに 1 つの ESTABLISHED エントリ。

この例では、tethys および dione に対する ESTABLISHED エントリがあります。構成されて実行しているクライアントごとに、そのクライアントがマウントされているかどうかにかかわらず、1つの ESTABLISHED エントリを含んでいるようにします。

6. クライアントで netstat(1M) コマンドを入力します。

コード例 A-10 は、クライアント dione で入力した netstat(1M) コマンドを示しています。

コード例 A-10 クライアントでの netstat(1M) コマンド

```
dione-client# netstat -a | grep sam-qfs
*.sam-qfs      *.*           0           0 24576       0 LISTEN
*.sam-qfs      *.*           0           0 24576       0 LISTEN
dione.32831    titan.sam-qfs 24820       0 24820       0 ESTABLISHED
*.sam-qfs      *.*           0           0 24576       0 LISTEN
```

7. 出力で次の内容を確認します。

- 3つの LISTEN エントリ。すべてのエントリが sam-fsd デーモンに対するものです。

- 1つの ESTABLISHED エントリ。

これらの行がある場合は、ネットワーク接続が確立されています。

ESTABLISHED の接続がレポートされない場合は、次の1つ以上の手順を実行します。

- 243 ページの「クライアントがサーバーにアクセスできるかどうかを確認する」
- 245 ページの「サーバーがクライアントにアクセスできるかどうかを確認する」
- 246 ページの「sam-sharefsd トレースログを調査する」

▼ クライアントがサーバーにアクセスできるかどうかを確認する

240 ページの「ネットワーク接続を確認する」で説明した手順で ESTABLISHED という接続が表示されない場合は、この手順を実行します。

1. samsharefs(1M) コマンドを使用して、サーバーのホストファイルを確認します。

samsharefs(1M) コマンドは、ホストの mcf エントリにファイルシステムに対する nodev 装置のない代替サーバーホストおよびクライアントホストで実行できます。この手順では、このコマンドを次の形式で使用します。

```
samsharefs -R filesystem
```

filesystem には、mcf ファイルで指定した Sun StorageTek QFS 共有ファイルシステムの名前を指定します。コード例 A-11 に、samsharefs(1M) -R コマンドを示します。

コード例 A-11 samsharefs(1M) -R コマンド

```
titan-server# samsharefs -R sharefs1
#
# Host file for family set 'sharefs1'
#
# Version: 3      Generation: 50      Count: 4
# Server = host 0/titan, length = 216
#
titan 173.26.2.129 1 - server
tethys 173.26.2.130 2 -
dione dione 0 -
mimas mimas 0 -
```

2. この出力を保存します。

この手順の操作が失敗した場合は、この出力が手順のあとの方で必要になります。

3. 出力が予想と一致することを確認します。

コマンドが失敗した場合は、ファイルシステムが作成されたことを確認します。その場合、次のいずれかが起きる可能性があります。

- mcf ファイルが正しく作成されない。samfsconfig(1M) コマンドを使用して、mcf ファイルが正しいかどうかを検証します。
- ファイルシステムがまったく作成されない。
- 初期ホスト構成ファイルが作成されていない。構成作業には、既存の mcf ファイルの編集、mcf ファイル変更のほかのシステムへの伝達、およびホストファイルの構成があります。

4. 最初の列で、サーバーの名前を含む行を検索します。

5. クライアントから、samsharefs(1M) 出力の 2 番目の列の各エントリに対して ping(1M) コマンドを使用して、サーバーにアクセスできることを確認します。

このコマンドは、次の形式で使用します。

```
ping servername
```

servername には、samsharefs(1M) コマンドの出力の 2 番目の列に示されるサーバー名を指定します。

コード例 A-12 は、ping(1M) コマンドの出力を示しています。

コード例 A-12 samsharefs(1M) 出力に表示されたシステムに対する ping(1M) の使用

```
dione-client# ping 173.26.2.129
ICMP Host Unreachable from gateway dione (131.116.7.218)
for icmp from dione (131.116.7.218) to 173.26.2.129
dione-client# ping titan
titan.foo.com is alive
```

6. ping(1M) コマンドによって到達不能なホストが見つかった場合は、クライアントの `hosts.filesystem.local` ファイルを確認してください。

samsharefs(1M) の出力の 2 列目に複数のエントリがある場合、および一部のエントリにアクセスできない場合は、共有ファイルシステムで使用するエントリの中に、アクセス可能なエントリのみが存在することを確認します。また、そのホストの `/etc/opt/SUNWsamfs/hosts.filesystem.local` ファイルに必要なエントリがあることも確認します。アクセスできないホストをこれらの場所に指定しないようにしてください。

sam-sharefsd デーモンが、アクセスできないサーバーインタフェースに接続しようとする、インストール、再起動、またはファイルシステムホスト再構成のあとでサーバーに接続するときに、大幅に遅延する可能性があります。これは、メタデータサーバーのフェイルオーバー動作に大きな影響を与えます。

コード例 A-13 に、`hosts.sharefs1.local` ファイルを示します。

コード例 A-13 `hosts.filesystem.local` ファイルの確認

```
dione-client# cat /etc/opt/SUNWsamfs/hosts.sharefs1.local
titan          titan # no route to 173.26.2.129
tethys         tethys # no route to 173.26.2.130
```

7. ping(1M) コマンドで到達可能なサーバーインタフェースがないことが見つかった場合は、正しいサーバーインタフェースを有効にしてください。

特定の動作用にサーバーネットワークインタフェースを構成または初期化するか、samsharefs(1M) コマンドを使用して `hosts` ファイル内のインタフェース名を更新し、実際の名前と一致するようにします。

▼ サーバーがクライアントにアクセスできるかどうかを確認する

240 ページの「ネットワーク接続を確認する」の手順で ESTABLISHED という接続が表示されない場合は、この手順を実行します。

1. `samsharefs(1M)` の出力を用意します。

243 ページの「クライアントがサーバーにアクセスできるかどうかを確認する」で生成された出力を使用するか、その最初の手順を使用してもう一度生成します。

2. 最初の列で、サーバーの名前を含む行を検索します。

3. クライアントで `hostname(1M)` コマンドを実行し、その出力が、`samsharefs(1M)` 出力の最初の列の名前と一致することを確認します。

コード例 A-14 は、`hostname(1M)` コマンドおよび出力を示しています。

コード例 A-14 `hostname(1M)` の出力

```
dione-client# hostname
dione
```

4. `hostname(1M)` コマンドの出力が `samsharefs(1M)` の出力の 2 番目の列の名前と一致する場合は、サーバーで `ping(1M)` コマンドを使用して、クライアントに到達できることを確認します。

コード例 A-15 は、`ping(1M)` コマンドおよび出力を示しています。

コード例 A-15 `ping(1M)` の出力

```
titan-server# ping dione
dione is alive
```

コード例 A-13 の第 2 列のすべてのエントリにアクセスできる必要はありませんが、潜在的なサーバーが接続されるすべてのインタフェースが列に指定されていることが必要です。サーバーは、共有ホストファイルで宣言されていないインタフェースからの接続を拒否します。

5. `ping(1M)` コマンドで到達可能なクライアントインタフェースがないことが見つかった場合は、正しいクライアントインタフェースを有効にしてください。

特定の動作用にクライアントネットワークインタフェースを構成または初期化するか、`samsharefs(1M)` コマンドを使用して `hosts` ファイル内のインタフェース名を更新し、実際の名前と一致するようにします。

▼ `sam-sharefsd` トレースログを調査する

トレースログファイルには、`sam-sharefsd(1M)` デーモンが動作中に生成した情報が保存されています。トレースログファイルには接続の試行、受信、拒否、拒絶など、およびホストファイルの変更やメタデータサーバーの変更などそのほかの操作が記録されます。

ログファイルにある問題を追跡するには、ログファイルを使用してホスト同士の動作順序の整合を取ることもあります。ホストのクロックが同期されていると、ログファイルの解析が非常に簡単になります。インストール手順の 1 つは、ネットワーク時刻

デーモン `xntpd(1M)` を有効にするものです。これにより、Sun StorageTek QFS 共有ファイルシステムの動作中に、メタデータサーバーやすべてのクライアントホストのクロックの同期が取られます。

初期構成を設定するときに、トレースログは特に便利です。クライアントログには、実行中の接続試行が表示されます。サーバーログファイル内の対応するメッセージは、Sun StorageTek QFS 共有ファイルシステムのネットワークおよび構成上の問題を診断するための、非常に役立つツールの 1 つです。ログファイルには、よく起きる問題を解決するための診断情報が含まれています。

次の手順では、ほとんどの `mount(1M)` の問題を解決できます。

- 240 ページの「ネットワーク接続を確認する」
- 243 ページの「クライアントがサーバーにアクセスできるかどうかを確認する」
- 245 ページの「サーバーがクライアントにアクセスできるかどうかを確認する」

これらの手順で問題を解決できない場合は、次の手順を実行してください。これらの手順は、サーバーとクライアントの両方のホストで実行します。

1. **ファイル `/var/opt/SUNWsamfs/trace/sam-sharefsd` の存在を確認します。**

このファイルが存在しない場合、またはこのファイルが最近変更されていない場合は、次の操作に進みます。

ファイルが存在する場合は、`tail(1)` などのコマンドを使用してファイルの最後の数行を調べます。疑わしい状態がある場合は、この節の 1 つ以上のほかの手順を使用して問題を調査します。

2. **手順 1 でファイル `/var/opt/SUNWsamfs/trace/sam-sharefsd` が存在しないか、ファイルが最近変更されていなかった場合は、ファイル `/etc/opt/SUNWsamfs/defaults.conf` を編集して `sam-sharefsd` のトレースを有効にする行を追加します。**

- a. `/etc/opt/SUNWsamfs` に `defaults.conf` がない場合は、`/opt/SUNWsamfs/examples/defaults.conf` からサンプルの `defaults.conf` ファイルを `/etc/opt/SUNWsamfs` にコピーします。

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

b. vi(1) などのエディタを使用してファイル

/etc/opt/SUNWsamfs/defaults.conf を編集し、トレースを有効にする行を追加します。

コード例 A-16 に、defaults.conf ファイルに追加する行を示します。

コード例 A-16 defaults.conf でトレースを有効にする行

```
trace
sam-sharefsd = on
sam-sharefsd.options = all
endtrace
```

c. samd(1M) config コマンドを実行して sam-fsd(1M) デーモンを再構成し、新しい defaults.conf ファイルを認識させます。

例:

```
# samd config
```

d. sam-fsd(1M) コマンドを実行して、構成ファイルを確認します。

コード例 A-17 に、sam-fsd(1M) コマンドの出力を示します。

コード例 A-17 sam-fsd(1M) コマンドの出力

```
# sam-fsd
Trace file controls:
sam-archiverd off
sam-catserverd off
sam-fsd         off
sam-rftd        off
sam-recycler    off
sam-sharefsd    /var/opt/SUNWsamfs/trace/sam-sharefsd
                 cust err fatal misc proc date
                 size  0    age 0
sam-stagerd     off
Would stop sam-archiverd()
Would stop sam-rftd()
Would stop sam-stagealld()
Would stop sam-stagerd()
Would stop sam-initd()
```

- e. /var/opt/SUNWsamfs/trace/sam-sharefsd のログファイルで、エラーを調べます。

```
# more /var/opt/SUNWsamfs/trace/sam-sharefsd
```

3. トレースファイルの最後の数十行で診断情報を調べます。

コード例 A-18 に、代表的な sam-sharefsd クライアントログファイルを示します。この例で、サーバーは titan、クライアントは dione です。このファイルには、パッケージのインストール後に生成される通常のログエントリが含まれ、最後ではデーモンがマウントされたファイルシステムに対して正常に動作しています。

コード例 A-18 クライアントのトレースファイル

```
dione# tail -18 /var/opt/SUNWsamfs/trace/sam-sharefsd
2004-03-23 16:13:11 shf-shsam2[13835:1]: FS shsam2: Shared file system daemon
started - config only
2004-03-23 16:13:11 shf-shsam2[13835:1]: FS shsam2: Host dione
2004-03-23 16:13:11 shf-shsam2[13835:1]: FS shsam2: Filesystem isn't mounted
2004-03-23 16:13:11 shf-shsam2[13837:1]: FS shsam2: Shared file system daemon
started
2004-03-23 16:13:11 shf-shsam2[13837:1]: FS shsam2: Host dione
2004-03-23 16:13:11 shf-shsam2[13837:1]: FS shsam2: Filesystem isn't mounted
2004-03-23 16:13:11 shf-shsam2[13837:1]: FS shsam2: Kill sam-sharefsd pid 13835
2004-03-23 16:13:12 shf-shsam2[13837:1]: FS shsam2: Killed sam-sharefsd pid
13835
2004-03-23 16:13:12 shf-shsam2[13837:1]: FS shsam2: Host dione; server = titan
2004-03-23 16:13:12 shf-shsam2[13837:1]: FS shsam2: Wakened from AWAIT_WAKEUP
2004-03-23 16:13:14 shf-shsam2[13837:5]: FS shsam2: Set Client (Server titan/3).
2004-03-23 16:13:14 shf-shsam2[13837:5]: FS shsam2: SetClientSocket dione
(flags=0)
2004-03-23 16:13:14 shf-shsam2[13837:5]: FS shsam2: rdsock dione/0 (buf=6c000).
2004-03-23 16:13:15 shf-shsam2[13837:1]: FS shsam2: Signal 1 received: Hangup
2004-03-23 16:13:15 shf-shsam2[13837:1]: FS shsam2: Wakened from AWAIT_WAKEUP
2004-03-23 16:13:15 shf-shsam2[13837:1]: FS shsam2: mount; flags=18889
2004-03-23 16:18:55 shf-shsam2[13837:1]: FS shsam2: Signal 1 received: Hangup
2004-03-23 16:18:55 shf-shsam2[13837:1]: FS shsam2: Wakened from AWAIT_WAKEUP
```

Linux クライアントのトラブルシューティング

Linux クライアントおよび Solaris クライアントは異なる手順を使用してシステム情報を探し、Sun StorageTek QFS の問題を診断します。

Linux カーネルのシステム情報が含まれたファイルは /proc ファイルシステムにあります。たとえば、/proc/cpuinfo ファイルにはハードウェア情報が含まれています。表 A-2 に、有用なトラブルシューティング情報が含まれたファイルをいくつか示します。

表 A-2 /proc のファイル

ファイル名	提供される情報
version	実行中のカーネルのバージョン
cpuinfo	ハードウェア情報
uptime	起動時からの時間、およびプロセスで使用されている合計時間 (秒数)
modules	読み込まれているモジュールに関する情報
cmdline	起動時にカーネルに渡されたコマンド行パラメータ
filesystems	存在するファイルシステムの実装
scsi/scsi	接続されている SCSI デバイス
fs/samfs/<QFS file system>/fsid	NFS の共有オプションに含まれている必要のあるファイルシステム ID

Linux のカーネルログメッセージは /var/log/messages ファイルに格納されます。

トラブルシューティングツール

Linux カーネルには多数のバリエーションがあるため、問題のトラブルシューティングは非常に困難です。デバッグに役立つツールがいくつかあります。

- lkcd および kgdb などのプロジェクトからは、カーネルクラッシュのダンプ情報が提供されます。
- kdb、kgdb、icebox などのプロジェクトからは、カーネルデバッグが提供されます。

注 – このようなプロジェクトは、Red Hat Linux または SUSE 用にはデフォルトで存在していません。RPM または SRPM を入手し、カーネルを再構成して使用する必要があります。

- strace コマンドは、システムコールおよび信号をトレースします。Solaris の truss コマンドに似たものです。
- Sun StorageTek QFS samtrace コマンドは、トレースバッファをダンプします。

- Sun StorageTek QFS samexplorer コマンドは、Sun StorageTek QFS 診断レポートスクリプトを生成します。

注 – Linux クライアントのトレースファイルは、Solaris クライアントと同じ `/var/opt/SUNWsamfs/trace` ディレクトリにあります。

よくある質問

Solaris プラットフォームの Sun StorageTek QFS を使い慣れたユーザーからよく寄せられる、Linux クライアントに関する質問を次に示します。

質問: Linux のインストールスクリプトから、負のスコアを受け取ったためにソフトウェアをインストールできないとレポートされます。ソフトウェアをインストールする方法はありますか。

回答: `-force-custom` および `-force-build` インストールオプションを試してください。ただしこの結果、モジュールのインストール時にシステムパニックが発生することがあります。カーネルが `spinlock` デバッグなどのカーネルハッキングオプション付きで構築されていると、特に危険性が高まります。

質問: `vmstat`、`iostat`、`top`、`truss` などのコマンドは Linux で使用できますか。

回答: `vmstat`、`top`、`iostat` の各コマンドは多くの Linux 環境にあります。インストールされていない場合は、`sysstat` および `procps` の RPM パッケージを使用して追加できます。Linux の `truss` 同等のコマンドは、`ltrace` および `strace` です。

質問: Sun StorageTek Traffic Manager は Sun StorageTek QFS の Linux クライアントで使用できますか。

回答: はい。Sun StorageTek Traffic Manager のマニュアルで説明されているように、マルチパスサポート付きでカスタムカーネルを構築しておきます。そのあと、Linux クライアントソフトウェアをインストールします。

質問: Sun StorageTek QFS の Linux クライアントで EFI (Extensible Firmware Interface) ラベルは使用できますか。

回答: 多くの Linux カーネルは、GPT (GUID Partition Table) パーティションの EFI ラベルのサポート付きで構築されていません。そのため、EFI ラベルを使用するには、`CONFIG_EFI_PARTITION` オプションセット付きでカーネルを再構築する必要があります。カスタムカーネルの構築方法についての詳細は、各ディストリビューションのマニュアルを参照してください。

質問: LVM (Logical Volume Management)、EVMS (Enterprise Volume Management System)、Device Mapper など、ほかの Linux のボリュームマネージャーと Sun StorageTek QFS Linux クライアントソフトウェアを使用できますか。

回答: ファイルシステムを EVMS で使用するには、そのファイルシステム用の FSIM (File System Interface Module) が必要です。Sun StorageTek QFS 製品用の FSIM は存在しません。LVM を使用するには、fdisk で表示されるパーティションの種類が LVM(8e) でなければなりません。Sun StorageTek QFS が使用するパーティションは SunOS にしてください。

質問: 2T バイトよりも大きいファイルシステムを使用できますか。

回答: はい。ただし、df などのファイルシステム情報を提供するユーティリティーの中には、Linux で実行したときに誤った情報を返すものがあります。また、NFS または Samba でファイルシステムを共有したときに問題が発生することがあります。

質問: Linux クライアントがサポートするマウントオプションと、Solaris クライアントがサポートするマウントオプションに違いがありますか。

回答: Linux クライアントでサポートされていない samfs マウントオプションがいくつかもあります。注意が必要なのは、nosuid と forcedirectio です。Linux クライアントでサポートされているマウントオプションの全リストについては、『Sun StorageTek QFS Linux クライアントガイド』を参照してください。

注 - mdadm (multiple devices admin) パッケージは、Sun StorageTek QFS Linux クライアントでのパスのフェイルオーバーに使用しないようにしてください。mdadm パッケージは、使用する装置にスーパーブロックを書き込みます。そのため、mdadm は Solaris が装置に書き込んだデータを壊す可能性があります。さらに、Solaris も、mdadm が装置に書き込んだスーパーブロックを壊す可能性があります。

Sun StorageTek QFS 共有ファイルシステムのマウントオプション

Sun StorageTek QFS 共有ファイルシステムをマウントするために、いくつかのマウントオプションを使用できます。この付録では、いくつかのオプションについて、その役割ごとに説明しています。

ほとんどのマウントオプションは、`mount(1M)` コマンドを使用したり、`/etc/vfstab` ファイルに記述したり、`samfs.cmd` ファイルに記述したりすることで指定できます。たとえば、次の `/etc/vfstab` ファイルには、Sun StorageTek QFS 共有ファイルシステムのための `mount(1M)` オプションが含まれています。

```
sharefs1 - /sfs samfs - no shared,mh_write
```

`samu(1M)` オペレータユーティリティを使用すると、一部のマウントオプションを動的に変更できます。これらのオプションの詳細については、261 ページの「`samu(1M)` オペレータユーティリティの使用法」を参照してください。

このあとの項に、Sun StorageTek QFS 共有ファイルシステムで使用可能なマウントオプションをまとめています。これらのマウントオプションの詳細は、`mount_samfs(1M)` のマニュアルページ、または各説明で示されている参照先をご覧ください。

- 254 ページの「バックグラウンドでのマウント: `bg` オプション」
- 254 ページの「ファイルシステムマウントの再試行: `retry` オプション」
- 254 ページの「Sun StorageTek QFS 共有ファイルシステムの宣言: `shared` オプション」
- 255 ページの「割り当てサイズの調整: `minallopsz=n` および `maxallopsz=n` オプション」
- 255 ページの「Sun StorageTek QFS 共有ファイルシステムでのリースの使用: `rdlease=n`、`wrlease=n`、および `aplease=n` オプション」
- 256 ページの「複数ホストの読み取りと書き込みの有効化: `mh_write` オプション」

- 258 ページの「並行スレッドの最小数の設定: min_pool=n オプション」
- 259 ページの「キャッシュした属性の保持: meta_timeo=n オプション」
- 259 ページの「ストライプ化割り当ての指定: stripe オプション」
- 259 ページの「メタデータの書き込み頻度の指定: sync_meta=n オプション」
- 260 ページの「WORM 機能の有効化: worm_capable および def_retention オプション」

バックグラウンドでのマウント: bg オプション

bg マウントオプションを指定すると、最初のマウント操作が失敗した場合に、そのあとのマウント試行がバックグラウンドで行われます。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。



注意 – Sun StorageTek QFS 共有ファイルシステムを Sun Cluster ノードにマウントする場合、またはクラスタの外部に共有 Sun StorageTek QFS クライアントがある場合は、このオプションを使用しないでください。

ファイルシステムマウントの再試行: retry オプション

retry マウントオプションを使用して、システムがファイルシステムのマウントを再試行する回数を指定します。デフォルトは 10000 です。

Sun StorageTek QFS 共有ファイルシステムの宣言: shared オプション

shared マウントオプションを使用して、Sun StorageTek QFS 共有ファイルシステムにするファイルシステムを宣言します。ファイルシステムを Sun StorageTek QFS 共有ファイルシステムとしてマウントするには、このオプションを /etc/vfstab

ファイルに指定する必要があります。このオプションを `samfs.cmd` ファイルまたは `mount(1M)` コマンドに指定しても、エラー条件は発生しませんが、ファイルシステムは Sun StorageTek QFS 共有ファイルシステムとしてマウントされません。

割り当てサイズの調整: `minallocsz=n` および `maxallocsz=n` オプション

`mount(1M)` コマンドの `-o minallocsz=n` および `-o maxallocsz=n` オプションは、容量をキロバイト単位で指定します。これが最小のブロック割り当てサイズになります。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。`-o minallocsz=n` オプションを使用すると、この割り当ての初期サイズを指定できます。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基づいて、`-o maxallocsz=n` オプションの設定値までブロック割り当てのサイズを増やすことができます (設定値を超えることはできません)。

これらの `mount(1M)` のオプションは、`mount(1M)` のコマンド行、`/etc/vfstab` ファイル、または `samfs.cmd` ファイルで指定できます。

Sun StorageTek QFS 共有ファイルシステムでのリースの使用: `rdlease=n`、`wrlease=n`、および `aplease=n` オプション

「リース」は、その有効期間に応じて、ファイルに対して操作を実行する共有ホスト権を認めます。メタデータサーバーは、自らを含む各共有ホストに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。対象となるファイル操作は次のとおりです。

- 「読み取りリース」は、既存のファイルデータの読み取りを可能にする。
- 「書き込みリース」は、既存のファイルデータの上書きを可能にする。
- 「追加リース」は、ファイルサイズの拡張を可能にし、新しく割り当てられたブロックへの書き込みを可能にする。

共有ホストは、必要であれば何度でもリースの更新を継続できます。リースはエンドユーザーから透過的に行われます。表 B-1 に、各リースタイプの期間を指定するマウントオプションを示します。

表 B-1 リース関連の mount(1M) のオプション

オプション	動作
-o rdlease= <i>n</i>	このオプションでは、読み取りリースの最長時間 (秒単位) を指定します。
-o wrlease= <i>n</i>	このオプションでは、書き込みリースの最長時間 (秒単位) を指定します。
-o aplease= <i>n</i>	このオプションでは、追加リースの最長時間 (秒単位) を指定します。

3 つのリースすべてにおいて、*n* は $15 \leq n \leq 600$ の範囲で指定できます。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。これらのリース設定の詳細は、mount_samfs(1M) 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、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。Sun Cluster 環境でのリース時間についての詳細は、『Sun StorageTek QFS インストールおよびアップグレードの手引き』を参照してください。

複数ホストの読み取りと書き込みの有効化: mh_write オプション

Sun StorageTek QFS 共有ファイルシステムでは、デフォルトで、複数のホストが同時に同じファイルを読み取ることができます。また、そのファイルにホストが書き込んでいない場合は、入出力はすべてのホストでページングされます。ファイルにデータを追加または書き込めるホストは、一度に 1 つだけです。

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のマウントオプションとして mh_write を指定すると、Sun StorageTek QFS 共有ファイルシステムによって、同じファイルへ

の複数ホストによる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、ファイルに書き込めるのは一度に1つのホストだけです。そのための時間は、`wrlease` マウントオプションの時間によって決まります。Sun StorageTek QFS 共有ファイルシステムが、`mh_write` オプションを有効にしてメタデータサーバーにマウントされている場合、同一ファイルへの読み取りと書き込みを同時に複数のホストから行うことができます。

表 B-2 では、複数ホストからのファイルアクセスが、メタデータサーバーでの `mh_write` オプションの有効/無効によってどのように影響されるかを示します。

表 B-2 `mh_write` オプションに基づくファイルアクセス

メタデータサーバーで <code>mh_write</code> が無効	メタデータサーバーで <code>mh_write</code> が有効
複数の読み取りホストが許可されます。ページ入出力を使用できます。	複数の読み取りホストが許可されます。ページ入出力を使用できます。
1つの書き込みホストだけが許可されます。ページ入出力を使用できます。そのほかのホストは待機します。	複数の読み取りおよび書き込みホストが許可されます。書き込みホストが存在する場合は、すべての入出力は直接行われます。
1つの追加ホストだけが許可されます。そのほかのホストは待機します。	1つの追加ホストだけが許可されます。そのほかのホストは読み取りまたは書き込みができます。書き込みホストが存在する場合は、すべての入出力は直接行われます。

`mh_write` オプションでは、ロック動作が変化しません。`mh_write` が有効と無効のどちらでも、ファイルロックの動作は同じです。`mh_write` オプションの効果は次のとおりです。

- `mh_write` が有効な場合、すべてのホストは1つのファイルを同時に読み取りおよび書き込みが可能です。
- `mh_write` が無効な場合、ある期間内にあるファイルに書き込めるホストは1つのみで、その期間内はどのホストもそのファイルを読み取ることができません。

Sun StorageTek QFS 共有ファイルシステムでは、ホスト間の整合性が維持されます。あるホストが読み取りまたは書き込みシステムコールを始めて実行したときに、そのホストはある期間内のファイルの読み取りまたは書き込みを可能にする「リース」を受け取ります。そのリースが存在することで、`mh_write` がない場合にほかのホストがそのファイルにアクセスすることが防止されます。特に、リースはその取得の元になったシステムコールよりも長い期間存在できます。

`mh_write` が無効な場合、Sun StorageTek QFS 共有ファイルシステムはデータの読み取りと書き込みに対して、POSIX に近い動作を行います。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。ファイルの変更は書き込みリースの最後に、読み取りリースが取得されたときにディスクへ反映され、新しく書き込まれたデータが表示されるよう、システムは無効なキャッシュページを破棄します。

`mh_write` が有効な場合、動作は一定でなくなります。読み取りと書き込みが同時にあった場合、Sun StorageTek QFS 共有ファイルシステムはすべてのホストのファイルアクセスを直接入出力モードに切り替えます。これにより、ページ整合入出力がすぐにほかのホストから確認されます。ただし、非ページ整合入出力によって無効なデータが表示されたり、ファイルに書き込まれることがあります。このような動作を防止している、通常のリースメカニズムが無効になるからです。

`mh_write` オプションを指定するのは、複数のホストが同時に同じファイルに書き込む必要があり、アプリケーションがページ整合入出力を実行しているときだけにしてください。それ以外の場合は、`flock()` (`mh_write` と組み合わせて動作) を使用してホスト間を調整しても、整合性は保証されないため、データの不整合が発生する危険性があります。

`mh_write` の詳細は、`mount_samfs(1M)` のマニュアルページを参照してください。

並行スレッドの最小数の設定: `min_pool=n` オプション

`min_pool=n` マウントオプションでは、Sun StorageTek QFS 共有ファイルシステムの並行スレッドの最小数が設定されます。デフォルトでは、Solaris システムで `min_pool=64`、Linux クライアントで 8 です。したがって、デフォルトの設定を使用した場合、Solaris 上のスレッドプールに少なくとも 64 個、Linux 上では 8 個のアクティブスレッドが常に存在します。`min_pool=n` マウントオプションは、Sun StorageTek QFS 共有ファイルシステムの動作状態に応じて、8 から 2048 まで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` ファイルで設定してください。`/etc/vfstab` ファイルまたはコマンド行で設定すると、無視されます。

注 - `min_pool` マウントオプションは、以前の `nstreams` マウントオプションに代わるものです。ソフトウェアの version 4U6 では、`nstreams` オプションは無視されます。これは、ソフトウェアの version 5 で完全に削除される予定です。

キャッシュした属性の保持: meta_timeo=*n* オプション

meta_timeo=*n* マウントオプションを使用して、メタデータ情報の検査の間にシステムが待機する時間を判別します。デフォルトでは、システムはメタデータ情報を3秒ごとに更新します。したがって、たとえば、新しくいくつかのファイルが作成された Sun StorageTek QFS 共有ファイルシステムに ls(1) コマンドを入力した場合、3秒経過しないとすべてのファイルの情報が戻されないことがあります。*n* には、 $0 \leq n \leq 60$ の値を指定します。

ストライプ化割り当ての指定: stripe オプション

デフォルトでは、Sun StorageTek QFS 共有ファイルシステム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で割り当てられます。ファイルデータがディスク間で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に対して stripe マウントオプションを指定します。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。このため、1ファイルの最大パフォーマンスは、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。割り当て方式の詳細については、7ページの「設計の基本」を参照してください。

メタデータの書き込み頻度の指定: sync_meta=*n* オプション

sync_meta=*n* オプションは sync_meta=1 または sync_meta=0 に設定できます。

デフォルトの sync_meta=1 では、Sun StorageTek QFS 共有ファイルシステムは、ファイルのメタデータが変更するごとにメタデータをディスクに書き込みます。この場合、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。この設定は、メタデータサーバーを変更するときに有効になっている必要があります。

sync_meta=0 と設定すると、Sun StorageTek QFS 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。このように書き込みを遅らせると、高いパフォーマンスが実現しますが、予期せずにマシンが停止したときのデータ整合性が低下します。

WORM 機能の有効化: worm_capable および def_retention オプション

オプションの WORM パッケージを使用している場合、ファイルシステムは worm_capable マウントオプションによって WORM ファイルをサポートできません。def_retention マウントオプションは、フォーマット def_retention=*MyNdOhPm* で、デフォルトの保存時間を設定します。

この形式では、M、N、O、P は負以外の整数です。また、y、d、h、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。

WORM 機能についての詳細は、194 ページの「WORM-FS ファイルシステムの構成」を参照してください。

samu(1M) オペレータユーティリティーの使用法

この章では、Sun StorageTek QFS 環境で構成した装置を samu(1M) で制御する方法について説明します。samu(1M) の表示の多くは、ストレージ/アーカイブ管理メカニズムを使用するサイトでのみ有益です。Sun StorageTek QFS 環境で samu(1M) を使用した場合、一部の表示は適用されません。

この章の内容は次のとおりです。

- 261 ページの「概要」
- 264 ページの「オペレータ表示」
- 314 ページの「オペレータコマンド」

samu(1M) の中から行える操作は、samcmd(1M) コマンドを使用して行うこともできます。samcmd(1M) については、samcmd(1M) のマニュアルページを参照してください。

概要

samu(1M) オペレータユーティリティーを使用するには、少なくとも横 80 文字、縦 24 行を表示する端末が必要です。ユーティリティーは、次の機能を備えています。

- Sun StorageTek QFS および SAM-QFS の装置とファイルシステムの活動を監視するための表示。
- 表示の選択、表示のオプションの設定、装置へのアクセスと装置の稼働状況の制御、およびウィンドウ表示のスナップショットの取得のためのコマンド。
- Sun StorageTek QFS または SAM-QFS のファイルシステムの実行を調整するコマンド。

この章で紹介するウィンドウ表示は、代表的な例です。端末に表示される情報の具体的な形式と量は、端末の種類と Sun StorageTek QFS 環境に構成されている装置によって異なる可能性があります。

このあとの項目では、samu(1M) の起動と停止、ユーティリティーとのやりとり、ヘルプウィンドウへのアクセス、およびオペレータ表示を行う方法について説明します。

▼ samu(1M) を起動する

- samu(1M) を起動するには、コマンド行で samu(1M) コマンドを入力します。

```
# samu
```

samu(1M) が起動され、ヘルプ画面が表示されます。これが、デフォルトの初期画面です。別の samu(1M) 画面を表示するには、262 ページの「samu(1M) 画面を表示する」の手順に従います。

samu(1M) ユーティリティーの初期画面は選択できます。samu(1M) のコマンド行オプションについては、samu(1M) のマニュアルページを参照してください。

注 - vi(1) 同様、samu(1M) は curses(3CURSES) ルーチンをベースとしています。samu(1M) を起動できない場合は、端末タイプが正しく定義されているかどうかを確認してください。

▼ samu(1M) 画面を表示する

samu(1M) コマンドには、別の samu(1M) 画面を表示するコマンド行オプションがあります。

1. コロン (:) を入力すると、samu(1M) プロンプトが表示されます。
コロンを入力すると、左下に次のプロンプトが表示されます。

```
Command:
```

2. 表示する画面に対応する文字を入力して、復帰改行を押します。

たとえば、v 画面を表示する場合は、Command: プロンプトの後ろに **v** を入力してから復帰改行を押します。

入力する文字とそれに対応する画面のリストについては、276 ページの「(h) - ヘルプ表示」を参照してください。

▼ samu(1M) を停止する

- samu(1M) を終了するには、次のどちらかを入力します。

- q
- :q

samu(1M) オペレータユーティリティが終了し、コマンドシェルに戻ります。

samu(1M) とのやりとり

samu(1M) とのやりとりは、順方向と逆方向のページ送り、コマンドの入力、画面の再表示、およびユーティリティの終了という点では、UNIX の vi(1) エディタとのやりとりと似ています。

この章には、各画面の項があります。各画面の項で、その画面のナビゲートに使用できるコントロールキーシーケンスを示します。samu(1M) のマニュアルページには、コントロールキーナビゲーションシーケンスの要約があります。

ウィンドウ表示の最終行には、コマンドエラーメッセージと表示エラーメッセージが表示されます。コマンドエラーが発生した場合、自動再表示機能は、オペレータが次の処理を行うまで停止します。

装置の入力

Sun StorageTek QFS 環境に含まれる装置には、mcf ファイルで装置番号 (10 など) が割り当てられます。多数の samu(1M) コマンドが装置番号を使用して 1 つの装置を参照します。

例:「:off」コマンドの構文は、次のとおりです。

```
:off eq
```

eq には、対象装置の装置番号を入力します。

オンラインヘルプの表示

samu(1M) を起動すると、最初のヘルプ画面をシステムが自動的に表示します。

ヘルプ (h) 画面の詳細については、276 ページの「(h) - ヘルプ表示」を参照してください。

▼ 表示画面からオンラインヘルプにアクセスする

- :h を入力します。

ある画面から次の画面に、順方向または逆方向に移動するには、次のキーシーケンスを入力します。

- 順方向にページ送りするには **Ctrl-f** を押す。
- 逆方向にページ送りするには **Ctrl-b** を押す。

h キーを押すことにより、いつでもヘルプ画面に戻ることができます。

オペレータ表示

samu(1M) オペレータ表示にするには、各表示に対応するキーを押します。a から w までの小文字のキーは、動作情報を表示します。

注 - 大文字の samu(1M) 表示 (C、D、F、I、J、K、L、M、N、P、R、S、T、U) は、技術サポートスタッフが支援するカスタマーサイトでのみ使用するよう設計されています。

この章では、大文字の表示について、小文字の表示ほど詳しくは説明しません。

表示が画面領域からはみ出る場合は、画面表示の下に単語 **more** が表示され、追加情報があることを示します。Ctrl-f を使用すると、次のページへ進んで残りの内容を表示できます。

コード例 C-1 では単語 **more** が表示され、後続の画面に情報があることを示します。

コード例 C-1 追加テキストがあることを示す samu(1M) の表示

```
xb54 54  exb8505  pt03  0  yes  2  0  on
lt55 55  dlt2000  pt02  1  yes  4  0  on  ml65
hp56 56  hpc1716  pt01  1  yes  3  0  on  hp70
hp57 57  hpc1716  pt01  1  yes  4  0  on  hp70
more
```

samu(1M) が装置の入力を指示した場合には、装置の装置番号を入力します。構成画面 (c) で、すべてのリムーバブルメディア装置の装置番号が表示されます。すべての表示の制御に、その表示用にリストされたコントロールキーを使用します。

このあとの項目では、オペレータ表示についてアルファベット順に説明します。例を示し、必要に応じて、表示されるフィールドについて説明する表を示します。

注 – Sun StorageTek SAM ソフトウェアなしで Sun StorageTek QFS ソフトウェアをインストールした場合は、一部の operator だけを表示できます。

(a) - アーカイバの状態表示

a 表示は、アーカイバの状態を表示します。

この表示は、次のように、必要な情報に応じて異なる方法で表示できます。

- アーカイバの状態の概要 (ファイルシステムごとのアーカイバの状態) を表示するには、次の形式でコマンドを入力します。

```
Command:a
```

- 特定のファイルシステムのアーカイブの詳細を表示するには、次の形式でコマンドを入力します。

```
Command:a filesystem
```

filesystem には、ファイルシステムの名前を指定します。

ナビゲーション

表 C-1 に、a 表示で使用するコントロールキーを示します。

表 C-1 a 表示のコントロールキー

キー	機能
Ctrl-b	直前のファイルシステム
Ctrl-f	次のファイルシステム
Ctrl-d	順方向に <i>arcopies</i> をページ送りする (下部)
Ctrl-u	逆方向に <i>arcopies</i> をページ送りする (下部)

表 C-2 に、`:a filesystem` 表示で使用するコントロールキーを示します。

表 C-2 `:a filesystem` 表示のコントロールキー

キー	機能
Ctrl-b	直前のファイルシステム
Ctrl-f	次のファイルシステム

表示例

コード例 C-2 は、単一ファイルシステムの活動と統計情報を概要表示で示します。

コード例 C-2 `samu(1M) a` の表示

```
Archiver status                samu 4.6 07:44:02 August 8 2005
sam-archiverd: Waiting for resources
sam-arfind: samfs1 mounted at /sam1
Waiting until 2005-05-08 07:54:02 to scan .inodes
sam-arfind: samfs2 mounted at /sam2
Waiting until 2005-05-08 07:52:57 to scan .inodes
sam-arfind: qfs1 mounted at /qfs1
Waiting until 2005-05-08 07:44:33 to scan .inodes
sam-arfind: qfs2 mounted at /qfs2
Waiting until 2005-05-08 07:53:21 to scan .inodes
sam-arfind: qfs3 mounted at /qfs3
Waiting until 2005-05-08 07:44:11 to scan .inodes

sam-arfind: qfs4 mounted at /qfs4
Waiting until 2005-05-08 07:53:35 to scan .inodes

sam-arfind: shareqfs1 mounted at /shareqfs1
Shared file system client. Cannot archive.

sam-arfind: shareqfs2 mounted at /shareqfs2
Shared file system client. Cannot archive.

sam-arcopy: qfs4.arset5.1.83 dt.DAT001
Waiting for volume dt.DAT001
```

フィールドの説明

表 C-3 に、詳細表示のフィールドを示します。

表 C-3 samu(1M) a の表示フィールドの説明

フィールド	説明
samfs1 mounted at	マウントポイント。
regular files	通常ファイルの数と全体サイズ。
offline files	オフラインファイルの数と全体サイズ。
archdone files	archdone ファイルの数とサイズ。アーカイバが処理を終了し、archdone ファイルに関する処理は残っていないことを意味します。archdone にマークされたファイルは、アーカイブ処理は終了しているが、必ずしもアーカイブ済みではありません。
copy1	アーカイブコピー 1 のファイル数と全体サイズ。
copy2	アーカイブコピー 2 のファイル数と全体サイズ。
copy3	アーカイブコピー 3 のファイル数と全体サイズ。
copy4	アーカイブコピー 4 のファイル数と全体サイズ。
directory	ディレクトリの数と全体サイズ。
sleeping until	アーカイバが次にいつ実行されるかを示します。

(c) - デバイス構成表示

c 表示には、構成の接続状況が表示されます。すべての装置名と装置番号が一覧表示されます。

デバイス構成表示を呼び出すには、次の形式でコマンドを入力します。

```
Command:c
```

ナビゲーション

表 C-4 に、この表示で使用するコントロールキーを示します。

表 C-4 c 表示のコントロールキー

キー	機能
Ctrl-b	逆方向にページ送りする
Ctrl-d	順方向に半ページ送る
Ctrl-f	順方向にページ送りする
Ctrl-u	逆方向に半ページ送る

表示例

コード例 C-3 は、デバイス構成表示の例です。

コード例 C-3 samu(1M) c の表示

```
Device configuration:          samu    4.6 07:48:11 Sept 8 2006
ty  eq state  device_name                fs family_set
sk  100 on    /etc/opt/SUNWsamfs/dcstkconf 100 dcL700
tp  120 off   /dev/rmt/1cbn                100 dcL700
sg  130 on    /dev/rmt/4cbn                100 dcL700
sg  140 on    /dev/rmt/5cbn                100 dcL700
tp  150 off   /dev/rmt/3cbn                100 dcL700
hy  151 on    historian                      151
```

フィールドの説明

表 C-5 で、この表示のフィールドを説明します。

表 C-5 samu(1M) c の表示フィールドの説明

フィールド	説明
ty	装置タイプ。
eq	装置番号。
state	装置の現在の動作状態。有効なデバイスの状態は、次のとおりです。 <ul style="list-style-type: none">• on - 装置をアクセスに利用可能。• ro - 装置は、読み込み専用アクセスだけに利用可能。• off - 装置をアクセスに利用できません。• down - 装置は、メンテナンスアクセスだけに利用可能。• idle - 装置は、新しい接続に利用できません。進行中の操作は、終了するまで続行されます。• nalloc - この装置への以後の割り当てを禁止する nalloc フラグが設定されています。詳細は、29 ページの「論理ユニット番号 (LUN) による割り当て制御」を参照してください。
device_name	装置のパス。
fs	ファミリーセットの装置番号。
family_set	装置が属するストレージのファミリーセットまたはライブラリ名。

(C) - メモリー表示

C 表示は、指定したメモリーアドレスの内容を表示します。アドレスの内容を表示するには、アドレスを 16 進数で入力します。

この表示を呼び出すには、次のコマンドを入力します。

```
Command: C hex-address
```

hex-address には、メモリー位置のアドレスを 16 進数で指定します。

例:

```
Command: C 0x1044a998
```

この表示はデバッグ用です。Sun のサポート要員の支援があるときにのみ使用してください。

表示例

コード例 C-4 は、メモリー表示の例です。この出力は長いため、マニュアルでは途中から省略しています。

コード例 C-4 samu(1M) C の表示

```
Memory base: 0x1234567 samu 4.6 07:52:25 Sept 8 2006
00000000 80921000 137ff801 edd05e21 40853420 .....x.mP^!@.4
00000010 00a00420 018528b0 01a604e0 02840640 . . .(0.&.'...@
00000020 02d030a1 a0853420 0080a0a0 100a6fff .P0! .4 . . .o.
00000030 f6921000 13c65e23 582d0000 0ba01020 v....F^#X-... .
00000040 00c45e20 48c608e0 2fd05e21 40920080 .D^ HF.'/P^!@...
00000050 037ff801 fa941000 16c45e20 48a600a0 ..x.z....D^ H&.
00000060 80921000 137ff801 d5d05e21 40853420 .....x.UP^!@.4
00000070 00a00420 018528b0 01a604e0 02840640 . . .(0.&.'...@
00000080 02d030a1 c0853420 0080a0a0 100a6fff .P0!@.4 . . .o.
00000090 f6921000 13c65e23 58a01020 00c45e20 v....F^#X . .D^
000000a0 48c608e0 2fd05e21 40920080 037ff801 HF.'/P^!@.....x.
000000b0 e39405a2 00c45e20 48a600a0 80921000 c..".D^ H&. ....
000000c0 137ff801 bed05e21 40853420 00a00420 ..x.>P^!@.4 . .
000000d0 018528b0 01a604e0 02840640 02d030a1 ..(0.&.'...@.P0!
000000e0 e0853420 0080a0a0 100a6fff f6921000 `.4 . . .o.v...
000000f0 13c65e23 58a01020 00c45e20 48c608e0 .F^#X . .D^ HF.'`
```

(d) - デーモントレースコントロールの表示

d 表示には、defaults.conf ファイルに指定されているとおりトレースされているイベントが表示されます。トレースファイルを使用可能にする方法については、defaults.conf(4) のマニュアルページを参照してください。

この表示を呼び出すには、次のコマンドを入力します。

```
Command:d
```

表示例

コード例 C-5 は、トレースファイル情報の表示例です。トレース対象のデーモンに関する情報、トレースファイルのパス、トレース対象イベント、トレースファイルのサイズと経過時間に関する情報などが表示されます。

コード例 C-5 samu(1M) d の表示

```
Daemon trace controls          samu 4.6 07:56:38 Sept 8 2006
sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
              cust err fatal misc proc debug date
              size  0      age  0
sam-archiverd /var/opt/SUNWsamfs/trace/sam-archiverd
              cust err fatal misc proc debug date
              size  0      age  0
sam-catserverd /var/opt/SUNWsamfs/trace/sam-catserverd
              cust err fatal misc proc debug date
              size  0      age  0
sam-fsd       /var/opt/SUNWsamfs/trace/sam-fsd
              cust err fatal misc proc debug date
              size  0      age  0
sam-rftd      /var/opt/SUNWsamfs/trace/sam-rftd
              cust err fatal misc proc debug date
              size  0      age  0
sam-recycler  /var/opt/SUNWsamfs/trace/sam-recycler
              cust err fatal misc proc debug date
              size  0      age  0
sam-sharefsd  /var/opt/SUNWsamfs/trace/sam-sharefsd
              cust err fatal misc proc debug date
              size  0      age  0
sam-stagerd   /var/opt/SUNWsamfs/trace/sam-stagerd
              cust err fatal misc proc debug date
              size  0      age  0
sam-serverd   /var/opt/SUNWsamfs/trace/sam-serverd
              cust err fatal misc proc debug date
              size  0      age  0
sam-clientd   /var/opt/SUNWsamfs/trace/sam-clientd
              cust err fatal misc proc debug date
              size  0      age  0
sam-mgmt      /var/opt/SUNWsamfs/trace/sam-mgmt
              cust err fatal misc proc debug date
              size  0      age  0
```

(D) - ディスクボリュームディクショナリ

D 表示は、`diskvols.conf` ファイルで定義されたディスクアーカイブ用のディスクメディアを追跡する、ディスクボリュームディクショナリを表示します。ディクショナリには、容量、空き容量、VSN の状態を示すフラグなど、各 VSN に関する情報が格納されています。フラグとしては、*unavailable*、*read only*、*bad media* などがあります。

この表示を呼び出すには、次のコマンドを入力します。

```
Command: D
```

表示例

コード例 C-6 は、デバイス構成表示の例です。

コード例 C-6 `samu(1M) D` の表示

```
Disk volume dictionary samu      4.6 07:48:11 May 8 2007

header
version 460

volumes
magic 340322 version 9 nkeys 2 ndata 2
index spacecapacityusedflagsvolume
   0 128824115201288778547210291200-----disk01
   1  6443827200 6443892736   70656-----disk02
clients
magic 340322 version 9 nkeys 1 ndata 1
```

フラグ

表 C-6 で、D 表示のフラグを示します。

表 C-6 samu(1M) D 表示の flags フィールド

フィールド	説明
l----	ボリュームにラベルが付けられ、seqnum ファイルが作成されました。ソフトウェアが新しい seqnum ファイルを作成することを防止するため、管理者が設定します。
-r---	ボリュームがリモートホストで定義されています。
--U--	ボリュームは利用不可です。
---R-	ボリュームは読み取り専用です。
----E	メディアエラー。ソフトウェアがディスクアーカイブディレクトリで書き込みエラーを検出したときに設定されます。

diskvols samu(1M) コマンドを使用すると、ディスクボリュームのディクショナリフラグを設定またはクリアできます。326 ページの「:diskvols volume [+flag | -flag] コマンド」を参照してください。

(f) - ファイルシステムの表示

f 表示は、Sun StorageTek QFS ファイルシステムのコンポーネントを表示します。

この表示を呼び出すには、次のコマンドを入力します。

```
Command: f
```

表示例

コード例 C-7 は、ファイルシステム表示の例です。

コード例 C-7 samu(1M) f の表示

```
File systems                                     samu 4.6 08:11:24 Sept 8 2006

ty      eq state      device_name      status high low mountpoint server
ms      10   on                samfs1   m----2----d 90% 70% /sam1
md      11   on      /dev/dsk/c5t8d0s3
md      12   on      /dev/dsk/c5t8d0s4
```

コード例 C-7 samu(1M) f の表示 (続き)

md	13	on	/dev/dsk/c5t8d0s5				
md	14	on	/dev/dsk/c5t8d0s6				
md	15	on	/dev/dsk/c5t8d0s7				
ms	20	on	samfs2	m----	2----	d 90% 70%	/sam2
md	21	on	/dev/dsk/c5t9d0s3				
md	22	on	/dev/dsk/c5t9d0s4				
md	23	on	/dev/dsk/c5t9d0s5				
md	24	on	/dev/dsk/c5t9d0s6				
md	25	on	/dev/dsk/c5t9d0s7				
ma	30	on	qfs1	m----	2----	d 90% 70%	/qfs1
mm	31	on	/dev/dsk/c5t10d0s0				
md	32	on	/dev/dsk/c5t10d0s1				
ma	40	on	qfs2	m----	2----	d 90% 70%	/qfs2
mm	41	on	/dev/dsk/c5t11d0s0				
md	42	on	/dev/dsk/c5t11d0s1				
ma	50	on	qfs3	m----	2---	r- 90% 70%	/qfs3
mm	51	on	/dev/dsk/c5t12d0s0				
mr	52	on	/dev/dsk/c5t12d0s1				
ma	60	on	qfs4	m----	2---	r- 90% 70%	/qfs4
mm	61	on	/dev/dsk/c5t13d0s0				
mr	62	on	/dev/dsk/c5t13d0s1				
ma	100	on	shareqfs1	m----	2c--	r- 80% 70%	/shareqfs1 spade
mm	101	on	/dev/dsk/c6t50020F2300004655d0s0				
mr	102	on	/dev/dsk/c6t50020F2300004655d0s1				
ma	110	on	shareqfs2	m----	2c--	r- 80% 70%	/shareqfs2 spade
mm	111	on	/dev/dsk/c6t50020F2300004655d0s6				
mr	112	on	/dev/dsk/c6t50020F2300004655d0s7				

フィールドの説明

表 C-7 で、この表示のフィールドを説明します。

表 C-7 samu(1M) f の表示フィールドの説明

フィールド	説明
ty	装置タイプ。
eq	装置番号。
state	装置の現在の動作状態。有効なデバイスの状態は、次のとおりです。 <ul style="list-style-type: none">• on - 装置をアクセスに利用可能。• ro - 装置は、読み込み専用アクセスだけに利用可能。• off - 装置をアクセスに利用できません。• down - 装置は、メンテナンスアクセスだけに利用可能。• idle - 装置は、新しい操作に利用できません。進行中の操作は、終了するまで続行されます。• nalloc - この装置への以後の割り当てを禁止する nalloc フラグが設定されています。詳細は、29 ページの「論理ユニット番号 (LUN) による割り当て制御」を参照してください。
device_name	ファイルシステム名または装置のパス。
status	デバイスの状態。状態コードの説明については、309 ページの「オペレータ表示の状態コード」を参照してください。
high	ディスク使用量の上限率。
low	ディスク使用量の下限率。
mountpoint	ファイルシステムのマウントポイント。
server	ファイルシステムがマウントされているホストシステム名。

(F) - 光磁気ディスクラベル表示

F 表示は、光磁気ディスクのラベルを表示します。

この表示を呼び出すには、次のコマンドを入力します。

```
Command: F
```

この表示はデバッグ用です。Sun のサポート要員の支援があるときにのみ使用してください。

(h) - ヘルプ表示

h 表示は、使用可能な samu(1M) の表示の概要を示します。デフォルトの場合、コマンド行で samu(1M) コマンドを入力したときに、システムが最初に表示する画面です。

この表示を呼び出すには、次のコマンドを入力します。

```
Command:h
```

ナビゲーション

表 C-8 に、この表示で使用するコントロールキーを示します。

表 C-8 h 表示のコントロールキー

キー	機能
Ctrl-b	逆方向にページ送りする
Ctrl-d	順方向にページ送りする (上部)
Ctrl-f	順方向にページ送りする
Ctrl-u	逆方向にページ送りする (上部)
Ctrl-k	パス表示をトグルする

表示例

ヘルプ画面は複数ページありますが、このマニュアルでは 1 ページ目だけを紹介します。2 ページ目以降のヘルプ画面には、各種の samu(1M) コマンドが表示されます。

コード例 C-8 は、Sun StorageTek SAM ファイルシステムの最初のヘルプ画面を示しています。Sun StorageTek QFS ファイルシステムでは、最初のヘルプ画面に含まれない表示があります。たとえば、Sun StorageTek QFS システムを実行している場合、リムーバブルメディア表示は使用できません。SAM-QFS の構成を使用している場合、ヘルプ画面はコード例 C-8 のように表示されます。

コード例 C-8 Sun StorageTek SAM システムの場合の samu(1M) の初期ヘルプ画面

```
Help informationpage 1/15samu 4.6 08:18:13 Sept 8 2006
Displays:
  a  Archiver statuswPending stage queue
  c  Device configurationCMemory
  d  Daemon trace controlsDDisk volume dictionary
  f  File systemsFOptical disk label
```

コード例 C-8 Sun StorageTek SAM システムの場合の samu(1M) の初期ヘルプ画面 (続き)

```
h Help informationIInode
l Usage informationJPreview shared memory
m Mass storage statusKKernel statistics
n Staging statusLShared memory tables
o Optical disk statusMShared memory
p Removable media load requestsNFile system parameters
r Removable mediaPActive Services
s Device statusRSAM-Remote
t Tape drive statusSSector data
u Staging queueTSCSI sense data
v Robot catalogUDevice table

more (ctrl-f)
```

(I) - i ノード表示

I 表示は、i ノードの内容を表示します。

この表示は、次のように、必要な情報に応じて異なる方法で表示できます。

- ファイルシステム全体の i ノードを表示するには、次の形式でコマンドを入力します。

```
Command :I filesystem
```

filesystem には、ファイルシステムの名前を指定します。

- 特定の i ノードを表示するには、次の形式でコマンドを入力します。

```
Command:I inode-number
```

inode-number には、i ノード番号を 16 進数または 10 進数で指定します。

ナビゲーション

表 C-9 に、この表示で使用するコントロールキーを示します。

表 C-9 I 表示のコントロールキー

キー	機能
Ctrl-b	直前の i ノード
Ctrl-f	次の i ノード
Ctrl-k	拡張表示形式

この表示はデバッグ用です。Sun のサポート要員の支援があるときにのみ使用してください。

表示例

コード例 C-9 は、i ノード表示の例です。

コード例 C-9 samu(1M) I の表示

```
Inode    0x1 (1) format: file          samu 4.6 08:27:14 Sept 8 2006
incore: y

00008100 mode      -r-----          409cdf57 access_time
00000001 ino        (1)          1d32ea20
00000001 gen        (1)          4096b499 modify_time
00000002 parent.ino (2)          02588660
00000002 parent.gen (2)          4096b499 change_time
00000000 size_u          02588660
000c0000 size_l      (786432)      4096b443 creation_time
01000000 rm:media/flags      409a8a7c attribute_time
00000000 rm:file_offset  409c0ce6 residence_time
00000000 rm:mau          00000000 unit/cs/arch/flg
00000000 rm:position  00000000 ar_flags
00000000 ext_attrs  -----          00000000 stripe/stride/sg
00000000 ext.ino   (0)          00000000 media  -- --
00000000 ext.gen   (0)          00000000 media  -- --
00000000 uid        root          00000000 psize   (0)
00000000 gid        root          000000c0 blocks  (192)
00000001 nlink     (1)          00000600 free_ino (1536)
00011840 status -n-----  -----  ---

Extents (4k displayed as 1k):
00_ 000000d0.00 000000e0.00 000000f0.00 00000100.00 00000110.00 00000120.00
```

コード例 C-9 samu(1M) I の表示 (続き)

```
06_ 00000130.00 00000140.00 00000150.00 00000160.00 00000170.00 00000180.00
12_ 00000190.00 000001a0.00 000001b0.00 000001c0.00 00000630.00 00000000.00
18_ 00000000.00
```

(J) - プレビュー共用メモリー表示

J 表示は、プレビューキューの共用メモリーセグメントを表示します。

この表示を呼び出すには、次のコマンドを入力します。

```
Command: J
```

ナビゲーション

表 C-10 に、この表示で使用するコントロールキーを示します。

表 C-10 J 表示のコントロールキー

キー	機能
Ctrl-b	逆方向にページ送りする
Ctrl-d	順方向に半ページ送る
Ctrl-f	順方向にページ送りする
Ctrl-u	逆方向に半ページ送る

この表示はデバッグ用です。Sun のサポート要員の支援があるときにのみ使用してください。

表示例

コード例 C-10 は、プレビュー共用メモリー表示の例です。この出力は長いため、マニュアルでは途中から省略しています。

コード例 C-10 samu(1M) J の表示

```
Preview shared memory size: 155648 samu 4.6 08:30:05 Sept 8 2006

00000000 00040000 00014d58 00000000 00000000 .....MX.....
00000010 00000000 00000000 73616d66 73202d20 .....samfs -
00000020 70726576 69657720 6d656d6f 72792073 preview memory s
```

コード例 C-10 samu(1M) J の表示 (続き)

```
00000030 65676d65 6e740000 00026000 00000000 egment.....
00000040 00025fff 00000000 00040000 00014d58 .._.....MX
00000050 00000000 00000000 00000000 00000000 .....
00000060 0000d9e0 00000064 00000000 000001b8 ..Y.....d.....8
00000070 3f800000 447a0000 0000d820 00000008 ?...Dz....X ....
```

(K) - カーネル統計情報表示

K 表示は、現在メモリーに存在する i ノードの数など、カーネル統計情報を表示します。

この表示を呼び出すには、次のコマンドを入力します。

```
Command:K
```

ナビゲーション

表 C-11 に、この表示で使用するコントロールキーを示します。

表 C-11 p 表示のコントロールキー

キー	機能
Ctrl-b	逆方向にページ送りする
Ctrl-f	順方向にページ送りする

この表示はデバッグ用です。Sun のサポート要員の支援があるときのみ使用してください。

表示例

コード例 C-11 は、カーネル統計情報表示の例です。

コード例 C-11 samu(1M) K の表示

```
Kernel statistics samu 4.6 08:33:19 Sept 8 2006

module: sam-qfs name: general instance: 0 class: fs
version 4.4.sam-qfs, gumball 2004-05-07 12:12:04
configured file systems 8
mounted file systems 8
```

コード例 C-11 samu(1M) K の表示 (続き)

```
nhino                16384
ninodes              129526
inocount             129527
inofree              128577
```

(1) - 使用率の表示

1 表示は、各ライブラリおよびファイルシステムに使用されている容量および空き容量など、ファイルシステムの使用率情報を表示します。

この表示を呼び出すには、次のコマンドを入力します。

```
Command:1
```

表示例

コード例 C-12 は、使用率表示の例です。

コード例 C-12 samu(1M) l の表示

```
Usage information                samu      4.6 08:36:27 Sept 8 2006

hostid = 80e69e6e OS name: SunOS Architecture: SPARC CPUs: 2 (2 online)

library 40: capacity389.3G bytes space291.1Gbytes, usage 25%
library 51: capacity9.5G bytes space9.5Gbytes, usage 0%
library 55: capacity0bytes space0bytes, usage 0%
library 56: capacity10.7G bytes space10.7Gbytes, usage 0%
library totals: capacity409.5G bytes space311.3Gbytes, usage 24%

filesystem samfs3: capacity54.5Mbytes space13.4Mbytes, usage 75%
filesystem samfs4: capacity319.5Mbytes space298.0Mbytes, usage 7%
filesystem samfs7: capacity96.6Mbytes space69.6Mbytes, usage 28%
filesystem samfs6: capacity5.0Gbytes space4.9Gbytes, usage 3%
filesystem samfs8: capacity5.0Gbytes space4.9Gbytes, usage 2%
filesystem totals: capacity10.5Gbytes space10.2Gbytes, usage 3%
```

注 - 4U3 よりも前のバージョンのソフトウェアでは、この表示にファイルシステムのライセンス情報が表示されます。

(L) - 共用メモリー表示

L 表示は、共用メモリーテーブルの位置を表示します。共用メモリーに保存されているシステムデフォルトも表示されます。

この表示を呼び出すには、次のコマンドを入力します。

```
Command: L
```

この表示はデバッグ用です。Sun のサポート要員の支援があるときのみ使用してください。

表示例

コード例 C-13 は、共用メモリーテーブルの例です。

コード例 C-13 samu(1M) L の表示

```
Shared memory tables          samu 4.6 08:38:31 May  8 2007

shm ptr tbl:                  defaults:
size          12000 (73728)    optical          mo
left          44c8 (17608)     tape             lt
scanner pid   1861            timeout          600
fifo path     01b0 /var/opt/SUNWsamfs/previews 100
dev_table     01cc            stages           1000
first_dev     0450            log_facility     184
scan_mess     cf50            dio minfilesize 100
preview_shmid 1                label barcode   FALSE
flags         0x20000000       barcodes low    FALSE
preview stages 55776           export unavail  FALSE
preview avail 100              attended        TRUE
preview count 0                start rpc       FALSE
preview sequence 445
age factor    1                vsn factor      1000
fs tbl ptr 0xd820             fs count        8
fseq 10 samfs1 state 0        0        0        0        0
fseq 20 samfs2 state 0        0        0        0        0
fseq 30 qfs1 state 0          0        0        0        0
fseq 40 qfs2 state 0          0        0        0        0
fseq 50 qfs3 state 0          0        0        0        0
fseq 60 qfs4 state 0          0        0        0        0
fseq 100 shareqfs1 state 0    0        0        0        0
fseq 110 shareqfs2 state 0    0        0        0        0
```

(m) - 外部ストレージの状態表示

m 表示には、外部記憶ファイルシステムとそのメンバードライブの状態が表示されます。この表示では、マウントされているファイルシステムのみが表示されます。

この表示を呼び出すには、次のコマンドを入力します。

```
Command:m
```

表示例

コード例 C-14 は、m 表示の例です。メンバードライブは、空白文字 1 つ分インデントされ、そのドライブが属するファイルシステムのすぐ下に表示されます。

コード例 C-14 samu(1M) m の表示

```
Mass storage status                               samu 4.6 08:41:11 Sept 8 2006
```

ty	eq	status	use	state	ord	capacity	free	ra	part	high	low
ms	10	m----2----d	1%	on		68.354G	68.343G	1M	16	90%	70%
md	11		1%	on	0	13.669G	13.666G				
md	12		1%	on	1	13.669G	13.667G				
md	13		1%	on	2	13.669G	13.667G				
md	14		1%	on	3	13.674G	13.672G				
md	15		1%	on	4	13.674G	13.672G				
ms	20	m----2----d	1%	on		68.354G	68.344G	1M	16	90%	70%
md	21		1%	on	0	13.669G	13.667G				
md	22		1%	on	1	13.669G	13.667G				
md	23		1%	on	2	13.669G	13.667G				
md	24		1%	on	3	13.674G	13.672G				
md	25		1%	on	4	13.674G	13.672G				
ma	30	m----2----d	4%	on		64.351G	61.917G	1M	16	90%	70%
mm	31		1%	on	0	4.003G	3.988G	[8363840 inodes]			
md	32		4%	on	1	64.351G	61.917G				
ma	40	m----2----d	1%	on		64.351G	64.333G	1M	16	90%	70%
mm	41		1%	on	0	4.003G	3.997G	[8382784 inodes]			
md	42		1%	on	1	64.351G	64.333G				
ma	50	m----2---r-	1%	on		64.351G	64.333G	1M	16	90%	70%
mm	51		1%	on	0	4.003G	3.997G	[8382784 inodes]			
mr	52		1%	on	1	64.351G	64.333G				
ma	60	m----2---r-	1%	on		64.351G	64.331G	1M	16	90%	70%
mm	61		1%	on	0	4.003G	3.997G	[8382784 inodes]			
mr	62		1%	on	1	64.351G	64.331G				
ma	100	m----2c--r-	2%	on		270.672G	265.105G	1M	16	80%	70%
mm	101		1%	on	0	2.000G	1.988G	[4168992 inodes]			
mr	102		2%	on	1	270.672G	265.469G				

コード例 C-14 samu(1M) m の表示 (続き)

ma	110	m----2c--r-	3%	on	270.656G	263.382G	1M	16	80%	70%
mm	111		1%	on	0	2.000G	1.987G	[4167616	inodes]	
mr	112		2%	on	1	270.656G	264.736G			

フィールドの説明

表 C-12 で、この表示のフィールドを説明します。

表 C-12 samu(1M) m の表示フィールドの説明

フィールド	説明
ty	装置タイプ。
eq	外部ストレージの装置番号。
status	デバイスの状態。状態コードの説明については、309 ページの「オペレータ表示の状態コード」を参照してください。
use	ディスク空間の使用率。
state	外部ストレージの現在の動作状態。
ord	ストレージファミリセット内のディスク装置の順番。
capacity	1024 バイト単位の使用可能なディスク空間ブロック数。
free	利用可能な 1024 バイト単位のディスク空間ブロック数。
ra	キロバイト単位による先読みのサイズ。
part	キロバイト単位による部分的な書き込みサイズ。
high	ディスク使用量の上限率。
low	ディスク使用量の下限率。

(M) - 共用メモリー表示

M 表示は、生の共用メモリーセグメントを 16 進数で表示します。これは装置テーブルです。

この表示を呼び出すには、次のコマンドを入力します。

```
Command:M
```

ナビゲーション

表 C-13 に、この表示で使用するコントロールキーを示します。

表 C-13 M 表示のコントロールキー

キー	機能
Ctrl-b	逆方向にページ送りする
Ctrl-d	順方向に半ページ送る
Ctrl-f	順方向にページ送りする
Ctrl-u	逆方向に半ページ送る

この表示はデバッグ用です。Sun のサポート要員の支援があるときにのみ使用してください。

表示例

コード例 C-15 は、共用メモリー表示の例です。この出力は長いので、マニュアルでは途中から省略しています。

コード例 C-15 samu(1M) M の表示

```
Shared memory      size: 73728      samu 4.6 08:43:20 May  8 2007
00000000 00040000 00014d58 00000000 00000000 .....MX.....
00000010 00000000 00000000 73616d66 73202d20 .....samfs -
00000020 73686172 6564206d 656d6f72 79207365 shared memory se
00000030 676d656e 74000000 00012000 000044c8 gment..... ..DH
00000040 0000dd20 00000000 00000742 00000745 ..] .....B...E
00000050 00000001 00000000 00000000 c0000000 .....@...
00000060 00000001 0001534d 00000000 00000000 .....SM.....
00000070 00000000 00000000 00000000 00000000 .....

00000080 00000000 00000000 00000000 00000000 .....
00000090 20000000 000001b0 000001cc 00000450 .....0...L...P
000000a0 0000cf50 00000001 00000001 4c696365 ..OP.....Lice
000000b0 6e73653a 204c6963 656e7365 206e6576 nse: License nev
000000c0 65722065 78706972 65732e00 00000000 er expires.....
000000d0 00000000 00000000 00000000 00000000 .....
000000e0 00000000 00000000 00000000 00000000 .....
000000f0 00000000 00000000 00000000 00000000 .....
```

(n) - 書き込みの状態表示

n 表示には、すべてのメディアのステージャーの状態が表示されます。未処理の書き込み要求のリストが表示されます。

この表示は、次のように、必要な情報に応じて異なる方法で表示できます。

- すべての書き込み活動の書き込み状態を表示するには、次の形式でコマンドを入力します。

```
Command:n
```

- 特定のメディアタイプの書き込み状態を表示するには、次の形式でコマンドを入力します。

```
Command:n mt
```

mt には、mcf(4) のマニュアルページに記載されているメディアタイプのいずれかを指定します。

表示例

コード例 C-16 は、書き込み状態表示の例です。

コード例 C-16 samu(1M) n の表示

```
Staging status                               samu 4.6 08:47:16 May  8 2007

Log output to: /var/opt/SUNWsamfs/stager/log

Stage request: dt.DAT001
Loading VSN DAT001

Staging queues
ty pid  user      status  wait files vsn
dt 16097 root      active  0:00   12 DAT001
```

(N) - ファイルシステムパラメータ表示

N 表示は、すべてのマウントポイントパラメータ、スーパーブロックバージョン、およびそのほかのファイルシステム情報を表示します。

この表示を呼び出すには、次のコマンドを入力します。

```
Command: N
```

ナビゲーション

表 C-14 に、この表示で使用するコントロールキーを示します。

表 C-14 N 表示のコントロールキー

キー	機能
Ctrl-b	直前のファイルシステム
Ctrl-d	順方向にページパーティションを送る
Ctrl-f	次のファイルシステム
Ctrl-i	詳細に状態を解釈する
Ctrl-u	逆方向にページパーティションを送る

この表示はデバッグ用です。Sun のサポート要員の支援があるときにのみ使用してください。

表示例

コード例 C-17 は、ファイルシステムパラメータ表示の例です。

コード例 C-17 samu(1M) N の表示

```
File system parameters                samu 4.6 08:55:19 Sept 8 2006

mount_point      : /sam1                partial           : 16k
fs_type          : 6                    maxpartial       : 16k
server           :                      partial_stage    : 16384
filesystem name: samfs1                 flush_behind     : 0
eq_type          : 10 ms                 stage_flush_beh: 0
state version    : 0 2                   stage_n_window   : 262144
(fs,mm)_count   : 5 0                   stage_retries    : 3
sync_meta       : 0                     stage timeout    : 0
stripe          : 0                      dio_consec r,w  : 0 0
```

コード例 C-17 samu(1M) N の表示 (続き)

```
mm_stripe      : 1                      dio_frm_min r,w: 256 256
high low       : 90% 70%                dio_ill_min r,w:  0  0
readahead     : 1048576                  ext_bsize      : 4096
writebehind    : 524288
wr_throttle    : 16777216
rd_ino_buf_size: 16384
wr_ino_buf_size: 512
config         : 0x08520530              mflag          : 0x00000044
status         : 0x00000001

Device configuration:
ty  eq state  device_name                fs family_set
md  11 on    /dev/dsk/c5t8d0s3            10 samfs1
md  12 on    /dev/dsk/c5t8d0s4            10 samfs1
md  13 on    /dev/dsk/c5t8d0s5            10 samfs1
md  14 on    /dev/dsk/c5t8d0s6            10 samfs1
md  15 on    /dev/dsk/c5t8d0s7            10 samfs1
```

(o) - 光磁気ディスクの状態表示

o 表示は、環境内で構成されているすべての光磁気ディスクドライブの状態を表示します。

この表示を呼び出すには、次のコマンドを入力します。

```
Command: o
```

ナビゲーション

表 C-15 に、この表示で使用するコントロールキーを示します。

表 C-15 o 表示のコントロールキー

キー	機能
Ctrl-b	逆方向にページ送りする
Ctrl-f	順方向にページ送りする
Ctrl-b	逆方向にページ送りする
Ctrl-d	順方向に半ページ送る

表 C-15 ◦ 表示のコントロールキー (続き)

キー	機能
Ctrl-f	順方向にページ送りする
Ctrl-k	選択する (手動、自動ライブラリ、両方、優先順位)
Ctrl-u	逆方向に半ページ送る

表示例

コード例 C-18 は、光磁気ディスクの状態表示の例です。

コード例 C-18 samu(1M) ◦ の表示

```
Optical disk status          samu    4.6 Thu Oct 11 13:15:40
ty  eq  status      act  use  state  vsn
mo 35  --l---wo-r    1 29%  ready  oper2
```

フィールドの説明

表 C-16 で、この表示のフィールドを説明します。

表 C-16 samu(1M) ◦ の表示フィールドの説明

フィールド	説明
ty	装置タイプ。
eq	光磁気ディスクの装置番号。
status	デバイスの状態。状態コードの説明については、309 ページの「オペレータ表示の状態コード」を参照してください。
act	動作カウント。

表 C-16 samu(1M) の表示フィールドの説明

フィールド	説明
use	カートリッジ空間の使用率。
state	光磁気ディスクの現在の動作状態。有効なデバイスの状態は、次のとおりです。 <ul style="list-style-type: none"> • ready - 装置はオンであり、ディスクがトランスポートに読み込まれています。アクセスに利用可能。 • notrdy - 装置はオンであるが、トランスポートにディスクが存在しません。 • idle - 装置は、新しい接続に利用できません。進行中の操作は、終了するまで続行されます。 • off - 装置をアクセスに利用できません。 • down - 装置は、メンテナンスアクセスだけに利用可能。
vsn	光磁気ディスクに割り当てられたボリュームシリアル名。ボリュームにラベルが付いていない場合には、キーワード nolabel。

(p) - リムーバブルメディアの読み込み要求の表示

p 表示は、保留中のリムーバブルメディアの読み込み要求に関する情報を一覧表示します。mt 引数を使用して、DLT テープなどの特定のメディアタイプ、またはテープなどのメディアファミリを選択できます。優先順位表示には、ユーザー待ち行列ではなくプレビュー待ち行列における優先順位が表示され、優先順位別にエントリがソートされます。

マウント要求は、次の形式で表示されます。

- 手動要求と自動ライブラリ要求の両方 (ユーザーごと)
- 手動要求と自動ライブラリ要求の両方 (優先順位ごと)
- 手動要求のみ
- 自動ライブラリ要求のみ

この表示は、次のように、必要な情報に応じて異なる方法で表示できます。

- 現在選択されているすべてのリムーバブルデバイスのマウント要求を表示するには、次の形式でコマンドを入力します。

```
Command:p
```

- 指定したリムーバブルメディアタイプの装置のマウント要求を表示するには、次の形式でコマンドを入力します。

```
Command: p mt
```

mt には、*mcf(4)* のマニュアルページに記載されているメディアタイプのいずれかを指定します。

ナビゲーション

表 C-17 に、この表示で使用するコントロールキーを示します。

表 C-17 p 表示のコントロールキー

キー	機能
Ctrl-b	逆方向にページ送りする
Ctrl-d	順方向に半ページ送る
Ctrl-f	順方向にページ送りする
Ctrl-k	別の表示形式に切り替える
Ctrl-u	逆方向に半ページ送る

表示例

コード例 C-19 は、リムーバブルメディアの読み込み要求表示の例です。

コード例 C-19 samu(1M) p の表示

```
Removable media load requests all both samu 4.6 09:14:19 Sept 8 2006
count: 1

index type pid      user      rb  flags      wait count  vsn
  0  dt   15533   root      150 W--f---   0:00      DAT001
```

フィールドの説明

表 C-18 で、この表示のフィールドを説明します。

表 C-18 samu(1M) p の表示フィールドの説明

フィールド	説明
index	プレビューテーブル内の索引番号。
type	リムーバブルメディアに割り当てられている装置タイプコード。
pid	UNIX プロセス識別子。プロセス識別子 1 は、NFS アクセスを示します。
user	読み込みを要求しているユーザーに割り当てられている名前。
priority	要求の優先順位。
rb	要求されている VSN が常駐している自動ライブラリの装置番号。
flags	装置のフラグ。表 C-19 を参照してください。
wait	マウント要求を受信してから経過した時間。
count	書き込みである場合、この VSN に対する要求数。
vsn	ボリュームのボリュームシリアル名。

フラグ

表 C-19 で、p 表示のフラグを示します。

表 C-19 samu(1M) p 表示の flags フィールド

フィールド	説明
W-----	書き込みアクセスが要求されました。
-b-----	エントリが混み合っている状態です。
--C-----	VSN のクリアが要求されました。
---f---	ファイルシステムが要求されました。
----N--	メディアがファイルシステムにとって異種です。
-----S-	すでにマウントされている側を切り替えます。
-----s	書き込み要求フラグです。

(P) - アクティブサービス表示

P 表示は、Sun StorageTek QFS の単一ポートマルチプレクサに登録されているサービスを一覧表示します。

この表示を呼び出すには、次のコマンドを入力します。

```
Command: P
```

ナビゲーション

表 C-20 に、この表示で使用するコントロールキーを示します。

表 C-20 P 表示のコントロールキー

キー	機能
Ctrl-b	逆方向にページ送りする
Ctrl-f	順方向にページ送りする

この表示はデバッグ用です。Sun のサポート要員の支援があるときにのみ使用してください。

表示例

コード例 C-20 は、アクティブサービス表示の例です。

コード例 C-20 samu(1M) P の表示

```
Active Services                samu      4.6 09:08:33 Sept 8 2006

Registered services for host 'pup':
  sharedfs.qfs2
  sharedfs.qfs1
  2 services registered.
```

(r) - リムーバブルメディアの状態表示

r 表示により、テープドライブなどのリムーバブルメディア装置の稼働状況を監視できます。ビデオテープなどの特定のメディアタイプ、またはすべてのテープ装置などのメディアファミリーを選択できます。

この表示は、次のように、必要な情報に応じて異なる方法で表示できます。

- すべてのリムーバブルメディア装置の状態を表示するには、次の形式でコマンドを入力します。

```
Command: r
```

- 特定の装置の状態を表示するには、次の形式でコマンドを入力します。

```
Command: r eq
```

eq には、装置の装置番号を指定します。

表示例

コード例 C-21 は、リムーバブルメディアの状態表示の例です。

コード例 C-21 samu(1M) r の表示

```
Removable media status: all                samu 4.6 09:11:27 Sept 8 2006
ty  eq  status      act  use  state  vsn
dt 150  --l-----r   0  63%  ready  DAT001
```

フィールドの説明

表 C-21 で、この表示のフィールドを説明します。

表 C-21 samu(1M) r の表示フィールドの説明

フィールド	説明
ty	装置タイプ。
eq	ドライブの装置番号。
status	デバイスの状態。状態コードの説明については、309 ページの「オペレータ表示の状態コード」を参照してください。
act	動作カウント。
use	カートリッジ空間の使用率。

表 C-21 samu(1M) r の表示フィールドの説明 (続き)

フィールド	説明
state	リムーバブルメディアの現在の動作状態。有効なデバイスの状態は、次のとおりです。 <ul style="list-style-type: none">• ready - 装置はオンであり、ディスクまたはテープがトランスポートに読み込まれています。アクセスに利用可能。• notrdy - 装置はオンであるが、トランスポートにディスクまたはテープが存在しません。• idle - 装置は、新しい接続に利用できません。進行中の操作は、終了するまで続行されます。• off - 装置をアクセスに利用できません。• down - 装置は、メンテナンスアクセスだけに利用可能。• nalloc - この装置への以後の割り当てを禁止する nalloc フラグが設定されています。詳細は、29 ページの「論理ユニット番号 (LUN) による割り当て制御」を参照してください。
vsn	ボリュームに割り当てられているボリュームシリアル名。ボリュームがラベル付けされていない場合には、キーワード nolabel。ボリュームがトランスポートに存在していない場合や装置がオフである場合には、空白のままです。

(R) - Sun SAM-Remote 情報表示

R 表示は、Sun SAM-Remote の構成に関する情報を表示します。

この表示を呼び出すには、次のコマンドを入力します。

```
Command: R
```

この表示はデバッグ用です。Sun のサポート要員の支援があるときにのみ使用してください。

(s) - デバイスの状態表示

s 表示は、環境内で構成されているすべての装置の状態を表示します。

この表示を呼び出すには、次のコマンドを入力します。

```
Command: s
```

ナビゲーション

表 C-22 に、この表示で使用するコントロールキーを示します。

表 C-22 s 表示のコントロールキー

キー	機能
Ctrl-b	逆方向にページ送りする
Ctrl-d	順方向に半ページ送る
Ctrl-f	順方向にページ送りする
Ctrl-u	逆方向に半ページ送る

表示例

コード例 C-22 は、デバイスの状態表示の例です。

コード例 C-22 samu(1M) s の表示

```
Device status                               samu      4.6 09:14:05 Sept 8 2006
ty      eq state  device_name                                fs status  pos
sk      100 on    /etc/opt/SUNWsamfs/dcstkconf                100 m-----r
        stk_dismount(2275) 0, volser 700073
sg      120 on    /dev/rmt/2cbn                                100 -----p
        empty
sg      130 on    /dev/rmt/5cbn                                100 --l----o-r
        Ready for data transfer
sg      140 on    /dev/rmt/6cbn                                100 -----p
        empty
sg      150 on    /dev/rmt/4cbn                                100 -----p
        empty
hy      151 on    historian                                      151 -----
```

フィールドの説明

表 C-23 で、この表示のフィールドを説明します。

表 C-23 samu(1M) s の表示フィールドの説明

フィールド	説明
ty	装置タイプ。
eq	装置の装置番号。
state	装置の現在の動作状態。
device_name	装置のパス。ファイルシステム装置の場合は、ファイルシステム名。
fs	装置が属するファミリーセットの装置番号。
status	デバイスの状態。状態コードの説明については、309 ページの「オペレータ表示の状態コード」を参照してください。

(S) - セクターデータ表示

S 表示は、生の装置データを表示します。

この表示を呼び出すには、次のコマンドを入力します。

```
Command: S
```

ナビゲーション

表 C-24 に、この表示で使用するコントロールキーを示します。

表 C-24 S 表示のコントロールキー

キー	機能
Ctrl-b	直前のセクター
Ctrl-d	順方向にページ送りする (上部)
Ctrl-f	次のセクター
Ctrl-k	拡張表示形式
Ctrl-u	逆方向にページ送りする (上部)

この表示はデバッグ用です。Sun のサポート要員の支援があるときにのみ使用してください。

(t) - テープドライブの状態表示

t 表示は、環境内で構成されているすべてのテープドライブの状態を表示します。

この表示を呼び出すには、次のコマンドを入力します。

```
Command:t
```

ナビゲーション

表 C-25 に、この表示で使用するコントロールキーを示します。

表 C-25 t 表示のコントロールキー

キー	機能
Ctrl-b	逆方向にページ送りする
Ctrl-f	順方向にページ送りする

表示例

コード例 C-23 は、テープドライブの状態表示の例です。

コード例 C-23 samu(1M) t の表示

```
Tape drive status                                samu      4.6 09:21:07 Sept 8 2006
ty  eq  status      act  use  state  vsn
sg 120 -----p    0   0%  notrdy
      empty
sg 130 -----p    0   0%  notrdy
      empty
sg 140 -----p    0   0%  notrdy
      empty
sg 150 --l-----r    0  41%  ready   700088
      idle
```

フィールドの説明

表 C-26 で、この表示のフィールドを説明します。

表 C-26 samu(1M) t の表示フィールドの説明

フィールド	説明
ty	装置タイプ。
eq	ドライブの装置番号。
status	デバイスの状態。状態コードの説明については、309 ページの「オペレータ表示の状態コード」を参照してください。
act	動作カウント。
use	カートリッジ空間の使用率。
state	リムーバブルメディアの現在の動作状態。有効なデバイスの状態は、次のとおりです。 <ul style="list-style-type: none">• ready - 装置はオンであり、ディスクまたはテープがトランスポートに読み込まれている。アクセスに利用可能。• notrdy - 装置はオンであるが、トランスポートにディスクまたはテープが存在しません。• idle - 装置は、新しい接続に利用できません。進行中の操作は、終了するまで続行されます。• off - 装置をアクセスに利用できません。• down - 装置は、メンテナンスアクセスだけに利用可能。• nalloc - この装置への以後の割り当てを禁止する nalloc フラグが設定されています。詳細は、29 ページの「論理ユニット番号 (LUN) による割り当て制御」を参照してください。
vsn	ボリュームに割り当てられているボリュームシリアル名。ボリュームがラベル付けされていない場合には、キーワード no label になります。ボリュームがトランスポートに存在していない場合や装置がオフである場合には、空白のままです。

(T) - SCSI センスタータ表示

T 表示は、SCSI 装置の SCSI の状態を表示します。

この表示を呼び出すには、次のコマンドを入力します。

```
Command: T
```

ナビゲーション

表 C-27 に、この表示で使用するコントロールキーを示します。

表 C-27 T 表示のコントロールキー

キー	機能
Ctrl-b	直前の装置
Ctrl-f	次の装置

この表示はデバッグ用です。Sun のサポート要員の支援があるときにのみ使用してください。

(u) - 書き込み待ち行列表示

u 表示には、書き込み待ち行列に入っているすべてのファイルが表示されます。

この表示を呼び出すには、次のコマンドを入力します。

```
Command:u
```

ナビゲーション

表 C-28 に、この表示で使用するコントロールキーを示します。

表 C-28 u 表示のコントロールキー

キー	機能
Ctrl-b	逆方向にページ送りする
Ctrl-d	順方向に半ページ送る
Ctrl-f	順方向にページ送りする
Ctrl-k	各エントリの 2 行目にパスを表示する
Ctrl-u	逆方向に半ページ送る

表示例

コード例 C-24 は、書き込み待ち行列表示の例です。

コード例 C-24 samu(1M) u の表示

```
Staging queue by media type: all          samu 4.6 09:24:23 Sept 8 2006
volumes 1 files 22
```

ty	length	fseq	ino	position	offset	vsn
dt	451.611k	20	1030	207cc	473	DAT001
dt	341.676k	20	1031	207cc	7fc	DAT001
dt	419.861k	20	1032	207cc	aa9	DAT001
dt	384.760k	20	1033	207cc	df2	DAT001
dt	263.475k	20	1034	207cc	10f5	DAT001
dt	452.901k	20	1035	207cc	1305	DAT001
dt	404.598k	20	1036	207cc	1690	DAT001
dt	292.454k	20	1037	207cc	19bb	DAT001
dt	257.835k	20	1038	207cc	1c05	DAT001
dt	399.882k	20	1040	207cc	1e0b	DAT001
dt	399.882k	40	1029	208d7	2	DAT001
dt	257.835k	40	1030	208d7	323	DAT001
dt	292.454k	40	1031	208d7	528	DAT001
dt	404.598k	40	1032	208d7	772	DAT001
dt	452.901k	40	1033	208d7	a9d	DAT001
dt	263.475k	40	1034	208d7	e28	DAT001
dt	384.760k	40	1035	208d7	1038	DAT001
dt	419.861k	40	1036	208d7	133b	DAT001
dt	341.676k	40	1037	208d7	1684	DAT001
dt	451.611k	40	1038	208d7	1931	DAT001
dt	161.326k	40	1039	208d7	1cba	DAT001
dt	406.400k	40	1040	208d7	1dfe	DAT001

フィールドの説明

表 C-29 で、この表示のフィールドを説明します。

表 C-29 samu(1M) u の表示フィールドの説明

フィールド	説明
ty	装置タイプ。
length	ファイルの長さ。
fseq	ファイルシステム装置番号。
ino	i ノード番号。

表 C-29 samu(1M) u の表示フィールドの説明

フィールド	説明
position	特定のメディア上にあるアーカイブファイルの位置。
offset	特定のメディア上にあるアーカイブファイルのオフセット。
vsn	ボリュームのボリュームシリアル名。

(U) - 装置テーブル表示

U 表示は、目で読める形式で装置テーブルを表示します。

この表示は、次のように、必要な情報に応じて異なる方法で表示できます。

- すべての装置の装置テーブルを表示するには、次の形式でコマンドを入力します。

```
Command:U
```

- 特定の装置の装置テーブルを表示するには、次の形式でコマンドを入力します。

```
Command:U eq
```

eq には、装置の装置番号を指定します。

ナビゲーション

表 C-30 に、この表示で使用するコントロールキーを示します。

表 C-30 U 表示のコントロールキー

キー	機能
Ctrl-b	直前の装置
Ctrl-f	次の装置

この表示はデバッグ用です。Sun のサポート要員の支援があるときのみ使用してください。

表示例

コード例 C-25 は、装置テーブル表示の例です。

コード例 C-25 samu(1M) U の表示

```
Device table: eq: 10      addr: 00000450  samu 4.6 09:28:40 Sept 8 2006

message:

0004000000014d58 0000000000000000          00000000 delay
0000000000000000 mutex                    00000000 unload_delay
00000aa8 next
73616d66 set:  samfs1
73310000
00000000
00000000
000a000a eq/fseq
08010801 type/equ_type
0000      state
00000000 st_rdev
00000000 ord/model
00000000 mode_sense
00000000 sense
00000000 space
00000000 capacity
00000000 active
00000000 open
00000000 sector_size
00000000 label_address
00000000 vsn:
00000000
00000000
00000000
00000000 status: -----
00000000 dt
73616d66 name: samfs1
```

(v) - 自動ライブラリカタログ表示

v 表示には、自動ライブラリに現在記録されているすべてのディスクやテープの場所と VSN が表示されます。

この表示は、次のように、必要な情報に応じて異なる方法で表示できます。

- すべての装置のカタログを表示するには、次の形式でコマンドを入力します。

```
Command: v
```

- 特定の装置のカタログ情報を表示するには、次の形式でコマンドを入力します。

```
Command: v eq
```

eq には、装置の装置番号を指定します。履歴カタログを表示するには、キーワード *historian* を入力します。

samu(1M) は、次のように、装置の入力を指示するプロンプトを表示することがあります。

```
Enter robot: eq
```

eq に装置の装置番号を指定するか、復帰改行を押します。復帰改行を押した場合は、直前に指定した装置の情報が表示されます。

装置名と装置番号のリストについては、267 ページの「(c) - デバイス構成表示」を参照してください。

ナビゲーション

表 C-31 に、この表示で使用するコントロールキーを示します。

表 C-31 v 表示のコントロールキー

キー	機能
Ctrl-b	逆方向にページ送りします。
Ctrl-d	次のライブラリカタログ。
Ctrl-f	順方向にページ送りします。
Ctrl-i	詳細、2行表示形式。Ctrl-i を 1 回入力すると、時間とパーコードが表示され、2 回入力すると、ボリューム予約が 2 行目に表示されます。
Ctrl-k	拡張ソートキー。Ctrl-k を入力したあと、次のいずれかを入力して、ソートキーを選択します。 <ul style="list-style-type: none"> 1 - スロット別にソート。 2 - カウント別にソート。 3 - 使用率別にソート。

表 C-31 v 表示のコントロールキー (続き)

キー	機能
	4 - VSN 別にソート。
	5 - アクセス時間別にソート。
	6 - バーコード別にソート。
	7 - ラベル時間別にソート。
Ctrl-u	直前の自動ライブラリカタログ。
/	VSN を検索します。
%	バーコードを検索します。
\$	スロットを検索します。

表示例

コード例 C-26 は、自動ライブラリカタログ表示の例です。

コード例 C-26 samu(1M) v の表示

```

Robot VSN catalog by slot      : eq 100samu  4.6 09:30:25 Sept 8 2006
count 32
slot      access time count use flags      ty vsn
  0      2004/05/08 08:35   64   0% -il-o-b----- sg 700071
  1      2004/05/08 09:08   27  12% -il-o-b----- sg 700073
  2      2004/05/08 09:12   26  12% -il-o-b----- sg 700077
  3      2004/05/08 08:39   37  40% -il-o-b----- sg 700079
  4      2004/05/08 09:16   24   6% -il-o-b----- sg 700084
  5      2004/05/08 09:18   24  41% -il-o-b----- sg 700088
  6      none                0   0% -il-o-b----- sg 700090
  7      none                0   0% -il-o-b----- sg 700092
  8      none                0   0% -il-o-b----- sg 000155
  9      none                0   0% -il-o-b----- sg 000156
 10      none                0   0% -il-o-b----- sg 000157
 11      none                0   0% -il-o-b----- sg 000158
 12      none                0   0% -il-o-b----- sg 000154
 13      none                0   0% -il-o-b----- sg 000153
 14      none                0   0% -il-o-b----- sg 000152

```

フィールドの説明

表 C-32 で、この表示のフィールドを説明します。

表 C-32 samu(1M) v の表示フィールドの説明

フィールド	説明
Robot VSN catalog	指定された自動ライブラリの名前と再表示された時間。
count	このライブラリのカatalogに割り当てられたスロットの数。
slot	指定ライブラリ内のスロット番号。
access time	ボリュームが最後にアクセスされた時刻。
count	最後の監査を行なってからのこのボリュームへのアクセス数。
use	ボリュームの空間使用率。
flags	装置のフラグ。フラグについては、表 C-33 を参照してください。
ty	装置タイプ。
vsn	ボリュームのボリュームシリアル名。

フラグ

表 C-33 に、表 C-32 の flags フィールドのフラグを示します。場合によっては、1 つのフィールドに複数のフラグが表示され、あるフラグがほかのフラグを上書きすることがあります。

表 C-33 samu(1M) v 表示の flags フィールド

フラグ	説明
A-----	ボリュームを監査する必要があります。
-i-----	使用中スロット。
--l-----	ラベルあり。N より優先されます。
--N-----	ラベルなし。このボリュームは環境にとって異種です。
---E-----	メディアエラー。ソフトウェアがカートリッジの書き込みエラーを検出したときに設定されます。
----o-----	スロットが占有されています。
----C-----	ボリュームはクリーニングテープです。p よりも優先されます。
----p-----	優先 VSN。
----b-----	バーコードが検出されました。

表 C-33 samu(1M) v 表示の flags フィールド (続き)

フラグ	説明
-----W----	書き込み保護。カートリッジに対して物理的な書き込み保護メカニズムが使用可能になるときに設定されます。
-----R----	読み込み専用。
-----c--	リサイクル。
-----d-	重複 VSN。U よりも優先されます。
-----U-	利用不可のボリューム。
-----f	ボリュームに空きがないことをアーカイバが検出しました。
-----X	エクスポートスロット。

(w) - 保留書き込み待ち行列の表示

w 表示には、ボリュームの読み込みがまだ行われていない待ち行列内の書き込み要求が表示されます。

この表示は、次のように、必要な情報に応じて異なる方法で表示できます。

- すべてのメディアの保留書き込み待ち行列を表示するには、次の形式でコマンドを入力します。

```
Command:w
```

- 特定のメディアタイプの保留書き込み待ち行列を表示するには、次の形式でコマンドを入力します。

```
Command:w mt
```

mt には、mcf(4) のマニュアルページに記載されているメディアタイプのいずれかを指定します。

ナビゲーション

表 C-34 に、この表示で使用するコントロールキーを示します。

表 C-34 w 表示のコントロールキー

キー	機能
Ctrl-b	逆方向にページ送りする
Ctrl-d	順方向に半ページ送る
Ctrl-f	順方向にページ送りする
Ctrl-k	各エントリの 2 行目にパスを表示する
Ctrl-u	逆方向に半ページ送る

表示例

コード例 C-27 は、保留書き込み待ち行列表示の例です。

コード例 C-27 samu(1M) w の表示

```
Pending stage queue by media type: all      samu      4.6 Thu Oct 11 13:20:27
volumes 1 files 13

ty      length  fseq  ino   position  offset  vsn
at      1.383M   1     42    3a786     271b   000002
at      1.479M   1     56    3a786     5139   000002
at     1018.406k  1     60    3a786     6550   000002
at      1.000M   1     65    3a786     7475   000002
at      1.528M   1     80    3a786     99be   000002
at      1.763M   1     92    3a786     ce57   000002
at      1.749M   1    123    3a786    11ece   000002
at     556.559k  1    157    3a786    1532f   000002
at     658.970k  1    186    3a786    17705   000002
at     863.380k  1    251    3a786    1dd58   000002
at     1.268M   1    281    3a786    1f2b7   000002
at     1.797M   1    324    3a786    23dfa   000002
at     1.144M   1    401    3a786    2bb6d   000002
```

フィールドの説明

表 C-35 で、この表示のフィールドを説明します。

表 C-35 samu(1M) w の表示フィールドの説明

フィールド	説明
ty	装置タイプ。
length	ファイルの長さ。
fseq	ファイルシステム装置番号。
ino	i ノード番号。
position	特定のメディア上にあるアーカイブファイルの位置 (10 進数の形式による)。
offset	特定のメディア上にあるアーカイブファイルのオフセット。
vsn	ボリュームのボリュームシリアル名。

オペレータ表示の状態コード

オペレータ表示には、リムーバブルメディア装置表示とファイルシステム表示で異なる状態コードが用意されています。これらの状態コードについて次に説明します。

リムーバブルメディア装置表示の状態コード

o、r、s、および t のオペレータ表示には、リムーバブルメディア装置の状態コードが表示されます。状態コードは 10 位置形式で表示され、左 (位置 1) から右 (位置 10) の方向に読みます。

この項で説明する状態コードは、samu(1M) f、m、および v 表示には適用されません。f および m 表示の状態コードについては、310 ページの「ファイルシステム表示の状態コード」を参照してください。v 表示の状態コードについては、303 ページの「(v) - 自動ライブラリカタログ表示」を参照してください。

表 C-36 に、各位置の有効な状態コードを示します。

表 C-36 リムーバブルメディア装置表示の状態コード

状態ビット	意味
s-----	メディアが走査中です。
m-----	自動ライブラリは動作可能。
M-----	メンテナンスモード。

表 C-36 リムーバブルメディア装置表示の状態コード (続き)

状態ビット	意味
-E-----	装置が走査中に回復不能エラーを受信しました。
-a-----	装置が監査モードにあります。
--l-----	メディアにラベルが付いています。
--N-----	外部メディア。
--L-----	メディアにラベルを付ける処理中。
---I-----	装置が休止状態となるのを待機しています。
---A-----	オペレータ操作が必要です。
----C-----	クリーニングが必要です。
----U-----	取り出しが要求されました。
-----R-----	装置が予約されています。
-----w---	プロセスがメディアに書き込みを行なっています。
-----o--	装置がオープン状態にあります。
-----P-	装置が位置付けられています (テープのみ)。
-----F-	自動ライブラリの場合、すべてのストレージスロットが占有されています。テープと光磁気ドライブの場合、メディアがいっぱいです。
-----R	装置はレディー状態にあり、メディアは読み込み専用です。
-----r	装置は回転立ち上げしており、準備完了状態にあります。
-----p	装置が存在しています。
-----W	装置は書き込み保護されています。

ファイルシステム表示の状態コード

f および m のオペレータ表示には、ファイルシステムの状態コードが表示されます。状態コードは 11 位置形式で表示され、左 (位置 1) から右 (位置 11) の方向に読みます。

この項で説明する状態コードは、samu(1M) c、o、r、s、t、または v 表示には適用されません。o、r、s、および t 表示の状態コードについては、309 ページの「リムーバブルメディア装置表示の状態コード」を参照してください。v 表示の状態コードについては、303 ページの「(v) - 自動ライブラリカタログ表示」を参照してください。

表 C-37 に、各位置の有効な状態コードを示します。

表 C-37 ファイルシステム表示の状態コード

状態ビット	ファイルシステムにおける意味
m-----	ファイルシステムが現在マウントされています。
M-----	ファイルシステムがマウント中です。
-u-----	ファイルシステムがマウント解除中です。
--A-----	ファイルシステムデータがアーカイブ中です。
---R-----	ファイルシステムデータが解放中です。
----S-----	ファイルシステムデータが書き込み中です。
-----1-----	Sun StorageTek SAM ファイルシステム version 1。
-----2-----	Sun StorageTek SAM ファイルシステム version 2。
-----c-----	Sun StorageTek QFS 共有ファイルシステム。
-----W---	単一書き込み。
-----R--	複数読み取り。
-----r-	mr 装置。
-----d	md 装置。

オペレータ表示のデバイスの状態

c、m、o、r、s、および t のオペレータ表示には、装置の状態コードが表示されます。これらのコードは、装置の現在のアクセス状態を表します。表 C-38 に、有効な状態コードを示します。

表 C-38 オペレータ表示のデバイスの状態

デバイスの状態	説明
on	装置をアクセスに利用可能。一部の表示の場合、ready や notrdy がこの状態より優先されることがあります。
ro	装置は、読み込み専用アクセスだけに利用可能。一部の表示の場合、ready や notrdy がこの状態より優先されることがあります。

表 C-38 オペレータ表示のデバイスの状態 (続き)

デバイスの状態	説明
off	<p>装置をアクセスに利用できません。テープドライブと光磁気ディスクドライブの場合、装置が off 状態にある原因としては次のように考えられます。</p> <ul style="list-style-type: none"> • クリーニングが要求されたが、自動ライブラリにクリーニングカートリッジがありません。 • クリーニングカートリッジをドライブに読み込んだりドライブから取り出したりできません。 • 初期化の結果、ドライブがいっぱいであることが判明し、ドライブのクリアが失敗しました。 • システムがカートリッジをドライブから除去できませんでした。 • 回転立ち上げ時、入出力操作のためのドライブのオープン処理が失敗しました。 • 取り出しのためドライブの回転を停止しようとしたときに NOT READY 以外のエラーが出力されました。 • 回転立ち上げ時、ドライブの標準テープドライバのオープン処理が失敗しました。
down	装置は、メンテナンスアクセスだけに利用可能。
idle	デバイスは、新しい接続に利用できません。進行中の操作は、終了するまで続行されます。
ready	デバイスがオンであり、トランスポートに読み込まれているディスクやテープをアクセスに利用可能。
notrdy	デバイスはオンであるが、トランスポートにディスクもテープも存在していません。
unavail	装置にアクセスできず、自動操作に使用できません。装置が unavail 状態でも、load(1M) コマンドと unload(1M) コマンドを使用してメディアを移動できます。
nalloc	この装置への以後の割り当てを禁止する nalloc フラグが設定されています。詳細は、29 ページの「論理ユニット番号 (LUN) による割り当て制御」を参照してください。

samu(1M) の down、off、および on 装置状態コマンドを使用して、装置の状態を down、off、または on に変更できます。これらのコマンドは、任意の samu(1M) の表示から入力できますが、c、m、o、r、s、または t 表示から入力すると、その表示中に装置の状態の変化が表示されます。たとえば、p 表示から装置の状態を off に設定できますが、装置の新しい状態は表示に反映されません。

次の手順は、装置の状態を down から on に変更し、on から down に変更する入力です。

▼ ドライブの状態を down から on に変更する

1. ドライブと自動ライブラリ装置の状態を表示する samu(1M) の表示を起動します。
装置の状態を表示する samu(1M) の表示は、c、m、o、r、s、および t です。

2. 装置が down 状態であることを、表示を見て確認します。

3. :off を入力します。

デバイスを off にすると、すべての活動が停止し、次の手順で装置を正常に起動できます。

例:

```
Command :off eq
```

eq には、装置の装置番号を指定します。

4. :on を入力します。

例:

```
Command :on eq
```

eq には、装置の装置番号を指定します。

▼ ドライブの状態を on から down に変更する

1. ドライブと自動ライブラリ装置の状態を表示する samu(1M) の表示を起動します。
装置の状態を表示する samu(1M) の表示は、c、m、o、r、s、および t です。

2. 装置が on 状態であることを、表示を見て確認します。

3. :off を入力します。

デバイスを off にすると、すべての活動が停止し、次の手順で装置を正常に停止できます。

例:

```
Command :off eq
```

eq には、装置の装置番号を指定します。

4. `:down` を入力します。

例:

```
Command:down eq
```

`eq` には、装置の装置番号を指定します。

オペレータコマンド

このあとの項では、`samu(1M)` オペレータユーティリティーのコマンドインタフェースから入力できるオペレータコマンドについて説明します。コマンドは、任意の表示から入力できます。

次の種類のオペレータコマンドが使用可能です。

- 315 ページの「デバイスコマンド」
- 315 ページの「ファイルシステムコマンド: 入出力管理」
- 318 ページの「ファイルシステムコマンド: 直接入出力管理」
- 320 ページの「ファイルシステムコマンド: Sun StorageTek QFS 共有ファイルシステム」
- 321 ページの「ファイルシステムコマンド: そのほか」
- 325 ページの「そのほかのコマンド」

Solaris OS のコマンド行からオペレータコマンドを入力するには、`samcmd(1M)` コマンドの引数として使用してください。`samcmd(1M)` コマンドについては、`samcmd(1M)` のマニュアルページを参照してください。

このあとの項では、一連のホットキーではなくコマンドを入力していることを示す場合に、各 `samu(1M)` コマンドの前にコロン (`:`) が付きます。

デバイスコマンド

表 C-39 は、デバイスコマンドとその動作です。

表 C-39 デバイスコマンドのアクション

コマンド	動作
down	装置 <i>eq</i> に対する処理を終了します。
idle	装置 <i>eq</i> への新しい接続を禁止することによって、この装置へのアクセスを制限します。既存の処理は、終了するまで続行されます。
off	装置 <i>eq</i> を論理的に停止します。
on	装置 <i>eq</i> を論理的に起動します。
unavail	装置 <i>eq</i> を選択し、ファイルシステムで使用できないようにします。たとえば、障害回復時にファイルシステムを復元するためにメディアを読み込もうとしているときに、ドライブの状態を <i>unavail</i> にすると、Sun StorageTek SAM のソフトウェアがこのドライブを使用しないようにできます。
unload	指定のリムーバブルメディア装置 <i>eq</i> 用にマウントされているメディアを取り出します。マガジンデバイスの場合、 <i>unload</i> コマンドはマウントされているカートリッジを読み込み解除してマガジンを取り出します。
nalloc	この装置への以後の割り当てを禁止する <i>nalloc</i> フラグを装置に設定します。詳細は、29 ページの「論理ユニット番号 (LUN) による割り当て制御」を参照してください。
alloc	デバイスから <i>nalloc</i> フラグを削除します。 <i>nalloc</i> フラグは、この装置への以後の割り当てを禁止します。 <i>on</i> コマンドも、このフラグを削除します。詳細は、29 ページの「論理ユニット番号 (LUN) による割り当て制御」を参照してください。

これらすべてのコマンドは、次の形式で使用します。:*command eq*。*eq* には、装置の装置番号を指定します。

ファイルシステムコマンド: 入出力管理

次のコマンドで、入出力特性を動的に管理できます。

:*flush_behind eq value* コマンド

flush_behind コマンドは、最大 *flush_behind* 値を設定します。0 より大きな値を設定すると、変更されたページが逐次書き込みされるときに非同期でディスクに書き込まれ、Solaris カーネルレイヤーがページを空にしておくのに役立ちます。このオプションは、最大 *flush_behind* 値を設定します。

value には、 $0 \leq \text{value} \leq 8192$ となるキロバイト単位の整数を指定します。デフォルトは *value*=0 で、*flush_behind* は使用不可です。

eq には、ファイルシステムの装置番号を指定します。

:force_nfs_async *eq*

コマンドと :noforce_nfs_async *eq* コマンド

これらのコマンドで、データをディスクに同期書き込みするように NFS が要求した場合でも、サーバに書き込まれる NFS データをファイルシステムがキャッシュするかどうかを制御できます。*force_nfs_async* コマンドは、NFS データをキャッシュします。*noforce_nfs_async* コマンドは、デフォルトで、データをディスクに同期書き込みします。

force_nfs_async コマンドは、ファイルシステムが NFS サーバーにマウントされ、クライアントが *noac* NFS マウントオプション付きでマウントされている場合のみ有効です。NFS ファイルシステムのマウントについては、*mount_nfs(1M)* のマニュアルページを参照してください。

eq には、ファイルシステムの装置番号を指定します。



注意 - *force_nfs_async* オプションは、NFS プロトコル違反です。このコマンドを使用する場合は、注意が必要です。サーバーが停止した場合、データが失われることがあります。データは NFS サーバーにキャッシュされ、複数の NFS サーバーがある場合、すぐにはすべてのクライアントで参照できません。Sun StorageTek QFS 共有ファイルシステム内で、複数の NFS サーバーを有効にすることができます。Sun StorageTek QFS 共有ファイルシステムの詳細については、71 ページの「Sun StorageTek QFS 共有ファイルシステムの構成」を参照してください。

:readahead *eq contig* コマンド

readahead コマンドは、ファイルシステムが先読みできる最大バイト数を指定します。

eq には、ファイルシステムの装置番号を指定します。

contig には、1K バイトブロックの単位数を指定します。この値は、 $1 < \text{contig} < 8192$ となる整数である必要があります。指定した *contig* 値は、8K バイトの倍数に切り捨てられます。デフォルトの *contig* は 8 (131072 バイト) です。

たとえば、次のコマンドは、装置番号 3 として定義されているファイルシステムに対し、262,144 バイトの最大連続ブロックサイズを設定しています。

```
:readahead 3 256
```

この値は、`readahead` 指示を指定することによって、`samfs.cmd` ファイルで構成することもできます。詳細については、`samfs.cmd(4)` のマニュアルページを参照してください。

`:sw_raid eq` コマンドと `:nosw_raid eq` コマンド

これらのオプションは、ファイルシステムが後書きバッファを割り当てるかどうかを指定します。このファイルシステムで `Solstice DiskSuite™` などのパッケージのソフトウェア RAID 機能も使用する場合は、`sw_raid` を指定します。デフォルト設定は `nosw_raid` です。

`eq` には、ファイルシステムの装置番号を指定します。

`:writebehind eq contig` コマンド

`writebehind` コマンドは、ファイルシステムが後書きできる最大バイト数を指定します。

`eq` には、ファイルシステムの装置番号を指定します。

`contig` には、1K バイトブロックの単位数を指定します。この値は、 $1 < contig < 8192$ となる整数である必要があります。デフォルトの `contig` は 8 (131072 バイト) です。

たとえば、次のコマンドは、装置番号 50 として定義されているファイルシステムに対し、262,144 バイトの最大連続ブロックサイズを設定しています。

```
:writebehind 50 256
```

この値は、`writebehind` 指示を指定することによって、`samfs.cmd` ファイルで構成することもできます。詳細については、`samfs.cmd(4)` のマニュアルページを参照してください。

`:wr_throttle eq value` コマンド

`wr_throttle` コマンドは、1 ファイルに対する未処理の書き込みデータのバイト数を `value` キロバイトに設定します。

eq には、ファイルシステムの装置番号を指定します。

value には、キロバイト単位の整数を指定します。*value=0* の場合、制限がなくなります。デフォルトは 16384 です。

ファイルシステムコマンド: 直接入出力管理

この項のコマンドは、Sun StorageTek QFS のファイルシステムの入出力を制御します。これらによって、入出力サイズおよび履歴に基づき、ファイルの入出力の種類を個別に変更できます。*setfa(1)* コマンドなどでファイルに対して直接入出力を指定した場合、オプションは無視され、通常ファイルに対するすべての入出力が、可能であれば直接入出力になります。

これらのコマンドは、境界割り当てされた入出力と境界割り当てされない入出力の両方で使用できます。境界割り当てされた入出力は、ファイルオフセットが 512 バイト境界内にあり、入出力転送の長さが 512 バイト以上の場合に発生します。境界割り当てされない入出力は、ファイルオフセットが 512 バイト境界内になく、転送長が 512 バイト未満の場合に発生します。

入出力と入出力管理の詳細については、187 ページの「高度な機能」を参照してください。

`:dio_rd_form_min eq value`

コマンドと `:dio_wr_form_min eq value` コマンド

これらのコマンドは、境界割り当てされた入出力の下限を *value* の 1024 バイトブロックに設定します。*dio_rd_form_min* コマンドを使用して読み取り用の *value* を設定し、*dio_wr_form_min* コマンドを使用して書き込み用の *value* を設定します。

eq には、ファイルシステムの装置番号を指定します。

value には、下限に使用する 1024 バイトブロックの数を整数で指定します。デフォルトの場合、*value=256* です。*value=0* の場合、自動入出力切り換えは使用不可になります。

`:dio_rd_ill_min eq value`

コマンドと `:dio_wr_ill_min eq value` コマンド

これらのコマンドは、境界割り当てされない入出力の下限を *value* の 1024 バイトブロックに設定します。*dio_rd_ill_min* コマンドを使用して読み取り用の *value* を設定し、*dio_wr_ill_min* コマンドを使用して書き込み用の *value* を設定します。

eq には、ファイルシステムの装置番号を指定します。

value には、下限に使用する 1024 バイトブロックの数を整数で指定します。デフォルトの場合、*value*=256 です。*value*=0 の場合、自動入出力切り換えは使用不可になります。

`:dio_rd_consec eq value`

コマンドと `:dio_wr_consec eq value` コマンド

これらのコマンドは、指定した下限よりバッファサイズが大きいときに連続して発生することが許可される入出力転送数を *value* 回に設定します。

eq には、ファイルシステムの装置番号を指定します。

value には、指定した下限よりバッファサイズが大きいときの連続入出力転送数を指定します。指定した下限とは、境界割り当てされた読み取りの場合は `dio_rd_form_min`、境界割り当てされない読み取りの場合は `dio_rd_ill_min` の *value* です。デフォルトの場合は、*value*=0 で、これはデフォルトの直接読み取りが入出力サイズに基づいて発生しないことを意味します。

詳細については、次のコマンドまたはマウントパラメータを参照してください。

- 318 ページの「`:dio_rd_form_min eq value` コマンドと `:dio_wr_form_min eq value` コマンド」
- 318 ページの「`:dio_rd_ill_min eq value` コマンドと `:dio_wr_ill_min eq value` コマンド」

`:dio_szero eq` コマンドと `:nodio_szero eq` コマンド

これらのコマンドは、直接入出力のスパースのゼロ化マウントオプションを設定またはクリアします。

`dio_szero` オプションでは、直接入出力で書き込まれたスパースファイルの未初期化領域が、この領域にアクセスされたときにゼロになります。これにより、スパースファイルの動作がページ入出力と同じになります。デフォルトでは、直接入出力で書き込まれたスパースファイルの未初期化領域は、パフォーマンス上の理由からゼロにされません。デフォルトは `nodio_szero` です。

eq には、ファイルシステムの装置番号を指定します。

`:forcedirectio eq`

コマンドと `:noforcedirectio eq` コマンド

これらのコマンドによって、デフォルトの入出力モードとして直接入出力を使用するかどうかを制御できます。デフォルトのモードはバッファ入出力で、ページキャッシュを使用します。`forcedirectio` コマンドは、すべての転送で直接入出力を使用可能にします。`noforcedirectio` コマンドは、デフォルトであるバッファ入出力を使用可能にします。

`eq` には、ファイルシステムの装置番号を指定します。

直接入出力を指定すると、ユーザーのバッファとディスクの間でデータが直接転送されます。直接入出力は、ブロックが境界割り当てされた大容量の逐次入出力だけに使用してください。

入出力の詳細については、187 ページの「高度な機能」を参照してください。

ファイルシステムコマンド: Sun StorageTek QFS 共有ファイルシステム

次のファイルシステムコマンドは、Sun StorageTek QFS 共有ファイルシステムでのみサポートされます。

`:meta_timeo eq interval` コマンド

`metatimeo` コマンドは、Sun StorageTek QFS 共有ファイルシステムのメタデータキャッシュタイムアウト値を設定します。この機能の使用方法については、259 ページの「キャッシュした属性の保持: `meta_timeo=n` オプション」を参照してください。

`eq` には、ファイルシステムの装置番号を指定します。

`interval` には、間隔を秒単位で指定します。デフォルトの `interval` は 3 です。この時間が経過した場合、クライアントホストシステムは、メタデータ情報の新しいコピーをメタデータサーバーホストから取得します。

`:mh_write eq` コマンドと `:nomh_write eq` コマンド

これらのコマンドは、マルチホスト読み取りおよび書き込みを使用可能または使用不可にします。この機能の詳細については、256 ページの「複数ホストの読み取りと書き込みの有効化: `mh_write` オプション」を参照してください。

`eq` には、ファイルシステムの装置番号を指定します。

`:minallocsz eq value`

コマンドと `:maxallocsz eq value` コマンド

これらのコマンドは、ブロック割り当てサイズの最小値と最大値を設定します。

`eq` には、ファイルシステムの装置番号を指定します。

`value` と、この機能の詳細については、255 ページの「割り当てサイズの調整: `minallocsz=n` および `maxallocsz=n` オプション」を参照してください。

`:rdlease eq interval` コマンド、`:wrlease eq interval` コマンド、`:aplease eq interval` コマンド

これらのコマンドは、読み取り、書き込み、および追加リースに許可される時間を調整します。この機能の詳細については、255 ページの「Sun StorageTek QFS 共有ファイルシステムでのリースの使用: `rdlease=n`、`wrlease=n`、および `aplease=n` オプション」を参照してください。

`eq` には、ファイルシステムの装置番号を指定します。

`interval` には、秒単位の整数を指定します。3 つのリースはすべて、 $15 \leq interval \leq 600$ となる `interval` を指定できます。デフォルトの `interval` は 30 です。

ファイルシステムコマンド: そのほか

次のコマンドで、リース、割り当てサイズ、およびそのほかのさまざまなファイルシステム特性を制御できます。

`:abr eq` コマンドと `:noabr eq` コマンド

これらのコマンドは、ABR (Application Binary Recovery) マウントオプションを設定またはクリアします。

Sun StorageTek QFS AIO (非同期入出力) のある Oracle RAC 環境専用です。このマウントオプションは、ソフトウェアミラーの ABR を使用可能または使用不可にします。ABR をサポートする Solaris Volume Manager ミラー化ボリューム上に構築された Sun StorageTek QFS ファイルシステムだけが対象になります。

`eq` には、ファイルシステムの装置番号を指定します。

:dmr eq コマンドと :nodmr eq コマンド

これらのコマンドは、DMR (Direct Mirror Reads) マウントオプションを設定またはクリアします。

Sun StorageTek QFS AIO のある Oracle RAC (Real Application Cluster) 環境専用です。このマウントオプションは、ソフトウェアミラーの DMR を使用可能または使用不可にします。DMR をサポートする Solaris Volume Manager ミラー化ボリューム上に構築された Sun StorageTek QFS ファイルシステムだけが対象になります。

eq には、ファイルシステムの装置番号を指定します。

:invalid eq interval コマンド

invalid コマンドは、ファイルが変更されたあと *interval* 秒以上、ファイルシステムがキャッシュされた属性を保持することを指定します。このコマンドは、*reader* マウントオプションを使用してファイルシステムをマウントした場合にのみ使用できます。マウントオプションについては、*mount_samfs(1M)* のマニュアルページを参照してください。

eq には、ファイルシステムの装置番号を指定します。

interval には、ファイルが変更されたあと、属性を保持する秒数を指定します。たとえば、*interval=30* と指定します。このファイルシステムで *ls(1)* コマンドを発行すると、書き込み側ホストに新しくファイルを作成したあと 30 秒間は、この新規ファイルが表示されないことがあります。

:mm_stripe eq value コマンド

mm_stripe コマンドは、ファイルシステムのメタデータのストライプの幅を *value* の 16K バイトディスクアロケーションユニット (DAU) に設定します。

eq には、ファイルシステムの装置番号を指定します。

value には、0 または 1 を指定します。*value=1* の場合、ファイルシステムは 1 DAU のメタデータを 1 LUN に書き込んでから、別の LUN に切り替えます。これがデフォルトです。*value=0* の場合、メタデータはすべての利用可能なメタデータ LUN にラウンドロビン式で書き込まれます。

:qwrite eq コマンドと :noqwrite eq コマンド

qwrite コマンドと *noqwrite* コマンドは、同じファイルを異なるスレッドから同時に読み取りおよび書き込みできるかどうかを制御します。*qwrite* は、ファイルシステムユーザーが同じファイルに対して同時に複数のトランザクションを処理する場合

にのみ指定します。たとえば、データベースアプリケーションなどに役立ちます。qwrite 機能は、複数の要求をドライブレベルでキューイングすることで入出力パフォーマンスを向上させます。qwrite 指定は、ファイルシステムの NFS 読み取りまたは書き込みには使用できません。

デフォルト設定は noqwrite で、ファイルシステムは同じファイルに対して同時に読み取りおよび書き込みを実行できません。これは、UNIX vnode インタフェース標準で定義されているモードです。排他的アクセスが与えられるのは 1 つの書き込み側だけで、そのほかの書き込み側および読み取り側は待機させられます。

eq には、ファイルシステムの装置番号を指定します。

:refresh_at_eof eq

コマンドと :norefresh_at_eof eq コマンド

refresh_at_eof コマンドおよび norefresh_at_eof コマンドを、reader マウントオプション付きで複数読み取りファイルシステムにマウントされた Sun StorageTek QFS ホストに対する高速更新に使用できます。このオプションを使用すると、読み取りバッファがファイルの末尾を超えたときに、システムが現在のファイルサイズを再表示します。たとえば、書き込み側ホストシステムがファイルに追加し、読み取り側が -f オプション付きで tail(1) コマンドを発行しているときに、このコマンドを使用できます。デフォルトは norefresh_at_eof です。

eq には、ファイルシステムの装置番号を指定します。

:suid eq コマンドと :nosuid eq コマンド

suid コマンドと nosuid コマンドは、実行中のプログラムが自分自身の所有者 ID を自動的に変更できるかどうかを制御します。これらのマウントオプションの使用に関する詳細については、mount_ufs(1M) のマニュアルページの suid および nosuid マウントオプションの説明、および suid(2) のマニュアルページを参照してください。

eq には、ファイルシステムの装置番号を指定します。

:stripe eq value コマンド

stripe コマンドは、ファイルシステムのストライプの幅を value のディスクアロケーションユニット (DAU) に設定します。ストライプの幅は、value × DAU バイトが 1 つの LUN に書き込まれてから、次の LUN に切り換わることを指定します。sammkfs(1M) -a コマンドを使用して、初期化時にファイルシステムの DAU サイズを設定できます。

eq には、ファイルシステムの装置番号を指定します。

value には、 $0 < value < 255$ となる整数を指定します。*value*=0 の場合、ファイルは各スライスにラウンドロビン式で書き込まれます。ms 装置タイプのファイルシステム、およびストライプ化グループ (gXXX) コンポーネントがない ma 装置タイプのファイルシステムでのデフォルトの *value* は、次のとおりです。

- DAU < 128K バイトの場合、128K バイト/DAU
- DAU > 128K バイトの場合、1

デフォルトの場合、Sun StorageTek QFS 共有ファイルシステムでは *value*=0 です。

デフォルトの場合、ストライプ化グループ (gXXX) コンポーネントがある ma 装置タイプのファイルシステムでは *value*=0 です。

不一致のストライプ化グループが存在する場合は、システムが *value*=0 を設定しません。

ファイルシステムの種類の詳細については、7 ページの「設計の基本」および 31 ページの「ファイル システムの構成」を参照してください。

:sync_meta eq *value* コマンド

sync_meta コマンドは、メタデータが変更されるたびにディスクに書き込むかどうかを決定します。Sun StorageTek QFS 共有ファイルシステムでこのコマンドを使用する場合は、259 ページの「メタデータの書き込み頻度の指定: *sync_meta*=n オプション」も参照してください。

eq には、ファイルシステムの装置番号を指定します。

value には、次のように、0 または 1 を指定します。

- *value* が 0 の場合、メタデータは変更されたあと、バッファに保持されます。より高いパフォーマンスが要求される非共有の Sun StorageTek QFS ファイルシステムでは、*value* を 0 に設定します。この場合、システムはメタデータをディスクに書き込む前にバッファに保持し、遅延書き込みを実行します。これは、非共有ファイルシステムおよび複数読み取りファイルシステムとしてマウントされていないファイルシステムのデフォルトです。
- *value* が 1 の場合、メタデータは変更されるたびにディスクに書き込まれます。この場合、パフォーマンスは低下しますが、データの整合性は向上します。これは、複数読み取りファイルシステムまたは共有ファイルシステムとしてマウントされた Sun StorageTek QFS ファイルシステムのデフォルトです。Sun StorageTek QFS 共有ファイルシステムで、フェイルオーバー機能が要求されている場合は、*value* を 1 に設定する必要があります。

`:trace eq` コマンドと `:notrace eq` コマンド

`trace` コマンドは、ファイルシステムのトレースを使用可能にします。`notrace` コマンドは、トレースを使用不可にします。これらは、すべての操作に影響を与えるグローバル指示です。ファイルシステムのトレースの詳細については、`defaults.conf(4)` のマニュアルページを参照してください。

`eq` には、ファイルシステムの装置番号を指定します。

その他のコマンド

次のコマンドで、トレースの制御、ディスク装置へのアクセスのオープン、およびいくつかのその他の作業を実行できます。

`:clear vsn [index]` コマンド

`clear` コマンドは、指定した VSN をリムーバブルメディアマウント要求表示から消去します。詳細は、290 ページの「(p) - リムーバブルメディアの読み込み要求の表示」を参照してください。

`vsn` には、マウントするボリュームを指定します。VSN マウントを待機していたプロセスは、すべて中止されます。

`index` には、リムーバブルメディア表示での VSN の 10 進数の順番を指定します。

`:devlog eq [option]` コマンド

`devlog` コマンドは、1 つ以上のイベントをログに記録する対象として設定します。

`eq` には、装置の装置番号を指定します。

`option` には、1 つまたは複数のイベントタイプを指定します。指定できるイベントタイプは、次のとおりです。all、date、default、detail、err、event、label、mig、module、msg、none、retry、stage、syserr、time。これらのオプションについては、`defaults.conf(4)` のマニュアルページを参照してください。`option` を指定しない場合、指定された `eq` でログに記録する現在のイベントは変更されません。

:diskvols *volume* [+*flag* | -*flag*] コマンド

diskvols コマンドは、ディスクボリュームのディクショナリにフラグを設定またはクリアします。

volume にはディスクボリュームディクショナリにあるボリュームを指定します。

flag には D samu(1M) 表示の 5 つのフラグのいずれかを指定します。ディスクボリュームディクショナリおよびフラグについての詳細は、272 ページの「(D)-ディスクボリュームディクショナリ」または samu(1M) のマニュアルページを参照してください。

:dtrace コマンド

dtrace コマンドは、次のとおりです。

- :dtrace *daemon_name* on
- :dtrace *daemon_name* off
- :dtrace *daemon_name.variable value*

dtrace コマンドは、さまざまなトレースオプションを指定します。表 C-40 に、トレース制御コマンドの引数を示します。

表 C-40 トレースコマンドの引数

引数	説明
<i>daemon_name</i>	all キーワードまたはプロセス名を指定します。all キーワードを指定した場合、トレースコマンドはすべてのデーモンに適用されます。次のプロセス名のいずれか 1 つを指定した場合、トレースコマンドはそのプロセスだけに適用されます。sam-archiverd、sam-catservrd、sam-fsd、sam-rftd、sam-recycler、sam-sharefsd、および sam-stagerd。キーワード on または off をプロセス名のあとに指定できます。on または off を指定した場合、指定されているすべてのプロセスに対してトレースが起動または停止されます。

表 C-40 トレースコマンドの引数 (続き)

引数	説明
<i>variable value</i>	<p>さまざまな <i>variable</i> 引数と <i>value</i> 引数を指定できます。defaults.conf(4) のマニュアルページには、これらの引数に関する総合的な情報が掲載されています。次の <i>variable</i> と <i>value</i> の組み合わせのいずれか 1 つを指定します。</p> <ul style="list-style-type: none"> • <i>file value</i>。 <i>value</i> には、トレースファイルを書き込めるファイル名を指定します。フルパス名を指定できます。 • <i>options value</i>。 <i>value</i> には、空白文字で区切られたトレースオプションを指定します。 • <i>age value</i>。 <i>age</i> には、トレースファイルのローテーション経過時間を指定します。 注: この値を 2 分以下に設定しないでください。ローテーションが行われなくなります。 • <i>size value</i>。 <i>value</i> には、ローテーションを開始するトレースファイルのサイズを指定します。

:fs *fsname* コマンド

fs コマンドは、N 表示で表示されるようにファイルシステムを設定します。

fsname には、検査するファイルシステムの名前を指定します。

:mount *mntpt* コマンド

mount コマンドは、Sun StorageTek QFS のファイルシステムを選択します。*mntpt* には、ファイルシステムのマウントポイントを指定します。

:open *eq* コマンド

open コマンドは、指定のディスク装置にアクセスできるようにします。read コマンド、ディスクセクター表示 (s) またはファイルラベル表示 (F) を使用する前に、このコマンドを発行する必要があります。

eq には、装置の装置番号を指定します。

:read *addr* コマンド

read コマンドは、現在オープン状態であるディスク装置から指定のセクターを読み取ります。読み取りを行う前に、装置を開く必要があります。

addr には、16 進数のセクターアドレスを指定します。

:refresh *i* コマンド

refresh コマンドは、samu(1M) 画面を再表示する間隔の時間を決定します。

i には、時間を秒単位で指定します。

:snap [*filename*] コマンド

snap コマンドは、ウィンドウのスナップショットを *filename* に送ります。これは、表示情報を受け取るファイルの名前です。

samu(1M) ユーティリティーのすべての画面のスナップショットを取れるため、障害レポートに活用できます。新しいスナップショットは、スナップショットファイルに追加されます。デフォルトのファイルは、現在の作業ディレクトリに入っている snapshots です。このファイルを印刷したり、vi(1) を使用して検査したり、Sun のカスタマーサポート要員にファクシミリ送信したりできます。

:! *shell_command* コマンド

! コマンドにより、samu(1M) オペレータユーティリティーを終了しないまま、シェルコマンドを実行できます。

用語集

D

DAU ディスク割り当て単位ディスク割り当て単位 (Disk Allocation Unit)。オンライン記憶装置の基本単位。「ブロックサイズ」とも呼ばれます。

F

FDDI Fiber-Distributed Data Interface の略語。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 ファイル転送プロトコル (File Transfer Protocol)。TCP/IP ネットワークを通して 2 つのホスト間でファイルを転送するためのインターネットプロトコルです。

I

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に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 (.inodes)。i ノードは長さが 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。

L

LAN ローカルエリアネットワーク (Local Area Network)

LUN 論理ユニット番号 (Logical Unit Number)

M

mcf マスター構成ファイル (Master Configuration File)。ファイルシステム環境でのデバイス間の関係 (トポロジ) を定義した、初期化時に読み込まれるファイル。

N

NFS ネットワークファイルシステム (Network File System)。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、Sun の分散ファイルシステムです。

NIS Sun OS 4.0 以上の Network Information Service。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です。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。

R

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 遠隔手続き呼び出し。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。

S

samfsdump 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。UNIX の tar(1) ユーティリティと似ていますが、通常、ファイルデータのコピーは行いません。「samfsrestore」も参照。

samfsrestore 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」も参照。

SAM-QFS Sun StorageTek SAM ソフトウェアと Sun StorageTek QFS ファイルシステムを組み合わせた構成。SAM-QFS は、ストレージ管理ユーティリティとアーカイブ管理ユーティリティにおいて、ユーザーと管理者に高速な標準の UNIX ファイルシステムのインタフェースを提供します。SAM-QFS は、Sun StorageTek SAM コマンドセット内の多くのコマンド、および標準の UNIX ファイルシステムのコマンドを使用します。

SCSI 小型コンピュータシステムインタフェース (Small Computer System Interface)。ディスクドライブ、テープドライブ、自動ライブラリといった周辺装置に通常使用される、電気通信の仕様です。

**Sun SAM-Remote
クライアント** クライアントデーモンにいくつかの擬似デバイスが含まれ、専用のライブラリデバイスも持つことがある Sun StorageTek SAM システム。クライアントは、Sun 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。

**Sun SAM-Remote
サーバー** 全容量の Sun StorageTek SAM ストレージ管理サーバーと、Sun SAM-Remote クライアントが共有するライブラリを定義する Sun SAM-Remote サーバーデーモンの両方。

T

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。

V

fstab ファイル **fstab** 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、`/etc/fstab` ファイル内の指定を上書きし、`/etc/fstab` に指定されたマウントオプションは `fs.mnt` ファイル内の指定を上書きします。

VSN ボリュームシリアル名 (Volume Serial Name)。リムーバブルメディアカートリッジへのアーカイブでは、**VSN** は、ボリュームラベルに書き込まれる磁気テープと光磁気ディスクの論理識別子。ディスクキャッシュへのアーカイブでは、**VSN** はディスクアーカイブセットに対して一意です。

W

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。

あ

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。

アーカイブ記憶領域 アーカイブメディア上で作成されたファイルデータのコピー。

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媒体。ライブラリ内の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を、アーカイブメディアとして使用できます。また、別のシステム上のマウントポイントをアーカイブメディアとすることもできます。

アドレスサブル記憶領域 Sun StorageTek QFS または Sun StorageTek SAM 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記憶領域。

い

イーサネット ローカルエリアの packets 交換網のテクノロジー。当初は同軸ケーブルが使用されていましたが、現在では遮蔽より対線ケーブルが利用されています。イーサネットは、10M バイトまたは 100M バイト/秒の LAN です。

え

遠隔手続き呼び出し 「RPC」を参照。

お

オフサイト記憶装置 サーバーから離れた遠隔地にあつて災害時の障害回復に使用される記憶装置。
オフライン記憶装置 読み込み時にオペレータの介入を必要とする記憶装置。
オンライン記憶装置 いつでも利用可能な記憶装置 (ディスクキャッシュ記憶領域など)。

か

カートリッジ テープ、光ディスクなど、データを記録するための媒体を含む物体。「メディア」、「ボリューム」、または「媒体」と呼ぶこともあります。
カーネル 基本的なシステム機能を提供する、中央制御プログラム。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。
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。
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。
書き込み ニアラインファイルやオフ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すること。

- カタログ** 自動ライブラリにある VSN 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。「VSN」も参照。
- 監査 (完全)** カートリッジを読み込んでカートリッジの 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。「VSN」も参照。
- 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。

き

- 擬似デバイス** 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。
samsharefs(1M) -u コマンドの実行時に、この情報を更新します。
- 共有ライター/
共有リーダー** 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(1M) 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(1M) コマンドの -o writer オプションによって指定します。mount(1M) コマンドの詳細については、mount_samfs(1M) のマニュアルページを参照してください。

く

- 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。
- グローバル指示** 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、アーカイブ指示とリリーサ指示。

こ
小型コンピュータ
システムインタフェース

「SCSI」を参照。

し
事前割り当て
自動ライブラリ

ディスクキャッシュ上の隣接する領域をファイルの書き込み用として予約することです。事前割り当ては、サイズがゼロのファイルに対してだけ指定できます。詳細については、setfa(1)のマニュアルページを参照してください。

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読み込んだり取り外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。

す
スーパーブロック
ストライプ化

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。

複数のファイルをインター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。SAM-QFS ファイルシステムには、2種類のストライプ化があります。SAM-QFS 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 に設定すると使用不可になります。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。

ストライプ化グループ	mcf ファイルで 1 つ以上の gXXX デバイスとして定義された、ファイルシステムにあるデバイスの集合。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、必ず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。
ストレージスロット	カートリッジがドライブ内で未使用のときに格納される、自動ライブラリ内の場所。
ストレージファミリセット	1 つのディスクファミリ装置にまとめられている、ディスクのセット。

せ

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 モジュールと別のマシン上の TCP モジュールをつなぎます。
----	--

た

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。
------	---

ち

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、Sun StorageTek SAM ソフトウェアから直接制御されます。

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(1) 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。

つ

強い制限値 ディスク割り当てにおいて、ユーザーが超えてはいけないファイルシステム資源 (ブロックと i ノード) の最大値です。

て

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ブ化 アクセスパフォーマンスの向上と全体的な記憶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。「ストライブ化」も参照。

ディスクバッファ Sun SAM-Remote ソフトウェアの構成において、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に使用するサーバーシステム上のバッファ。

ディスク容量しきい値 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。

ディスク割り当て単位 「DAU」を参照。

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。

デバイスログ機能 デバイスの問題の解析に使用するデバイス固有のエラー情報を提供する、構成可能な機能。

と

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。

な

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。

に

ニアライン記憶装置 アクセスする前に無人マウントが必要なリムーバブルメディア記憶装置。通常、ニアライン記憶装置はオンライン記憶装置よりも安価ですが、アクセスに多少時間がかかります。

ね

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。Sun StorageTek SAM のファイルシステムは、自動ライブラリ用に設計された Sun StorageTek SAM 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。

は

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。

バックアップ記憶装置 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せん。

ふ

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 固有指示	<code>archiver.cmd</code> ファイル内のグローバル指示のあとのアーカイバ指示とリリーサ指示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指示は、次の <code>fs =</code> 指示行まで、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とした指示が複数存在する場合、ファイルシステム固有指示がグローバル指示より優先されます。
ファミリセット	自動ライブラリ内の複数のディスクやドライブなどの、独立した物理デバイスのグループによって表される記憶装置。「ストレージファミリセット」も参照。
ファミリデバイスセット	「ファミリセット」を参照。
ブロックサイズ	「DAU」を参照。
ブロック割り当てマップ	ディスク上の記憶装置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。

ほ

ホストファイル	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。Sun StorageTek QFS 共有ファイルシステムとしてファイルシステムを初期化する場合は、ファイルシステムを作成する前に <code>/etc/opt/SUNWsamfs/hosts.fs-name</code> というホストファイルを作成します。 <code>sammkfs(1M)</code> コマンドは、ファイルシステムの作成時にこのホストファイルを使用します。 <code>samsharefs(1M)</code> コマンドを使用して、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。
ボリューム	データ共有のための、カートリッジ上の名前付きの領域。カートリッジは、1 つまたは複数のボリュームで構成されます。両面カートリッジには、片面に 1 つずつ、合計 2 つのボリュームが含まれています。
ボリューム オーバーフロー	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。

ま

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。

み

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。

め

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、セグメントファイル、およびセグメントファイルの索引に関する情報で構成されます。

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、mcf ファイルにおいて、ma ファイルシステム内の mm デバイスとして宣言されます。

メディア テープカートリッジまたは光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。

ゆ

猶予期間 ディスク割り当てにおいて、弱い制限値に達したユーザーがファイルの作成や記憶領域の割り当てを行うことのできる時間。

よ

弱い制限値

ディスク割り当てにおいて、ユーザーが一時的に超えてもよい最大ファイルシステム資源 (ブロックと i ノード) の限界値です。弱い制限値を超えると、タイマーが起動します。指定した時間の間弱い制限値を超えると、弱い制限値未満のレベルにファイルシステムの使用を削減しないかぎり、システム資源の割り当ては行われません。

ら

ライブラリ

「自動ライブラリ」を参照。

ライブラリカタログ

「カタログ」を参照。

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」と「ストライプ化」も参照。

り

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、Sun StorageTek SAM のユーティリティ。

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解放することで、利用可能なディスクキャッシュ空間を増やす、Sun StorageTek SAM のコンポーネント。リリーサは、オンラインディスク記憶装置の容量を、上限値と下限値に合わせて自動的に調整します。

ろ

ローカルファイルシステム

Sun 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 記憶装置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、自動ライブラリの一部分。「トランスポート」とも呼ばれます。

わ

割り当て ユーザーが使用できるシステム資源の容量。

索引

数字

0 割り当て, 107

64 ビットアドレッシング, 3

A

abr samu(1M) コマンド, 321

abr マウントオプション, 127

additional parameters フィールド, 40

alloc コマンド, 29

API ルーチン, 6, 52

aplease

 samu(1M) コマンド, 321

 マウントオプション, 255

archive(1) コマンド, 51

archiver(1M) コマンド, 240, 242

B

bg マウントオプション, 254

C

chmod 4000 WORM トリガー, 198

chmod -w WORM トリガー, 200

clear samu(1M) コマンド, 325

D

DAU, 6

 md 装置, 11

 mr または gXXXX, 11

 概要, 8

 シングル割り当て方式, 10

 設定, 9

 デュアル割り当て方式, 10

 割り当ておよび DAU, 101

dd(1M) コマンド, 103, 105

defaults.conf ファイル

 トレースの有効化, 247

 変更の反映, 56

def_retention マウントオプション, 260

device state フィールド, 40

devlog samu(1M) コマンド, 325

DID サポート, 126

dio_rd_consec

 samu(1M) コマンド, 319

 マウントオプション, 222

dio_rd_form_min samu(1M) コマンド, 318

dio_rd_ill_min samu(1M) コマンド, 318

dio_szero samu(1M) コマンド, 319

dio_wr_consec

 samu(1M) コマンド, 319

 マウントオプション, 222

dio_wr_form samu(1M) コマンド, 318

dio_wr_ill_min samu(1M) コマンド, 318

directio(3C) 関数呼び出し, 3, 222
diskvols.conf ファイル
 samu(1M) の表示, 272
diskvols samu(1M) コマンド, 326
dmr samu(1M) コマンド, 322
dmr マウントオプション, 127
down samu(1M) コマンド, 315
dtrace samu(1M) コマンド, 326

E

EDOM エラー, 231
EDQUOT エラー, 101
EFI ラベル
 Linux クライアントで, 251
emul_lite マウントオプション (WORM), 195
ENOCSEI エラー, 231
equipment
 identifier フィールド, 37
 ordinal フィールド, 38
 type フィールド, 38
EVMS (Enterprise Volume Management System), 251

F

File System Manager
 アカウントの作成, 32
 概要, 31
 管理対象サーバーの追加, 35
 権限レベルの割り当て, 33
 使用, 32
 セッションタイムアウトの設定, 36
flush_behind
 samu(1M) コマンド, 315
 マウントオプション, 228
forcedirectio
 samu(1M) コマンド, 320
 マウントオプション, 222
force_nfs_async samu(1M) コマンド, 316

fsck(1M) コマンド 「samfsck(1M) コマンド」も参照, 3
fsmadm(1M) コマンド, 35
fs samu(1M) コマンド, 327

G

gXXX 装置, 39

H

HA-NFS
 構成, 153, 168
 無効化, 181, 182, 184
HA-SAM, 186
HASStoragePlus リソースタイプ, 146, 154
hostname(1M) コマンド, 246
hosts.fsname.local ファイル, 77, 90, 91
hosts ファイル, 76, 82
 Sun Cluster での, 142
 ホストの削除, 87
 例, 92

I

idle samu(1M) コマンド, 315
.inodes ファイル, 3
invalid samu(1M) コマンド, 322
IPMP
 アダプタ, 153
 検証テスト, 151
i ノード
 attributes, 51
 samu(1M) の表示, 277
 ファイルの内容, 7
i ノードハッシュテーブル、調整, 229

L

Linux クライアント, 5

トラブルシューティング, 249
マウントオプション, 252
よくある質問, 251
ログファイル, 249
LogicalHost リソースタイプ, 182
ls(1) コマンド 「sls(1) コマンド」 も参照, 53
LUN
LUN による割り当て制御, 29
LUN による割り当て制御, 29
LVM (Logical Volume Management), 251

M

maxallocsz
samu(1M) コマンド, 321
マウントオプション, 255
maxphys パラメータ, 223
ma ファイルシステム, 9, 39
mcf ファイル
/dev/dsk のエントリ, 38
/dev/rmt エントリ, 38
/dev/samst エントリ, 38
additional parameters フィールド, 40
device state フィールド, 40
equipment identifier フィールド, 37
equipment ordinal フィールド, 38
equipment type フィールド, 38
shared キーワード, 75
Sun Cluster 上の共有ファイルシステム, 139
Sun Cluster での, 165
Sun Cluster の非共有ファイルシステム, 149
Sun Cluster の例, 158
エラーの確認, 69
確認, 234
共有ファイルシステム, 88
更新, 88
構成, 36
ファイルシステムサイズの増加, 66
ファミリーセット (Family Set) フィールド, 39
フィールド, 37
変更の反映, 56
例, 41, 44

mdadm、Linux クライアントでの使用禁止, 252
md 装置, 39
metadb(1M) コマンド, 156
metaset コマンド, 159
meta_timeo
samu(1M) コマンド, 320
マウントオプション, 259
mh_write
samu(1M) コマンド, 320
マウントオプション, 256
minallocsz
samu(1M) コマンド, 321
マウントオプション, 255
min_pool マウントオプション, 258
mm_stripe samu(1M) コマンド, 322
mm 装置, 39
mount(1M) コマンド, 42, 62
共有ファイルシステム, 72
失敗, 235
トラブルシューティング, 98, 233, 240
mount samu(1M) コマンド, 327
mr 装置, 39
ms ファイルシステム, 9, 39

N

nalloc コマンド, 29
Network Appliance インタフェースのエミュレーション, 195
NFS
共有ポイント, 154
リソースグループ, 161
nhino パラメータ, 229, 230
ninodes パラメータ, 229, 230
noabr samu(1M) コマンド, 321
nodev キーワード, 38
nodio_szero samu(1M) コマンド, 319
nodmr samu(1M) コマンド, 322
noforcedirectio samu(1M) コマンド, 320
noforce_nfs_async samu(1M) コマンド, 316

nomh_write samu(1M) コマンド, 320
noquota マウントオプション, 119
noqwrite samu(1M) コマンド, 322
norefresh_at_eof samu(1M) コマンド, 323
noseduid samu(1M) コマンド, 323
nosw_raid samu(1M) コマンド, 317
notrace samu(1M) コマンド, 325
nsswitch.conf ファイル, 151
nstreams マウントオプション, 258

O

OCR (Oracle Cluster Registry) ファイル, 145
OCR ファイル, 145
off samu(1M) コマンド, 315
on samu(1M) コマンド, 315
open samu(1M) コマンド, 327
Oracle RAC, 140
 Sun Cluster での, 144

Q

qfsrestore(1M) コマンド, 69
quota.group ファイル, 110
quota マウントオプション, 100
Qwrite, 226
qwrite
 samu(1M) コマンド, 322
 マウントオプション, 227

R

rdlease
 samu(1M) コマンド, 321
 マウントオプション, 255
readahead
 samu(1M) コマンド, 316
 マウントオプション, 225
reader マウントオプション, 210
read samu(1M) コマンド, 327

Red Hat Linux、サポートされるディストリビュー
 ション, 5
refresh_at_eof samu(1M) コマンド, 323
refresh samu(1M) コマンド, 328
release(1) コマンド, 52
retry マウントオプション, 254

S

sam_archive(3) API ルーチン, 52
Samba, 215
sam-catservd デーモン, 188
samchaid(1M) コマンド, 107
samcmd(1M) aridle コマンド, 58
samcmd(1M) コマンド, 62
samd(1M) コマンド, 59, 67, 69, 76
samfs.cmd ファイル, 42, 62, 222
 Sun Cluster での, 140
 例, 64
samfsck(1M) コマンド, 40, 75, 79, 232, 233
samfsconfig(1M) コマンド, 239
 出力例, 88
sam-fsd(1M) コマンド, 67, 69, 190
sam-fsd デーモン, 83, 188
samfsinfo(1M) コマンド, 43
 出力例, 237
samfsrestore(1M) コマンド, 69
sam-genericd デーモン, 188
samgrowfs(1M) コマンド, 66, 67, 220
sammkfs(1M) コマンド, 28, 40, 43, 69
 トラブルシューティング, 233
samquota(1M) コマンド, 102
 引数, 114
sam_release(3) API ルーチン, 52
SAM-Remote、samu(1M) の表示, 295
sam-rpcd デーモン, 188
sam_segment(3) API ルーチン, 52
sam_setfa(3) API ルーチン, 52, 222
sam-sharedfsd デーモン, 188
samsharefs(1M) コマンド, 60, 76, 81

- 出力例, 238
- sam-sharefsd デーモン, 84, 188
 - トラブルシューティング, 98
- sam-sharefsd のトレースログ, 246
- sam_ssum(3) API ルーチン, 52
- sam_stage(3) API ルーチン, 52
- !samu(1M) コマンド, 328
- samu(1M) によるメディアの読み込み解除, 315
- samu(1M) のデバイスコマンド, 315
- samu(1M) ユーティリティ
 - mount コマンド, 62
 - nalloc コマンドおよび alloc コマンド, 29
 - インタフェース, 263
 - オペレータコマンド, 314
 - オペレータ表示, 264
 - 概要, 261
 - 共有ファイルシステムコマンド, 320
 - 状態コード, 309
 - 直接入出力コマンド, 318
 - ディスプレイ制御キー, 263
 - デバイスコマンド, 315
 - ファイルシステムコマンド, 315, 321
 - ヘルプ表示, 276
 - 呼び出し, 261
- samunhold(1M) コマンド, 219
- SANergy, 214
 - インストール, 216
 - ファイルホールド, 219
- SANERGY_SMBPATH 環境変数, 215
- SAN-QFS ファイルシステム
 - clients, 216
 - Sun StorageTek QFS 共有ファイルシステムとの比較, 221
 - 概要, 212
 - 拡大, 220
 - 制限事項, 214
 - トラブルシューティング, 219
 - マウント解除, 217
 - メタデータコントローラ, 215
 - 有効化, 215
- scconf コマンド, 165, 169
- scdidadm(1M) コマンド, 129, 164
- scrgadm(1M) コマンド, 134, 150, 154, 160
- SCSI ディスク、転送, 224
- SCSI の状態、samu(1M) の表示, 299
- scswitch(1M) コマンド, 134, 144, 162
- sd.conf ファイル, 224
- sd_max_xfer_size 定義, 224
- segment(1) コマンド, 52, 209
- setfa(1) コマンド, 3, 28, 52, 192, 222
- setfa ファイル属性, 52
- setuid samu(1M) コマンド, 323
- sfind(1M) コマンド
 - WORM オプション, 208
- share(1M) コマンド, 215
- shared マウントオプション, 254
- sls(1) コマンド, 51, 53
 - WORM オプション, 207
 - 出力の説明, 53
- snap samu(1M) コマンド, 328
- Solstice DiskSuite/Solaris ボリュームマネージャー, 155
- squota(1) コマンド, 102
- ssd.conf ファイル, 224
- ssd_max_xfer_size 定義, 224
- ssum(1) コマンド, 51
- stage(1) コマンド, 52
- stage_flush_behind
 - マウントオプション, 228
- Sun Cluster
 - DID サポート, 126
 - HA-NFS の構成, 153, 161, 168
 - HA-NFS の無効化, 181, 182, 184
 - HA-SAM 構成, 186
 - HAStoragePlus リソースタイプ, 146, 154
 - IPMP 検証テスト, 151, 161, 167
 - NFS リソースグループ, 154
 - Oracle RAC, 140, 144
 - Resource Group Manager, 146
 - Solstice DiskSuite/Solaris ボリュームマネージャーで, 155
 - Sun Cluster 用 Solaris Volume Manager, 127
 - VERITAS Clustered Volume マネージャーで

- , 162
- 概要, 124
- 共有ファイルシステム, 135
- 共有ファイルシステムの例, 136
- 共有または非共有ファイルシステム, 126
- クラスタの外部にあるクライアント, 168
- クラスタの外部にあるクライアントの構成, 168
- 構成の検証, 142
- 構成の変更, 180
- 構成例, 134
- 制限事項, 125
- ネットワークネームサービス, 151, 161, 167
- 非共有ファイルシステムの構成, 146
- 非共有ファイルシステムの例, 147
- プライベートインターコネクト名, 141
- マニュアル, 124
- メタデータサーバーのリソース, 136
- Sun Cluster 用 Solaris Volume Manager, 127
- SUNW.qfs リソースタイプ, 134, 144
- SuSE Linux、サポートされるディストリビューション, 5
- sw_raid samu(1M) コマンド, 317
- sync_meta
 - samu(1M) コマンド, 324
 - マウントオプション, 259
- syslog.conf ファイル, 232

T

- touch ユーティリティー、WORM-FS 保存期間の設定, 205
- trace_rotate(1M) コマンド, 190
- trace samu(1M) コマンド, 325

U

- unavail samu(1M) コマンド, 315
- unfuse コマンド, 218
- unload samu(1M) コマンド, 315
- umount(1M) コマンド, 73
 - 共有ファイルシステム, 86

- unshare(1M) コマンド
 - SAN-QFS ファイルシステム, 218

V

- VERITAS Clustered Volume マネージャー, 162
- vfstab ファイル, 42, 62
 - shared キーワード, 75
 - Sun Cluster での, 150, 160
 - 共有ファイルシステム, 85
 - フィールド, 63
 - 保存期間、WORM-FS, 204
- VFS、vnode インタフェース
- vnode インタフェース
 - 「VFS」を参照
- vrddiskadd コマンド, 164
- vrddmpadm コマンド, 164
- vxdg コマンド, 164
- VxVM, 162
 - 構成, 163
 - 装置, 185
 - パッケージ, 163

W

- worm_capable マウントオプション, 195, 260
- worm_emul マウントオプション, 195
- WORM-FS
 - 「lite」オプション, 197
 - touch を使用した保存期間の設定, 205
 - WORM ファイル属性の表示, 207
 - WORM ファイルの作成, 197, 202
 - エミュレーションモード, 195
 - オプション, 202
 - 概要, 194
 - 検索, 208
 - デフォルト保存期間の設定, 204
 - マウントオプション, 196, 260
- worm_lite マウントオプション, 195
- WORM 保存, 55
- writebehind

samu(1M) コマンド, 317
マウントオプション, 224
writer マウントオプション, 210
wrlease
samu(1M) コマンド, 321
マウントオプション, 255
wr_throttle
samu(1M) コマンド, 317
マウントオプション, 227

X

xntpd(1M) デーモン, 246

あ

アーカイバ
samu(1M) の表示, 265
アーカイブ
コピー, 54
アップグレード
ディスク, 68

う

うるう年、WORM 保存期間, 204

え

エミュレーションモード、WORM, 195

お

オペレータユーティリティ 「samu(1M)」を参照

か

カーネル統計情報、samu(1M) の表示, 280
外部ストレージ、samu(1M) の表示, 283
書き込みスロットル, 227

管理セット、割り当て, 99, 107

き

キャッシュ

入出力 「入出力」を参照、ページ

強制的マウント解除, 65

共有ファイルシステム, 5

aplease マウントオプション, 255

hosts ファイル, 76, 82

Linux クライアント, 5

maxallocsz マウントオプション, 255

mcf ファイル, 88

meta_timeo マウントオプション, 259

mh_write マウントオプション, 256

minallocsz マウントオプション, 255

min_pool マウントオプション, 258

nodev キーワード, 38

sam-fsd デーモン, 83

sam-sharefsd デーモン, 84

SAN-QFS 共有ファイルシステムとの比較, 221

stripe マウントオプション, 259

Sun Cluster での, 135, 141

Sun Cluster での例, 136

Sun Cluster の外部にあるクライアント, 168

sync_meta マウントオプション, 259

wrlease マウントオプション, 255

キャッシュした属性, 259

クライアントサーバー通信, 98

クライアントホストの削除, 86

クライアントホストの追加, 81

構成, 71

コマンド、samu(1M) の使用, 320

デーモン, 188

入出力, 98

非共有から共有への変換, 74

非共有への変換, 78

ファイルロック, 5

フェイルオーバー, 93, 95, 96

マウント, 72

マウントオプション, 253

マウント解除, 86

マウントポイント, 4

- メタデータサーバーの変更, 93, 96
- リース, 255
- ローカルホストファイル, 77, 83
- 共有ファイルシステムのスレッド, 258
- 共有ホストファイル
 - 変更, 59, 60
- 共有ライター/共有リーダー, 210
- 共用メモリー、samu(1M) の表示, 279, 282, 284

く

- クライアントサーバー通信, 98
- クラスタの外部にあるクライアント, 168

こ

- 構成
 - mcf ファイルの作成, 36
 - samu(1M) の表示, 267
 - ディスク使用, 8
 - ファイル割り当て, 15
 - 変更、伝達, 55
- 構成変更の伝達, 55

さ

- サービス、samu(1M) の表示, 293

し

- システムファイル, 223
- 自動ライブラリ
 - samu(1M) の表示, 290, 303
- 状態コード
 - samu(1M) による表示, 309, 310
- シングル割り当て方式, 10

す

- スーパブロック、version 1 および 2, 11, 43

- ステージャー
 - samu(1M) の保留書き込み待ち行列の表示, 307
 - 状態、samu(1M) の表示, 286, 300

- ストライプ
 - samu(1M) コマンド, 323
 - マウントオプション, 226, 259

- ストライプ化
 - 概要, 18
- ストライプ化グループ, 6, 21, 39
 - デバイス, 194
 - ファイル属性, 52
 - 不一致, 24
 - 例, 47

- ストライプ化割り当て, 18, 39, 259
 - ストライプ幅, 12, 193, 226
 - ユーザー指定, 193
 - 例, 46

- ストライプ幅
 - データディスク, 12
 - メタデータディスク, 14

せ

- セクターデータ、samu(1M) の表示, 297

そ

- 装置テーブル、samu(1M) の表示, 302
- 装置の状態、samu(1M) の表示, 295
- 装置の状態コード、samu(1M) による表示, 311

た

- 高可用性 SAM 構成, 186

つ

- 強い制限値、割り当て, 101

て

ディスク

追加またはアップグレード, 68

ディスクキャッシュ

超過, 209

追加, 66

ディスクボリュームディクショナリ

samu(1M) の表示, 272

ディスク割り当て単位、「DAU」を参照

ディレクトリ属性, 192

ディレクトリ名参照キャッシュ (DNLC), 7, 98

データ整合, 11

テープドライブ

samu(1M) の表示, 298

デーモン

sam-catservrd, 188

sam-fsd, 83, 188

sam-genericd, 188

sam-rpcd, 188

sam-sharedfsd, 188

sam-sharefsd, 84, 188

samu(1M) の表示, 270

xntpd(1M), 246

概要, 188

共有ファイルシステム, 188

トレース, 189

トレース、samu(1M) コマンド, 326

デュアル割り当て方式, 10

と

トレースファイル, 189

イベント, 189

切り換え, 190

ディレクトリ, 189

トレースログファイル, 246

に

入出力

切り換え, 222

調整, 221

直接, 2, 221, 222

直接入出力 samu(1M) コマンド, 318

直接入出力ファイル属性, 52

ページ, 2, 224, 225

ね

ネットワークネームサービス, 151

は

パーティション (追加, 変更, 削除), 68

ハッシュテーブル, 230

バッファ入出力

「入出力」を参照、ページ

ひ

光磁気ディスクの状態

samu(1M) の表示, 288

光磁気ディスクラベル、samu(1M) の表示, 275

ふ

ファイル

大容量ファイル転送, 223

大容量ファイルの格納, 209

ファイルシステム

ma, 39

ms, 39

ms と ma, 9

samu(1M) の表示, 273

回復, 3

機能, 2

基本操作, 51

キャッシュの超過, 209

共有、「共有ファイルシステム」を参照

共有への変換, 74

コマンド、samu(1M) の使用, 315, 321

再作成, 68

サポートされるファイル数, 3

修復, 233

- 状態コード、samu(1M)による表示, 310
- 使用率、samu(1M)の表示, 281
- 設計, 7
- 直接入出力コマンド、samu(1M)の使用, 318
- ディスクキャッシュの追加, 66
- 名前の変更, 67
- バックアップ, 68
- パラメータ、samu(1M)の表示, 287
- ファイル情報, 53
- ファイル割り当て, 15
- ボリューム管理, 2
- マウント, 62
- マウント解除, 65
- メタデータ, 4
- 容量, 3
 - 割り当て、「割り当て」を参照
- ファイルシステムの回復, 3
- ファイルシステムの修復, 233
- ファイルシステムの名前の変更, 67
- ファイル属性, 7, 51, 192
- ファイルのバックアップ, 68
- ファイル保存, 55
- ファイル領域の事前割り当て, 6, 192
- ファイル割り当て
 - ストライブ化, 15, 18
 - ストライブ化グループ, 21
 - 不一致のストライブ化グループ, 24
 - 方法, 193
 - ラウンドロビン, 15, 16
 - 領域の事前割り当て, 192
- ファミリーセット (Family Set) フィールド, 39
- プロセス、「デーモン」も参照
- ブロック割り当て、SAN-QFS ファイルシステム, 220

へ

- ページ入出力「入出力」を参照
- ヘルプ
 - samu(1M)の表示, 276

ほ

- 保存, 55
 - touchを使用した期間の設定, 205
 - 期間、WORM-FS, 202
 - デフォルト期間、WORM-FS, 204
 - マウントオプション, 260
- 保留書き込み待ち行列、samu(1M)の表示, 307
- ボリューム管理, 2

ま

- マウントオプション
 - abr, 127
 - bg, 254
 - dmr, 127
 - flush_behind, 228
 - Linuxクライアント, 252
 - meta_timeo, 259
 - mh_write, 256
 - minallocsz と maxallocsz, 255
 - min_pool, 258
 - rdlease、wrlease、および aplease, 255
 - retry, 254
 - shared, 254
 - stage_flush_behind, 228
 - sync_meta, 259
 - worm_capable と def_retention, 260
 - WORM-FS, 196
 - wr_throttle, 227
 - 共有ファイルシステム, 253
 - ストライブ, 259
 - 割り当て, 100
- マウント解除
 - SAN-QFS ファイルシステム, 217
 - ファイルシステム, 65
- マウントパラメータ, 62
- マウントポイント、確認, 235

む

- 無限割り当て, 107

め

- メタデータ, 7
 - mcf 内の装置, 39
 - 概要, 4
 - サーバー, 4
 - 内容, 7
 - 分離, 7
 - 割り当て, 15
- メタデータサーバー、「共有ファイルシステム」を参照
- メッセージファイル, 66, 232
- メモリー、samu(1M) の表示, 269

よ

- 弱い制限値、割り当て, 101

ら

- ライブラリカタログ、samu(1M) の表示, 303
- ラウンドロビン式割り当て, 16, 39
 - ユーザー指定, 193
 - 例, 45

り

- リース, 255
- リムーバブルメディア
 - samu(1M) の表示, 290
 - 状態、samu(1M) の表示, 293
 - 状態コード、samu(1M), 309

ろ

- ローカルホストファイル, 77, 83
 - 作成, 90
- ログファイル
 - Linux クライアント, 249
 - sam-sharefsd のトレースログ, 246, 249
 - ファイルシステムのトラブルシューティング, 232

わ

- 割り当て
 - 0, 107
 - nalloc コマンドおよび alloc コマンド, 29
 - SAN-QFS でのブロック割り当て, 220
 - 概要, 99
 - 管理セット, 99, 107
 - グループ割り当ての表示, 118
 - 検査, 111
 - 構成, 103, 104
 - コマンド, 102
 - 資源の割り当ての禁止, 117
 - 修正, 121
 - 強い制限値, 101
 - ディスクブロックとファイル割り当て, 101
 - デフォルト, 108
 - 取り出し, 113, 119
 - 変更, 113
 - 無限, 107
 - 有効化, 102
 - 猶予期間, 114, 115
 - 弱い制限値, 101
 - 割り当て値の変更, 109
 - 割り当てでの DAU, 101
 - 割り当てファイル, 100
 - 割り当てを有効にする, 109
- 割り当てサイズの調整, 255
- 割り当ての削除, 119
- 割り当ての猶予期間, 114, 115

