



# Sun StorEdge™ QFS, Sun StorEdge™ SAM-FS ファイルシステム管理マニュアル

---

リリース 4.2

Sun Microsystems, Inc.  
[www.sun.com](http://www.sun.com)

Part No. 817-7383-10  
2004 年 10 月, Revision A

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

Copyright 2004 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, および StorEdge 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 の商標または登録商標です。

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 StorEdge QFS and Sun StorEdge SAM-FS File System Administration Guide</i> Part No: 817-7721-10 Revision A
-----	--



# 目次

---

はじめに	xxi
マニュアルの構成	xxii
UNIX コマンド	xxiii
シェルプロンプトについて	xxiii
書体と記号について	xxiv
関連マニュアル	xxv
Sun のオンラインマニュアル	xxv
サン以外の Web サイト	xxvi
Sun の技術サポート	xxvi
ライセンス	xxvii
診断	xxvii
インストールのサポート	xxvii
コメントをお寄せください	xxvii
1. 概要	1
共通機能	2
vnode インタフェース	2
拡張ボリューム管理	2
ページ入出力と直接入出力のサポート	3
ファイル領域の事前割り当て	3

アプリケーションプログラミングインタフェースルーチン	3
大容量	4
ファイルシステムの高速回復	4
調整可能なディスク割り当て単位	5
ファイルシステムの違い	5
メタデータストレージ	5
複数ストライプ化グループのサポート	5
SAM の相互運用性	6
Sun Cluster の相互運用性	6
Sun StorEdge QFS 共有ファイルシステムのサポート	6
2. ファイルシステム設計	9
設計の基本	9
i ノードファイルとファイルの特徴	10
ファイル属性とファイル状態	11
ユーザー指定ファイル属性	12
システム指定のファイル状態	14
ファイル情報の表示	15
アーカイブのコピーの行の説明	16
検査合計行の説明	17
ディスク割り当て単位とストライプ幅の指定	18
DAU 設定とファイルシステムジオメトリ	19
デュアル割り当て方式	19
シングル割り当て方式	20
割り当て方式のまとめ	22
データディスクでのストライプ幅	23
Sun StorEdge SAM-FS のストライプ幅	23
Sun StorEdge QFS のストライプ幅 - ストライプ化グループを使用しない	24

Sun StorEdge QFS のストライプ幅 - ストライプ化グループを使用	25
Sun StorEdge QFS のデータ整合	25
メタデータディスクでのストライプ幅	26
ファイル割り当て方式	26
メタデータ割り当て	26
ラウンドロビン式割り当て	27
ストライプ化割り当て	29
ストライプ化グループ (Sun StorEdge QFS ファイルシステムのみ)	33
不一致のストライプ化グループ (Sun StorEdge QFS ファイルシステムのみ)	36
例	36
3. ボリューム管理	41
mcf ファイルの作成	42
Equipment Identifier フィールド	43
Equipment Ordinal フィールド	44
Equipment Type フィールド	44
Family Set フィールド	45
Device State フィールド	46
Additional Parameters フィールド	46
mcf ファイルの例	46
Sun StorEdge SAM-FS ボリューム管理の例	46
Sun StorEdge QFS および Sun SAM-QFS ボリューム管理の例	47
例 1	48
例 2	48
例 3	49
ファイル設定、オプション、指示の相互関係	50
ファイルシステムの初期化	50
例	51

## 構成の例 52

- ▼ Sun StorEdge QFS ラウンドロビン式ディスクの構成の作成 52
- ▼ Sun StorEdge SAM-FS ラウンドロビン式ディスクの構成の作成 54
- ▼ Sun StorEdge QFS ストライプ化ディスクの構成の作成 55
- ▼ Sun StorEdge SAM-FS ストライプ化ディスクの構成の作成 56
- ▼ Sun StorEdge QFS ストライプ化グループの構成の作成 58

## 4. ファイルシステムの操作 61

### ファイルシステムの初期化 62

### システムに対する構成ファイルの変更の伝達 62

- ▼ `mcf(4)` または `defaults.conf(4)` の情報を Sun StorEdge QFS 環境で変更する 63
- ▼ `mcf(4)` または `defaults.conf(4)` ファイルシステム情報を Sun StorEdge SAM-FS または Sun SAM-QFS 環境で変更する 64
- ▼ `mcf(4)` または `defaults.conf(4)` リムーバブルメディアドライブ情報を変更する 65
- ▼ `archiver.cmd(4)` または `stager.cmd(4)` の情報を変更する 67
- ▼ マウントされたファイルシステムの共有ホストファイル情報を変更する 67
- ▼ マウント解除されたファイルシステムの共有ホストファイル情報を変更する 68

### ファイルシステムのマウント 70

`mount(1M)` コマンド 70

`/etc/vfstab` ファイル 71

`samfs.cmd` ファイル 72

### ファイルシステムのマウント解除 74

- ▼ スタンドアロン Sun StorEdge SAM-FS および Sun SAM-QFS ファイルシステムをマウント解除する 74
- ▼ Sun StorEdge QFS および Sun SAM-QFS 共有ファイルシステムをマウント解除する 75

### 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 76

▼ ファイルシステムの検査	76
▼ ファイルシステムの修復	77
アップグレードのための情報の保持	78
例 1	79
例 2	82
例 3	82
ハードウェア装置のアップグレードの準備	83
ファイルシステムへのディスクキャッシュの追加	84
▼ ファイルシステムへのディスクキャッシュの追加	84
ファイルシステムのディスクの交換	86
▼ ファイルシステムのバックアップと再作成	86
ホストシステムのアップグレード	89
Solaris OS のアップグレード	90
▼ Sun StorEdge SAM-FS または Sun SAM-QFS 環境で Solaris OS をアップグレードする	90
▼ Sun StorEdge QFS 環境で Solaris OS をアップグレードする	93
5. Sun StorEdge QFS 共有ファイルシステム	97
概要	98
Sun StorEdge QFS 共有ファイルシステムの構成	98
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	99
▼ 非共有ファイルシステムを共有ファイルシステムに変換する	99
▼ 非共有ファイルシステムを各クライアントで共有に変換する	101
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	102
▼ 共有ファイルシステムを各クライアントで非共有に変換する	102
▼ 共有ファイルシステムをサーバーで非共有に変換する	103
Sun StorEdge QFS 共有ファイルシステムのマウントとマウント解除	105
▼ Sun StorEdge QFS 共有ファイルシステムをマウントする	105
▼ Sun StorEdge QFS 共有ファイルシステムをマウント解除する	105

- クライアントホストの追加と削除 106
  - ▼ クライアントホストを追加する 106
  - ▼ クライアントホストを削除する 117
- Sun StorEdge QFS 環境でのメタデータサーバーの変更 119
  - ▼ 使用可能なメタデータサーバーを変更する 120
  - ▼ 使用不可のメタデータサーバーを変更する 120
- デーモン 122
- Sun StorEdge QFS 共有ファイルシステムのマウントオプション 123
  - バックグラウンドでのマウント: `bg` オプション 123
  - ファイルシステムマウントの再試行: `retry` オプション 124
  - Sun StorEdge QFS 共有ファイルシステムの宣言: `shared` オプション 124
  - 割り当てサイズの調整: `minallocsz=n` および `maxallocsz=n` オプション 124
  - Sun StorEdge 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 125
  - 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 126
  - 並行スレッド数の設定: `nstreams=n` オプション 127
  - キャッシュした属性の保持: `meta_timeo=n` オプション 128
  - ストライプ化割り当ての指定: `stripe` オプション 128
  - メタデータの書き込み頻度の指定: `sync_meta=n` オプション 128
- Sun StorEdge QFS 共有ファイルシステムのマウントの意味 129
- Sun StorEdge QFS 共有ファイルシステムでのファイルロック 130
- パフォーマンスについて 130
- 失敗またはハングアップした `sammkfs(1M)` コマンドまたは `mount(1M)` コマンドの障害追跡 130
  - 失敗した `sammkfs(1M)` コマンドの回復 131
    - ▼ `mcf(4)` ファイルを確認し、`mcf(4)` ファイルの変更をシステムに伝達する 131
  - 失敗した `mount(1M)` コマンドの回復 131
    - ▼ ファイルシステムがマウント可能かどうかを確認する 132



- ▼ samfsinfo(1M) コマンドと samsharefs(1M) コマンドを使用する 134
- ▼ samfsconfig(1M) コマンドを使用する 136
- ハングアップした mount(1M) コマンドの回復 138
  - ▼ ネットワーク接続を確認する 138
  - ▼ 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 (省略可能) 141
  - ▼ 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 (省略可能) 143
  - ▼ sam-sharefsd トレースログを調査する (省略可能) 144
- 6. samu(1M) オペレータユーティリティの使用法 149
  - 概要 149
    - ▼ samu(1M) を起動する 150
    - ▼ samu(1M) 画面を表示する 150
    - ▼ samu(1M) を停止する 151
    - samu(1M) とのやりとり 151
    - 装置の入力 151
    - オンラインヘルプの表示 152
      - ▼ 表示画面からオンラインヘルプにアクセスする 152
  - オペレータ表示 152
    - (a) - アーカイバの状態表示 153
      - ナビゲーション 154
      - 表示例 154
      - フィールドの説明 155
    - (c) - デバイス構成表示 156
      - ナビゲーション 156
      - 表示例 156
      - フィールドの説明 157
    - (C) - メモリー表示 157

- 表示例 158
- (d) - デーモントレースコントロールの表示 159
  - 表示例 159
- (e) - ファイルシステムの表示 160
  - 表示例 160
  - フィールドの説明 161
- (F) - 光磁気ディスクラベル表示 162
- (h) - ヘルプ表示 162
  - ナビゲーション 163
  - 表示例 163
- (I) - i ノード表示 164
  - ナビゲーション 164
  - 表示例 165
- (J) - プレビュー共用メモリー表示 166
  - ナビゲーション 166
  - 表示例 166
- (K) - カーネル統計情報表示 167
  - ナビゲーション 167
  - 表示例 167
- (l) - ライセンスの表示 168
  - 表示例 168
- (L) - 共用メモリー表示 169
  - 表示例 169
- (m) - 外部ストレージの状態表示 170
  - 表示例 170
  - フィールドの説明 171
- (M) - 共用メモリー表示 172
  - ナビゲーション 172

表示例	172
(n) - 書き込みの状態表示	173
表示例	174
(N) - ファイルシステムパラメタ表示	174
ナビゲーション	174
表示例	175
(o) - 光磁気ディスクの状態表示	176
ナビゲーション	176
表示例	176
フィールドの説明	177
(p) - リムーバブルメディアの読み込み要求の表示	178
ナビゲーション	178
表示例	179
フィールドの説明	179
フラグ	180
(P) - アクティブサービス表示	180
ナビゲーション	180
表示例	181
(r) - リムーバブルメディアの状態表示	181
表示例	181
フィールドの説明	182
(R) - Sun SAM-Remote 情報表示	182
(s) - デバイスの状態表示	183
ナビゲーション	183
表示例	183
フィールドの説明	184
(S) - セクターデータ表示	184
ナビゲーション	184

(t) - テープドライブの状態表示	185
ナビゲーション	185
表示例	185
フィールドの説明	186
(T) - SCSI センスデータ表示	187
ナビゲーション	187
(u) - 書き込み待ち行列表示	187
ナビゲーション	188
表示例	188
フィールドの説明	189
(U) - 装置テーブル表示	189
ナビゲーション	190
表示例	190
(v) - 自動ライブラリカタログ表示	191
ナビゲーション	192
表示例	192
フィールドの説明	193
フラグ	194
(w) - 保留書き込み待ち行列の表示	195
ナビゲーション	195
表示例	196
フィールドの説明	196
オペレータ表示の状態コード	197
リムーバブルメディア装置表示の状態コード	197
ファイルシステム表示の状態コード	198
オペレータ表示のデバイスの状態	199
▼ ドライブの状態を down から on に変更する	200
▼ ドライブの状態を on から down に変更する	201

オペレータコマンド	201
デバイスコマンド	202
SAM コマンド - アーカイバ制御	203
:hwm_archive eq コマンドと :nohwm_archive eq コマンド	205
:thresh eq high low コマンド	205
SAM コマンド - リリーサ制御	206
:maxpartial eq value コマンド	206
:partial eq value コマンド	206
SAM コマンド - ステージャ制御	206
:partial_stage eq value コマンド	206
:stage_flush_behind eq value コマンド	207
:stage_n_window eq value コマンド	207
:stage_retries eq value コマンド	207
:stclear mt.vsn コマンド	208
:stidle コマンド	208
:strun コマンド	208
ファイルシステムコマンド - 入出力管理	208
:flush_behind eq value コマンド	208
:force_nfs_async eq	
コマンドと :noforce_nfs_async eq コマンド	209
:readahead eq contig コマンド	209
:sw_raid eq コマンドと :nosw_raid eq コマンド	210
:writebehind eq contig コマンド	210
:wr_throttle eq value コマンド	210
ファイルシステムコマンド - 直接入出力管理	211
:dio_rd_form_min eq value	
コマンドと :dio_wr_form_min eq value コマンド	211
:dio_rd_ill_min eq value	
コマンドと :dio_wr_ill_min eq value コマンド	211

:dio\_rd\_consec *eq value*  
     コマンドと :dio\_wr\_consec *eq value* コマンド 212

:forcedirectio *eq*  
     コマンドと :noforcedirectio *eq* コマンド 212

ファイルシステムコマンド - Sun StorEdge QFS 共有ファイルシステム 212

:meta\_timeo *eq interval* コマンド 213

:mhwrite *eq* コマンドと :nomh\_write *eq* コマンド 213

:minallocsz *eq value*  
     コマンドと :maxallocsz *eq value* コマンド 213

:rdlease *eq interval*, :wrlease *eq interval*,  
     コマンドと :aplease *eq interval* コマンド 213

ファイルシステムコマンド - その他 214

:invalid *eq interval* コマンド 214

:mm\_stripe *eq value* コマンド  
     (Sun StorEdge QFS ファイルシステムのみ) 214

:qwrite *eq*  
     コマンドと :noqwrite *eq* コマンド  
     (Sun StorEdge QFS ファイルシステムのみ) 214

:refresh\_at\_eof *eq*  
     コマンドと :norefresh\_at\_eof *eq* コマンド  
     (Sun StorEdge QFS ファイルシステムのみ) 215

:setuid *eq* コマンドと :nosetuid *eq* コマンド 215

:stripe *eq value* コマンド 215

:sync\_meta *eq value* コマンド 216

:trace *eq* コマンドと :notrace *eq* コマンド 217

自動ライブラリコマンド 217

:audit [-e] *eq* [ :slot [ :side ]] コマンド 217

:export *eq:slot* コマンドと :export *mt.vsn* コマンド 217

:import *eq* コマンド 218

:load *eq:slot* [ :side ] コマンドと :load *mt.vsn* コマンド 218

:priority *pid newpri* コマンド 218

その他のコマンド 219

:clear *vsn* [ *index* ] コマンド 219  
:devlog *eq* [ *option* ] コマンド 219  
:dtrace コマンド 219  
:fs *fsname* コマンド 220  
:mount *mntpt* コマンド 220  
:open *eq* コマンド 220  
:read *addr* コマンド 221  
:refresh *i* コマンド 221  
:snap [ *filename* ] コマンド 221  
:! *shell\_command* コマンド 221

## 7. ファイルシステム割り当て 223

### 概要 223

割り当てのタイプ、割り当てファイル、割り当てレコード 224

弱い制限値と強い制限値 225

割り当てとアーカイブメディア 226

ディスクブロックとファイル割り当て 226

### 割り当ての有効化 226

割り当て設定のガイドライン 227

▼ 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 228

▼ 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 230

▼ 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 233

無限割り当てを設定する 234

▼ 無限割り当てを設定する 234

デフォルトの割り当て値を有効にする 235

▼ 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 235

割り当てを有効にする 236

▼ 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 236

- ▼ 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 236

#### 割り当ての検査 238

- ▼ 超過した割り当てを検査する 239

#### 割り当ての変更と削除 241

- ▼ 猶予期間を変更する 241
- 猶予期間の期限を変更する 244
- ▼ 追加のファイルシステム資源の割り当てを禁止する 246
- ▼ ファイルシステム割り当てを削除する 249
- ▼ 割り当てを修正する 250

### 8. Sun Cluster 環境での Sun StorEdge QFS 253

#### 始める前に 254

#### 制限事項 255

#### Sun Cluster と Sun StorEdge QFS ソフトウェアの相互作用 256

##### 共有ファイルシステムでのデータアクセス 256

##### 非共有ファイルシステムでのデータアクセス 256

#### 構成例について 257

#### Sun Cluster での Sun StorEdge QFS 共有ファイルシステムの構成 258

##### メタデータサーバーのリソースの検討事項 258

##### 構成例 259

- ▼ Sun StorEdge QFS 共有ファイルシステムの作成を準備する 259
- ▼ Sun StorEdge 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 263
- ▼ 構成を検証する 265
- ▼ Oracle Real Application Cluster 用 Sun Cluster データサービスを構成する 266

#### Sun Cluster での非共有ファイルシステムの構成 268

##### 例 1 269

- ▼ 非共有 Sun StorEdge QFS ファイルシステムの作成を準備する 270



- ▼ ステップ 2 : Sun StorEdge QFS ファイルシステムの作成と Sun Cluster ノードの構成 271
- ▼ ネットワークネームサービスと IPMP 検証テストを構成する 273
- ▼ HA-NFS の構成と高可用性 Sun StorEdge QFS ファイルシステムの構成 276

#### 例 2 278

- ▼ Solstice DiskSuite/Solaris ボリュームマネージャソフトウェアを準備する 278
- ▼ Sun StorEdge QFS ファイルシステムの作成を準備する 279
- ▼ Sun StorEdge QFS ファイルシステムを作成し、Sun Cluster ノードを構成する 281
- ▼ ネットワークネームサービスと IPMP 検証テストを構成する 283
- ▼ HA-NFS を構成し、高可用性 Sun StorEdge QFS ファイルシステムを構成する 286

#### 例 3 287

- ▼ VxVM ソフトウェアを構成する 288
- ▼ Sun StorEdge QFS ファイルシステムの作成を準備する 290
- ▼ Sun StorEdge QFS ファイルシステムを作成し、Sun Cluster ノードを構成する 291
- ▼ 構成を検証する 292
- ▼ ネットワークネームサービスと IPMP 検証テストを構成する 293
- ▼ HA-NFS を構成し、高可用性 Sun StorEdge QFS ファイルシステムを構成する 295

#### Sun StorEdge QFS の構成の変更 296

- ▼ 共有ファイルシステムの構成を変更する 297
- ▼ 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 298
- ▼ Solaris ボリュームマネージャ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 299
- ▼ VxVM で制御されるボリュームを使用する Sun StorEdge QFS ファイルシステム上の HA-NFS を無効にする 301

## 9. 高度な機能 303

デーモン、プロセス、およびトレース	303
デーモンとプロセス	303
トレースファイル	305
トレースファイルの内容	305
トレースファイルの切り換え	306
トレース対象プロセスの判別	306
ファイル属性を設定するための setfa(1) コマンドの使用	307
ファイルやディレクトリのファイル属性の選択	308
ファイル領域の事前割り当て	308
ファイル割り当て方式とストライプ幅の選択	309
ストライプ化グループ装置の選択	309
大容量ファイルの格納	310
複数読み取りファイルシステム	310
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	312
事前確認	315
SAN-QFS ファイルシステムの使用開始	315
▼ 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	315
▼ クライアントで SAN-QFS ファイルシステムを有効にする	316
▼ クライアントで SANergy ソフトウェアをインストールする	317
SAN-QFS ファイルシステムのマウント解除	318
▼ 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	318
▼ 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	319
▼ Sun StorEdge 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	319
▼ Sun StorEdge 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	319
障害追跡: 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	320

- ▼ 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 321
- SAN-QFS ファイルシステムにおけるブロック割り当て 321
- 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 321
- samgrowfs(1M) による SAN-QFS ファイルシステムの拡張 321
- SAN-QFS 共有ファイルシステムと Sun StorEdge QFS 共有ファイルシステムの比較 322
- 入出力パフォーマンス 322
  - ページ入出力 323
  - 直接入出力 323
  - 入出力切り換え 323
- 大容量ファイル転送パフォーマンスの向上 324
  - ▼ 大容量ファイル転送パフォーマンスを向上する 324
  - Qwrite 328
  - 書き込みスロットルの設定 329
  - 遅延フラッシュ率の設定 329
  - iノードの数とiノードハッシュテーブルの調整 330
    - ninodes パラメタ 330
    - nhino パラメタ 331
    - ninodes パラメタと nhino パラメタの設定が必要な場合 332
- 用語集 333
- 索引 349



# はじめに

---

このマニュアルでは、Sun StorEdge QFS および Sun StorEdge SAM-FS 4.2 リリースに含まれているファイルシステムソフトウェアについて説明します。ソフトウェア製品と含まれているファイルシステムは次のとおりです。

- Sun StorEdge SAM-FS ファイルシステム。Sun StorEdge SAM-FS 環境には、汎用ファイルシステムとストレージアーカイブマネージャ SAM が含まれています。Sun StorEdge SAM-FS 環境のファイルシステムを使用すると、デバイス対応速度で自動ライブラリにデータをアーカイブできます。また、ディスクアーカイブというプロセスを使用して、別のファイルシステムのファイルにデータをアーカイブすることもできます。Sun StorEdge SAM-FS 環境のファイルシステムは、完成度の高いファイルシステムです。ユーザーに対して標準ファイルシステムインタフェースが表示され、すべてのファイルが主ディスクストレージに存在するかのよう読み取りと書き込みを行えます。
- Sun StorEdge QFS ファイルシステム。Sun StorEdge QFS ファイルシステムは、スタンドアロンファイルシステムとして使用することも、ストレージアーカイブマネージャ SAM と組み合わせて使用することもできます。SAM と組み合わせた場合は、*Sun SAM-QFS* と呼ばれます。Sun StorEdge QFS は、Sun StorEdge SAM-FS ファイルシステムの機能の大部分を共有します。ただし、Sun StorEdge QFS ファイルシステムは、高パフォーマンスを得るように設計されているため、Sun StorEdge SAM-FS 環境でサポートされるよりも多くの機能が含まれます。

---

**注** – Sun StorEdge SAM-FS ソフトウェアが提供するストレージおよびアーカイブ管理機能を利用しながら Sun StorEdge QFS ファイルシステムを実行する場合は、Sun StorEdge QFS および Sun StorEdge SAM-FS ソフトウェア両方のライセンスを購入することができます。こうしたシステムは、「Sun SAM-QFS」と呼ばれます。

紛らわしくなることを避けるために、このマニュアルでは、必要がない限り、Sun SAM-QFS 構成という用語を使用していません。このマニュアルのストレージおよびアーカイブ管理についての説明では、Sun StorEdge SAM-FS という用語は Sun SAM-QFS 構成も意味します。同様に、ファイルシステムの設計および機能に関する説明でも、Sun StorEdge QFS という用語は Sun SAM-QFS 構成も意味します。

---

Sun StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムは技術的には似ていますが、このマニュアルでは、必要に応じて相違点を注記します。

Sun StorEdge QFS および Sun StorEdge SAM-FS 4.2 リリースは、次の Sun Solaris™ オペレーティングシステム (OS) プラットフォームレベルでサポートされます。

表 P-1 最低限の Sun Solaris OS プラットフォームレベル

製品	最低限のプラットフォームレベル
Sun StorEdge QFS	Solaris 8 07/01
	Solaris 9 04/03
Sun StorEdge SAM-FS	Solaris 8 07/01
	Solaris 9 04/03
Sun Cluster 環境 Sun StorEdge QFS	Solaris 8 02/02
	Solaris 9 04/03

このマニュアルは、Sun StorEdge QFS および Sun StorEdge SAM-FS ソフトウェアのインストールと構成、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、アカウントの作成、システムバックアップの実行、その他 Solaris に関する基本的なシステム管理作業をはじめとする、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、ファイルシステム設計の情報を説明しています。
- 第 3 章 では、ボリューム管理情報を説明しています。
- 第 4 章 では、Sun StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムでさまざまな作業を行う方法を説明しています。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、およびその他のシステム管理アクティビティが含まれます。
- 第 5 章 では、Sun StorEdge QFS 共有ファイルシステムの使用方法を説明しています。
- 第 6 章 では、samu(1M) オペレータユーティリティの使用方法を説明しています。
- 第 7 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 8 章 では、Sun Cluster 環境における Sun StorEdge QFS の動作について説明しています。

- 第9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などのその他の高度な事項について説明しています。

用語集は、このマニュアルとその他の Sun StorEdge QFS、Sun StorEdge SAM-FS のマニュアルで使用されている用語の定義集です。

---

## 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の基本的な UNIX® コマンドと操作手順に関する説明はありません。これらについては、以下を参照してください。

- ご使用のシステムに付属のソフトウェアマニュアル
- 下記にある Solaris™ オペレーティング環境のマニュアル  
<http://docs.sun.com>

---

## シェルプロンプトについて

表 P-2 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-2 シェルプロンプト

シェル	プロンプト
UNIX の C シェル	マシン名%
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-3 にこのマニュアルで使用する書体と記号を示します。

表 P-3 書体と記号

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.
<b>AaBbCc123</b>	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。	% <b>su</b> Password:
AaBbCc123 またはゴシック	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。 rm <b>ファイル名</b> と入力します。
『 』	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。
\	枠で囲まれたコード例で、テキストがページ行幅をこえる場合に、継続を示します。	% <b>grep</b> `^#define \ XV_VERSION_STRING`

\* 使用しているブラウザにより、これら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 StorEdge QFS および Sun StorEdge SAM-FS 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 1 つです。表 P-4 は、これらの製品に関するリリース 4.2 のマニュアルセット一式の内容を示しています。

表 P-4 関連マニュアル

タイトル	Part No.
Sun SAM-Remote 管理者マニュアル	816-7835-10
Sun QFS, Sun SAM-FS, Sun SAM-QFS 障害回復マニュアル	816-7678-10
Sun StorEdge QFS, Sun StorEdge SAM-FS ファイルシステム管理マニュアル	817-7383-10
Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き	817-7394-10
Sun StorEdge SAM-FS ストレージ/アーカイブ管理マニュアル	817-7388-10
Sun StorEdge QFS, Sun StorEdge SAM-FS ご使用にあたって	817-7398-10

---

## Sun のオンラインマニュアル

Sun StorEdge QFS および Sun StorEdge SAM-FS ソフトウェアのディストリビューションには、これら製品のマニュアルの PDF ファイルが含まれています。これらの PDF ファイルは、以下のサイトで見ることができます。

- Sun の Network Storage ドキュメント Web サイト。

このウェブサイトには、数多くのストレージソフトウェア製品のマニュアルが用意されています。

- a. このウェブサイトには、次の URL でアクセスできます。

[www.sun.com/products-n-solutions/hardware/docs/Software/Storage\\_Software](http://www.sun.com/products-n-solutions/hardware/docs/Software/Storage_Software)  
Storage Software ページが表示されます。

- b. 次のリストの該当するリンクをクリックします。

- Sun StorEdge QFS Software
- Sun StorEdge SAM-FS Software

- docs.sun.com.

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

- a. このウェブサイトには、次の URL でアクセスできます。

docs.sun.com

docs.sun.com のページが表示されます。

- b. サーチボックスで次の項目を検索し、目的の製品のマニュアルを見つけます。

- Sun StorEdge QFS
- Sun StorEdge SAM-FS

---

## サン以外の Web サイト

このマニュアルには、サン以外の団体/個人の Web サイトに関する情報が含まれています。サンは、これらのサイトあるいはリソースに関する、あるいはこれらのサイト、リソースから利用可能であるコンテンツ、広告、製品、あるいは資料に関して一切の責任を負いません。サンは、これらのサイトあるいはリソースに関する、あるいはこれらのサイトから利用可能であるコンテンツ、製品、サービスのご利用あるいは信頼によって、あるいはそれに関連して発生するいかなる損害、損失、申し立てに対する一切の責任を負いません。

---

## Sun の技術サポート

このマニュアルに記載されていない技術的な問い合わせについては、次の URL にアクセスしてください。

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

---

## ライセンス

Sun StorEdge QFS および Sun StorEdge SAM-FS 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。

---

## 診断

Sun StorEdge QFS および Sun StorEdge SAM-FS ソフトウェアには、samexplorer(1M) スクリプトが付属しています。この診断スクリプトは、読者にとっても、サンのカスタマーサポート要員にとっても、非常に役に立ちます。このスクリプトは、サーバー構成の診断レポートを出力し、ログ情報を収集します。このスクリプトの詳細は、ソフトウェアインストール後に samexplorer(1M) のマニュアルページにアクセスしてください。

---

## 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

弊社では、マニュアルの改善に努力しており、お客様からのコメントおよびご忠告をお受けしております。コメントは下記よりお送りください。

<http://www.sun.com/hwdocs/feedback>

コメントには下記のタイトルと Part No. を記載してください。

Sun StorEdge QFS, Sun StorEdge SAM-FS ファイルシステム管理マニュアル, Part No. 817-7383-10



# 第1章

## 概要

Sun StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムは構成可能なファイルシステムで、ユーザーからは標準の UNIX ファイルシステムと同じインタフェースに見えます。表 1-1 に、これらのファイルシステムとストレージ・アーカイブ管理 (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 StorEdge QFS ファイルシステム	Sun StorEdge QFS スタンドアロンファイルシステム
Sun StorEdge SAM-FS ファイルシステム	標準ファイルシステムおよびストレージ・アーカイブ管理 (SAM) ユーティリティ
Sun SAM-QFS ファイルシステム	Sun StorEdge SAM-FS ソフトウェアに含まれる、ストレージ・アーカイブ管理ユーティリティと組み合わせられた Sun StorEdge QFS ファイルシステム

技術的に 2 つのファイルシステムは同じですが、違いもあります。この章では、これらのファイルシステム共通の機能の概要を示し、ファイルシステムごとに異なる機能や各ファイルシステムで使用できるコマンドについても説明します。この章の節は次のとおりです。

- 2 ページの「共通機能」
- 5 ページの「ファイルシステムの違い」

---

## 共通機能

Sun StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムでは、ユーザープログラムを変更したり、UNIX カーネルを変更する必要がありません。この節では、これらのファイルシステムの共通機能について説明します。

### vnode インタフェース

Sun StorEdge QFS および Sun StorEdge SAM-FS の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、これらのファイルシステムは標準の Solaris OS カーネルとともに作動し、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 StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムに存在するものも含めて、ファイルに対するすべての要求を受け取ります。Sun StorEdge QFS または Sun StorEdge SAM-FS のファイルと見なされるファイルの場合、カーネルは要求を該当するファイルシステムへ渡して、処理できるようにします。Sun StorEdge QFS および Sun StorEdge SAM-FS の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。

### 拡張ボリューム管理

Sun StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムは、ストライプ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 (`mc.f`) とマウントパラメタを使用して、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するようにします。これは、装置または装置の一部だけを指定できる多くの UNIX システムとは対照的です。Sun StorEdge QFS および Sun StorEdge SAM-FS 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。Sun StorEdge QFS または Sun StorEdge SAM-FS 環境のいずれか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してください。

Sun StorEdge QFS および Sun StorEdge SAM-FS 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します。Sun StorEdge QFS および Sun StorEdge SAM-FS ソフトウェアは、各ファイルシステムが格納された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 StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムは、次の異なる種類の入出力をサポートしています。: ページ (キャッシュ、バッファとも呼ばれます) および直接。これらの入出力タイプは、次のとおりです。

- 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。

## 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ルーチン

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。

## 大容量

Sun StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムは、長さが最大  $2^{63}$  バイトのファイルをサポートしています。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プ化できます。これは、Sun StorEdge QFS および Sun StorEdge SAM-FS 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からです。これは、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 (UFS) 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 4 テラ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 StorEdge SAM-FS ファイルシステムでは、ファイル数は事前に制限されていません。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 StorEdge 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分かれています。実際には、メタデータ (mm) 装置によって、Sun StorEdge QFS ファイルシステム内のファイル数の上限が決まります。メタデータ装置を追加すると、最大のファイル数を増加できます。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、システム障害の後で、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 StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムでは、ファイルシステムによるディスクへの書き込みができなくなる障害の後でも、ほとんどの場合、ファイルシステムの検査は不要です (`sync(1M)` を使用)。また、これらの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、Sun StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムはテラバイト単位のサイズの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。



## 調整可能なディスク割り当て単位

ディスク割り当て単位 (DAU) はオンライン記憶領域の基本単位です。Sun StorEdge QFS ファイルシステムには、調整可能な DAU が含まれています。これは、物理ディスクストレージの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、4 キロバイトの倍数で調整可能です。

---

## ファイルシステムの違い

Sun StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムは共通の機能が数多くあり、それらについては 2 ページの「共通機能」で説明しています。それに対して、この節ではその違いについて説明します。違いの 1 つはパフォーマンスです。Sun StorEdge QFS ファイルシステムでは、装置のディスク速度そのものに迫る速度と、ファイルシステムの管理の容易さが得られます。この後の各項では、ファイルシステムの違いについて、さらに説明します。

## 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。これは、Sun StorEdge SAM-FS ファイルシステムの場合です。

Sun StorEdge QFS ファイルシステムは、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 StorEdge 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 StorEdge QFS と Sun StorEdge SAM-FS の両方のファイルシステムは、どちらも 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。

## 複数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 StorEdge QFS ファイルシステムではストライプ化グループを定義できます。ストライプ化グループのディスクブロック割り当ては最適化できます。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。

## SAM の相互運用性

Sun StorEdge SAM-FS ファイルシステムは、ファイルシステム機能をストレージ・アーカイブ管理 (SAM) ユーティリティと組み合わせています。ユーザーは、磁気ディスクに対してファイルを直接読み書きできます。または、ファイルのアーカイブのコピーが主ディスクストレージにあるかのようにアクセスできます。

Sun StorEdge QFS ファイルシステムは、スタンドアロンファイルシステムとして使用することも、ストレージアーカイブマネージャー SAM と組み合わせて使用することもできます。Sun StorEdge QFS と Sun StorEdge SAM-FS の両方のライセンスを受けている場合は、「Sun SAM-QFS」と呼ばれます。

可能な場合、Sun StorEdge SAM-FS ソフトウェアは標準の Solaris OS ディスク装置ドライバとテープ装置ドライバを使用します。一部の自動ライブラリおよび光磁気装置など、Solaris OS で直接サポートされていない装置は、Sun Microsystems が提供する特殊なデバイスドライバが Sun StorEdge SAM-FS ソフトウェアパッケージに含まれています。

## Sun Cluster の相互運用性

Sun StorEdge 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。Sun StorEdge SAM-FS ファイルシステムは、Sun Cluster 環境ではサポートされません。

## Sun StorEdge QFS 共有ファイルシステムのサポート

共有ファイルシステムは、Sun StorEdge QFS 共有ファイルシステムまたは Sun SAM-QFS 共有ファイルシステムとして実装することができます。共有ファイルシステムを使用すると、複数の Sun Solaris 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を実装できます。

Sun StorEdge QFS 共有ファイルシステムは次のファイルタイプをサポートしていません。

- b - ブロックスペシャルファイル
- c - キャラクタペシャルファイル

■ p - FIFO (名前付きパイプ) スペシャルファイル

共有ファイルシステムは、セグメントファイルをサポートしません。Sun Cluster 環境では、Sun SAM-QFS 共有ファイルシステムは実装できません。

ファイルシステムの詳細については、97 ページの「Sun StorEdge QFS 共有ファイルシステム」を参照してください。



## 第2章

---

# ファイルシステム設計

---

ファイルシステムをうまく設計することは、迅速かつ中断されない情報へのアクセスを保証するために重要です。また、優れた設計は、ファイルシステムの回復にも不可欠となります。この章の内容は次のとおりです。

- 9 ページの「設計の基本」
- 10 ページの「i ノードファイルとファイルの特徴」
- 18 ページの「ディスク割り当て単位とストライプ幅の指定」
- 26 ページの「ファイル割り当て方式」

---

## 設計の基本

Sun StorEdge QFS および Sun StorEdge SAM-FS 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。これらの機能を最大限に活用するには、可能なら必ず複数のファイルシステムを作成してください。

Sun StorEdge QFS および Sun StorEdge SAM-FS 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ます。この場合、ディレクトリの先頭から末尾に向かって検索されます。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。数千のファイルがディレクトリに含まれる場合は、検索時間が非常に長くなる可能性があります。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 のディレクトリ内のファイル数を 10,000 個未満にする必要があります。

ディレクトリ名参照キャッシュ (DNLC) 機能とディレクトリ DNLC 機能は、どちらもファイルシステムのパフォーマンスを向上させます。ディレクトリ DNLC は、すべての Solaris オペレーティングシステム (OS) リリース 9 および Solaris OS リリース 8 アップデート以降で使用できます。

## 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の `.inodes` ファイルに格納されます。Sun StorEdge SAM-FS の `.inodes` ファイルは、ファイルデータと同じ物理デバイスに常駐し、ファイルデータによってインタリーブされます。対照的に、Sun StorEdge 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常駐します。

標準の Solaris オペレーティングシステム (OS) の i ノードのように、Sun StorEdge QFS または Sun StorEdge SAM-FS ファイルシステム i ノードは、次のファイルの POSIX 標準 i ノードの時間を含みます。ファイルアクセス、ファイル変更、i ノードが変更された時間。Sun StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムでは、作成時刻、属性変更時刻、および存在時間が追加されています。表 2-1 は、i ノードに記録される時間をまとめたものです。

表 2-1 .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 StorEdge QFS または Sun StorEdge SAM-FS ファイルシステム固有の属性が最後に変更された時刻。サンによる拡張機能。
creation	ファイルが作成された時刻。サンによる拡張機能。
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。サンによる拡張機能。

Sun StorEdge QFS および Sun StorEdge SAM-FS 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。この節では、これらの特徴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。`s1s(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。`s1s(1)` のオプションの詳細は、`s1s(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)`

## ユーザー指定ファイル属性

表 2-2 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 2-2 ユーザー指定ファイル属性

コマンド	定義	対象
archive -C	ファイルが並行アーカイブの対象としてマークされる。つまり、書き込み操作のためにファイルを開いている場合でも、アーカイブできるようになる。この属性は archive(1) コマンドを使用して設定できる。	Sun StorEdge SAM-FS
archive -n	ファイルがアーカイブされないようにマークされる。この属性は、スーパーユーザーが archive(1) コマンドを使用して設定できる。	Sun StorEdge SAM-FS
release -a	このファイルは、アーカイブのコピーが 1 つ作成されるとすぐに解放されるようにマークされる。この属性は、archiver.cmd ファイルから、または release(1) コマンドを使用して設定できる。	Sun StorEdge SAM-FS
release -n	ファイルが解放されないようにマークされる。この属性は、archiver.cmd ファイルから、またはスーパーユーザーが release(1) コマンドを使用して設定できる。	Sun StorEdge SAM-FS
release -p	ファイルが部分解放の対象としてマークされる。この属性は、archiver.cmd ファイルから、または release(1) コマンドを使用して設定できる。	Sun StorEdge SAM-FS
stage -a	ファイルが関連する書き込みの対象としてマークされる。この属性は、archiver.cmd ファイルから、または stage(1) コマンドを使用して設定できる。	Sun StorEdge SAM-FS
stage -n	ファイルが書き込まれないようにマークされる。これは、リムーバブルメディアカートリッジへの直接アクセスを意味する。この属性は、archiver.cmd ファイルから、またはスーパーユーザーが stage(1) コマンドを使用して設定できる。 Sun StorEdge QFS 共有ファイルシステムクライアントではサポートされない	Sun StorEdge SAM-FS
setfa -D	ファイルが直接入出力の対象としてマークされる	Sun StorEdge QFS Sun StorEdge SAM-FS
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る。	Sun StorEdge QFS



表 2-2 ユーザー指定ファイル属性 (続き)

コマンド	定義	対象
<code>setfa -sm</code>	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る。	Sun StorEdge QFS Sun StorEdge SAM-FS
<code>segment nm stage Ahead x</code>	ファイルがセグメント化の対象としてマークされる。 <i>nm</i> 指定によって、セグメントのサイズが <i>n</i> メガバイトであることが示される。 <code>stage Ahead x</code> 属性は、その数のセグメント ( <i>x</i> ) があらかじめ書き込まれることを示す。この属性は <code>segment(1)</code> コマンドを使用して設定できる。	Sun StorEdge SAM-FS

表 2-2 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、その後でそのディレクトリに作成されるファイルは、作成時に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。

ファイル属性に関する情報は、15 ページの「ファイル情報の表示」で説明する `sls(1)` コマンドを使用して収集できます。

## システム指定のファイル状態

表 2-3 に、ファイルシステムがファイルに対して設定するさまざまな状態を示します。これらの状態は i ノードに格納されます。

表 2-3 システム指定のファイル状態

属性	定義	対象
archdone	ファイルのアーカイブ要求が満たされたことを示す。アーカイバがファイルに対して実行する作業はなくなった。この属性はアーカイバが設定する。ユーザーは設定できない。archdone は、必ずしもファイルがアーカイブされたことを意味するものではないことに注意してください。	Sun StorEdge SAM-FS
damaged	ファイルが破壊されていることを示す。この属性は、ステージまたは <code>samfsrestore(1M)</code> コマンドで設定。undamage(1M) コマンドを使用して、破壊されていない状態にこの属性をリセットできる。この属性が <code>samfsrestore(1M)</code> ユーティリティによって設定されていた場合は、 <code>samfsdump(1M)</code> が取得されたときにはこのファイルのアーカイブコピーは存在していなかったことを意味する。破壊されていない状態にこの属性をリセットできるが、ファイルは回復不可能のままの場合がある。	Sun StorEdge SAM-FS
offline	ファイルデータが解放されたことを示す。この属性はリリーサが設定する。この属性は、 <code>release(1)</code> コマンドを使用して設定することもできる。	Sun StorEdge SAM-FS

ファイル状態に関する情報は、15 ページの「ファイル情報の表示」で説明する `sls(1)` コマンドを使用して収集できます。

## ファイル情報の表示

Sun StorEdge QFS および Sun StorEdge SAM-FS の `sls(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 2-1 に、ファイル `hgc2` の `i` ノード情報を表示した `sls(1)` コマンドの出力の詳細を示します。

コード例 2-1 Sun StorEdge SAM-FS 環境での `sls(1)` の出力

```
# sls -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 17:10:00
creation: Jun 13 17:08 residence: Jun 13 17:08
```

表 2-4 に、コード例 2-1 に示した `sls(1)` コマンドの各行の意味を示します。表 2-4 では、アーカイブに関する行が、Sun StorEdge QFS 環境の `sls(1)` 出力に表示されないことに注意してください。

表 2-4 `sls(1)` 出力の説明

行番号	先頭の文字	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、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、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る。ファイルシステム割り当ての詳細については、223 ページの「ファイルシステム割り当て」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る。
3	<code>archdone;</code>	ファイル固有のファイル属性。この行の詳細は、 <code>sls(1)</code> のマニュアルページを参照してください。

表 2-4 sls(1) 出力の説明 (続き)

行番号	先頭の文字	内容
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ない。この行の一般的な書式は次のとおり。</p> <p>segments <i>noffline o</i>, archdone <i>a</i>, damaged <i>d</i>;</p> <p>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す。この例では 3。</p> <p><i>offline o</i> は、オフラインのデータセグメントの数を示す。この例では、オフラインセグメントがない。</p> <p>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す。この例では 3。</p> <p>damaged <i>d</i> は、破壊されたセグメントの数を示す。この例では、破壊されたセグメントがない。</p>
5	copy 1:	<p>最初のアーカイブコピーの行。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 行表示する。詳細は、16 ページの「アーカイブのコピーの行の説明」を参照。</p>
6	copy 2:	<p>2 つめのアーカイブコピーの行。詳細は、16 ページの「アーカイブのコピーの行の説明」を参照。</p>
7	access:	<p>ファイルが最後にアクセスおよび変更された時刻</p>
8	changed:	<p>ファイルの内容が最後に変更され、ファイルの属性が最後に変更されてからの時間。</p>
9	creation:	<p>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。</p>

## アーカイブのコピーの行の説明

アーカイブのコピーの行のフィールドは次のとおりです。

- 最初のフィールドは、アーカイブのコピーの番号を示します。

- 2 番目のフィールドには、4 つのインジケータ (ハイフン (-) または文字) があります。表 2-5 に、インジケータが表す情報を、左から右の順で示します。

表 2-5 アーカイブのコピーの行のインジケータ

位置	意味
1	<p>エントリが期限切れかアクティブかを示す</p> <p>S は、アーカイブコピーが期限切れであることを示す。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ある。</p> <p>U は、コピーがアーカイブ解除されたことを示す。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。</p> <p>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す。</p>
2	<p>アーカイブのコピーが再アーカイブされるかどうかを示す</p> <p>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す。</p> <p>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す。</p>
3	未使用
4	<p>コピーが破壊されているかどうかを示す</p> <p>D は、アーカイブのコピーが破壊されていることを示す。破壊されたアーカイブのコピーは、書き込みの対象外。</p> <p>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す。このアーカイブのコピーは書き込みの対象外。</p>

- 3 番目のフィールドには、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が示されます。
- 4 番目のフィールドには、小数点 (.) で区切られた 2 つの 16 進数が含まれます。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。
- アーカイブのコピー行の 5 番目と 6 番目のフィールドは、アーカイブのコピーが常駐するメディアの種類とボリュームシリアル名 (VSN) を示します。

## 検査合計行の説明

ファイルに検査合計に関連する属性がある場合、s1s(1) コマンドによって checksum 行が返されます。これらの属性 (generate、use、または valid) は、ssum(1) コマンドを使用して設定できます。この行は、Sun StorEdge SAM-FS 環境の s1s(1) の出力に表示されます。検査合計行の書式は次のとおりです。

```
checksum: gen use val algo: 1
```

検査合計属性がファイルに設定されている場合、システムはこの行を表示します。意味は次のとおりです。

- generate 属性が設定されていない場合は、gen の代わりに no\_gen が表示されます。
- use 属性が設定されていない場合は、no\_use が表示されます。
- ファイルがアーカイブされ、検査合計が計算されている場合は、val が表示されます。
- ファイルがアーカイブされていない場合、または検査合計が計算されていない場合は、not\_val が表示されます。
- algo キーワードは、検査合計値の生成に使用されるアルゴリズムを指定する数値アルゴリズムインジケータの前に表示されます。

---

## ディスク割り当て単位とストライプ幅の指定

ディスク領域はブロック単位で割り当てられます。ブロックは、「ディスク割り当て単位 (DAU)」とも呼ばれます。これは、オンラインディスクストレージの基本単位です。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。

例：

Sun StorEdge SAM-FS ファイルシステムを使用しているとします。DAU を 16 キロバイトに設定し、stripe=0 と設定してストライプを無効にします。stripe=0 と設定したラウンドロビン式割り当てを使用します。ファイルは次の 2 つです。

- 1 番目のファイルは 15 キロバイトです。これは 1 つの DAU を占有します。ファイルデータが DAU の 15 キロバイト分を占め、残りの 1 キロバイトは使用されません。
- 2 番目のファイルは 20 キロバイトです。これは 2 つの DAU を占有します。ファイルデータは、1 番目の DAU の 16 キロバイトすべてを占め、2 番目の DAU の 4 キロバイト分を占めます。2 番目の DAU には、使用されない 12 キロバイトが含まれます。

sammkfs(1M) コマンドの -a *allocation\_unit* オプションは、DAU 設定を指定します。

ストライプ化割り当てを使用すると、ストライプ幅マウントオプションによって、1 回の入出カイベントで書き込まれる最大 DAU 数が決まります。この設定は、`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。`mount(1M)` コマンドを実行する前に、`sammkfs(1M)` コマンドを実行する必要があります。

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。

---

注 – 特記のない限り、このマニュアル内での Sun StorEdge QFS「ファイルシステム」に関する情報は、Sun SAM-QFS の構成にも適用されます。

---

## DAU 設定とファイルシステムジオメトリ

Sun StorEdge QFS および Sun StorEdge SAM-FS 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。これにより、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が軽減されます。大容量のファイル进行操作するアプリケーションでは、この機能は非常に有益です。読み取り/変更/書き込み操作の制御方法については、324 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固有の DAU 設定を使用できます。1 つのサーバー上で、複数のマウント済みのファイルシステムを、それぞれ異なる DAU 設定でアクティブにすることができます。DAU 設定は、`sammkfs(1M)` コマンドを使用してファイルシステムを作成するときに決まります。動的に変更することはできません。

使用可能な DAU 設定は、使用しているファイルシステムによって異なります。この項では、各ファイルシステムの DAU 設定について説明します。また、マスター構成 (mcf) ファイルの概念についても説明します。この ASCII ファイルはシステム構成時に作成します。このファイルは、Sun StorEdge QFS または Sun StorEdge SAM-FS 環境で使用される装置とファイルシステムを定義したものです。mcf ファイルについては次の項で述べていますが、詳細は 41 ページの「ボリューム管理」で説明します。

ファイル割り当て方式として、デュアル割り当て方式、シングル割り当て方式が利用可能です。これらの方式について次に説明します。

### デュアル割り当て方式

Sun StorEdge SAM-FS ファイルシステムは、mcf ファイルで装置タイプ `ms` と定義されます。Sun StorEdge SAM-FS ファイルシステムで使用可能なデバイスタイプは、タイプ `md` だけです。メタデータとファイルデータは、どちらも Sun StorEdge SAM-FS ファイルシステムでは `md` 装置に書き込まれます。デフォルトでは、`md` 装置の DAU は 16 キロバイトです。

Sun StorEdge QFS ファイルシステムは、mcf ファイルで装置タイプ `ma` と定義されます。メタデータは `mm` 装置に書き込まれます。データは `md`、`mr`、または `gXXX` 装置に書き込むことができます。

`md` 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 4 キロバイト、大きな割り当ては `DAU` になります。Sun StorEdge QFS 共有ファイルシステムのデフォルト `DAU` は 64 キロバイトです。Sun StorEdge QFS (非共有) ファイルシステムのデフォルト `DAU` は 16 キロバイトです。`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せん。

`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で割り当てます。ファイルを拡張するときにさらに領域が必要な場合は、ファイルシステムは 1 つ以上の大きな割り当て (`DAU`) を使用します。この結果、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、多数の小容量ファイルによるディスクの断片化は最小限に抑えられます。

- `mm` メタデータ装置では、小さな割り当ては 4 キロバイト、大きな割り当ては 16 キロバイトになります。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、`DAU` サイズとして大きな値を選択すると、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の詳細については、303 ページの「高度な機能」を参照してください。

## シングル割り当て方式

Sun StorEdge 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。Sun StorEdge QFS ファイルシステムは、mcf ファイルで装置タイプ `ma` です。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` 装置は、Sun StorEdge SAM-FS ファイルシステムのデュアル割り当て方式と同じく、`DAU` サイズが 16 キロバイト、32 キロバイト、または 64 キロバイトに制限されています。

`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 StorEdge 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 2-6 に、これらの DAU サイズを示します。

表 2-6 Sun StorEdge QFS の装置タイプ

装置タイプ	DAU のサイズ
mr または gXXX	8 キロ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る。DAU サイズは、16 キロバイト ~ 65,528 キロバイト (64M バイト) の範囲で指定できる。Sun StorEdge QFS 環境の mr または gXXX 装置のデフォルト DAU は 64 キロバイト。
md	このタイプの装置は、Sun StorEdge SAM-FS ファイルシステムと同じ方法でデュアル割り当てを使用する。DAU のサイズは、16K、32K、または 64K バイトに構成できる。Sun StorEdge QFS 環境の md 装置のデフォルト DAU は 64 キロバイト。 Sun StorEdge QFS ファイルシステムの md 装置はデータの格納のみに使用され、メタデータの格納には使用されない。これが、Sun StorEdge QFS ファイルシステムの md 装置と Sun StorEdge SAM-FS ファイルシステムの md 装置の違いである。

注 – Sun StorEdge QFS ソフトウェアをインストールしたときに、ファイルシステム上で `sammkfs(1M)` を実行しなかった場合は、バージョン 1 のスーパーブロックを使用します。バージョン 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しません。バージョン 1 のスーパーブロックでは、mm 装置の割り当ては 16 キロバイトです。Sun StorEdge QFS ファイルシステムの md 装置を定義できるのは、バージョン 2 のスーパーブロックのみです。

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。次のコマンドでは、128 キロバイトの DAU が指定されます。

```
# sammkfs -a 128 samqfs1
```

`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。

## 割り当て方式のまとめ

表 2-7 に、Sun StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムで使用可能な装置タイプを示します。

表 2-7 ファイルシステム装置の装置タイプ

mc f ファイル の装置タイプ	格納するデータのタイプ	割り当て方式	この装置タイプを使用できる ファイルシステム
md	ファイルデータとメタデータ	デュアル	Sun StorEdge SAM-FS
md	ファイルデータ	デュアル	Sun StorEdge QFS
mm	メタデータ	デュアル	Sun StorEdge QFS
mr	ファイルデータ	シングル	Sun StorEdge QFS
gXXX	ファイルデータ	シングル	Sun StorEdge QFS

Sun StorEdge SAM-FS ファイルシステム (ms ファイルシステム) にあるのは md 装置のみです。

Sun StorEdge QFS ファイルシステム (ma ファイルシステム) では、次の装置を混在できます。

- 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

表 2-8 に、さまざまなファイルシステムで使用される割り当て方式を示します。

表 2-8 ファイル割り当て

ファイルシステムとデバイスタイプ	割り当ての増分値
Sun StorEdge SAM-FS と md 装置	8 つのアドレスまでは 4 キロバイトのブロック、それ以降は DAU。
Sun StorEdge QFS と mr 装置	DAU
Sun StorEdge QFS と gX 装置	DAU
Sun StorEdge QFS と md 装置	8 つのアドレスまでは 4 キロバイトのブロック、それ以降は DAU。

表 2-9 に、DAU のデフォルトを示します。

表 2-9 デフォルトの DAU サイズ

ファイルシステムとデバイスタイプ	デフォルトの DAU サイズ
Sun StorEdge SAM-FS の md 装置	16 キロバイト
Sun StorEdge QFS の mr 装置と md 装置	64 キロバイト
Sun StorEdge QFS の gX 装置	256 キロバイト

## 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は、Sun StorEdge QFS と Sun StorEdge SAM-FS の各ファイルシステムで異なります。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。

この項では、ストライプ幅に影響する各ファイルシステムの違いについて説明します。

### Sun StorEdge SAM-FS のストライプ幅

Sun StorEdge SAM-FS ファイルシステムのストライプ幅はマウント時に設定されます。表 2-10 に、デフォルトのストライプ幅を示します。

表 2-10 Sun StorEdge SAM-FS 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	1 ディスクに書き込まれるデータ容量
16 キロバイト (デフォルト)	8 DAU	128 キロバイト
32 キロバイト	4 DAU	128 キロバイト
64 キロバイト	2 DAU	128 キロバイト

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 16 キロバイトになります。`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 8 に設定されます。

表 2-10 の 1 列目の数値に 2 列目の数値を乗じると、結果は 128 キロ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 128 キロバイト以上あると、Sun StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムはさらに効率的に動作します。

## Sun StorEdge QFS のストライプ幅 - ストライプ化グループを使用しない

Sun StorEdge QFS 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」は、グループとしてストライプ化された装置のコレクションです。ストライプ化グループの詳細については、26 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 StorEdge 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、DAU とストライプ幅の関係は、Sun StorEdge SAM-FS ファイルシステムでの DAU とストライプ幅の関係のようになります。違いは、64 キロバイトを超える DAU が可能なことと、DAU を 8 キロ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 65528 キロバイトです。

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容量は 128 キロバイトまたはその前後になります。書き込み操作で 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 StorEdge QFS ファイルシステムの効率が高まります。表 2-11 に、デフォルトのストライプ幅を示します。これは、ストライプ幅を指定しない場合に使用される幅です。

表 2-11 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	1 ディスクに書き込まれるデータ容量
16 キロバイト	8 DAU	128 キロバイト
24 キロバイト	5 DAU	120 キロバイト
32 キロバイト	4 DAU	128 キロバイト
40 キロバイト	3 DAU	120 キロバイト
48 キロバイト	2 DAU	96 キロバイト
56 キロバイト	2 DAU	112 キロバイト
64 キロバイト (デフォルト)	2 DAU	128 キロバイト
72 キロバイト	1 DAU	72 キロバイト
128 キロバイト	1 DAU	128 キロバイト
> 128 キロバイト	1 DAU	DAU サイズ

## Sun StorEdge QFS のストライプ幅 - ストライプ化グループを使用

ストライプ化グループが Sun StorEdge 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、同時にいくつかのディスク装置に書き込まれます。この割り当てでは、ディスクのグループが 1 つの装置のように扱われます。ストライプ化グループでの割り当ては、論理的には、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、ラウンドロビンです。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。不一致のストライプ化グループが存在すると、ファイルは 1 つのストライプ化グループのみに格納されます。

## Sun StorEdge QFS のデータ整合

「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 StorEdge QFS ファイルシステムの最適な整合式は次のとおりです。

$$\text{allocation\_unit} = \text{RAID\_stripe\_width} \times \text{number\_of\_data\_disks\_in\_the\_RAID}$$

たとえば、RAID-5 ユニットには合計 9 個のディスクがありますが、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 個です。RAID のストライプ幅が 64 キロバイトの場合、最適な割り当て単位は  $64 \times 8$  で 512 キロバイトです。

データファイルは、同じファイルシステム内に定義された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) において、ストライプ化またはラウンドロビン処理されます。

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。この章の後の部分では、DAU 設定やストライプ幅の決定時に考慮すべき事項について説明します。

## 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に関する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。16 キロバイトの DAU が 1 つメタデータディスクに書き込まれてから、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 4 キロバイトの 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、ストライプ割り当てまたはラウンドロビン式割り当てを使用して割り当てられます。これは、mount(1M) コマンドの `-o mm_stripe=n` オプションの指定によって決まります。最小の設定は `-o mm_stripe=0` で、ストライプ化が無効になります。最大の設定は `-o mm_stripe=255` です。

---

## ファイル割り当て方式

Sun StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムでは、ラウンドロビンとストライプ化の両方の割り当て方式を指定できます。表 2-12 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 2-12 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 StorEdge SAM-FS	ストライプ化	ストライプ化
Sun StorEdge QFS	ストライプ化	ストライプ化
Sun StorEdge 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 StorEdge 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

この後の各節で、これらの割り当てについて詳しく説明します。

## 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。

- Sun StorEdge SAM-FS 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る
- Sun StorEdge QFS 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る

Sun StorEdge QFS ファイルシステムでは、mm 装置にファイルデータが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 4 キロバイトです。表 2-13 に、システムによるメタデータの割り当てを示します。

表 2-13 メタデータ割り当て

メタデータタイプ	Sun StorEdge QFS ファイルシステム用の割り当ての増分値	Sun StorEdge SAM-FS 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)	16 キロバイト DAU	16、32、または 64 キロバイト (DAU)。
間接ブロック	16 キロバイト DAU	16、32、または 64 キロバイト (DAU)。
ディレクトリ	4 キロバイトブロックおよび 16 キロバイト DAU	4 キロバイト、最大で合計 32 キロバイト、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は、複数データストリームの場合に役立ちます。これは、このような環境では、全体のパフォーマンスがストライブ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。次のファイルは次の論理ディスクに書き込まれます。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。

次の図に、ラウンドロビン式割り当てを示します。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最初から再度開始します。

図 2-1 に、5 つのデバイス上でラウンドロビン割り当て方式を使用する Sun StorEdge SAM-FS ファイルシステムを示します。図 2-2 に、5 つのデバイス上でラウンドロビン割り当て方式を使用する Sun StorEdge QFS ファイルシステムを示します。

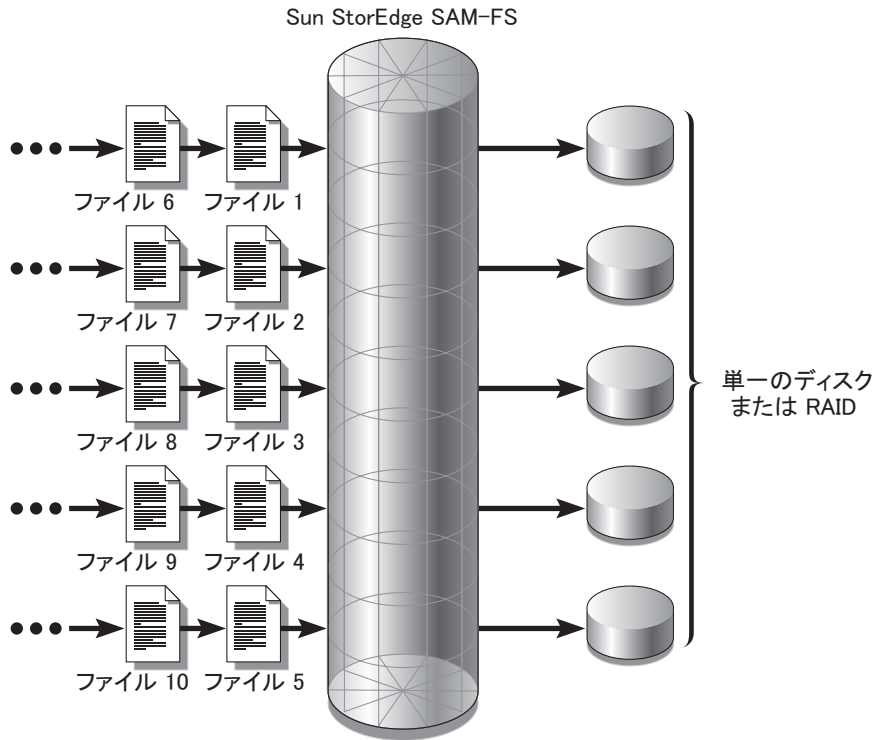


図 2-1 5つの装置を使用するラウンドロビン式 Sun StorEdge SAM-FS ファイルシステム



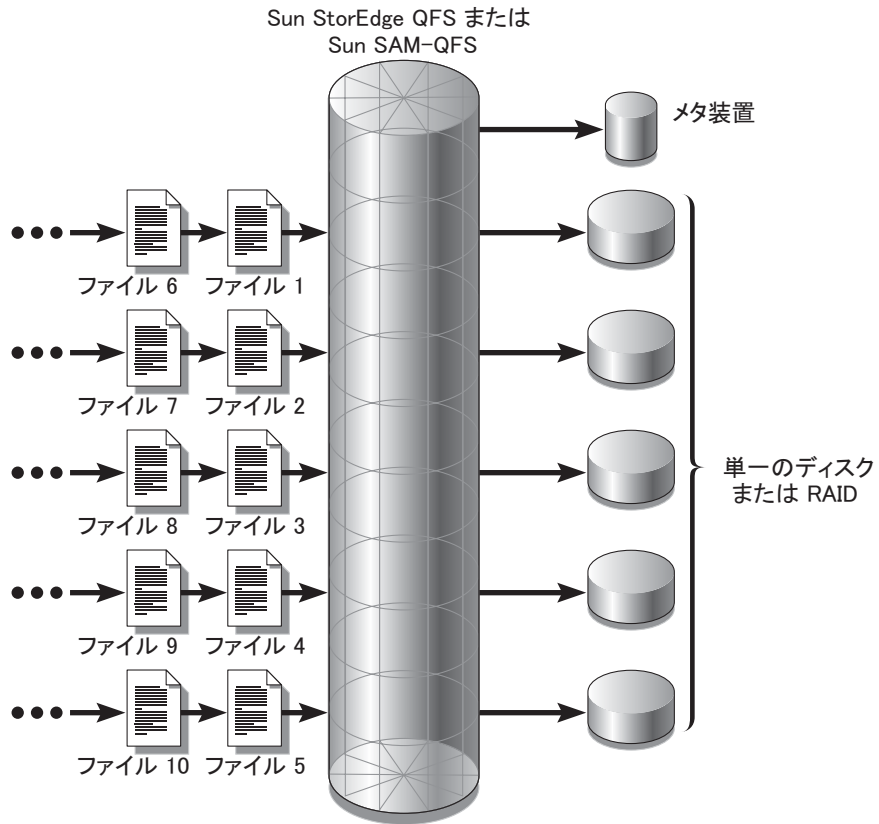


図 2-2 5つの装置を使用するラウンドロビン式 Sun StorEdge QFS ファイルシステム

## ストライプ化割り当て

デフォルトでは、Sun StorEdge QFS および Sun StorEdge SAM-FS 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で、すべての装置を合計したパフォーマンスを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。通常、ストライプ化によってパフォーマンスは向上します。これは、ディスクの読み取りと書き込みが、複数のディスクヘッドに並行して分散されるため

す。ストライプ化ディスクアクセスによって、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 StorEdge QFS または Sun StorEdge SAM-FS ファイルシステムが、ファイルへの *md* 装置に書き込みを開始するとき、ファイルのサイズは4キロバイトの小さな DAU に収まると仮定します。ファイルが、割り当て済みの最初の8個の小さな DAU (32キロバイト) 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 StorEdge QFS ファイルシステムが、ファイルへの *mr* 装置に書き込みを開始するときは、最初に1つの DAU、次の DAU、その次の DAU という順に書き込みます。*mr* 装置の DAU サイズは1つだけです。Sun StorEdge QFS ファイルシステムは、ストライプ化された *mm* 装置にもメタデータを書き込みます。

ストライプ化割り当てが使用されると、複数のファイルがアクティブになり、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。

次の図では、ストライプ割り当てを使用するファイルシステムを示します。これらの図では、ファイルの  $DAU \times stripe\_width$  バイトがディスク1、 $DAU \times stripe\_width$  バイトがディスク2、 $DAU \times stripe\_width$  バイトがディスク3という順に書き込まれます。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。

図 2-3 に、5 つのストライプ化されたデバイスを使用する Sun StorEdge SAM-FS ファイルシステムを示します。図 2-4 に、5 つのストライプ化されたデバイスを使用する Sun StorEdge QFS ファイルシステムを示します。

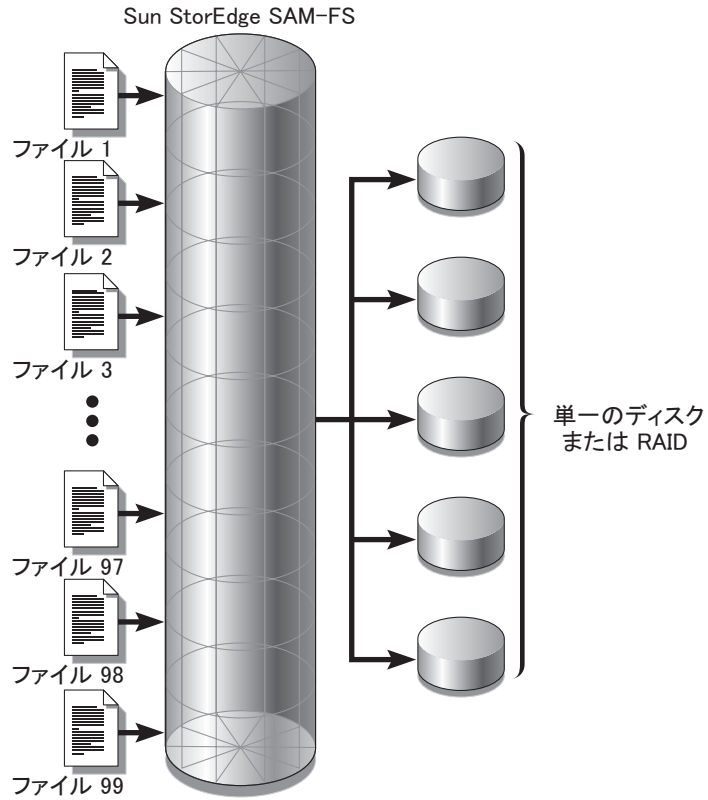


図 2-3 Sun StorEdge SAM-FS ファイルシステム

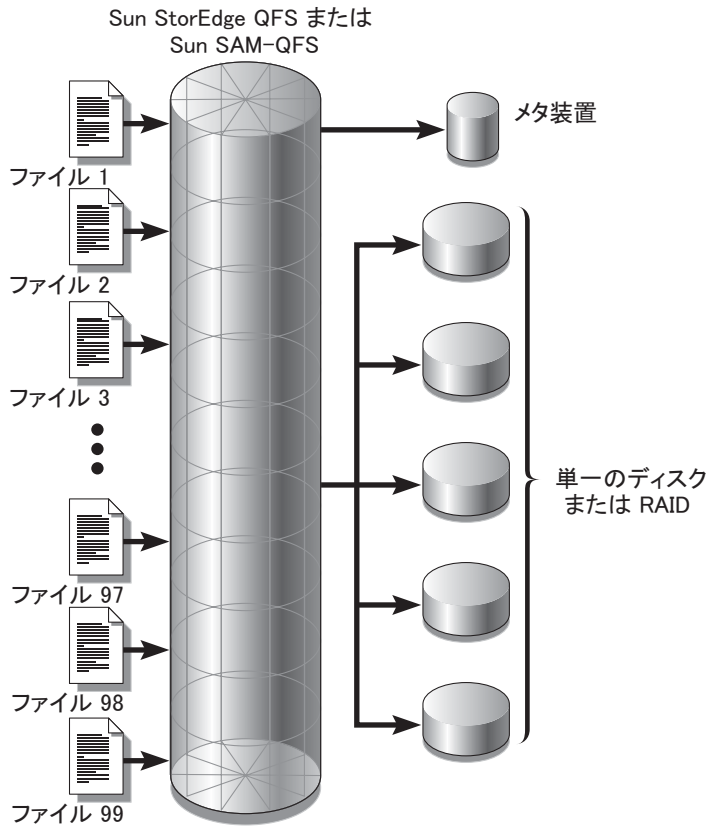


図 2-4 Sun StorEdge QFS ファイルシステム

## ストライプ化グループ (Sun StorEdge QFS ファイルシステムのみ)

「ストライプ化グループ」とは、非常に大規模な入出力条件とテラバイト単位のディスクキャッシュのあるファイルシステム用に設計された、特殊な Sun StorEdge 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含む装置タイプを指定できます。複数のストライプ化グループの装置タイプを、単一の Sun StorEdge 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 StorEdge QFS ファイルシステム内の装置のコレクションです。ストライプ化グループは、gXXX 装置として mcf ファイルで定義する必要があります。ストライプ化グループを使用すると、1 つのファイルの書き込みや読み取りを複数の装置との間で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 2-5 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 StorEdge QFS ファイルシステムを示しています。図 2-5 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 でラウンドロビンされます。3 つのストライプ化グループが定義されています (g0、g1、g2)。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。

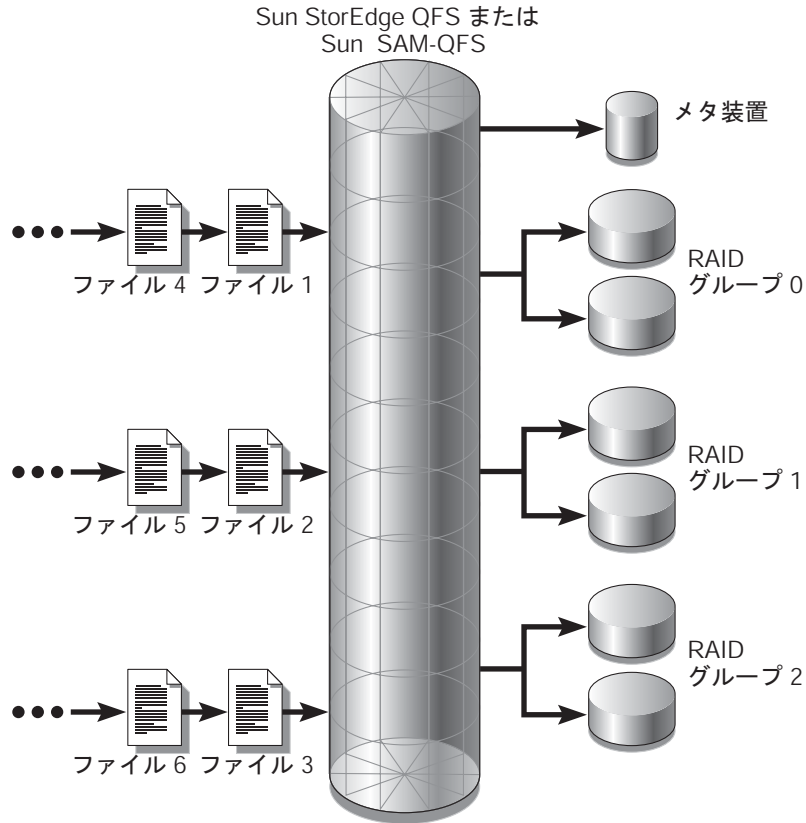


図 2-5 Sun StorEdge QFS のラウンドロビン式ストライプ化グループ

図 2-5 の構成で、`/etc/fstab` 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 2-2 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 2-2 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	-	

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 (続き)

/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 2-6 は、データがグループにストライプ化されるストライプ化グループを使用する Sun StorEdge QFS ファイルシステムを示しています。図 2-6 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 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/fstab 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。

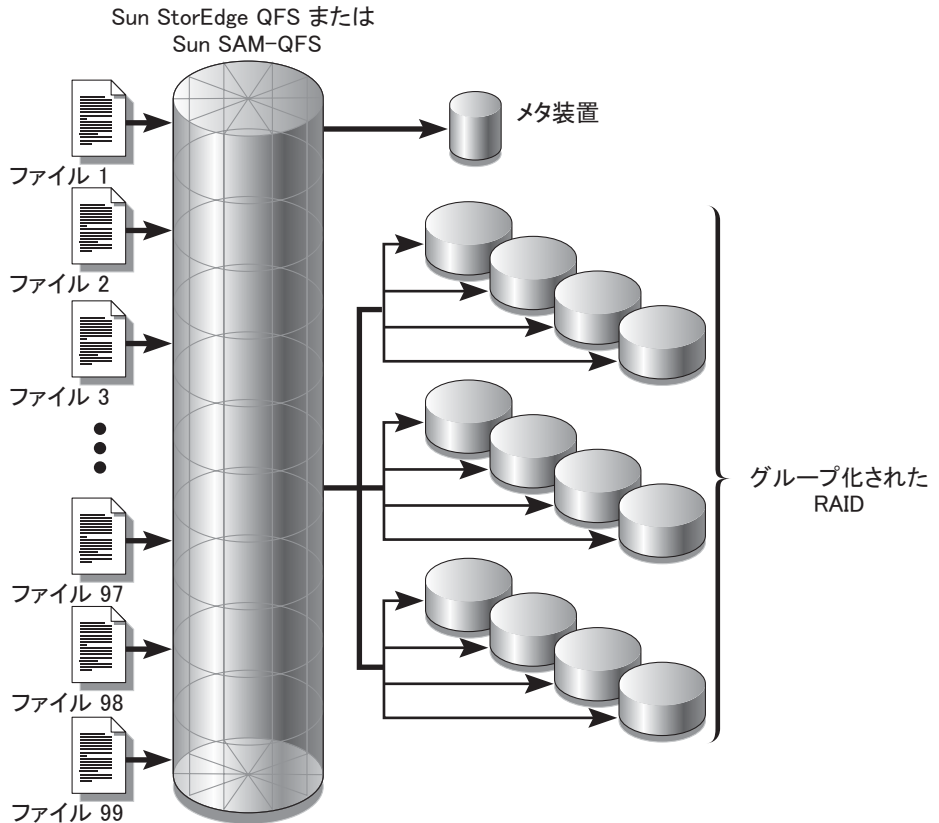


図 2-6 Sun StorEdge QFS のストライプ化グループ割り当て

## 不一致のストライプ化グループ (Sun StorEdge QFS ファイルシステムのみ)

不一致のストライプ化グループを含むファイルシステムを構築することは可能です。不一致のストライプ化グループのあるファイルシステムとは、グループごとに装置の数が異なる複数のストライプ化グループを持つファイルシステムです。Sun StorEdge 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ていますが、不一致のグループでのストライプ化はサポートしていません。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** – ファイルシステムに不一致のストライプ化グループが含まれている場合、1つのファイルが複数のストライプ化グループに分散されることはありません。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。

ストライプ化グループをどこまで一杯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を使用し、外部ストレージの状態の表示するために `m` 画面にアクセスします。

---

次の例に、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。

### 例

Sun StorEdge QFS のライセンスがあり、映像と音声の両方のデータを含むファイルシステムをサイトで作成する必要があるとします。

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 2-7 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 Sun StorEdge QFS ファイルシステムです。

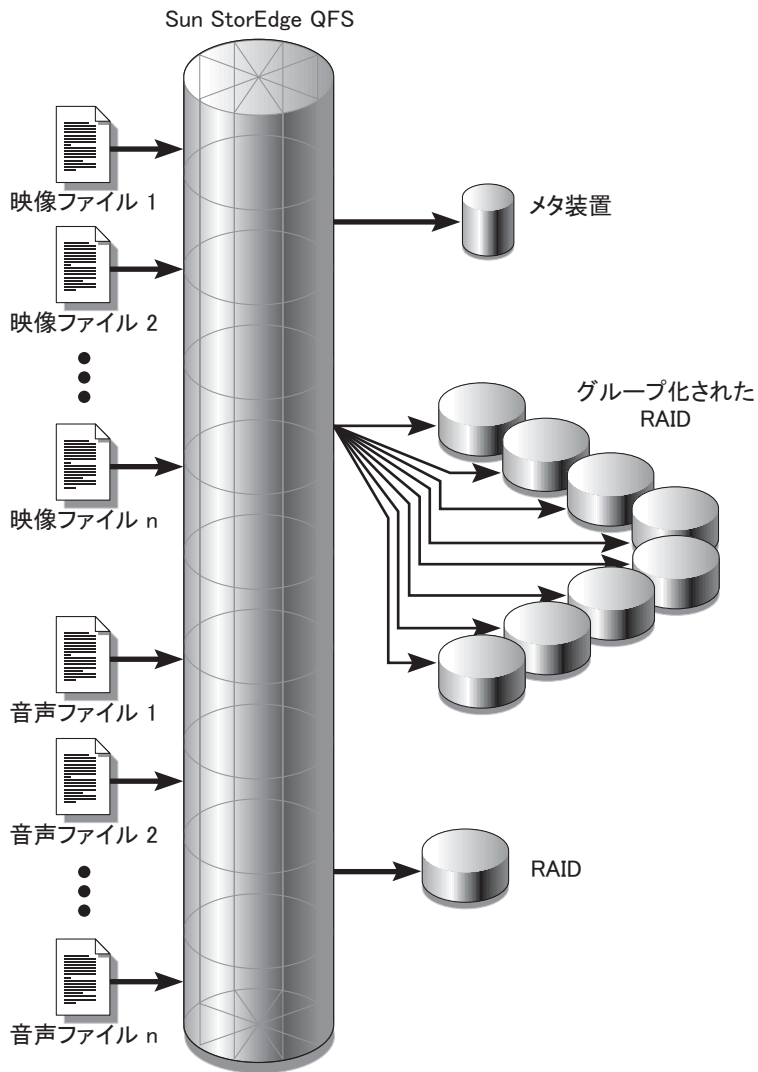


図 2-7 Sun StorEdge QFS ファイルシステム

表 2-14 に、このファイルシステムの特徴を示します。

表 2-14 ファイルシステム avfs 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	128 キロバイト
g0 のディスク数	8
g0 の最大ブロックサイズ	8 ディスク × 128 キロバイトの DAU = 1024 キロバイト。 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 128 キロ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 1024 キロバイト。
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ることに注意してください。0 よりも大きな値 (stripe > 0) 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 2-3 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 2-3 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 -
/dev/dsk/c05t0d0s6 106  g0   avfs -
/dev/dsk/c06t0d0s6 107  g0   avfs -
```

コード例 2-3 ファイルシステム avfs の mcf ファイル (続き)

```
/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 ファイルを用意してから、コード例 2-4 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 2-4 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 2-5 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 2-5 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 2-6 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 `video`、小容量ストライプ化グループを `audio` 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 2-6 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。



## 第3章

---

# ボリューム管理

---

マスター構成ファイル (mcf) は、Sun StorEdge QFS または Sun StorEdge SAM-FS 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。この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を構成する装置をファミリセットにグループ化します。

インストールおよび構成操作の詳細は、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』を参照してください。この章では、Sun StorEdge QFS および Sun StorEdge SAM-FS 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。

- 42 ページの「mcf ファイルの作成」
- 46 ページの「mcf ファイルの例」
- 50 ページの「ファイル設定、オプション、指示の相互関係」
- 50 ページの「ファイルシステムの初期化」
- 52 ページの「構成の例」

---

**注** – ストレージおよびアーカイブ管理に関して、Sun StorEdge SAM-FS への参照は、Sun SAM-QFS の構成にも当てはまります。ファイルシステムの設計および機能に関して、Sun StorEdge QFS への参照は、Sun SAM-QFS の構成にも当てはまりません。この節では、明確にするために必要な場合にのみ「Sun SAM-QFS」という名前を使用します。

---

## mc f ファイルの作成

Sun StorEdge QFS または Sun StorEdge SAM-FS ファイルシステムを構成する最初のステップでは、`/etc/opt/SUNWsamfs/mcf` にマスター構成ファイルを作成します。mcf ファイルには、RAID やディスク装置をファイルシステムに指定したり編成したりするために、これらのファイルシステムで必要な情報が含まれます。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 ASCII ファイルで、6 つの列またはフィールドに分けられる指定コードの行から構成されます。コード例 3-1 は、mcf ファイル行の 6 つのフィールドを示しています。

コード例 3-1 mcf ファイルのフィールド

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

mcf ファイルへのデータの入力方法には、次の規則があります。

- 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指定できます。コメント行は先頭にシャープ (#) を付けます。
- 有用な情報を入力する必要のないフィールドもあります。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。SAM-QFS マネージャを使用して、mcf ファイルを作成することもできます。SAM-QFS マネージャのインストールについては、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』を参照してください。SAM-QFS マネージャ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。

次の項で、mcf ファイルの各フィールドについて説明します。

- 43 ページの「Equipment Identifier フィールド」
- 44 ページの「Equipment Ordinal フィールド」
- 44 ページの「Equipment Type フィールド」
- 45 ページの「Family Set フィールド」
- 46 ページの「Device State フィールド」
- 46 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。このフィールドにファイルシステム名を指定する場合、その名前はファミリーセット名と同じにして、mcf ファイルの次の行で、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 ファイルでその他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のコメント行を前に付けるとよいでしょう。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ます。英字、数字、下線記号 () だけが使用できます。
- nodev キーワード。このフィールドに nodev キーワードを指定すると、mcf ファイルは Solaris ホスト上の Sun StorEdge QFS 共有ファイルシステムにおいてクライアントホストとして使用されます。Sun Cluster 環境ではこのキーワードは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 StorEdge 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 StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成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。このフィールドに /dev/samst エントリがある場合は、自動ライブラリまたは光磁気ディスクドライブが指定されます。ネットワーク接続自動ライブラリを構成する場合は、詳細について、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』および『Sun StorEdge SAM-FS ストレージ/アーカイブ管理マニュアル』を参照してください。
- テープドライブの説明。このフィールドがテープドライブの場合、エントリは次のいずれかの形式になります。
  - このフィールドには、/dev/rmt エントリを指定できます。
  - このフィールドには、/dev/rmt リンクのポイント先と同じ特殊なファイルをポイントするシンボリックリンクへのパスを指定できます。この方法でテープドライブを指定する場合は、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 フィールドにファミリーセット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。その他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 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。  $1 \leq eq\_ord \leq 65534$  のような一意の整数を指定します。これは必須フィールドです。

## 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。

表 3-1 に示すように、Sun StorEdge SAM-FS ファイルシステムの Equipment Type フィールドには ms または md を指定できます。

表 3-1 Sun StorEdge SAM-FS Equipment Type フィールド

Equipment Type フィールドの内容	意味
ms	Sun StorEdge SAM-FS ファイルシステムの定義
md	ファイルデータおよびメタデータの情報を格納するため、ストライプ装置またはラウンドロビン式の装置を定義。

表 3-2 に示すように、Sun StorEdge QFS または Sun SAM-QFS ファイルシステムの Equipment Type フィールドには ma、md、mm、mr、または gXXX を指定できます。

表 3-2 Sun StorEdge QFS または Sun SAM-QFS の Equipment Type フィールド

Equipment Type フィールドの内容	意味
ma	Sun StorEdge QFS または Sun SAM-QFS ファイルシステムの定義
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 ノードおよびその他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。
mr	ラウンドロビン式の装置またはストライプデータ装置を定義



表 3-2 Sun StorEdge QFS または Sun SAM-QFS の Equipment Type フィールド (続き)

Equipment Type フィールドの内容	意味
gXXX	<p>ストライプ化グループデータ装置。ストライプ化グループは、先頭の g の後に数値を付けて指定。number には、<math>0 \leq XXX \leq 127</math> のような整数を指定する必要がある。たとえば、g12 と指定する。</p> <p>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る。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ない。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ない。</p>

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、他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。

ファイルシステムを定義する行と、ファイルシステム内のディスク装置を定義する行には、すべて同じファミリーセット名を指定する必要があります。ソフトウェアは、ファミリーセットを使用して、同じファミリーセット名の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 オプションをともに使用して、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。

自動ライブラリとその関連ドライブを定義する行と、装置を定義する行では、同じファミリーセット名を指定する必要があります。

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。

## 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 または off です。このフィールドはオプション。on または off を入力したくない場合は、ダッシュ (-) を入力して、このフィールドを省略することを示します。

## Additional Parameters フィールド

Sun StorEdge SAM-FS ファイルシステムでは、Additional Parameters フィールドはオプションです。空のままにしておくことができます。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 StorEdge 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` を指定する必要があります。

Sun StorEdge 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。次の節では、サンプル mcf ファイルを示します。

## Sun StorEdge SAM-FS ポリューム管理の例

Sun StorEdge SAM-FS ファイルシステムでは、次の装置タイプを使用するとき、`/etc/opt/SUNWsamfs/mcf` ファイルの `Equipment Type` フィールドにファミリーセットを定義できます。

- Sun StorEdge SAM-FS ファイルシステムタイプに対して `ms` を使用するとき
- 装置に対して `md` を使用するとき データは、これらの装置にストライプ化またはラウンドロビン処理される。ストライプ幅は、`mount(1M)` コマンドの `-o stripe=n` オプションで設定する。デフォルトのストライプ幅は DAU サイズに基づいて設定される。ストライプ幅と DAU サイズの詳細は、9 ページの「ファイルシステム設計」を参照してください。

Sun StorEdge SAM-FS ファイルシステムのメタデータ (i ノード、ディレクトリ、割り当てマップなど) とファイルデータは、同じディスクに格納されます。データファイルは、同じファイルシステム内に定義されている各ディスクパーティションに対して、ストライプ化またはラウンドロビン処理されます。

コード例 3-2 は、Sun StorEdge SAM-FS ファイルシステムの mcf ファイルを示しています。

コード例 3-2 Sun StorEdge SAM-FS ファイルシステムのサンプル mcf ファイル

```
# Sun StorEdge SAM-FS ファイルシステムの構成例
#
# Equipment      Eq   Eq   Fam.  Dev.  Additional
# Identifier     Ord  Type Set   State Parameters
#-----
samfs1          10   ms   samfs1
/dev/dsk/c1t1d0s6 11   md   samfs1 -
/dev/dsk/c2t1d0s6 12   md   samfs1 -
/dev/dsk/c3t1d0s6 13   md   samfs1 -
/dev/dsk/c4t1d0s6 14   md   samfs1 -
/dev/dsk/c5t1d0s6 15   md   samfs1 -
```

## Sun StorEdge QFS および Sun SAM-QFS ボリューム管理の例

Sun StorEdge QFS および Sun SAM-QFS ファイルシステムでは、次の装置タイプを使用するとき、/etc/opt/SUNWsamfs/mcf ファイルの Equipment Type フィールドにファミリーセットを定義します。

- Sun StorEdge QFS または Sun SAM-QFS ファイルシステムのタイプに対して ma を使用するとき。
- メタデータ装置に対して mm を使用するとき。ファイルデータはこの装置には書き込まれない。複数のメタデータ装置を指定できる。Sun StorEdge QFS および Sun SAM-QFS ファイルシステムのメタデータ (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ず。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る。
- ファイルデータがストライプ化またはラウンドロビン処理される装置に対して mr または md を使用するとき。
- ファイルデータがグループとしてストライプ化される装置に対して gXXX を使用するとき。ストライプ化グループは、1つの単位としてストライプ化される装置の論理グループ。データは、各グループのメンバー上でストライプ化される

グループは、g0 ~ g127 までの装置タイプ番号で指定され、各装置でのストライプ幅は DAU になる。ストライプ化グループのすべての装置は、同じサイズであること。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ない。1つのファイルシステムに、mr および gXXX 装置を混合させることはできるが、md 装置は、1つのファイルシステム内の mr または gXXX 装置のいずれとも混合させることはできない。

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る。デフォルトはラウンドロビン。

データファイルは、同じファイルシステム内に定義されている各データディスクパーティション (mr または gXXX) 上で、ストライプ化またはラウンドロビン処理されません。

## 例 1

コード例 3-3 は、2つのストライプ化グループを含む Sun StorEdge QFS または Sun SAM-QFS ファイルシステムの mcf ファイルです。

コード例 3-3 ストライプ化グループを示す mcf ファイルの例

```
# Sun StorEdge 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  -
```

## 例 2

コード例 3-4 は、3つの Sun SAM-QFS ファイルシステムの mcf ファイルを示しています。

コード例 3-4 3つの Sun SAM-QFS ファイルシステムの mcf ファイルの例

```
# Sun SAM-QFS file system configuration example
#
# Equipment      Eq      Eq      Fam.  Dev.   Additional
# Identifier     Ord    Type   Set   State  Parameters
#-----
qfs1             10     ma     qfs1  -
```

コード例 3-4 3つの Sun SAM-QFS ファイルシステムの mcf ファイルの例 (続き)

```

/dev/dsk/c1t13d0s6 11 mm qfs1 -
/dev/dsk/c1t12d0s6 12 mr qfs1 -
#
qfs2
/dev/dsk/c1t5d0s6 21 mm qfs2 -
/dev/dsk/c5t1d0s6 22 mr qfs2 -
#
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 -

```

### 例 3

コード例 3-5 は、md 装置を使用する 1 つの Sun SAM-QFS ファイルシステムの mcf ファイルです。この mcf ファイルではテープライブラリも定義されています。

コード例 3-5 Sun SAM-QFS ファイルシステムおよびライブラリの mcf ファイルの例

```

# Sun SAM-QFS file system configuration example
#

# 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

```

mcf ファイルのファイルシステム構成を示す詳細な例は、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』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) コマンド行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、複数の方法で指定できます。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、70 ページの「ファイルシステムのマウント」を参照してください。

---

## ファイルシステムの初期化

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。*allocation\_unit* に指定する数値で DAU 設定が決まります。

これらのファイルシステムのリリース 4.2 では、2 種類のスーパーブロック設計がサポートされています。リリース 4.2 では両方のスーパーブロック設計を使用できます。コード例 3-6 で、samfsinfo(1M) コマンド出力はファイルシステムが使用しているスーパーブロックを示します。

コード例 3-6 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
```

この出力の最初の行に、これがバージョン 2 のスーパーブロックであることが示されています。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。

- 4.0 より前のリリースは、バージョン 1 スーパーブロック設計のみをサポートしています。
- 4.0 以降のリリースは、バージョン 2 スーパーブロックをサポートしています。4.0 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は、バージョン 2 スーパーブロックに依存する機能を使用するには、あらかじめリリース 4.0 または 4.2 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 StorEdge QFS 共有ファイルシステムなど、特定の機能は、バージョン 2 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、4.2 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後であればいつでも実行できます。



---

**注意** – バージョン 2 のスーパーブロックを使用するファイルシステムは、4.0 より前のリリースに戻すことができません。4.2 リリースのソフトウェアを使用して、バージョン 1 スーパーブロックを作成することはできません。

---

バージョン 2 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バージョン 2 スーパーブロックの作成の詳細は、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』を参照してください。

## 例

コード例 3-7 は、`sammkfs(1M)` コマンドを使用して、バージョン 2 スーパーブロックで Sun StorEdge SAM-FS ファイルシステムを初期化しています。

コード例 3-7      バージョン 2 スーパーブロックでのファイルシステムの初期化

```
# sammkfs -a 64 samfs1
Building 'samfs1' will destroy the contents of devices:
    /dev/dsk/c1t9d0s2
    /dev/dsk/c8t1d0s2
    /dev/dsk/c8t5d0s2
    /dev/dsk/c8t6d0s2
Do you wish to continue? [y/N] y
total data kilobytes      = 1715453952
total data kilobytes free = 1715453760
total meta kilobytes     = 17684128
total meta kilobytes free = 17680304
```

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。

## 構成の例

この節では、構成サンプルを示し、サーバーに 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。

- 52 ページの「Sun StorEdge QFS ラウンドロビン式ディスクの構成の作成」
- 54 ページの「Sun StorEdge SAM-FS ラウンドロビン式ディスクの構成の作成」
- 55 ページの「Sun StorEdge QFS ストライプ化ディスクの構成の作成」
- 56 ページの「Sun StorEdge SAM-FS ストライプ化ディスクの構成の作成」
- 58 ページの「Sun StorEdge QFS ストライプ化グループの構成の作成」

Sun StorEdge QFS のすべての構成サンプルでは、自動ライブリヤ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 StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』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、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と仮定しています。

## ▼ Sun StorEdge QFS ラウンドロビン式ディスクの構成の作成

この構成サンプルは、応答時間の遅いディスクにメタデータを分離する Sun StorEdge 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る。

この手順では、以下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、mcf ファイルを作成します。

コード例 3-8 に、mcf ファイルの例を示します。

コード例 3-8 Sun StorEdge QFS ラウンドロビン式 mcf ファイルの例

```
# Sun StorEdge QFS ディスクキャッシュの構成
# 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。

コマンドの例は次のとおりです。

```
# 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 StorEdge 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。

コマンドの例は次のとおりです。

```
# mount /qfs
```

## ▼ Sun StorEdge SAM-FS ラウンドロビン式ディスクの構成の作成

この構成サンプルは、Sun StorEdge SAM-FS ファイルシステムのもので、4つのパーティションに対して、デフォルトで、ストライプ化割り当てが使用されます。ラウンドロビン式割り当てを使用するには、`stripe=0` と設定する必要があります。ファイルシステムは、`sammkfs(1M)` コマンドを使用して作成されます。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る。

1. エディタを使用して、`mcf` ファイルを作成します。

コード例 3-9 に、`mcf` ファイルの例を示します。

コード例 3-9 Sun StorEdge SAM-FS ラウンドロビン式 `mcf` ファイルの例

```
# Sun StorEdge SAM-FS ディスクキャッシュの構成
# Round-robin mcf example

# Equipment      Eq  Eq  Fam.  Dev  Additional
# Identifier     Ord Type Set   State Parameters
#-----
samfs1           1  ms  samfs1
/dev/dsk/c1t1d0s6 11  md  samfs1 on
/dev/dsk/c2t1d0s6 12  md  samfs1 on
/dev/dsk/c3t1d0s6 13  md  samfs1 on
/dev/dsk/c4t1d0s6 14  md  samfs1 on
```

2. `mkdir(1)` コマンドを使用して、`/samfs1` ファイルシステムの `/samfs` マウントポイントを作成します。

コマンドの例は次のとおりです。

```
# mkdir /samfs
```

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 16 K バイトですが、次の例では DAU のサイズを 64 K バイトに設定しています。

```
# sammkfs -a 64 samfs1
```

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

Sun StorEdge SAM-FS 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。

```
samfs1 - /samfs 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。

コマンドの例は次のとおりです。

```
# mount /samfs
```

## ▼ Sun StorEdge QFS ストライプ化ディスクの構成の作成

この構成サンプルは、Sun StorEdge QFS ファイルシステムのもので、4つのデータパーティションに対してデフォルトで、ファイルデータがストライプ化されます。

この手順では、以下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。メタデータは装置 11 だけ書き込まれます。
- 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る。

1. エディタを使用して、`mcf` ファイルを作成します。

コード例 3-10 に、`mcf` ファイルの例を示します。

コード例 3-10 Sun StorEdge QFS ストライプ化 `mcf` ファイルの例

```
# Sun StorEdge QFS ディスクキャッシュの構成
# 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。

コマンドの例は次のとおりです。

```
# 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 64 K バイトですが、次の例では DAU のサイズを 128 K 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 StorEdge SAM-FS 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` DAU 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。

コマンドの例は次のとおりです。

```
# mount /qfs
```

## ▼ Sun StorEdge SAM-FS ストライプ化ディスクの構成の作成

この構成サンプルは、Sun StorEdge SAM-FS ファイルシステムのものであります。

1. エディタを使用して、mcf ファイルを作成します。

コード例 3-11 に、mcf ファイルの例を示します。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 LUN にある。

コード例 3-11 Sun StorEdge SAM-FS ストライプ化 mcf ファイルの例

```
# Sun StorEdge SAM-FS ディスクキャッシュの構成
# Striped Disk mcf example

# Equipment      Eq   Eq   Fam.  Dev.  Additional
# Identifier     Ord  Type Set   State Parameters
#-----
samfs1           10   ms   samfs1
/dev/dsk/c1t1d0s6 11   md   samfs1 on
/dev/dsk/c2t1d0s6 12   md   samfs1 on
/dev/dsk/c3t1d0s6 13   md   samfs1 on
/dev/dsk/c4t1d0s6 14   md   samfs1 on
```

2. mkdir(1) コマンドを使用して、/samfs1 ファイルシステムの /samfs マウントポイントを作成します。

コマンドの例は次のとおりです。

```
# mkdir /samfs
```

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 16K バイトの DAU を使用します。

```
# sammkfs samfs1
```

このストライプ化ディスク構成では、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、16 K 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。

/etc/vfstab でこのファイルシステムのマウントポイントを指定します。

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。

コマンドの例は次のとおりです。

```
# mount /samfs
```

## ▼ Sun StorEdge QFS ストライプ化グループの構成の作成

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 ビットだけが、 $n \times \text{DAU}$  を表すために使用されます。RAID 装置にまたがって大容量 DAU を書き込むこの方法は、ビットマップの領域やシステム更新時間を節約できます。ストライプ化グループは、RAID 装置のグループに非常に大きなファイルを書き込んだり、ディスク間で大量のデータのストリーミングを行う場合に役立ちます。

---

注 – 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。

$$\text{minimum\_disk\_space\_allocated} = \text{DAU} \times \text{number\_of\_disks\_in\_the\_group}$$

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  $\times$  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。

この構成サンプルは、応答時間の遅いディスクにメタデータを分離する Sun StorEdge 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、以下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) で構成される。各ディスクは、別の LUN にある。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る。

1. エディタを使用して、mcf ファイルを作成します。

コード例 3-12 に、mcf ファイルの例を示します。

コード例 3-12 Sun StorEdge QFS ストライプ化グループ mcf ファイルの例

```
# Sun StorEdge QFS ディスクキャッシュの構成
# 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。

コマンドの例は次のとおりです。

```
# 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 128 K 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、装置 12 と 13 がストライプ化され、ファイルは 2 つのストライプ化グループでラウンドロビン式割り当てが行われます。ストライプ化グループは結合されたエンティティとして扱っていることとなります。つまり、いったんストライプ化グループを作成すると、その構成を変更することはできません。`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。

コマンドの例は次のとおりです。

```
# mount /qfs
```



# ファイルシステムの操作

---

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。

- 62 ページの「ファイルシステムの初期化」
- 62 ページの「システムに対する構成ファイルの変更の伝達」
- 70 ページの「ファイルシステムのマウント」
- 74 ページの「ファイルシステムのマウント解除」
- 76 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」
- 78 ページの「アップグレードのための情報の保持」
- 83 ページの「ハードウェア装置のアップグレードの準備」
- 84 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」
- 86 ページの「ファイルシステムのディスクの交換」
- 89 ページの「ホストシステムのアップグレード」
- 90 ページの「Solaris OS のアップグレード」

Sun StorEdge QFS および Sun StorEdge SAM-FS 環境では、その他の種類の操作およびアップグレードも行なう必要があります。この他の操作については、次の資料で説明しています。

- 『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』は、Sun StorEdge QFS および Sun StorEdge SAM-FS ソフトウェアのインストール、アップグレード、および構成について説明しています。また、Sun StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムのダンプファイルの作成方法についても説明しています。
- 『Sun StorEdge SAM-FS ストレージ/アーカイブ管理マニュアル』は、自動ライブラリにスロットを追加し、自動ライブラリをアップグレードまたは交換し、DLT テープドライブをアップグレードする方法について説明しています。
- 『Sun QFS, Sun SAM-FS, Sun SAM-QFS 障害回復マニュアル』は、Sun StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムを復元する方法について説明しています。

---

## ファイルシステムの初期化

`sammkfs(1M)` コマンドを使用して、Sun StorEdge QFS または Sun StorEdge SAM-FS ファイルシステムの初期化または再初期化を行うことができます。次の例では、`sammkfs(1M)` コマンドの最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 Sun StorEdge QFS または Sun SAM-QFS ファイルシステムのバージョン 2 スーパーブロックを構築します。Sun StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムは、2 種類のスーパーブロックをサポートしています。Sun StorEdge QFS および Sun StorEdge SAM-FS 4.2 リリースは、バージョン 1 スーパーブロックの既存のファイルシステムをサポートしますが、バージョン 1 スーパーブロックを作成することはできません。

`sammkfs(1M)` コマンド、そのオプション、バージョン 1 およびバージョン 2 のスーパーブロックの詳細は、50 ページの「ファイルシステムの初期化」、または `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 StorEdge QFS ファイルシステムを初期化する方法については、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』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

この節では、システムに対して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` ファイル
- `defaults.conf` ファイル
- `archiver.cmd` ファイル (Sun SAM-QFS ファイルシステムのみ)
- `stager.cmd` ファイル (Sun SAM-QFS ファイルシステムのみ)
- 共有ホストファイル (Sun StorEdge QFS shared and Sun 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 StorEdge QFS または Sun StorEdge SAM-FS システムがすでに動作しているときに、`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

この節では、これらの手順について説明します。

- 63 ページの「`mcf(4)` または `defaults.conf(4)` の情報を Sun StorEdge QFS 環境で変更する」
- 64 ページの「`mcf(4)` または `defaults.conf(4)` ファイルシステム情報を Sun StorEdge SAM-FS または Sun SAM-QFS 環境で変更する」
- 65 ページの「`mcf(4)` または `defaults.conf(4)` リムーバブルメディアドライブ情報を変更する」
- 67 ページの「`archiver.cmd(4)` または `stager.cmd(4)` の情報を変更する」
- 67 ページの「マウントされたファイルシステムの共有ホストファイル情報を変更する」
- 68 ページの「マウント解除されたファイルシステムの共有ホストファイル情報を変更する」

## ▼ `mcf(4)` または `defaults.conf(4)` の情報を Sun StorEdge QFS 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 のすべての参加ノードでこの手順を実行します。

1. `vi(1)` または別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. `sam-fsd(1M)` コマンドを使用して、`mcf` ファイルにエラーがないかどうかを確認します。(省略可能)

`mcf` ファイルを変更する場合は、この操作を実行してください。コマンドの例は次のとおりです。

```
# 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` ファイルまたは `defaults.conf` ファイルの変更を伝達します。

コマンドの例は次のとおりです。

```
# 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(4)` または `defaults.conf(4)` ファイルシステム情報を Sun StorEdge SAM-FS または Sun SAM-QFS 環境で変更する

1. `vi(1)` または別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. `sam-fsd(1M)` コマンドを使用して `mcf` ファイルにエラーがないかどうかを確認します。(省略可能)

`mcf` ファイルを変更する場合は、この操作を実行してください。このコマンドの書式は次のとおりです。

```
# 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. `samcmd(1M) aridle` コマンドを実行して、`mcf` ファイルで定義された各ファイルシステムのアーカイバをアイドル状態にします。(省略可能)

1 つまたは複数のファイルシステムに関連する情報を削除または変更している場合は、この操作を実行する必要があります。このコマンドは、次の形式で使用します。

```
samcmd aridle fs.fsname
```

*fsname* には、`mcf` ファイルで定義されているファイルシステムの名前を指定します。`mcf` ファイル内で変更の対象となるすべてのファイルシステムに対してこのコマンドを実行します。

4. `samcmd(1M) idle` コマンドを実行して、`mcf` ファイルでドライブに割り当てられた各装置番号のアーカイバをアイドル状態にします。(省略可能)

1 つまたは複数のドライブに関連する情報を削除または変更している場合は、この操作を実行する必要があります。このコマンドは、次の形式で使用します。

```
samcmd idle eq
```

`eq` には、`mcf` ファイルで定義されているドライブの装置番号を指定します。`mcf` ファイル内で変更の対象となるすべてのドライブについて、必要であればこのコマンドを繰り返します。

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

ファイルシステムのマウント解除の詳細については、74 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. `samd(1M) config` コマンドを使用して、変更を伝達します。

コマンドの例は次のとおりです。

```
# 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(4)` または `defaults.conf(4)` 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. `sam-fsd(1M)` コマンドを使用して、`mcf` ファイルにエラーがないかどうかを確認します。(省略可能)

`mcf` ファイルを変更する場合は、この操作を実行してください。このコマンドは、次の形式で使用します。

```
# 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。

3. `samcmd(1M) aridle` コマンドを実行して、`mcf` ファイルで定義された各ファイルシステムのアーカイバをアイドル状態にします。(省略可能)

1 つまたは複数のファイルシステムに関連する情報を削除または変更している場合は、この操作を実行します。このコマンドは、次の形式で使用します。

```
samcmd aridle fs.fsname
```

*fsname* には、`mcf` ファイルで定義されているファイルシステムの名前を指定します。`mcf` ファイル内で変更の対象となるすべてのファイルシステムに対してこのコマンドを実行します。

4. `mcf` ファイルでドライブに割り当てられた各装置番号に対して、`samcmd(1M) idle` コマンドを実行します。(省略可能)

1 つまたは複数のドライブに関連する情報を削除または変更している場合は、この操作を実行します。このコマンドは、次の形式で使用します。

```
samcmd idle eq
```

*eq* には、`mcf` ファイルで定義されているドライブの装置番号を指定します。`mcf` ファイル内で変更の対象となるすべてのドライブについて、必要であればこのコマンドを繰り返します。

5. `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。

コマンドの例は次のとおりです。

```
# samd stop
```

6. `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。

コマンドの例は次のとおりです。

```
# samd config
```

7. `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開始します。

コマンドの例は次のとおりです。

```
# 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。

## ▼ `archiver.cmd(4)` または `stager.cmd(4)` の情報を変更する

1. `vi(1)` または別のエディタを使用して、`archiver.cmd(4)` または `stager.cmd(4)` ファイルを編集します。
2. `archiver(1M) -lv` コマンドを使用して、`archiver.cmd(4)` ファイルに加えた変更を確認します。(省略可能)  
既存の `archiver.cmd(4)` ファイルを変更する場合にのみ、この操作を実行してください。
3. ファイルを保存し、閉じます。
4. `samd(1M) config` コマンドを使用してファイルの変更を伝達し、システムを再起動します。  
コマンドの例は次のとおりです。

```
# samd config
```

## ▼ マウントされたファイルシステムの共有ホストファイル情報を変更する

この手順を使用して、新しいホストエントリを追加するか、既存の共有ホストファイルエントリの列 2～5 を変更します。

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) file-system-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。  
コマンドの例は次のとおりです。

```
# 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。  
コマンドの例は次のとおりです。

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

3. 共有ホストファイルのコピーを保存します。(省略可能)  
コマンドの例は次のとおりです。

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

4. vi(1) または別のエディタを使用して、/tmp/file 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。

---

**注** – マウントされたファイルシステムの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ことはできません。このような変更を加えるには、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、メタデータサーバーをマウント解除して、68 ページの「マウント解除されたファイルシステムの共有ホストファイル情報を変更する」の指示に従います。

---

5. 共有ホストファイルを保存し、閉じます。
6. 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。  
コマンドの例は次のとおりです。

```
# 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。  
コマンドの例は次のとおりです。

```
# samsharefs -u sharefs1
```

## ▼ マウント解除されたファイルシステムの共有ホストファイル情報を変更する

この手順を使用して、共有ホストファイルのホスト名の変更、エントリの並び替え、およびエントリの挿入を行います。



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) -R file-system-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。

コマンドの例は次のとおりです。

```
# 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。コマンドの例は次のとおりです。

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

4. 共有ホストファイルのコピーを保存します。(省略可能)  
コマンドの例は次のとおりです。

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

5. `vi(1)` または別のエディタを使用して、`/tmp/file` ファイルを編集します。  
マウント解除されたファイルシステムでは、ホスト名の変更、エントリの並び替え、エントリの挿入、新しいホストエントリの追加、既存のエントリの列 2～5 の編集を行うことができます。
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。  
コマンドの例は次のとおりです。

```
# 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。  
コマンドの例は次のとおりです。

```
# 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。

---

# ファイルシステムのマウント

Solaris OS の `mount(1M)` コマンドを使用して、Sun StorEdge QFS または Sun StorEdge SAM-FS ファイルシステムをマウントできます。この節では、マウントオプションのさまざまな指定方法について説明します。

マウントパラメタを使用して、ファイルシステムの特性を操作します。マウントパラメタの指定方法はいくつかあり、指定方法によっては他の方法による指定が無効になります。マウントオプションは、次の方法で指定できます。

1. `mount(1M)` コマンドのコマンド行オプション。優先順位は最も高い。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他のオプション、`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る。
2. `/etc/vfstab` ファイルの設定。優先順位は 2 番目。
3. `samfs.cmd` ファイルでの指示優先順位は 3 番目。
4. システムのデフォルト。優先順位は 4 番目 (最低)。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る。

また、`samu(1M)` オペレータユーティリティまたは `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。これらのいずれかのユーティリティ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。

以下では、マウントオプションを指定する方法について詳しく説明し、どのような場合にファイルやコマンドを使用するか、また、その優先順位を示します。『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』にも、ファイルシステムのマウントに関する情報があります。

## `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ウォーターマークと最低ウォーターマークなどを指定できます。

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。mount(1M) コマンドのオプションによって、/etc/vfstab のエントリと samfs.cmd ファイルの指示はどちらも無効になります。

例：次のコマンドは、setuid の実行を禁止し、qwrite を有効にして、/work のファイルシステム qfs1 をマウントします。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、コマンドで各オプションを区切ります。

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

Sun StorEdge QFS または Sun 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する必要があります。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な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

/etc/vfstab Solaris OS システムファイルには、mcf ファイルで定義された各 Sun StorEdge QFS または Sun StorEdge SAM-FS ファイルシステムの行が必要です。これは、ファイルシステムをマウントするために要求されます。ファイルシステムごとに、表 4-1 に示す 7 つのフィールドの情報を指定する必要があります。

表 4-1 /etc/vfstab ファイルのフィールド

フィールド番号	内容
1	ファイルシステムのファミリセット名
2	samfsck(1M) に対するファイルシステム
3	マウントポイント
4	ファイルシステムタイプ。これは、Sun StorEdge QFS ファイルシステムの場合でも、常に samfs です。

表 4-1 /etc/vfstab ファイルのフィールド (続き)

フィールド 番号	内容
5	samfsck(1M) パス
6	ブートオプションでのマウント
7	マウントパラメタ

/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。ただし、7 番目のフィールドのマウントパラメタは、コンマ (,) の後に空白文字を付けずに区切る必要があります。

例 : /etc/vfstab ファイルの例です。

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

マウントパラメ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タをどれでも指定できます。これらのパラメタは、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 (SAM) のさまざまな設定、Qwrite、およびその他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 StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムに対してマウントパラメタを指定できます。このファイルは、複数のファイルシステムに、同じマウントパラメタを指定して構成する場合に役立ちます。

使用可能なマウントパラメタのリストは、非常に広範に及びます。指定できる使用可能なマウントパラメタは、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 (SAM) のさまざまな設定、および Qwrite、その他の機能に関連するものです。

このファイルを使用すると、すべてのマウントパラメタを 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 StorEdge QFS および Sun StorEdge SAM-FS 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

パラメタを指定できます。共通のパラメタを一度に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タを、7番目のフィールドに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タとほとんど同じです。このファイルに指定できるマウントパラメタの詳細は、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。

fs = 行よりも前の指示は、すべてのファイルシステムにグローバルに適用されます。特定のファイルシステム固有の指示の前には、fs = で始まる行を指定する必要があります。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 4-1 のサンプル samfs.cmd ファイルでは、ディスクキャッシュ利用率の最低ウォーターマークと最高ウォーターマークが設定され、2つのファイルシステムの個別のパラメタが指定されています。

#### コード例 4-1 samfs.cmd ファイルのサンプル

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

samfs.cmd ファイルの指示がデフォルトとして使用され、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。

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。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。

# 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 StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムをマウント解除できます。

Sun StorEdge SAM-FS および Sun SAM-QFS ファイルシステムでは、ファイルシステムをマウント解除する前にアーカイバを停止するためのコマンドを実行する必要があります。次の手順では、アーカイバをアイドル状態にして、ファイルシステムをマウント解除する方法を示します。Sun StorEdge QFS ファイルシステムを使用している場合は、アーカイバをアイドル状態にする必要はありません。

## ▼ スタンドアロン Sun StorEdge SAM-FS および Sun SAM-QFS ファイルシステムをマウント解除する

1. ファイルシステムに対して `samcmd(1M) aridle fs.fsname` コマンドを実行します。(省略可能)

Sun StorEdge SAM-FS または Sun SAM-QFS ファイルシステムをマウント解除する場合は、この操作を実行します。コマンドの例は次のとおりです。

```
# samcmd aridle fs.samqfs2
```

この操作によって、`samqfs2` ファイルシステムのアーカイブが正常に停止します。特に、デーモンを停止する前の論理位置でアーカイブ操作を停止できます。

2. `samd(1M) stop` コマンドを実行します。(省略可能)

コマンドの例は次のとおりです。

```
# samd stop
```

3. ファイルシステムをマウント解除します。

```
# umount /samqfs
```

マウント解除するときには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ます。このため、場合によっては `umount(1M)` コマンドを 2 回実行する必要があります。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)、その他のコマンドを umount(1M) コマンドと合わせて使用してください。マウント解除の手順については、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』でも説明されています。

## ▼ Sun StorEdge QFS および Sun SAM-QFS 共有ファイルシステムをマウント解除する

umount(1M) コマンドによって、共有ファイルシステムが Solaris システムからマウント解除されます。umount(1M) コマンドの詳細については、umount(1M) のマニュアルページを参照してください。

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。

```
# umount /samqfs
```

2. メタデータサーバーで、samcmd(1M) aridle fs.fsname コマンドを実行して、論理位置でファイルシステムのアーカイブを停止します。(省略可能)

この手順は、Sun SAM-QFS ファイルシステム環境で実行します。

コマンドの例は次のとおりです。

```
# samcmd aridle fs.samqfs2
```

この例では、ファイルシステム samqfs2 のアーカイブを停止します。

3. 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。

```
# umount /samqfs
```

メタデータサーバーのマウント解除を行うのは、すべてのクライアントホストをマウント解除してからです。

マウント解除するときには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ます。このため、場合によっては umount(1M) コマンドを 2 回実行する必要があります。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)、その他のコマンドを umount(1M) コマンドと合わせて使用してください。マウント解除の手順については、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』でも説明されています。

# 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 StorEdge QFS および Sun StorEdge SAM-FS 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レコードを書き込みます。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 4-2 は、これらのエラー表示の概要を示しています。

表 4-2 エラー表示

エラー	Solaris OS での意味	Sun StorEdge QFS および Sun StorEdge SAM-FS 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 4-2 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。

## ▼ ファイルシステムの検査

- `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。

`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。

## ▼ ファイルシステムの修復

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、74 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。

---

## アップグレードのための情報の保持

環境のディスク、コントローラ、その他の装置を追加または変更するときに、mcf ファイルのすべてのファイルシステムの指定の修正や再生成が難しい場合があります。samfsconfig(1M) コマンドは、このような変更を行った後で、ファイルシステムおよびファイルシステム構成要素の情報を生成する場合に役立ちます。

samfsconfig(1M) コマンドは装置を検査して、Sun StorEdge QFS または Sun StorEdge SAM-FS スーパーブロックがあるかどうかを判断します。検出されたスーパーブロックの情報を使用して、mcf ファイルと同様の書式で装置についてまとめます。この書式を保存して編集すると、mcf ファイルが破壊されたり、紛失したり、正しくない場合に、ファイルを再作成できます。

このコマンドでは、指定する各装置についての情報が戻され、stdout にその情報が書き込まれます。ベース装置 (ファイルシステムそのもの) のファミリセット番号、ファイルシステムのタイプ (ma または ms)、ファイルシステムが Sun StorEdge QFS 共有ファイルシステムかどうかを検出できます。

異常がある場合は、次のようにフラグが付けられます。

- ハッシュ記号 (#)。ファミリセットの情報が不完全であることを示す。
- 不等号 (>)。複数の装置名が特定のファイルシステム要素を参照していることを示す。

システムを再構成する場合や、システムで障害が起きた場合には、必要であれば、このコマンドの出力を使用して、mcf ファイルのファイルシステムの部分を再生成できます。次の例は、samfsconfig(1M) コマンドの出力です。

## 例 1

この例では、システム管理者が装置名のリストをファイルに入力しています。これらの装置名は環境では報告されません。システム管理者は、Sun StorEdge QFS および Sun StorEdge SAM-FS ファミリセットに対して、これらの装置のみを検査します。結果には、ファミリセットの古いフラグメントといくつかの完全なインスタンスが示されています。

コード例 4-2 例 1 - samfsconfig(1M) コマンドの出力

```
mn# samfsconfig -v `cat /tmp/dev_files`
Device '/dev/dsk/c0t0d0s0' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c0t0d0s1' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c0t0d0s3' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c0t0d0s4' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c0t0d0s5' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c0t0d0s6' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c0t0d0s7' doesn't have a SAM-FS superblock (SBLK).
Couldn't open '/dev/dsk/c0t1d0s0'; errno=5.
Couldn't open '/dev/dsk/c0t1d0s1'; errno=5.
Device '/dev/dsk/c0t1d0s3' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c0t1d0s4' has a SAM-FS superblock.
Device '/dev/dsk/c0t1d0s5' has a SAM-FS superblock.
Device '/dev/dsk/c0t1d0s6' has a SAM-FS superblock.
Couldn't open '/dev/dsk/c0t1d0s7'; errno=5.
Couldn't open '/dev/dsk/c0t6d0s0'; errno=16.
Couldn't open '/dev/dsk/c0t6d0s1'; errno=16.
Couldn't open '/dev/dsk/c0t6d0s3'; errno=16.
Couldn't open '/dev/dsk/c0t6d0s4'; errno=16.
Couldn't open '/dev/dsk/c0t6d0s5'; errno=16.
Couldn't open '/dev/dsk/c0t6d0s6'; errno=16.
Couldn't open '/dev/dsk/c0t6d0s7'; errno=16.
Couldn't open '/dev/dsk/c1t0d0s3'; errno=5.
Couldn't open '/dev/dsk/c1t0d0s4'; errno=5.
Couldn't open '/dev/dsk/c1t0d0s5'; errno=5.
Device '/dev/dsk/c1t0d0s6' doesn't have a SAM-FS superblock (SBLK).
Couldn't open '/dev/dsk/c1t0d0s7'; errno=5.
Couldn't open '/dev/dsk/c1t1d0s0'; errno=2.
Couldn't open '/dev/dsk/c1t2d0s3'; errno=5.
Couldn't open '/dev/dsk/c1t2d0s4'; errno=5.
Couldn't open '/dev/dsk/c1t2d0s5'; errno=5.
Device '/dev/dsk/c1t2d0s6' doesn't have a SAM-FS superblock (SBLK).
Couldn't open '/dev/dsk/c1t2d0s7'; errno=5.
Could not read from device '/dev/dsk/c1t3d0s0'; errno=5.
Couldn't open '/dev/dsk/c1t4d0s3'; errno=5.
Couldn't open '/dev/dsk/c1t4d0s4'; errno=5.
Couldn't open '/dev/dsk/c1t4d0s5'; errno=5.
Device '/dev/dsk/c1t4d0s6' doesn't have a SAM-FS superblock (SBLK).
```

コード例 4-2 例 1 - samfsconfig(1M) コマンドの出力 (続き)

```
Device '/dev/dsk/c1t4d0s7' doesn't have a SAM-FS superblock (SBLK).
Couldn't open '/dev/dsk/c1t5d0s3'; errno=5.
Couldn't open '/dev/dsk/c1t5d0s4'; errno=5.
Couldn't open '/dev/dsk/c1t5d0s5'; errno=5.
Device '/dev/dsk/c1t5d0s6' doesn't have a SAM-FS superblock (SBLK).
Couldn't open '/dev/dsk/c1t5d0s7'; errno=5.
Device '/dev/dsk/c3t0d0s0' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c3t0d0s1' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c3t0d0s3' has a SAM-FS superblock.
Device '/dev/dsk/c3t0d0s4' has a SAM-FS superblock.
Couldn't open '/dev/dsk/c3t0d0s7'; errno=5.
Device '/dev/dsk/c3t1d0s0' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c3t1d0s1' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c3t1d0s3' has a SAM-FS superblock.
Device '/dev/dsk/c3t1d0s4' has a SAM-FS superblock.
Couldn't open '/dev/dsk/c3t1d0s7'; errno=5.
Device '/dev/dsk/c4t0d0s0' has a SAM-FS superblock.
Could not read from device '/dev/dsk/c4t0d0s1'; errno=5.
Could not read from device '/dev/dsk/c4t0d0s3'; errno=5.
Could not read from device '/dev/dsk/c4t0d0s4'; errno=5.
Could not read from device '/dev/dsk/c4t0d0s5'; errno=5.
Device '/dev/dsk/c4t0d0s6' has a SAM-FS superblock.
Device '/dev/dsk/c4t0d0s7' has a SAM-FS superblock.
Device '/dev/dsk/c4t1d0s0' has a SAM-FS superblock.
Could not read from device '/dev/dsk/c4t1d0s1'; errno=5.
Could not read from device '/dev/dsk/c4t1d0s3'; errno=5.
Could not read from device '/dev/dsk/c4t1d0s4'; errno=5.
Could not read from device '/dev/dsk/c4t1d0s5'; errno=5.
Device '/dev/dsk/c4t1d0s6' has a SAM-FS superblock.
Device '/dev/dsk/c4t1d0s7' has a SAM-FS superblock.
Device '/dev/dsk/c4t2d0s0' has a SAM-FS superblock.
Could not read from device '/dev/dsk/c4t2d0s1'; errno=5.
Could not read from device '/dev/dsk/c4t2d0s3'; errno=5.
Could not read from device '/dev/dsk/c4t2d0s4'; errno=5.
Could not read from device '/dev/dsk/c4t2d0s5'; errno=5.
Device '/dev/dsk/c4t2d0s6' has a SAM-FS superblock.
Device '/dev/dsk/c4t2d0s7' has a SAM-FS superblock.
Device '/dev/dsk/c4t3d0s0' has a SAM-FS superblock.
Could not read from device '/dev/dsk/c4t3d0s1'; errno=5.
Could not read from device '/dev/dsk/c4t3d0s3'; errno=5.
Could not read from device '/dev/dsk/c4t3d0s4'; errno=5.
Could not read from device '/dev/dsk/c4t3d0s5'; errno=5.
Device '/dev/dsk/c4t3d0s6' has a SAM-FS superblock.
Device '/dev/dsk/c4t3d0s7' has a SAM-FS superblock.
19 SAM-FS devices found.
#
# Family Set 'samfs2' Created Mon Jun 25 10:37:52 2001
```

コード例 4-2 例 1 - samfsconfig(1M) コマンドの出力 (続き)

```
#
# Missing slices
# Ordinal 1
# /dev/dsk/c0t1d0s6    12    md    samfs2  -
#
# Family Set 'samfs1' Created Wed Jul 11 08:47:38 2001
#
# Missing slices
# Ordinal 1
# /dev/dsk/c0t1d0s4    12    md    samfs1  -
# Ordinal 2
# /dev/dsk/c0t1d0s5    13    md    samfs1  -
#
# Family Set 'samfs2' Created Sat Nov  3 17:22:44 2001
#
samfs2 ma 30 samfs2 - shared
/dev/dsk/c4t0d0s6    31    mm    samfs2  -
/dev/dsk/c4t1d0s6    32    mr    samfs2  -
/dev/dsk/c4t2d0s6    33    mr    samfs2  -
#
# Family Set 'qfs1' Created Wed Nov  7 15:16:19 2001
#
qfs1 ma 10 qfs1 -
/dev/dsk/c3t0d0s3    11    mm    qfs1    -
/dev/dsk/c3t0d0s4    12    g0    qfs1    -
/dev/dsk/c3t1d0s3    13    g0    qfs1    -
/dev/dsk/c3t1d0s4    14    g0    qfs1    -
#
# Family Set 'sharefsx' Created Wed Nov  7 16:55:19 2001
#
sharefsx ma 200 sharefsx - shared
/dev/dsk/c4t0d0s0    210   mm    sharefsx -
/dev/dsk/c4t1d0s0    220   mr    sharefsx -
/dev/dsk/c4t2d0s0    230   mr    sharefsx -
/dev/dsk/c4t3d0s0    240   mr    sharefsx -
#
# Family Set 'samfs5' Created Tue Nov 27 16:32:28 2001
#
samfs5 ma 80 samfs5 -
/dev/dsk/c4t3d0s6    82    mm    samfs5  -
/dev/dsk/c4t3d0s7    83    g0    samfs5  -
/dev/dsk/c4t0d0s7    84    g0    samfs5  -
/dev/dsk/c4t1d0s7    85    g1    samfs5  -
/dev/dsk/c4t2d0s7    86    g1    samfs5  -
```

## 例 2

この例では、不等号 (>) のフラグが付いた装置が重複しています。スライス s0 はディスクの先頭から開始し、ディスク全体 (s2) のスライスも同様になっています。これは、Solaris 9 OE で出力できるスタイルです。

コード例 4-3 は、`samfsconfig(1M)` コマンドおよび出力を示しています。

コード例 4-3      例 2 - `samfsconfig` コマンドの出力

```
# samfsconfig /dev/dsk/c3t*
#
# Family Set 'shsam1' Created Wed Oct 17 14:57:29 2001
#
shsam1 160 ma shsam1 shared
> /dev/dsk/c3t50020F23000055A8d0s2    161    mm    shsam1  -
> /dev/dsk/c3t50020F23000055A8d0s0    161    mm    shsam1  -
/dev/dsk/c3t50020F23000055A8d0s1    162    mr    shsam1  -
> /dev/dsk/c3t50020F23000078F1d0s0    163    mr    shsam1  -
> /dev/dsk/c3t50020F23000078F1d0s2    163    mr    shsam1  -
/dev/dsk/c3t50020F23000078F1d0s1    164    mr    shsam1  -
```

## 例 3

この例では、ディスクスライス全体 (スライス 2) がコマンド行で省かれています。これは、Solaris 9 OE で出力できるスタイルです。

コード例 4-4 は、`samfsconfig(1M)` コマンドおよび出力を示しています。

コード例 4-4      例 3 - `samfsconfig(1M)` コマンドの出力

```
# samfsconfig /dev/dsk/c3t*s[013-7]
#
# Family Set 'shsam1' Created Wed Oct 17 14:57:29 2001
#
shsam1 160 ma shsam1 shared
/dev/dsk/c3t50020F23000055A8d0s0    161    mm    shsam1  -
/dev/dsk/c3t50020F23000055A8d0s1    162    mr    shsam1  -
/dev/dsk/c3t50020F23000078F1d0s0    163    mr    shsam1  -
/dev/dsk/c3t50020F23000078F1d0s1    164    mr    shsam1  -
```

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

# ハードウェア装置のアップグレードの準備

サーバーのアップグレード、新しいテープドライブの追加、自動ライブラリの追加、または既存の自動ライブラリへの別のドライブのインストールを行うときは、どの場合も、あらかじめ計画することが大切です。この節では、お使いの環境の装置に対するハードウェアのアップグレードの準備について説明します。

アップグレードの前に、次の操作を行うことをお勧めします。

- ハードウェアの追加や変更によってサン新しいライセンスが必要になるかどうかを判別する。  
ライセンスのアップグレードが必要ない変更の例としては、メモリーの追加やディスクキャッシュの増加があります。ライセンスのアップグレードが必要な変更の例としては、自動ライブラリでのスロットの追加やサーバーモデルの変更があります。
- ハードウェアの製造元のインストールの指示をよく読む。Sun Solaris のシステム管理者用のマニュアルで、ハードウェア追加に関する箇所も読みます。
- 古い mcf ファイルと新しい mcf ファイルで装置番号を調べる。mcf ファイルの詳細は、mcf(4) のマニュアルページを参照してください。
- 手元にあるバックアップコピーに問題がないかどうかを確認する。データおよびメタデータのバックアップについては、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』で説明されている手順を参照してください。
  - Sun StorEdge QFS 環境では、qfsdump(1M) コマンドで、すべてのデータとメタデータのダンプが取得されます。この処理の詳細は、qfsdump(1M) のマニュアルページを参照してください。
  - Sun StorEdge SAM-FS および Sun SAM-QFS 環境では、samfsdump(1M) コマンドで、すべてのメタデータのダンプが取得されます。アーカイブする必要のあるすべてのファイルのアーカイブのコピーがあることを確認してください。Sun StorEdge SAM-FS または Sun SAM-QFS の各ファイルシステムで archive\_audit(1) コマンドを使用すると、アーカイブのコピーがないファイルを確認できます。次の例では、/sam がマウントポイントです。

```
# archive_audit /sam
```

- システムにユーザーをログインさせない。
- Sun StorEdge SAM-FS および Sun SAM-QFS 環境で、アーカイバが wait モードになるようにする。アップグレード中は、アーカイバは wait モードになっている必要があり、実行してはいけません。

次のいずれかの方法でアーカイバをアイドル状態にできます。

- /etc/opt/SUNWsamfs/archiver.cmd ファイルに wait 指示を指定する。  
wait 指示や archiver.cmd ファイルの詳細は、archiver.cmd(4) のマニュアルページを参照してください。
- samu(1M) オペレータユーティリティを使用する。
- 次のコマンドを実行する。

```
# samcmd aridle
```

詳細は、samcmd(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ために、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する場合があります。これは、mcf ファイルを更新し、samgrowfs(1M) コマンドを使用して実行します。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。

Sun StorEdge SAM-FS および Sun SAM-QFS 環境では、ディスクまたはパーティションの追加時に、履歴の装置番号が更新されることがあります。明示的に無効にしないかぎり、システムは履歴の装置番号を自動的に生成します。詳細は、historian(7) のマニュアルページを参照してください。

### ▼ ファイルシステムへのディスクキャッシュの追加

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については、74 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。(省略可能)  
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 ファイルを編集します。

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 StorEdge 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 mm が必要です。0 以上のデータパーティションを追加できます。

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。変更内容を保存して、エディタを終了します。

/etc/opt/SUNWsamfs/mcf ファイルでは、装置の識別名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
```

4. sam-fsd(1M) コマンドを入力して、mcf ファイルにエラーがないかどうかを確認します。

コマンドの例は次のとおりです。

```
# sam-fsd
```

sam-fsd(1M) コマンドの出力で、mcf ファイルのエラーが示された場合は、エラーを修正してから次の操作に進みます。

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

コマンドの例は次のとおりです。

```
# samd config
```

samd(1M) コマンドについては、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)` コマンドを入力します。  
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、新しい名前に対して `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。  
ファイルシステムの名前を変更した場合は、`mcf` ファイルに新しい名前を入力し、手順 4 および 5 の説明に従って、`sam-fsd(1M)` コマンドと `samd(1M) config` コマンドを実行します。
8. Sun StorEdge QFS 共有ファイルシステムのみで、各参加クライアントホストの `mcf` ファイルを編集して、メタデータサーバーの `mcf` ファイルと一致させます。

---

## ファイルシステムのディスクの交換

次の作業を実行する必要がある場合があります。

- 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

これらの作業を行うには、後に説明する手順に従って、ファイルシステムのバックアップと再作成を実行する必要があります。

### ▼ ファイルシステムのバックアップと再作成

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。

ソフトウェアによって異なりますが、ファイルは、`mcf`、`archiver.cmd`、`defaults.conf`、`samfs.cmd`、`inquiry.conf` などです。Sun StorEdge QFS および Sun StorEdge SAM-FS 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、ライブラリカタログ、履歴、共有ホストファイル、ネットワーク接続自動ライブラリ用のパラメタファイルに対応するバックアップコピーがあることを確認してください。

Sun StorEdge SAM-FS および Sun SAM-QFS 環境では、カタログファイルの名前と場所がわからない場合は、`vi(1)` コマンドや別の表示コマンドで `mcf` ファイルを調べて、`mcf` ファイルの最初の `rb` エントリを検索します。このエントリに、ライブラリカタログファイルの名前が指定されています。カタログファイルの場所が指定されていない場合は、デフォルトの場所 (`/var/opt/SUNWsamfs/catalog`) が使用されています。

## 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。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。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。ダンプファイルの作成方法については、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』を参照してください。

Sun StorEdge SAM-FS または Sun SAM-QFS ファイルシステムを使用している場合は、`samfsdump(1M)` を使用してダンプファイルを作成するときに、ファイルシステムでアーカイブされていないファイルが検出されると、警告が表示されるので注意してください。警告が表示された場合は、ファイルシステムをマウント解除する前に、それらのファイルをアーカイブする必要があります。

## 3. ファイルシステムをマウント解除します。

ファイルシステムのマウント解除については、74 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、名前を変更します。(省略可能)  
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。

5. /etc/opt/SUNWsamfs/mcf ファイルを編集します。

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。mcf ファイルを編集して、ディスクまたはパーティションを追加または削除します。新しいパーティションは、既存のディスクパーティションの後に追加する必要があります。変更内容を保存して、エディタを終了します。

Sun StorEdge 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 mm が必要です。0 以上のデータパーティションを追加できます。

/etc/opt/SUNWsamfs/mcf ファイルでは、装置の識別名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
```

6. sam-fsd(1M) コマンドを入力して、mcf ファイルにエラーがないかどうかを確認します。

コマンドの例は次のとおりです。

```
# sam-fsd
```

sam-fsd(1M) コマンドの出力で、mcf ファイルのエラーが示された場合は、エラーを修正してから次の操作に進みます。

詳細は、sam-fsd(1M) のマニュアルページを参照してください。

7. samd(1M) config コマンドを使用して、mcf ファイルの変更を伝達します。

コマンドの例は次のとおりです。

```
# 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) コマンドを使用して、新しいファイルシステムを作成します。

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)` コマンドを入力して、ファイルシステムをマウントします。  
Sun StorEdge QFS または Sun StorEdge SAM-FS 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)` コマンドを使用して、ファイルシステムのマウントポイントに移動します。
11. `samfsrestore(1M)` コマンドまたは `qfsrestore(1M)` コマンドを使用して、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 2 で作成したダンプファイルから復元します。  
これらのコマンドの使い方については、`samfsdump(1M)` または `qfsdump(1M)` のマニュアルページや、『Sun QFS, Sun SAM-FS, Sun SAM-QFS 障害回復マニュアル』を参照してください。
12. `restore.sh(1M)` スクリプトを使用して、オンラインになっていたすべてのファイルを書き込みます。  
このコマンドは、次の形式で使用します。

```
# restore.sh log_file mount_point
```

`log_file` では、`sammkfs(1M)` コマンドまたは `samfsrestore(1M)` コマンドによって作成されたログファイルの名前を指定します。

`mount_point` では、復元するファイルシステムのマウントポイントを指定します。

`restore.sh(1M)` スクリプトについては、`restore.sh(1M)` のマニュアルページを参照してください。

---

## ホストシステムのアップグレード

ファイルシステムに対して使用しているホストシステムをアップグレードする場合は、次の項目を考慮に入れてください。

- 既存のホストがまだ作動している間に、新しいホストへ移行することをお奨めします。こうすることで、新しいハードウェアプラットフォームのインストール、構成、およびアプリケーションに対するテストの時間が確保できます。
- 新しいホストシステムへの移行は、Sun StorEdge QFS または Sun StorEdge SAM-FS ソフトウェアの初回インストールと同様です。Sun StorEdge SAM-FS および Sun SAM-QFS の環境では、ソフトウェアを再インストールし、構成ファイル (特に、`mcf` ファイル、`/kernel/drv/st.conf` ファイル、`/etc/opt/SUNWsamfs/inquiry.conf` ファイル) を更新する必要があります。また、既存の `archiver.cmd` ファイルと `defaults.conf` ファイルを新しいシステムにコピーしたり、システムログを構成したりすることも必要です。

ソフトウェアを再インストールする場合は、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』のインストールの説明を参照してください。

- ライセンスキーの更新が必要になることがあります。ライセンスキーは CPU ホスト ID に関連付けられています。システムを交換すると、新しいライセンスが必要になります。
- 古いホストシステムの電源を切断する前に、手元にあるバックアップコピーに不備がないかどうかを確認します。場合によっては、このとき新しいダンプファイルを作成する必要があります。ダンプファイルを使用して、新しいサーバーでファイルシステムを再作成します。ダンプファイルの作成の詳細については、qfsdump(1M) または samfsdump(1M) のマニュアルページや、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』を参照してください。

---

## Solaris OS のアップグレード

この節では、Solaris OS のアップグレード方法について説明します。

- 90 ページの「Sun StorEdge SAM-FS または Sun SAM-QFS 環境で Solaris OS をアップグレードする」
- 93 ページの「Sun StorEdge QFS 環境で Solaris OS をアップグレードする」

### ▼ Sun StorEdge SAM-FS または Sun SAM-QFS 環境で Solaris OS をアップグレードする

Solaris OS レベルのアップグレードに必要な手順の多くは、Sun StorEdge SAM-FS または Sun SAM-QFS 環境のアップグレードに必要な手順と同じです。この手順の一部のステップでは、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』の手順を参照します。

#### 1. ソフトウェアアップグレードを用意します。

Sun StorEdge SAM-FS および Sun SAM-QFS ソフトウェアは、さまざまなレベルの Solaris OS をサポートしています。新しくアップグレードした Solaris OS に古い Sun StorEdge SAM-FS または Sun SAM-QFS ソフトウェアを再インストールしないでください。

また、現在インストールされているバージョンや、アップグレード後のレベルによっては、新しいソフトウェアライセンスが必要になる場合があります。

ご購入先に連絡して、ソフトウェアの新しいコピーを入手し、サイトで新しいライセンスが必要かどうかを確認してください。

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。

これらのファイルは、mcf、archiver.cmd、defaults.conf、samfs.cmd、inquiry.conf などです。Sun StorEdge SAM-FS および Sun SAM-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、ライブラリカタログ、履歴、ネットワーク接続自動ライブラリ用のパラメタファイルに対応するバックアップコピーがあることを確認してください。

カタログファイルの名前と場所がわからない場合は、vi(1) コマンドや別の表示コマンドで mcf ファイルを調べて、mcf ファイルの最初の rb エントリを検索します。このエントリに、ライブラリカタログファイルの名前が指定されています。カタログファイルの場所が指定されていない場合は、デフォルトの場所 (/var/opt/SUNWsamfs/catalog) が使用されています。

3. 影響を受け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。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。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。

Sun StorEdge SAM-FS または Sun SAM-QFS ファイルシステムを使用している場合は、samfsdump(1M) を使用してダンプファイルを作成するときに、ファイルシステムでアーカイブされていないファイルが検出されると、警告が表示されるので注意してください。警告が表示された場合は、ファイルシステムをマウント解除する前に、それらのファイルをアーカイブする必要があります。

4. 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については、74 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. samd(1M) stop コマンドを実行して、すべてのアーカイブおよび SAM 操作を停止します。(省略可能)

Sun StorEdge SAM-FS または Sun SAM-QFS ファイルシステムをマウント解除する場合は、この操作を実行します。コマンドの例は次のとおりです。

```
# samd stop
```

6. 既存の Sun StorEdge SAM-FS または Sun SAM-QFS ソフトウェアを削除します。

pkgrm(1M) コマンドを使用して、既存のソフトウェアを削除します。新しいパッケージまたは新しいオペレーティングシステムレベルをインストールする前に、既存の Sun StorEdge SAM-FS および Sun SAM-QFS パッケージをすべて削除する必要があります。

たとえば、次のコマンドは、Sun StorEdge SAM-FS または Sun SAM-QFS 環境で、SUNWsamtp、SUNWsamfsu、および SUNWsamfsr パッケージを削除します。SUNWsamfs パッケージは、最後に削除する必要があります。SUNWsamtp パッケージはオプションのツールパッケージなので、システムにインストールされていない場合もあることに注意してください。pkgrm(1M) コマンドの例を次に示します。

```
# pkgrm SUNWsamtp SUNWsamfsu SUNWsamfsr
```

この操作の情報は、リリースレベル 4.2 以上でソフトウェアパッケージを削除していると仮定しています。ソフトウェアパッケージ名はリリース 4.2 で変更しています。システムに以前のリリースのソフトウェアパッケージがある場合は、その削除について、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』を参照してください。

## 7. Solaris OS のアップグレード

インストールしている OS レベルの Sun Solaris アップグレード手順を使用して、新しい Solaris OS のバージョンをインストールします。

### 8. 手順 1 で取得した SUNWsamfsr パッケージと SUNWsamfsu パッケージを追加します。

Sun StorEdge SAM-FS および Sun SAM-QFS のソフトウェアパッケージでは、ソフトウェアの追加や削除に Solaris OS パッケージユーティリティを使用します。ソフトウェアパッケージを変更するには、スーパーユーザーでログインする必要があります。pkgadd(1M) コマンドでは、Sun StorEdge SAM-FS および Sun SAM-QFS パッケージをアップグレードするために必要なさまざまな処理の確認を要求されません。

Sun StorEdge SAM-FS および Sun SAM-QFS のパッケージは、インストール CD-ROM の /cdrom/cdrom0 ディレクトリにあります。

次のように、pkgadd(1M) コマンドを実行してパッケージをアップグレードします。メッセージが表示されたら「yes」と入力します。

```
# pkgadd -d .SUNWsamfsr SUNWsamfsu
```

インストール中に、矛盾するファイルの存在がシステムによって検出されると、インストールを継続するかどうかの確認を求められます。別のウィンドウを開いて、ファイルをコピーして他の場所に保存できます。

### 9. ライセンスキーを更新します。(省略可能)



既存の Sun StorEdge SAM-FS および Sun SAM-QFS ソフトウェアリビジョンと、アップグレード先のリビジョンによって、ソフトウェアの新しいライセンスキーの入手が必要になることがあります。新しいライセンスが必要かどうかについては、ご購入先にお問い合わせください。

4.2 よりも前のリリースからアップグレードしている場合は、次のファイルに新しいライセンスキーを指定する必要があります。

```
/etc/opt/SUNWsamfs/LICENSE.4.2
```

詳細は、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』のライセンスに関する情報を参照してください。

#### 10. ファイルシステムをマウントします。(省略可能)

/etc/vfstab ファイルを変更して「Mount at Boot」フィールドで「yes」を指定していない場合は、この操作を実行する必要があります。

mount(1M) コマンドを使用して、ファイルシステムをマウントし、アップグレードした Sun SAM-FS または Sun SAM-QFS ソフトウェアの操作を続行します。

次の例で、samfs1 はマウントするファイルシステムの名前です。

```
# mount samfs1
```

## ▼ Sun StorEdge QFS 環境で Solaris OS をアップグレードする

Solaris OS レベルのアップグレードに必要な手順の多くは、Sun StorEdge QFS 環境のアップグレードに必要な手順と同じです。この手順の一部のステップでは、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』の手順を参照します。

#### 1. ソフトウェアアップグレードを用意します。

Sun StorEdge QFS ソフトウェアは、さまざまなレベルの Solaris OS をサポートしています。新しくアップグレードした Solaris OS に古い Sun StorEdge QFS ソフトウェアを再インストールしないでください。

また、現在インストールされているバージョンや、アップグレード後のレベルによっては、Sun StorEdge QFS の新しいライセンスが必要になる場合があります。

ご購入先に連絡して、ソフトウェアの新しいコピーを入手し、サイトで新しいライセンスが必要かどうかを確認してください。

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。

ファイルは、mcf、defaults.conf、samfs.cmd、共有ホストファイルなどです。Sun StorEdge 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. 影響を受け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。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。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。ダンプファイルの作成方法については、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』を参照してください。

4. 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については、74 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. 既存の Sun StorEdge QFS ソフトウェアを削除します。

pkgrm(1M) コマンドを使用して、既存のソフトウェアを削除します。新しいパッケージまたは新しいオペレーティングシステムレベルをインストールする前に、既存の Sun StorEdge QFS パッケージを削除する必要があります。

たとえば、次のコマンドでは Sun StorEdge QFS ソフトウェアが削除されます。

```
# pkgrm SUNWqfsu SUNWqfsr
```

この操作の情報は、リリースレベル 4.2 以上でソフトウェアパッケージを削除していると仮定しています。ソフトウェアパッケージ名はリリース 4.2 で変更しています。システムに以前のリリースのソフトウェアパッケージがある場合は、その削除について、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』を参照してください。

6. Solaris OS のアップグレード

インストールしている Solaris OS レベルの Sun Solaris アップグレード手順を使用して、新しい Solaris OS のバージョンをインストールします。

7. 手順 1 で取得したパッケージを追加します。

Sun StorEdge QFS のソフトウェアパッケージでは、ソフトウェアの追加や削除に Solaris OS パッケージユーティリティを使用します。ソフトウェアパッケージを変更するには、スーパーユーザーでログインする必要があります。pkgadd(1M) コマンドでは、Sun StorEdge QFS パッケージのアップグレードに必要な処理を確認するメッセージが表示されます。

Sun StorEdge QFS のパッケージは、インストール CD-ROM の /cdrom/cdrom0 ディレクトリにあります。

pkgadd(1M) コマンドを実行してパッケージをアップグレードします。メッセージが表示されたら「yes」と入力します。

```
# pkgadd -d .SUNWqfsr SUNWqfsu
```

インストール中に、矛盾するファイルの存在がシステムによって検出されると、インストールを継続するかどうかの確認を求められます。残すファイルは、別のウィンドウで別の場所にコピーできます。

## 8. ライセンスキーを更新します。(省略可能)

既存の Sun StorEdge QFS ソフトウェアリビジョンと、アップグレード先のリビジョンによって、Sun StorEdge QFS ソフトウェアの新しいライセンスキーの入手が必要になることがあります。新しいライセンスが必要かどうかについては、ご購入先にお問い合わせください。

4.2 よりも前の Sun StorEdge QFS リリースからアップグレードしている場合は、次のファイルに新しいライセンスキーを指定する必要があります。

```
/etc/opt/SUNWsamfs/LICENSE.4.2
```

詳細は、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』のライセンスに関する情報を参照してください。

## 9. mcf ファイルを更新します。(省略可能)

装置名を変更した場合は、新しい装置名と一致するように mcf ファイルを更新する必要があります。新しい装置名を検証し、62 ページの「システムに対する構成ファイルの変更の伝達」の手順を行います。

## 10. ファイルシステムをマウントします。(省略可能)

/etc/vfstab ファイルを変更して「yes」を指定していない場合は、この操作を実行する必要があります。

70 ページの「ファイルシステムのマウント」で説明する手順を使用します。アップグレードした Sun StorEdge QFS ソフトウェアで操作を続けます。



# Sun StorEdge QFS 共有ファイルシステム

Sun StorEdge QFS 共有ファイルシステムは、複数の Solaris オペレーティングシステム (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 StorEdge 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作動し、その他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 StorEdge 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 StorEdge 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、ローカルパス入出力 (「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。

この章では、Sun StorEdge 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。この章には、以下の節があります。

- 98 ページの「概要」
- 98 ページの「Sun StorEdge QFS 共有ファイルシステムの構成」
- 99 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」
- 102 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」
- 105 ページの「Sun StorEdge QFS 共有ファイルシステムのマウントとマウント解除」
- 106 ページの「クライアントホストの追加と削除」
- 119 ページの「Sun StorEdge QFS 環境でのメタデータサーバーの変更」
- 122 ページの「デーモン」
- 123 ページの「Sun StorEdge QFS 共有ファイルシステムのマウントオプション」
- 129 ページの「Sun StorEdge QFS 共有ファイルシステムのマウントの意味」
- 130 ページの「Sun StorEdge QFS 共有ファイルシステムでのファイルロック」
- 130 ページの「失敗またはハングアップした `sammkfs(1M)` コマンドまたは `mount(1M)` コマンドの障害追跡」

---

## 概要

Sun StorEdge QFS 共有ファイルシステムでは、アーカイブや書き込みは行われなため、アーカイブメディアとのネットワーク接続は必要ありません。Sun StorEdge QFS を共有ファイルシステム環境で稼働している場合は、この章のアーカイブメディアに関する情報は無視してください。

Sun SAM-QFS 共有ファイルシステム環境の場合、メタデータサーバーにできる各ホストは、次のいずれかの同一のアーカイブメディアリポジトリに接続する必要があります。

- リムーバブルメディア装置 (テープまたは光磁気ドライブ) のあるライブラリ
- 1 つまたは複数のファイルシステム内のディスク領域

アーカイブメディアは、メタデータサーバーにできるホストごとに、`mcF` ファイルまたは `diskvols.conf` ファイルに指定する必要がある

Sun SAM-QFS 共有ファイルシステム環境では、アクティブなメタデータサーバーだけで、書き込み (`sam-stagerd`) とアーカイブ (`sam-archiverd`) のデーモンがアクティブになる。メタデータサーバーは、すべてのファイル要求の書き込み元のサーバーとして指定される

Sun Cluster 環境に、Sun SAM-QFS 共有ファイルシステムを構成できない。

この章では、Sun StorEdge QFS 共有ファイルシステムの保守方法について説明します。『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』の説明に従って、ホストシステムに Sun StorEdge QFS または Sun SAM-QFS ソフトウェアがインストールされていることが前提になっています。

---

注 – Sun StorEdge QFS 共有ファイルシステムは、Sun StorEdge SAM-FS (`ms` ファイルシステム) 環境に構成できません。

---

---

## Sun StorEdge QFS 共有ファイルシステムの構成

Sun StorEdge 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 StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』の説明に従ってください。この章の多くの例では、このガイド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。

---

# 非共有ファイルシステムから共有ファイルシステムへの変換

非共有 Sun StorEdge QFS ファイルシステムを Sun StorEdge QFS 共有ファイルシステムに変換するには、以下の手順をこの順序で実行します。

## ▼ 非共有ファイルシステムを共有ファイルシステムに変換する

1. 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。

この手順を実行するためには `root` の権限が必要です。

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`、`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、ライブリカatalog、履歴、ネットワーク接続自動ライブラリ用のパラメータファイルに対応するバックアップコピーがあることを確認してください。

Sun SAM-QFS 環境では、カタログファイルの名前と場所がわからない場合、`mcf` ファイルの **Additional Parameters** フィールドの自動ライブラリを参照してください。**Additional Parameters** フィールドが空の場合、システムはデフォルトパスの `/var/opt/SUNWsamfs/catalog/catalog_name` を使用します。カタログファイルの場所の詳細は、`mcf(4)` のマニュアルページを参照してください。

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。(省略可能)

既存の Sun StorEdge QFS または Sun SAM-QFS のファイルシステムから新しい Sun StorEdge QFS 共有ファイルシステムにファイルを移動する場合は、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。

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

*fsname* には、新しい Sun StorEdge 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。コマンドの例は次のとおりです。

```
# sammkfs -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。

コマンドの例は次のとおりです。

コード例 5-1 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/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
```

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。

コマンドの例は次のとおりです。

コード例 5-2 `/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。

コマンドの例は次のとおりです。

コード例 5-3 Sun StorEdge 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 Not Server
# Name Addresses Priority Used Host
# ----
host1 172.16.0.129,titan.xyzco.com 1 - server
host2 172.16.0.130,tethys.xyzco.com 2 -
```



9. `samsharefs(1M) -u -R fsname` 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。

コマンドの例は次のとおりです。

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

10. `samd(1M) config` コマンドを実行します。

これで、`sam-fsd` デーモンに構成変更を知らせます。コマンドの例は次のとおりです。

```
# samd config
```

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。

## ▼ 非共有ファイルシステムを各クライアントで共有に変換する

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。

コマンドの例は次のとおりです。

```
# mkdir /sharefs1
```

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

コード例 5-4      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
```

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。(省略可能)

既存の Sun StorEdge QFS または Sun SAM-QFS のファイルシステムから新しい Sun StorEdge QFS 共有ファイルシステムにファイルを移動する場合は、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

ツ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。
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。  
コマンドの例は次のとおりです。

コード例 5-5 `/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
```

6. `/etc/opt/SUNWsamfs/hosts.fsname` ホスト構成ファイルを作成します。  
コマンドの例は次のとおりです。

コード例 5-6 Sun StorEdge 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  Not  Server
# Name  Addresses              Priority Used Host
# ----  -
host1   172.16.0.129,titan.xyzco.com  1      -   server
host2   172.16.0.130,tethys.xyzco.com  2      -
```

---

## 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 StorEdge QFS 共有ファイルシステムを非共有 Sun StorEdge QFS ファイルシステムに変換するには、以下の手順をこの順序で実行します。

### ▼ 共有ファイルシステムを各クライアントで非共有に変換する

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。

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。  
これで、`sam-fsd` デーモンに構成変更を知らせます。コマンドの例は次のとおりです。

```
# samd config
```

5. ファイルシステムのマウントポイントを削除します。

## ▼ 共有ファイルシステムをサーバーで非共有に変換する

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。  
この手順を実行するためには `root` の権限が必要です。
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`、`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、ライブラリカタログ、履歴、ネットワーク接続自動ライブラリ用のパラメタファイルに対応するバックアップコピーがあることを確認してください。

Sun SAM-QFS 環境では、カタログファイルの名前と場所がわからない場合は、`vi(1)` などの表示コマンドで `mcf` ファイルを調べて、自動ライブラリのエントリを検索します。各ライブラリのカタログファイルのパスは、**Additional Parameters** フィールドにあります。**Additional Parameters** フィールドが空の場合、システムはデフォルトパスの `/var/opt/SUNWsamfs/catalog/catalog_name` を使用します。カタログファイルの場所の詳細は、`mcf(4)` のマニュアルページを参照してください。

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。(省略可能)  
既存の Sun StorEdge QFS 共有または Sun SAM-QFS 共有ファイルシステムから新しい Sun StorEdge QFS 共有ファイルシステムにファイルを移動する場合は、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。
5. `sammkfs(1M) -F -U fsname` を実行して、Sun StorEdge 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。

*fsname* には、新しい非共有ファイルシステムに変換する Sun StorEdge QFS 共有ファイルシステムのファミリセット名を指定します。コマンドの例は次のとおりです。

```
# sammkfs -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。

コマンドの例は次のとおりです。

コード例 5-7      ファイルシステム `samfs1` の `mcf` ファイル

# Equipment	Eq	Eq	Family	Dev	Add
# Identifier	Ord	Type	Set	State	Params
# -----	---	---	---	---	---
<code>samfs1</code>	10	ma	<code>samfs1</code>	on	
<code>/dev/dsk/c2t50020F23000065EEd0s6</code>	11	mm	<code>samfs1</code>	on	
<code>/dev/dsk/c7t50020F2300005D22d0s6</code>	12	mr	<code>samfs1</code>	on	
<code>/dev/dsk/c7t50020F2300006099d0s6</code>	13	mr	<code>samfs1</code>	on	
<code>/dev/dsk/c7t50020F230000651Cd0s6</code>	14	mr	<code>samfs1</code>	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。

コマンドの例は次のとおりです。

コード例 5-8      `/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
samfs1    -                /samfs1  samfs    -        no
```

8. `/etc/opt/SUNWsamfs/hosts.fsname` 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。  
これで、`sam-fsd` デーモンに構成変更を知らせます。コマンドの例は次のとおりです。

```
# samd config
```

10. `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。

---

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

Sun StorEdge 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、Solaris OS のマウントまたはマウント解除の順序で行うことが重要です。

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(4)` 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 StorEdge QFS 共有ファイルシステムのマウントの詳細については、123 ページの「Sun StorEdge 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。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、61 ページの「ファイルシステムの操作」を参照してください。

## ▼ Sun StorEdge QFS 共有ファイルシステムをマウントする

`mount(1M)` コマンドによって、Sun StorEdge QFS 共有ファイルシステムが Solaris OS にマウントされます。`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。

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。

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

75 ページの「Sun StorEdge QFS および Sun SAM-QFS 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。

# クライアントホストの追加と削除

この後の節では、クライアントホストシステムの追加や削除について説明します。

- 106 ページの「クライアントホストを追加する」
- 117 ページの「クライアントホストを削除する」

## ▼ 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 StorEdge 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。詳細は、『Sun Cluster System Administration Guide for Solaris OS』を参照してください。

各クライアントホストで、次の手順に従ってください。

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 StorEdge 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 StorEdge 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。コマンドの例は次のとおりです。

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

- Sun StorEdge 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どれかから、`-R` オプションを付けて `samsharefs(1M)` コマンドを実行します。コマンドの例は次のとおりです。

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

`samsharefs(1M)` コマンドを実行できるのは、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だけです。詳細は、`samsharefs(1M)` のマニュアルページを参照してください。

---

注 – ファイルシステムのマウント解除時には、任意の潜在的なメタデータサーバーでホスト情報を変更できます。サンでは、常にホスト情報を取り出すことで、ホスト情報を最新状態に保つことをお勧めします。

---

3. vi(1) などのエディタを使用して、Sun StorEdge 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 5-9 にこの手順を示します。

コード例 5-9 編集前の hosts.sharefs1

```
# vi /etc/opt/SUNWsamfs/hosts.sharefs1
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host      Host IP                Server  Not  Server
# Name      Addresses                Priority Used Host
# ----      -
titan      172.16.0.129,titan.xyzco.com  1      -   server
tethys     172.16.0.130,tethys.xyzco.com  2      -
mimas      mimas.xyzco.com              -      -
dione      dione.xyzco.com              -      -
```

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 5-10 は、helene の行を最終行として追加した後のファイルを示しています。

コード例 5-10 編集した hosts.sharefs1

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host      Host IP                Server  Not  Server
# Name      Addresses                Priority Used Host
# ----      -
titan      172.16.0.129,titan.xyzco.com  1      -   server
tethys     172.16.0.130,tethys.xyzco.com  2      -
mimas      mimas.xyzco.com              -      -
dione      dione.xyzco.com              -      -
helene     helene.xyzco.com            -      -
```

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 StorEdge 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。

- Sun StorEdge 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) -u` コマンドを実行します。コマンドの例は次のとおりです。

```
# samsharefs -u sharefs1
```

- Sun StorEdge 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、アクティブなメタデータサーバーまたはいずれかの潜在的なメタデータサーバーから、`samsharefs(1M) -R -u` コマンドを実行します。コマンドの例は次のとおりです。

```
# 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. クライアントホストのどれかに、スーパーユーザーでログインします。
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にアクセスまたはマウントするには、その共有ファイルシステムが `mcf` ファイルで定義されている必要があります。

`vi(1)` などのエディタを使用して、クライアントホストシステムのどれかで `mcf` ファイルを編集します。Sun StorEdge QFS 共有ファイルシステムに組み込むすべてのクライアントホストの `mcf` ファイルを更新する必要があります。ファイルシステムとディスクの宣言情報では、ファミリーセット名、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の `mcf` ファイルにも `shared` キーワードを指定する必要があります。ただし、コントローラの割り当てがホストごとに変更できるのと同様に、装置名も変更できます。

`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 StorEdge 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。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。



例 1: コード例 5-11 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であり、`titan` と同じメタデータディスクに接続していることに注意してください。

コード例 5-11 `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 5-12 に、編集後の `mcf` ファイルを示します。

コード例 5-12 `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
```

コード例 5-12 では、装置番号が、mcf ファイルのメタデータサーバー titan のものと一致することに注意してください。これらの装置番号は、クライアントホスト tethys またはその他のクライアントホストですすでに使用されている必要があります。

例 2: コード例 5-13 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。mimas はメタデータサーバーにできないことと、メタデータディスクに接続していないことに注意してください。

コード例 5-13 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。装置がないため、samfsconfig(1M) コマンドでは、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する。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ごとに 1 つまたは複数の nodev 行を作成する。これらの行では、アクセスできない各装置の Equipment Identifier フィールドに nodev キーワードを指定する必要がある。この例では、存在しないメタデータディスクを表すために mcf ファイルに nodev というデバイスエントリを作成する
- 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 5-14 に、できあがった mimas の mcf ファイルを示します。

コード例 5-14 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
```

コード例 5-14 クライアントホスト mimas の mcf ファイル (続き)

```
/dev/dsk/c1t50020F2300005D22d0s6 12 mr sharefs1 on  
/dev/dsk/c1t50020F2300006099d0s6 13 mr sharefs1 on  
/dev/dsk/c1t50020F230000651Cd0s6 14 mr sharefs1 on
```

注 – Sun SAM-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 ファイルを必要に応じて必ず更新してください。

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。

これで、`sam-fsd` デーモンに構成変更を知らせます。コマンドの例は次のとおりです。

```
# samd config
```

10. 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。(省略可能)

Sun StorEdge 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、メタデータサーバーとクライアントホストがファイルシステムのアクセスに使用できる、ホストインタフェースを定義されます。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.fsname.local
```

*fsname* には、Sun StorEdge QFS 共有ファイルシステムのファミリセット名になります。

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 (#) 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せん。

表 5-1 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 5-1 ローカルホストの構成ファイルのフィールド

フィールド番号	内容
1	<b>Host Name</b> フィールド。このフィールドには、Sun StorEdge 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) を指定する。
2	<b>Host Interfaces</b> フィールド。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する。このフィールドは、 <code>ifconfig(1M) -a</code> コマンドの出力を使用して作成できる。次のどれかの方法で個々のインタフェースを指定できる。 <ul style="list-style-type: none"><li>ドット付き 10 進数 (dotted-decimal) の IP アドレス形式</li><li>IP バージョン 6 の 16 進数のアドレス形式</li><li>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</li></ul> 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。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る (このアドレスは共有ホストファイルにも含まれる)。

Sun StorEdge 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.fs_name` 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、次のプロセスでは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)。

1. 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。
2. 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、接続が成功するまで、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タスクを実行します。
  - i. メタデータサーバーの `/etc/opt/SUNWsamfs/hosts.fsname` 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。
  - ii. 両方のファイルに含まれるアドレスのリストを作成し、サーバーへの接続に成功するまで、これらのアドレスに接続しようとします。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：この例は、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』に示されている例を拡張したものです。コード例 5-15 は、4つのホストのあるホストファイルの例です。

コード例 5-15 Sun StorEdge 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	Not	Server	
# Name	Addresses	Priority	Used	Host	
#	----	-----	----	----	----
titan	172.16.0.129,titan.xyzco.com	1	-	server	
tethys	172.16.0.130,tethys.xyzco.com	2	-		
mimas	mimas.xyzco.com	-	-		
dione	dione.xyzco.com	-	-		

図 5-1 に、これらのシステムのインタフェースを示します。

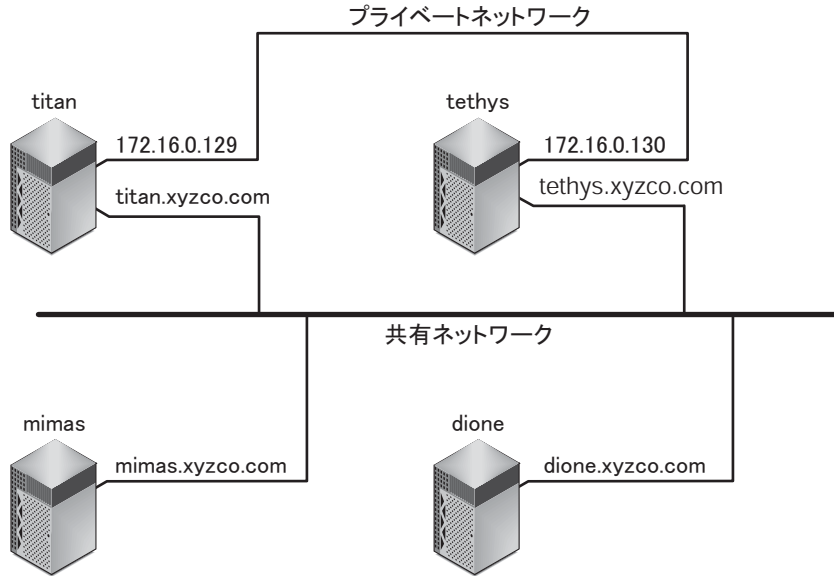


図 5-1 ネットワークインタフェース

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 5-16 は、これらのファイルの内容を示しています。

コード例 5-16 titan と tethys の両方の 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。これらが titan および tethys の共有インタフェースで titan および tethys と接続され、titan または tethys の到達不能なプライベートインタフェースに接続し

ようとしなことを保証するため、システム管理者は mimas および dione の /etc/opt/SUNWsamfs/hosts.sharefs1.local ファイルと同一のコピーを作成してあります。コード例 5-17 に、これらのファイルの内容を示します。

コード例 5-17 mimas と dione の両方の hosts.sharefs1.local ファイル

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

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

これで、`sam-fsd` デーモンに構成変更を知らせます。コマンドの例は次のとおりです。

```
# samd config
```

#### 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 5-18 に示す `ps(1)` および `grep(1)` コマンドを使用します。

コード例 5-18 `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 5-18 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。このような場合は、この次の手順に進むことができます。ただし、システムに戻された出力で、`sam-sharefsd` デーモンが Sun StorEdge 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、138 ページの「ハングアップした `mount(1M)` コマンドの回復」で説明するいくつかの診断手順を実行します。

#### 13. マウントポイントが存在しない場合は、新しい Sun StorEdge QFS 共有ファイルシステム用のマウントポイントを作成します。

`mkdir(1)` コマンドを使用して、マウントポイント用のディレクトリを作成します。コマンドの例は次のとおりです。

```
# 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。

コマンドの例は次のとおりです。

```
# 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 StorEdge QFS 共有ファイルシステムのエントリを指定する必要があります。Mount Parameters フィールドに `shared` を指定します。

起動時に Sun StorEdge 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、`/etc/vfstab` システムに次の変更を加えます。

- `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。

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。どちらの場合も、コード例 5-19 に示すように `Mt params` フィールドの `shared` は必須エントリです。

コード例 5-19 `/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。

コマンドの例は次のとおりです。

```
# df -k
```



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

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(4)` 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 StorEdge QFS 共有ファイルシステムのマウントの詳細については、123 ページの「Sun StorEdge 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。

コマンドの例は次のとおりです。

```
# mount /sharefs1
```

## ▼ 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。

---

参考 – `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、最初のクライアントホストの Sun StorEdge QFS 共有ファイルシステムをマウント解除します。

Sun StorEdge QFS 共有ファイルシステムがマウントされているすべてのクライアントホストで、この操作を繰り返します。

コマンドの例は次のとおりです。

```
client# umount sharefs1
```

3. `umount(1M)` コマンドを使用して、Sun StorEdge 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。

コマンドの例は次のとおりです。

```
metaserver# umount sharefs1
```

4. まだログインしていない場合は、Sun StorEdge 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 StorEdge 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 5-20 は、クライアントホストを削除する前のファイルを示しています。

コード例 5-20 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 IP                Server  Not  Server
# Name      Addresses                Priority Used Host
# ----      -
titan       172.16.0.129,titan.xyzco.com  1      -    server
tethys      172.16.0.130,tethys.xyzco.com  2      -
mimas       mimas.xyzco.com             -      -
dione       dione.xyzco.com             -      -
helene      helene.xyzco.com            -      -
```

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 5-21 は、`helene` の行を削除した後のファイルを示しています。

コード例 5-21 クライアントホストを削除した後の `hosts.sharefs1`

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host      Host IP                Server  Not  Server
# Name      Addresses                Priority Used Host
# ----      -
titan       172.16.0.129,titan.xyzco.com  1      -    server
tethys      172.16.0.130,tethys.xyzco.com  2      -
mimas       mimas.xyzco.com             -      -
dione       dione.xyzco.com            -      -
```

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。

コマンドの例は次のとおりです。

```
# 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。  
コマンドの例は次のとおりです。

```
# samsharefs -R sharefs1
```

10. `mount(1M)` コマンドを使用して、Sun StorEdge 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。
11. `mount(1M)` コマンドを使用して、Sun StorEdge QFS 共有ファイルシステムを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 StorEdge QFS 環境でのメタデータサーバーの変更

この節では、Sun Cluster 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 StorEdge 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。

---

**注** – Sun SAM-QFS 環境のメタデータサーバーの変更で不明な点がある場合は、サンにお問い合わせください。

---

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。

- 120 ページの「使用可能なメタデータサーバーを変更する」
- 120 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

この手順では、メタデータサーバーが使用可能なときに、Sun StorEdge QFS 環境で Sun StorEdge QFS 共有ファイルシステムのメタデータサーバーを変更する方法について説明します。

- 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。

コマンドの例は次のとおりです。

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

## ▼ 使用不可のメタデータサーバーを変更する

この手順では、既存のメタデータサーバーが使用不可のときに、Sun StorEdge QFS 環境で Sun StorEdge QFS 共有ファイルシステムのメタデータサーバーを変更する方法について説明します。

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、再書き込みできないようにする) ことです。

コード例 5-22 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 5-22      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]::c            # Forces a dump  
kadb[1]:$q            # Exits the debugger for prom
```

コード例 5-23 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 5-23      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。



---

**注意** – 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後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。サーバーを停止するために、以下のどの方法も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
- それまでの停止したメタデータサーバーへの `go` (続行) 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ないで、サーバーで `:c` (続行) を実行してください。変更を行うと、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。

---

## 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。

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必要があるため、待機する必要があります。新しいメタデータサーバーで次のようなコマンドを実行します。

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

リース時間が期限切れかどうか分からない場合は、`samu(1M) N` の表示を使用します。`samu(1M)` については、149 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、125 ページの「Sun StorEdge 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. ファイルシステムをマウント解除します。(省略可能)

このステップは、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。

105 ページの「Sun StorEdge 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. `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。(省略可能)

このステップは、この時点で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。

Sun StorEdge QFS または Sun SAM-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、ファイルの長さが増え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、クライアントが実行しようとする、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。

---

## デーモン

Sun StorEdge 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、Sun StorEdge QFS 共有ファイルシステムに構成されている各マウントポイントでは、1 つの `sam-sharefsd` デーモンがアクティブになっています。

`sam-fsd` デーモンは、Sun StorEdge QFS 共有ファイルシステムを認識すると、共有ファイルシステムデーモン (`sam-sharefsd`) を起動します。サーバー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。メタデータサーバー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。

1 つの Sun StorEdge QFS 共有ファイルシステムのデーモンは、各クライアントホストの Sun StorEdge QFS 共有ファイルシステムの共有マウントポイントごとに起動されます。このデーモンによって、メタデータサーバーへの接続が確立されます。メタデータサーバーの `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 StorEdge QFS のインストール時に、`sam-qfs` のエントリが `/etc/services` に自動的に追加されますが、このエントリは削除する必要があります。共有ファイルシステムポートは `/etc/inet/services` ファイルに定義されます。`/etc/inet/services` ファイルにインストールされるポート番号は 7105 です。このポートが、他のサービスと競合しないことを確認してください。

---

**注** – Sun StorEdge QFS 4.2 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。

---

## Sun StorEdge QFS 共有ファイルシステムのマウントオプション

Sun StorEdge 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。この章では、多くのオプションについて、その役割ごとに説明しています。ただし、オプションの中には、特定の状況だけで使用できるものもあります。この節では、特別な目的のために使用するマウントオプションについて説明します。

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(4) ファイルに記述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 StorEdge 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、149 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。

この後の項に、Sun StorEdge 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。

### 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、その後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** – Sun StorEdge 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 StorEdge 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 StorEdge 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 StorEdge 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(4) 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 StorEdge 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：minallopsz=*n* および maxallopsz=*n* オプション

mount(1M) コマンドの -o minallopsz=*n* および -o maxallopsz=*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 minallopsz=*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 maxallopsz=*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 StorEdge 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 5-2 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 5-2 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) を指定する。
-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) を指定する。
-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) を指定する。

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こと、リースが期限切れになるごとに更新する必要があるため、クライアントホストとメタデータサーバーの間のトラフィックが増加します。Sun Cluster 環境でのリース時間についての詳細は、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』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 StorEdge 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 StorEdge 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、wrelease マウントオプションの時間によって決まります。Sun StorEdge 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 5-3 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 5-3 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 mh_write が無効	メタデータサーバーで mh_write が有効
複数の読み取りホストが許可される ページ入出力を使用できる	複数の読み取りホストが許可される ページ入出力を使用できる
1つの書き込みホストだけが許可される ページ入出力を使用できる その他のホストは待機する	複数の読み取りおよび書き込みホストが許可される 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る。
1つの追加ホストだけが許可される その他のホストは待機する	1つの追加ホストだけが許可される その他のホストは読み取りまたは書き込みができる 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る。

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できる

- `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ない

Sun StorEdge 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許可する「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` がない場合に他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 StorEdge 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、他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 StorEdge 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他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。

## 並行スレッド数の設定：`nstreams=n` オプション

`nstreams=n` マウントオプションでは、Sun StorEdge QFS 共有ファイルシステムの並行スレッドの数が設定されます。デフォルトでは、`nstreams=256` です。デフォルト設定では、256 個までの操作は同時に処理されますが、257 番目の操作は他の操作が終了してから開始します。`nstreams=n` マウントオプションは、Sun StorEdge QFS 共有ファイルシステムの動作状態に応じて調整できます。 $n$  には、 $76 \leq n \leq 1024$  の値を指定します。

## 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 15 秒ごとに更新します。たとえば、新しくいくつかのファイルが作成された Sun StorEdge 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、15 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 StorEdge 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、9 ページの「ファイルシステム設計」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 StorEdge 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 StorEdge QFS 共有ファイルシステムは、メタデータをバッファ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。

---

# Sun StorEdge QFS 共有ファイルシステムのマウントの意味

Sun StorEdge 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 StorEdge 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.
```

---

## Sun StorEdge QFS 共有ファイルシステムでのファイルロック

必須のロックはサポートされていません。必須のロックを設定すると、EACCES エラーが戻されます。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。

---

## パフォーマンスについて

メタデータサーバーは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) のサイズを拡大すると、パフォーマンスを向上できます。これにより、クライアントが頻繁に多数のファイルを開くときに、パフォーマンスが向上します。このキャッシュサイズは、デフォルトの 2 倍または 3 倍にするとよいでしょう。

方法については、『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 `nssize` です。

---

## 失敗またはハングアップした `sammkfs(1M)` コマンドまたは `mount(1M)` コマンドの障害追跡

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。

## 失敗した `sammkfs(1M)` コマンドの回復

`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(4)` ファイルを確認し、`mcf(4)` 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(4)` ファイルを確認し、`mcf(4)` 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(4)` ファイルを確認します。

コマンドの例は次のとおりです。

```
# 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. `mcf(4)` ファイルを編集して、診断された問題を解決します。(省略可能)

`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と示されている場合は、この操作を実行します。

3. `sam-fsd(1M)` コマンドをもう一度使用して、`mcf(4)` ファイルを確認します。

`sam-fsd(1M)` コマンドの出力で `mcf(4)` 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(4)` ファイルの変更を伝達するために必要です。

コマンドの例は次のとおりです。

```
# samd config
```

## 失敗した `mount(1M)` コマンドの回復

`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、138 ページの「ハングアップした `mount(1M)` コマンドの回復」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する
- mount コマンドが失敗し、ファイルシステムと mcf(4) 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、次のことを確認する
  - mcf(4) ファイルの構文が有効かどうか。詳細は、131 ページの「mcf(4) ファイルを確認し、mcf(4) ファイルの変更をシステムに伝達する」を参照。
  - 最近 mcf(4) ファイル対してに行っ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、131 ページの「mcf(4) ファイルを確認し、mcf(4) ファイルの変更をシステムに伝達する」を参照。
  - mcf(4) 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(4) 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る。samfsconfig(1M) コマンドの使用方法については、136 ページの「samfsconfig(1M) コマンドを使用する」を参照してください。
- その他の理由で 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。
  - 132 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」
  - 134 ページの「samfsinfo(1M) コマンドと samsharefs(1M) コマンドを使用する」
  - 136 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、134 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 StorEdge 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 5-24 に出力例を示します。

### コード例 5-24      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 StorEdge 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 5-25 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 5-25      /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。コマンドの例は次のとおりです。

```
# unshare mountpoint
```

*mountpoint* には、Sun SAM-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(4) ファイルで指定した Sun StorEdge QFS 共有ファイルシステムの名前を指定します。コード例 5-26 に、samfsinfo(1M) コマンドと出力を示します。

コード例 5-26 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 5-26 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
```

次の行の後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(4)` ファイルで指定した Sun StorEdge QFS 共有ファイルシステムの名前を指定します。コード例 5-27 に、`samsharefs(1M)` コマンドと出力を示します。

### コード例 5-27 `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` 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。

`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、136 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、次の形式で使用します。

```
samfsconfig list_of_devices
```

*list\_of\_devices* には、*mcf(4)* 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 5-28 に、*mcf* ファイルに *nodev* エントリのないホストで実行した *samfsconfig(1M)* コマンドを示します。コード例 5-28 に、ホスト *tethys* の *mcf* ファイルを示します。

コード例 5-28      *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 5-29 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 5-29      *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   -
```

コード例 5-29     nodelv エントリがある場合の `samfsconfig(1M)` コマンドの例  
                  (続き)

```
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)` コマンドの最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、その後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。
2. `samu(1M)` コマンドを入力して、`samu(1M)` オペレータユーティリティを起動します。

コマンドの例は次のとおりです。

```
# samu
```

### 3. P を押して、Active Services 画面を表示します。

コード例 5-30 に P の表示を示します。

#### コード例 5-30      メタデータサーバーの P の表示

```
Active Services                               samu  4.2 09:02:22 Sept 22 2004

Registered services for host 'titan':
  sharedfs.sharefs1
  1 service(s) registered.
```

この出力を確認します。コード例 5-30 で、`sharedfs.filesystem-name` を含む行を調べます。この例では、行に `sharedfs.sharefs1` が含まれている必要があります。そのような行が表示されない場合は、`sam-fsd` と `sam-sharefsd` の両方のデーモンが起動していることを確認します。次の手順を行います。

#### a. `defaults.conf` ファイルでデーモントレースを有効にします。

トレースを有効にする方法については、`defaults.conf(4)` を参照するか、144 ページの「`sam-sharefsd` トレースログを調査する (省略可能)」の手順 2 を参照してください。

#### b. 構成ファイル、特に `/etc/opt/SUNWsamfs/mcf` を確認します。

#### c. 構成ファイルを検査し、デーモンが有効であることを確認してから、この手順をもう一度行います。

### 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 5-34 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 5-31      `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,titan.foo.com 1 - server
tethys 173.26.2.130,tethys.foo.com 2 -
dione dione.foo.com 0 -
mimas mimas.foo.com 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 5-32 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 5-32 サーバーでの netstat(1M) の例



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。
- そのホストに対して 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必要があります。

## 6. クライアントで netstat(1M) コマンドを入力します。

コード例 5-33 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 5-33 クライアントでの netstat(1M) コマンド



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、手順 7 に進んでください。

#### 7. 次のいずれか 1 つまたは複数の手順を行います。

- 141 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 (省略可能)」
- 143 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 (省略可能)」
- 144 ページの「sam-sharefsd トレースログを調査する (省略可能)」

### ▼ 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 (省略可能)

138 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、この手順を行います。

#### 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(4) ファイルで指定した Sun StorEdge QFS 共有ファイルシステムの名前を指定します。コード例 5-34 に、samsharefs(1M) -R コマンドを示します。

#### コード例 5-34 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,titan.xyzco.com 1 - server
tethys 173.26.2.130,tethys.xyzco.com 2 -
dione dione.foo.com 0 -
mimas mimas.foo.com 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、この出力が手順の後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。このようなファイルの構成についての詳細は、この章の前半の手順を参照してください。構成作業には、既存の mcf(4) ファイルの編集、mcf(4) ファイル変更の他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 5-35 は、ping(1M) コマンドの出力を示しています。

コード例 5-35 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.xyzco.com
titan.foo.com is alive
```

### 6. クライアントで、hosts.filesystem.local ファイルを調べます。(省略可能)

ping(1M) コマンドでアクセスできないホストが見つかった場合に、この操作を実行してください。

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後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 5-36 に、hosts.sharefs1.local ファイルを示します。

コード例 5-36 hosts.filesystem.local ファイルの確認

```
dione-client# cat /etc/opt/SUNWsamfs/hosts.sharefs1.local
titan          titan.xyzco.com          # no route to 173.26.2.129
tethys         tethys.xyzco.com         # no route to 173.26.2.130
```

## 7. 正しいサーバーインタフェースを有効にします。(省略可能)

アクセスできるサーバーインタフェースがないことが ping(1M) コマンドによって明らかになった場合は、サーバーのネットワークインタフェースを一般的な操作に対応するように構成または初期化するか、samsharefs(1M) コマンドを使用して、ホストファイル内のインタフェース名を実際の名前と一致するように更新する必要があります。

## ▼ 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 (省略可能)

138 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、この手順を行います。

### 1. samsharefs(1M) の出力を用意します。

141 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 (省略可能)」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 5-37 は、hostname(1M) コマンドおよび出力を示しています。

コード例 5-37 hostname(1M) の出力

```
dione-client# hostname
dione
```

4. 2 番目の列の各エントリに対してサーバーで ping(1M) コマンドを使用して、クライアントにアクセスできることを確認します。(省略可能)

hostname(1M) コマンドの出力が、samsharefs(1M) 出力の 2 列目の名前と一致する場合に、この操作を実行します。コード例 5-38 に、ping(1M) コマンドと出力を示します。

コード例 5-38 ping(1M) の出力

```
titan-server# ping dione.xyzco.com
dione.xyzco.com is alive
```

コード例 5-36 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. 正しいクライアントインタフェースを有効にします。(省略可能)

アクセスできるクライアントインタフェースがないことが ping(1M) コマンドによって明らかになった場合は、クライアントのネットワークインタフェースを一般的な操作に対応するように構成または初期化するか、samsharefs(1M) コマンドを使用して、ホストファイル内のインタフェース名を実際の名前と一致するように更新する必要があります。

## ▼ sam-sharefsd トレースログを調査する (省略可能)

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他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 StorEdge 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 StorEdge 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、mount(1M) の問題を解決できます。

- 138 ページの「ネットワーク接続を確認する」
- 141 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 (省略可能)」

- 143 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 (省略可能)」

これらの手順で問題を解決できなかった場合に、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 つ以上の他の手順を使用して問題を調査します。
2. **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。(省略可能)**

手順 1 でファイル `/var/opt/SUNWsamfs/trace/sam-sharefsd` が存在しなかった、あるいはこのファイルが最近変更されていなかった場合に、この手順を実行します。

- a. **defaults.conf ファイルのサンプルを、`/opt/SUNWsamfs/examples/defaults.conf` から `/etc/opt/SUNWsamfs` にコピーします。(省略可能)**

この時点で `defaults.conf` ファイルが `/etc/opt/SUNWsamfs` がない場合に、この手順を実行します。コード例 5-39 に手順を示します。

コード例 5-39 defaults.conf ファイルのコピー

```
# 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 `/opt/SUNWsamfs/examples/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 5-40 に、`defaults.conf` ファイルに追加する行を示します。

コード例 5-40 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(4)` ファイルを認識させます。

コマンドの例は次のとおりです。

```
# 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 5-41 に、`sam-fsd(1M)` コマンドの出力を示します。

コード例 5-41 `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 5-42 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 5-42 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
```





## 第6章

# samu(1M) オペレータユーティリティの使用法

この章では、Sun StorEdge QFS 環境または Sun StorEdge SAM-FS 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 StorEdge 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。

- 149 ページの「概要」
- 152 ページの「オペレータ表示」
- 197 ページの「オペレータ表示の状態コード」
- 199 ページの「オペレータ表示のデバイスの状態」
- 201 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。ユーティリティは、次の機能を備えています。

- Sun StorEdge QFS および Sun StorEdge SAM-FS 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 StorEdge QFS または Sun StorEdge SAM-FS 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 StorEdge QFS 環境または Sun StorEdge SAM-FS 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。

この後の節では、samu(1M) の起動と停止、ユーティリティとのやりとり、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。コマンドの例は次のとおりです。

```
# 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、150 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:
```

---

注 - samu(1M) のヘルプ画面とすべての画面表示も、同じ方法で表示できます。コロン文字を入力すると、Command プロンプトが表示されます。

---

## 2. 表示したい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 画面を表示したい場合は、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、162 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 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 StorEdge QFS 環境または Sun StorEdge SAM-FS 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 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。このヘルプ画面は、Sun StorEdge QFS と Sun StorEdge SAM-FS のどちらのファイルシステムを使用しているかによって異なります。

ヘルプ (h) 画面の詳細については、162 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 (A、C、F、I、J、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に設計されています。

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。コード例 6-1 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 6-1 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。

この後の節では、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。

## (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 6-1 に、a 表示で使用するコントロールキーを示します。

表 6-1 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 6-2 に、:a *filesystem* 表示で使用するコントロールキーを示します。

表 6-2 :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 6-2 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 6-2 samu(1M) a の表示

```
Archiver status                               samu 4.2 07:44:02 Sept 8 2004
License: License never expires.

sam-archiverd:  Waiting for resources

sam-arfind:  samfs1 mounted at /sam1
Waiting until 2004-05-08 07:54:02 to scan .inodes

sam-arfind:  samfs2 mounted at /sam2
Waiting until 2004-05-08 07:52:57 to scan .inodes

sam-arfind:  qfs1 mounted at /qfs1
Waiting until 2004-05-08 07:44:33 to scan .inodes

sam-arfind:  qfs2 mounted at /qfs2
```

コード例 6-2 samu(1M) a の表示 (続き)

```
Waiting until 2004-05-08 07:53:21 to scan .inodes

sam-arfind: qfs3 mounted at /qfs3
Waiting until 2004-05-08 07:44:11 to scan .inodes

sam-arfind: qfs4 mounted at /qfs4
Waiting until 2004-05-08 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 6-3 に、詳細表示のフィールドを示します。

表 6-3 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が残っていないことを意味する。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ない。
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
Directories	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 6-4 に、この表示で使用するコントロールキーを示します。

表 6-4 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 6-3 は、デバイス構成表示の例です。

コード例 6-3 samu(1M) c の表示

```
Device configuration:                samu      4.2 07:48:11 Sept
8 2004
License: License never expires.

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 6-5 で、この表示のフィールドを説明します。

表 6-5 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 <ul style="list-style-type: none"><li>• on - 装置をアクセスに利用可能</li><li>• ro - 装置は、読み込み専用アクセスだけに利用可能</li><li>• off - 装置をアクセスに利用できない。</li><li>• down - 装置は、メンテナンスアクセスだけに利用可能。</li><li>• idle - 装置は、新しい接続に利用できない。進行中の操作は、終了するまで続行される。</li></ul>
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。コマンドの例は次のとおりです。

```
Command: C 0x1044a998
```

この表示はデバッグ用です。サン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 6-4 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 6-4 samu(1M) C の表示

Memory	base: 0x1234567				samu 4.2 07:52:25 Sept 8 2004
License: License never expires.					
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.`\
00000100	02d05e21	40920080	037ff801	cc941020	.P^!@.....x.L..
00000110	10c45e20	48a600a0	80921000	137ff801	.D^ H& . . . . .x.
00000120	a7d05e21	40853420	00a00420	018528b0	'P^!@.4 . . . .(0
00000130	01a604e0	02840640	02d030a2	00853420	.&.`'...@.P0" . .4
00000140	0080a0a0	400a6fff	f6921000	13c65e23	.. @.o.v....F^#
00000150	58a01020	00c45e20	48c608e0	02d05e21	X . .D^ HF.`\P^!
00000160	40920080	037ff801	b5941020	20c45e20	@.....x.5.. D^
00000170	48a600a0	80921000	137ff801	90d05e21	H& . . . . .x..P^!
00000180	40853420	00a00420	018528b0	01a604e0	@.4 . . . .(0.&.`
00000190	02840640	02d030a2	80853420	0080a0a0	...@.P0" . .4 . .
000001a0	400a6fff	f6921000	13c65e23	58a01020	@.o.v....F^#X .
000001b0	00c45e20	48c608e0	02d05e21	40920080	.D^ HF.`\P^!@...
000001c0	037ff801	9e941020	30c45e20	48a600a0	..x.... 0D^ H&.
000001d0	80921000	137ff801	79d05e21	40853420	.....x.yP^!@.4
000001e0	00a00420	018528b0	01a604e0	02840640	. . . .(0.&.`'...@
000001f0	02d030a3	00853420	0080a0a0	400a6fff	.P0# . .4 . . @.o.

## (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 6-5 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 6-5 samu(1M) d の表示

```
Daemon trace controls                samu 4.2 07:56:38 Sept 8 2004
License: License never expires.

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
```

コード例 6-5 samu(1M) d の表示 (続き)

```
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
```

## (f) - ファイルシステムの表示

f 表示は、Sun StorEdge QFS ファイルシステムまたは Sun StorEdge SAM-FS 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 6-6 は、ファイルシステム表示の例です。

コード例 6-6 samu(1M) f の表示

```
File systems samu 4.2 08:11:24 Sept 8 2004
License: License never expires.

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
md 13 on /dev/dsk/c5t8d0s5
md 14 on /dev/dsk/c5t8d0s6
```

コード例 6-6 samu(1M) f の表示 (続き)

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 6-6 で、この表示のフィールドを説明します。

表 6-6 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 <ul style="list-style-type: none"> <li>• on - 装置をアクセスに利用可能</li> <li>• ro - 装置は、読み込み専用アクセスだけに利用可能</li> <li>• off - 装置をアクセスに利用できない。</li> <li>• down - 装置は、メンテナンスアクセスだけに利用可能。</li> <li>• idle - 装置は、新しい操作に利用できない。進行中の操作は、終了するまで続行される。</li> </ul>
device_name	ファイルシステム名または装置のパス

表 6-6 samu(1M) f の表示フィールドの説明 (続き)

フィールド	説明
status	デバイスの状態。状態コードの説明については、197 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。サン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 6-7 に、この表示で使用するコントロールキーを示します。

表 6-7 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 6-7 は、Sun StorEdge SAM-FS ファイルシステムの最初のヘルプ画面を示します。Sun StorEdge 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 StorEdge 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。Sun SAM-QFS のソフトウェアを使用している場合、ヘルプ画面はコード例 6-7 のように表示されます。

コード例 6-7 Sun StorEdge SAM-FS システムの場合の samu(1M) の初期ヘルプ画面

```
Help information           page 1/15      samu 4.2           08:18:13 Sept 8 2004
License: License never expires.

Displays:

  a  Archiver status           w      Pending stage queue
  c  Device configuration      C      Memory
  d  Daemon trace controls     F      Optical disk label
  f  File systems              I      Inode
  h  Help information          J      Preview shared memory
  l  License information       K      Kernel statistics
  m  Mass storage status       L      Shared memory tables
  n  Staging status           M      Shared memory
  o  Optical disk status       N      File system parameters
  p  Removable media load requests P      Active Services
  r  Removable media          R      SAM-Remote
  s  Device status            S      Sector data
  t  Tape drive status        T      SCSI sense data
  u  Staging queue            U      Device table
```

```
v Robot catalog
```

```
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 6-8 に、この表示で使用するコントロールキーを示します。

表 6-8 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。サン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 6-8 は、i ノード表示の例です。

コード例 6-8 samu(1M) I の表示

```
Inode      0x1 (1) format: file          samu 4.2          08:27:14 Sept 8 2004
License: License never expires.          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
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 6-9 に、この表示で使用するコントロールキーを示します。

表 6-9 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。サン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 6-9 は、プレビュー共用メモリー表示の例です。この出力は長いので、マニュアルでは途中から省略しています。

コード例 6-9 samu(1M) J の表示

```
Preview shared memory size: 155648 samu 4.2 08:30:05 Sept 8 2004
License: License never expires.
00000000 00040000 00014d58 00000000 00000000 .....MX.....
00000010 00000000 00000000 73616d66 73202d20 .....samfs -
00000020 70726576 69657720 6d656d6f 72792073 preview memory s
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 6-10 に、この表示で使用するコントロールキーを示します。

表 6-10 K 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。サン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 6-10 は、カーネル統計情報表示の例です。

コード例 6-10 samu(1M) K の表示

```
Kernel statistics                samu 4.2                08:33:19 Sept 8 2004
License: License never expires.

module: sam-qfs  name: general instance: 0 class: fs
version                4.2.sam-qfs, gumball 2004-05-07 12:12:04
configured file systems      8
mounted file systems        8
nhino                    16384
ninodes                  129526
inocount                  129527
inofree                   128577
```

## (1) - ライセンスの表示

1 表示は、Sun StorEdge QFS および Sun StorEdge SAM-FS のソフトウェアのライセンスと有効期限を表示します。

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 6-11 は、ライセンス表示の例です。

コード例 6-11 samu(1M) 1 の表示

```
License information                samu      4.2      08:36:27 Sept 8 2004
License: License never expires.

hostid = 80e69e6e

License never expires
Remote sam server feature enabled
Remote sam client feature enabled
Migration toolkit feature enabled
Fast file system feature enabled
Data base feature enabled
Foreign tape support enabled
Segment feature enabled
Shared filesystem support enabled
SAN API support enabled

Robot type STK ACSLS Library is present and licensed
  350 sg slots present and licensed
```

表示例は、Sun StorEdge SAM-FS ファイルシステムのライセンス情報を示しています。ライセンス情報は、次のファイルに定義されているライセンスキーから引き出されます。

```
/etc/opt/SUNWsamfs/LICENSE.4.2
```

この表示は、システムの次の情報を表示します。

- 有効期限の情報
- ホスト ID

- 使用可能な Sun StorEdge QFS および Sun StorEdge SAM-FS の製品と機能
- 装置とメディアの組み合わせ

## (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。サン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 6-12 は、共用メモリーテーブルの例です。

コード例 6-12 samu(1M) L の表示

```
Shared memory tables          samu 4.2 08:38:31 May  8 2004
License: License never expires.

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
fseq 20 samfs2 state 0      0      0      0
fseq 30 qfs1 state 0        0      0      0
fseq 40 qfs2 state 0        0      0      0
```

コード例 6-12 samu(1M) L の表示 (続き)

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	0
fseq	110	shareqfs2	state	0	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 6-13 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 6-13 samu(1M) m の表示

```
Mass storage status                samu 4.2                08:41:11 Sept 8 2004
License: License never expires.

ty      eq  status      use state ord  capacity      free      ra  part high low
ms      10  m----2----d  1% on          0  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          0  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          0  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          0  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
```

コード例 6-13 samu(1M) m の表示 (続き)

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				
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 6-11 で、この表示のフィールドを説明します。

表 6-11 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、197 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 6-12 に、この表示で使用するコントロールキーを示します。

表 6-12 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。サン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 6-14 は、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 6-14 samu(1M) M の表示

```
Shared memory      size: 73728          samu 4.2          08:43:20 May  8 2004
License: License never expires.
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  .....
```



コード例 6-14 samu(1M) M の表示 (続き)

```

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 .....

00000100 00000000 00000000 00000000 00000000 .....
00000110 00000000 00000000 00000000 00000000 .....
00000120 00000000 00000000 00000000 00000000 .....
00000130 00000000 00000000 00000000 00000000 .....
00000140 00000000 00000000 00000000 00000000 .....
00000150 00000000 00000000 00000000 00000000 .....
00000160 00000000 00000000 00000000 00000000 .....
00000170 00000000 00000000 00000000 00000000 .....

00000180 00000000 00000000 00000000 00000000 .....
00000190 00000000 00000000 00000000 00000000 .....
000001a0 00000000 00000000 00000000 00000000 .....
000001b0 2f766172 2f6f7074 2f53554e 5773616d /var/opt/SUNWsam
000001c0 66732f61 6d6c6400 00000000 00040000 fs/amld.....
000001d0 00014d58 00000000 00000000 00000000 ..MX.....
000001e0 00000000 00000097 00000000 00000000 .....
000001f0 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 6-15 は、書き込み状態表示の例です。

コード例 6-15 samu(1M) n の表示

```
Staging status          samu 4.2          08:47:16 May  8 2004
License: License never expires.

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タ表示

N 表示は、すべてのマウントポイントパラメタ、スーパーブロックバージョン、およびその他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 6-13 に、この表示で使用するコントロールキーを示します。

表 6-13 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

表 6-13 N 表示のコントロールキー (続き)

キー	機能
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。サン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 6-16 は、ファイルシステムパラメタ表示の例です。

コード例 6-16 samu(1M) N の表示

```

File system parameters          samu 4.2          08:55:19 Sept 8 2004
License: License never expires.
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
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、Sun StorEdge SAM-FS 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 6-14 に、この表示で使用するコントロールキーを示します。

表 6-14 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
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 6-17 は、光磁気ディスクの状態表示の例です。

コード例 6-17 samu(1M) o の表示

```
Optical disk status          samu    4.2   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 6-15 で、この表示のフィールドを説明します。

表 6-15 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、197 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	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 <ul style="list-style-type: none"><li>• 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る。アクセスに利用可能</li><li>• 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ない</li><li>• idle - 装置は、新しい接続に利用できない。進行中の操作は、終了するまで続行される。</li><li>• off - 装置をアクセスに利用できない。</li><li>• down - 装置は、メンテナンスアクセスだけに利用可能。</li></ul>
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ーを選択できます。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 6-16 に、この表示で使用するコントロールキーを示します。

表 6-16 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

表 6-16 p 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	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 6-18 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 6-18 samu(1M) p の表示

```
Removable media load requests all both samu 4.2 09:14:19 Sept 8 2004
License: License never expires. 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 6-17 で、この表示のフィールドを説明します。

表 6-17 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 6-18 を参照。
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 6-18 で、p 表示のフラグを示します。

表 6-18 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た
-b-----	エントリがビジー状態である
--C----	VSN のクリアが要求された
---f---	ファイルシステムが要求された
----N--	メディアが Sun StorEdge SAM-FS ファイルシステムにとって異種
-----S-	すでにマウントされている側を切り替える
-----s	書き込み要求フラグ

## (P) - アクティブサービス表示

P 表示は、Sun StorEdge QFS および Sun StorEdge SAM-FS 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 6-19 に、この表示で使用するコントロールキーを示します。

表 6-19 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。サン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 6-19 は、アクティブサービス表示の例です。

コード例 6-19 samu(1M) P の表示

```
Active Services                               samu      4.2      09:08:33 Sept 8 2004
License: License never expires.

Registered services for host 'pup':
  sharedfs.qfs2
  sharedfs.qfs1
  2 service(s) 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 6-20 は、リムーバブルメディアの状態表示の例です。

コード例 6-20 samu(1M) r の表示

```
Removable media status: all                   samu 4.2      09:11:27 Sept 8 2004
License: License never expires.

ty  eq  status      act  use  state  vsn
dt 150 --l-----r   0  63% ready  DAT001
```

## フィールドの説明

表 6-20 で、この表示のフィールドを説明します。

表 6-20 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、197 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 <ul style="list-style-type: none"><li>• 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。</li><li>• 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ない。</li><li>• idle - 装置は、新しい接続に利用できない。進行中の操作は、終了するまで続行される。</li><li>• off - 装置をアクセスに利用できない。</li><li>• down - 装置は、メンテナンスアクセスだけに利用可能。</li></ul>
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。

## (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。サン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、Sun StorEdge SAM-FS 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 6-21 に、この表示で使用するコントロールキーを示します。

表 6-21 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 6-21 は、デバイスの状態表示の例です。

コード例 6-21 samu(1M) s の表示

```
Device status          samu      4.2      09:14:05 Sept 8 2004
License: License never expires.

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 6-22 で、この表示のフィールドを説明します。

表 6-22 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、197 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 6-23 に、この表示で使用するコントロールキーを示します。

表 6-23 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。サン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、Sun StorEdge SAM-FS 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 6-24 に、この表示で使用するコントロールキーを示します。

表 6-24 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 6-22 は、テープドライブの状態表示の例です。

コード例 6-22 samu(1M) t の表示

```
Tape drive status          samu      4.2      09:21:07 Sept 8 2004
License: License never expires.

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 6-25 で、この表示のフィールドを説明します。

表 6-25 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、197 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 <ul style="list-style-type: none"><li>• 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。</li><li>• 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ない。</li><li>• idle - 装置は、新しい接続に利用できない。進行中の操作は、終了するまで続行される。</li><li>• off - 装置をアクセスに利用できない。</li><li>• down - 装置は、メンテナンスアクセスだけに利用可能。</li></ul>
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。

## (T) - SCSI センスデータ表示

T 表示は、SCSI 装置の SCSI 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 6-26 に、この表示で使用するコントロールキーを示します。

表 6-26 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。サン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 6-27 に、この表示で使用するコントロールキーを示します。

表 6-27 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 6-23 は、書き込み待ち行列表示の例です。

コード例 6-23 samu(1M) u の表示

```
Staging queue by media type: all          samu 4.2          09:24:23 Sept 8 2004
License: License never expires.          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
```



コード例 6-23 samu(1M) u の表示 (続き)

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 6-28 で、この表示のフィールドを説明します。

表 6-28 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
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 6-29 に、この表示で使用するコントロールキーを示します。

表 6-29 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。サン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 6-24 は、装置テーブル表示の例です。

コード例 6-24 samu(1M) U の表示

```
Device table: eq: 10      addr: 00000450  samu 4.2          09:28:40 Sept 8 2004
License: License never expires.

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
```

コード例 6-24 samu(1M) U の表示 (続き)

```
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、156 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 6-30 に、この表示で使用するコントロールキーを示します。

表 6-30 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する
Ctrl-d	次のライブラリカタログ
Ctrl-f	順方向にページ送りする
Ctrl-i	詳細、2行表示形式。Ctrl-i を 1 回入力すると、時間とバ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 行目に表示される。
Ctrl-k	拡張ソートキー。Ctrl-k を入力した後、次のいずれかを入力して、ソートキーを選択する。 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 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 を検索する
%	バーコードを検索する
\$	スロットを検索する

## 表示例

コード例 6-25 は、自動ライブラリカタログ表示の例です。

コード例 6-25 samu(1M) v の表示

```
Robot VSN catalog by slot      : eq 100samu      4.2      09:30:25 Sept 8 2004
License: License never expires.                count 32
slot      access time count use flags          ty vsn
  0      2004/05/08 08:35:00   64 0% -il-o-b----- sg 700071
  1      2004/05/08 09:08:00   27 12% -il-o-b----- sg 700073
  2      2004/05/08 09:12:00   26 12% -il-o-b----- sg 700077
```

コード例 6-25 samu(1M) v の表示 (続き)

3	2004/05/08 08:39:00	37	40%	-il-o-b-----	sg 700079
4	2004/05/08 09:16:00	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 6-31 で、この表示のフィールドを説明します。

表 6-31 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 6-32 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

1 つのフィールドに複数のフラグが設定され、あるフラグが別のフラグを上書きすることがあります。表 6-32 は、表 6-31 の flags フィールドのフラグです。

表 6-32 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る
-i-----	使用中スロット
--l-----	ラベルあり。N より優先される。
--N-----	ラベルなし。このボリュームは、Sun StorEdge SAM-FS 環境にとって異種。
---E-----	メディアエラー。Sun StorEdge SAM-FS の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る。
----o-----	スロットが占有されている
----C-----	ボリュームはクリーニングテープである。p よりも優先される。
----p-----	優先 VSN
-----b-----	バーコードが検出された
-----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る。
-----R---	読み込み専用
-----c--	リサイクル
-----d-	重複 VSN。u よりも優先される。
-----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 6-33 に、この表示で使用するコントロールキーを示します。

表 6-33 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 6-26 は、保留書き込み待ち行列表示の例です。

コード例 6-26 samu(1M)w の表示

```
Pending stage queue by media type: all      samu      4.2  Thu Oct 11 13:20:27
License: License never expires.           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 6-34 で、この表示のフィールドを説明します。

表 6-34 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。この後の項で、これらの表示について説明します。

### 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、198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、191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 6-35 に、各位置の有効な状態コードを示します。

表 6-35 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ある
m-----	自動ライブラリは動作可能
M-----	メンテナンスモード
-E-----	装置が走査中に回復不能エラーを受信した
-a-----	装置が監査モードにある
--l-----	メディアにラベルが付いている
--N-----	外部メディア
--L-----	メディアにラベルを付ける処理中
---I-----	装置がアイドル状態となるのを待機している
---A-----	オペレータ操作が必要である
----C-----	クリーニングが必要である
----U-----	取り出しが要求された
-----R----	装置が予約されている
-----w---	プロセスがメディアに書き込みを行っている
-----o--	装置がオープン状態にある
-----P-	装置が位置付けられている (テープのみ)

表 6-35 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
-----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る。 テープと光磁気ドライブの場合、メディアがいっぱいである
-----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ある
-----r	装置は回転立ち上げしており、レディー状態にある
-----p	装置が存在している
-----W	装置は書き込み保護されている

## 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。c、o、r、s、および t 表示の状態コードについては、197 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、191 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 6-36 に、各位置の有効な状態コードを示します。

表 6-36 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る
M-----	ファイルシステムがマウント中である
-u-----	ファイルシステムがマウント解除中である
--A-----	ファイルシステムデータがアーカイブ中である
---R-----	ファイルシステムデータが解放中である
----S-----	ファイルシステムデータが書き込み中である
-----1----	Sun StorEdge SAM-FS ファイルシステムバージョン 1
-----2----	Sun StorEdge SAM-FS ファイルシステムバージョン 2
-----c----	Sun StorEdge QFS 共有ファイルシステム
-----W---	単一書き込み

表 6-36 ファイルシステム表示の状態コード (続き)

状態ビット	ファイルシステムにおける意味
-----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 6-37 に、有効な状態コードを示します。

表 6-37 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る。
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る。
off	装置をアクセスに利用できない。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以下が考えられる。 <ul style="list-style-type: none"> <li>• 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なかった。</li> <li>• 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ない</li> <li>• 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た。</li> <li>• システムがカートリッジをドライブから除去できなかった</li> <li>• 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た。</li> <li>• 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た</li> <li>• 回転立ち上げ時、ドライブの標準テープドライブのオープン処理が失敗した。</li> </ul>
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ない。進行中の操作は、終了するまで続行される。

表 6-37 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
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ない。
unavail	装置にアクセスできず、自動 Sun StorEdge SAM-FS 操作に使用できない。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る。

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、次のステップで装置を正常に起動できます。コマンドの例は次のとおりです。

```
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。  
コマンドの例は次のとおりです。

```
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、次のステップで装置を正常に停止できます。コマンドの例は次のとおりです。

```
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。

コマンドの例は次のとおりです。

```
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

この後の項では、`samu(1M)` オペレータユーティリティ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。

- 202 ページの「デバイスコマンド」
- 203 ページの「SAM コマンド - アーカイバ制御」
- 206 ページの「SAM コマンド - ステージャ制御」
- 206 ページの「SAM コマンド - リリーサ制御」
- 208 ページの「ファイルシステムコマンド - 入出力管理」
- 211 ページの「ファイルシステムコマンド - 直接入出力管理」
- 212 ページの「ファイルシステムコマンド - Sun StorEdge QFS 共有ファイルシステム」
- 214 ページの「ファイルシステムコマンド - その他」

- 217 ページの「自動ライブラリコマンド」
- 219 ページの「その他のコマンド」

Solaris オペレーティングシステム (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する必要があります。`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。

この後の項では、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 6-38 は、デバイスコマンドとその動作です。

表 6-38 デバイスコマンドのアクション

コマンド	動作
<code>down</code>	装置 <code>eq</code> に対する処理を終了する
<code>idle</code>	装置 <code>eq</code> 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する。既存の処理は、終了するまで続行される。
<code>off</code>	装置 <code>eq</code> を論理的に停止する
<code>on</code>	装置 <code>eq</code> を論理的に起動する
<code>unavail</code>	装置 <code>eq</code> を選択し、Sun StorEdge SAM-FS ファイルシステムで使用できないようにする。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 <code>unavail</code> にすると、Sun StorEdge SAM-FS 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る。
<code>unload</code>	指定のリムーバブルメディア装置 <code>eq</code> 用にマウントされているメディアを取り出す。マガジンデバイスの場合、 <code>unload</code> 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す。

コード例 6-27 は、デバイス制御コマンドの形式です。

コード例 6-27 デバイス制御コマンドの形式

```
:down eq
:idle eq
:off eq
:on eq
:unavail eq
:unload eq
```

`eq` には、装置の装置番号を指定します。

## SAM コマンド - アーカイバ制御

表 6-39 は、アーカイバコマンドとその動作です。

表 6-39 アーカイバコマンドのアクション

コマンド	動作
aridle	すべてのアーカイブを次の適切なポイントで停止する。たとえば、 <code>sam-arcopy</code> 操作の場合、現在の <code>tar(1)</code> ファイルの末尾で停止する。このコマンドは、ファイルシステムをマウント解除する前に、すべてのファイルシステムのアーカイブ稼働状況を停止するときにも使用できる。
arrrun	アーカイバでソフト再起動を実行する。アーカイバデーモンが再起動し、進行中だったすべての作業が復元される。
arrestart	アーカイバの中断と再起動を行う。このアクションは、アーカイバの状態とは関係なく行われる。そのため、 <code>arrestart</code> の使用には注意が必要。メディアをアーカイブするためのコピー操作の中には、完了しないものがある可能性があり、その場合には再度実施する必要がある。この結果、メディア空間が浪費されることになる。
armarchreq	アーカイブ要求を削除する
arrun	アーカイバにアーカイブ処理を開始させる。このコマンドは、 <code>archiver.cmd</code> ファイルに定義されている <code>wait</code> コマンドよりも優先される。
arscan	ファイルシステムを走査する
arstop	すべてのアーカイブ処理をただちに停止する
artrace	アーカイバのトレースを実行する

コード例 6-28 は、アーカイバコマンドの形式です。

コード例 6-28 アーカイバコマンドの形式

```
:aridle [ dk | rm | fs.fsname ]
:arrrun
:arrestart
:armarchreq fsname.[* | archreq]
:arrun [ dk | rm | fs.fsname ]
:arscan fsname[.dir | ..inodes][int]
:arstop [ dk | rm | fs.fsname ]
:artrace [fs.fsname]
```

これらのコマンドの引数は、省略可能です。引数が指定されていない場合、全ファイルシステムが処理対象となります。引数が指定されている場合、指定されたアーカイブファイル (dk や rm) および指定されたファイルシステムに基づいてコマンドが実行されます。表 6-40 に、アーカイバコマンドの引数を示します。

表 6-40 アーカイバコマンドの引数

引数	説明
dk	ディスクアーカイブファイルに関するコマンドであることを指定する
rm	リムーバブルメディアファイルに関するコマンドであることを指定する
<i>fsname</i>	特定のファイルシステムに関するコマンドであることを指定する。ファイルシステム名を <i>fsname</i> に入力する。
<i>archreq</i>	特定のアーカイブ要求ファイルを次の形式で指定する <i>arset.copy.seq_num</i> このファイル名には、3つの構成要素がある。各構成要素は、ピリオドで区切る。最初の構成要素は、アーカイブセットの名前。2番目の構成要素は、コピー番号 (1～4)。3番目の構成要素は、アーカイバが割り当てるシーケンス番号。同時に複数のアーカイブ要求が存在することがある。 システムに存在するアーカイブ要求ファイルの名前を取得するには、 <code>showqueue(1M)</code> コマンドを使用する。このコマンドの使用方法をコード例 6-29 で示す。または、次のディレクトリに移動して、存在するファイルを一覧表示する。 <code>/var/opt/SUNWsamfs/archiver/<i>fsname</i>/ArchReq</code>
*	すべてのファイルを意味する
<i>dir</i>	特定のディレクトリ名を指定する。これが、走査対象ディレクトリになる。
<code>.inodes</code>	i ノードを走査することを指定する
<i>int</i>	走査を待機する秒数を示す整数

コード例 6-29 は、`showqueue(1M)` コマンドを使用して、`arrmarchreq samu(1M)` コマンドの入力として使用できる *archreq* ファイル名を取得する方法です。

コード例 6-29 `showqueue(1M)` の使用方法

```
# showqueue samfs9
Filesystem samfs9:
Scan list: empty
Archive requests
arset1.2.0 schedule 2004-01-22 16:23:07
  files:697 space:  4.934G flags: offline
(min:  1.000k)  priority: 0 0
  No volumes available
  Drive 1
  Files: 695, bytes:  1.932G (min:  1.000k)
```



コード例 6-29 showqueue(1M) の使用方法 (続き)

```
Stage volumes:
  lt.CFX600
  lt.CFX601

arset1.1.1 schedule 2004-01-22 16:23:07
  files:3 space: 6.236M flags:
(min: 826.000k) priority: 0 0
  No volumes available
  Drive 1
  Files: 3, bytes: 6.236M (min: 826.000k)
```

コード例 6-29 は、ファイル arset1.2.0 および arset1.1.1 がアーカイブ要求ファイルであることを示しています。

## :hwm\_archive eq コマンドと :nohwm\_archive eq コマンド

`hwm_archive` コマンドは、ファイルシステム中のデータ量が増えてリリーサの最高ウォーターマークを超えたときに、アーカイバを起動します。リリーサの最高ウォーターマークは、`thresh` コマンドを使用して設定します。`thresh` コマンドについては、205 ページの「`:thresh eq high low` コマンド」を参照してください。

`nohwm_archive` コマンドは、この機能を使用不可にして、デフォルトの状態にします。

`eq` には、ファイルシステムの装置番号を指定します。

## :thresh eq high low コマンド

`thresh` コマンドは、ファイルのアーカイブを制御するため、ファイルシステムにおける上限と下限のしきい値を設定します。

`eq` には、ストレージファミリセットの装置番号を指定します。

`high` には、上限しきい値を指定します。

`low` には、下限しきい値を指定します。

たとえば、次のコマンドは、ファイルシステムの装置番号が 10 であるストレージファミリセットに対し、上限しきい値 50 パーセント、下限しきい値 40 パーセントを設定します。

```
:thresh 10 50 40
```

## SAM コマンド - リリーサ制御

次のコマンドで、部分的解放機能を制御できます。部分的解放機能の詳細については、『Sun StorEdge SAM-FS ストレージ/アーカイブ管理マニュアル』の「解放処理」の章を参照してください。

### `:maxpartial eq value` コマンド

`maxpartial` コマンドは、ファイルシステムの部分的解放の最大サイズを *value* キロバイトに設定します。部分的解放のサイズを、この `maxpartial` 設定より大きくすることはできません。

*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、 $0 \leq value \leq 2097152$  となる整数を指定します。

### `:partial eq value` コマンド

`partial` コマンドは、ファイルの解放後にオンラインのままにするキロバイト数を設定します。詳細については、『Sun StorEdge SAM-FS ストレージ/アーカイブ管理マニュアル』の「解放処理」の章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。

*value* には、オンラインのままにするキロバイト数を指定します。デフォルトのサイズは 16 です。

## SAM コマンド - ステージャ制御

次のコマンドで、書き込みの稼働状況を制御できます。

### `:partial_stage eq value` コマンド

`partial_stage` コマンドは、ファイルシステムの部分的な書き込みサイズを *value* キロバイトに設定します。部分的解放属性を持つファイルは、*value* で指定されたファイルオフセットを超えてアクセスされると、ファイル全体がディスクに書き込まれます。

*eq* には、ファイルシステムの装置番号を指定します。

*value* には、0 より大きく、*maxpartial* 設定で指定した *value* より小さい整数を指定します。*maxpartial* 設定の詳細については、206 ページの「:*maxpartial eq value* コマンド」を参照してください。部分的解放機能の詳細については、『Sun StorEdge SAM-FS ストレージ/アーカイブ管理マニュアル』の「解放処理」の章を参照してください。

### :*stage\_flush\_behind eq value* コマンド

*stage\_flush\_behind* コマンドは、最大書き込み遅延フラッシュ値を設定します。Sun Solaris VM レイヤーがページを空にしておけるように、書き込みされるページは非同期でディスクに書き込まれます。

*eq* には、ファイルシステムの装置番号を指定します。

*value* には、 $0 \leq \textit{value} \leq 8192$  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、*stage\_flush\_behind* は使用不可です。

### :*stage\_n\_window eq value* コマンド

*stage\_n\_window* コマンドは、*stage(1)* コマンドの *-n* オプションと協調して動作します。この *samu(1M)* コマンドは、ファイルシステムに対する *stage(1)* コマンドの *-n* オプションを *value* に設定します。このコマンドは、アーカイブメディアから直接読み取られ、*stage -n* が指定されているファイルに有効です。*stage -n* 属性が設定されているファイルの場合、*value* は、アプリケーションのバッファーに一度に書き込まれるデータの量になります。

*eq* には、ファイルシステムの装置番号を指定します。

*value* には、 $64 \leq \textit{value} \leq 2097152$  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、Sun StorEdge QFS 共有ファイルシステム以外のすべてのファイルシステムで 256 です。共有ファイルシステムの場合は、*minallopsz* マウントオプションの値に設定されます。

### :*stage\_retries eq value* コマンド

*stage\_retries* コマンドは、エラーが発生した場合のアーカイブコピーごとの書き込み再試行の回数を設定します。

*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、 $0 \leq \textit{value} \leq 20$  となる数値を指定します。*value*=0 の場合、再試行は行われません。デフォルトは 3 です。

## :stclear *mt.vsn* コマンド

stclear コマンドは、書き込み要求をクリアします。

*mt* にはメディアタイプを指定します。有効なメディアタイプについては、mcf(4) のマニュアルページを参照してください。

*vsn* には、マウントするボリュームを指定します。

## :stidle コマンド

stidle コマンドは、ステージャをアイドル状態にします。このコマンドを使用すると、ステージャは現在のタスクを終了し、追加の書き込みを開始しません。

## :strun コマンド

strun コマンドは、書き込み処理を再開します。stidle コマンドを発行した後で、ステージャを再起動するときに、このコマンドを使用します。

# 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、 $0 \leq value \leq 8192$  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にのみ有効です。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 StorEdge QFS 共有ファイルシステム内で、複数の NFS サーバーを使用できます。Sun StorEdge QFS 共有ファイルシステムの詳細については、97 ページの「Sun StorEdge QFS 共有ファイルシステム」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、 $1 < contig < 8192$  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 StorEdge QFS および Sun StorEdge SAM-FS 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、303 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タを参照してください。

- 211 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」
- 211 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」

`: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、303 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 StorEdge 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 StorEdge 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 StorEdge 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、128 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 15 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。

## `:mhwrite 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、126 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、124 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`, `:wrlease eq interval`, コマンドと `: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、125 ページの「Sun StorEdge 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 - その他

次のコマンドで、リース、割り当てサイズ、およびその他のさまざまなファイルシステム特性を制御できます。

### `:invalid eq interval` コマンド

`invalid` コマンドは、ファイルが変更された後 *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後、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後 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 (Sun StorEdge QFS ファイルシステムのみ)

`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 16 キロ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 (Sun StorEdge QFS ファイルシステムのみ)

`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、その他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 (Sun StorEdge QFS ファイルシステムのみ)

`refresh_at_eof` コマンドと `norefresh_at_eof` コマンドを使用して、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ホストの Sun StorEdge QFS 複数読み取りファイルシステムを高速更新できます。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。

## `:setuid eq` コマンドと `:nosetuid eq` コマンド

`setuid` コマンドと `nosetuid` コマンドは、このファイルシステムで `setuid` の実行を許可するかどうかを制御します。これらのマウントオプションは、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 `setuid(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 < 128 キロバイトの場合、128 キロバイト/DAU
- DAU > 128 キロバイトの場合、1

デフォルトの場合、Sun StorEdge 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、9 ページの「ファイルシステム設計」および 41 ページの「ボリューム管理」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 StorEdge 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、128 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後、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 Sun StorEdge QFS または Sun StorEdge SAM-FS の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 StorEdge QFS ファイルシステムのデフォルトです。Sun StorEdge 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大域指示です。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。

## 自動ライブラリコマンド

次のコマンドは、自動ライブラリのメディアの稼働状況を制御します。

### `:audit [ -e ] eq [ :slot [ :side ]]` コマンド

`audit` コマンドを使用すると、指定の自動ライブラリ装置が各ボリュームをマウントし、VSN を読み込み、ライブラリカタログを再構築します。

`-e` が指定され、ボリュームがテープカートリッジの場合、テープはデータの末尾 (EOD) にスキップし、使用可能な容量が更新されます。EOD へのスキップは割り込み不可であることに注意してください。条件によっては、完了まで数時間かかることがあります。

`eq` には、自動ライブラリ装置の装置番号を指定します。

`slot` には、読み込みたいボリュームが入っているスロット番号を指定します。

`side` には、光磁気ディスクの面を指定します。1 または 2 です。テープカートリッジの場合には、この引数を使用しない。

このコマンドは、ネットワークに接続されたライブラリではサポートされません。

### `:export eq:slot` コマンドと `:export mt.vsn` コマンド

`export` コマンドを使用すると、指定の自動ライブラリがボリュームをメールスロットにエクスポートします。ボリュームは、自動ライブラリ内のスロット位置によって識別されます。

- 装置番号とスロット番号を指定してエクスポートする場合、指定の自動ライブラリがボリュームをメールスロットに移動します。`eq` には、装置番号または装置名を指定します。`slot` には、エクスポートしたいボリュームが入っているスロット番号を指定します。

- 論理識別子を指定してエクスポートする場合、指定の自動ライブラリがボリュームをメールスロットに移動します。*mt* にはメディアタイプを指定します。有効なメディアタイプについては、*mcf(4)* のマニュアルページを参照してください。*vsn* には、エクスポートするボリュームを指定します。

## :import *eq* コマンド

*import* コマンドを使用すると、指定の自動ライブラリにカートリッジを追加できます。*eq* には、自動ライブラリの装置番号を指定します。

## :load *eq:slot* [ *:side* ] コマンドと :load *mt.vsn* コマンド

*load* コマンドにより、次のように、物理識別子と論理識別子のどちらによっても読み込みを行えます。

- 装置番号とスロット番号を指定して読み込む場合、指定の自動ライブラリがボリュームをドライブに読み込みます。  
*eq* には、装置番号または装置名を指定します。  
*slot* には、読み込みたいボリュームが入っているスロット番号を指定します。  
*side* には、光磁気ディスクの面を指定します。1 または 2 です。テープカートリッジの場合には、この引数を使用しない。
- 論理識別子を指定して読み込む場合、指定の読み込み対象自動ライブラリがラベル付きボリュームをドライブにマウントします。  
*mt* にはメディアタイプを指定します。有効なメディアタイプについては、*mcf(4)* のマニュアルページを参照してください。  
*vsn* には、マウントするボリュームを指定します。

## :priority *pid newpri* コマンド

*priority* コマンドは、プロセスの読み込み優先順位を設定します。このコマンドは、リムーバブルメディアマウント要求表示から指定できます。詳細は、178 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。

*pid* には、p 表示で表示される優先順位を指定します。

*newpri* には、その要求に設定する優先順位を指定します。整数にする必要があります。

## 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タスクを実行できます。

### `: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、178 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。

`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。

### `: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`

dttrace 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 6-41 に、トレース制御コマンドの引数を示します。

表 6-41 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する。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る。次のプロセス名のどれか 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る。sam-archiverd、sam-catserverd、sam-fsd、sam-rftd、sam-recycler、sam-sharefsd、および sam-stagerd。キーワード on または off をプロセス名の後に指定できる。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る。
<i>variable value</i>	さまざまな <i>variable</i> 引数と <i>value</i> 引数を指定できる。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る。次の <i>variable</i> と <i>value</i> の組み合わせのどれか 1 つを指定する。 <ul style="list-style-type: none"><li>• <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する。フルパス名を指定できる。</li><li>• <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する。</li><li>• <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する。</li><li>• <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する。</li></ul>

## : 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 StorEdge QFS または Sun StorEdge SAM-FS 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、サン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を終了しないまま、シェルコマンドを実行できます。



## 第7章

---

# ファイルシステム割り当て

---

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。「管理」セットは、サイトで指定されるユーザーグループです。

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。

この章の内容は次のとおりです。

- 223 ページの「概要」
- 226 ページの「割り当ての有効化」
- 238 ページの「割り当ての検査」
- 241 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は、ユーザー、グループ、またはサイトで定義される管理セットごとに設定できます。システム管理者は、ファイルの数、オンラインブロックの数、およびブロックの合計数に制限値を設定できます。

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。

- 224 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」
- 225 ページの「弱い制限値と強い制限値」

- 226 ページの「割り当てとアーカイブメディア」
- 226 ページの「ディスクブロックとファイル割り当て」

表 7-1 に、この章の割り当ての説明でよく使用される用語を示します。

表 7-1 割り当ての用語

用語	定義
猶予期間	ユーザーが弱い制限値に到達してから、ファイルの作成または記憶領域の割り当てを許可される期間。
弱い制限値	ディスク割り当てについて、ユーザーが一時的に超過できるファイルシステム資源 (ブロックおよび i ノード) のしきい値。弱い制限値を超えると、タイマーが起動します。指定の時間 (猶予期間) が過ぎてもユーザーが弱い制限値を超えていると、ユーザーがファイルシステム使用量を弱い制限値未満に減らさないかぎり、それ以上のシステム資源を割り当てられなくなる。
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。
割り当て	ユーザーが使用できるシステム資源の容量。
タイマー	ユーザーが弱い制限値に到達してからの経過時間を追跡する機能。猶予期間が過ぎると、強い制限値がユーザーに適用される。

## 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。

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、quota マウントオプションが有効な場合に、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になっているため、noquota マウントオプションを指定して割り当てを無効にしないでください。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のためのレコードです。システム管理者の資源使用量は、レコード 0 に累積されます。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、236 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。異なるユーザーのために異なる割り当てを設定するために、レコード 1 およびすべての後続のレコードを編集することができます。表 7-2 に、割り当てファイル名と /root で利用可能な割り当てを示します。

表 7-2 割り当てファイル名

/root ディレクトリの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 の値を他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。

## 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。新しいユーザーが、自分の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。この場合、操作 (通常は write(2) または creat(2)) は失敗し、EDQUOT エラーが生成されます。

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が、強い制限値を超えることはできません。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限り、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。sqquota(1) コマンドは、ユーザー用の samquota(1M) コマンドです。sqquota(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。

## 割り当てとアーカイブメディア

割り当てを使用して、Sun StorEdge QFS および Sun StorEdge SAM-FS ファイルシステムのアーカイブメディアでユーザーに許可するデータ量を制限できます。

例：stage(1) コマンドを使用して、アーカイブメディアのデータをオンライン化します。システムレベルで stage(1) コマンドを呼び出すと、次のようにユーザー割り当てを超えることが可能になります。

```
# stage -r *
```

ユーザー割り当ては、ユーザーが次のように stage(1) -w コマンドを実行したときに監視されます。

```
# stage -w *
```

ユーザーの割り当てが満たされるまで、ファイルは書き込まれます。その後は、ファイルは書き込まれません。

## 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の最大ブロック数は、最も近い DAU 数に切り捨てられます。

---

## 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。

表 7-3 に、割り当ての操作時に使用するコマンドを示します。

表 7-3 割り当てのコマンド

コマンド	説明
squota(1)	あるユーザーの割り当て統計を表示する。samquota(1M) コマンドのサブセット。
samchaid(1M)	ファイル管理セット ID 属性を変更する
samquota(1M)	あるユーザー、グループ、または管理セットの割り当て統計を表示する。また、このコマンドを使用すると、管理者が割り当てレコードを編集できる。
samquotastat(1M)	ファイルシステムにアクティブな割り当てがある場合に、その割り当てについて報告する。

samfsck(1M) コマンドを実行すると、ファイルシステムを確認し、割り当てファイルに記録されている使用量が実際のファイルシステム使用量の合計と一致することを確認します。一致しないと、samfsck(1M) コマンドによって通知されます。ファイルシステム修復が実行される場合は、正しくない既存の割り当てレコードがこのコマンドによってすべて更新されます。

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近づく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 バイトのスライスを共有する場合は、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。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。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。**samquota(1M)** コマンドの **-U**、**-G**、および **-A** オプションは、ユーザー、グループ、または管理セットのいずれれに対してコマンドを使用するかを決定する。コード例 7-1 にこれを示す

コード例 7-1 **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

次の手順では、割り当てを使用するために新しいファイルシステムを構成する方法を説明します。この手順が適用されるのは、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です。

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には、230 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。

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。

ファイルシステムを作成するには、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』に概要を示す操作を行うか、52 ページの「構成の例」の例を参照して、**mcf** ファイルおよび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。

**mount(1M)** コマンドを使用して、次のようにファイルシステムをマウントします。

```
# 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、ファイルシステムをマウント解除します。

割り当てファイルを作成したファイルシステムを、umount(1M) コマンドを使用してマウント解除します。コマンドの例は次のとおりです。

```
# 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。

ファイルシステムで samfsck(1M) コマンドを実行します。たとえば、次のコマンドによってファイルシステムの検査が実行されます。-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、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有効になります。



---

**注意** - /etc/vfstab ファイルまたは samfs.cmd ファイルに割り当て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。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。

割り当てファイルが存在し、割り当てを有効にしないでファイルシステムをマウントし、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。割り当てを含むファイルシステムが quota マウントオプションを指定しないでマウントおよび実行している場合は、-F オプションを指定して samfsck(1M) 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。

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。

この章のこの後の節では、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、228 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。

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、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。

次のように引数を指定しないで mount(1M) コマンドを使用して、/etc/mnttab ファイルを確認します。

```
# mount
```

### 3. cd(1) コマンドを使用して、ルートディレクトリに移動します。

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。コマンドの例は次のとおりです。

```
# cd /oldfs1
```

### 4. 割り当てがまだファイルシステムに存在していないことを確認します。

ルートディレクトリで、`ls(1) -a` コマンドを使用して、そのディレクトリのファイルのリストを取り出します。少なくとも 1 つの割り当てタイプ (u、g、または a) がファイルシステムに対して設定されている場合、後で任意のその他のタイプを設定できます。新しいファイルを後で追加するときに、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。

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

### 5. 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応じた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 番号で最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。

#### 例 1

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最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。

- $(1024+1)*128 = 131200$

- $131200/4096 = 32.031...$

次のコマンドを使用します。

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

#### 例 2

グループ割り当てを有効にするときに、最大 2000 までのグループ ID を使用している場合は、次のように計算します。

- $(2000+1)*128 = 256128$

- $256128/4096 = 62.531\dots$

次のコマンドを使用します。

```
# dd if=/dev/zero of=/oldfs1/.quota_g bs=4096 count=63
```

### 例 3

ユーザー ID 割り当てを有効にするときに、最大 4799 までのユーザー ID が使用中の場合は、次のように計算します。

- $(4799+1)*128 = 1228800$
- $1228800/4096 = 300.0$

次のコマンドを使用します。

```
# dd if=/dev/zero of=/oldfs1/.quota_u bs=4096 count=300
```

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。

コマンドの例は次のとおりです。

```
# 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せん。ファイルシステムのマウント解除の詳細は、74 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)` 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) -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、割り当てファイルが更新されます。

コマンドの例は次のとおりです。

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

---

**注** – この操作のコマンドは、割り当てファイルにすでに割り当てられているレコードしか更新しないことに注意してください。

---

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。



---

**注意** - `/etc/vfstab` ファイルまたは `samfs.cmd` ファイルで、`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。`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。

割り当てファイルが存在し、割り当てを有効にしないでファイルシステムをマウントし、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。割り当てを含むファイルシステムが `quota` マウントオプションを指定しないでマウントおよび実行している場合は、`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。

この章のこの後の節では、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。コマンドの例は次のとおりです。

```
# 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。コマンドの例は次のとおりです。

```
# 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。無限割り当ては、ブロックとファイルの強い制限値を両方とも 0 に設定することで、ユーザー、グループ、または管理セットごとに設定できます。ファイルシステムは、無限割り当てを特別な割り当てとして処理します。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。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。

### ▼ 無限割り当てを設定する

- `samquota(1M)` コマンドを使用して、無限割り当てを設定します。  
たとえば、次のコマンドを使用して無限割り当てを設定します。

```
# 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 7-2 に、無限割り当ての設定方法を示します。

## コード例 7-2 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) コマンドを使用して、無限割り当てを設定します。

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
```

最初の参照時に、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は、15,000 ブロックに設定される
- 総ブロックの弱い制限値は、12 ギガブロックに設定される
- 総ブロックの強い制限値は、15 ギガブロックに設定される
- ファイルの弱い制限値は、1000 ファイルに設定される
- ファイルの強い制限値は、1200 ファイルに設定される

- 猶予期間は 1 週間に設定される

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であることに注意してください。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。

コード例 7-3 に、さまざまな割り当てを有効にするコマンドを示します。

コード例 7-3 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後で、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`。出力先を一時ファイルにします。

```
# samquota -G sam -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 40M:s:t -b 60M:h:t -t 0s:o -t 0s:t /sam6
```

任意の一時ファイルを使用できます。手順 2 では、1 つまたは複数のフィールドを変更するためにエディタを使用します。ユーザー割り当てエントリを作成するためのテンプレートとして、グループ割り当てエントリを使用することができます。コード例 7-4 に、テンプレートとして使用するために、`quota.group` 作成したり取り出す方法を示します。

コード例 7-4 ファイル `quota.group`

```
# samquota -G sam -e /sam6 > /tmp/quota.group  
# cat /tmp/quota.group  
  
# Type ID  
# Online Limits Total Limits  
# soft hard 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、手順 1 からファイルを編集します。

たとえば、コード例 7-5 は、`vi(1)` エディタで開き、手順 1 で作成したファイルを示しています。このファイルは、グループ ID 101 が 102 に変更されたことも示しています。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 7-5 編集後のファイル `quota.group`

```
# Type ID  
# Online Limits Total Limits  
# soft hard soft hard  
# Files  
# Blocks  
# Grace Periods  
#
```

## コード例 7-5 編集後のファイル quota.group (続き)

```
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. シェルを使用してファイルを実行します。

この操作は、エディタで加えた変更に応用されます。コマンドの例は次のとおりです。

```
# 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。

同様に、この手順を使用して、ユーザー、グループ、管理 ID、ファイルシステム、およびその他のエンティティ間で割り当て制限値をコピーする割り当てコマンドを生成できます。

## 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後で、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。表 7-4 に、割り当てを検査するために使用できるコマンドを示します。

表 7-4 割り当て検査のコマンド

コマンド	処理
<code>squota(1)</code>	ユーザーコマンド。1 ユーザーのユーザー割り当てとその他の情報が表示される。詳細は、 <code>squota(1)</code> のマニュアルページを参照してください。
<code>samquota(1M)</code>	管理者コマンド。ユーザー、グループ、管理セットの割り当てが表示され、現在のディスク使用量も表示される。また、このコマンドを使用すると、割り当てを超過しているユーザーの情報も表示される。詳細は、 <code>samquota(1M)</code> のマニュアルページを参照してください。

## ▼ 超過した割り当てを検査する

次の手順では、使用超過した割り当てを検査する方法を示します。

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、有効な割り当てを表示します。

次のどれかの方法で `samquota(1M)` コマンドを使用して、割り当てが有効になっているマウント済みファイルシステム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 7-6 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 7-6 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 7-7 によって、すべてのマウントされた Sun StorEdge QFS および Sun StorEdge SAM-FS 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 7-7 ユーザー memil の割り当て超過の検査

```
# samquota -U memil

                Online Limits          Total Limits
                Soft    Hard           In Use    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。

表 7-5 に、samquota(1M) 出力のフィールドを示します。

表 7-5 samquota(1M) の出力のフィールド

フィールド名	内容
In Use	現在のブロック使用量
Soft	ブロックの弱い制限値
Hard	ブロックの強い制限値
Grace Period	ユーザーが弱い制限値を超過できる期間

- 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。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。たとえば、次のコマンドでは、すべてのマウントされた Sun StorEdge QFS および Sun StorEdge SAM-FS 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 /qfs3
```

---

## 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。項目は次のとおりです。

- 241 ページの「猶予期間を変更する」
- 244 ページの「猶予期間の期限を変更する」
- 246 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」
- 249 ページの「ファイルシステム割り当てを削除する」
- 250 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、割り当て統計を取り出します。

`samquota(1M)` コマンドは、ユーザー、グループ、または管理セットに対して使用できます。コード例 7-8 に、割り当て統計を取り出す方法を示します。

コード例 7-8 `samquota(1M)` を使用して割り当て統計を取り出す

```
# samquota -U userID [ file ]
# samquota -G groupID [ file ]
# samquota -A adminsetID [ file ]
```

表 7-6 に、これらのコマンドの引数を示します。

表 7-6 `samquota(1M)` コマンドの引数

引数	説明
<i>userID</i>	割り当てを変更するユーザーのユーザー ID (数値) またはユーザー名を指定
<i>groupID</i>	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) またはグループ名を指定
<i>adminsetID</i>	割り当てを変更するサイト固有管理者セットの管理セット ID (数値) を指定
ファイル	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。 <i>file</i> 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、 <i>file</i> 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。

2. `samquota(1M)` コマンドの出力を確認します。

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 7-9 は、`samquota(1M)` コマンドオプションを使用して、弱い制限値の猶予期間を変更しています。

コード例 7-9 `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
```

表 7-7 に、これらのコマンドの引数を示します。

表 7-7 samquota(1M) コマンドの引数

引数	説明
<i>userID</i>	割り当てを変更するユーザーのユーザー ID (数値) またはユーザー名を指定
<i>groupID</i>	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) またはグループ名を指定
<i>adminsetID</i>	割り当てを変更するサイト固有管理者セットの管理セット ID (数値) を指定
<i>interval</i>	猶予期間に使用する間隔を指定。 <i>interval</i> 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。デフォルトでは、単位乗数は s で、間隔が秒単位で指定されます。また、w (週)、d (日)、h (時間)、または m (分) も指定できます。
ファイル	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。 <i>file</i> 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、 <i>file</i> 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: ユーザー `memil` の猶予期間を変更すると仮定します。コード例 7-10 に、割り当てとその出力を確認する `samquota(1M)` コマンドを示します。

コード例 7-10 猶予期間を変更する

```
# samquota -U memil /sam6

          Type      ID      In Use      Online Limits      Total Limits
          Type      ID      In Use      Soft      Hard      In Use      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
```

次のコマンドを入力して、弱い制限値の猶予期間を短縮します。

```
# samquota -U memil -t 1d /sam6
```

コード例 7-11 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 7-11 新しい割り当てを確認する

```
# samquota -U memil /sam6

                Online Limits                Total Limits
                Type  ID   In Use   Soft   Hard   In Use   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。

- 猶予期間をクリアする。ユーザーが次にファイルまたはブロックを割り当てると (まだ猶予期間を超えていても)、猶予期間タイマーがリセットされ、猶予期間のカウントが最初から始まる
- 猶予期間をリセットする。有効期限をリセットすると、タイマーが現在の猶予期間にリセットされ、すぐにカウントを開始する
- 猶予期間の値を設定する。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る。値の制限はない。猶予期間を超える値も指定できる
- 猶予期間を終了する。タイマーがすぐに終了するように設定する

例：

コード例 7-12 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 7-12 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)
```



コード例 7-13 では、タイマーがクリアされます。このため、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、カウントが始まります。

コード例 7-13 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
```

			Online Limits		In Use		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			
---> Warning:  online soft limits to be enforced in 6d23h59m56s								

コード例 7-14 では、猶予期間をリセットします。

コード例 7-14 猶予期間をリセットする

```
# samquota -G sam -x reset /sam6
Setting Grace Timer:  continue? y
# samquota -G sam /sam6
```

			Online Limits		In Use		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			
---> Warning:  online soft limits to be enforced in 6d23h59m52s								

コード例 7-15 では、猶予期間を終了します。

コード例 7-15 猶予期間を終了する

```
# samquota -G sam -x expire /sam6
Setting Grace Timer:  continue? y
# samquota -G sam /sam6
```

			Online Limits		In Use		Total Limits	
Type	ID	In Use	Soft	Hard	In Use	Soft	Hard	
/sam6								
Files	group	101	32	2000	2000	32	2000	2000

#### コード例 7-15 猶予期間を終了する (続き)

```
Blocks group 101 41888 40000 60000000 43208 60000000
60000000
Grace period 1w 1w
---> Online soft limits under enforcement (since 6s ago)
```

コード例 7-16 では、非常に長い有効期限を設定します。

#### コード例 7-16 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

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
```

## ▼ 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。samquota(1M) コマンドは、矛盾する値を検出し、その出力でレポートします。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、ソフトウェアは追加の割り当てを禁止します。

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 7-17 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 7-17      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。

`samquota(1M)` コマンドを使用して、割り当てを無効な値にリセットします。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。

`samquota(1M)` コマンドを使用して、割り当てが正しく変更されていることを確認します。次の例では、グループ `sam` のグループ割り当て情報を取得します。

```
# samquota -G sam /qfs1
```

コード例 7-18 に示すように、`samquota(1M)` コマンドを再び入力して、変更した割り当てを確認します。

コード例 7-18 変更した割り当てを確認する

```
# samquota -G sam /sam6
```

			Online Limits			Total Limits		
Type	ID	In Use	Soft	Hard	In Use	Soft	Hard	
/sam6								
Files	group	101	32!	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)` コマンドと `samquota(1M)` コマンドを使用して、グループの割り当てを復元します。

コード例 7-19 に、変更した割り当てを復元および確認するコマンドを示します。

コード例 7-19 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
```

			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		

---> 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指定を削除する必要があります。この手順では、ファイルシステム割り当てを無効にする方法を示します。

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. `/etc/vfstab` 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。(省略可能)

`/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がある場合にのみこの操作を行います。

`vi(1)` または `cat(1)` などのビューアを使用して、`/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があるかどうかを確認します。このマウントオプションがある場合は、ファイルを編集して `quota` マウントオプションを削除します。

---

注 – Sun StorEdge QFS および Sun StorEdge SAM-FS の 4.1 リリース以降では、`/etc/vfstab` ファイルおよび `samfs.cmd` ファイルを使用して、割り当てを有効または無効にする必要はありません。

---

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。  
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。  
コマンドの例は次のとおりです。

```
# 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、74 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。  
コマンドの例は次のとおりです。

```
# mount -o noquota /myfs
```

## 5. 割り当てファイルを削除する。

後日、割り当て機能を元に戻す場合は、割り当てファイルを破棄しないでください。割り当てファイルを保存して、後で割り当てを元に戻すには、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、**quota** マウントオプションを使用して再度ファイルシステムをマウントします。**quota** マウントオプションは、**/etc/vfstab** ファイルまたは **samfs.cmd** ファイルにマウントオプションとして指定できます。または、**mount(1M)** コマンドの **-o quota** オプションとして指定できます。

割り当て機能を元に戻さない場合、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。コマンドの例は次のとおりです。

```
# 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. **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。

コマンドの例は次のとおりです。

```
# 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、74 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、割り当てファイルが更新されます。ただし、割り当てファイルにすでに割り当てられているレコードしか更新されないことに注意してください。コマンドの例は次のとおりです。

```
# 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。  
コマンドの例は次のとおりです。

```
# mount /myfs
```





# Sun Cluster 環境での Sun StorEdge QFS

---

この章では、Sun Cluster 環境での Sun StorEdge QFS の動作について説明します。また、Sun Cluster 環境での Sun StorEdge QFS 共有ファイルシステム、および Sun Cluster 環境での非共有 Sun StorEdge QFS ファイルシステムの構成例も示します。

この章には、以下の節があります。

- 254 ページの「始める前に」
- 255 ページの「制限事項」
- 256 ページの「Sun Cluster と Sun StorEdge QFS ソフトウェアの相互作用」
- 257 ページの「構成例について」
- 258 ページの「Sun Cluster での Sun StorEdge QFS 共有ファイルシステムの構成」
- 268 ページの「Sun Cluster での非共有ファイルシステムの構成」
- 296 ページの「Sun StorEdge QFS の構成の変更」

---

## 始める前に

Sun StorEdge QFS ソフトウェアのバージョン 4.2 では、Sun StorEdge 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 StorEdge 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 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なリソースとしてファイルシステムを構成したことがある。
- Sun StorEdge QFS と Sun StorEdge QFS 共有ファイルシステムの両方をインストールおよび構成したことがある。

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 StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』の以下の章
  - 第 1 章。Sun Cluster 環境での Sun StorEdge QFS のハードウェアとソフトウェアの必須情報が書かれています。
  - 第 3 章。高可用性を得るための、Sun Cluster システムへの Sun StorEdge QFS ファイルシステムのインストール方法が説明されています。

- 以下の Sun Cluster 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、このガイドの以下の節を参照してください。

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 3.1 Software Installation Guide』。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 3.1 Data Services Installation and Configuration Guide』。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 用 Sun Cluster データサービスで Sun StorEdge 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 – 特記のない限り、このマニュアルで「Oracle Real Application Cluster」とは「Oracle Parallel Server」のことも指しています。

---

## 制限事項

Sun Cluster 環境の Sun StorEdge QFS ソフトウェアには、以下の制限があります。

- Sun StorEdge QFS でストレージ・アーカイブ管理ソフトウェア (SAM) 構成を使用する Sun StorEdge SAM-FS ソフトウェアはサポートされていません。
- 共有ファイルシステムには、以下の制限があります。
  - SUNW.qfs リソースタイプとしてだけ構成できます。HAStoragePlus リソ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、スケーラブルなファイルシステムとして構成できます。
  - 共有ファイルシステムのデータは、Sun Cluster 以外のノードからアクセスできません。Sun Cluster 以外のノードが共有 Sun StorEdge QFS リーダーとしてだけ構成されていても、アクセスできません。ファイルシステムのデータにアクセスするには、すべてのノードが同じ Sun Cluster のメンバーになる必要があります。
- 非共有ファイルシステムには、以下の制限があります。
  - HAStoragePlus リソースタイプとしてだけ構成できます。SUNW.qfs リソ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 StorEdge 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 StorEdge QFS ファイルシステムは、Sun Cluster 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。

---

# Sun Cluster と Sun StorEdge 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 Disk ID (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 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 StorEdge 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だけアクセス可能です。

DID 装置に Sun StorEdge 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 StorEdge 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。

## 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 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 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 内のすべてのノードからアクセス可能です。非共有 Sun StorEdge 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 `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他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。

## 構成例について

この章では、Sun Cluster での Sun StorEdge QFS 共有ファイルシステム、および Sun Cluster での非共有 Sun StorEdge 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、以下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 8-1 にも基づいています。このコード例で、`scdidadm (1M)` コマンドはディスク識別子 (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 8-1 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 8-1 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 StorEdge 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 での Sun StorEdge QFS 共有ファイルシステムの構成

Sun StorEdge QFS 共有ファイルシステムを Sun Cluster 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 StorEdge QFS 共有ファイルシステムを、Sun Cluster ない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 StorEdge QFS 共有ファイルシステムはスケラブルなアプリケーションと関連付けられます。Sun StorEdge 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 StorEdge 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他のノードのアクセスが影響を受けなくなります。

---

**注** – Sun Cluster で制御された Sun StorEdge 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 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。

後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

この節では、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 StorEdge 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 StorEdge 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 8-1 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 に Sun StorEdge 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 StorEdge 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 StorEdge QFS のファイルデータに使用されます。

この構成には、以下の項で説明するように 5 つのステップがあります。

1. Sun StorEdge 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 StorEdge QFS 共有ファイルシステムの作成を準備する

この手順のステップ 1 ~ 3 は、Sun Cluster システムの 1 つのノードから実行する必要があります。この例で、このステップはノード scnode-A から実行しています。

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 8-2      /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 Volume Table of Contents (VTOC) に含まれていなかったものが、50 G 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。



- format(1M) ユーティリティを使用して /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 8-3      /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。

- 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
```

- 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、以下を実行します。

- a. mcf ファイルに 2 つの新しいエントリ (qfs1 と qfs2) を追加して、6 つのパーティションが 2 つの Sun StorEdge 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 8-4 mcf ファイルへの構成エントリの追加

```
# cat >> /etc/opt/SUNWsamfs/mcf <<EOF
#
# Sun StorEdge 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、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』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 8-5 samfs.cmd ファイルの例

```
fs = qfs2
  stripe = 1
  sync_meta = 1
  mh_write
  qwrite
  forcedirectio
  nstreams = 1024
  rdlease = 600
```

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後で行ってください。

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

## ▼ Sun StorEdge 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。

コード例 8-6 Sun Cluster のプライベートインターコネクト名の取得

```
# /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すべてのノードで、以下を実行します。
  - a. `samd(1M) config` コマンドを使用して、新しい Sun StorEdge QFS 構成があることを Sun StorEdge QFS デーモンに知らせます。

```
# samd config
```

- b. ステップ 1 で調べた Sun Cluster 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 StorEdge QFS 共有ホストファイルを作成します。

3. 一意の Sun StorEdge 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 のインターコネクト名を指定します。

Sun Cluster のフェイルオーバーおよびフェンシング動作用として、Sun StorEdge 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 8-7 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 StorEdge 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すべてのノードで、以下を実行します。
  - 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、マウントポイントを `other` が読み書きできるように (755) 変更します。

コード例 8-8 qfs1 ファイルシステムのグローバルマウントポイントの作成

```
# mkdir /global/qfs1
# chmod 755 /global/qfs1
# chown root:other /global/qfs1
```

- b. `/etc/vfstab` ファイルに Sun StorEdge QFS 共有ファイルシステムのエントリを追加します。

コード例 8-9 `/etc/vfstab` ファイルへの共有ファイルシステムのエントリの追加

```
# 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
```

## ▼ 構成を検証する

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 8-10      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                    -
```

この例で、`qfs1` のメタデータサーバーは `scnode-A` です。

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。  
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は、非常に重要です。

コード例 8-11      ファイルシステム `qfs1` の Sun Cluster ノードへのマウント

```
# mount qfs1
# ls /global/qfs1
lost+found/
```

3. `samsharefs(1M) -s` コマンドを実行して Sun StorEdge 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。

コード例 8-12      ファイルシステム `qfs1` の切り替えによる任意のフェイルオーバーの検証

```
# 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。

コード例 8-13 `SUNW.qfs` リソースの構成

```
# 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。

コード例 8-14 構成済みノードでのリソースグループのテスト

```
# scswitch -z -g qfs-rg -h scnode-B
# scswitch -z -g qfs-rg -h scnode-A
```

## ▼ Oracl Real Application Cluster 用 Sun Cluster データサービスを構成する

この節では、Sun StorEdge 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 StorEdge 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。

コード例 8-15      ファイルシステム qfs1 および qfs2 の所有権とアクセス権の設定

```
# 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。

コード例 8-16      ファイルシステム qfs1 および qfs2 へのサブディレクトリの作成

```
$ 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、ステップ 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 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。

---

# Sun Cluster での非共有ファイルシステムの構成

非共有 Sun StorEdge 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 上の非共有 Sun StorEdge QFS ファイルシステムは、HA-NFS、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と関連付けられます。非共有 Sun StorEdge 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 StorEdge 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 StorEdge 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 StorEdge 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 StorEdge 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または VERITAS Clustered Volume Manager) が必要で、1 つは Sun StorEdge QFS メタデータ (i ノード) 用、もう 1 つは Sun StorEdge 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 StorEdge 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、9 ページの「設計の基本」を参照してください。

この節では、非共有 Sun StorEdge QFS ファイルシステムを使用する Sun Cluster 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以下の 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。

- 例 1 は raw グローバル装置
- 例 2 は 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 3 は VERITAS Clustered Volume Manager (VxVM) で制御されるボリューム



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 StorEdge QFS のメタデータに使用され、残りの容量が Sun StorEdge 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 StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』を参照してください。

## 例 1

この例は、raw グローバル装置上の HA-NFS での非共有 Sun StorEdge 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 8-1 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 (この例は、装置 d4 ~ d7 を使用しています)。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 の検証テストの構成
4. HA-NFS の構成と高可用性ファイルシステムの構成

## ▼ 非共有 Sun StorEdge 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。

コード例 8-17 /dev/global/dsk/d4 のパーティションをレイアウトするコマンド

```
# format /dev/global/rdsk/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、20 G 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/rdsk/d4s2 | fmthard \
-s - /dev/global/rdsk/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、以下を実行します。

- 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 StorEdge QFS に構成します。

コード例 8-18 mcf ファイルへの新しいファイルシステムの追加

```
# cat >> /etc/opt/SUNWsamfs/mcf <<EOF
#
# Sun StorEdge 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、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』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ます。

Sun StorEdge 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、このステップを完了しておくことが重要です。

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

## ▼ ステップ 2 : Sun StorEdge QFS ファイルシステムの作成と Sun Cluster ノードの構成

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で、`samd(1M) config` コマンドを使用して、新しい Sun StorEdge QFS 構成があることを Sun StorEdge 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すべてのノードで、以下を実行します。
  - 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、マウントポイントを `other` が読み書きできるように (755) 変更します。

コード例 8-19 `qfsnfs1` ファイルシステムのグローバルマウントポイントの作成

```
# mkdir /global/qfsnfs1
# chmod 755 /global/qfsnfs1
# chown root:other /global/qfsnfs1
```

- b. `/etc/vfstab` ファイルに Sun StorEdge 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。

コード例 8-20 `/etc/vfstab` ファイルへのファイルシステムのエントリの追加

```
# 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。

コード例 8-21 構成の検証

```
# 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。

コード例 8-22      必要な Sun Cluster リソースタイプの検索

```
# 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。

コード例 8-23      必要な Sun Cluster リソースタイプの追加

```
# 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 検証テストを構成する

この節では、Sun Cluster ノード用のネットワークネームサービスと IPMP 検証テストの構成方法の例を示します。詳細は、『Sun Cluster 3.1 Software Installation Guide』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 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。

このステップは、NIS サーバーを構成する前に実行します。

コード例 8-24 ノード名として Sun Cluster およびファイルを参照するための /etc/nsswitch ファイルの編集

```
# 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検証します。

コード例 8-25 /etc/nsswitch.conf ファイルの変更の検証

```
# 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 検証テストを設定します。  
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。

コード例 8-26      アダプタ qfe2 と qfe3 の IPMP テストアドレスの静的構成

```
#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 アダプタの動的構成

コード例 8-27      IPMP アダプタ qfe2 と qfe3 の動的構成

```
# 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。

コード例 8-28 IPMP アダプタ qfe2 と qfe3 の構成の確認

```
# 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 の構成と高可用性 Sun StorEdge QFS ファイルシステムの構成

この節では、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 StorEdge 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 StorEdge 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。

コード例 8-29 2 つのファイルシステム用の NFS 共有ポイントの作成

```
# 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。

コード例 8-30 /etc/hosts テーブルへの NFS 論理ホストの追加

```
# 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。

コード例 8-31 HAStoragePlus リソースタイプの構成

```
# 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 リソースに依存性を設定します。

コード例 8-32 HAStoragePlus リソースに依存する NFS リソースタイプの構成

```
# 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、高可用性になりました。

9. 高可用性 NFS ファイルシステムを Sun StorEdge 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確認します。

コード例 8-33 リソースグループのテスト

```
# 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

この例は、Solstice DiskSuite/Solaris ボリュームマネージャ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 StorEdge 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 8-1 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では、Solaris ボリュームマネージャ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 StorEdge QFS ファイルシステムの作成に使用されます。

この例には以下のステップがあります。

1. Solstice DiskSuite/Solaris ボリュームマネージャ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 の検証テストの構成
5. HA-NFS の構成と高可用性ファイルシステムの構成

### ▼ Solstice DiskSuite/Solaris ボリュームマネージャソフトウェアを準備する

1. Solaris ボリュームマネージャのメタデータベース (metadb) が、Sun StorEdge 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。

コード例 8-34      Solaris ボリュームマネージャのメタデータベースが構成されているかどうかの確認

#	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 3.1 Software Installation Guide』を参照してください。

2. この Sun StorEdge 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 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がボリュームを作成する raw 装置のプールに追加します。

コード例 8-35 raw 装置プールへの DID 装置 d4 ~ d7 の追加

```
# metaset -s nfsdg -a /dev/did/dsk/d4 /dev/did/dsk/d5 \  
/dev/did/dsk/d6 /dev/did/dsk/d7
```

## ▼ Sun StorEdge 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。

コード例 8-36 /dev/global/dsk/d4 のパーティションをレイアウトするコマンド

```
# 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。

コード例 8-36 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、20 G 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 StorEdge QFS メタデータボリュームおよび 2 つの RAID-5 (パリティストライブ化) Sun StorEdge QFS ファイルデータボリュームを構成します。

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。

コード例 8-37      パーティションから RAID-1 メタデータボリュームおよび RAID-5 データボリュームの構成と、4 つのドライブのパーティション 0 から RAID-1 セットの作成

```
# 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
```

4. 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。

コード例 8-38      4 つのドライブのパーティション 1 から 2 つの RAID-5 セットの構成

```
# 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
```

5. ファイルシステムの潜在的ホストとなる各ノードで、Sun StorEdge 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。

コード例 8-39      メタデータサーバーの `mcf` ファイルへの Sun StorEdge QFS ファイルシステムの追加

```
# cat >> /etc/opt/SUNWsamfs/mcf <<EOF

# Sun StorEdge 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、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』を参照してください。

6. 各ノードの `mcf` の構成が正しいことを検証します。

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

## ▼ Sun StorEdge 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 StorEdge QFS 構成ファイルがあることを Sun StorEdge 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。

コード例 8-40      Solaris ボリュームマネージャのディスクグループの調停検出の有効化

```
# 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 Sun StorEdge QFS 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、以下を実行します。
  - 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、マウントポイントを `other` が読み書きできるように (755) 変更します。

コード例 8-41 qfsnfs1 ファイルシステムのグローバルマウントポイントの作成

```
# mkdir /global/qfsnfs1
# chmod 755 /global/qfsnfs1
# chown root:other /global/qfsnfs1
```

- b. `/etc/vfstab` ファイルに Sun StorEdge 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。

コード例 8-42 ファイルシステムのエントリを追加するための `/etc/vfstab` ファイルの編集

```
# 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することで、ノードが正しく構成されたことを確認します。

このステップは、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。

コード例 8-43 構成の検証

```
# 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"
```

必要な Sun Cluster リソースタイプが見つからない場合は、以下の 1 つまたは複数のコマンドで追加します。

コード例 8-44 リソース構成へのリソースタイプの追加

```
# 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 検証テストを構成する

この節では、Sun StorEdge QFS ソフトウェアで使用するための、ネットワークネームサービスと IPMP 検証テストの構成方法の例を示します。詳細は、『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 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。

このステップは、NIS サーバーを構成する前に実行します。

コード例 8-45 ノード名として Sun Cluster およびファイルを参照するための /etc/nsswitch ファイルの編集

```
# 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検証します。

コード例 8-46 /etc/nsswitch.conf ファイルの変更の検証

```
# 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 検証テストを設定します。  
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。

コード例 8-47 各アダプタの IPMP テストアドレスの静的構成

```
# 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
#
# IP Addresses for LogicalHostnames
#
192.168.2.10     lh-qfs1

EOF
```

- b. IPMP アダプタを動的に構成します。

コード例 8-48 IPMP アダプタの動的構成

```
# 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. 構成を検証します。

コード例 8-49 IPMP アダプタの動的構成

```
# 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、高可用性 Sun StorEdge QFS ファイルシステムを構成する

この節では、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 StorEdge 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 StorEdge 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。

コード例 8-50 ファイルシステム用の NFS 共有ポイントの作成

```
# 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。

コード例 8-51 HAStoragePlus リソースタイプの構成

```
# 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
```

- リソースグループをオンラインにします。

```
# scswitch -Z -g nfs-rg
```

- NFS リソースタイプを構成し、HAStoragePlus リソースに依存性を設定します。

コード例 8-52 NFS リソースタイプの構成

```
# scrgadm -a -g nfs-rg -j nfs1-res -t SUNW.nfs -y \  
Resource_dependencies=qfsnfs1-res
```

- 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、高可用性になりました。

- 高可用性 NFS ファイルシステムを Sun StorEdge 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確認します。

コード例 8-53 リソースグループのテスト

```
# 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

この例は、VERITAS Clustered Volume マネージャー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 StorEdge 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 8-1 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 StorEdge 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
- VRTSvmdoc
- VRTSvmman
- VRTSvmpro
- VRTSvxvm
- VRTSob および VRTSobgui (オプションの GUI パッケージ)

この例には以下のステップがあります。

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 の検証テストの構成
6. HA-NFS の構成と高可用性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 StorEdge 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。

コード例 8-54 物理装置の HBA コントローラ番号の確認

```
# 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 ctrlr=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。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。

```
# /usr/sbin/vxdg init qfs-dg qfs-dg00=disk0 \  
qfsdg01=disk1 qfsdg02=disk2 qfsdg03=disk3
```

コード例 8-55 ディスクグループがこのシステム上で有効なことの検証

```
# vxdg import nfsdg  
# vxdg free
```

6. Sun StorEdge QFS メタデータ用の 2 つのミラー化ボリュームと、Sun StorEdge 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。

コード例 8-56      メタデータボリュームとデータボリュームの構成

```
# 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&
```

7. 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 StorEdge 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 StorEdge QFS ファイルシステムのエントリを追加します。

コード例 8-57      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 StorEdge 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、『Sun StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』を参照してください。

2. `mcf` の構成が正しいことを検証します。

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

## ▼ Sun StorEdge 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 StorEdge QFS 構成ファイルがあることを Sun StorEdge 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 Sun StorEdge QFS 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、以下を実行します。

- 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、マウントポイントを `other` が読み書きできるように (755) 変更します。

コード例 8-58 `qfsnfs1` ファイルシステムのグローバルマウントポイントの作成

```
# mkdir /global/qfsnfs1
# chmod 755 /global/qfsnfs1
# chown root:other /global/qfsnfs1
```

- b. /etc/vfstab ファイルに Sun StorEdge 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。

コード例 8-59 /etc/vfstab ファイルへのファイルシステムのエントリの追加

```
# 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
```

## ▼ 構成を検証する

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証します。

それには、288 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。この検証は、ノード 1 つずつで実行してください。

コード例 8-60 構成の検証

```
# 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。必要な Sun Cluster リソースタイプが見つからない場合は、以下の 1 つまたは複数のコマンドで追加します。

```
# scrgadm -p | egrep "SUNW.HAStoragePlus|SUNW.LogicalHostname|SUNW.nfs"
```

コード例 8-61 リソース構成への Sun Cluster リソースの追加

```
# 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 検証テストを構成する

この節では、ネットワークネームサービスと IPMP 検証テストの構成方法の例を示します。詳細は、『Sun Cluster 3.1 Software Installation Guide』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 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。

このステップは、NIS サーバーを構成する前に実行します。

コード例 8-62 ノード名として Sun Cluster およびファイルを参照するための /etc/nsswitch ファイルの編集

```
# 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
netmaks: 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検証します。

コード例 8-63 /etc/nsswitch.conf ファイルの変更の検証

```
# 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 検証テストを設定します。  
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。

コード例 8-64 各アダプタの IPMP テストアドレスの静的構成

```
# 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
#
# IP Addresses for LogicalHostnames
#
192.168.2.10     lh-qfs1
EOF
```

- b. IPMP アダプタを動的に構成します。

コード例 8-65 IPMP アダプタの動的構成

```
# 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. 構成の検証

コード例 8-66 IPMP アダプタの動的構成

```
# 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、高可用性 Sun StorEdge QFS ファイルシステムを構成する

この節では、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. ファイルシステムの潜在的ホストとなる各ノードで、Sun StorEdge 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 StorEdge 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。

コード例 8-67 ファイルシステム用の NFS 共有ポイントの作成

```
# mkdir -p /global/qfsnfs1/SUNW.nfs
# echo "share -F nfs -o rw /global/qfsnfs1" > \
/global/qfsnfs1/SUNW.nfs/dfstab.nfs1-res
```

2. Sun Cluster システムの 1 つのノードから、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。

コード例 8-68 HAStoragePlus リソースタイプの構成

```
# 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 リソースに依存性を設定します。

```
# scrgadm -a -g nfs-rg -j nfs1-res -t SUNW.nfs -y \
  Resource_dependencies=qfsnfs1-res
```

7. 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、高可用性になりました。

8. 高可用性 NFS ファイルシステムを Sun StorEdge 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、オンラインとオフラインになることを検証します。

コード例 8-69 リソースグループのテスト

```
# 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
```

---

## Sun StorEdge QFS の構成の変更

この節では、Sun StorEdge 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変更、無効化、および削除する方法を示します。次の項があります。

- 297 ページの「共有ファイルシステムの構成を変更する」
- 298 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」
- 299 ページの「Solaris ボリュームマネージャ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」
- 301 ページの「VxVM で制御されるボリュームを使用する Sun StorEdge 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

この手順は、259 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。

コード例 8-70      データベースインスタンスとリスナーの停止

```
$ 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。

コード例 8-71      リソースグループの管理対象外状態への移行

```
# 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、259 ページの「構成例」の手順に従います。

メタデータサーバーのリソースグループ構成、または Sun StorEdge QFS ソフトウェアに変更を加える (たとえば、新しいパッケージにアップグレードする) 場合は、手順 3 に進みます。

3. スーパーユーザーとして、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。

コード例 8-72 リソースグループの無効化

```
# 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 StorEdge 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

この手順を使用すると、raw グローバル装置を使用している非共有 Sun StorEdge 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、269 ページの「例 1」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 StorEdge QFS、および LogicalHost の各リソースタイプを無効にします。

コード例 8-73 リソースタイプの無効化

```
# 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。

コード例 8-74 リソースの削除

```
# 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。

コード例 8-75 不要になったリソースタイプの無効化

```
# 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

この手順を使用すると、Solstice DiskSuite/Solaris ボリュームマネージャで制御されるボリュームを使用している非共有 Sun StorEdge 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、278 ページの「例 2」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 StorEdge QFS、および LogicalHost の各リソースタイプを無効にします。

コード例 8-76 リソースタイプの無効化

```
# 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。

コード例 8-77 以前に構成したリソースの削除

```
# 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。

コード例 8-78 リソースタイプの無効化

```
# 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。

コード例 8-79 RAID-5 および RAID-1 セットの削除

```
# 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。

コード例 8-80 ドライブエラーの調停検出の削除

```
# 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。

コード例 8-81      メタデータベースの削除

```
# 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 StorEdge QFS ファイルシステム上の HA-NFS を無効にする

この手順を使用すると、VxVM で制御されるボリュームを使用している非共有 Sun StorEdge 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、287 ページの「例 3」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 StorEdge QFS、および LogicalHost の各リソースタイプを無効にします。

コード例 8-82      リソースタイプの無効化

```
# 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。

コード例 8-83      リソースの削除

```
# 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。

コード例 8-84 不要になったリソースタイプの無効化

```
# 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
```

# 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。

- 303 ページの「デーモン、プロセス、およびトレース」
- 307 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」
- 310 ページの「大容量ファイルの格納」
- 310 ページの「複数読み取りファイルシステム」
- 312 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」
- 322 ページの「入出力パフォーマンス」
- 324 ページの「大容量ファイル転送パフォーマンスの向上」
- 328 ページの「Qwrite」
- 329 ページの「書き込みスロットルの設定」
- 329 ページの「遅延フラッシュ率の設定」

---

## デーモン、プロセス、およびトレース

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 StorEdge QFS および Sun StorEdge SAM-FS 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 StorEdge QFS と Sun StorEdge SAM-FS のデーモンの名前はすべて `sam-daemon_named` という形式になっています。つまり、`sam-` にデーモン名が続き、最後に小文字の `d` が付くという形式です。このような規約があるため、デーモンは簡単に見分けることができます。プロセスの名前の付け方も似ていますが、最後に小文字

の d が付きません。表 9-1 に、システムで実行している可能性のあるデーモンとプロセスの一部を示します (システムの稼働状況によっては、この他にも sam-genericd や sam-catserverd などを実行しています)。

表 9-1 デーモンとプロセス

プロセス	説明
sam-archiverd	Sun StorEdge SAM-FS ファイルを自動的にアーカイブする。このプロセスは、Sun StorEdge SAM-FS ファイルシステムがマウントされている間ずっと実行される。
sam-fsd	マスターデーモン
sam-rftd	複数の Sun StorEdge SAM-FS ホストシステム間でデータを転送する
sam-robotd	自動ライブラリメディアチェンジャー制御デーモンを開始して監視する
sam-scannerd	手動でマウントされたすべてのリムーバブルメディア装置を監視する。アーカイブメディアのカートリッジが挿入されていないか、スキャナが定期的に各装置を検査する。
sam-sharefsd	Sun StorEdge QFS 共有ファイルシステムデーモンを呼び出す
sam-releaser	最低ウォーターマークに達するまで、Sun StorEdge SAM-FS ファイルシステムでアーカイブファイルが占有していたディスク領域の解放を試みる。リリーサは、ディスクキャッシュが最高ウォーターマークに到達したときに自動的に開始し、ファイルの解放が終了すると停止する。これはプロセスであり、デーモンではない。
sam-stagealld	Sun StorEdge SAM-FS ファイルの結合書き込みを制御する
sam-stagerd	Sun StorEdge SAM-FS ファイルの書き込みを制御する
sam-rpcd	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) サーバープロセスを制御する

Sun StorEdge QFS または Sun StorEdge SAM-FS 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、停止時または障害発生時には自動的に再起動します。

Sun StorEdge SAM-FS ソフトウェアを実行するとき、sam-fsd デーモンは次のプロセスを作成します。

- sam-archiverd。sam-archiverd デーモンは sam-arcopy プロセスと sam-arfind プロセスを開始します。
- sam-catserverd。このデーモンは samd(1M) stop コマンドを実行すると停止します。
- sam-rftd
- sam-initd
- sam-robotd。このデーモンは samd(1M) stop コマンドを実行すると停止します。

- `sam-scannerd`。このデーモンは `samd(1M)` `stop` コマンドを実行すると停止します。
- `sam-sharefsd`。Sun StorEdge QFS 共有ファイルシステムごとにプロセスが 1 つ作成されます。
- `sam-stagealld`
- `sam-stagerd`

## トレースファイル

Sun StorEdge QFS および Sun StorEdge SAM-FS 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の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、主にサン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。通常、トレースファイルへの書き込みはできません。Sun StorEdge SAM-FS ソフトウェアの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
- その他のイベント

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
- その他のイベント

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)、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 (時刻は常に含まれる)
- 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るないように、`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)` やその他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 9-1 に、このコマンドからの出力を示します。

コード例 9-1 `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
```

コード例 9-1 sam-fsd コマンドの出力 (続き)

```
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
License:License never expires.
```

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 StorEdge QFS および Sun StorEdge SAM-FS 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式を指定したり、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。

次の節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

ファイル属性は、`setfa(1)` コマンドを使用して設定されます。`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。その後で、他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他のファイルは、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。`-L` オプションは、既存ファイルに対して使います。ファイルは空でも、データが入っていてもかまいません。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す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後で、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、このようにファイルを事前割り当てすると、後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、ファイルを作成すると、マウント時に指定した割り当て方式とストライプ幅が使用されます (mount\_samfs(1M) のマニュアルページを参照してください)。ただし、一般ユーザーが、ファイルやファイルを含むディレクトリについて別の割り当て方式を使用することもできます。これは、setfa(1) コマンドの `-s` (ストライプ) オプションを使用して行います。

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。 `-s` オプションを使用して、割り当て方式とストライプ幅を指定します。表 9-2 に、このオプションの効果を示します。

表 9-2 ファイル割り当てとストライプ幅

<code>-s</code> <i>stripe</i>	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る。
1-255	ストライプ化	1 ~ 255 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る。

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 StorEdge 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 9-2 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させることを指定しています。

コード例 9-2      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。

---

## 大容量ファイルの格納

非常に大容量のファイルを操作するときは、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異なりますが、動作は次のようになります。

- Sun StorEdge 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` エラーが戻される。
- Sun StorEdge SAM-FS 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機する。これは、要求を処理するために使用可能な十分なディスク領域がないためです。

Sun StorEdge SAM-FS の環境で操作していて、アプリケーションが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る。`segment(1)` コマンドについては、`segment(1)` のマニュアルページか、『Sun StorEdge SAM-FS ストレージ/アーカイブ管理マニュアル』を参照してください。

---

## 複数読み取りファイルシステム

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 StorEdge 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 StorEdge 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 StorEdge 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 StorEdge 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 StorEdge 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 StorEdge QFS 共有ファイルシステムのクライアントホストを読み込み専用のホストにするには、Sun StorEdge 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、`shared` と `reader` の両方のマウントオプションを指定します。Sun StorEdge 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も設定します。Sun StorEdge QFS 共有ファイルシステムの詳細については、97 ページの「Sun StorEdge QFS 共有ファイルシステム」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限らないため、行をコピーした後にディスクコントローラの情報を更新する必要があります。

複数読み取りファイルシステム環境では、Sun StorEdge 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 StorEdge 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す。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ホストシステムにファイルシステムと同期の取れていないものがないようにできます。

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)、またはその他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います。サイトの必要に応じて、mount 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 StorEdge QFS ファイルシステムによるディレクトリや他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 StorEdge 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 StorEdge 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 StorEdge QFS ソフトウェアの両方が必要です。Sun StorEdge 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 オペレーティングシステム (OS) 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 StorEdge QFS 共有ファイルシステムを使用します。

Sun StorEdge QFS 共有ファイルシステムの詳細は、97 ページの「Sun StorEdge QFS 共有ファイルシステム」を参照してください。

Sun StorEdge 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、322 ページの「SAN-QFS 共有ファイルシステムと Sun StorEdge QFS 共有ファイルシステムの比較」を参照してください。

---

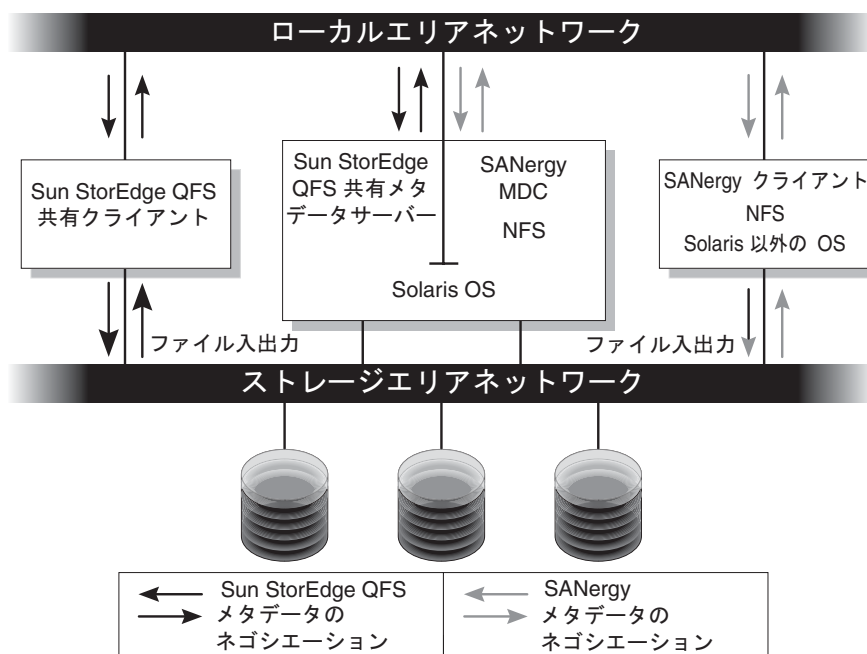
この節では、SAN-QFS ファイルシステムのその他の面について説明します。

- 315 ページの「事前確認」
- 315 ページの「SAN-QFS ファイルシステムの使用開始」
- 318 ページの「SAN-QFS ファイルシステムのマウント解除」
- 320 ページの「障害追跡: 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」
- 321 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」
- 321 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」
- 321 ページの「samgrowfs(1M) による SAN-QFS ファイルシステムの拡張」
- 322 ページの「SAN-QFS 共有ファイルシステムと Sun StorEdge QFS 共有ファイルシステムの比較」

図 9-1 は、Sun StorEdge 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 MDC システムが 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 StorEdge 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 StorEdge 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 メタデータコントローラ (MDC) としても機能します。

図 9-1 Sun StorEdge 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



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。

## 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 StorEdge 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。

- 315 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」
- 316 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」
- 317 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 StorEdge QFS ファイルシステムの常駐先となります。

1. `superuser` として、Sun StorEdge 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 StorEdge 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、Sun StorEdge QFS ファイルシステムの名前を指定します (例: qfs1)。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 環境変数を追加します。

8. 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。(省略可能)

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。

9. 316 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」に進みます。

## ▼ クライアントで SAN-QFS ファイルシステムを有効にする

SAN-QFS ファイルシステムでは、複数のクライアントホストをサポートしており、IRIX、Windows、AIX、LINUX などのホストがその対象例として挙げられます。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、同ファイルに手順 6 の `qfs_file_system_name` を追加します。結果は次のようになります。

```
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 StorEdge QFS, Sun StorEdge SAM-FS ソフトウェアインストールおよび構成の手引き』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。

3. 317 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」に進みます。

## ▼ クライアントで SANergy ソフトウェアをインストールする

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。特定のプラットフォームに関する事柄については、SANergy のマニュアルを参照してください。

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。

このコマンドを使用する場合は、次のいずれかのマウントオプションリストを指定してください。

オプションリスト 1:

```
# mount -o acregmin=0, acregmax=0, actimeo=0, noac \  
host:/mount_point/ local_mount_point
```

オプションリスト 2:

```
# mount -noac 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 StorEdge 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。

コマンドの例は次のとおりです。

```
# 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。

- 318 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」
- 319 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」
- 319 ページの「Sun StorEdge 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」
- 319 ページの「Sun StorEdge 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. `superuser` 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 StorEdge 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、Sun StorEdge QFS ファイルシステムの名前を指定します (例: `qfs1`)。 `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 StorEdge 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. `superuser` として、Sun StorEdge 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。

```
# umount /qfs1
```

## ▼ Sun StorEdge 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. `superuser` として、Sun StorEdge 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。

## 障害追跡: 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 StorEdge 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。

該当ホールドは、SANergy ファイル共有機能で消去しますが、緊急時や SANergy File Sharing のシステム障害で再起動を回避する必要がある場合は、次の手順に従って、`samunhold(1M)` コマンドを使用します。

## ▼ 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、226 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。オープンファイルのデータおよび属性については、SANergy の各クライアント共通ではありません。

## 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。このタスクを実行するには、84 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。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。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は、(作成時の) `mcf` ファイルで検出された順に番号が付いています。

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、その後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 StorEdge QFS 共有ファイルシステムの比較

SAN-QFS ファイルシステムと Sun StorEdge 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 9-3 に、両ファイルシステムの相違点を示します。

表 9-3 SAN-QFS 共有ファイルシステムと Sun StorEdge QFS 共有ファイルシステム

SAN-QFS ファイルシステム	Sun StorEdge 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る (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 環境に適している
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つ	複数のホストが書き込み可能。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る。

## 入出力パフォーマンス

Sun StorEdge QFS と Sun StorEdge SAM-FS の両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

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だけに指定してください。

`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 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は、一時的な設定になります。設定が有効なのはファイルがアクティブである間だけです。

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)` コマンドに `-o forcedirectio` オプションを指定する
- `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う

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

`Sun StorEdge QFS` と `Sun StorEdge SAM-FS` の両ファイルシステムは、自動入出力切り替えに対応しています。入出力切り換えとは、一定の容量のページ入出力が発生したら、システムによって直接入出力に切り換えるように指定することです。このような自動的 directio 切り換えを使用することで、システムでは、サイトで定義した容量の連続入出力操作を実行することができ、ページ入出力から直接入出力に自動的に切り換えることができます。

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、`dio_wr_consec` パラメタと `dio_rd_consec` パラメタを `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。`samu(1M)` を使用して有効にすることもできます。

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 StorEdge QFS および Sun StorEdge SAM-FS 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タを変更してください。

---

### ▼ 大容量ファイル転送パフォーマンスを向上する

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` パラメタで制御します。`maxphys` パラメ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
```



## 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、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、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;
```

## 4. システムを再起動します。

## 5. writebehind パラメタを設定します。

この操作はページ入出力だけに影響します。

Sun StorEdge QFS と Sun StorEdge SAM-FS のいずれかの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後書きするバイト数は、writebehind パラメタで指定します。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 StorEdge QFS ファイルシステムの場合、DAU (samkfs(1M) -a コマンド) も RAID 5 ストライプサイズの倍数である必要があります。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後で、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) ユーティリティのコマンド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 StorEdge QFS と Sun StorEdge SAM-FS のいずれかの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 パラメタで指定します。このパラメ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後で、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
```

readahead パラメタは、ページ入出力の入出力パフォーマンスを向上するサイズに設定する必要があります。また、readahead のサイズを大きくしすぎると、パフォーマンスが損なわれる可能性があることにも注意してください。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 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が重要です。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 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 readahead があるため、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) ユーティリティのコマンドで有効にできます。マウントオプションで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。

## 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、 $n * \text{DAU}$  バイトが装置に書き込まれてから、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。各ファイルは次の装置に作成されます。その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す。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

Qwrite 機能は、Sun StorEdge QFS 環境で有効にできます。

デフォルトの Sun StorEdge 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、その他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 (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

デフォルトの Sun StorEdge QFS と Sun StorEdge SAM-FS の両ファイルシステムでは、mount(1M) コマンドの -o wr\_throttle=*n* オプションが 16M バイトに設定されます。-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。

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)、メモリー無効を回避するために -o wr\_throttle=*n* オプションを調整できます。一般的に、-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。つまり、次のようになります。

```
 $number\_of\_streams * n * 1024 < total\_memory - Solaris\_OS\_memory\_needs$ 
```

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 16,384 キロ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、Sun Solaris VM 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  $16 \leq n \leq 8192$  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、Sun Solaris VM 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  $16 \leq n \leq 8192$  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 StorEdge QFS および Sun StorEdge SAM-FS 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タを設定できます。

- ninodes
- nhino

これらのパラメ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。

この後の各節で、これらのパラメタについて詳しく説明します。

### ninodes パラメタ

ninodes パラメタは、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 StorEdge QFS と Sun StorEdge SAM-FS 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  $16 \leq \text{value} \leq 2000000$  です。ninode のデフォルトの *value* は次のどれかになります。

- *ncsize* 設定に等しい *value*。ncsize パラメタは 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 または範囲外である場合、ファイルシステムは ninode を 2000 に設定する  
コマンドの例は次のとおりです。

```
set samfs:ninode = 4000
```

## nhino パラメタ

nhino パラメタは、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  $1 \leq \text{value} \leq 1048756$  で、かつ 2 の累乗 (0 を除く) にする必要があります。nhino のデフォルトの *value* は次のどれかになります。

- *ninode* の値に等しい値を 8 で割り、必要であれば最も近い 2 の累乗に切り上げる。たとえば、/etc/system に次の行があるとします。

```
set samfs:ninode 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、最も近い 2 の累乗に切り上げたものです。

- 512。ninode の設定が範囲外である場合、ファイルシステムは nhino を 512 に設定する

コマンドの例は次のとおりです。

```
set samfs:nhino = 1024
```

## ninodes パラメタと nhino パラメ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 StorEdge QFS および Sun StorEdge SAM-FS 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、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す。

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)。大量のファイル进行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) ファイルに他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。



# 用語集

---

---

## D

**DAU** (ディスク割り当て単位) オンライン記憶領域の基本単位。ブロックサイズとも呼ばれます。

また、Sun StorEdge QFS のファイルシステムでは、16 K バイトから 65,528 K バイトまでのサイズの完全に調整可能な DAU もサポートしています。DAU は、8 K バイトの倍数で指定する必要があります。

Sun StorEdge SAM-FS のファイルシステムは、小型 DAU と大型 DAU の両方をサポートしています。小型 DAU は、4 K バイトです ( $2^{14}$  つまり 4096 バイト)。大型 DAU は、16 K、32 K、または 64 K バイトです。利用できる DAU のサイズのペアは、4/16、4/32、および 4/64 です。

---

## F

**FDDI** (Fiber Distributed Data Interface) 100 M バイト/秒の光ファイバ LAN。

**fiber-distributed data interface** 「FDDI」参照。

**FTP** (ファイル転送プロトコル) 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。

---

## 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)。Sun StorEdge QFS および Sun StorEdge SAM-FS のすべての i ノードは長さが 512 バイトです。i ノードファイルは、Sun StorEdge QFS の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。

---

## M

- mcf** マスター構成ファイル (Master Configuration File)。Sun StorEdge QFS または Sun StorEdge SAM-FS 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。

---

## 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)

---

## 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、サンの分散ファイルシステムです。
- NIS** SunOS 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。

---

## P

**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。

---

## 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。

---

## 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** 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。

**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。

**small computer system interface** 「SCSI」参照。

**Sun StorEdge QFS** 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た、高速な UNIX ファイルシステム。Sun StorEdge QFS ソフトウェアは、保管されているすべてのファイルへのアクセス、およびマスター構成ファイル (mcf) に設定されているすべての装置へのアクセスを制御します。

**Sun StorEdge SAM-FS** Sun Storage Archive Manager File System。Sun StorEdge SAM-FS ソフトウェアは、保管されているすべてのファイルへのアクセス、およびマスター構成ファイル (mcf) に設定されているすべての装置へのアクセスを制御します。

**Sun SAM-QFS** Sun SAM-QFS ソフトウェアは、Sun StorEdge SAM-FS ソフトウェアと Sun StorEdge QFS のファイルシステムを結合します。Sun 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。Sun SAM-QFS は、Sun StorEdge SAM-FS 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。

#### Sun SAM-Remote クライアント

Sun SAM-Remote クライアントは、多数の擬似デバイスが含まれた Sun SAM-Remote クライアントデーモンを構成する Sun StorEdge SAM-FS システムです。Sun SAM-Remote クライアントには、専用のライブラリデバイスがある場合とない場合とがあります。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

Sun SAM-Remote サーバーは、全容量の Sun StorEdge SAM-FS 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です。

---

## T

**tar** テープアーカイブ。Sun StorEdge SAM-FS ソフトウェアで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

**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にアーカイブを行っている場合、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です。ディスクキャッシュにアーカイブを行っている場合は、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 StorEdge QFS または Sun StorEdge SAM-FS 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、10 M バイトまたは 100 M バイト/秒の LAN です。
--------	---

---

## え

遠隔手続き呼び出し	「RPC」参照。
-----------	----------

---

## お

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。
オフサイト記憶装置	サーバーから遠隔地にあつて災害回復に使用される記憶装置。
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。
- カートリッジ** データを記録するための媒体を含む物体。(テープまたは光磁気ディスク)「メディア」、「ボリューム」、または「媒体」と呼ぶこともあります。
- 外部配列** ファイルに割り当てられた各データブロックが、ディスク上のどこにあるかを定義する、ファイルの i ノード内の配列。
- 解放優先順位** さまざまなウェイトにそれぞれ対応するファイル属性を乗算し、その結果を集計することによって、ファイルシステム内のファイルの解放優先順位を求める方法。
- 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。
- 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。
- 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。Sun StorEdge QFS および Sun StorEdge SAM-FS の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。
- 共有ライター/共有リーダー**  
Sun StorEdge QFS の共有ライター/共有リーダー機能は、複数のサーバーで共有するファイルシステムを指定する機能です。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。

---

## し

### しきい値

オンライン記憶装置に適した利用可能な記憶装置ウィンドウを定義するメカニズム。しきい値により、リリーサのストレージ目標が設定されます。「ディスク容量しきい値」も参照してください。

### 事前割り当て

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。この結果、この領域が隣接することが保証されます。事前割り当ては、サイズがゼロのファイルに対してだけ行えます。つまり、`setfa -l` コマンドは、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。

### 自動ライブラリ

オペレータが処置を必要としない、リムーバブルメディアカートリッジを自動的に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

ファイルシステムの基本パラメータを定義する、ファイルシステム内のデータ構造。スーパーブロックは、ストレージファミリセット内のすべてのパーティションに書き込まれ、セットにおけるパーティションのメンバーシップを識別します。

ストライプ化	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。Sun StorEdge QFS および Sun StorEdge SAM-FS のすべてのファイルシステムでは、個別のファイルシステムに対してストライプ化アクセスとラウンドロビンアクセスのどちらでも宣言できます。Sun StorEdge QFS のファイルシステムでは、各ファイルシステム内でストライプ化グループを宣言できます。「ラウンドロビン」に関する項目も参照。
ストライプ化グループ	mcf ファイルで 1 つ (通常は 2 つ) 以上の gXXX デバイスとして定義された、Sun StorEdge QFS の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。1 つのファイルシステム内に指定できるストライプ化グループは最大 128 個ですが、指定可能な総デバイス数は 252 個です。
ストライプサイズ	ストライプの次のデバイスに移動する前に割り当てるディスク割り当て単位 (DAU) の数。stripe=0 の場合、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。
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。ライブラリが直接接続されている場合、ストレージスロットの内容は自動ライブラリのカタログに保管されます。

---

## 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 接続されたライブラリは、自動ライブラリに SCSI 規格を使用することで、Sun StorEdge SAM-FS ソフトウェアによって直接制御されます。

**直接入出力** 大型ブロック整合逐次入出力に使用される属性の 1 つ。setfa(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** Sun StorEdge SAM-FS 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。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します。個々のディスクパーティションまたはディスク全体で、ディスクキャッシュとして使用できます。

**ディスクのストライプ化** アクセスパフォーマンスの向上と全体的な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。「ストライプ化」に関する項目も参照。

**ディスクバッファ** Sun SAM-Remote ソフトウェアを使用している場合、ディスクバッファとは、クライアントからサーバーにデータをアーカイブするとき使用するサーバー上のバッファです。

**ディスク容量しきい値** 管理者によって定義された、ユーザーが利用できるディスク容量。この値によって、望ましいディスクキャッシュ利用率の範囲が決まります。上限値は、ディスクキャッシュ利用率の最大レベルを示します。下限値は、ディスクキャッシュ利用率の最小レベルを示します。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」参照。
ディレクトリ	ファイルシステム内のその他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	Sun StorEdge QFS または Sun StorEdge SAM-FS の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、ユーザーや他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、Sun StorEdge SAM-FS のファイルシステム内にあるソフトウェア。
デバイスのログ	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 StorEdge SAM-FS 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 StorEdge SAM-FS 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  
バックアップ記憶装置

デバイスの一部または光磁気カートリッジの片面。

不注意によるファイルの消去を防ぐことを目的とした、ファイル群のスナップショット。バックアップには、ファイルの属性と関連データの両方が含まれます。

---

## ふ

ファイバチャネル  
ファイルシステム  
ファイルシステム固有指示

デバイス間の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。

階層構造によるファイルとディレクトリの集まり。

グローバル指示の後のアーカイブ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、fs = で始まります。ファイルシステム固有指示は、次の fs = 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  
ファミリデバイスセット  
プレビュー要求の優先順位決定

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。

「ファミリセット」参照。

すぐには応答できないアーカイブ要求と書き込み要求に優先順位を設定すること。

ブロックサイズ 「DAU」 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。  
新されます。

---

## ほ

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すること。

---

## め

メディア テープカートリッジまたは光磁気ディスクカートリッジ。

メディアリサイクリング 利用率の低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。利用率の低いアーカイブメディアとは、アーカイブファイル数の少ないアーカイブメディアのことです。

**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必要な索引情報です。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。データが消去されたとき、データを表すメタデータを復元しなければ消去データを取り戻せないため、メタデータは保護される必要があります。

**メタデータデバイス** Sun StorEdge QFS のファイルシステムのメタデータを保存する、独立したデバイス (ソリッドステートディスクやミラーデバイスなど)。メタデータからファイルデータを切り離すと、パフォーマンスを向上させることができます。メタデータデバイスは、`ma` ファイルシステム内の `mm` デバイスであると、`mcf` ファイルにおいて宣言されます。

---

## ら

**ライブラリ** 「自動ライブラリ」参照。

**ライブラリカタログ** 「カタログ」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。

デフォルトでは、ストライプ化グループが存在する場合を除いて、Sun StorEdge QFS および Sun StorEdge SAM-FS のファイルシステムはストライプ化データアクセスを実装します。ラウンドロビン式アクセスが指定されている場合、ファイルはラウンドロビンされます。正しくないストライプ化グループがファイルシステムに存在している場合、ストライプ化はサポートされず、ラウンドロビンが強制されます。

「ディスクのストライプ化」と「ストライプ化」の項目も参照。

---

## り

**リリーサ** アーカイブされたファイルを識別し、そのディスクキャッシュコピーを開放することで、利用可能なディスクキャッシュ空間を増やす、Sun StorEdge SAM-FS 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。

**リース** Sun StorEdge QFS 共有ファイルシステムにおいて、リースは、リースが有効である期間中、ファイルに対する操作を行うためのクライアントホストのアクセス権を許可します。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ローカルファイルシステム

Sun Cluster 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。他のノードからは、あまり利用されません。スタンドアロン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。

**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 StorEdge SAM-FS 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。

---

## ろ

**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。

---

## 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です。

---

## 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。指定時間 (デフォルトは 1 週間) 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。

---

# 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

---

## 記号

! コマンド (samu(1M) コマンド), 221  
/dev/dsk のエントリ, 43  
/dev/rmt エントリ, 43  
/dev/samst エントリ, 43  
/etc/opt/SUNWsamfs/LICENSE.4.1  
「LICENSE.4.1 ファイル」を参照  
/etc/system ファイル, 324  
/etc/vfstab ファイル  
「vfstab ファイル」を参照  
/kernel/drv/sd.conf ファイル, 325

## 数字

0 割り当て, 234

## A

Additional parameters フィールド, 46  
API ルーチン  
aplease コマンド (samu(1M) コマンド), 213  
aplease マウントオプション, 125  
archdone ファイル属性, 14  
archive(1) コマンド, 11  
archiver.cmd, 89  
aridle samu(1M) コマンド, 203  
arrerrun samu(1M) コマンド, 203

arrestart samu(1M) コマンド, 203  
arrmarchreq samu(1M) コマンド, 203  
arrun samu(1M) コマンド, 203  
arscan samu(1M) コマンド, 203  
arstop samu(1M) コマンド, 203  
artrace samu(1M) コマンド, 203  
audit ロボットコマンド (samu(1M) コマンド), 217

## C

clear コマンド (samu(1M) コマンド), 219

## D

damaged ファイル属性, 14

### DAU

gXXXX, 21  
md, 21  
mr, 21  
概要, 5, 18  
シングル割り当て方式, 20  
設定, 19  
デュアル割り当て方式, 19  
割り当ておよび DAU, 226

defaults.conf ファイル, 89

devlog コマンド (samu(1M) コマンド), 219

dio\_rd\_consec コマンド (samu(1M) コマンド), 212

dio\_rd\_consec パラメタ, 324

dio\_rd\_form\_min コマンド (samu(1M) コマンド), 211  
dio\_rd\_ill\_min コマンド (samu(1M) コマンド), 211  
dio\_wr\_consec コマンド (samu(1M) コマンド), 212  
dio\_wr\_consec パラメタ, 324  
dio\_wr\_form コマンド (samu(1M) コマンド), 211  
dio\_wr\_ill\_min コマンド (samu(1M) コマンド), 211  
directio(3C) 関数呼び出し, 3, 322  
diskvols.conf ファイル, 98  
down samu(1M) コマンド, 202  
dsk エントリ, 43  
dtrace コマンド (samu(1M) コマンド), 219

## E

EDOM エラー, 76  
EDQUOT エラー, 225  
ENOCSE エラー, 76  
export ロボットコマンド (samu(1M) コマンド), 217

## F

flush\_behind コマンド (samu(1M) コマンド), 208  
flush\_behind マウントパラメタ, 329  
forcedirectio コマンド (samu(1M) コマンド), 212  
force\_nfs\_async コマンド (samu(1M) コマンド), 209  
fsck(1M) コマンド 「samfsck(1M) コマンド」 も参照, 4  
fs コマンド (samu(1M) コマンド), 220

## G

gXXX 装置, 45

## H

hwm\_archive コマンド (samu(1M) コマンド), 205

## I

idle samu(1M) コマンド, 202  
import ロボットコマンド (samu(1M) コマンド), 218  
invalid コマンド (samu(1M) コマンド), 214  
i ノード  
    ファイルの内容, 10

## L

LICENSE.4.1 ファイル, 93, 95, 168  
load ロボットコマンド (samu(1M) コマンド), 218

## M

maxallocsz コマンド (samu(1M) コマンド), 213  
maxallocsz マウントオプション, 124  
maxcontig 設定, 327  
maxpartial コマンド (samu(1M) コマンド), 206  
maxphys パラメタ, 324  
ma ファイルシステム, 44  
mcf  
    /dev/dsk のエントリ, 43  
    /dev/rmt エントリ, 43  
    /dev/samst エントリ, 43  
    additional parameters フィールド, 46  
    device state フィールド, 46  
    equipment identifier フィールド, 43  
    equipment ordinal フィールド, 44  
    equipment type フィールド, 44  
    エラーの確認, 88  
    エントリ, 42  
    構成, 42  
    サーバーのアップグレード, 89  
    サンプルファイル, 52  
    ファイルシステムサイズの増加, 85, 88  
    フィールド, 42

md ファイルシステム, 44  
meta\_timeo  
マウントオプション, 128  
meta\_timeo ファイルシステムコマンド (samu(1M)  
コマンド), 213  
mhwrite コマンド (samu(1M) コマンド), 213  
mh\_write マウントオプション, 126  
minallocsz コマンド (samu(1M) コマンド), 213  
minallocsz マウントオプション, 124  
mm\_stripe コマンド (samu(1M) コマンド), 214  
mm ファイルシステム, 44  
mount(1M) コマンド, 50, 70, 72, 93  
mount コマンド (samu(1M) コマンド), 220  
mr ファイルシステム, 44  
ms ファイルシステム, 44

## N

nodev キーワード, 43  
noforcedirectio コマンド (samu(1M) コマン  
ド), 212  
noforce\_nfs\_async コマンド (samu(1M) コマン  
ド), 209  
nohwm\_archive コマンド (samu(1M) コマン  
ド), 205  
nomhwrite コマンド (samu(1M) コマンド), 213  
noqwrite コマンド (samu(1M) コマンド), 214  
norefresh\_at\_eof コマンド (samu(1M) コマン  
ド), 215  
nosetuid コマンド (samu(1M) コマンド), 215  
nosw\_raid コマンド (samu(1M) コマンド), 210  
notrace ファイルシステムコマンド (samu(1M) コ  
マンド), 217  
nstreams マウントオプション, 127

## O

offline ファイル属性, 14  
off samu(1M) コマンド, 202  
on samu(1M) コマンド, 202

open コマンド (samu(1M) コマンド), 220

## P

partial\_stage コマンド (samu(1M) コマンド), 206  
partial ファイルシステムコマンド (samu(1M) コマ  
ンド), 206  
pkgadd(1M) コマンド, 92, 95  
pkgrm(1M) コマンド, 92, 94  
priority コマンド (samu(1M) コマンド), 218

## Q

qfsdump(1M) コマンド, 90  
qfsrestore(1M) コマンド, 89  
Qwrite, 328  
qwrite コマンド (samu(1M) コマンド), 214

## R

rdlease コマンド (samu(1M) コマンド), 213  
rdlease マウントオプション, 125  
readahead  
ファイルシステムコマンド (samu(1M) コマン  
ド), 209  
マウントパラメタ, 327  
README ファイル, 121  
read コマンド (samu(1M) コマンド), 221  
refresh\_at\_eof コマンド (samu(1M) コマンド), 215  
release(1) コマンド, 11, 12  
restore.sh(1M) コマンド, 89  
rmt エントリ, 43

## S

sam\_archive(3) API ルーチン, 11  
sam-archiverd デーモン, 304  
samchaid(1M) コマンド, 227, 234  
samcmd(1M) コマンド, 70, 74

samd(1M) コマンド, 74, 85, 88, 91, 304  
 samfsck(1M) コマンド, 45, 71, 76, 77  
 samfs.cmd ファイル, 50, 70, 72, 324, 326  
 sam-fsd, 122  
 sam-fsd(1M) コマンド, 306  
 samfsd(1M) コマンド, 88  
 samfsdump(1M) コマンド, 90  
 sam-fsd デーモン, 111, 115, 304  
 samfsinfo(1M) コマンド, 50  
 samfsrestore(1M) コマンド, 89  
 sam-genericd デーモン, 304  
 samgrowfs(1M) コマンド, 84, 86  
 sam-log ファイル, 76  
 sammkfs(1M) コマンド, 21, 45, 50, 88  
 samquota(1M) コマンド, 227, 228  
 samquotastat(1M) コマンド, 227  
 sam\_release(3) API ルーチン, 11  
 sam-releaser プロセス, 304  
 sam-rftd デーモン, 304  
 sam-robotsd デーモン, 304  
 sam-rpcd デーモン, 304  
 sam-scannerd デーモン, 304  
 sam\_segment(3) API ルーチン, 11  
 sam\_setfa(3) API ルーチン, 11, 323  
 sam-sharedfsd, 122  
 sam\_ssum(3) API ルーチン, 11  
 sam\_stage(3) API ルーチン, 11  
 sam-stagealld デーモン, 304  
 sam-stagerd デーモン, 304  
 samst エントリ, 43  
 samu(1M)  
   ! コマンド, 221  
   aplease コマンド, 213  
   aridle コマンド, 203  
   arrerun コマンド, 203  
   arrestart コマンド, 203  
   arrmarchreq コマンド, 203  
   arrun コマンド, 203  
   arscan コマンド, 203  
   arstop コマンド, 203  
   artrace コマンド, 203  
   audit ロボットコマンド, 217  
   clear コマンド, 219  
   devlog コマンド, 219  
   dio\_rd\_consec コマンド, 212  
   dio\_rd\_form\_min コマンド, 211  
   dio\_rd\_ill\_min コマンド, 211  
   dio\_wr\_consec コマンド, 212  
   dio\_wr\_form\_min コマンド, 211  
   dio\_wr\_ill\_min コマンド, 211  
   down コマンド, 202  
   dtrace コマンド, 219  
   export ロボットコマンド, 217  
   flush\_behind コマンド, 208  
   forcedirectio コマンド, 212  
   force\_nfs\_async コマンド, 209  
   fs コマンド, 220  
   hwm\_archive コマンド, 205  
   idle コマンド, 202  
   import ロボットコマンド, 218  
   invalid コマンド, 214  
   load ロボットコマンド, 218  
   maxallocsz コマンド, 213  
   maxpartial コマンド, 206  
   mcf ファイルとのやりとり, 151  
   meta\_timeo ファイルシステムコマンド, 213  
   mhwwrite コマンド, 213  
   minallocsz コマンド, 213  
   mm\_stripe コマンド, 214  
   mount コマンド, 70, 220  
   noforcedirectio コマンド, 212  
   noforce\_nfs\_async コマンド, 209  
   nohwm\_archive コマンド, 205  
   nomhwwrite コマンド, 213  
   noqwrite コマンド, 214  
   norefresh\_at\_eof コマンド, 215  
   nosetuid コマンド, 215  
   nosw RAID コマンド, 210  
   notrace ファイルシステムコマンド, 217  
   off コマンド, 202  
   on コマンド, 202  
   open コマンド, 220  
   partial\_stage コマンド, 206

partial ファイルシステムコマンド, 206  
 priority コマンド, 218  
 qwrite コマンド, 214  
 rdlease コマンド, 213  
 readahead ファイルシステムコマンド, 209  
 read コマンド, 221  
 refresh\_at\_eof コマンド, 215  
 samu(1M) による状態の表示, 202  
 setuid コマンド, 215  
 snap コマンド, 221  
 stage\_flush\_behind コマンド, 207  
 stage\_n\_window コマンド, 207  
 stage\_retries コマンド, 207  
 stclear コマンド, 208  
 stidle コマンド, 208  
 stripe コマンド, 215  
 strun コマンド, 208  
 sw\_raid コマンド, 210  
 sync\_meta コマンド, 216  
 thresh ファイルシステムコマンド, 205  
 trace ファイルシステムコマンド, 217  
 unavail コマンド, 202  
 unload コマンド, 202  
 writebehind ファイルシステムコマンド, 210  
 wrlease コマンド, 213  
 wr\_throttle コマンド, 210  
 インタフェース, 151  
 オペレータ表示, 152  
 状態コード, 197  
 ディスプレイ制御キー, 151  
 デーモントレースコマンド, 219  
 表示制御 samu(1M) コマンド, 202  
 ファイルシステムコマンド, 211, 214  
 呼び出しコマンド, 149  
 ロボットコマンド, 217  
 samu(1M) の保留書き込み, 195  
 samunhold(1M) コマンド, 320  
 SAN-QFS ファイルシステム  
   SANergy ファイルホルドの解放, 320  
   Sun StorEdge QFS 共有ファイルシステムとの比較, 322  
   概要, 312  
   有効化, 315  
 sd\_max\_xfer\_size 定義, 325  
 segment(1) コマンド, 11, 13, 310  
 setfa(1) コマンド, 3, 11, 307, 323  
 setuid コマンド (samu(1M) コマンド), 215  
 sls(1) コマンド, 15  
 snap コマンド (samu(1M) コマンド), 221  
 Solaris のアップグレード, 90, 93  
 squash(1) コマンド, 227  
 ssd\_max\_xfer\_size 定義, 325  
 ssum(1) コマンド, 11  
 stage(1) コマンド, 11, 12  
 stage\_flush\_behind コマンド (samu(1M) コマンド), 207  
 stage\_flush\_behind マウントパラメタ, 329  
 stage\_n\_window コマンド (samu(1M) コマンド), 207  
 stage\_retries コマンド (samu(1M) コマンド), 207  
 stclear コマンド (samu(1M) コマンド), 208  
 st.conf ファイル, 89  
 stidle コマンド (samu(1M) コマンド), 208  
 stripe コマンド (samu(1M) コマンド), 215  
 stripe マウントオプション, 128  
 strun コマンド (samu(1M) コマンド), 208  
 Sun SAM-QFS  
   「Sun StorEdge SAM-FS」も参照  
   共有ファイルシステム  
     「共有ファイルシステム」を参照  
     定義, xxi  
 Sun StorEdge QFS  
   共有ファイルシステム  
     「共有ファイルシステム」を参照  
     定義, xxi  
 Sun StorEdge QFS 共有ファイルシステム  
   「共有ファイルシステム」を参照  
 Sun StorEdge SAM-FS  
   相互運用性, 6  
   定義, xxi  
 sw\_raid コマンド (samu(1M) コマンド), 210  
 sync\_meta コマンド (samu(1M) コマンド), 216  
 sync\_meta マウントオプション, 128

## T

tee(1M) コマンド, 77  
thresh ファイルシステムコマンド (samu(1M) コマンド), 205  
trace\_rotate(1M) コマンド, 306  
trace ファイルシステムコマンド (samu(1M) コマンド), 217

## U

unavail samu(1M) コマンド, 202  
unload samu(1M) コマンド, 202

## V

VFS, 2  
vfstab ファイル, 2, 50, 70, 71, 93, 95, 116  
vnode インタフェース  
「VFS」を参照

## W

writebehind  
ファイルシステムコマンド (samu(1M) コマンド), 210  
マウントパラメータ, 326  
wrlase コマンド (samu(1M) コマンド), 213  
wrlase マウントオプション, 125  
wr\_throttle コマンド (samu(1M) コマンド), 210  
wr\_throttle マウントパラメータ, 329

## あ

アーカイバ  
samu(1M) の表示, 153  
wait モード, 83  
アーカイブメディアの指定, 98  
割り当てとアーカイブメディア, 226  
アップグレード  
Solaris, 93

Sun Solaris OS, 90  
一般的な情報, 61  
サーバー, 89  
ディスク, 86  
パーティション, 86  
アプリケーションプログラミングインタフェース  
ルーチン  
「API ルーチン」を参照

## お

大型  
DAU 「DAU」を参照  
ファイル, 310  
オペレータユーティリティ 「samu(1M)」を参照

## か

管理者  
ユーティリティ 「samu(1M)」を参照  
管理セット割り当て, 223, 233

## き

キャッシュ  
attributes, 128  
入出力 「入出力」を参照、ページ  
共有ファイルシステム  
aplease マウントオプション, 125  
diskvols.conf ファイル, 98  
maxallocsz マウントオプション, 124  
mcf ファイル, 43  
meta\_timeo マウントオプション, 128  
mh\_write マウントオプション, 126  
minallocsz マウントオプション, 124  
nodev キーワード, 43  
nstreams マウントオプション, 127  
sam-fsd デーモン, 111, 115  
SAN-QFS 共有ファイルシステムとの比較, 322  
stripe マウントオプション, 128  
sync\_meta マウントオプション, 128  
wrlase マウントオプション, 125

- アーカイブメディアの指定, 98
- キャッシュした属性, 128
- クライアントホストの削除, 117
- クライアントホストの追加, 106
- デーモン, 122
- ファイルシステムのマウント, 105
- ファイルロック, 130
- フェイルオーバー, 119
- マウントオプション, 123
- マウントの意味, 129
- マウントポイント, 97
- メタデータサーバーの変更, 119
- リース, 125
- 共有ファイルシステムの `bg` マウントオプション, 123
- 共有ファイルシステムの `retry` マウントオプション, 124
- 共有ファイルシステムの `shared` マウントオプション, 124
- 共有ファイルシステムのスレッド, 127

## こ

### 構成

- `mcf` ファイルの作成, 42
- `samu(1M)` の表示, 156
- ディスク使用, 18
- ファイル「`mcf`」を参照
- ファイル割り当て, 26

高度な機能, 303

### 小型 DAU

「DAU」を参照

この, 51

### コマンド

- `archive(1)`, 11
- `directio(3C)`, 3, 322
- `fsck(1M)` 「`samfsck(1M)` コマンド」も参照, 4, 70
- `ls(1)` 「`sls(1)` コマンド」も参照, 15
- `mount(1M)`, 50, 70, 72, 93
- `pkgadd(1M)`, 92, 95
- `pkgrm(1M)`, 92, 94
- `qfsdump(1M)`, 90

- `qfsrestore(1M)`, 89
- `release(1)`, 11
- `restore.sh(1M)`, 89
- `sam_archive(3)` API ルーチン, 11
- `samchaid(1M)`, 227, 234
- `samcmd(1M)`, 70, 74
- `samd(1M)`, 74, 85, 88, 91, 304
- `samfsck(1M)`, 45, 71, 76, 77
- `sam-fsd(1M)`, 88, 306
- `samfsdump(1M)`, 90
- `samfsinfo(1M)`, 50
- `samfsrestore(1M)`, 89
- `samgrowfs(1M)`, 84, 86
- `sammkfs(1M)`, 21, 45, 50, 88
- `samquota(1M)`, 227, 228
- `samquotastat(1M)`, 227
- `sam_release(3)` API ルーチン, 11
- `sam_segment(3)` API ルーチン, 11
- `sam_setfa(3)` API ルーチン, 11, 323
- `sam_ssum(3)` API ルーチン, 11
- `sam_stage(3)` API ルーチン, 11
- `samu(1M)`, 149
- `samunhold(1M)`, 320
- `segment(1)`, 11, 310
- `setfa(1)`, 3, 11, 307, 323
- `sls(1)`, 15
- `squota(1)`, 227
- `ssum(1)`, 11
- `stage(1)`, 11
- `tee(1M)`, 77
- `trace_rotate(1M)`, 306

## さ

サーバー, アップグレード, 89

### 削除

- ソフトウェア, 92, 94
- 割り当て, 241, 249

## し

### 状態コード

`samu(1M)` による表示, 197, 198

## す

- スーパーブロック, 21, 51, 62, 321
- ステージ
  - samu(1M) による状態表示, 173, 187
  - samu(1M) の保留書き込み, 195
- ストライプ幅
  - データディスク, 23
  - メタデータディスク, 26
- ストライプ化割り当て, 128
  - mcf 内の装置, 44
  - Sun SAM-FS サンプルファイル, 56
  - Sun StorEdge QFS サンプルファイル, 55
  - Sun StorEdge QFS ストライプ化グループのサンプルファイル, 58
  - 概要, 5
  - ストライプ化グループ, 45
  - ストライプ幅, 309, 327
  - ユーザー指定, 309
- ストライプ化
  - グループ割り当て (ファイル属性), 12
- ストレージの状態表示 (samu(1M)), 170
- ストレージ・アーカイブマネージャー
  - 「Sun StorEdge SAM-FS」または「Sun SAM-QFS」を参照

## そ

- 装置
  - identifier フィールド, 43
  - ordinal フィールド, 44
  - type フィールド, 44
- ソフトウェア
  - アップグレード, 61
  - 取り出し, 92, 94
- ソフトウェアのアンインストール, 92, 94

## ち

- 直接入出力
  - 「入出力」を参照

## つ

- 強い制限値, 225

## て

- ディスク
  - キャッシュの追加, 84
  - キャッシュを超えるファイル, 310
  - 追加, 変更, 削除, 86
  - 割り当て単位「DAU を参照」
- ディレクトリ属性, 308
- データ
  - ストライプ化「ストライプ化割り当て」を参照
  - 整合, 25
- テープ
  - samu(1M) によるドライブ状態表示, 185
- sam-catserverd デーモン, 304
- デーモン
  - sam-archiverd, 304
  - sam-catserverd, 304
  - sam-fsd, 111, 115, 122, 304
  - sam-genericd, 304
  - sam-releaser, 304
  - sam-rftd, 304
  - sam-robotsd, 304
  - sam-rpcd, 304
  - sam-scannerd, 304
  - sam-sharedfsd, 122
  - sam-stagealld, 304
  - sam-stagerd, 304
  - samu(1M) の表示, 159
  - 概要, 303
  - 共有ファイルシステム, 122
  - トレース, 305
  - トレースコマンド (samu(1M) コマンド), 219
- デバイス
  - devlog samu(1M) コマンド, 202
  - samu(1M) による状態表示, 183
  - samu(1M) のコマンド, 202
  - state フィールド, 46
  - コード、samu(1M) による表示, 197, 198
  - 状態、samu(1M) による表示, 199
- デュアル割り当て方式, 19



## と

- トレースファイル, 305
- トレースファイルの切り換え, 306

## に

- 入出力
  - direct, 3, 322, 323
  - 切り換え, 323
  - 調整, 322
  - 直接入出力ファイル属性, 12
  - ページ, 3, 322, 326, 327

## は

- パーティション (追加, 変更, 削除), 86
- ハードウェアのアップグレード, 61
- 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表示, 176

## ふ

- ファイル
  - archdone 属性, 14
  - attributes, 10, 11, 14, 307
  - damaged 属性, 14
  - i ノードの内容, 10
  - offline 属性, 14
  - メタデータ, 10
  - ユーザー設定, 11
- ファイルシステム
  - capacity, 4
  - samu(1M) による表示, 168
  - samu(1M) のコマンド, 211, 214
  - 回復, 4
  - 基本操作, 61
  - 共有 「共有ファイルシステム」 を参照

## 修復, 77

- タイプ ma, 44
- タイプ md, 44
- タイプ mm, 44
- タイプ mr, 44
- タイプ ms, 44
- 妥当性検査, 76
- 名前の変更, 84
- 破損, 76
  - 割り当て 「割り当て」 を参照
- ファイルシステムの回復, 4
- ファイルシステムの修復, 77
- ファイルシステムの名前の変更, 84
- ファイル領域の事前割り当て, 3, 308
- ファイル割り当て
  - ストライプ化, 26, 29
  - ストライプ化グループ, 33
  - 不一致のストライプ化グループ, 36
  - 方法, 309
  - ラウンドロビン, 26, 27
  - 領域の事前割り当て, 308
- 複数のホストの読み取りと書き込み, 126
- 複数読み取りファイルシステム, 310
- プロセス
  - 「デーモン」 を参照

## へ

- ページ入出力
  - 「入出力」 を参照

## ほ

- ボリューム管理, 41, 47

## ま

- マウントオプション
  - wr\_throttle, 329
  - 共有ファイルシステム, 123
  - 割り当て, 224

## む

無限割り当て, 234

## め

メタデータ

mcf 内の装置, 44

概要, 5

サーバー「共有ファイルシステム」を参照

内容, 10

分離, 10

メッセージファイル, 76, 85, 88

メディア

samu(1M) による状態表示, 181

samu(1M) による読み込み要求の表示, 178

基本操作, 149

メディアのインポート

samu(1M), 218

メディアのエクスポート

samu(1M), 217

メディアの読み込み解除

samu(1M), 202

## よ

弱い制限値, 225

## ら

ライセンス

samu(1M) の表示, 168

インストールキー, 93, 95

概要, xxvii

ライセンスのアップグレード, 90

ラウンドロビン式割り当て

mcf 内の装置, 44

Sun SAM-FS サンプルファイル, 54

Sun StorEdge QFS サンプルファイル, 52

ユーザー指定, 309

## り

リリース, 125

## ろ

ロボット

samu(1M) のコマンド, 217

## わ

割り当て

0, 234

default, 235

アーカイブメディア, 226

概要, 223

管理セット, 223, 233

検査, 238

構成, 228, 230

修正, 250

ディスクブロックとファイル割り当て, 226

取り出し, 241, 249

変更, 241

無限, 234

無効化, 246

有効化, 226

猶予期間, 241, 244

弱い制限値, 225

割り当てでの DAU, 226

割り当てファイル, 224

割り当てサイズの調整, 124

割り当ての検査, 238

割り当ての修正, 250

割り当ての変更, 241

割り当ての無効化, 246

割り当ての有効化, 226

割り当ての猶予期間, 241, 244