



# Sun StorageTek™ QFS 파일 시스템 구성 및 관리 안내서

---

버전 4, 업데이트 6

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

부품 번호: 820-1717-10  
2007년 5월, 개정판 A

본 설명서에 대한 의견은 다음 사이트로 보내 주십시오. <http://www.sun.com/hwdocs/feedback>

Copyright 2007 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 모든 권리는 저작권자의 소유입니다.

Sun Microsystems, Inc.는 본 설명서에서 사용하는 기술과 관련한 지적 재산을 보유하고 있습니다. 특히 이러한 지적 재산권에는 <http://www.sun.com/patents>에 나열된 하나 이상의 미국 특허 및 추가 특허 또는 미국 및 기타 국가에서 특허 출원 중인 응용 프로그램이 포함될 수 있습니다.

본 제품 또는 설명서는 사용, 복사, 배포 및 역컴파일을 제한하는 라이선스 하에서 배포됩니다. 본 제품 또는 설명서의 어떠한 부분도 Sun 및 해당 사용권자의 사전 서면 승인 없이는 형식이나 수단에 상관없이 재생이 불가능합니다.

글꼴 기술을 포함한 타사 소프트웨어는 저작권이 등록되어 있으며 Sun 공급업체로부터 라이선스를 취득한 것입니다.

본 제품의 일부는 Berkeley BSD 시스템일 수 있으며 University of California로부터 라이선스를 취득했습니다. UNIX는 X/Open Company, Ltd.를 통해 독점 라이선스를 취득한 미국 및 기타 국가의 등록 상표입니다.

Sun, Sun Microsystems, Sun 로고, AnswerBook2, docs.sun.com, Solaris, Java, Solstice DiskSuite, SunPlex 및 Sun StorageTek는 미국 및 기타 국가에서 Sun Microsystems, Inc.의 상표 또는 등록 상표입니다.

모든 SPARC 상표는 라이선스 하에 사용되며 미국 및 기타 국가에서 SPARC International, Inc.의 상표 또는 등록 상표입니다. SPARC 상표가 부착된 제품은 Sun Microsystems, Inc.가 개발한 아키텍처를 기반으로 합니다.

Mozilla는 미국 및 기타 국가에서 Netscape Communications Corporation의 상표 또는 등록 상표입니다.

OPEN LOOK 및 Sun™ Graphical User Interface는 Sun Microsystems, Inc.가 해당 사용자 및 라이선스 소유자를 위해 개발했습니다. Sun은 컴퓨터 업계에서 시각적 또는 그래픽 사용자 인터페이스 개념을 연구하고 개발하는 데 있어 Xerox의 선구자적 업적을 인정합니다. Sun은 Xerox Graphical User Interface에 대한 Xerox의 비독점 라이선스를 보유하고 있으며 이 라이선스는 OPEN LOOK GUI를 구현하거나 그 외의 경우 Sun의 서면 라이선스 계약을 준수하는 Sun의 라이선스 소유자에게도 적용됩니다.

U.S. 정부 권한 - 상용. 정부 사용자는 Sun Microsystems, Inc. 표준 사용권 계약과 FAR의 해당 규정 및 추가 사항의 적용을 받습니다.

본 설명서는 "있는 그대로" 제공되며 상업성, 특정 목적에 대한 적합성 또는 비침해성에 대한 모든 묵시적 보증을 포함하여 모든 명시적 또는 묵시적 조건, 표현 및 보증에 대해 어떠한 책임도 지지 않습니다. 이러한 보증 부인은 법적으로 허용된 범위 내에서만 적용됩니다.



재활용  
가능



Adobe PostScript

# 목차

---

머리말 xxv

1. 파일 시스템 개요 1

파일 시스템 기능 1

볼륨 관리 2

페이지된 I/O 및 직접 I/O 지원 2

대용량 2

빠른 파일 시스템 복구 3

메타 데이터 저장소 3

vnode 인터페이스 4

공유 파일 시스템 지원 4

Linux 클라이언트 지원 5

추가 파일 시스템 기능 6

디자인의 기본 7

Inode 파일 및 파일의 특징 7

디스크 할당 단위 지정 8

DAU 설정 및 파일 시스템 구조 8

ms 및 ma Sun StorageTek SAM 파일 시스템 9

이중 및 단일 할당 체계 9

데이터 맞춤 11

데이터 디스크의 스트라이프 너비 11

파일 할당 방식	13
메타 데이터 할당	14
라운드 로빈 할당	14
스트라이프 할당	17
스트라이프 그룹	19
일치하지 않는 스트라이프 그룹	23
일치하지 않는 스트라이프 그룹의 예제	23
논리 장치 번호(LUN)당 할당 제어	27

## 2. 파일 시스템 구성 29

File System Manager 소프트웨어 사용	29
▼ 처음으로 File System Manager 호출	30
추가 관리자 및 사용자 계정 만들기	30
▼ 추가 계정 만들기	31
권한 수준 지정	31
다수 사용자를 위한 계정 만들기	32
▼ File System Manager 액세스를 위한 추가 서버 추가	33
세션 시간 초과 설정	34
mcf 파일의 기능	34
Equipment Identifier 필드	35
Equipment Ordinal 필드	36
Equipment Type 필드	36
Family Set 필드	37
Device State 필드	38
Additional Parameters 필드	38
mcf 파일의 예제	38
파일 설정, 옵션 및 명령 사이의 관계	40
파일 시스템 초기화	41

구성 예제	42
▼ Sun StorageTek QFS 라운드 로빈 디스크 구성 만들기	42
▼ Sun StorageTek QFS 스트라이프 디스크 구성 만들기	44
▼ Sun StorageTek QFS 스트라이프 그룹 구성 만들기	45
<b>3. 작업 수행</b>	<b>49</b>
파일 및 파일 속성 보기	49
파일 속성 및 파일 상태	49
파일 정보 표시	51
s1s(1) 출력 정보	51
보유 행 정보	53
시스템에 대한 구성 파일 변경 전파	53
▼ Sun StorageTek QFS Sun Cluster 환경에서 mcf 또는 defaults.conf 정보 변경	54
▼ SAM-QFS 환경에서 mcf 또는 defaults.conf 파일 시스템 정보 변경	55
▼ mcf 또는 defaults.conf 제거 가능한 매체 드라이브 정보 변경	56
공유 호스트 파일 변경	57
▼ 새 항목 추가 또는 기존 항목 변경	57
▼ 호스트 이름 변경, 항목 순서 변경 또는 항목 삽입	58
마운트 매개변수 설정	59
mount(1M) 명령	60
/etc/vfstab 파일	60
samfs.cmd 파일	61
파일 시스템 마운트 해제	62
▼ 독립형 QFS 또는 SAM-QFS 파일 시스템의 마운트 해제	62
▼ Sun StorageTek QFS 공유 파일 시스템 마운트 해제	63
파일 시스템에 디스크 캐시 추가	63
▼ 파일 시스템에 디스크 캐시 추가	63
파일 시스템 재작성	64
▼ 파일 시스템 백업 및 다시 만들기	65

<b>4. Sun StorageTek QFS 공유 파일 시스템 구성</b>	<b>67</b>
Sun StorageTek QFS 공유 파일 시스템 마운트 및 마운트 해제	67
▼ 공유 파일 시스템 마운트	68
▼ 공유 파일 시스템 마운트 해제	68
공유되지 않은 파일 시스템을 공유 파일 시스템으로 변환	69
▼ 메타 데이터 서버에 대한 변환 수행	69
▼ 각 클라이언트에서 변환 수행	71
공유 파일 시스템을 공유되지 않은 파일 시스템으로 변환	73
▼ 각 클라이언트에서 변환 수행	73
▼ 서버에서 변환 수행	74
클라이언트 호스트 추가 또는 제거	75
▼ 클라이언트 호스트 추가	76
▼ 클라이언트 호스트 제거	80
Sun StorageTek QFS 공유 환경에서 mcf 파일 업데이트	82
로컬 호스트 구성 파일 작성	85
Sun StorageTek QFS 환경에서 메타 데이터 서버 변경	88
▼ 메타 데이터 서버가 사용 가능할 때 메타 데이터 서버 변경	89
▼ 메타 데이터 서버가 사용 불가능할 때 메타 데이터 서버 변경	89
SAM-QFS 환경에서 메타 데이터 서버 변경	91
▼ SAM-QFS 환경에서 메타 데이터 서버 변경	91
Sun StorageTek QFS 공유 파일 시스템에서의 클라이언트-서버 통신	93
<b>5. 파일 시스템 할당량 관리</b>	<b>95</b>
개요	95
할당량 유형, 할당량 파일 및 할당량 레코드	96
소프트 제한 및 하드 제한	97
디스크 블록 및 파일 제한	97

## 할당량 활성화 98

할당량 설정을 위한 지침 98

▼ 새로운 파일 시스템이 할당량을 사용하도록 구성 99

▼ 기존 파일 시스템이 할당량을 사용하도록 구성 101

▼ 디렉토리 및 파일에 관리자 세트 ID 할당 103

무한 할당량 설정 104

▼ 무한 할당량 설정 104

기본 할당량 값 활성화 104

▼ 사용자, 그룹 또는 관리자 세트에 대한 기본 할당량 값 활성화 105

제한 활성화 106

▼ 사용자, 그룹 또는 관리자 세트에 대한 제한 활성화 106

▼ 기존 할당량 파일을 사용하는 사용자, 그룹 또는 관리자 세트에 대한 제한 활성화 또는 변경하기 106

## 할당량 확인 108

▼ 초과된 할당량 확인 108

## 할당량 변경 및 제거 110

▼ 유예 기간 변경 110

유예 기간 만료 변경 113

▼ 추가 파일 시스템 자원 할당 방지 114

▼ 파일 시스템의 할당량 제거 117

▼ 할당량 정정 118

## 6. Sun Cluster 환경의 Sun StorageTek QFS 구성 119

시작하기 전에 120

제한사항 121

Sun Cluster 시스템과 Sun StorageTek QFS 소프트웨어의 상호 작용 방법 122

공유 파일 시스템의 데이터 액세스 122

공유되지 않는 파일 시스템의 데이터 액세스 122

Sun Cluster용 Solaris Volume Manager의 Sun StorageTek QFS 지원	123
▼ Sun Cluster용 Solaris Volume Manager가 있는 파일 시스템 구성	124
구성 예제	129
Sun Cluster 환경에서의 Sun StorageTek QFS 공유 파일 시스템 구성	131
메타 데이터 서버 자원 고려사항	131
구성 예제	132
▼ Sun StorageTek QFS 공유 파일 시스템 작성 준비	133
▼ Sun StorageTek QFS 공유 파일 시스템 작성 및 Sun Cluster 노드 구성	136
▼ 구성 검증	137
▼ Oracle Real Application Clusters용 Sun Cluster 데이터 서비스 구성	139
Sun Cluster 환경에서 공유 해제된 파일 시스템 구성	140
예제 1: 원시 전역 장치의 HA-NFS	141
▼ 공유되지 않는 Sun StorageTek QFS 파일 시스템 작성 준비	142
▼ Sun StorageTek QFS 파일 시스템 작성 및 Sun Cluster 노드 구성	143
▼ 네트워크 이름 서비스 및 IPMP 검증 테스트 구성	145
▼ 고가용성을 위한 HA-NFS 및 Sun StorageTek QFS 파일 시스템 구성	148
예제 2: Solstice DiskSuite/Solaris Volume Manager가 제어하는 볼륨의 HA-NFS	149
▼ Solstice DiskSuite/Solaris Volume Manager 소프트웨어 준비	150
▼ Sun StorageTek QFS 파일 시스템 준비	151
▼ Sun StorageTek QFS 파일 시스템 작성 및 Sun Cluster 노드 구성	152
▼ 네트워크 이름 서비스 및 IPMP 검증 테스트 구성	154
▼ 고가용성을 위한 HA-NFS 및 Sun StorageTek QFS 파일 시스템 구성	154
예제 3: VxVM 볼륨의 HA-NFS	156
▼ VxVM 소프트웨어 구성	157
▼ Sun StorageTek QFS 파일 시스템 작성 준비	158

▼ Sun StorageTek QFS 파일 시스템 작성 및 Sun Cluster 노드 구성	159
▼ 구성 검증	160
▼ 네트워크 이름 서비스 및 IPMP 검증 테스트 구성	160
▼ 고가용성을 위한 HA-NFS 및 Sun StorageTek QFS 파일 시스템 구성	160
클러스터 외부에 공유 클라이언트 구성	161
구성 전제조건	161
Sun StorageTek QFS 메타 데이터 서버 Sun Cluster 노드	161
Sun StorageTek QFS 메타 데이터 클라이언트 노드	162
Sun Cluster 장치 구성	162
클러스터 외부의 클라이언트 구성을 위한 요구 사항	162
최소 소프트웨어 릴리스 레벨	163
지원되는 하드웨어 아키텍처	163
저장소 요구 사항	163
구성 지침	163
▼ /etc/hosts 파일 편집	163
▼ 메타 데이터 서버 네트워크 구성	164
▼ 메타 데이터 클라이언트 네트워크 구성	165
▼ Sun Cluster 설치 및 구성	165
▼ Sun StorageTek QFS 메타 데이터 서버 구성	166
▼ Sun StorageTek QFS 메타 데이터 클라이언트 구성	170
Sun StorageTek QFS 구성 변경	172
▼ 공유 파일 시스템 구성 변경	173
▼ 원시 전역 장치를 사용하는 파일 시스템에서 HA-NFS 비활성화	174
▼ Solaris Volume Manager 제어 볼륨을 사용하는 파일 시스템에서 HA-NFS 비활성화	175
▼ VxVM 제어 볼륨을 사용하는 Sun StorageTek QFS 파일 시스템에서 HA-NFS 비활성화	176
Sun Cluster를 사용한 고가용성 Sun StorageTek SAM 구성	178

7. 고급 항목	179
데몬, 프로세스 및 추적 사용	179
데몬 및 프로세스	180
추적 파일	181
추적 파일 내용	181
추적 파일 교환	182
추적되고 있는 프로세스 확인	182
setfa(1) 명령을 사용하여 파일 속성 설정	184
파일 및 디렉토리에 대한 파일 속성 선택	184
파일 공간 사전 할당	184
파일 할당 방식 및 스트라이프 너비 선택	185
스트라이프 그룹 장치 선택	186
WORM-FS 파일 시스템 구성	186
WORM-FS 기능 활성화	187
WORM "Lite" 옵션	188
WORM 파일 생성	189
예제 1: WORM 트리거 chmod 4000	190
예제 2: WORM 트리거 chmod 4000	191
예제 3: WORM 트리거 chmod -w	192
예제 4: WORM 트리거 chmod -w	193
보유 기간	194
기본 보유 기간 설정	196
touch를 사용하여 보유 기간 설정	197
파일 보유 기간 연장	199
sfs를 사용한 WORM-FS 파일 보기	199
sfind를 사용한 WORM-FS 파일 찾기	200
용량이 큰 파일 작업	201
다중판독기 파일 시스템 구성	202

이기종 컴퓨팅 환경에서 SAN-QFS 파일 시스템 사용	204
시작하기 전에	206
SAN-QFS 파일 시스템 활성화	206
▼ 메타 데이터 제어기에서 SAN-QFS 파일 시스템 활성화	206
▼ 클라이언트에서 SAN-QFS 파일 시스템 활성화	207
▼ 클라이언트에 SANergy 소프트웨어 설치	208
SAN-QFS 파일 시스템 마운트 해제	209
▼ SANergy 클라이언트에서 SAN-QFS 파일 시스템 마운트 해제	209
▼ 메타 데이터 제어기에서 SAN-QFS 파일 시스템 마운트 해제	210
▼ Sun StorageTek QFS 클라이언트에서 SAN-QFS 파일 시스템 마운트 해제	210
▼ Sun StorageTek QFS 서버에서 SAN-QFS 파일 시스템 마운트 해제	210
문제 해결: SANergy 파일 보류를 갖는 SAN-QFS 파일 시스템 마운트 해제	211
▼ SANergy 파일 보류가 존재하는 경우 파일 시스템 마운트 해제	211
SAN-QFS 파일 시스템의 블록 할당량	211
SAN-QFS 파일 시스템의 파일 데이터 및 파일 속성	211
samgrowfs(1M)을 사용한 SAN-QFS 파일 시스템 확장	212
SAN-QFS 공유 파일 시스템 및 Sun StorageTek QFS 공유 파일 시스템 비교	212
I/O 유형 이해	213
페이지된 I/O	213
직접 I/O	213
I/O 전환	214
대형 파일에 대한 파일 전송 성능 향상	214
▼ 파일의 전송 성능 향상	214
Qwrite 기능 활성화	217
쓰기 스로틀 설정	218
Flush-Behind 속도 설정	219

inode 수 및 inode 해시 테이블 조정 219

ninodes 매개변수 220

nhino 매개변수 220

ninodes 및 nhino 매개변수 설정 시기 221

## A. 문제 해결 **Sun StorageTek QFS** 223

파일 시스템 무결성 확인 및 파일 시스템 복구 223

▼ 파일 시스템 확인 224

▼ 파일 시스템 복구 225

공유 파일 시스템에서 실패하거나 멈춘 `sammkfs(1M)` 또는 `mount(1M)` 명령 문제 해결 225

실패한 `sammkfs(1M)` 명령 복구 226

▼ `mcf` 파일 확인 및 `mcf` 파일 변경을 시스템으로 전파 226

실패한 `mount(1M)` 명령 복구 226

▼ 파일 시스템의 마운트 가능 여부 확인 227

▼ `samfsinfo(1M)` 및 `samsharefs(1M)` 명령 사용 229

▼ `samfsconfig(1M)` 명령 사용 230

멈춘 `mount(1M)` 명령 복구 232

▼ 네트워크 연결 확인 232

▼ 클라이언트가 서버에 연결할 수 있는지 확인 234

▼ 서버가 클라이언트에 연결할 수 있는지 확인 237

▼ `sam-sharefsd` 추적 로그 확인 238

Linux 클라이언트 문제 해결 240

문제 해결 도구 241

FAQ 242

## B. **Sun StorageTek QFS** 공유 파일 시스템의 마운트 옵션 245

백그라운드에서 마운트: `bg` 옵션 246

파일 시스템 마운트 재시도: `retry` 옵션 246

Sun StorageTek QFS 공유 파일 시스템 선언: `shared` 옵션 246

할당 크기 조정: `minallocsz=n` 및 `maxallocsz=n` 옵션 247

Sun StorageTek QFS 공유 파일 시스템에서 임대 사용: `rdlease=n`, `wrlease=n` 및 `aplease=n` 옵션 247

다중 호스트 읽기 및 쓰기 활성화: `mh_write` 옵션 248

최소 동시 스레드 수 설정: `min_pool=n` 옵션 250

캐시된 속성 유지: `meta_timeo=n` 옵션 250

스트라이프 할당 지정: `stripe` 옵션 251

메타 데이터가 기록되는 빈도 지정: `sync_meta=n` 옵션 251

WORM 기능 활성화: `worm_capable` 및 `def_retention` 옵션 252

### C. samu(1M) 운영자 유틸리티 사용 253

개요 253

- ▼ samu(1M)를 호출하려면 254

- ▼ samu(1M) 화면 표시 254

- ▼ samu(1M) 중지 255

samu(1M)와 상호 작용 255

장치 입력 255

온라인 도움말 가져오기 256

- ▼ 디스플레이 화면에서 온라인 도움말 액세스하기 256

운영자 디스플레이 256

(a) - 아카이버 상태 디스플레이 257

탐색 258

예제 디스플레이 258

필드 설명 259

(c) - 장치 구성 디스플레이 260

탐색 260

예제 디스플레이 260

필드 설명 261

- (C) - 메모리 디스플레이 261
  - 예제 디스플레이 262
- (d) - 데몬 추적 컨트롤 디스플레이 262
  - 예제 디스플레이 263
- (D) - 디스크 볼륨 사전 264
  - 예제 디스플레이 264
  - 플래그 264
- (E) - 파일 시스템 디스플레이 265
  - 예제 디스플레이 265
  - 필드 설명 266
- (F) - 광 디스크 레이블 디스플레이 267
- (h) - 도움말 디스플레이 267
  - 탐색 267
  - 예제 디스플레이 268
- (I) - Inode 디스플레이 269
  - 탐색 269
  - 예제 디스플레이 270
- (J) - 미리보기 공유 메모리 디스플레이 270
  - 탐색 271
  - 예제 디스플레이 271
- (K) - 커널 통계 디스플레이 271
  - 탐색 272
  - 예제 디스플레이 272
- (l) - 사용량 디스플레이 272
  - 예제 디스플레이 273
- (L) - 공유 메모리 표 273
  - 예제 디스플레이 274

- (m) - 대량 저장 상태 디스플레이 274
  - 예제 디스플레이 275
  - 필드 설명 276
- (M) - 공유 메모리 디스플레이 276
  - 탐색 277
  - 예제 디스플레이 277
- (n) - 스테이지 상태 디스플레이 278
  - 예제 디스플레이 278
- (N) - 파일 시스템 매개변수 디스플레이 279
  - 탐색 279
  - 예제 디스플레이 280
- (o) - 광 디스크 상태 디스플레이 281
  - 탐색 281
  - 예제 디스플레이 281
  - 필드 설명 282
- (p) - 제거 가능한 매체 로드 요청 디스플레이 282
  - 탐색 283
  - 예제 디스플레이 283
  - 필드 설명 284
  - 플래그 284
- (P) - 활성 서비스 디스플레이 285
  - 탐색 285
  - 예제 디스플레이 285
- (r) - 제거 가능한 매체 상태 디스플레이 286
  - 예제 디스플레이 286
  - 필드 설명 286
- (R) - Sun SAM-Remote 정보 디스플레이 287

- (s) - 장치 상태 디스플레이 287
  - 탐색 288
  - 예제 디스플레이 288
  - 필드 설명 289
- (S) - 섹터 데이터 디스플레이 289
  - 탐색 289
- (t) - 테이프 드라이브 상태 디스플레이 290
  - 탐색 290
  - 예제 디스플레이 290
  - 필드 설명 291
- (T) - SCSI 감지 데이터 디스플레이 292
  - 탐색 292
- (u) - 스테이지 대기열 디스플레이 292
  - 탐색 293
  - 예제 디스플레이 293
  - 필드 설명 294
- (U) - 장치 테이블 디스플레이 294
  - 탐색 295
  - 예제 디스플레이 295
- (v) - 자동화된 라이브러리 카탈로그 디스플레이 296
  - 탐색 297
  - 예제 디스플레이 297
  - 필드 설명 298
  - 플래그 299
- (w) - 보류 스테이지 대기열 299
  - 탐색 300
  - 예제 디스플레이 300
  - 필드 설명 301

운영자 디스플레이 상태 코드	301
제거 가능한 매체 장치 디스플레이 상태 코드	301
파일 시스템 디스플레이 상태 코드	302
운영자 디스플레이 장치 상태	303
▼ down에서 on으로 드라이브 상태 변경하기	304
▼ on에서 down으로 드라이브 상태 변경하기	305
운영자 명령	306
장치 명령	306
파일 시스템 명령: I/O 관리	307
:flush_behind <i>eq value</i> 명령	307
:force_nfs_async <i>eq</i> 및 :noforce_nfs_async <i>eq</i> 명령	307
:readahead <i>eq contig</i> 명령	308
:sw_raid <i>eq</i> 및 :nosw_raid <i>eq</i> 명령	308
:writebehind <i>eq contig</i> 명령	309
:wr_throttle <i>eq value</i> 명령	309
파일 시스템 명령: 직접 I/O 관리	309
:dio_rd_form_min <i>eq value</i> 및 :dio_wr_form_min <i>eq value</i> 명령	310
:dio_rd_ill_min <i>eq value</i> 및 :dio_wr_ill_min <i>eq value</i> 명령	310
:dio_rd_consec <i>eq value</i> 및 :dio_wr_consec <i>eq value</i> 명령	310
:dio_szero <i>eq</i> 및 :nodio_szero <i>eq</i> 명령	311
:forcedirectio <i>eq</i> 및 :noforcedirectio <i>eq</i> 명령	311
파일 시스템 명령: Sun StorageTek QFS 공유 파일 시스템	311
:meta_timeo <i>eq interval</i> 명령	311
:mh_write <i>eq</i> 및 :nomh_write <i>eq</i> 명령	312
:minallocsz <i>eq value</i> 및 :maxallocsz <i>eq value</i> 명령	312
:rdlease <i>eq interval</i> , :wrlease <i>eq interval</i> 및 :aplease <i>eq interval</i> 명령	312

파일 시스템 명령: 기타 312

- :abr *eq* 및 :noabr *eq* 명령 312
- :dmr *eq* 및 :nodmr *eq* 명령 313
- :invalid *eq interval* 명령 313
- :mm\_stripe *eq value* 명령 313
- :qwrite *eq* 및 :noqwrite *eq* 명령 314
- :refresh\_at\_eof *eq* 및 :norefresh\_at\_eof *eq* 명령 314
- :suid *eq* 및 :nosuid *eq* 명령 314
- :stripe *eq value* 명령 315
- :sync\_meta *eq value* 명령 315
- :trace *eq* 및 :notrace *eq* 명령 316

기타 명령 316

- :clear *vsname [ index ]* 명령 316
- :devlog *eq [ option ]* 명령 316
- :diskvols *volume [+flag | -flag]* 명령 316
- :dtrace 명령 317
- :fs *filename* 명령 317
- :mount *mntpt* 명령 317
- :open *eq* 명령 318
- :read *addr* 명령 318
- :refresh *i* 명령 318
- :snap [*filename*] 명령 318
- :! *shell\_command* 명령 318

용어 해설 319

색인 329

# 그림

---

그림 1-1	5개 장치를 사용한 ms 파일 시스템의 라운드 로빈 할당	15
그림 1-2	5개 장치를 사용한 ma 파일 시스템의 라운드 로빈 할당	16
그림 1-3	5개 장치를 사용하는 ms 파일 시스템에서의 스트라이프	18
그림 1-4	5개 장치를 사용하는 ma 파일 시스템에서의 스트라이핑	19
그림 1-5	Sun StorageTek QFS 라운드 로빈 스트라이프 그룹	20
그림 1-6	Sun StorageTek QFS 스트라이프 그룹 할당	22
그림 1-7	스트라이프 할당에서 일치하지 않는 스트라이프 그룹을 사용하는 Sun StorageTek QFS 파일 시스템	24
그림 4-1	네트워크 인터페이스	87
그림 7-1	Sun StorageTek QFS 소프트웨어 및 SANergy 소프트웨어를 사용하는 SAN-QFS 파일 시스템	205



# 표

---

표 1-1	제품 개요	1
표 1-2	.inode 파일의 내용	7
표 1-3	Sun StorageTek QFS 장비 유형 값 및 DAU 크기	10
표 1-4	ms 파일 시스템 기본 스트라이프 너비	11
표 1-5	기본 스트라이프 너비	12
표 1-6	기본 할당 방식	13
표 1-7	메타 데이터 할당	14
표 1-8	예제 파일 시스템 특징	25
표 2-1	File System Manager 권한 수준	31
표 2-2	Sun StorageTek QFS 또는 SAM-QFS Equipment Type 필드	36
표 3-1	사용자 파일 속성	50
표 3-2	sls(1) 수행 결과 설명	51
표 4-1	로컬 호스트 구성 파일 필드	85
표 5-1	할당량 파일 이름	96
표 7-1	데몬 및 프로세스	180
표 7-2	파일 할당 및 스트라이프 너비	185
표 7-3	SAN-QFS 공유 파일 시스템 및 Sun StorageTek QFS 공유 파일 시스템	212
표 A-1	오류 표시기	223
표 A-2	/proc 파일	241
표 B-1	임대 관련 mount(1M) 옵션	247

표 B-2	mh_write 옵션에 따른 파일 액세스	249
표 C-1	a 디스플레이에 대한 컨트롤 키	258
표 C-2	:a <i>filesystem</i> 디스플레이에 대한 컨트롤 키	258
표 C-3	samu(1M) a 디스플레이 필드 설명	259
표 C-4	c 디스플레이에 대한 컨트롤 키	260
표 C-5	samu(1M) c 디스플레이 필드 설명	261
표 C-6	samu(1M) D 디스플레이의 플래그 필드	264
표 C-7	samu(1M) f 디스플레이 필드 설명	266
표 C-8	h 디스플레이에 대한 컨트롤 키	267
표 C-9	I 디스플레이에 대한 컨트롤 키	269
표 C-10	J 디스플레이에 대한 컨트롤 키	271
표 C-11	K 디스플레이에 대한 컨트롤 키	272
표 C-12	samu(1M) m 디스플레이 필드 설명	276
표 C-13	M 디스플레이에 대한 컨트롤 키	277
표 C-14	N 디스플레이에 대한 컨트롤 키	279
표 C-15	o 디스플레이에 대한 컨트롤 키	281
표 C-16	samu(1M) o 디스플레이 필드 설명	282
표 C-17	p 디스플레이에 대한 컨트롤 키	283
표 C-18	samu(1M) p 디스플레이 필드 설명	284
표 C-19	samu(1M) P 디스플레이의 플래그 필드	284
표 C-20	P 디스플레이에 대한 컨트롤 키	285
표 C-21	samu(1M) r 디스플레이 필드 설명	286
표 C-22	s 디스플레이에 대한 컨트롤 키	288
표 C-23	samu(1M) s 디스플레이 필드 설명	289
표 C-24	S 디스플레이에 대한 컨트롤 키	289
표 C-25	t 디스플레이에 대한 컨트롤 키	290
표 C-26	samu(1M) t 디스플레이 필드 설명	291
표 C-27	T 디스플레이에 대한 컨트롤 키	292
표 C-28	u 디스플레이에 대한 컨트롤 키	293
표 C-29	samu(1M) u 디스플레이 필드 설명	294

표 C-30	U 디스플레이에 대한 컨트롤 키	295
표 C-31	V 디스플레이에 대한 컨트롤 키	297
표 C-32	samu(1M) V 디스플레이 필드 설명	298
표 C-33	samu(1M) V 디스플레이의 플래그 필드	299
표 C-34	w 디스플레이에 대한 컨트롤 키	300
표 C-35	samu(1M) w 디스플레이 필드 설명	301
표 C-36	제거 가능한 매체 장치 디스플레이 상태 코드	302
표 C-37	파일 시스템 디스플레이 상태 코드	303
표 C-38	운영자 디스플레이 장치 상태	303
표 C-39	장치 명령 작업	306
표 C-40	추적 명령 인수	317



# 머리말

---

본 설명서 Sun StorageTek™ QFS 파일 시스템 구성 및 관리 설명서는 Sun StorageTek QFS 버전 4, 업데이트 6(4U6)에 포함되어 있는 파일 시스템 소프트웨어에 대해 설명합니다.

Sun StorageTek QFS 제품에는 독립형 파일 시스템이나 공유 파일 시스템으로 사용할 수 있는 파일 시스템이 있으며, 이는 또한 Sun StorageTek SAM 제품에 포함된 저장소 및 아카이브 관리자와 병행하여 사용할 수 있습니다. Sun StorageTek SAM 소프트웨어와 병행하여 사용하는 경우 SAM-QFS라고 합니다.

Sun StorageTek QFS 소프트웨어 패키지는 다음 운영 체제(OS) 환경에서 실행됩니다.

- SPARC® 플랫폼용 Solaris™ 9 운영 체제(OS) 04/03 이후 버전
- SPARC 또는 x64 플랫폼용 Solaris 10 OS
- x86(32비트)용 Solaris 10 OS(Sun StorageTek QFS 공유 클라이언트 전용)
- x86/x64 플랫폼용 Red Hat Enterprise Linux 3.0(UD-5 및 UD8)(Sun StorageTek QFS 공유 클라이언트 전용)
- x64 플랫폼용 Red Hat Enterprise Linux 4.0(UD-2 및 UD-4)(Sun StorageTek QFS 공유 클라이언트 전용)
- x64 플랫폼용 SuSE Linux Enterprise Server 8(서비스 팩 4)(Sun StorageTek QFS 공유 클라이언트 전용)
- x64 플랫폼 및 SGI Altix Itanium 시스템용 SuSE Linux Enterprise Server 9(서비스 팩 2)(Sun StorageTek QFS 공유 클라이언트 전용)
- x64 플랫폼용 SuSE Linux Enterprise Server 10(Sun StorageTek QFS 공유 클라이언트 전용)

---

주 - 이것은 SuSE Enterprise Linux 8을 지원하는 Sun StorageTek QFS의 최종 버전입니다. 이 소프트웨어의 다음 버전은 지원되지 않습니다.

---

이 설명서는 Sun StorageTek QFS 파일 시스템의 설치, 구성 및 유지 관리에 대한 책임이 있는 시스템 관리자용입니다. 이 설명서를 읽는 사용자인 시스템 관리자가 이미 설치, 구성, 계정 작성, 시스템 백업 수행, 기타 기본적인 Solaris OS 시스템 관리 업무를 포함한 Solaris OS 절차에 해박한 지식을 보유한 상태임을 가정합니다.

---

## 본 설명서의 구성

본 설명서는 다음 장으로 구성되어 있습니다.

- **1장**에서는 개요 정보를 제공합니다.
- **2장**에서는 파일 시스템 구성 정보를 제공합니다.
- **3장**에서는 파일 시스템 초기화, 서버 추가, 디스크 캐시 추가 및 기타 시스템 관리 작업 같은 다양한 작업을 수행하는 방법에 대해 설명합니다.
- **4장**에서는 Sun StorageTek QFS 공유 파일 시스템을 구성하는 방법에 대해 설명합니다.
- **5장**에서는 파일 시스템 할당량 사용 방법에 대해 설명합니다.
- **6장**에서는 Sun<sup>TM</sup> Cluster 환경에서 Sun StorageTek QFS 소프트웨어를 구성하는 방법에 대해 설명합니다.
- **7장**에서는 다중 판독기 파일 시스템 사용 및 성능 사양과 같은 기타 고급 항목에 대해 설명합니다.
- **부록 A**에는 Sun StorageTek QFS 소프트웨어의 문제 해결에 대한 정보를 포함하고 있습니다.
- **부록 B**에는 Sun StorageTek QFS 공유 파일 시스템에 대한 마운트 옵션 목록이 있습니다.
- **부록 C**에서는 samu(1M) 운영자 유틸리티 사용 방법에 대해 설명합니다.

---

## UNIX 명령 사용

이 설명서에는 시스템 종료, 시스템 부팅 및 장치 구성과 같은 기본적인 UNIX<sup>®</sup> 명령어 및 절차에 대한 정보는 포함되어 있지 않을 수 있습니다. 이러한 정보에 대해서는 다음을 참조하여 주십시오.

- 시스템에 포함되어 있는 소프트웨어 설명서
- Solaris<sup>TM</sup> 운영 체제 설명서는 다음 URL을 참조하여 주시기 바랍니다.

<http://docs.sun.com>

---

## 셸 프롬프트

표 P-1은 본 설명서에서 사용하는 셸 프롬프트를 표시합니다.

표 P-1 셸 프롬프트

셸	프롬프트
C 셸	<i>machine-name%</i>
C 셸 슈퍼유저	<i>machine-name#</i>
Bourne 셸 및 Korn 셸	\$
Bourne 셸 및 Korn 셸 슈퍼유저	#

---

## 활자체 규약

표 P-2는 본 설명서에서 사용된 활자체 규약을 나열한 것입니다.

표 P-2 활자체 규약

활자체 또는 기호	의미	예
AaBbCc123	명령 및 파일, 디렉토리 이름; 컴퓨터 화면에 출력되는 내용입니다.	.login 파일을 편집하십시오. 모든 파일 목록을 보려면 <code>ls -a</code> 명령을 사용하십시오. % You have mail.
<b>AaBbCc123</b>	사용자가 입력하는 내용으로 컴퓨터 화면의 출력 내용과 대조됩니다.	% <b>su</b> Password:
AaBbCc123	설명서 제목; 새로운 단어 또는 용어; 강조할 단어; 실제 이름 또는 값으로 대체될 명령줄 변수	<i>class</i> 옵션입니다. 이를 실행하기 위해서는 반드시 슈퍼 유저여야 합니다. 파일 삭제 명령은 <b>rm filename</b> 입니다.
AaBbCc123	책 제목, 장, 절	Solaris 사용자 설명서 6장 데이터 관리를 참조하시기 바랍니다.

표 P-2 활자체 규약(계속)

활자체 또는 기호	의미	예
[ ]	구문에서 대괄호는 인수가 옵션임을 나타냅니다.	scmadm [-d sec] [-r n[:n][,n]...] [-z]
{ arg   arg }	구문에서 중괄호와 파이프 기호 ( )는 인수들 중 하나가 지정되어야 함을 나타냅니다.	sndradm -b { phost   shost }
\	명령줄 끝의 백슬래시(\)는 명령이 다음 행에서 계속됨을 나타냅니다.	atm90 /dev/md/rdsk/d5 \ /dev/md/rdsk/d1 atm89

## 관련 설명서

본 설명서는 Sun StorageTek QFS 및 Sun StorageTek SAM 소프트웨어 제품의 운영을 기술한 설명서 세트의 일부입니다. 표 P-3은 이 제품들에 대한 전체 릴리스 4U6 설명서 세트를 표시합니다.

표 P-3 관련 Sun StorageTek 설명서

제목	부품 번호
Sun StorageTek QFS 설치 및 업그레이드 안내서	820-1722-10
Sun StorageTek Storage Archive Manager 파일 시스템 구성 및 관리 안내서	820-1742-10
Sun StorageTek Storage Archive Manager 아카이브 구성 및 관리 안내서	820-1727-10
Sun StorageTek Storage Archive Manager 설치 및 업그레이드 안내서	820-1732-10
Sun StorageTek Storage Archive Manager 문제 해결 안내서	820-1737-10
Sun StorageTek QFS 및 Sun StorageTek Storage Archive Manager 4.6 릴리스 노트	820-1747-10
Sun StorageTek QFS Linux 클라이언트 안내서	820-1752-10

---

# Sun 설명서를 온라인으로 이용하시려면

Sun StorageTek QFS 소프트웨어 배포에는 Sun의 Network Storage 설명서 웹 사이트 또는 docs.sun.com에서 볼 수 있는 PDF 파일도 포함됩니다.

## docs.sun.com에서 설명서 액세스

이 웹 사이트에는 Solaris 및 기타 여러 가지 Sun 소프트웨어 제품에 대한 설명서가 있습니다.

1. 다음 URL로 이동합니다.

<http://docs.sun.com>

docs.sun.com 페이지가 나타납니다.

2. 검색 상자에서 **Sun StorageTek QFS**를 검색하여 해당 제품 설명서를 찾으십시오.

## Sun의 Network Storage 설명서 웹 사이트에서 설명서 액세스

이 웹 사이트에는 Network Storage에 대한 설명서가 있습니다.

1. 다음 URL로 이동합니다.

[http://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 페이지가 나타납니다.

2. **Sun StorageTek QFS Software** 링크를 누릅니다.

---

## 타사 웹 사이트

Sun은 본 설명서에서 언급된 타사 웹 사이트의 가용성 여부에 대해 책임을 지지 않습니다. 또한 해당 사이트나 리소스를 통해 제공되는 내용, 광고, 제품 및 기타 자료에 대해 어떠한 보증도 하지 않으며 그에 대한 책임도 지지 않습니다. 따라서 타사 웹 사이트의 내용, 제품 또는 리소스의 사용으로 인해 발생한 실제 또는 주장된 손상이나 피해에 대해서도 책임을 지지 않습니다.

---

## Sun 기술 지원

본 제품과 관련하여 설명서에 나와 있지 않은 기술 문제가 발생할 경우 다음 URL을 참조하십시오.

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

---

## 라이센싱

Sun StorageTek QFS 및 Sun StorageTek SAM 소프트웨어에 대한 라이선스를 구할 수 있는 정보는 Sun 영업 담당자나 공인 서비스 제공자(ASP)에게 문의하십시오.

---

## 설치 지원

설치 및 구성 서비스를 받으려면, 1-800-USA4SUN으로 전화하여 Sun Enterprise Services에 문의하거나 해당 지역 Enterprise Services 영업 담당자에게 문의하십시오.

---

## Sun은 여러분의 의견을 환영합니다.

Sun은 설명서의 내용을 개선하기 위해 노력을 기울이고 있으며 언제나 여러분의 의견과 제안을 환영합니다. 다음 사이트에 여러분의 의견을 제출하여 주십시오.

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

아래와 같이 설명서의 제목과 부품 번호를 함께 적어 보내주시기 바랍니다 (Sun StorageTek QFS 파일 시스템 구성 및 관리 안내서, 부품 번호 820-1717-10).

## 파일 시스템 개요

본 장에는 다음 절이 포함되어 있습니다.

- 1페이지의 "파일 시스템 기능"
- 7페이지의 "디자인의 기본"
- 13페이지의 "파일 할당 방식"

## 파일 시스템 기능

Sun StorageTek QFS 파일 시스템은 사용자에게 표준 UNIX 파일 시스템(UFS) 인터페이스를 제공하는 구성 가능한 파일 시스템입니다. 표 1-1에서는 파일 시스템과 저장소 및 아카이브 관리(Sun StorageTek SAM) 소프트웨어가 어떻게 사용 또는 결합될 수 있는지 보여줍니다.

표 1-1      제품 개요

제품	구성 요소
Sun StorageTek QFS 파일 시스템	독립형 파일 시스템
Sun StorageTek QFS 공유 파일 시스템	다중 호스트 시스템에 마운트할 수 있는 분산 파일 시스템
SAM 파일 시스템	Sun StorageTek SAM 소프트웨어에 포함되어 있는 파일 시스템. 이 파일 시스템은 Sun StorageTek QFS 파일 시스템에 있는 일부 기능을 포함하지 않습니다.
SAM-QFS	Sun StorageTek QFS 및 Sun StorageTek SAM 소프트웨어가 함께 사용될 경우 Sun StorageTek QFS 제품의 고급 파일 시스템 기능뿐만 아니라 Sun StorageTek SAM 제품의 저장소 관리 기능도 이용할 수 있습니다. 이 조합을 SAM-QFS라고 부릅니다. 주: 달리 언급되지 않는 한, 본 설명서 전반에 걸쳐 언급되는 Sun StorageTek QFS 파일 시스템 정보는 SAM-QFS 구성에도 적용됩니다.

Sun StorageTek QFS 파일 시스템은 사용자 프로그램이나 UNIX 커널에 대한 변경이 필요하지 않습니다. Sun StorageTek QFS 파일 시스템의 일부 기능이 다음 절에 설명되어 있습니다.

## 볼륨 관리

Sun StorageTek QFS 파일 시스템은 스트라이프 및 라운드 로빈 디스크 액세스 모두를 지원합니다. 마스터 구성 파일(mcf) 및 마운트 매개 변수는 볼륨 관리 기능을 지정하고 파일 시스템이 제어하는 장치 사이의 관계를 알 수 있도록 해줍니다. 이것은 하나의 장치 또는 장치의 일부만 지정할 수 있는 대개의 UNIX 파일 시스템과 대조적입니다. Sun StorageTek QFS 파일 시스템에는 추가 볼륨 관리자 응용 프로그램이 필요하지 않습니다. 그러나 Sun StorageTek QFS 환경의 장치에 대해 미러링을 사용하려는 경우 논리 볼륨 관리자와 같은 추가 패키지를 구해야 합니다.

Sun StorageTek QFS 통합 볼륨 관리 기능은 표준 Solaris OS 장치 드라이버 인터페이스를 사용하여 I/O 요청을 기본 장치로(부터) 전달합니다. Sun StorageTek QFS 소프트웨어는 각 파일 시스템이 상주하는 패밀리 세트로 저장 장치를 그룹화합니다.

## 페이지된 I/O 및 직접 I/O 지원

Sun StorageTek QFS 파일 시스템은 두 가지 유형의 I/O를 지원하는데, 페이지된(또한 캐시된 또는 버퍼된 I/O라고도 함) 및 직접 I/O입니다. 이들 I/O 유형은 다음과 같이 수행합니다.

- 페이지된 I/O가 사용될 경우 사용자 데이터는 가상 메모리 페이지에 캐시되며 커널은 데이터를 디스크에 기록합니다. 표준 Solaris OS 인터페이스가 페이지된 I/O를 관리합니다. 이는 Sun StorageTek QFS에 대한 기본 I/O 유형입니다.
- 직접 I/O가 사용되면 사용자 데이터는 사용자 메모리에서 디스크로 직접 쓰여집니다. Solaris OS `directio(3C)` 기능 호출 또는 `-D` 옵션이 있는 `setfa(1)` 명령을 사용하여 직접 I/O를 지정할 수 있습니다. 직접 I/O를 사용하면 대형 블록, 순차적으로 정렬된 I/O의 경우 성능을 크게 향상시킬 수 있습니다.

## 대용량

Sun StorageTek QFS 소프트웨어는 최대 2<sup>63</sup>바이트 길이의 파일을 지원합니다. 이와 같이 큰 용량의 파일은 단일 파일 시스템 내에서도 여러 디스크 또는 RAID 장치에 걸쳐 스트라이프할 수 있습니다. 실제 64비트 파일 시스템이 아닌 표준 UNIX 파일 시스템과는 대조적으로 Sun StorageTek QFS 파일 시스템이 실제 64비트 주소지정을 사용하기 때문에 가능합니다.

구성할 수 있는 파일 시스템의 수는 거의 제한이 없습니다. 볼륨 관리자는 각 파일 시스템이 최대 252개의 장치 분할 영역(일반적으로 디스크)을 포함하도록 합니다. 각 분할 영역은 최대 16테라바이트의 데이터를 포함할 수 있습니다. 이 구성은 거의 무제한의 저장 용량을 제공합니다.

Sun StorageTek QFS 파일 시스템의 파일 수에 대해 미리 정의된 제한이 없습니다. inode 공간(파일에 대한 정보 보유)은 동적으로 할당되기 때문에 파일의 최대 수는 사용 가능한 디스크의 저장 용량에 의해서만 제한됩니다. inode는 마운트 지점에 있는 .inodes 파일에 분류되어 있습니다. .inodes 파일은 파일당 512바이트의 저장 공간이 필요합니다.

Sun StorageTek QFS 파일 시스템의 경우 inode는 메타 데이터 장치에 있으며 파일 데이터 장치와 구분됩니다. 실제로 메타 데이터(mm) 장치의 크기는 Sun StorageTek QFS 파일 시스템의 파일 수를 제한하지만 메타 데이터 장치를 추가하여 최대 파일 수를 증가시킬 수 있습니다. 파일 수에 대한 하드 제한은  $2^{32}-1$  파일이며 추천 제한은  $10^7$  파일입니다.

## 빠른 파일 시스템 복구

파일 시스템의 핵심 기능은 갑작스런 정전 후 빠르게 복구하는 능력입니다. 표준 UNIX 파일 시스템의 경우, 시스템 장애 후 일관성 오류를 복구하려면 시간이 오래 소요되는 파일 시스템 검사(fsck(1M))가 필요합니다.

Sun StorageTek QFS 파일 시스템은 대개 파일 시스템이 디스크에 기록되지 않게 하는 장애 발생 후 파일 시스템 점검이 필요하지 않습니다(sync(1M) 사용). 또한 Sun StorageTek QFS 파일 시스템은 파일 오류 상태에서 저널링 없이 복구됩니다. 파일 시스템은 모든 중요한 I/O 작업에 대해 ID 기록, 직렬 쓰기 및 오류 점검을 사용하여 이 작업을 동적으로 수행합니다. 시스템 장애가 발생한 후에도 멀티테라바이트급 크기의 Sun StorageTek QFS 파일 시스템을 즉시 다시 마운트할 수 있습니다.

## 메타 데이터 저장소

파일 시스템은 메타 데이터를 사용하여 파일 및 디렉토리 정보를 참조합니다. 일반적으로 메타 데이터는 파일 데이터와 동일한 장치에 상주합니다. Sun StorageTek QFS 파일 시스템에는 파일 시스템 메타 데이터를 별도의 장치에 저장하여 파일 데이터로부터 분리하는 옵션이 있습니다. Sun StorageTek QFS 파일 시스템에서는 장치 헤드 이동 및 회전 지연 시간 감소, RAID 캐시 사용률 향상 또는 파일 데이터 미러링 없이 메타 데이터를 미러링하는 데 하나 이상의 별도 메타 데이터 장치를 정의할 수 있습니다.

Sun StorageTek QFS 파일 시스템은 별도의 파일에 inode 메타 데이터 정보를 저장합니다. 이렇게 하여 파일의 수 및 파일 시스템 전체가 동적으로 커지도록 합니다.

## vnode 인터페이스

Sun StorageTek QFS 파일 시스템은 표준 Solaris OS 가상 파일 시스템(vfs/vnode) 인터페이스를 통해 구현됩니다.

vfs/vnode 인터페이스를 사용함으로써 파일 시스템은 표준 Solaris OS 커널과 함께 작동하며 파일 관리 지원을 위해 커널에 대한 수정을 필요로 하지 않습니다. 따라서 파일 시스템은 운영 체제 변경으로부터 보호되고 운영 체제가 업데이트될 때 대개 집중적인 회귀 테스트가 필요하지 않습니다.

커널은 Sun StorageTek QFS 파일 시스템에 상주하는 요청을 포함하여 파일에 대한 모든 요청을 차단합니다. 해당 파일이 Sun StorageTek QFS 파일로 식별된 경우, 커널은 요청을 적절한 파일 시스템에 통과시켜 처리합니다. Sun StorageTek QFS 파일 시스템은 /etc/vfstab 파일의 samfs 유형으로서, mount(1M) 명령을 통해 식별됩니다.

## 공유 파일 시스템 지원

Sun StorageTek QFS 공유 파일 시스템은 다중 Solaris 운영 체제(OS) 호스트 시스템에 마운트할 수 있는 분산 파일 시스템입니다. Sun StorageTek QFS 공유 파일 시스템 환경에서 하나의 Solaris OS 호스트는 파일 시스템에 대한 메타 데이터 서버의 역할을 수행하고, 나머지 호스트는 클라이언트로 구성할 수 있습니다. 두 개 이상의 호스트를 가능한 메타 데이터 서버로 구성할 수 있지만, 오직 하나의 호스트만 한 시점에서 메타 데이터 서버로 구성할 수 있습니다. Sun StorageTek QFS 공유 파일 시스템 마운트 지점의 수에는 제한이 없습니다.

Sun StorageTek QFS 공유 파일 시스템의 장점은 파일 데이터가 광섬유 채널 디스크에서 호스트로 직접 전달된다는 것입니다. 데이터는 로컬 경로 I/O(직접 액세스 I/O라고도 함)를 통해 이동합니다. 이것은 네트워크를 통해 데이터를 전송하는 네트워크 파일 시스템(NFS)과 대조됩니다.

공유 파일 시스템은 Sun StorageTek QFS 공유 파일 시스템이나 SAM-QFS 공유 파일 시스템으로서 구현될 수 있습니다. 공유 파일 시스템은 ms 또는 ma 파일 시스템 유형을 사용할 수 있습니다.

Sun StorageTek QFS 공유 파일 시스템은 다음 사항을 지원하지 않습니다.

- 해당 파일 유형:
  - b - 블록 특수 파일
  - c - 문자 특수 파일
  - p - FIFO(명명된 파이프) 특수 파일

- 세그먼트된 파일. 세그먼트된 파일 환경에서 SAM-QFS 공유 파일 시스템을 구현할 수 없습니다.
- 강제 잠금. 강제 잠금이 설정되면 EACCES 오류가 반환됩니다. 그러나 권고 잠금은 지원됩니다. 권고 잠금에 대한 자세한 내용은 fcmt1(2) 시스템 호출을 참조하십시오.

공유 파일 시스템에 대한 자세한 내용은 4장, 67페이지의 "Sun StorageTek QFS 공유 파일 시스템 구성"을 참조하십시오.

## Linux 클라이언트 지원

공유 파일 시스템 내에서 Solaris 클라이언트뿐만 아니라 Linux 클라이언트에도 Sun StorageTek QFS 소프트웨어가 설치될 수 있습니다. Sun StorageTek QFS Linux 클라이언트 소프트웨어는 다음 Linux 배포를 지원합니다.

- x86/x64 플랫폼용 Red Hat Enterprise Linux 3.0(UD-5 및 UD8)(Sun StorageTek QFS 공유 클라이언트 전용)
- x64 플랫폼용 Red Hat Enterprise Linux 4.0(UD-2 및 UD-4)(Sun StorageTek QFS 공유 클라이언트 전용)
- x64 플랫폼용 SuSE Linux Enterprise Server 8(서비스 팩 4)(Sun StorageTek QFS 공유 클라이언트 전용)
- x64 플랫폼 및 SGI Altix Itanium 시스템용 SuSE Linux Enterprise Server 9(서비스 팩 2)(Sun StorageTek QFS 공유 클라이언트 전용)
- x64 플랫폼용 SuSE Linux Enterprise Server 10(Sun StorageTek QFS 공유 클라이언트 전용)

---

주 - 이것은 SuSE Enterprise Linux 8을 지원하는 Sun StorageTek QFS의 마지막 버전입니다. 이 소프트웨어의 다음 버전에서는 지원되지 않습니다.

---

공유 Sun StorageTek QFS Solaris 클라이언트와 달리 Linux 클라이언트는 클라이언트 전용 동작으로만 제한됩니다. 가능한 메타 데이터 서버로 구성될 수 없습니다. Linux 클라이언트는 Sun StorageTek SAM 소프트웨어와의 상호 작용을 지원하지만 stage, archive, release 및 samu와 같은 Sun StorageTek SAM 명령은 지원하지 않습니다. Sun StorageTek QFS 파일 시스템 기능만 제공합니다.

Sun StorageTek QFS Linux 클라이언트 소프트웨어에 대한 자세한 내용은 Sun StorageTek QFS Linux 클라이언트 안내서를 참조하십시오.

## 추가 파일 시스템 기능

다음 추가 기능도 Sun StorageTek QFS 파일 시스템에 의해 지원됩니다.

- 파일 공간의 사전 할당 - 빠른 속도의 순차적 읽기 및 쓰기를 위해 `setfa(1)` 명령을 사용하여 연속적인 디스크 공간을 사전 할당할 수 있습니다.
- 응용 프로그램 프로그래밍 인터페이스(API) 루틴 - API 루틴을 이용하면 프로그램에서 연속적인 디스크 공간 사전 할당이나 특정 스트라이프 그룹 액세스와 같은 다양한 특수 기능을 수행할 수 있습니다. 이러한 루틴에 대한 자세한 내용은 `intro_libsam(3)` 매뉴얼 페이지를 참조하십시오.
- 조정 가능한 디스크 할당 단위(DAU) - DAU는 온라인 저장소의 기본 단위입니다. Sun StorageTek QFS 파일 시스템 소프트웨어에는 물리적 디스크 저장 장치가 있는 파일 시스템 조정 및 읽기-수정-쓰기 작업으로 인한 시스템 오버헤드 제거에 유용하게 사용될 수 있는 조정 가능한 DAU가 포함되어 있습니다. 4KB의 배수로 DAU 크기를 조정할 수 있습니다. 자세한 내용은 8페이지의 "디스크 할당 단위 지정"을 참조하십시오.
- 다중 스트라이프 그룹에 대한 지원 - 단일 파일 시스템에서 다중 RAID 장치를 지원하기 위해 Sun StorageTek QFS 소프트웨어는 스트라이프 그룹 정의를 지원합니다. 스트라이프 그룹에 대해 디스크 블록 할당을 최적화할 수 있으므로, 디스크 상의 할당 맵을 업데이트하기 위해 오버헤드를 줄일 수 있습니다. 사용자는 API 루틴 또는 `setfa(1)` 명령을 사용하여 스트라이프 그룹에 파일을 지정할 수 있습니다.
- Sun Cluster 시스템 상호 운용성 - Sun Cluster 환경의 경우 로컬 파일 시스템 및 고가용성 파일 시스템으로서 Sun StorageTek QFS 파일 시스템이 지원됩니다. 자세한 내용은 119페이지의 "Sun Cluster 환경의 Sun StorageTek QFS 구성"을 참조하십시오.

# 디자인의 기본

Sun StorageTek QFS 파일 시스템은 다중 스레드된 고급 저장소 관리 시스템입니다. 소프트웨어의 기능을 최대한으로 활용하려면 가능할 때마다 여러 파일 시스템을 만들어야 합니다.

Sun StorageTek QFS 파일 시스템은 디렉토리 검색에 대해 디렉토리의 시작부터 끝까지 검색하는 선형 검색을 사용합니다. 디렉토리에 파일 수가 많을수록 전체 디렉토리의 검색 시간도 길어집니다. 수 천개의 파일이 있는 디렉토리들이 있으면 검색 시간이 커질 수 있습니다. 이렇게 오래 걸리는 검색 시간은 파일 시스템을 복원할 때도 마찬가지로입니다. 성능을 향상시키고 파일 시스템 덤프 및 복원 속도를 높이려면 디렉토리 내의 파일 수를 10,000개 이하로 유지하십시오.

디렉토리 이름 검색 캐시(DNLC) 기능이 파일 시스템 성능을 향상시킵니다. 이 캐시는 경로가 짧은(30문자 이하) 파일에 대한 디렉토리 검색 정보를 저장하여 진행 중인 디렉토리 검색을 수행할 필요성을 제거합니다. DNLC 기능은 Solaris OS 9 이상의 모든 릴리스에서 사용 가능합니다.

다음 절에서는 파일 시스템 디자인에 영향을 주는 몇 가지 추가 기능을 다룹니다.

- 7페이지의 "Inode 파일 및 파일의 특징"
- 8페이지의 "디스크 할당 단위 지정"

## Inode 파일 및 파일의 특징

파일 시스템에 저장되는 파일의 유형이 파일 시스템 디자인에 영향을 미칩니다. inode는 파일 또는 디렉토리의 특징을 설명하는 512바이트 블록의 정보입니다. 이 정보는 파일 시스템 내에서 동적으로 할당됩니다.

Inode는 파일 시스템 마운트 지점에 있는 `.inodes` 파일에 저장됩니다. Sun StorageTek QFS `.inodes` 파일은 파일 데이터 장치와 구분되는 메타 데이터 장치에 상주합니다.

표준 Solaris 운영 체제(OS) inode와 같이 Sun StorageTek QFS 파일 시스템 inode에는 파일의 POSIX 표준 inode 시간 즉, 파일 액세스, 파일 수정 및 inode 변경 시간이 포함됩니다. Sun StorageTek QFS 파일 시스템 inode는 표 1-2에 표시된 다른 시간도 포함합니다.

표 1-2 `.inode` 파일의 내용

시간	활동
access	파일이 마지막으로 액세스된 시간. POSIX 표준
modification	파일이 마지막으로 수정된 시간. POSIX 표준

표 1-2 .inode 파일의 내용(계속)

시간	활동
changed	inode 정보가 마지막으로 변경된 시간. POSIX 표준
속성	Sun StorageTek QFS 파일 시스템에 특정한 속성이 마지막으로 변경된 시간. Sun Microsystems 확장
creation	파일이 만들어진 시간. Sun Microsystems 확장
residence	파일이 오프라인에서 온라인 또는 온라인에서 오프라인으로 변경된 시간. Sun Microsystems 확장

주 - WORM-FS(한 번 쓰고 많이 읽음) 패키지가 설치되는 경우 inode는 retention-end 날짜도 포함합니다. 자세한 내용은 186페이지의 "WORM-FS 파일 시스템 구성"을 참조하십시오.

Inode 파일 정보 보기에 대한 자세한 내용은 49페이지의 "파일 및 파일 속성 보기"를 참조하십시오.

## 디스크 할당 단위 지정

디스크 공간은 디스크 할당 단위(DAU)라고 부르는 온라인 디스크 저장소의 기본 단위로 할당됩니다. 섹터, 트랙 및 실린더는 물리적인 디스크 구조를 나타내는 반면, DAU는 파일 시스템 구조를 나타냅니다. 적절한 DAU 크기 및 스트라이프 크기를 설정하면 성능을 향상시키고 자기 디스크 사용법을 최적화할 수 있습니다. DAU 설정은 파일이 할당될 때 사용되는 최소 연속적 공간입니다.

다음 하위 절은 DAU 설정 및 스트라이프 너비 구성 방법을 설명합니다.

## DAU 설정 및 파일 시스템 구조

Sun StorageTek QFS 파일 시스템은 조정 가능 DAU를 사용합니다. 이 구성 가능한 DAU를 사용하여 파일 시스템을 물리적 디스크 저장 장치로 조정할 수 있습니다. 이 기능은 읽기-수정-쓰기 작업에 의한 시스템 오버헤드를 최소화하며 따라서 아주 큰 파일을 조작하는 응용 프로그램의 경우에 특히 유용합니다. 읽기-수정-쓰기 작업 제어 방법에 대한 정보는 214페이지의 "대형 파일에 대한 파일 전송 성능 향상"을 참조하십시오.

각 파일 시스템은 서버에서 활성 상태인 마운트된 여러 파일 시스템 중 하나인 경우에도 고유한 DAU 설정을 가질 수 있습니다. 가능한 DAU 설정은 사용하는 파일 시스템 유형에 따라 다릅니다. DAU 설정은 파일 시스템이 작성될 때 `sammkfs(1M)` 명령을 통해 결정됩니다. 이 값을 동적으로 변경할 수는 없습니다.

DAU 설정은 마스터 구성(mcf) 파일에 지정된 장치 및 파일 시스템 정의와 함께 작동합니다. mcf 파일에 대한 자세한 내용은 29페이지의 "파일 시스템 구성"을 참조하십시오.

## ms 및 ma Sun StorageTek SAM 파일 시스템

사용할 수 있는 두 개의 파일 할당 체계에는 ms 파일 시스템 유형 및 ma 파일 시스템 유형이 있습니다.

단일 분할 영역의 파일 시스템과 같이 단순한 Sun StorageTek QFS Sun StorageTek SAM 파일 시스템의 경우, 해당 파일 시스템은 mcf 파일에서 ms의 장비 유형 값으로 정의됩니다. ms 파일 시스템에서, 허용되는 유일한 장치 유형은 유형 md이며, 메타 데이터 및 파일 데이터가 모두 md 장치에 기록됩니다. 기본적으로 md 장치에서 DAU는 64KB입니다.

다중 분할 영역에 설치된 더욱 복잡한 Sun StorageTek QFS 파일 시스템은 mcf 파일에 장비 유형 ma로 정의됩니다. ma 파일 시스템에서는 메타 데이터가 mm 장치에 기록되며, 데이터는 md, mr 또는 gXXX 장치에 기록될 수 있습니다.

ma 파일 시스템 내에서는 다음과 같은 장치들을 혼합할 수 있습니다.

- mm 및 mr 장치
- mm 및 gXXX 장치
- mm, mr 및 gXXX 장치
- mm 및 md 장치

이 장치 유형에 대한 자세한 내용은 표 2-2를 참조하십시오.

## 이중 및 단일 할당 체계

md 및 mm 장치는 다음과 같이 이중 할당 체계를 사용합니다.

- md 데이터 장치에서 작은 할당은 4KB이고 큰 할당은 DAU입니다. 기본 DAU는 64KB입니다. 이 기본값은 `sammkfs(1M)` 명령에 `-a allocation_unit` 옵션을 사용하여 파일 시스템을 초기화할 때 변경할 수 있습니다. DAU 크기는 16, 32 또는 64KB가 될 수 있습니다.
- 파일이 md 장치에서 만들어질 때 시스템은 파일의 첫 8개 주소를 작은 할당 크기에 할당합니다. 더 많은 공간이 필요한 경우, 파일 시스템은 하나 이상의 큰 할당 크기 (DAU)를 사용하여 파일을 확장합니다. 결과적으로, 큰 파일에 대한 I/O 성능은 향상되고 여러 개의 작은 파일로 인해 발생할 수 있는 디스크 조각화가 최소화됩니다.

---

주 - ms 파일 시스템을 사용할 때, 스트라이프 너비를 0보다 크게 설정하여 디스크 전반에 걸쳐 메타 데이터 정보를 스트라이프해야 합니다. 그러나, 스트라이프 너비 및 DAU 크기를 설정하기 전에 [11페이지의 "데이터 디스크의 스트라이프 너비"](#)를 읽고 이해해야 합니다.

---

- mm 메타 데이터 장치에서 작은 할당은 4KB이고 큰 할당은 16KB입니다. 이중 할당 체계를 사용하면 파일 시스템이 메타 데이터를 더 효율적으로 디스크에 쓸 수 있으며 디스크 조각화를 최소화하는 데 도움이 됩니다.

파일 시스템에 저장되는 파일 데이터의 유형에 따라 더 큰 DAU 크기가 파일 시스템 성능을 크게 향상시킬 수 있습니다. 파일 시스템 성능 조정에 대한 자세한 내용은 [7장, 179페이지의 "고급 항목"](#)을 참조하십시오.

ma Sun StorageTek QFS 파일 시스템만 단일 할당 체계를 사용하는 장치를 포함할 수 있습니다. 이러한 파일 시스템은 다음과 같이 별도의 메타 데이터 장치와 데이터 장치로 구성되어 있습니다.

- 메타 데이터 장치는 장비 유형 mm으로만 정의될 수 있습니다.
- 데이터 장치는 장비 유형 md, mr 또는 gXXX로 정의될 수 있습니다. md 장치는 DAU 크기가 16KB, 32KB 또는 64KB로 제한됩니다.

mr 및 gXXX 장치는 단일 할당 체계를 따릅니다. 파일 시스템에서 mr 및 gXXX 장치를 혼합할 수 있지만 md 장치를 파일 시스템에서 mr 또는 gXXX 장치와 혼합할 수는 없습니다.

mr 및 gXXX 데이터 장치를 사용하는 Sun StorageTek QFS 파일 시스템에 대한 DAU 크기는 구성 가능합니다. 데이터 장치에서 사용될 수 있는 가능한 DAU 크기는 mcf 파일의 각 데이터 장치에 지정된 장비 유형 값에 따라 다릅니다. [표 1-3](#)은 이러한 DAU 크기를 표시합니다.

**표 1-3 Sun StorageTek QFS 장비 유형 값 및 DAU 크기**

장비 유형	DAU 크기
mr 또는 gXXX	8KB 증분으로 기본 크기를 조정하여 다른 DAU 크기를 지정할 수 있습니다. DAU 크기는 16KB부터 65,528KB(64MB) 사이의 값이 될 수 있습니다. 기본 DAU 크기는 mr의 경우 64KB이며 gXXX의 경우 256KB입니다.
md	이 장치 유형은 이중 할당 체계를 사용합니다. DAU는 16, 32 또는 64KB 길이로 구성할 수 있습니다. 기본 DAU 크기는 64KB입니다. ma 파일 시스템의 md 장치는 메타 데이터가 아닌 데이터를 저장하는 데에만 사용됩니다. ms 파일 시스템의 md 장치는 파일 데이터와 메타 데이터를 모두 저장하는 데 사용됩니다.

주 - 소프트웨어 버전 3.5를 사용하거나 소프트웨어 버전 4에서 sammkfs 호환 모드 플러그를 사용하여 파일 시스템을 구축한 경우 버전 1 수퍼 블록을 사용할 수 있습니다. 버전 1 수퍼 블록에서 mm 장치는 이중 할당 체계를 사용하지 않으며 mm 장치의 할당은 16KB입니다. 버전 2 수퍼 블록을 사용해야 Sun StorageTek QFS 파일 시스템의 md 장치를 정의할 수 있습니다. 버전 1 수퍼 블록을 사용하고 있는지 확인하려면 samfsinfo(1M) 명령을 사용합니다.

## 데이터 맞춤

데이터 맞춤은 RAID 컨트롤러의 할당 단위를 파일 시스템의 할당 단위와 일치시키는 것을 말합니다. 최적의 Sun StorageTek QFS 파일 시스템 맞춤 공식은 다음과 같습니다.

$$\text{할당\_단위} = \text{RAID\_스트라이프\_너비} \times \text{데이터\_디스크\_수}$$

예를 들어 RAID-5 장치에 총 9개의 디스크가 있는 경우 9개 중에서 하나는 패리티 디스크이므로 데이터 디스크의 수는 8개입니다. RAID 스트라이프 너비가 64KB인 경우 최적의 할당 단위는 64에 8을 곱한 512KB입니다.

동일한 파일 시스템 내에서 데이터 파일은 각 스트라이프 그룹(gXXX) 또는 데이터 디스크(mr 또는 md)를 통해 스트라이프되거나 라운드 로빈으로 할당됩니다.

일치하지 않는 맞춤은 읽기-수정-쓰기 작업을 발생시킬 수 있으므로 성능이 저하됩니다.

## 데이터 디스크의 스트라이프 너비

Sun StorageTek QFS ms 및 ma 파일 시스템 간에 스트라이프 너비 기본값은 다릅니다. 스트라이프 너비는 mount(1M) 명령에서 `-o stripe=n` 옵션으로 지정됩니다. 스트라이프 너비가 0으로 설정되면 라운드 로빈 할당이 사용됩니다.

다음 하위 절은 다양한 파일 시스템에서의 스트라이프 너비를 설명합니다.

ms 파일 시스템에서 스트라이프 너비는 마운트 시에 설정됩니다. 표 1-4는 기본 스트라이프 너비를 보여줍니다.

표 1-4 ms 파일 시스템 기본 스트라이프 너비

DAU	기본 스트라이프 너비	디스크에 쓰여지는 데이터의 양
16KB	8 DAU	128KB
32KB	4 DAU	128KB
64KB(기본값)	2 DAU	128KB

예를 들어 `sammkfs(1M)`가 기본 설정으로 실행되면 기본적으로 큰 DAU는 64KB입니다. `mount(1M)` 명령을 실행하는 경우 스트라이프 너비를 지정하지 않으면, 기본값이 사용되며 마운트 시 설정되는 스트라이프 너비는 2가 됩니다.

주 - ms 파일 시스템에서 스트라이프 너비를 0보다 크게 설정해야 메타 데이터 정보가 디스크 전반에 걸쳐 스트라이프됩니다.

표 1-4의 첫 번째 열에 있는 숫자를 두 번째 열의 숫자와 곱하는 경우 결과 숫자는 128KB입니다. Sun StorageTek QFS 파일 시스템은 디스크에 기록되는 데이터의 양이 최소한 128KB인 경우에 가장 효율적으로 동작합니다.

## 스트라이프 그룹을 사용하지 않는 ma 파일 시스템의 스트라이프 너비

Sun StorageTek QFS ma 파일 시스템에서 마운트 시 설정되는 스트라이프 너비는 스트라이프 그룹의 구성 여부에 따라 다릅니다. 스트라이프 그룹은 그룹으로 스트라이프된 장치의 모음입니다. 스트라이프 그룹에 대한 자세한 내용은 [13페이지의 "파일 할당 방식"](#)을 참조하십시오. 이 절은 스트라이프 그룹 없이 구성된 Sun StorageTek QFS 파일 시스템에 대한 스트라이프 너비를 설명합니다.

스트라이프 그룹이 구성되지 않은 경우, ma 파일 시스템에서 DAU와 스트라이프 너비의 관계는 ms 파일 시스템의 경우와 유사합니다. 차이점은 64KB 이상인 DAU가 가능하다는 것과 DAU를 8KB 블록으로 구성할 수 있다는 것입니다. 최대 DAU 크기는 65,528KB입니다.

스트라이프 너비가 지정되지 않으면 디스크에 기록되는 데이터 양은 기본적으로 약 128KB입니다. Sun StorageTek QFS 파일 시스템은 기록 연산이 I/O 요청당 적어도 하나의 전체 스트라이프를 기록하는 경우 가장 효율적입니다. [표 1-5](#)는 기본 스트라이프 너비를 나타냅니다.

표 1-5 기본 스트라이프 너비

DAU	기본 스트라이프 너비	디스크에 쓰여지는 데이터의 양
16KB	8 DAU	128KB
24KB	5 DAU	120KB
32KB	4 DAU	128KB
40KB	3 DAU	120KB
48KB	2 DAU	96KB
56KB	2 DAU	112KB
64KB(기본값)	2 DAU	128KB
72KB	1 DAU	72KB
128KB	1 DAU	128KB
> 128KB	1 DAU	DAU 크기

## 스트라이프 그룹을 사용하는 ma 파일 시스템의 스트라이프 너비

Sun StorageTek QFS 파일 시스템에 대한 스트라이프 그룹이 구성된 경우, 할당되는 최소 공간은 DAU에 스트라이프 그룹의 장치 수를 곱한 값입니다. 스트라이프 그룹을 사용하면 할당량이 매우 커질 수 있습니다.

스트라이프 그룹을 사용하는 경우 데이터는 하나의 장치에 있는 것처럼 여러 디스크 장치에 한 번에 기록됩니다. 스트라이프 그룹의 할당 크기는 DAU 크기에 스트라이프 그룹의 요소 수를 곱한 값과 같습니다.

-o stripe=*n* 마운트 옵션은 해당 할당이 다른 스트라이프 그룹으로 이동되기 전에 각 스트라이프 그룹에서 발생하는 할당의 수를 결정합니다. 파일 시스템이 -o stripe=0으로 마운트되면 할당은 언제나 하나의 스트라이프 그룹입니다.

기본적으로 설정값은 라운드 로빈 할당 방식을 지정하는 -o stripe=0입니다. 설정값은 최저 -o stripe=0(스트라이프 비활성) 또는 최고 -o stripe=255일 수 있습니다. 일치하지 않는 스트라이프 그룹이 있으면 시스템은 -o stripe=0으로 설정되며, 이 경우 파일이 하나의 스트라이프 그룹에만 상주할 수 있습니다.

할당 방식에 대한 자세한 내용은 [13페이지의 "파일 할당 방식"](#)을 참조하십시오.

## 메타 데이터 디스크의 스트라이프 너비

mount\_samfs(1M) 명령에 -o mm\_stripe=*n* 옵션을 사용하여 메타 데이터 디스크에 관한 메타 데이터 정보를 스트라이프할 수 있습니다. 기본 스트라이프 너비는 -o mm\_stripe=1이며, 이 설정은 파일 시스템이 하나의 16KB DAU를 메타 데이터 디스크에 쓴 후 다음 메타 데이터 디스크로 전환하도록 합니다. 작은 4KB DAU는 메타 데이터 디스크에 대해 사용됩니다.

기본적으로 여러 메타 데이터 장치가 있는 경우, mount(1M) 명령의 -o mm\_stripe=*n* 옵션에 지정된 대로 메타 데이터가 할당됩니다. 설정값은 최저 -o mm\_stripe=0으로 설정하여 스트라이프를 비활성화하거나, 최고 -o mm\_stripe=255로 설정할 수 있습니다.

---

# 파일 할당 방식

Sun StorageTek QFS 소프트웨어를 사용하면 라운드 로빈 및 스트라이프 할당 방식 모두를 지정할 수 있습니다. [표 1-6](#)은 사용된 기본 파일 할당 방식을 표시합니다.

**표 1-6** 기본 할당 방식

파일 시스템	메타 데이터	파일 데이터
Sun StorageTek QFS	스트라이프	스트라이프
Sun StorageTek QFS(스트라이프 그룹)	스트라이프	라운드 로빈
Sun StorageTek QFS 공유 파일 시스템	스트라이프	라운드 로빈

이 절의 나머지는 할당을 보다 자세하게 설명합니다.

## 메타 데이터 할당

메타 데이터 할당은 사용 중인 파일 시스템 유형에 따라 다릅니다.

- ms 파일 시스템의 경우, 메타 데이터는 md 장치 전반에 걸쳐 할당됩니다.
- ma 파일 시스템의 경우, 메타 데이터는 mm 장치 전반에 걸쳐 할당됩니다. mm 장치에는 파일 데이터가 할당되지 않습니다.

Inode의 길이는 512바이트입니다. 디렉토리의 길이는 처음에 4KB입니다. 표 1-7은 시스템이 메타 데이터를 할당하는 방법을 표시합니다.

표 1-7 메타 데이터 할당

메타 데이터 유형	ma 파일 시스템의 할당 증분	ms 파일 시스템의 할당 증분
Inodes(.inodes 파일)	16KB DAU	16, 32 또는 64KB DAU
간접 블록	16KB DAU	16, 32 또는 64KB DAU
디렉토리	4KB 블록 및 16KB DAU	4KB, 총 최대 32KB, 다음에는 DAU 크기

## 라운드 로빈 할당

라운드 로빈 할당 방식은 패밀리 세트의 연속하는 각 장치에 한 번에 하나의 데이터 파일을 씁니다. 라운드 로빈 할당은 다중 데이터 스트림에 유용합니다. 이러한 유형의 환경에서는 전체적인 성능이 스트라이프 성능보다 뛰어나기 때문입니다.

라운드 로빈 디스크 할당은 하나의 파일을 하나의 논리 디스크에 쓰는 방식입니다. 다음 파일은 그 다음 논리 디스크에 쓰여지는 방식으로 수행됩니다. 쓰여지는 파일의 수가 패밀리 세트에서 정의된 장치의 수와 일치할 경우, 파일 시스템은 선택된 첫 번째 장치부터 다시 시작합니다. 파일이 물리적 장치의 크기보다 크면, 파일의 첫 번째 부분은 첫 번째 장치에 쓰여지고 나머지 부분은 사용 가능한 저장 공간이 있는 다음 장치에 쓰여집니다. 각 파일의 크기는 I/O의 크기를 결정합니다.

라운드 로빈 할당은 /etc/vfstab 파일에 stripe=0을 입력하여 명시적으로 지정할 수 있습니다.

다음 그림은 ms 및 ma 파일 시스템의 라운드 로빈 할당을 표시합니다. 이 그림에서 파일 1은 디스크 1에, 파일 2는 디스크 2에, 파일 3은 디스크 3에 쓰여지는 방식으로 계속 이어집니다. 파일 6이 만들어지면 디스크 1에 쓰여져 라운드 로빈 방식 할당 체계가 다시 시작됩니다.

ms 파일 시스템

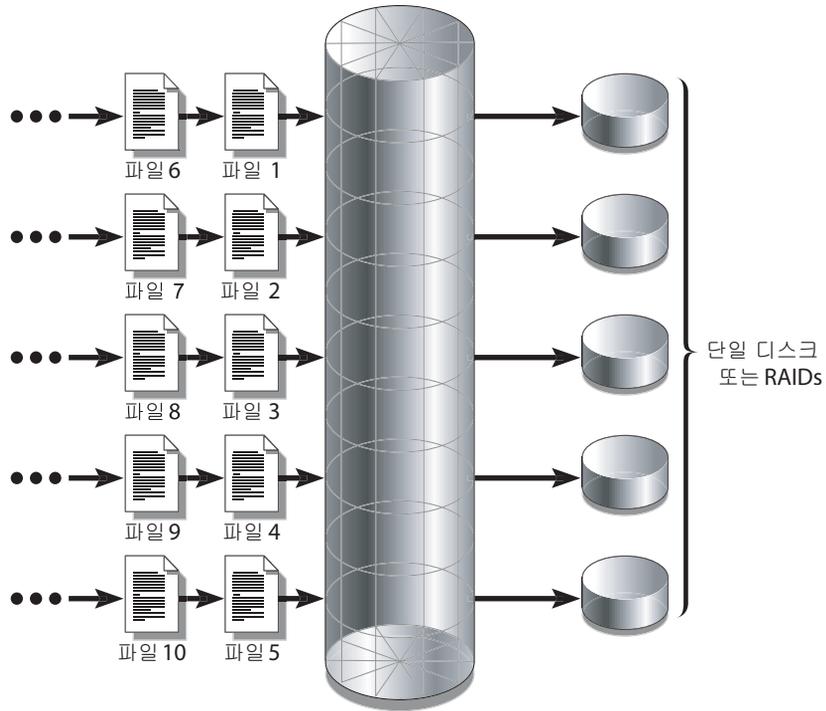


그림 1-1 5개 장치를 사용한 ms 파일 시스템의 라운드 로빈 할당

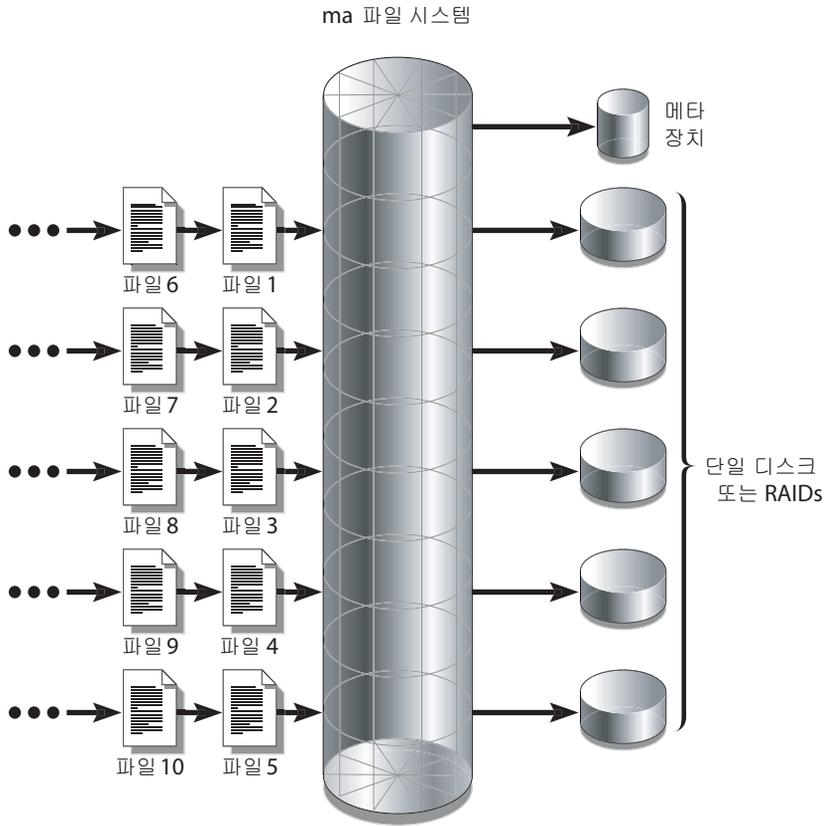


그림 1-2 5개 장치를 사용한 ma 파일 시스템의 라운드 로빈 할당

## 스트라이프 할당

기본적으로 Sun StorageTek QFS 파일 시스템은 스트라이프 할당 방법을 사용하여 파일 시스템 패밀리 세트의 모든 장치에 걸쳐 데이터를 분산시킵니다. 스트라이프는 여러 장치에 걸쳐 인터레이스 방식으로 동시에 파일을 쓰는 방법입니다.

스트라이프는 한 파일에 대한 성능을 위해 모든 장치의 추가 성능이 필요할 때 사용됩니다. 스트라이프 장치를 사용하는 파일 시스템은 순차 방식이 아닌 인터레이스 방식으로 블록을 어드레스합니다. 스트라이프는 여러 I/O 스트림이 여러 디스크에 걸쳐 파일을 동시에 쓸 수 있기 때문에 일반적으로 성능을 향상시킵니다. DAU 및 스트라이프 너비는 I/O 전송의 크기를 결정합니다.

스트라이프를 사용하는 파일 시스템에서 파일 1은 디스크 1, 디스크 2, 디스크 3, 디스크 4 및 디스크 5에 쓰여집니다. 파일 2 또한 디스크 1~5에 기록됩니다. DAU에 스트라이프 너비를 곱한 값이 한 블록에 있는 각 디스크에 쓰여질 데이터의 양을 결정합니다.

Sun StorageTek QFS 파일 시스템이 md 장치에 파일을 쓰는 경우, 먼저 해당 파일을 4KB의 작은 DAU에 맞추어서 시작합니다. 할당된 첫 번째 8개의 작은 DAU(32KB)에 파일이 적합하지 않으면 파일 시스템은 파일의 나머지 부분을 하나 이상의 큰 DAU에 씁니다.

Sun StorageTek QFS 파일 시스템이 mr 장치에 파일을 쓰는 경우, 먼저 하나의 DAU에 쓴 다음 또 다른 DAU에 쓰는 방식으로 이어집니다. mr 장치는 오직 하나의 DAU 크기만 가집니다.

다중 활성 파일은 라운드 로빈 할당의 경우보다 스트라이프의 경우에 훨씬 더 많은 디스크 헤드 이동을 유발합니다. I/O가 여러 파일에 동시에 발생할 경우 라운드 로빈 할당 방식을 사용합니다.

다음 그림은 스트라이프된 할당을 사용하는 ms 및 ma 파일 시스템을 나타낸 것입니다. 이 그림에서, 파일의 DAU x 스트라이프 너비 바이트가 디스크 1에 기록됩니다. 파일의 DAU x 스트라이프 너비 바이트는 디스크 2에 기록되며, 그 다음도 마찬가지입니다. 스트라이프의 순서는 파일에 대해 FIFO(선입선출) 방식으로 수행됩니다. 스트라이프는 모든 디스크에 걸쳐 I/O 로드가 분산됩니다.

ms 파일 시스템

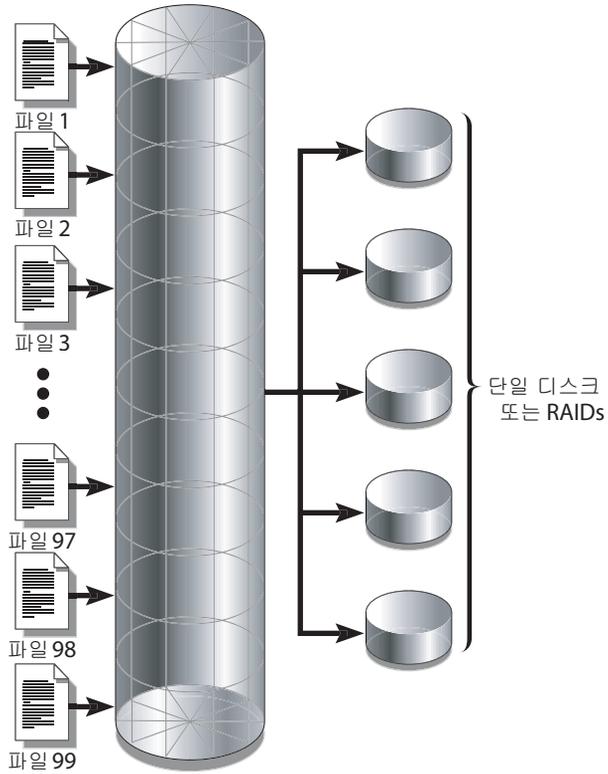


그림 1-3 5개 장치를 사용하는 ms 파일 시스템에서의 스트라이프

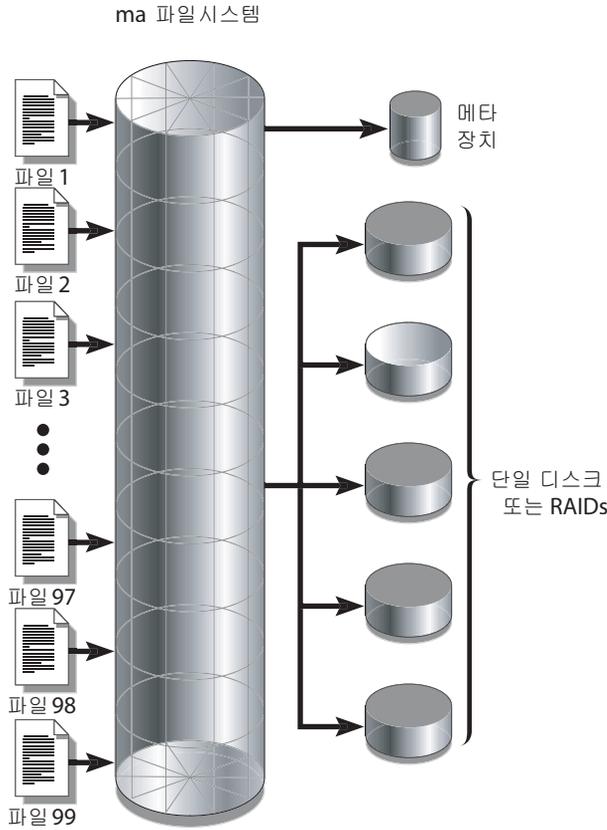


그림 1-4 5개 장치를 사용하는 ma 파일 시스템에서의 스트라이핑

## 스트라이프 그룹

스트라이프 그룹은 아주 큰 I/O 요구 사항 및 테라바이트 디스크 캐시가 있는 파일 시스템용으로 디자인된 Sun StorageTek QFS 할당 방식입니다. 스트라이프 그룹을 이용하면 여러 개의 물리적 디스크를 나타내는 장비 유형을 지정할 수 있습니다. 다중 스트라이프 그룹 장비 항목은 단일 Sun StorageTek QFS 파일 시스템을 구성할 수 있습니다. 스트라이프 그룹은 매우 큰 RAID 구성을 위해 비트맵 공간 및 시스템 업데이트 시간을 절약합니다.

스트라이프 그룹은 Sun StorageTek QFS 파일 시스템 내의 장치 모음입니다. mcf 파일에 gXXX 장치로 정의된 스트라이프된 그룹을 이용하여 하나의 파일이 두 개 이상의 장치에 쓰여지거나 읽혀질 수 있습니다. 파일 시스템 내에서 최대 128개의 스트라이프 그룹을 지정할 수 있습니다.

그림 1-5는 스트라이프 그룹 및 라운드 로빈 할당을 사용하는 Sun StorageTek QFS ma 파일 시스템을 나타낸 것입니다. 그림 1-5에서, qfs1 파일 시스템에 쓰여지는 파일은 정의된 스트라이프 그룹 g0, g1 및 g2 사이에 라운드 로빈으로 할당됩니다. 각 그룹은 두 개의 물리적 RAID 장치로 구성됩니다.

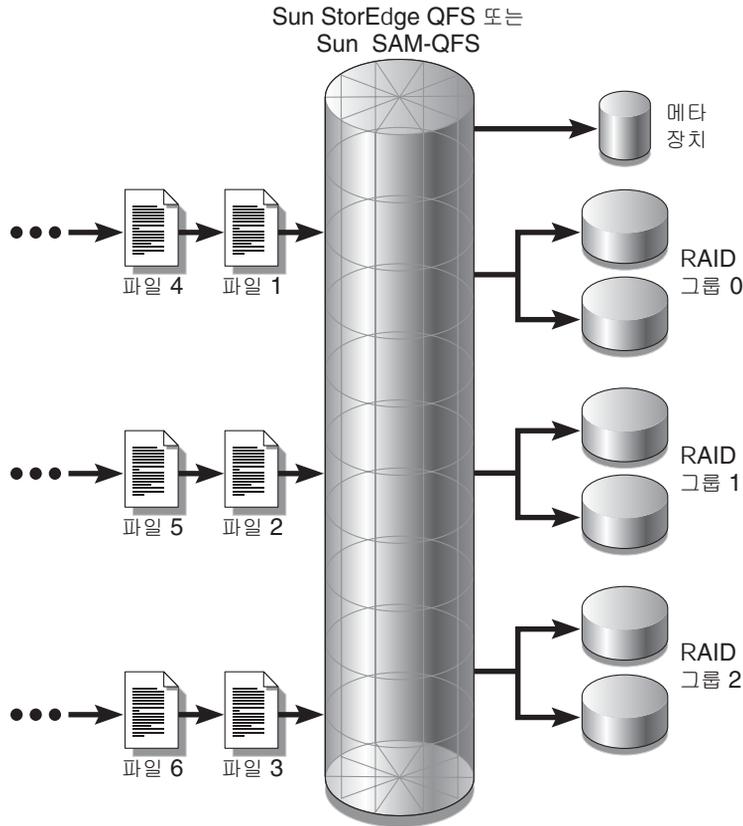


그림 1-5 Sun StorageTek QFS 라운드 로빈 스트라이프 그룹

그림 1-5의 구성에서 /etc/vfstab의 마운트 지점 옵션이 stripe=0으로 설정됩니다. 코드 예 1-1은 이 스트라이프 그룹을 선언하는 mcf 파일을 보여줍니다.

코드 예 1-1 스트라이프 그룹을 표시하는 mcf 파일 예제

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

그림 1-6은 스트라이프 그룹 및 스트라이프 할당을 사용하는 Sun StorageTek QFS ma 파일 시스템을 나타낸 것입니다. qfs1 파일 시스템에 쓰여지는 파일은 그룹 g0, g1 및 g2를 통해 스트라이프됩니다. 각 그룹에는 네 개의 물리적 RAID 장치가 포함됩니다. /etc/vfstab에서 마운트 지점 옵션은 stripe=1 이상으로 설정됩니다.

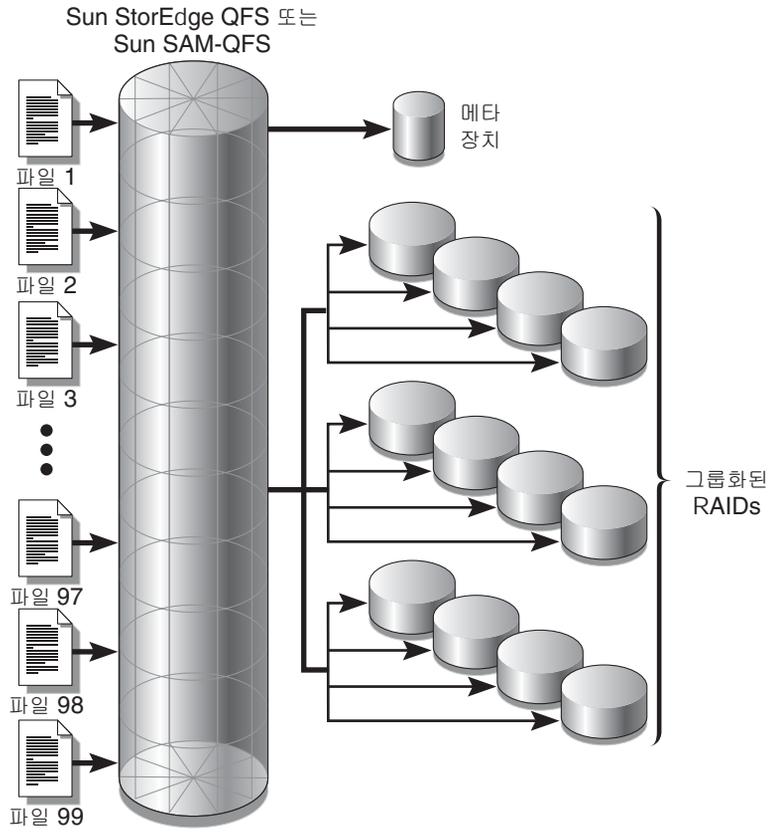


그림 1-6 Sun StorageTek QFS 스트라이프 그룹 할당

## 일치하지 않는 스트라이프 그룹

그룹마다 서로 다른 장치 수를 갖는 다중 스트라이프 그룹인 일치하지 않는 스트라이프 그룹으로 파일 시스템을 구축할 수 있습니다. Sun StorageTek QFS 파일 시스템은 일치하지 않는 스트라이프 그룹을 지원하지만 일치하지 않는 그룹에서의 스트라이핑은 지원하지 않습니다. 일치하지 않는 스트라이프 그룹이 있는 파일 시스템은 라운드 로빈 파일 시스템으로 마운트됩니다.

---

주 - 일치하지 않는 스트라이프 그룹이 포함된 파일 시스템에서 단일 파일은 여러 스트라이프 그룹에 확장될 수 없습니다. 파일이 상주하는 스트라이프 그룹이 가득 차면 확장될 수 없습니다. 일치하지 않는 스트라이프 그룹이 있는 경우 `setfa(1)` 명령의 `-g` 옵션을 사용하여 원하는 그룹으로 파일을 경로 지정합니다. 자세한 내용은 `setfa(1)` 매뉴얼 페이지를 참조하십시오.

스트라이프 그룹이 얼마나 채워졌는지 판별하려면 `samu(1M)` 운영자 유틸리티를 사용하고 `m` 디스플레이에 액세스하여 대용량 저장 상태를 표시합니다.

---

다음은 서로 다른 유형의 파일을 저장하기 위해 일치하지 않는 스트라이프 그룹을 사용하여 파일 시스템을 설정하는 방법에 대한 예제입니다.

## 일치하지 않는 스트라이프 그룹의 예제

비디오 및 오디오 데이터 모두가 있는 사이트에서 파일 시스템을 만들어야 한다고 가정합니다.

비디오 파일은 오디오 파일보다 상당히 크고 더 높은 성능이 필요합니다. 스트라이프 그룹은 매우 큰 파일에 대한 성능을 극대화하므로 큰 스트라이프 그룹이 있는 파일 시스템에 비디오 파일을 저장하기로 합니다.

오디오 파일은 비디오 파일보다 크기가 작으며 더 낮은 성능이 필요합니다. 오디오 파일은 작은 스트라이프 그룹에 저장하기로 합니다. 하나의 파일 시스템은 비디오 및 오디오 파일을 모두 지원할 수 있습니다.

**그림 1-7**은 필요한 파일 시스템을 나타낸 것입니다. 이것은 스트라이프 할당에서 일치하지 않는 스트라이프 그룹을 사용하는 `ma` 파일 시스템입니다.

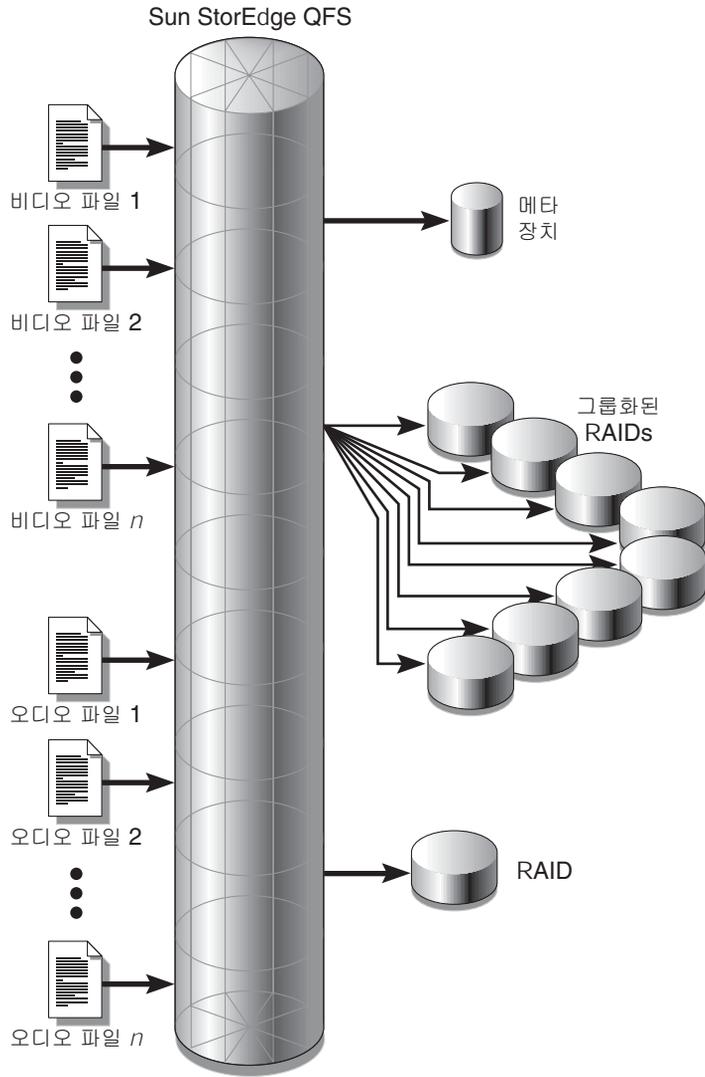


그림 1-7 스트라이프 할당에서 일치하지 않는 스트라이프 그룹을 사용하는 Sun StorageTek QFS 파일 시스템

표 1-8은 이 예제 파일 시스템의 특징을 나타낸 것입니다.

표 1-8 예제 파일 시스템 특징

특징	참고 사항
파일 시스템 이름	avfs
스트라이프 그룹 수	2개. 비디오 파일 그룹은 g0입니다. 오디오 파일 그룹은 g1입니다.
스트라이프 너비	0
DAU	128KB
g0에 대한 디스크 수	8
g0에 대한 최소 블록 크기	디스크 8개 X 128KB DAU = 1024KB 이 크기는 한 번의 블록 쓰기에서 쓰여지는 데이터의 양입니다. 각 디스크는 128KB의 데이터를 받으므로 한 번에 모든 디스크에 쓰여지는 총 데이터 양은 1024KB입니다.
g1에 대한 디스크의 수	1
g1에 대한 최소 블록 크기	디스크 1개 X 128KB DAU = 128KB

/etc/vfstab 파일에 다음 행을 추가하면 해당 환경에서 avfs 파일 시스템을 인식합니다.

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

/etc/vfstab 파일에서 라운드 로빈 파일 시스템을 지정하기 위해 stripe=0 이 사용된다는 점에 유의하십시오. 이 값이 사용되는 이유는 일치하지 않는 스트라이프 그룹에 대해 0보다 큰 값이 지원되지 않기 때문입니다.

코드 예 1-2는 avfs 파일 시스템을 위한 mcf 파일을 표시합니다.

코드 예 1-2 avfs 파일 시스템을 위한 mcf 파일

```
# 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 -
```

코드 예 1-2 avfs 파일 시스템을 위한 mcf 파일(계속)

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

이 파일 시스템을 위한 mcf 파일이 준비되면 코드 예 1-3에 표시된 `sammkfs(1M)` 및 `mount(1M)` 명령을 입력하여 avfs 파일 시스템을 작성 및 마운트할 수 있습니다.

코드 예 1-3 avfs 파일 시스템을 작성 및 마운트하는 명령

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

파일 시스템을 마운트한 후 코드 예 1-4에 표시된 명령을 사용하여 두 가지 유형의 파일에 대해 두 개의 디렉토리를 작성할 수 있습니다.

코드 예 1-4 avfs 파일 시스템에 디렉토리를 작성하는 명령

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

디렉토리가 만들어진 후, 코드 예 1-5에 표시된 `setfa(1)` 명령을 사용하여 큰 스트라이프 그룹을 비디오에 할당하고, 작은 스트라이프 그룹을 오디오에 할당할 수 있습니다. 이러한 디렉토리에 만들어지는 파일은 속성을 상속 받으므로 해당하는 각 스트라이프 그룹에 할당됩니다.

코드 예 1-5 파일 속성 설정을 위한 명령

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

`sammkfs(1M)` 명령에 대한 자세한 내용은 `sammkfs(1M)` 매뉴얼 페이지를 참조하십시오. `mount(1M)` 명령에 대한 자세한 내용은 `mount_samfs(1M)` 매뉴얼 페이지를 참조하십시오. `setfa(1)` 명령에 대한 자세한 내용은 `setfa(1)` 매뉴얼 페이지를 참조하십시오.

## 논리 장치 번호(LUN)당 할당 제어

필요할 경우 `nalloc` 명령으로 특정 Sun StorageTek QFS 데이터 분할 영역에 대한 할당을 비활성화하여 해당 장치에 추가로 할당되지 않도록 합니다. 이 기능은 현재 데이터 분할 영역에만 사용할 수 있으며 메타 데이터 분할 영역에는 사용할 수 없습니다.

`alloc` 또는 `on` 명령을 사용하여 분할 영역에 대한 할당을 다시 시작할 수 있습니다.

분할 영역의 할당 상태(`allocflag`)는 시스템이 재부트하더라도 지속됩니다.

`nalloc` 및 `alloc` 명령은 `samu` 인터페이스에서 사용할 수 있으며 `samu on` 명령도 할당을 `on`으로 설정합니다. `samu` 화면은 비활성화된 분할 영역에 대한 `nalloc` 상태를 표시합니다. `samtrace` 및 `samfsinfo` 출력은 할당 상태도 포함하고 있습니다.

`samu` 인터페이스에 대한 자세한 내용은 [253페이지의 "samu\(1M\) 운영자 유틸리티 사용"](#)을 참조하십시오.



# 파일 시스템 구성

---

설치 및 구성 프로세스는 Sun StorageTek QFS 설치 및 업그레이드 안내서에 자세히 설명되어 있습니다. 이 장에서는 Sun StorageTek QFS 환경에서 사용되는 파일 시스템의 구성에 대한 추가 정보를 제공합니다. 본 장에는 다음 절이 포함되어 있습니다.

- 29페이지의 "[File System Manager 소프트웨어 사용](#)"
- 34페이지의 "[mcf 파일의 기능](#)"
- 41페이지의 "[파일 시스템 초기화](#)"
- 42페이지의 "[구성 예제](#)"

---

## File System Manager 소프트웨어 사용

File System Manager 소프트웨어는 중앙 위치에서 네트워크에 있는 하나 이상의 파일 시스템을 구성, 제어, 보호 및 모니터링할 수 있는 브라우저 인터페이스 도구입니다. 이 중앙 위치에 액세스하기 위해 네트워크의 임의의 호스트에서 웹 브라우저를 사용할 수 있습니다.

이 소프트웨어의 목표는 파일 시스템과 연관된 대부분의 일반 작업을 수행하기 위해 명령줄 인터페이스(CLI) 명령보다 덜 복잡한 방법을 제공하는 것입니다. File System Manager 소프트웨어 설치에 대한 지침에 대해서는 Sun StorageTek QFS 설치 및 업그레이드 안내서를 참조하십시오.

기본적으로 File System Manager가 설치되는 서버를 관리하도록 설정됩니다. 또한 Sun StorageTek QFS 소프트웨어를 실행 중인 다른 서버를 관리하는 데도 사용할 수 있지만, 해당 추가 서버가 먼저 File System Manager 액세스를 허용하도록 구성되어야 합니다. 추가 관리 서버 추가에 대한 지침은 [33페이지의 "File System Manager 액세스를 위한 추가 서버 추가"](#)를 참조하십시오.

## ▼ 처음으로 File System Manager 호출

CLI 명령 대신 File System Manager를 호출하고 사용하여 파일 시스템 관리 및 구성을 수행하려는 경우 이 절차를 수행하십시오.

1. **File System Manager**가 설치된 서버 또는 해당 서버에 네트워크를 통해 액세스할 수 있는 임의의 컴퓨터에 로그인합니다.
2. 소프트웨어의 이전 버전에서 업그레이드한 경우 웹 브라우저를 열고 브라우저 캐시를 지웁니다.
3. 웹 브라우저에서 **File System Manager** 소프트웨어를 호출합니다.

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

*hostname*의 경우에는 File System Manager 소프트웨어가 설치되는 호스트의 이름을 입력합니다. 호스트 이름에 추가하여 도메인 이름을 지정해야 하는 경우 이 형식으로 *hostname*을 지정합니다. *hostname.domainname*. 이 URL은 http가 아니라 https로 시작합니다.

Sun Java™ Web Console 로그인 페이지가 표시됩니다.

4. **User Name** 프롬프트에 root 또는 다른 유효한 사용자 이름을 입력합니다.

---

주 - File System Manager 소프트웨어를 이전 버전으로부터 업그레이드했다면 samadmin 사용자 계정도 사용할 수 있습니다. **User Name** 필드에 samadmin을 입력한 후 samadmin 암호를 입력하면 File System Manager의 모든 작업에 대한 전체 액세스 권한을 얻게 됩니다.

---

5. **Password** 프롬프트에서 암호를 입력합니다.
6. **Log In**을 누릅니다.
7. 저장소 섹션에서 **File System Manager**를 누릅니다.  
이제 File System Manager에 로그인되었습니다.

## 추가 관리자 및 사용자 계정 만들기

초기 File System Manager 구성 후 언제라도 추가 관리자 및 게스트 계정을 만들 수 있습니다. 이러한 게스트 계정은 관리 스테이션에 국한됩니다.

File System Manager 소프트웨어를 제거한 경우 제거 스크립트는 수동으로 만든 추가 계정을 제거하지 않습니다. 사용자는 다음 절차 중 하나 또는 두 가지 모두를 사용하여 수동으로 추가한 모든 계정을 관리해야 합니다.

## ▼ 추가 계정 만들기

1. 브라우저 인터페이스 외부에서 관리 스테이션 서버에 root로 로그인합니다.
2. useradd 및 passwd 명령을 사용하여 각 사용자를 추가합니다.  
예를 들어 bobsmith 계정 이름을 가진 사용자를 추가하려면 다음을 입력합니다.

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

이러한 방법으로 추가한 사용자 계정에는 File System Manager 기능에 대해 읽기 전용인 보기 권한만 부여됩니다. 권한을 추가하려면 다음에 나오는 [31페이지의 "권한 수준 지정"](#) 절을 참조하십시오.

## 권한 수준 지정

사용자에게 File System Manager 기능에 대한 전체 또는 부분적인 액세스 권한을 지정할 수 있습니다. 다음 표에는 File System Manager 사용자에게 지정할 수 있는 5개의 권한 수준이 나열되어 있습니다.

표 2-1 File System Manager 권한 수준

관리 권한 수준	설명
com.sun.netstorage.fsmgr.config	사용자에게 액세스 권한 제한이 없습니다.
com.sun.netstorage.fsmgr.operator.media	사용자가 라이브러리를 추가 또는 제거할 수 있고, 독립형 드라이브를 추가 또는 제거할 수 있으며, VSN을 예약하고, 가져오며, 로드 및 언로드하고, 내보낼 수 있습니다.
com.sun.netstorage.fsmgr.operator.sam.control	사용자가 아카이브 작업을 시작 또는 중지하거나 유틸리티 상태로 지정할 수 있습니다.
com.sun.netstorage.fsmgr.operator.file	사용자가 스테이징을 시작 또는 중지할 수 있으며 파일 시스템을 복구할 수 있습니다.
com.sun.netstorage.fsmgr.operator.filesystem	사용자가 파일 시스템을 마운트 또는 마운트 해제하고, 마운트 옵션을 편집하며, 파일 시스템 검사(fsck)를 수행할 수 있습니다.

사용자에게 전체 또는 일부 구성 권한을 지정하려면 /etc/user\_attr 파일에 다음 행을 추가합니다.

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

account-name은 사용자 계정 이름이고 privilege-level은 사용자에게 지정할 권한 수준입니다.

예를 들어 사용자 계정 bobsmith에 전체 권한(권한 수준 com.sun.netstorage.fsmgr.config)을 지정하려면 /etc/user\_attr 파일에 다음 행을 추가합니다.

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

파일 시스템을 스테이징 및 복구하고(권한 수준 com.sun.netstorage.fsmgr.operator.file) VSN을 내보내고 가져오며 지정하기(권한 수준 com.sun.netstorage.operator.media) 위해서만 bobsmith 권한을 지정하려면 /etc/user\_attr 파일에 다음 행을 추가합니다.

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

## 다수 사용자를 위한 계정 만들기

다수의 사용자가 사용할 수 있는 일반적인 File System Manager 계정을 만든 후 해당 사용자 중 일부만 액세스할 수 있는 권한을 보유한 역할을 추가할 수 있습니다.

1. useradd와 passwd 명령을 사용하여 계정을 추가합니다.

예를 들어 다수의 사용자가 사용할 수 있는 guest라는 사용자 계정을 추가하려면 다음을 입력합니다.

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

2. roleadd와 passwd 명령을 사용하여 역할을 추가합니다.

guest 계정 내에 특수 권한이 있는 admin이라는 역할을 만들려면 다음을 입력합니다.

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

3. /etc/user\_attr 파일에 권한 수준을 지정합니다.

파일 시스템을 복구 및 스테이징하도록 admin 역할 권한을 지정하려면 /etc/user\_attr 파일에 다음 행을 추가합니다.

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

이 예제에서 사용자가 guest로 로그인하면 File System Manager가 사용자에게 No Role 또는 Admin 중 하나를 선택하라는 메시지를 표시합니다. 사용자가 Admin 역할 암호를 알면 Admin을 선택하여 Admin 암호를 입력하고 파일 시스템을 복구 및 스테이징할 수 있는 권한을 갖게 됩니다. 다른 모든 사용자들은 No Role을 선택하여 읽기 전용 권한만을 가져야 합니다.

동일한 권한 수준을 가진 다수의 사용자가 소프트웨어에 동시에 로그인할 수 있기 때문에 한 사용자의 변경 사항이 다른 사용자의 이전 변경 사항을 덮어쓸 위험이 있습니다. 이러한 상황을 방지하려면 어떤 사용자가 변경 작업을 수행할 수 있으며 이러한 변경 작업에 대해 다른 사용자에게 통지하는 방법에 대한 정책을 수립해야 합니다.

## ▼ File System Manager 액세스를 위한 추가 서버 추가

File System Manager는 기본적으로 이 소프트웨어가 설치된 서버를 관리하도록 설정됩니다. 또한 Sun StorageTek QFS 소프트웨어를 실행 중인 다른 서버를 관리하는 데도 사용할 수 있지만, 해당 추가 서버가 먼저 File System Manager 액세스를 허용하도록 구성되어야 합니다.

1. 브라우저 인터페이스 외부에서 telnet 유틸리티를 사용하여 추가하려는 서버에 연결합니다. root로 로그인합니다.
2. fsmadm(1M) add 명령을 사용하여 이 서버를 원격으로 관리할 수 있는 호스트 목록에 관리 스테이션(**File System Manager** 소프트웨어가 설치된 시스템)을 추가합니다.  
이 명령을 통해 목록에 추가된 호스트만이 서버를 원격으로 관리할 수 있습니다.  
예를 들어, 다음과 같습니다.

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

3. 관리 스테이션이 제대로 추가되었는지 확인하려면 fsmadm(1M) list 명령을 사용하여 관리 스테이션이 출력에 나열되는지 확인합니다.
4. **File System Manager** 브라우저 인터페이스에 관리자 사용자로 로그인합니다.
5. **Servers** 페이지에서 **Add**를 누릅니다.  
Add Server 창이 표시됩니다.
6. **Server Name** 또는 **IP Address** 필드에 새 서버의 이름이나 **IP** 주소를 입력합니다.
7. **OK**를 누릅니다.

## 세션 시간 초과 설정

Java Web Console 프레임워크의 기본 세션 시간 초과 값은 15분입니다. File System Manager 설치 프로그램에서 세션 시간 초과 값이 60분으로 변경됩니다. 세션 시간 초과를 다른 값으로 변경할 수는 있지만 보안 유지를 위해 60분을 초과하지 않는 값으로 설정하는 것이 좋습니다.

세션 시간 초과 값을 변경하려면 관리 스테이션에서 다음 명령을 입력합니다.

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

예를 들어 시간 초과 값을 45분으로 변경하려면 다음을 입력합니다.

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

---

## mcf 파일의 기능

/etc/opt/SUNWsamfs/mcf에 있는 마스터 구성 파일(mcf)은 Sun StorageTek QFS 소프트웨어의 제어 하에 있거나 해당 소프트웨어에 의해 사용되는 모든 장치에 대해 설명합니다. 시스템 구성 시에 이 ASCII 파일을 만들 때 각 장치에 대한 속성을 선언하고 각 파일 시스템의 장치를 패밀리 세트로 그룹화합니다.

mcf 파일에는 이러한 파일 시스템에서 RAID 및 디스크 장치를 식별하고 파일 시스템으로 구성하는 데 필요한 정보가 들어 있습니다. 또한 파일 시스템에 포함된 각 자동화된 라이브러리 또는 장치에 대한 항목도 들어 있습니다. 예제 mcf 파일은 /etc/opt/SUNWsamfs/examples/mcf에 있습니다.

mcf 파일은 코드 예 2-1에 표시된 것처럼 6개 열 또는 필드로 나누어진 스펙 코드의 행으로 구성됩니다.

코드 예 2-1 mcf 파일의 필드

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

mcf 파일에 데이터를 입력할 때 다음 규칙을 따릅니다.

- 파일의 필드 사이에는 공백 또는 탭 문자를 입력합니다.
- mcf 파일에 주석 행을 포함할 수 있습니다. 주석 행은 우물정자(#)로 시작됩니다.
- 일부 필드는 선택사항입니다. 대시(-) 문자를 사용하여 옵션 필드에 의미 있는 정보가 포함되지 않았음을 나타낼 수 있습니다.

mcf 파일 작성에 대한 자세한 내용은 mcf(4) 매뉴얼 페이지를 참조하십시오. 또한 File System Manager를 사용하여 mcf 파일을 작성할 수 있습니다. File System Manager 설치에 대한 정보는 Sun StorageTek QFS 설치 및 업그레이드 안내서를 참조하십시오. File System Manager 사용에 대한 정보는 해당 온라인 도움말을 참조하십시오.

다음 하위 절에서는 mcf 파일의 각 필드에 대해 설명합니다.

- 35페이지의 "Equipment Identifier 필드"
- 36페이지의 "Equipment Ordinal 필드"
- 36페이지의 "Equipment Type 필드"
- 37페이지의 "Family Set 필드"
- 38페이지의 "Device State 필드"
- 38페이지의 "Additional Parameters 필드"

## Equipment Identifier 필드

Equipment Identifier(장비 식별자) 필드는 필수 필드입니다. Equipment Identifier 필드를 사용하여 다음 종류의 정보를 지정합니다.

- 파일 시스템 이름. 이 필드에 파일 시스템 이름이 포함되는 경우 Family Set 필드의 이름과 동일해야 하며 mcf 파일의 후속 행들이 파일 시스템에 포함된 모든 디스크 또는 장치를 정의해야 합니다. 하나의 mcf 파일에 두 개 이상의 파일 시스템을 선언할 수 있습니다. 일반적으로 mcf 파일의 첫 데이터 행에서는 첫 번째 파일 시스템을 선언하고 나머지 행에서는 해당 파일 시스템에 포함된 장치를 지정합니다. mcf 파일에 선언된 다른 파일 시스템은 읽기 쉽도록 그 앞에 빈 주석 행이 올 수 있습니다. 파일 시스템 이름은 알파벳 문자로 시작되어야 하며 알파벳 문자, 숫자 또는 밑줄(\_) 문자만 포함할 수 있습니다.
- nodev 키워드. 키워드 nodev는 mcf 파일이 상주하는 시스템이 Solaris 호스트의 Sun StorageTek QFS 공유 파일 시스템에서 클라이언트 호스트로 사용된다고 표시합니다. 파일 시스템이 Sun Cluster 환경에 있는 경우 이 키워드를 사용하지 마십시오. 이 키워드는 메타 데이터 서버에 상주하는 하나 이상의 메타 데이터 장치에 대해 Equipment Identifier 로만 이 필드에 나타날 수 있습니다. Sun StorageTek QFS 공유 파일 시스템의 mcf 파일 작성에 대한 자세한 내용은 Sun StorageTek QFS 설치 및 업그레이드 안내서를 참조하십시오.
- 디스크 분할 영역 또는 슬라이스 설명. 이 필드의 /dev/ 항목은 디스크 분할 영역 또는 슬라이스를 나타냅니다.
- 자동화된 라이브러리 또는 광자기 설명. /dev/samst 항목은 자동화된 라이브러리 또는 광 드라이브를 식별합니다. 네트워크로 연결된 자동화된 라이브러리를 구성 중인 경우 자세한 내용은 Sun StorageTek Storage Archive Manager 설치 및 업그레이드 안내서를 참조하십시오.

- 테이프 드라이브 설명. 이 항목은 다음 두 양식 중 하나일 수 있습니다.
  - /dev/rmt 항목
  - /dev/rmt 링크가 가리키는 것과 동일한 파일을 가리키는 심볼릭 링크에 대한 경로. 이런 방법으로 테이프 드라이브를 지정하는 경우 파일 시스템을 마운트하기 전에 링크를 작성해야 합니다.

Equipment Identifier 필드에 파일 시스템의 이름이 포함되는 경우, 31자로 제한됩니다. 기타 모든 내용의 경우, 이 필드는 127자로 제한됩니다.

## Equipment Ordinal 필드

mcf 파일의 각 행에 대해 Equipment Ordinal (*eq*) 필드는 정의될 파일 시스템 구성 요소 또는 장치에 대한 숫자 식별자를 포함해야 합니다. 1부터 65534까지의 고유한 정수를 지정하십시오. 이 필드는 필수 필드입니다.

## Equipment Type 필드

Equipment Type(장비 유형) 필드에 2, 3 또는 4 문자의 코드를 입력합니다. 이 필드는 필수 필드입니다.

Sun StorageTek QFS 또는 SAM-QFS 파일 시스템의 Equipment Type 필드는 표 2-2에 표시된 임의의 값을 포함할 수 있습니다.

표 2-2 Sun StorageTek QFS 또는 SAM-QFS Equipment Type 필드

Equipment Type 필드 내용	의미
ma	별도 장치(mm 장치)에 메타 데이터를 저장하는 Sun StorageTek QFS 파일 시스템을 정의합니다.
ms	데이터와 메타 데이터를 동일한 장치에 저장하는 Sun StorageTek QFS 파일 시스템을 정의합니다.
md	파일 데이터 저장을 위한 스트라이프 또는 라운드 로빈 장치를 정의합니다.
mm	inode 및 데이터가 아닌 기타 정보 저장을 위한 메타 데이터 장치를 정의합니다. 여러 메타 데이터 장치를 지정할 수 있습니다. Sun StorageTek QFS <sub>ma</sub> 파일 시스템의 메타 데이터(inode, 디렉토리, 할당 맵 등 포함)는 메타 데이터 장치에 위치하고 파일 데이터 장치와 구분됩니다. 여러 메타 데이터 장치가 있는 경우, 기본적으로 메타 데이터는 라운드 로빈 할당을 사용하여 할당됩니다.
mr	라운드 로빈 또는 스트라이프 데이터 장치를 정의합니다.

Equipment Type 필드 내용	의미
gXXX	<p>스트라이프 그룹 데이터를 정의합니다. 스트라이프 그룹은 문자 g로 시작하고 그 다음에 숫자가 옵니다. 예를 들어 숫자 g12는 반드시 0과 127 사이(0 및 127 포함)의 정수여야 합니다. 스트라이프 그룹의 모든 구성원은 동일한 유형과 크기여야 합니다. 한 파일 시스템 내의 다른 스트라이프 그룹은 동일한 수의 멤버가 필요하지 않습니다. md, mr 및 gXXX 장치는 한 파일 시스템에서 혼합될 수 없습니다.</p> <p>데이터는 그룹 간에 스트라이프되거나(모든 그룹이 동일한 수의 장치를 포함하는 경우) 라운드 로빈될 수 있습니다. 기본값은 라운드 로빈입니다.</p>

파일 시스템 장비 유형 이외에도, 자동화된 라이브러리 및 기타 장치를 식별하기 위해 다른 코드가 사용됩니다. 특정 장비 유형에 대한 자세한 내용은 mcf(4) 매뉴얼 페이지를 참조하십시오.

## Family Set 필드

패밀리 세트는 장치 그룹에 대한 이름을 포함합니다. 이 필드는 필수 필드입니다.

패밀리 세트 이름은 영문자로 시작되어야 하며 영문자, 숫자 또는 밑줄(\_) 문자만 포함할 수 있습니다.

파일 시스템의 디스크 장치를 정의하는 행은 모두 동일한 패밀리 세트 이름을 포함해야 합니다. 소프트웨어는 패밀리 세트 이름을 사용하여 장치를 하나의 파일 시스템으로 그룹화합니다. sammkfs(1M) 명령이 실행될 때 파일 시스템의 모든 장치에 물리적으로 패밀리 세트 이름을 기록합니다. samfsck(1M) 명령에서 -F 및 -R 옵션을 사용하여 이 이름을 변경할 수 있습니다. sammkfs(1M) 명령에 대한 자세한 내용은 sammkfs(1M) 매뉴얼 페이지를 참조하십시오. samfsck(1M) 명령에 대한 자세한 내용은 samfsck(1M) 매뉴얼 페이지를 참조하십시오.

자동화된 라이브러리의 장치를 정의하는 행과 라이브러리의 연관된 드라이브에 있는 장치를 정의하는 행은 동일한 패밀리 세트 이름을 포함해야 합니다.

수동으로 로드된 제거 가능한 독립형 매체 장치의 경우 이 필드에 대시(-)를 포함할 수 있습니다.

식별자 #family-set-name을 패밀리 세트의 첫 번째 장치 바로 앞에 삽입하여 특정 패밀리 세트와 연관된 주석을 만들 수 있습니다. 주석 행과 패밀리 세트의 마지막 장치 사이에 추가된 임의의 주석이 해당 패밀리 세트와 연관됩니다. 패밀리 세트가 File System Manager 소프트웨어를 통해 나중에 삭제된 경우, 관련된 주석도 mcf 파일에서 삭제됩니다.

## Device State 필드

Device State(장치 상태) 필드는 파일 시스템이 초기화될 때 장치의 상태를 지정합니다. 유효한 장치 상태는 on 및 off입니다. 이 필드는 옵션 필드입니다. 값을 지정하지 않으려는 경우 대시 문자(-)를 삽입하여 이 필드가 생략되었음을 표시하십시오.

## Additional Parameters 필드

SAM-QFS 파일 시스템의 경우, Additional Parameters(추가 매개변수) 필드는 옵션이며 공백으로 비워둘 수 있습니다. 기본적으로 라이브러리 카탈로그 파일은 `/var/opt/SUNWsamfs/catalog/family-set-name`에 쓰여집니다. 라이브러리 카탈로그 파일에 대한 대체 경로를 지정하려면 이 필드를 사용하십시오.

Sun StorageTek QFS 공유 파일 시스템의 경우, 이 필드에 키워드 `shared`가 포함되어야 합니다.

Sun StorageTek QFS 공유 해제된 파일 시스템의 경우, 대시(-)를 입력하거나 이 필드를 공백으로 둡니다.

## mcf 파일의 예제

각 파일 시스템 구성은 고유합니다. 시스템 요구 사항 및 실제 하드웨어는 사이트마다 다릅니다. 다음 코드 예제는 예제 mcf 파일을 나타냅니다.

**코드 예 2-2**는 스트라이프 그룹이 두 개인 Sun StorageTek QFS 파일 시스템에 대한 mcf 파일을 보여줍니다.

**코드 예 2-2**                    스트라이프 그룹을 표시하는 mcf 파일 예제

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

코드 예 2-3은 세 개의 Sun StorageTek QFS 파일 시스템이 있는 mcf 파일을 나타냅니다.

코드 예 2-3                    세 개의 SAM-QFS 파일 시스템을 표시하는 mcf 파일 예제

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

코드 예 2-4는 md 장치를 사용하는 하나의 SAM-QFS 파일 시스템이 있는 mcf 파일을 나타낸 것입니다. 이 mcf 파일은 테이프 라이브러리도 정의합니다.

```
# 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
```

파일 시스템 구성의 추가 예제는 Sun StorageTek QFS 설치 및 업그레이드 안내서를 참조하십시오.

## 파일 설정, 옵션 및 명령 사이의 관계

mcf 파일은 각 파일 시스템을 정의하지만 파일 시스템 작동은 기본 시스템 설정, /etc/vfstab 파일의 설정, samfs.cmd 파일의 설정 및 mount(1M) 명령의 옵션 사이의 관계에 따라 다릅니다.

스트라이프 너비와 같은 일부 마운트 옵션은 둘 이상의 위치에 지정할 수 있습니다. 이렇게 지정할 경우, 한 위치의 설정은 다른 위치의 설정을 무효화하게 됩니다.

마운트 옵션을 지정하는 여러 가지 방법에 대한 자세한 내용은 [59페이지의 "마운트 매개변수 설정"](#)을 참조하십시오.

# 파일 시스템 초기화

새 파일 시스템을 작성하거나 이전 또는 손상된 파일 시스템을 바꾸려는 경우 `sammkfs(1M)` 명령을 사용하여 초기화해야 합니다.

`sammkfs(1M)` 명령으로 새로운 파일 시스템을 구축할 수 있으며, `-a allocation-unit` 옵션을 사용하여 DAU 설정을 지정할 수 있습니다.

Sun StorageTek QFS 소프트웨어의 버전 4U2 및 후속 릴리스는 두 가지 수퍼 블록 디자인을 지원합니다. 코드 예 2-5에서 `samfsinfo(1M)` 명령 출력이 `samfs1` 파일 시스템이 버전 2 수퍼 블록을 사용 중임을 표시합니다.

코드 예 2-5 `samfsinfo(1M)` 명령 예제

```
# 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
```

이러한 수퍼 블록과 관련한 다음의 작동 및 기능적 차이점을 알아 두십시오.

- 4U0 이전의 릴리스는 버전 수퍼 블록 디자인만 지원합니다.
- 4U0 이상의 릴리스는 버전 2 수퍼 블록을 지원합니다. 4U0 소프트웨어를 업그레이드로 설치한 경우, 버전 2 수퍼 블록에 의존하는 기능을 사용하기 전에 4U0 이상의 `sammkfs(1M)` 명령을 사용하여 기존 파일 시스템을 다시 초기화해야 합니다. 액세스 제어 목록(ACL)과 같은 특정 기능 및 Sun StorageTek QFS 공유 파일 시스템은 버전 2 수퍼 블록에서만 지원됩니다. 파일 시스템 재초기화는 소프트웨어 설치 업그레이드 프로세스에서 하나의 단계로 설명되어 있지만, 이 작업은 소프트웨어가 설치된 후 언제든지 수행할 수 있습니다.



주의 - 버전 2 수퍼 블록을 사용하는 파일 시스템은 4U0 이전의 릴리스로 되돌릴 수 없습니다. 4U6 릴리스 소프트웨어를 사용하여 버전 1 수퍼 블록을 작성할 수 없습니다.

버전 2 수퍼 블록을 필요로 하는 기능 또는 `sammkfs(1M)` 명령을 사용하여 버전 2 수퍼 블록을 만드는 방법에 대한 자세한 내용은 Sun StorageTek QFS 설치 및 업그레이드 안내서를 참조하십시오.

다음 예제는 파일 시스템 이름만 유일한 인수로 가지는 가장 간단한 형식으로 된 `sammkfs` 명령입니다.

```
# sammkfs samqfs1
```

앞선 명령은 독립형 Sun StorageTek QFS 또는 SAM-QFS 파일 시스템을 위한 버전 2 수퍼 블록을 구축합니다.

`sammkfs(1M)` 명령, 해당 옵션 및 버전 1과 버전 2 수퍼 블록의 의미에 대한 자세한 내용은 `sammkfs(1M)` 매뉴얼 페이지를 참조하십시오. `sammkfs(1M)` 명령을 사용한 공유 Sun StorageTek QFS 파일 시스템 초기화에 대한 정보는 Sun StorageTek QFS 설치 및 업그레이드 안내서를 참조하십시오.

## 구성 예제

이 절에서는 구성 예제를 설명하고 서버에서 `mcf` 파일을 설정하는 것과 관련된 여러 단계 및 고려 사항을 보여줍니다. 여기에는 다음과 같은 절차가 포함됩니다.

- 42페이지의 "Sun StorageTek QFS 라운드 로빈 디스크 구성 만들기"
- 44페이지의 "Sun StorageTek QFS 스트라이프 디스크 구성 만들기"
- 45페이지의 "Sun StorageTek QFS 스트라이프 그룹 구성 만들기"

SAM-QFS 구성의 모든 예제는 본질적으로 디스크 캐시 크기 이상으로 파일 시스템을 확장하고 여기에는 자동화된 라이브러리 및 정의된 기타 제거 가능한 매체 장치도 있을 수 있다는 점에 유의하십시오. 제거 가능한 매체 장치 구성은 하나의 예제에서만 나타납니다. 제거 가능한 매체 장치 구성에 대한 내용은 Sun StorageTek Storage Archive Manager 설치 및 업그레이드 안내서를 참조하십시오.

구성 예제에서는 파일 시스템이 시스템에 로드되어 있고 모든 파일 시스템은 마운트 해제되어 있다고 가정합니다.

### ▼ Sun StorageTek QFS 라운드 로빈 디스크 구성 만들기

이 구성 예제는 메타 데이터를 지연 시간이 낮은 디스크에 분리하는 Sun StorageTek QFS 파일 시스템입니다. 라운드 로빈 할당은 4개의 분할 영역에서 사용됩니다. 각 디스크는 별도의 컨트롤러에 있습니다.

이 절차에서는 다음과 같은 사항을 가정합니다.

- 메타 데이터 장치는 장비 서수 11로 지정된 장치의 컨트롤러 5, 논리 장치 번호(LUN) 0에서 사용되는 단일 분할 영역(s6)입니다.
- 데이터 장치는 4개의 컨트롤러에 연결된 4개의 디스크로 구성됩니다.

1. 코드 예 2-6에 나타난 것과 같이 편집기를 사용하여 mcf 파일을 만듭니다.

코드 예 2-6

Sun StorageTek QFS 라운드 로빈 mcf 파일 예제

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

2. mkdir(1) 명령을 사용하여 /qfs1 파일 시스템에 대한 /qfs 마운트 지점을 만듭니다.

```
# mkdir /qfs
```

3. sammkfs(1M) 명령을 사용하여 파일 시스템을 초기화합니다.

다음 예제에서는 기본값인 64KB DAU를 사용합니다.

```
# sammkfs qfs1
```

4. 편집기를 사용하여 /etc/vfstab 파일을 수정합니다.

mr 데이터 장치가 있는 Sun StorageTek QFS 파일 시스템은 기본적으로 스트라이프 할당을 사용하므로 라운드 로빈 할당하려면 stripe=0을 설정해야 합니다. 파일 시스템에서 라운드 로빈을 명시적으로 설정하려면 다음과 같이 stripe=0을 설정합니다.

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

5. mount(1M) 명령을 사용하여 파일 시스템을 마운트합니다.

```
# mount /qfs
```

## ▼ Sun StorageTek QFS 스트라이프 디스크 구성 만들기

이 예제 구성에서 파일 데이터는 기본적으로 네 개의 데이터 분할 영역으로 스트라이프됩니다.

이 절차에서는 다음과 같은 사항을 가정합니다.

- 메타 데이터 장치는 컨트롤러 0, LUN 1에서 사용되는 단일 분할 영역(s6)입니다. 메타 데이터는 장치 서수 11에만 쓰여집니다.
- 데이터 장치는 4개의 컨트롤러에 연결된 4개의 디스크로 구성됩니다. 각 디스크는 별도의 컨트롤러에 있습니다.

1. 코드 예 2-7에 나타난 것과 같이 편집기를 사용하여 mcf 파일을 만듭니다.

코드 예 2-7 Sun StorageTek QFS 스트라이프 디스크 mcf 파일 예제

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

2. `mkdir(1)` 명령을 사용하여 `/qfs1` 파일 시스템에 대한 `/qfs` 마운트 지점을 만듭니다.

```
# mkdir /qfs
```

3. `sammkfs(1M)` 명령을 사용하여 파일 시스템을 초기화합니다.

기본 DAU는 64KB이지만, 다음 예제에서는 DAU 크기를 128KB로 설정합니다.

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

이 구성에서 이 파일 시스템에 쓰여진 모든 파일은 128KB 증분으로 모든 장치에 걸쳐 스트라이프됩니다.

4. 편집기를 사용하여 /etc/vfstab 파일을 수정합니다.

Sun StorageTek QFS 파일 시스템은 기본적으로 스트라이프 할당을 사용합니다. 이 예제에서는 스트라이프 너비를 기본값인 `stripe=1`로 설정합니다. 다음 설정은 1 DAU의 스트라이프 너비로 4개의 모든 `mr` 장치에 걸쳐 데이터를 스트라이프합니다.

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

5. `mount(1M)` 명령을 사용하여 파일 시스템을 마운트합니다.

```
# mount /qfs
```

## ▼ Sun StorageTek QFS 스트라이프 그룹 구성 만들기

스트라이프 그룹을 이용하면 매우 큰 파일의 경우 RAID 장치도 함께 그룹화할 수 있습니다. 하나의 DAU는 비트맵에서 1비트로 나타납니다. 스트라이프 그룹에  $n$ 개의 장치가 있는 경우,  $n$ 에 DAU를 곱한 값이 최소 할당 크기입니다. 비트맵에서는 오직 1비트만  $n \times$  DAU를 나타내는 데 사용됩니다.

스트라이프 그룹에 할당된 최소 디스크 공간은 다음과 같습니다.

$$\text{minimum-disk-space-allocated} = \text{DAU} \times \text{number-of-disks-in-the-group}$$



주의 - 단일 바이트의 데이터 쓰기는 스트라이프 그룹에 할당된 전체 최소 디스크 공간을 채웁니다. 스트라이프 그룹은 매우 특수한 애플리케이션에 대해 사용됩니다. 파일 시스템에서 스트라이프 그룹 사용으로 인한 영향을 이해해야 합니다.

총 스트라이프 너비에 장치의 수를 곱한 값보다 작은 너비를 가진 파일(이 예제에서는 길이에 있어서  $128\text{KB} \times 4$  디스크 =  $512\text{KB}$ 보다 작은 파일)도  $512\text{KB}$ 의 디스크 공간을 사용합니다.  $512\text{KB}$ 보다 큰 파일에 대해서는  $512\text{KB}$  증분으로 총 공간에 필요한 공간이 할당됩니다.

스트라이프 그룹 내의 장치는 동일한 크기여야 합니다. 스트라이프 그룹의 크기를 늘리기 위해 장치를 추가하는 것은 불가능합니다. 그러나 `samgrowfs(1M)` 명령을 사용하여 스트라이프 그룹을 추가할 수는 있습니다. 이 명령에 대한 자세한 내용은 `samgrowfs(1M)` 매뉴얼 페이지를 참조하십시오.

이 구성 예제는 메타 데이터를 지연 시간이 낮은 디스크에 분리하는 Sun StorageTek QFS 파일 시스템입니다. 4개의 드라이브에 2개의 스트라이프 그룹이 설정되어 있습니다.

이 절차에서는 다음과 같은 사항을 가정합니다.

- 메타 데이터 장치는 컨트롤러 0, LUN 1에서 사용되는 단일 분할 영역(s6)입니다.
- 데이터 장치는 4개의 컨트롤러에 연결된 4개의 디스크(2개의 동일한 디스크로 구성된 2개의 그룹)로 구성됩니다. 각 디스크는 별도의 LUN에 존재합니다. 분할 영역 6이 전체 디스크를 차지한다고 가정하여 데이터 저장을 위해 전체 디스크가 사용됩니다.

1. 코드 예 2-8에 나타난 것과 같이 편집기를 사용하여 mcf 파일을 만듭니다.

코드 예 2-8 Sun StorageTek QFS 스트라이프 그룹 mcf 파일 예제

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

2. mkdir(1) 명령을 사용하여 /qfs1 파일 시스템에 대한 /qfs 마운트 지점을 만듭니다.

```
# mkdir /qfs
```

3. sammkfs(1M) 명령을 사용하여 파일 시스템을 초기화합니다.

다음 예제에서는 DAU 크기를 128KB로 설정합니다.

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

4. 편집기를 사용하여 /etc/vfstab 파일을 수정합니다.

다음 예제에서는 스트라이프 그룹 g0부터 스트라이프 그룹 g1까지 라운드 로빈 할당을 지정하는 데 필요한 기본 설정값인 stripe=0을 사용합니다.

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

이 /etc/vfstab 파일은 stripe= 옵션을 사용하여 스트라이프 너비를 설정합니다. 이 예제에는 2개의 스트라이프 그룹인 g0과 g1이 있습니다. stripe=0으로 지정된 경우, 파일은 두 개의 스트라이프 그룹에 라운드 로빈 방식으로 기록됩니다.

---

주 - 스트라이프 그룹의 구성이 만들어진 후에 이 구성을 변경하려면 다른 `sammkfs(1M)` 명령을 실행해야 합니다.

---

5. `mount(1M)` 명령을 사용하여 파일 시스템을 마운트합니다.

```
# mount /qfs
```



# 작업 수행

이 장에서는 파일 시스템 작업과 관련된 항목을 다룹니다. 본 장에는 다음 절이 포함되어 있습니다.

- 49페이지의 "파일 및 파일 속성 보기"
- 53페이지의 "시스템에 대한 구성 파일 변경 전파"
- 59페이지의 "마운트 매개변수 설정"
- 62페이지의 "파일 시스템 마운트 해제"
- 63페이지의 "파일 시스템에 디스크 캐시 추가"
- 64페이지의 "파일 시스템 재작성"

## 파일 및 파일 속성 보기

Sun StorageTek QFS 파일 시스템에 특정한 속성에 사용자 설정 및 일반 파일 상태 모두가 포함됩니다. 이 절에서는 이러한 속성을 설명하고 `sfs` 명령을 사용하여 해당 속성을 보는 방법을 설명합니다.

## 파일 속성 및 파일 상태

파일의 사용자 지정 속성 및 시스템 지정 상태는 해당 파일의 `inode`에 저장됩니다. `sfs(1) -D` 명령을 사용하여 이러한 `inode` 속성을 표시할 수 있습니다. `sfs(1)` 옵션에 대한 자세한 내용은 `sfs(1)` 매뉴얼 페이지를 참조하십시오.

사용자는 다음 명령을 지정하여 속성을 설정할 수 있습니다.

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

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

사용자는 다음 응용 프로그램 프로그래밍 인터페이스(API) 루틴을 지정하여 응용 프로그램 내에서 속성을 설정할 수 있습니다.

- sam\_archive(3)
- sam\_release(3)
- sam\_segment(3)
- sam\_setfa(3)
- sam\_ssum(3)
- sam\_stage(3)

표 3-1은 inode에 나열되는 사용자 속성을 나타낸 것입니다.

표 3-1 사용자 파일 속성

파일 속성	설명
setfa -D	파일이 직접 I/O로 표시됩니다.
setfa -gn	파일이 스트라이프 그룹 <i>n</i> 에 할당되도록 표시합니다.
setfa -sm	파일이 스트라이프 너비 <i>m</i> 으로 할당되도록 표시됩니다.

주 – Sun StorageTek SAM 아카이빙 기능에만 특정한 다수의 추가 사용자 지정 파일 속성이 있습니다. 자세한 내용은 Sun StorageTek Storage Archive Manager 파일 시스템 구성 및 관리 안내서를 참조하십시오.

모든 파일 및 디렉토리에서 표 3-1에 있는 속성을 설정할 수 있습니다. 디렉토리 속성이 설정된 후 디렉토리에 작성되는 파일이 모든 디렉토리 속성을 상속합니다. 상위 디렉토리에 속성이 적용되기 전에 만들어진 파일은 디렉토리 속성을 상속 받지 않습니다.

WORM-FS 패키지가 설치된 경우 WORM(한 번 쓰고 많이 읽음) 속성을 파일에 적용하고 파일의 보유 기간을 설정할 수 있습니다. 자세한 내용은 186페이지의 "WORM-FS 파일 시스템 구성"을 참조하십시오.

## 파일 정보 표시

Sun StorageTek QFS `sls(1)` 명령은 표준 UNIX `ls(1)` 명령을 확장하고 파일에 대한 자세한 내용을 제공합니다. 코드 예 3-1은 `hgc2` 파일에 대한 `inode` 정보를 표시하는 자세한 `sls(1)` 명령 출력을 표시합니다.

코드 예 3-1 SAM-QFS 환경에서의 `sls(1)` 출력

```
# 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
```

### `sls(1)` 출력 정보

표 3-2는 코드 예 3-1에 표시된 `sls(1)` 출력에서 각 행에 대한 의미를 설명한 것입니다.

주 - 아카이빙에 관련된 행은 Sun StorageTek SAM 환경에 있는 `sls(1)` 출력에만 나타납니다.

표 3-2 `sls(1)` 수행 결과 설명

행 번호	태그	내용
1	<code>mode:</code>	파일의 모드 및 권한, 파일에 대한 하드 링크의 수, 파일의 소유자, 소유자가 속한 그룹
2	<code>length:</code>	바이트 단위의 파일 길이, 파일의 <code>admin ID</code> 번호, 파일의 <code>inode</code> 번호 기본적으로 <code>admin ID</code> 번호는 0입니다. 이 번호가 0보다 클 경우에는 파일 및 블록을 계산하기 위한 파일의 계산 범주를 나타냅니다. 이 파일 시스템에서 파일 시스템 할당량이 활성화되어 있지 않더라도 이 번호는 0보다 큰 값으로 설정할 수 있습니다. 파일 시스템 할당량에 대한 자세한 내용은 95페이지의 "파일 시스템 할당량 관리"를 참조하십시오. <code>inode</code> 번호는 <code>inode</code> 번호 자체, 마침표(.), <code>inode</code> 생성 번호의 두 부분으로 구성됩니다.
3	<code>archdone;</code>	파일에 대한 파일 속성. 이 행에 대한 자세한 내용은 <code>sls(1)</code> 매뉴얼 페이지를 참조하십시오.

표 3-2 sls(1) 수행 결과 설명(계속)

행 번호	태그	내용
4	segments	<p>세그먼트 인덱스 정보. 이 행은 파일이 세그먼트 인덱스가 아닌 경우에는 나타나지 않습니다. 이 행의 일반적인 형식은 다음과 같습니다. segments <i>n</i>, offline <i>o</i>, archdone <i>a</i>, damaged <i>d</i>;</p> <ul style="list-style-type: none"> <li>segments <i>n</i>은 이 파일에 대한 데이터 세그먼트의 총 수를 표시합니다. 이 예제에서는 3개가 있습니다.</li> <li>offline <i>o</i>는 오프라인 데이터 세그먼트 수를 표시합니다. 이 예제에서는 오프라인 세그먼트가 없습니다.</li> <li>archdone <i>a</i>는 아카이브 요구 조건을 충족시키는 세그먼트의 수를 표시합니다. 이 예제에서는 3개가 있습니다.</li> <li>damaged <i>d</i>는 손상된 세그먼트의 수를 표시합니다. 이 예제에서는 손상된 세그먼트가 없습니다.</li> </ul>
5, 6	copy 1: , copy 2:	<p>아카이브 복사본 행. sls(1) 명령은 각 활성 또는 만료된 아카이브 복사본에 대한 아카이브 복사본 행 하나를 표시합니다.</p> <p>이 행의 4가지 위치는 다음을 표시합니다.</p> <p>1 - 만료 또는 활성 항목 중 하나입니다.</p> <ul style="list-style-type: none"> <li>S는 아카이브 복사본이 만료되었음을 나타냅니다. 즉, 파일이 수정되었고 이 아카이브 복사본은 이전 버전의 파일입니다.</li> <li>U는 복사본이 아카이브되지 않았음을 나타냅니다. 언아카이브는 파일 또는 디렉토리에 대한 아카이브 항목이 삭제되는 프로세스입니다.</li> <li>대시(-)는 아카이브 복사본이 활성 상태이고 유효함을 나타냅니다.</li> </ul> <p>2 - 아카이브 복사본을 다시 아카이브할 것인지의 여부를 나타냅니다.</p> <ul style="list-style-type: none"> <li>r은 아카이브 복사본이 아카이버에 의해 다시 아카이브될 예정임을 나타냅니다.</li> <li>대시(-)는 아카이브 복사본이 아카이버에 의해 다시 아카이브되지 않음을 나타냅니다.</li> </ul> <p>3 - 사용되지 않음</p> <p>4 - 복사본이 손상되었는지의 여부를 나타냅니다.</p> <ul style="list-style-type: none"> <li>D는 아카이브 복사본이 손상되었음을 나타냅니다. 손상된 아카이브 복사본은 스테이징 대상이 아닙니다.</li> <li>대시(-)는 아카이브 복사본이 손상되지 않았음을 나타냅니다. 이러한 아카이브 복사본은 스테이징 대상입니다.</li> </ul> <p>나머지 아카이브 복사본 행의 형식은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>아카이브 복사본이 아카이브 매체에 기록된 날짜 및 시간</li> <li>소수점(.)으로 구분되는 두 개의 16진수. 첫 번째 16진수(2239a)는 카트리지에서 아카이브 파일의 시작 위치를 나타냅니다. 두 번째 16진수(48)는 아카이브 파일에서 이 복사본의 파일 바이트 오프셋(512로 나눔)을 나타냅니다.</li> <li>아카이브 복사본이 상주하는 매체 유형 및 볼륨 일련 번호(VSN)</li> </ul>

표 3-2 sls(1) 수행 결과 설명(계속)

행 번호	태그	내용
7	access:	파일이 마지막으로 액세스된 시간과 수정된 시간
8	changed:	파일 내용 및 파일 속성이 마지막으로 변경된 시간
9	creation:	파일이 작성되고 파일 시스템에 상주하게 된 시간

## 보유 행 정보

선택적 WORM-FS 패키지를 사용 중인 경우 sls(1) 출력에 보유 행도 표시됩니다. 보유 행의 형식은 다음과 같습니다.

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

이 행은 이 파일에 대한 보유 기간이 설정되었는지, 설정되었다면 길이가 얼마인지를 표시합니다. retention-end 날짜는 보유 기간이 만료되는 날짜를 표시합니다.

WORM-FS 기능의 사용에 대한 자세한 내용은 [186페이지의 "WORM-FS 파일 시스템 구성"](#)을 참조하십시오.

## 시스템에 대한 구성 파일 변경 전과

이 절에서는 시스템 전체로 구성 파일 변경사항을 전과하는 방법을 설명합니다. 이 절차는 다음 파일에 대한 변경 사항 전과를 설명합니다.

- mcf(4)
- defaults.conf
- archiver.cmd(SAM-QFS 파일 시스템만 해당)
- stager.cmd(SAM-QFS 파일 시스템만 해당)
- 공유 호스트 파일(Sun StorageTek QFS 공유 및 SAM-QFS 공유 파일 시스템만 해당)

다음 환경에서 이러한 절차를 수행해야 합니다.

- 정보를 추가, 삭제 또는 수정하기 위해 이 모든 파일을 업데이트하는 경우.
- Sun StorageTek QFS 파일 시스템이 이미 작동한 후 Sun StorageTek SAM archiver.cmd, defaults.conf 또는 stager.cmd 파일을 작성 또는 업데이트 하는 경우

다음 절에서는 이러한 절차에 대해 설명합니다.

- 54페이지의 "Sun StorageTek QFS Sun Cluster 환경에서 mcf 또는 defaults.conf 정보 변경"
- 55페이지의 "SAM-QFS 환경에서 mcf 또는 defaults.conf 파일 시스템 정보 변경"
- 56페이지의 "mcf 또는 defaults.conf 제거 가능한 매체 드라이브 정보 변경"
- 57페이지의 "공유 호스트 파일 변경"
- 57페이지의 "새 항목 추가 또는 기존 항목 변경"
- 58페이지의 "호스트 이름 변경, 항목 순서 변경 또는 항목 삽입"

## ▼ Sun StorageTek QFS Sun Cluster 환경에서 mcf 또는 defaults.conf 정보 변경

Sun Cluster 환경에서 고가용성을 위해 구성된 공유 파일 시스템에 대한 mcf 또는 defaults.conf 정보를 변경하려면 Sun Cluster 시스템의 모든 참여 노드에서 이 절차를 수행하십시오.

1. vi(1) 또는 다른 편집기를 사용하여 파일을 편집하고 파일 시스템 정보를 변경합니다.
2. mcf 파일을 변경하려는 경우 sam-fsd(1M) 명령을 사용하여 mcf 파일의 오류를 점검합니다.

```
# sam-fsd
```

이 명령의 출력에 오류가 나타나면 다음 단계를 진행하기 전에 오류를 수정하십시오.

3. samd(1M) config 명령을 사용하여 mcf(4) 또는 defaults.conf 파일 변경 사항을 전파합니다.

```
# samd config
```

이러한 파일에 대한 자세한 내용은 defaults.conf(4) 또는 mcf(4) 매뉴얼 페이지를 참조하십시오.

## ▼ SAM-QFS 환경에서 mcf 또는 defaults.conf 파일 시스템 정보 변경

1. vi(1) 또는 다른 편집기를 사용하여 파일을 편집하고 파일 시스템 정보를 변경합니다.
2. mcf 파일을 변경하려는 경우 sam-fsd(1M) 명령을 사용하여 mcf 파일의 오류를 점검합니다.

```
# sam-fsd
```

이 명령의 출력에 오류가 나타나면 다음 단계를 진행하기 전에 오류를 수정하십시오.

3. 하나 이상의 파일 시스템과 관련된 정보를 제거 또는 변경하려는 경우 samcmd(1M) aridle 명령을 실행하여 mcf 파일에 정의되어 영향을 받는 각 파일 시스템에 대해 아카이버를 대기 상태로 만듭니다.

이 명령을 다음 형식으로 사용합니다.

```
samcmd aridle fs.fsname
```

*fsname*에 파일 시스템의 이름을 지정합니다.

4. 하나 이상의 드라이브와 관련된 정보를 제거 또는 변경하려는 경우 samcmd(1M) idle 명령을 실행하여 mcf 파일의 영향을 받는 각 드라이브에 지정된 각 장비 서수에 대해 아카이버를 대기 상태로 만듭니다.

이 명령을 다음 형식으로 사용합니다.

```
samcmd idle eq
```

*eq*에는 해당 장치의 장비 서수 번호를 지정합니다.

5. umount(1M) 명령을 실행하여 변경 사항의 영향을 받은 각 파일 시스템을 마운트 해제합니다.

파일 시스템 마운트 해체에 대한 자세한 내용은 62페이지의 "파일 시스템 마운트 해제"를 참조하십시오.

6. samd(1M) config 명령을 사용하여 변경 사항을 전파합니다.

```
# samd config
```

7. mount(1M) 명령을 사용하여 마운트 해제했던 파일 시스템을 다시 마운트합니다.

이러한 파일에 대한 자세한 내용은 defaults.conf(4) 또는 mcf(4) 매뉴얼 페이지를 참조하십시오.

## ▼ mcf 또는 defaults.conf 제거 가능한 매체 드라이브 정보 변경

1. 파일을 편집하고 제거 가능한 매체 드라이브 정보를 변경합니다.
2. mcf 파일을 변경하려는 경우 **sam-fsd(1M)** 명령을 사용하여 mcf 파일의 오류를 점검합니다.

```
# sam-fsd
```

이 명령의 출력에 오류가 나타나면 다음 단계를 진행하기 전에 오류를 수정하십시오.

3. 하나 이상의 파일 시스템과 관련된 정보를 제거 또는 변경하려는 경우 **samcmd(1M)** **aridle** 명령을 실행하여 mcf 파일에 정의되어 영향을 받는 각 파일 시스템에 대해 아카이버를 대기 상태로 만듭니다.

이 명령을 다음 형식으로 사용합니다.

```
samcmd aridle fs.fsname
```

*fsname*에 파일 시스템의 이름을 지정합니다.

4. 하나 이상의 드라이브와 관련된 정보를 제거 또는 변경하려는 경우 mcf 파일의 영향을 받는 각 드라이브에 지정된 각 장비 서수 번호에 대해 **samcmd(1M)** **idle** 명령을 실행합니다.

이 명령을 다음 형식으로 사용합니다.

```
samcmd idle eq
```

*eq*에는 해당 장치의 장비 서수 번호를 지정합니다.

5. **samd(1M)** **stop** 명령을 사용하여 모든 제거 가능한 매체 작업을 중지합니다.

```
# samd stop
```

6. **samd(1M)** **config** 명령을 사용하여 변경 사항을 전파하고 시스템을 재시작합니다.

```
# samd config
```

7. `samd(1M)` `start` 명령을 사용하여 모든 제거 가능한 매체 작업을 다시 시작합니다.

```
# samd start
```

이러한 파일에 대한 자세한 내용은 `defaults.conf(4)` 또는 `mcf(4)` 매뉴얼 페이지를 참조하십시오.

## 공유 호스트 파일 변경

파일 시스템을 마운트 해제하지 않고 공유 호스트 파일에서 새 호스트 항목을 추가하거나 기존 항목을 변경할 수 있습니다. 호스트 이름을 변경하거나, 항목 순서를 변경하거나, 항목을 삽입하려면 먼저 파일 시스템을 마운트 해제해야 합니다. 공유 호스트 파일을 변경하려면 다음 절차를 사용합니다.

### ▼ 새 항목 추가 또는 기존 항목 변경

공유 호스트 파일의 끝에 새 호스트 항목을 추가하거나 공유 호스트 파일에 있는 기존 항목의 2열에서 5열까지를 변경하려면 이 절차를 사용합니다.

1. 메타 데이터 서버로 작동 중인 호스트를 모르는 경우 `samsharefs(1M)` `family-set-name` 명령을 실행하여 메타 데이터 서버 이름을 표시합니다.

파일 시스템이 구성된 임의의 호스트에서 이 명령을 실행합니다.

예를 들어, 다음과 같습니다.

```
# samsharefs sharefs1
```

2. 메타 데이터 서버에서 공유 호스트 파일을 임시 작업 파일에 저장합니다.

예를 들어, 다음과 같습니다.

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

3. (선택 사항) 공유 호스트 파일의 사본을 저장합니다.

예를 들어, 다음과 같습니다.

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

4. `vi(1)` 또는 다른 편집기를 사용하여 임시 작동 파일을 편집합니다.

마운트된 파일 시스템의 경우 파일의 끝에 새 호스트 항목을 추가할 수 있으며 기존 항목의 2열에서 5열까지를 변경할 수 있습니다.

5. 임시 작동 파일을 저장하고 닫습니다.
6. 임시 작동 파일을 SUNWsamfs 디렉토리에 복사합니다.  
예를 들어, 다음과 같습니다.

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

7. 새 공유 호스트 파일을 파일 시스템에 적용합니다.  
예를 들어, 다음과 같습니다.

```
# samsharefs -u sharefs1
```

## ▼ 호스트 이름 변경, 항목 순서 변경 또는 항목 삽입

공유 호스트 파일에서 호스트 이름 변경, 항목 순서 변경 또는 항목 삽입을 위한 절차를 수행하려면 파일 시스템을 마운트 해제해야 합니다.

1. 메타 데이터 서버로 작동 중인 호스트를 모르는 경우 **samsharefs(1M)** `-R family-set-name` 명령을 실행하여 메타 데이터 서버 이름을 표시합니다.  
파일 시스템이 구성된 모든 호스트에서 이 명령을 실행합니다.  
예를 들어, 다음과 같습니다.

```
# samsharefs -R sharefs1
```

2. 참여하는 각 클라이언트 및 메타 데이터 서버의 파일 시스템을 차례로 마운트 해제합니다.
3. 메타 데이터 서버에서 공유 호스트 파일을 임시 작동 파일에 저장합니다.  
예를 들어, 다음과 같습니다.

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

4. (선택 사항) 공유 호스트 파일의 사본을 저장합니다.  
예를 들어, 다음과 같습니다.

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

5. **vi(1)** 또는 다른 편집기를 사용하여 임시 작동 파일을 편집합니다.
6. 공유 호스트 파일을 저장하고 닫습니다.

7. 새 공유 호스트 파일을 SUNWsamfs 디렉토리에 복사합니다.  
예를 들어, 다음과 같습니다.

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

8. 새 공유 호스트 파일을 파일 시스템에 적용합니다.  
예를 들어, 다음과 같습니다.

```
# samsharefs -uR sharefs1
```

9. 메타 데이터 서버와 클라이언트에서 차례로 파일 시스템을 마운트합니다.

---

## 마운트 매개변수 설정

Solaris OS mount(1M) 명령을 사용하여 Sun StorageTek QFS 파일 시스템을 마운트할 수 있습니다.

마운트 매개변수는 파일 시스템의 특성을 조작하는 데 사용됩니다. 마운트 매개변수를 지정하는 여러 가지 방법이 있습니다. 계층 구조의 맨 위에 있는 방법이 계층 구조의 아래에 있는 방법을 대체합니다. 맨 위에서 아래쪽으로 계층 순서로 나열된 다음 방법으로 마운트 옵션을 지정할 수 있습니다.

- mount(1M) 명령에서 명령줄 옵션 사용. Solaris OS mount(1M) 명령에서 지정된 옵션은 /etc/vfstab 파일에서 지정된 다른 옵션, samfs.cmd 파일에서 지정된 명령 및 시스템 기본 설정값보다 우선합니다.
- /etc/vfstab 파일 설정
- samfs.cmd 파일에서 명령어 사용
- 시스템 기본값. 시스템 기본 설정값은 Solaris OS에 대해 미리 정의된 구성 가능한 설정값입니다. samfs.cmd 파일, /etc/vfstab 파일 및 mount(1M) 명령의 스펙으로 시스템 설정을 대체할 수 있습니다.

또한 samu(1M) 운영자 유틸리티나 samcmd(1M) 명령을 사용하여 마운트 옵션을 지정할 수도 있습니다. 이 방법으로 활성화되거나 비활성화된 마운트 옵션은 파일 시스템이 마운트 해제될 때까지 지속됩니다.

다음 하위 절에서는 마운트 옵션 지정 방법을 설명합니다. 특정 마운트 옵션에 대한 내용은 [부록 B, 245페이지의 "Sun StorageTek QFS 공유 파일 시스템의 마운트 옵션"](#)을 참조하십시오. Sun StorageTek QFS 설치 및 업그레이드 안내서에도 파일 시스템 마운트에 대한 내용이 있습니다.

## mount(1M) 명령

Solaris OS mount(1M) 명령을 사용하여 파일 시스템을 마운트하고, /etc/vfstab 파일 및 /etc/opt/SUNWsamfs/samfs.cmd 파일에서 지정된 설정보다 우선하는 설정을 지정할 수 있습니다. 예를 들어 스트라이프 너비, 먼저 읽기, 후에 쓰기, 디스크 캐시 사용률에 대한 High/Low 워터마크 등을 지정할 수 있습니다.

samfs.cmd 파일과 함께 mount(1M) 명령을 사용하는 한 가지 방법은 시스템을 시험하거나 조정할 때 samfs.cmd 파일을 마운트 옵션에 대한 기본 위치로 사용하고 mount(1M) 명령의 옵션을 사용하는 것입니다.

예를 들어 다음 명령은 파일 시스템 qfs1을 /work에 마운트하고 setuid 실행은 허용하지 않으며 qwrite를 활성화합니다. qfs1 파일 시스템 이름은 장비 식별자입니다. 이 파일 시스템에 대한 mcf 파일의 장비 식별자 필드에도 나타납니다. 두 개 이상의 마운트 옵션을 지정하려면 콤마를 사용하여 각각을 분리합니다.

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

Sun StorageTek QFS 또는 SAM-QFS 공유 파일 시스템을 마운트하는 경우 먼저 메타데이터 서버에 파일 시스템을 마운트한 후 참여하는 각 클라이언트 호스트에 파일 시스템을 마운트해야 합니다. mount 명령에 shared 옵션을 포함시키는데, 해당 명령이 메타데이터 서버와 참여 호스트에서 동일함에 주의하십시오.

mount(1M) 명령에 대한 자세한 내용은 mount\_samfs(1M) 매뉴얼 페이지를 참조하십시오.

## /etc/vfstab 파일

mcf 파일에 정의되는 각 Sun StorageTek QFS 파일 시스템은 /etc/vfstab Solaris OS 시스템 파일에 하나의 행을 가져야 합니다. 이 행은 파일 시스템을 마운트하기 위해 필요합니다.

다음은 /etc/vfstab 파일에 있는 파일 시스템 행의 예입니다.

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

왼쪽에서 오른쪽으로 가면서 표시된 필드는 다음을 나타냅니다.

- 파일 시스템 패밀리 세트 이름
- samfsck(1M)에 대한 파일 시스템
- 마운트 지점
- 파일 시스템 유형. Sun StorageTek QFS 파일 시스템의 경우에도 이것은 항상 samfs입니다.

- samfsck(1M) 패스
- 부팅 시 마운트 옵션
- 중간에 삽입된 공백 없이 쉼표로 구분되는 마운트 매개변수

/etc/vfstab 파일의 필드는 공백 또는 탭 문자로 구분되어야 합니다.

마운트 매개변수 필드는 mount\_samfs(1M) 매뉴얼 페이지에 설명된 -o 옵션에 대한 인수로 나열된 모든 마운트 매개변수를 포함할 수 있습니다. 이러한 매개변수는 samfs.cmd 파일의 명령줄 또는 mount(1M) 명령의 -o 옵션에 대한 인수로 지정할 수 있는 매개변수와 거의 동일합니다. samfs.cmd 파일에서와 같이, 다양한 I/O 설정, 먼저 읽기, 후에 쓰기, 스트라이프 너비, 다양한 저장소 및 아카이브 관리 설정, Qwrite 및 기타 기능에 대한 지정을 포함할 수 있습니다.

사용 가능한 마운트 매개변수에 대한 자세한 내용은 mount\_samfs(1M) 매뉴얼 페이지를 참조하십시오. /etc/vfstab 파일 수정에 대한 자세한 내용은 vfstab(4) 매뉴얼 페이지를 참조하십시오.

## samfs.cmd 파일

/etc/opt/SUNWsamfs/samfs.cmd 파일을 사용하면 사용자의 모든 Sun StorageTek QFS 파일 시스템에 대한 마운트 매개변수를 지정할 수 있습니다. 이 파일은 여러 개의 파일 시스템이 구성되어 있고, 이러한 파일 시스템에 대해 동일한 마운트 매개변수를 지정하려는 경우에 유용합니다.

이 파일을 사용하면 한 위치에서 쉽게 읽을 수 있는 형식으로 모든 마운트 매개변수를 정의할 수 있습니다. 이 파일의 시작에 대해 지정된 명령은 전역 명령이며 모든 Sun StorageTek QFS 파일 시스템에 적용됩니다. 이 파일의 두 번째 부분에서는 각 개별 파일 시스템에 적용할 특정 매개변수를 지정할 수 있습니다. 각 파일 시스템에 대해 모든 마운트 매개변수를 지정해야 하는 /etc/vfstab 파일과 이 파일이 다른 점은 공통 매개변수를 한 위치에서만 한 번만 지정하는 기능입니다.

samfs.cmd 파일에 지정할 수 있는 마운트 매개변수는 /etc/vfstab 파일에서 또는 mount(1M) 명령의 -o 옵션에 대한 인수로 지정할 수 있는 매개변수와 거의 동일합니다. 지정할 수 있는 사용 가능한 마운트 매개변수는 I/O 설정, Read-Ahead, Write-Behind, 스트라이프 너비, 다양한 저장소 및 아카이브 관리 설정, WORM-FS, Qwrite 및 기타 기능에 해당됩니다. 이 파일에서 지정할 수 있는 마운트 매개변수에 대한 자세한 내용은 samfs.cmd(4) 매뉴얼 페이지를 참조하십시오.

samfs.cmd 파일에서 명령은 한 행에 하나씩 쓰여집니다. 이 파일은 주석을 포함할 수 있으며, 주석은 우물정자(#)로 시작됩니다. 우물정자 오른쪽에 나타나는 문자는 주석으로 간주됩니다.

모든 파일 시스템에 전역적으로 적용되는 명령의 경우 fs = 행 앞에 해당 행을 위치하십시오. 특정 파일 시스템에 특정한 명령은 fs = 로 행을 시작하고 모든 전역 명령 뒤에 위치시킵니다. 특정 파일 시스템에 사용되는 명령은 전역 명령보다 우선합니다.

코드 예 3-2는 모든 파일 시스템에 대한 디스크 캐시 사용률에 대한 High/Low 위터마크를 설정하고 두 개의 특정 파일 시스템에 대한 개별화된 매개변수를 지정하는 samfs.cmd 파일 예제를 보여줍니다.

코드 예 3-2 samfs.cmd 파일 예제

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

samfs.cmd 파일의 명령은 모든 기본 시스템 설정 값보다 우선하지만, mount(1M) 명령에 대한 인수가 이 파일의 모든 명령보다 우선합니다. /etc/vfstab 파일의 항목도 samfs.cmd 파일에서 지정된 명령보다 우선합니다.

mount(1M) 명령에 대한 자세한 내용은 mount\_samfs(1M) 매뉴얼 페이지를 참조하십시오. samfs.cmd 파일에 입력할 수 있는 명령에 대한 자세한 내용은 samfs.cmd(4) 매뉴얼 페이지를 참조하십시오.

## 파일 시스템 마운트 해제

Solaris OS umount(1M) 명령을 사용하여 Sun StorageTek QFS 파일 시스템을 마운트 해제할 수 있습니다.

### ▼ 독립형 QFS 또는 SAM-QFS 파일 시스템의 마운트 해제

- umount(1M) 명령을 사용하여 파일 시스템을 마운트 해제합니다.

```
# umount /samqfs
```

파일 시스템을 마운트 해제하기 전에 NFS 사용을 위해 공유되는 경우 unshare(1M)할 수도 있습니다. 공유를 해제한 후 두 번째로 마운트 해제 명령을 실행합니다. 이 명령 실행이 실패하여 파일 시스템을 마운트 해제해야 하는 경우 umount(1M) 명령에 -f 옵션을 사용합니다. -f 옵션은 파일 시스템을 강제로 마운트 해제합니다.

마운트 해제 절차에 대한 자세한 내용은 Sun StorageTek QFS 설치 및 업그레이드 안내서를 참조하십시오.

## ▼ Sun StorageTek QFS 공유 파일 시스템 마운트 해제

68페이지의 "공유 파일 시스템 마운트 해제"의 지침을 따르십시오.

### 파일 시스템에 디스크 캐시 추가

파일 시스템에 대한 디스크 캐시를 늘리려면 디스크 분할 영역이나 디스크 드라이브를 추가한 후 `mcf` 파일을 업데이트하고 `samgrowfs(1M)` 명령을 사용하여 파일 시스템을 확장합니다. 파일 시스템을 다시 초기화하거나 복원할 필요는 없습니다.

`mcf` 파일을 변경할 때 다음에 유의하십시오.

- 파일 시스템에 최대 252개의 디스크 분할 영역을 구성할 수 있습니다.
- Sun StorageTek QFS 파일 시스템의 크기를 늘리려면 하나 이상의 새로운 메타 데이터 분할 영역을 추가해야 합니다. 메타 데이터 분할 영역에는 장비 유형 값 `mm`이 필요합니다.
- 메타 데이터 또는 데이터에 대해 새 분할 영역을 추가하려면 기존 디스크 분할 영역 다음으로 `mcf` 파일에 새 분할 영역을 추가합니다.
- `mcf` 파일의 장비 식별자 이름을 변경하지 마십시오. `mcf` 파일에 있는 이름이 슈퍼블록에 있는 이름과 일치하지 않으면 파일 시스템을 더 이상 마운트할 수 없습니다. 대신 다음 메시지가 `/var/adm/messages`에 기록됩니다.

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

## ▼ 파일 시스템에 디스크 캐시 추가

1. `umount(1M)` 명령을 사용하여 확장하려는 파일 시스템을 마운트 해제합니다.

파일 시스템이 공유되는 경우 모든 클라이언트 호스트 및 메타 데이터 서버에서 차례로 파일 시스템을 마운트 해제하십시오. 그런 다음 메타 데이터 서버에서 이 절차의 나머지 단계를 수행할 수 있습니다.

파일 시스템 마운트 해제에 대한 자세한 내용은 62페이지의 "파일 시스템 마운트 해제"를 참조하십시오.

2. 이 절차 수행 중에 파일 시스템의 이름을 변경하려면 `samfsck(1M)` 명령에 `-R` 및 `-F` 옵션을 함께 사용하여 파일 시스템 이름을 바꿉니다.

이 명령에 대한 자세한 내용은 `samfsck(1M)` 매뉴얼 페이지를 참조하십시오.

3. `/etc/opt/SUNWsamfs/mcf` 파일을 편집하여 디스크 캐시를 추가합니다.

4. `sam-fsd(1M)` 명령을 실행하여 `mcf` 파일에서 오류를 검사합니다.

```
# sam-fsd
```

이 명령의 출력에 오류가 나타나면 다음 단계를 진행하기 전에 오류를 수정하십시오.

5. `samd(1M)` `config` 명령을 실행하여 `mcf` 파일 변경 사항을 시스템으로 전파합니다.

```
# samd config
```

자세한 내용은 `samd(1M)` 매뉴얼 페이지를 참조하십시오.

6. 확장될 파일 시스템에 대해 `samgrowfs(1M)` 명령을 실행합니다.  
예를 들어 다음 명령을 입력하여 파일 시스템 `samfs1`을 확장합니다.

```
# samgrowfs samfs1
```

파일 시스템 이름을 변경한 경우, 새로운 이름을 사용하여 `samgrowfs(1M)` 명령을 실행하십시오. 이 명령에 대한 자세한 내용은 `samgrowfs(1M)` 매뉴얼 페이지를 참조하십시오.

7. 파일 시스템을 마운트합니다.

Sun StorageTek QFS 파일 시스템 마운트에 대한 정보는 `mount_samfs(1M)` 매뉴얼 페이지를 참조하십시오.

8. 파일 시스템이 **Sun StorageTek QFS** 공유 파일 시스템인 경우 각 참여 클라이언트 호스트에 있는 `mcf` 파일을 편집하여 메타 데이터 서버의 `mcf` 파일과 일치시킵니다.

---

## 파일 시스템 재작성

다음 중 하나를 수행하기 위해서는 파일 시스템을 다시 작성해야 합니다.

- 디스크 또는 분할 영역 변경
- 디스크 또는 분할 영역 추가
- 디스크 또는 분할 영역 제거

이 절에서 이 절차를 설명합니다.

## ▼ 파일 시스템 백업 및 다시 만들기

1. 모든 사이트별 시스템 파일 및 구성 파일을 백업합니다.  
소프트웨어에 따라 이러한 파일에는 `mcf`, `archiver.cmd`, `defaults.conf`, `samfs.cmd` 또는 `inquiry.conf` 등이 포함될 수 있습니다. Sun StorageTek QFS 환경의 모든 파일 시스템에 대해 이러한 파일을 백업하십시오. `/etc/opt/SUNWsamfs` 디렉토리의 파일, `/var/opt/SUNWsamfs` 디렉토리의 파일 및 공유 호스트 파일에 대한 백업 복사본도 있는지 확인합니다.
2. 수정할 각 파일 시스템이 백업되었는지 확인합니다.  
파일 시스템은 해당 사이트의 정책에 따라 정기적으로 백업되어야 합니다. 파일 시스템에 대한 백업 파일에 문제가 없다면 지금 다시 백업할 필요는 없습니다. 그러나 마지막 덤프 파일이 작성된 후 생성된 정보를 보존하려면 지금 파일 시스템을 백업해야 합니다. `qfsdump`를 사용하여 덤프 파일을 작성하는 방법에 대한 자세한 내용은 Sun StorageTek QFS 설치 및 업그레이드 안내서를 참조하십시오.
3. 파일 시스템을 마운트 해제합니다.  
지침에 대해서는 [62페이지의 "파일 시스템 마운트 해제"](#)를 참조하십시오.
4. 이 절차 수행 중에 파일 시스템의 이름을 변경하려면 `samfsck(1M)` 명령에 `-R` 및 `-F` 옵션을 함께 사용합니다.  
자세한 내용은 `samfsck(1M)` 매뉴얼 페이지를 참조하십시오.
5. `/etc/opt/SUNWsamfs/mcf` 파일을 편집하여 분할 영역을 추가, 변경 또는 제거합니다.  
자세한 내용은 [63페이지의 "파일 시스템에 디스크 캐시 추가"](#)를 참조하십시오.
6. `sam-fsd(1M)` 명령을 입력하여 `mcf` 파일에서 오류를 검사합니다.

```
# sam-fsd
```

이 명령의 출력이 `mcf` 파일에 오류가 있음을 나타내는 경우 다음 단계로 가기 전에 오류를 수정하십시오.

7. `samd(1M) config` 명령을 실행하여 `mcf` 파일 변경 사항을 시스템으로 전파합니다.

```
# samd config
```

자세한 내용은 `samd(1M)` 매뉴얼 페이지를 참조하십시오.

8. `sammkfs(1M)` 명령을 실행하여 파일 시스템을 다시 작성합니다.  
예를 들어 다음 명령은 `samfs10`을 만듭니다.

```
# sammkfs samfs10
```

9. **mount(1M)** 명령을 실행하여 파일 시스템을 마운트합니다.

Sun StorageTek QFS 파일 시스템 마운트에 대한 정보는 **mount\_samfs(1M)** 매뉴얼 페이지를 참조하십시오.

10. **cd(1)** 명령을 입력하여 파일 시스템의 마운트 지점으로 변경합니다.

11. **qfsrestore(1M)** 명령을 사용하거나 **File System Manager**를 사용하여 각 파일을 복구합니다.

갖고 있던 덤프 파일로부터 또는 1단계에서 만든 덤프 파일로부터 복원합니다.

자세한 내용은 **qfsdump(1M)** 매뉴얼 페이지 또는 **File System Manager** 온라인 도움말을 참조하십시오.

# Sun StorageTek QFS 공유 파일 시스템 구성

이 장에서는 Sun StorageTek QFS 공유 파일 시스템을 구성 및 유지하는 방법에 대해 설명합니다. 본 장에는 다음 절이 포함되어 있습니다.

- 67페이지의 "Sun StorageTek QFS 공유 파일 시스템 마운트 및 마운트 해제"
- 69페이지의 "공유되지 않은 파일 시스템을 공유 파일 시스템으로 변환"
- 73페이지의 "공유 파일 시스템을 공유되지 않은 파일 시스템으로 변환"
- 75페이지의 "클라이언트 호스트 추가 또는 제거"
- 88페이지의 "Sun StorageTek QFS 환경에서 메타 데이터 서버 변경"
- 93페이지의 "Sun StorageTek QFS 공유 파일 시스템에서의 클라이언트-서버 통신"

## Sun StorageTek QFS 공유 파일 시스템 마운트 및 마운트 해제

Sun StorageTek QFS 공유 파일 시스템을 마운트 또는 마운트 해제하는 경우 메타 데이터 서버와 클라이언트를 마운트 또는 마운트 해제하는 순서가 중요합니다.

장에 조치를 위해 메타 데이터 서버 및 모든 가능한 메타 데이터 서버에 대한 마운트 옵션이 동일해야 합니다. 예를 들어 마운트 옵션이 포함된 `samfs.cmd` 파일을 만들고 이것을 모든 호스트에 복사할 수 있습니다.

Sun StorageTek QFS 공유 파일 시스템 마운트에 대한 자세한 내용은 245페이지의 "Sun StorageTek QFS 공유 파일 시스템의 마운트 옵션"을 참조하고 `mount_samfs(1M)` 매뉴얼 페이지를 참조하십시오. 파일 시스템의 마운트 및 마운트 해제에 대한 자세한 정보는 3장, 49페이지의 "작업 수행"을 참조하십시오.

## ▼ 공유 파일 시스템 마운트

1. 메타 데이터 서버 및 모든 클라이언트 호스트에서 수퍼유저가 되십시오.
2. `mount(1M)` 명령을 사용하여 메타 데이터 서버를 마운트합니다.  
모든 클라이언트 호스트에서 파일 시스템을 마운트하기 전에 메타 데이터 서버에서 파일 시스템을 마운트합니다.
3. `mount(1M)` 명령을 사용하여 클라이언트 호스트를 마운트합니다.  
원하는 순서대로 클라이언트 호스트에 파일 시스템을 마운트할 수 있습니다.  
`mount(1M)` 명령에 대한 자세한 내용은 `mount(1M)` 매뉴얼 페이지를 참조하십시오.

## ▼ 공유 파일 시스템 마운트 해제

1. `umount(1M)` 명령을 사용하여 모든 참여 클라이언트 호스트에서 파일 시스템을 마운트 해제합니다.  
예를 들어, 다음과 같습니다.

```
client# umount /samqfs
```

필요할 경우, `-f` 옵션을 `umount(1M)` 명령으로 사용합니다. `-f` 옵션은 파일 시스템을 강제로 마운트 해제합니다.

---

주 - 파일 시스템이 메타 데이터 서버에 마운트되지 않은 경우 공유 클라이언트의 강제 마운트 해제는 완료되지 않습니다.

---

2. 메타 데이터 서버의 파일 시스템을 마운트 해제합니다.

```
metaserver# umount /samqfs
```

마운트 해제 시에 파일 시스템에 여러 조건이 존재할 수 있으므로 `umount(1M)` 명령을 두 번 실행해야 합니다. 그래도 파일 시스템이 마운트 해제되지 않으면 `umount(1M)` 명령에 `unshare(1M)`, `fuser(1M)` 또는 기타 명령과 함께 사용하십시오. 마운트 해제 절차에 대한 자세한 내용은 `umount(1M)` 매뉴얼 페이지와 Sun StorageTek QFS 설치 및 업그레이드 안내서를 참조하십시오.

umount 명령과 `-o await_clients#` 플래그를 사용할 수도 있습니다. 그러면 마운트 해제 프로세스는 클라이언트가 마운트를 해제할 때까지 지정된 초(#) 동안 대기합니다. 대기 시간이 종료되거나 모든 클라이언트가 마운트를 해제하는 즉시 마운트 해제가 진행됩니다. 이 인수가 비공유 파일 시스템에 지정된 경우나 호스트가 공유 파일 시스템의 메타 데이터 서버가 아닌 경우 옵션은 무시됩니다.

이 플래그는 `-f` 옵션과 함께 사용될 수도 있습니다. 이 경우 소프트웨어는 마운트 해제를 실행하기 전에 지정된 시간 동안 대기합니다.

---

## 공유되지 않은 파일 시스템을 공유 파일 시스템으로 변환

Sun StorageTek QFS 공유 파일 시스템에 대한 초기 설치 및 구성을 수행하려면 Sun StorageTek QFS 설치 및 업그레이드 안내서의 모든 지침을 따르십시오. 이 장의 많은 예는 해당 안내서에서 설명한 호스트 이름과 구성 정보를 사용합니다.

공유 해제된 Sun StorageTek QFS 파일 시스템을 Sun StorageTek QFS 공유 파일 시스템으로 변환하려면 먼저 메타 데이터 서버에서 변환을 수행한 다음 각 클라이언트에 변환을 수행해야 합니다. 이 절에서 해당 절차를 설명합니다.

### ▼ 메타 데이터 서버에 대한 변환 수행

이 절차의 단계를 완료하려면 root 권한이 있어야 합니다.

1. 기본 메타 데이터 서버로 사용되는 시스템에 수퍼유저로 로그인합니다.
2. 모든 사이트별 시스템 파일 및 구성 파일을 백업합니다.

소프트웨어에 따라 이러한 파일에는 `mcf`, `archiver.cmd`, `defaults.conf`, `samfs.cmd` 또는 `inquiry.conf` 등이 포함될 수 있습니다. 모든 파일 시스템에 대해 이 파일들을 백업하십시오. `/etc/opt/SUNWsamfs` 디렉토리의 파일과 `/var/opt/SUNWsamfs` 디렉토리의 파일에 대한 백업 복사본이 있는지 확인하십시오.

3. 수정할 각 파일 시스템이 백업되었는지 확인합니다.

파일 시스템은 해당 사이트의 정책에 따라 정기적으로 백업되어야 합니다. 파일 시스템에 대한 백업 파일에 문제가 없다면 지금 다시 백업할 필요는 없습니다.

4. `umount(1M)` 명령을 사용하여 파일 시스템을 마운트 해제합니다.

지침에 대해서는 [62페이지의 "파일 시스템 마운트 해제"](#)를 참조하십시오.

5. `samfsck(1M) -S -F family-set-name` 명령을 사용하여 파일 시스템을 **Sun StorageTek QFS** 공유 파일 시스템으로 변환하십시오

`family-set-name`에 대해 새로운 Sun StorageTek QFS 공유 파일 시스템으로 변환하려는 파일 시스템의 패밀리 세트 이름을 지정하십시오. 예를 들어, 다음과 같습니다.

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

6. `/etc/opt/SUNWsamfs/mcf` 파일을 편집하여 파일 시스템의 **Additional Parameters** 필드에 `shared` 키워드를 추가합니다.

예를 들어, 다음과 같습니다.

코드 예 4-1            공유 파일 시스템 sharefs1에 대한 mcf 파일

```
# 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` 파일을 편집하여 파일 시스템의 **Mount Parameters** 필드에 `shared` 키워드를 추가합니다.

예를 들어, 다음과 같습니다.

코드 예 4-2            `/etc/vfstab` 파일 예제

```
# 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* 호스트 구성 파일을 작성합니다.  
예를 들어, 다음과 같습니다.

코드 예 4-3 Sun StorageTek QFS 공유 파일 시스템 호스트 파일 예제

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

호스트 구성 파일 작성에 대한 자세한 내용은 Sun StorageTek QFS 설치 및 업그레이드 안내서를 참조하십시오.

9. **amsharefs(1M)** `-u -R family-set-name` 명령을 실행하여 파일 시스템 및 호스트 구성을 초기화합니다.

예를 들어, 다음과 같습니다.

```
# amsharefs -u -R sharefs1
```

---

주 - 이 명령으로 오류 메시지가 나타날 수 있으나, 무시할 수 있습니다.

---

10. **samd(1M)** `config command` 명령을 실행합니다.

```
# samd config
```

이 명령은 `sam-fsd` 데몬에 구성 변경 사항을 알립니다.

11. **mount(1M)** 명령을 실행하여 파일 시스템을 마운트합니다.

## ▼ 각 클라이언트에서 변환 수행

1. **mkdir(1M)** 명령을 사용하여 파일 시스템용 마운트 지점을 만듭니다.

예를 들어, 다음과 같습니다.

```
# mkdir /sharefs1
```

2. (선택 사항) /etc/opt/SUNWsamfs/hosts.*file-system-name*.local 로컬 호스트 구성 파일을 작성합니다.

Sun StorageTek QFS 공유 호스트 시스템이 여러 호스트 인터페이스를 가지고 있는 경우 이 단계를 수행해야 할 경우도 있습니다. 로컬 호스트 구성 파일이 메타 데이터 서버와 클라이언트 호스트가 파일 시스템에 액세스할 때 사용할 수 있는 호스트 인터페이스를 정의합니다. 이 파일을 사용하여 파일 시스템 트래픽이 해당 환경의 공용 및 개인 네트워크에서 처리되는 방식을 지정합니다.

코드 예 4-4에서 예제 로컬 호스트 구성 파일을 보여줍니다.

코드 예 4-4 hosts.sharefs1.local 파일

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

로컬 호스트 파일을 만드는 방법에 대한 자세한 내용은 85페이지의 "로컬 호스트 구성 파일 작성"을 참조하십시오.

3. 기존 Sun StorageTek QFS 파일 시스템에서 새 Sun StorageTek QFS 공유 파일 시스템으로 파일을 옮기는 경우 수정될 각 파일 시스템을 백업해야 합니다.

파일 시스템은 해당 사이트의 정책에 따라 정기적으로 백업되어야 합니다. 파일 시스템에 대한 백업 파일에 문제가 없다면 지금 다시 백업할 필요는 없습니다.

4. umount(1M) 명령을 사용하여 파일 시스템을 마운트 해제합니다.

지침에 대해서는 62페이지의 "파일 시스템 마운트 해제"를 참조하십시오.

5. /etc/vfstab 파일을 편집하여 파일 시스템의 Mount Parameters 필드에 shared 키워드를 추가합니다.

예를 들어, 다음과 같습니다.

```
# 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* 호스트 구성 파일을 작성합니다.  
코드 예 4-5는 예제를 나타냅니다.

코드 예 4-5 Sun StorageTek QFS 공유 파일 시스템 호스트 파일 예제

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

호스트 구성 파일을 만드는 방법에 대한 자세한 내용은 Sun StorageTek QFS 설치 및 업그레이드 안내서를 참조하십시오.

---

## 공유 파일 시스템을 공유되지 않은 파일 시스템으로 변환

Sun StorageTek QFS 공유 파일 시스템을 공유 해제된 Sun StorageTek QFS 파일 시스템으로 변환하려면 먼저 각 클라이언트에서 변환을 수행한 다음 메타 데이터 서버에서 변환을 수행해야 합니다. 이 절에서 해당 절차를 설명합니다.

### ▼ 각 클라이언트에서 변환 수행

1. `umount(1M)` 명령을 사용하여 파일 시스템을 마운트 해제합니다.  
지침에 대해서는 [62페이지의 "파일 시스템 마운트 해제"](#)를 참조하십시오.
2. /etc/opt/SUNWsamfs/mcf 파일에서 파일 시스템 항목을 삭제합니다.
3. /etc/vfstab 파일에서 파일 시스템 항목을 삭제합니다.
4. `samd(1M) config` 명령을 실행합니다.

```
# samd config
```

이 명령은 `sam-fsd` 때문에 구성 변경 사항을 알립니다.

5. 파일 시스템에 대한 마운트 지점을 삭제합니다.

## ▼ 서버에서 변환 수행

이 절차의 단계를 완료하려면 root 권한이 있어야 합니다.

1. 슈퍼유저로서 메타 데이터 서버 시스템에 로그인합니다.

2. 모든 사이트별 시스템 파일 및 구성 파일을 백업합니다.

소프트웨어에 따라, 이러한 파일에는 mcf(4), archiver.cmd, defaults.conf, samfs.cmd, inquiry.conf 등이 포함될 수 있습니다. 모든 파일 시스템에 대해 이 파일들을 백업하십시오. /etc/opt/SUNWsamfs 디렉토리의 파일과 /var/opt/SUNWsamfs 디렉토리의 파일에 대한 백업 복사본도 있는지 확인하십시오.

3. 기존 **Sun StorageTek QFS** 공유 파일 시스템에서 새 **Sun StorageTek QFS** 파일 시스템으로 파일을 옮기는 경우 수정될 각 파일 시스템을 백업해야 합니다.

파일 시스템은 해당 사이트의 정책에 따라 정기적으로 백업되어야 합니다. 이 작업은 설치 절차의 마지막 단계로 설명되어 있습니다. 파일 시스템에 대한 백업 파일에 문제가 없다면 지금 다시 백업할 필요는 없습니다.

4. umount(1M) 명령을 사용하여 파일 시스템을 마운트 해제합니다.

지침에 대해서는 [62페이지의 "파일 시스템 마운트 해제"](#)를 참조하십시오.

5. samfsck(1M) -F -U file-system-name을 실행하여 **Sun StorageTek QFS** 공유 파일 시스템을 공유 해제된 파일 시스템으로 변환합니다.

file-system-name에 대해 새로운 공유 해제된 파일 시스템으로 변환하는 Sun StorageTek QFS 공유 파일 시스템의 이름을 지정합니다. 예를 들어, 다음과 같습니다.

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

6. /etc/opt/SUNWsamfs/mcf 파일을 편집하여 파일 시스템의 **Additional Parameters** 필드에서 shared 키워드를 제거합니다.

예를 들어, 다음과 같습니다.

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

7. /etc/vfstab 파일을 편집하여 파일 시스템의 **Mount Parameters** 필드에서 shared 키워드를 제거합니다.

예를 들어, 다음과 같습니다.

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

8. /etc/opt/SUNWsamfs/hosts.*file-system-name* 구성 파일을 삭제합니다.
9. **samd(1M)** config 명령을 실행합니다.

```
# samd config
```

이 명령은 sam-fsd 때문에 구성 변경 사항을 알립니다.

10. **mount(1M)** 명령을 실행하여 파일 시스템을 마운트합니다.

---

## 클라이언트 호스트 추가 또는 제거

다음 하위 절에서는 Sun StorageTek QFS 공유 파일 시스템에 있는 클라이언트 호스트 시스템을 추가 및 제거하는 방법에 대한 지침을 제공합니다.

- 76페이지의 "[클라이언트 호스트 추가](#)"
- 80페이지의 "[클라이언트 호스트 제거](#)"
- 82페이지의 "[Sun StorageTek QFS 공유 환경에서 mcf 파일 업데이트](#)"
- 85페이지의 "[로컬 호스트 구성 파일 작성](#)"

## ▼ 클라이언트 호스트 추가

모든 해당 호스트에서 파일 시스템을 구성되고 마운트한 후 Sun StorageTek QFS 공유 파일 시스템에 클라이언트 호스트를 추가할 수 있습니다. Sun Cluster 환경의 노드인 클라이언트 호스트를 추가하는 경우 클러스터의 기존 자원 그룹에 해당 노드를 추가해야 합니다. 자세한 내용은 Sun Cluster System Administration Guide for Solaris OS를 참조하십시오.

1. 메타 데이터 서버에서 슈퍼유저가 됩니다.
2. `samsharefs(1M)` 명령을 사용하여 현재의 Sun StorageTek QFS 공유 파일 시스템 정보를 얻고 이것을 편집 가능한 파일에 씁니다.
  - Sun StorageTek QFS 공유 파일 시스템이 마운트되었으면 현재의 메타 데이터 서버에서 `samsharefs(1M)` 명령을 실행합니다. 예를 들어, 다음과 같습니다.

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

- Sun StorageTek QFS 공유 파일 시스템이 마운트 해제되는 경우 메타 데이터 서버 또는 모든 가능한 메타 데이터 서버에서 `samsharefs(1M)` 명령을 `-R` 옵션과 함께 실행하십시오. 예를 들어, 다음과 같습니다.

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

`samsharefs(1M)` 명령은 활성 메타 데이터 서버 또는 가능한 메타 데이터 서버로 구성된 클라이언트 호스트에서만 실행할 수 있습니다. 자세한 내용은 `samsharefs(1M)` 매뉴얼 페이지를 참조하십시오.

3. `vi(1)` 또는 다른 편집기를 사용하여 Sun StorageTek QFS 공유 파일 시스템 정보 파일을 엽니다.

코드 예 4-6은 이 단계를 표시합니다.

코드 예 4-6            편집 전 `hosts.sharefs1`

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

4. 편집기를 사용하여 새 클라이언트 호스트에 대한 행을 추가합니다.

코드 예 4-7은 helene에 대한 행이 마지막 행으로 추가된 후의 파일을 나타낸 것입니다.

코드 예 4-7 편집 후 hosts.sharefs1

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

5. samsharefs(1M) 명령을 사용하여 이진 파일에 현재의 정보를 업데이트합니다.

이 명령에서 사용되는 옵션 및 이 명령이 실행되는 시스템은 다음과 같이 Sun StorageTek QFS 공유 파일 시스템이 마운트되었는지 여부에 따라 다릅니다.

- 파일 시스템이 마운트된 경우 현재 메타 데이터 서버에서 samsharefs(1M) -u 명령을 실행합니다. 예를 들어, 다음과 같습니다.

```
# samsharefs -u sharefs1
```

- 파일 시스템이 마운트 해제된 경우 활성 메타 데이터 서버 또는 모든 가능한 메타 데이터 서버에서 samsharefs(1M) -R -u 명령을 실행합니다. 예를 들어, 다음과 같습니다.

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

이제 클라이언트 호스트 helene이 인식됩니다.

6. 슈퍼유저로서 추가하려는 클라이언트 호스트에 로그인합니다.

7. format(1M) 명령을 사용하여 클라이언트 호스트 디스크가 있는지 확인합니다.

8. 클라이언트 호스트의 mcf 파일을 업데이트합니다.

호스트 시스템이 공유 파일 시스템을 액세스 또는 마운트할 수 있으려면, 호스트 시스템의 mcf 파일에 정의된 해당 파일 시스템이 있어야 합니다. mcf 파일은 Sun StorageTek QFS 공유 파일 시스템의 모든 클라이언트 호스트와 일치하도록 업데이트해야 합니다. 파일 시스템 및 디스크 선언 정보는 Family Set Name, Equipment Ordinal 및 Equipment Type에 대해 메타 데이터 서버의 구성과 동일한 데이터를 포함해야 합니다. 클라이언트 호스트의 mcf 파일에는 shared 키워드도 포함되어야 합니다. 그러나 제어기 지정은 호스트마다 다를 수 있으므로 장치 이름을 변경할 수 있습니다.

mcf 파일 편집 방법에 대한 내용은 82페이지의 "Sun StorageTek QFS 공유 환경에서 mcf 파일 업데이트"를 참조하십시오.

9. 메타 데이터 서버 호스트에서 `samd(1M) config` 명령을 실행합니다.

```
# samd config
```

이 명령은 `sam-fsd` 데몬에 구성 변경 사항을 알립니다.

10. (선택 사항) 새 클라이언트 호스트에 대한 로컬 호스트 구성 파일을 만듭니다.

Sun StorageTek QFS 공유 파일 시스템이 여러 호스트 인터페이스를 가지고 있는 경우 이 단계를 수행해야 할 경우도 있습니다. 로컬 호스트 구성 파일이 메타 데이터 서버와 클라이언트 호스트가 파일 시스템에 액세스할 때 사용할 수 있는 호스트 인터페이스를 정의합니다. 이 파일을 사용하여 파일 시스템 트래픽이 해당 환경의 공용 및 개인 네트워크에서 처리되는 방식을 지정합니다.

로컬 호스트 파일을 만드는 방법에 대한 내용은 85페이지의 "로컬 호스트 구성 파일 작성"을 참조하십시오.

11. 클라이언트 호스트에서 `samd(1M) config` 명령을 실행합니다.

```
# samd config
```

이 명령은 `sam-fsd` 데몬에 구성 변경 사항을 알립니다.

12. `sam-sharefsd` 데몬이 파일 시스템에 대해 실행 중인지 확인합니다.

이 작업을 마치려면 코드 예 4-8에 표시된 대로 `ps(1)` 및 `grep(1)` 명령을 사용합니다.

코드 예 4-8 `ps(1)` 명령 출력

```
# 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
```

코드 예 4-8은 `sam-sharefsd` 데몬이 `sharefs1` 파일 시스템에 대해 활성화되어 있음을 나타냅니다. 시스템에서 반환한 출력 결과에 `sam-sharefsd` 데몬이 Sun StorageTek QFS 공유 파일 시스템에 대해 활성화되어 있지 않다고 나타나면 225페이지의 "공유 파일 시스템에서 실패하거나 멈춘 `sammkfs(1M)` 또는 `mount(1M)` 명령 문제 해결"에 설명된 진단 절차를 수행합니다.

13. 새 **Sun StorageTek QFS** 공유 파일 시스템에 마운트 지점이 아직 없는 경우 **mkdir(1)** 명령을 사용하여 마운트 지점에 대한 디렉토리를 만듭니다.

예를 들어, 다음과 같습니다.

```
# mkdir /sharefs1
```

14. **chmod(1M)** 명령을 사용하여 마운트 지점에 755 권한 세트를 부여합니다.

예를 들어, 다음과 같습니다.

```
# chmod 755 /sharefs1
```

권한은 모든 참여 호스트에 대해 동일해야 합니다. 755는 마운트된 후 파일 시스템을 사용할 수 있기 위해서 사용자가 마운트 지점에 대해 실행 권한을 가져야 하기 때문에 초기 권한 설정으로 권장됩니다. 파일 시스템을 마운트한 후 **root** 디렉토리의 권한이 이 설정을 무효화합니다.

15. **/etc/vfstab** 파일을 수정합니다.

**/etc/vfstab** 파일에 **Sun StorageTek QFS** 공유 파일 시스템에 대한 항목이 있어야 합니다. **Mount Parameters** 필드에 **shared**를 지정합니다. 또한, 다음 중 하나의 작업을 수행합니다.

- 부팅 시 이 파일 시스템이 자동으로 부팅되지 않도록 하려면 **Mt@boot** 필드에 **no**를 입력하십시오.
- **Sun StorageTek QFS** 공유 파일 시스템이 부트할 때 자동으로 마운트되도록 하려면 다음을 수행합니다.
  - **Mt@boot** 필드에 **yes**를 입력합니다.
  - **Mt params** 필드에 **bg** 마운트 옵션을 추가합니다. **bg** 마운트 옵션은 메타 데이터 서버가 응답하지 않는 경우 파일 시스템을 백그라운드에서 마운트합니다.

**코드 예 4-9**는 **Mt params** 필드의 **shared** 및 **bg** 항목을 나타냅니다.

**코드 예 4-9**                    **/etc/vfstab** 파일 예제

```
# 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. 메타 데이터 서버에서 **df(1M)** 명령을 실행하여 파일 시스템이 메타 데이터 서버에 마운트되었는지 확인합니다.

예를 들어, 다음과 같습니다.

```
# df -k
```

파일 시스템이 표시된 목록에 포함되어 있어야 합니다.

17. 클라이언트 호스트에서 **mount(1M)** 명령을 실행하여 **Sun StorageTek QFS** 공유 파일 시스템을 마운트합니다.

예를 들어, 다음과 같습니다.

```
# mount /sharefs1
```

Sun StorageTek QFS 공유 파일 시스템 마운트에 대한 자세한 내용은 [245페이지의 "Sun StorageTek QFS 공유 파일 시스템의 마운트 옵션"](#)을 참조하거나 `mount_samfs(1M)` 매뉴얼 페이지를 참조하십시오.

## ▼ 클라이언트 호스트 제거

1. 메타 데이터 서버 및 모든 클라이언트 호스트에서 슈퍼유저가 되십시오.

---

주 - `samsharefs(1M)` 명령을 사용하여 메타 데이터 서버 또는 클라이언트 호스트에 로그인했는지 정확하게 확인할 수 있습니다.

---

2. `umount(1M)` 명령을 사용하여 **Sun StorageTek QFS** 공유 파일 시스템이 마운트된 각 클라이언트 호스트에서 **Sun StorageTek QFS** 공유 파일 시스템을 마운트 해제합니다.  
예를 들어, 다음과 같습니다.

```
client# umount sharefs1
```

3. `umount(1M)` 명령을 사용하여 메타 데이터 서버에서 **Sun StorageTek QFS** 공유 파일 시스템을 마운트 해제하십시오.

예를 들어, 다음과 같습니다.

```
metaserver# umount sharefs1
```

4. 아직 하지 않은 경우, **Sun StorageTek QFS** 공유 파일 시스템에 대해 메타 데이터 서버에 슈퍼유저로 로그인합니다.
5. **sharefs(1M)** 명령을 사용하여 현재의 구성 정보를 얻습니다.  
다음 예제 명령은 현재의 구성 정보를 `/etc/opt/SUNWsamfs/hosts.sharefs1` 파일에 씁니다.

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

6. **vi(1)** 또는 다른 편집기를 사용하여 **Sun StorageTek QFS** 공유 파일 시스템 정보 파일을 엽니다.  
**코드 예 4-10**은 클라이언트 호스트가 삭제되기 전 파일을 나타냅니다.

**코드 예 4-10**                    클라이언트 호스트 삭제 전 `hosts.sharefs1`

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

7. 편집기를 사용하여 더 이상 지원하지 않을 클라이언트 호스트 또는 호스트들을 삭제합니다.  
**코드 예 4-11**은 `helene`에 대한 행이 삭제된 후의 파일을 나타낸 것입니다.

**코드 예 4-11**                    클라이언트 호스트 삭제 후 `hosts.sharefs1`

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

8. `samsharefs(1M) -R -u` 명령을 사용하여 현재의 호스트 정보를 업데이트합니다.  
예를 들어, 다음과 같습니다.

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

호스트 `helene`이 제거되었습니다.

9. `samsharefs(1M) -R` 명령을 사용하여 현재의 구성을 표시합니다.  
예를 들어, 다음과 같습니다.

```
# samsharefs -R sharefs1
```

10. `mount(1M)` 명령을 사용하여 먼저 메타데이터 서버에 **Sun StorageTek QFS** 공유 파일 시스템을 마운트한 후 파일 시스템의 각 클라이언트 호스트에 마운트합니다.  
`mount(1M)` 명령에 대한 자세한 내용은 `mount_samfs(1M)` 매뉴얼 페이지를 참조하십시오.

## Sun StorageTek QFS 공유 환경에서 mcf 파일 업데이트

`samfsconfig(1M)` 명령은 Sun StorageTek QFS 공유 파일 시스템에 포함된 장치를 식별하는 데 도움이 될 수 있는 구성 정보를 생성합니다. 그런 다음 이 정보를 사용하여 각 클라이언트 호스트에 있는 `mcf` 파일을 업데이트합니다.

각 클라이언트 호스트에 개별적인 `samfsconfig(1M)` 명령을 입력합니다. 컨트롤러 번호는 각 클라이언트 호스트에 의해 할당되므로 컨트롤러 번호는 메타 데이터 서버의 컨트롤러 번호와 다를 수 있습니다.

---

주 – Sun StorageTek QFS 공유 파일 시스템이 마운트된 후 메타 데이터 서버의 `mcf` 파일을 업데이트하는 경우 해당 공유 파일 시스템에 액세스할 수 있는 모든 호스트에서 `mcf` 파일을 업데이트해야 합니다.

---

예제 1. 코드 예 4-12는 tethys 클라이언트에서 패밀리 세트 sharefs1에 대한 장치 정보를 얻기 위해 samfsconfig(1M) 명령이 어떻게 사용되는지 보여줍니다. tethys는 가능한 메타 데이터 서버이므로 공유 파일 시스템의 다른 메타 데이터 서버인 titan과 동일한 메타 데이터 디스크에 연결되어 있습니다.

코드 예 4-12 tethys에 대한 samfsconfig(1M) 명령 예제

```
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
```

samfsconfig(1M) 명령 출력에서 마지막 다섯 개 행을 클라이언트 호스트 tethys의 mcf 파일에 복사하여 클라이언트 호스트 tethys의 mcf 파일을 편집합니다. 다음 사항을 확인하십시오.

- 각 Device State 필드는 on으로 설정됩니다.
- shared 키워드가 파일 시스템 이름의 Additional Parameters 필드에 나타납니다.

코드 예 4-13은 mcf 파일의 결과를 보여줍니다.

코드 예 4-13 sharefs1 클라이언트 호스트 tethys에 대한 mcf 파일

```
# 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
```

예제 2. 코드 예 4-14는 클라이언트 호스트 mimas에서 패밀리 세트 sharefs1에 대한 장치 정보를 얻기 위해 samfsconfig(1M) 명령이 어떻게 사용되는지 보여줍니다. 이 예제에서 mimas는 메타 데이터 서버가 될 수 없으며 메타 데이터 디스크에 연결되어 있지 않습니다.

코드 예 4-14 mimas에 대한 samfsconfig(1M) 명령 예제

```
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
```

mimas에 대한 samfsconfig(1M) 명령 수행 결과에서 메타 데이터 디스크인 Ordinal 0이 없다는 것을 알 수 있습니다. 장치가 없는 경우 samfsconfig(1M) 프로세스는 파일 시스템의 요소를 주석 처리하고 파일 시스템 패밀리 세트 선언 행을 생략합니다. 다음과 같이 mcf 파일에 대해 다음 편집 작업 유형을 수행하십시오.

- 클라이언트 호스트 mimas에 대한 mcf 파일에서 sharefs1으로 시작되는 파일 시스템의 패밀리 세트 선언 행을 작성합니다. 파일 시스템의 Family Set 선언 행의 Additional Parameter 필드에 shared 키워드를 입력합니다.
- 누락된 각 Equipment Ordinal 항목에 대해 하나 이상의 nodev 행을 작성합니다. 이러한 각 행에서 액세스할 수 없는 장치에 대한 Equipment Identifier 필드에 키워드 nodev가 나타나야 합니다.
- 각 Device State 필드를 on으로 설정합니다.
- 장치 행에 대한 주석 처리를 해제합니다.

코드 예 4-15는 mimas에 대한 mcf 파일의 결과를 보여줍니다.

코드 예 4-15 클라이언트 호스트 mimas에 대한 mcf 파일

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

# 로컬 호스트 구성 파일 작성

로컬 호스트 구성 파일은 다음 위치에 있어야 합니다.

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

로컬 호스트 구성 파일에서 주석이 허용됩니다. 주석 행은 우물정자(#)로 시작되어야 합니다. 우물정자 오른쪽에 있는 문자는 무시됩니다.

표 4-1은 로컬 호스트 구성 파일의 필드에 대한 설명입니다.

표 4-1 로컬 호스트 구성 파일 필드

필드	내용
Host Name	이 필드는 Sun StorageTek QFS 공유 파일 시스템의 일부인 메타 데이터 서버 또는 가능한 메타 데이터 서버의 영숫자 이름을 포함해야 합니다.
호스트 인터페이스	이 필드는 콤마로 구분된 호스트 인터페이스 주소 목록을 포함해야 합니다. 이 필드는 ifconfig(1M) -a 명령을 수행한 출력 결과에서 만들 수 있습니다. 각 인터페이스는 다음 중 하나의 방법으로 지정할 수 있습니다. <ul style="list-style-type: none"><li>• 10진수와 점으로 이루어진 IP 주소 형식</li><li>• IP 버전 6개의 16진수 주소 형식</li><li>• 로컬 DNS(도메인 이름 서비스)가 특정 호스트 인터페이스로 변환할 수 있는 심볼 이름</li></ul> 각 호스트는 이 필드를 사용하여 지정된 호스트 인터페이스에 연결을 시도할지 여부를 결정합니다. 시스템은 왼쪽에서 오른쪽으로 주소를 검사하며 공유 시스템 호스트 파일에 포함되어 있는 목록의 첫 번째 응답 주소를 사용하여 연결이 이루어집니다.

Sun StorageTek QFS 공유 파일 시스템에서 각 클라이언트 호스트는 메타 데이터 서버 호스트에서 메타 데이터 서버 IP 주소의 목록을 얻습니다.

메타 데이터 서버 및 클라이언트 호스트는 메타 데이터 서버의 /etc/opt/SUNWsamfs/hosts.fsname 파일과 각 클라이언트 호스트의 hosts.fsname.local 파일(있는 경우)을 모두 사용하여 파일 시스템에 액세스할 때 사용할 호스트 인터페이스를 결정합니다. 이 프로세스는 다음과 같습니다(network client 의 client는 클라이언트 호스트와 메타 데이터 서버를 모두 가리킴).

1. 클라이언트는 파일 시스템의 온 디스크 호스트 파일에서 메타 데이터 서버 호스트 IP 인터페이스 목록을 얻습니다.  
이 파일을 검토하려면 메타 데이터 서버 또는 가능한 메타 데이터 서버에서 samsharefs(1M) 명령을 실행하십시오.
2. 클라이언트는 hosts.fs\_name.local에 대해 해당 파일을 검색합니다.

3. 검색 결과에 따라 다음 중 하나의 작업을 수행해야 합니다.

- `hosts.fsname.local` 파일이 없는 경우, 클라이언트는 성공할 때까지 시스템 호스트 구성 파일의 각 주소에 차례로 연결을 시도합니다.
- `hosts.fsname.local` 파일이 있으면 클라이언트는 다음 작업을 수행합니다.
  - a. 메타 데이터 서버의 `/etc/opt/SUNWsamfs/hosts.fsname` 파일과 `hosts.fsname.local` 파일 모두에서 메타 데이터 서버에 대한 주소 목록을 비교합니다.
  - b. 두 위치에 존재하는 주소 목록을 작성한 후 성공할 때까지 이러한 각 주소에 차례로 연결을 시도합니다. 이러한 파일의 주소 순서가 다른 경우 클라이언트는 `hosts.fsname.local` 파일에 있는 순서를 사용합니다.

예제. 코드 예 4-16은 4개의 호스트를 나열한 예제 호스트 파일을 나타냅니다.

코드 예 4-16                      Sun StorageTek QFS 공유 파일 시스템 호스트 파일 예제

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

그림 4-1은 이러한 시스템에 대한 인터페이스를 보여줍니다.

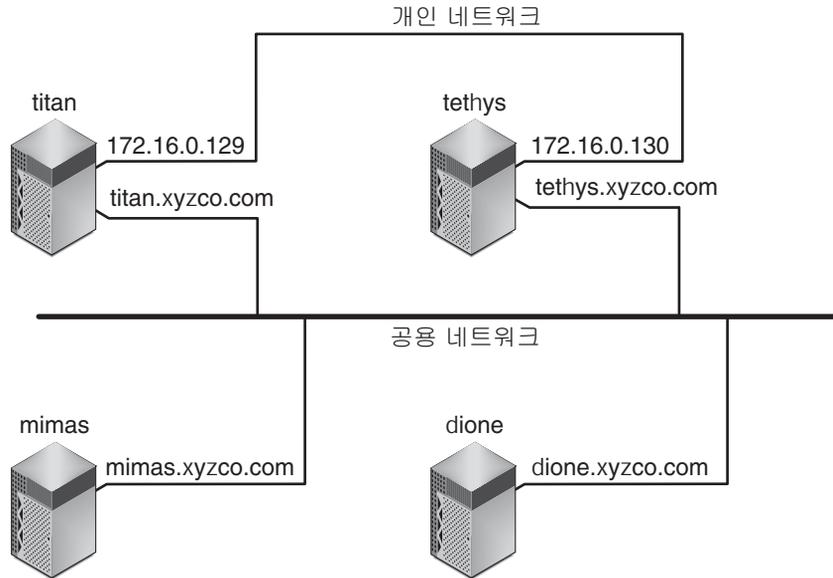


그림 4-1 네트워크 인터페이스

시스템 titan 및 tethys는 인터페이스 172.16.0.129 및 172.16.0.130으로 개인 네트워크 연결을 공유합니다. 시스템 관리자는 titan 및 tethys가 항상 개인 네트워크 연결을 통해 통신하도록 각 시스템에 동일한

/etc/opt/SUNWsamfs/hosts.sharefs1.local 복사본을 만들었습니다.

코드 예 4-17은 이러한 파일의 정보를 표시합니다.

코드 예 4-17 titan 및 tethys에 있는 hosts.sharefs1.local 파일

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

시스템 mimas 및 dione은 개인 네트워크에 없습니다. titan 및 tethys의 공용 인터페이스를 통해 항상 titan 및 tethys에 연결되어 있도록 하려면 시스템 관리자가 mimas 및 dione에 있는 /etc/opt/SUNWsamfs/hosts.sharefs1.local과 동일한 복사본을 만듭니다. 코드 예 4-18은 이러한 파일의 정보를 보여줍니다.

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

## Sun StorageTek QFS 환경에서 메타 데이터 서버 변경

이 절의 절차에서는 Sun Cluster 소프트웨어와 같은 소프트웨어 패키지의 자동 구성원 서비스 기능을 사용하지 않고 Sun StorageTek QFS 공유 파일 시스템에서 메타 데이터 서버로서 작동하는 호스트를 변경하는 방법에 대해 설명합니다.

다음 상황에서 수동으로 메타 데이터 서버를 변경할 수 있습니다.

- 메타 데이터 서버를 사용할 수 없게 되는 경우
- 메타 데이터 서버 또는 가능한 메타 데이터 서버를 변경하려는 경우

메타 데이터 서버 변경이 성공하려면 기존 메타 데이터 서버 및 모든 가능한 메타 데이터 서버의 마운트 옵션이 같아야 합니다.

변경이 수행될 시점에 기존 메타 데이터 서버가 사용 가능한지 여부에 따라 다음 절차 중 하나를 선택하십시오.

- 89페이지의 "메타 데이터 서버가 사용 가능할 때 메타 데이터 서버 변경"
- 89페이지의 "메타 데이터 서버가 사용 불가능할 때 메타 데이터 서버 변경"

## ▼ 메타 데이터 서버가 사용 가능할 때 메타 데이터 서버 변경

- 기존 메타 데이터 서버에서 `samsharefs(1M) -s` 명령을 입력하여 새로운 메타 데이터 서버를 선언합니다.

예를 들어, 다음과 같습니다.

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

---

주 – Sun SAM-QFS 환경에서 모든 아카이빙 작업은 이 명령을 실행하기 전에 메타 데이터 서버에서 중지되어야 합니다.

---

## ▼ 메타 데이터 서버가 사용 불가능할 때 메타 데이터 서버 변경

공유 파일 시스템의 메타 데이터 서버가 충돌한 경우, 메타 데이터 서버를 재부트한 후에만 메타 데이터 서버를 변경하거나 메타 데이터 서버가 재부트되기 전에 해당 서버가 I/O를 실행할 수 없도록 하는 것이 안전합니다. 파일 시스템이 손상될 수 있으므로 서버를 중지하기 위해 다음 방법을 사용하지 마십시오.

- L1-A 키 시퀀스 실행
- 다른 호스트에 반사적 장애 조치 수행
- `go(계속)` 명령 실행, 덤프 파일 재요청 또는 기존 메타 데이터 서버에 대한 `sync` 명령 실행

마찬가지로 메타 데이터 서버 패닉이 발생하고 커널 `adb(1)`로 드롭되는 경우, 메타 데이터 서버를 변경하지 말고 서버에서 `:c(계속)` 명령을 실행합니다. 이 작업은 기존의 메타 데이터 서버가 스테일 버퍼를 현재의 활성 파일 시스템으로 밀어 내게 합니다.

메타 데이터 서버를 변경하려면 다음의 단계를 사용합니다.

1. 다시 부트하지 않고 기존 메타 데이터 서버를 다시 시작할 수 있는지 확인합니다.  
특히 서버의 전원이 꺼졌는지, 재부팅되었는지, 중지되었는지 또는 메타 데이터 디스크에서 연결이 끊어졌는지 확인하십시오. 기존 메타 데이터 서버를 불러와 모든 버퍼를 지우거나 제거하는 것(또는 다시 기록할 수 없도록 하는 것)이 목적입니다.

코드 예 4-19는 kadb 프롬프트에서 사용하는 키 순서를 표시합니다.

코드 예 4-19            메타 데이터 서버가 kadb 프롬프트에서 다시 시작할 수 없도록 만드는 키 순서

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

코드 예 4-20은 PROM 프롬프트에서 사용하는 키 순서를 표시합니다.

코드 예 4-20            메타 데이터 서버가 PROM 프롬프트에서 시작할 수 없도록 만드는 키 순서

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

인수에 대해 `-r` 또는 `-v`와 같은 `boot(1M)` 명령에 대한 인수를 지정합니다. 자세한 내용은 `boot(1M)` 매뉴얼 페이지를 참조하십시오.

2. 새로운(가능한) 메타 데이터 서버에서 최소한 최대 임대 시간이 경과한 후에 `samsharefs(1M)` 명령을 실행합니다.

예를 들어, 다음과 같습니다.

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

`samsharefs(1M)` 명령을 실행하기 전에 모든 클라이언트 임대가 만료되어야 하므로 대기합니다. 임대 시간의 만료 여부를 확실히 알 수 없는 경우에는 `samu(1M)` N 디스플레이를 사용합니다. `samu(1M)`에 대한 자세한 내용은 253페이지의 "[samu\(1M\) 운영자 유틸리티 사용](#)"을 참조하십시오. 임대 및 해당 기간에 대한 자세한 내용은 247페이지의 "[Sun StorageTek QFS 공유 파일 시스템에서 임대 사용: rdlease=n, wrlease=n 및 aplease=n 옵션](#)"을 참조하십시오.



주의 - 마운트된 파일 시스템에서 `samsharefs(1M)` 명령에 `-R` 옵션을 사용하여 메타 데이터 서버 호스트를 변경하는 경우, 먼저 활성 중인 메타 데이터 서버를 중지하고 비활성화한 후 연결을 끊어야 합니다. 이렇게 하지 않으면 파일 시스템이 손상될 수 있습니다.

3. (선택 사항) 파일 시스템을 마운트 해제합니다.

파일 시스템 검사를 수행하려는 경우에만 이 단계를 수행합니다.

63페이지의 "[Sun StorageTek QFS 공유 파일 시스템 마운트 해제](#)"에 설명된 절차를 따르십시오.

#### 4. (선택 사항) samfsck(1M) 명령을 사용하여 파일 시스템 검사를 수행합니다.

Sun StorageTek QFS 공유 파일 시스템의 메타 데이터 서버가 충돌하는 경우 서버를 재부트해야 하고 samfsck(1M)가 실행되기 전에 모든 클라이언트에서 파일 시스템을 마운트 해제해야 합니다. 서버 및 클라이언트는 파일의 길이를 변경하기 전에 블록을 사전 할당합니다. samfsck(1M) 명령은 추가 블록이 할당되어 있는 파일을 제거하고 이 추가 블록에는 데이터가 포함되어 있을 수도 있습니다. 제거된 파일이 클라이언트로부터 크기 업데이트를 기다리는 경우, 해당 파일은 클라이언트가 계속하면 블록을 손실하게 됩니다. 결과적으로 파일은 데이터를 손실하게 되고 손실된 데이터는 0으로 읽습니다.

---

## SAM-QFS 환경에서 메타 데이터 서버 변경

이 절의 절차에서는 Sun Cluster 소프트웨어와 같은 소프트웨어 패키지의 자동 구성원 서비스 기능을 사용하지 않고 SAM-QFS 공유 파일 시스템에서 메타 데이터 서버로서 작동하는 호스트를 변경하는 방법에 대해 설명합니다.

다음 상황에서 수동으로 메타 데이터 서버를 변경할 수 있습니다.

- 메타 데이터 서버를 사용할 수 없게 되는 경우
- 메타 데이터 서버 또는 가능한 메타 데이터 서버를 변경하려는 경우

메타 데이터 서버 변경이 성공하려면 기존 메타 데이터 서버 및 모든 가능한 메타 데이터 서버의 마운트 옵션이 같아야 합니다.

### ▼ SAM-QFS 환경에서 메타 데이터 서버 변경

Sun StorageTek SAM는 항상 하나의 호스트에서만 실행될 수 있습니다. 이 절차에서는 두 개 시스템이 모두 전송 시에 작동하는 것으로 가정합니다. 이 예제의 경우 호스트 A에서 호스트 B로 Sun StorageTek SAM 아카이빙 기능을 이동합니다.

이 절차를 수행하기 전에 호스트 B가 호스트 A의 로봁 카탈로그에 액세스했는지 확인하십시오. archiver.cmd 파일, mcf 파일, stager.cmd 및 기타 구성 파일은 호스트 A에 있는 파일과 동일해야 합니다.

1. 다음 단계를 수행하여 호스트 A의 Sun StorageTek SAM 아카이빙 프로세스를 유훈 상태로 전환합니다.
  - a. samcmd aridle 및 samcmd stidle을 실행하여 호스트 A에서 아카이빙 및 스테이징을 정지합니다.

이 명령을 사용하면 현재 아카이빙 및 스테이징은 완료되지만 새 작업은 시작되지 않습니다.

b. 호스트 A의 모든 테이프 드라이브를 유틸리티 상태로 전환합니다.

`samcmd eq idle`을 사용하여 이 작업을 수행할 수 있는데, 여기에서 `eq`는 드라이브의 장비 서수입니다. 이렇게 하면 현재 I/O가 완료된 후 드라이브가 "off" 상태로 전환됩니다.

c. 아카이버와 스테이지가 유틸리티 상태이며 테이프 드라이브가 "off" 상태이면 `samd stop` 명령을 실행하여 모든 로봇 및 테이프 관련 데몬을 정지합니다.

d. 리사이클러를 실행하는 cron 작업이 있는 경우 `crontab`에서 이 항목을 제거하고 현재 리사이클러가 실행되고 있지 않는지 확인합니다.

이 시점에서 Sun StorageTek SAM는 정지 상태이며 호스트 B에 대한 파일 시스템의 장애 조치를 수행할 수 있습니다.

2. 호스트 B에서 `samd config`를 실행하여 호스트 B의 Sun StorageTek SAM를 시작합니다.

이렇게 하면 `sam-fsd`와 해당 하위 프로세스(아카이버, 스테이지 등)가 구성 파일을 재구성하고 다시 읽습니다. `sam-amld`와 테이프 라이브러리 관련 데몬도 시작됩니다. 이 시점에서 스테이지를 대기하는 모든 Sun StorageTek QFS 공유 클라이언트 응용 프로그램은 스테이지 요청을 다시 실행해야 합니다.

이제 호스트 B는 모든 Sun StorageTek QFS 파일 시스템의 Sun StorageTek SAM 서버 및 메타 데이터 서버로서 완전히 기능을 수행하게 됩니다.

---

# Sun StorageTek QFS 공유 파일 시스템에 서의 클라이언트-서버 통신

Sun StorageTek QFS 공유 파일 시스템의 작동은 인터럽트 가능한 하드 연결 작동입니다. 각 클라이언트는 메타 데이터 서버를 사용할 수 없더라도 계속해서 서버와 통신을 시도합니다. 메타 데이터 서버가 응답하지 않는 경우, 사용자가 Ctrl-C를 눌러 모든 보류되고 차단된 I/O 전송을 마칠 수 있습니다.

시스템은 다음 메시지를 생성하여 상태 조건을 알려줍니다.

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

이 메시지는 클라이언트 sam-sharefsd 때문 또는 서버 sam-sharefsd 때문이 활성화되어 있지 않은 경우에도 생성됩니다. 서버가 응답할 때 다음 메시지가 생성됩니다.

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

파일 시스템이 메타 데이터 서버가 아니라 클라이언트에 마운트된 경우, 시스템에서 다음 메시지를 생성합니다.

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

Sun StorageTek QFS 공유 파일 시스템이 서버에서 마운트될 때 다음 메시지가 생성됩니다.

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

모든 클라이언트 대신 메타 데이터 서버가 파일 이름을 찾기 때문에 메타 데이터 서버에서 Solaris DNLC(Directory Name Lookup Cache)의 크기가 기본값이면 성능이 저하될 수 있습니다. 클라이언트가 자주 많은 수의 파일을 여는 경우 성능을 높이려면 이 캐시의 크기를 기본값보다 두 배 또는 심지어 세 배까지 늘릴 수 있습니다.

이 절차는 *Solaris Tunable Parameters Reference Manual*에 기록되어 있습니다. 디렉토리 이름 조회 캐시의 크기를 제어하는 매개변수는 ncsizes입니다.



# 파일 시스템 할당량 관리

---

이 장에서는 파일 시스템 할당량을 활성화 및 관리하는 방법을 설명합니다. 본 장에는 다음 절이 포함되어 있습니다.

- 95페이지의 "개요"
- 98페이지의 "할당량 활성화"
- 108페이지의 "할당량 확인"
- 110페이지의 "할당량 변경 및 제거"

---

## 개요

파일 시스템 할당량은 특정 사용자, 사용자 그룹 또는 관리자 세트라고 부르는 사이트가 결정하는 사용자 그룹이 이용할 수 있는 온라인 및 총 디스크 공간의 양을 제어합니다.

할당량은 각 사용자가 소비할 수 있는 공간의 양 및 inode의 수를 제한함으로써 파일 시스템의 크기를 제어하게 됩니다. 할당량은 사용자 홈 디렉토리를 포함하는 파일 시스템에서 특히 유용할 수 있습니다. 할당량이 활성화된 후에는 사용량을 모니터링하고 필요에 따라 할당량을 조정할 수 있습니다.

파일 시스템은 사용자에게 데이터에 대한 블록과 파일에 대한 inode를 제공합니다. 각 파일은 하나의 inode를 사용하고, 파일 데이터는 디스크 할당 단위(DAU)에 저장됩니다. DAU 크기는 파일 시스템이 만들어질 때 결정됩니다. 할당량은 디스크 사용량을 512바이트의 배수로 계산합니다.

다음 하위 절에서는 할당량 사용에 대한 기본적인 정보를 제공합니다.

- 96페이지의 "할당량 유형, 할당량 파일 및 할당량 레코드"
- 97페이지의 "소프트 제한 및 하드 제한"
- 97페이지의 "디스크 블록 및 파일 제한"

## 할당량 유형, 할당량 파일 및 할당량 레코드

할당량은 사용자 ID, 그룹 ID 또는 관리자의 사이트별 그룹 지정 방식에 따라 설정할 수 있습니다. 이 사이트별 그룹 지정 방식을 관리자 세트 ID라고 합니다. 예를 들어 관리자 세트 ID는 파일 시스템 할당량이 적용되는 프로젝트에서 작업하는 사용자 그룹을 식별하는 데 사용할 수 있습니다.

시스템에서 파일 시스템의 루트 디렉토리에 하나 이상의 할당량 파일이 있음을 감지할 때 및 quota 마운트 옵션이 사용 중일 때 할당량을 사용할 수 있습니다. quota 마운트 옵션은 기본적으로 활성화됩니다. noquota를 사용 중인 파일 시스템을 마운트하는 경우 할당량을 사용할 수 없습니다. 마운트 옵션에 대한 정보는 mount\_samfs(1M) 매뉴얼 페이지를 참조하십시오.

각 할당량 파일에는 레코드 시퀀스가 포함되어 있습니다. 레코드 0은 시스템 관리자의 할당량 및 자원 사용량에 대한 레코드입니다. 시스템 관리자 할당량은 강요되지 않지만 할당량 파일의 다음 레코드에 대한 템플릿으로 시스템 관리자의 레코드를 포함하는 모든 레코드를 사용할 수 있습니다. 이 연습에 대한 자세한 내용은 [106페이지의 "기존 할당량 파일을 사용하는 사용자, 그룹 또는 관리자 세트에 대한 제한 활성화 또는 변경하기"](#)를 참조하십시오.

레코드 1은 할당량 파일의 유형에 따라 사용자 1, 그룹 1 또는 관리자 세트 ID 1을 위한 할당량 파일의 레코드입니다. 다른 사용자에게 다른 할당량을 설정하기 위해 레코드 1 및 모든 후속 레코드를 편집할 수 있습니다. [표 5-1](#)은 /root에서 사용 가능한 할당량 및 할당량 파일 이름을 표시합니다.

표 5-1 할당량 파일 이름

할당량 파일 이름	할당량 유형
.quota_u	UID(시스템 사용자 ID)
.quota_g	GID(시스템 그룹 ID)
.quota_a	AID(시스템 관리자 세트 ID)

할당량 파일에서 레코드 0을 편집하고 레코드 0의 값이 모든 다른 사용자에게 대한 초기 할당량 설정값으로 사용되도록 허용하여 사용자에게 대한 기본 할당량 제한을 설정할 수 있습니다. 기본적으로 사용자 할당량 제한이 명시적으로 설정되지 않으면 레코드 0의 값이 사용됩니다.

각 할당량 파일은 128바이트의 공간이 필요합니다. 초기 할당량 0 파일에 필요한 크기를 계산하려면 다음 공식을 사용하십시오.

$$\begin{aligned}(\text{가장 높은 ID} + 1) \times 128 &= x \\ x / 4096 &= \text{할당량 0 파일 크기}\end{aligned}$$

## 소프트 제한 및 하드 제한

소프트 제한 및 하드 제한을 모두 설정할 수 있습니다. 하드 제한은 사용 가능한 시스템 자원에 대해 고정된 양을 지정하므로, 사용자는 이 한도를 절대 초과할 수 없습니다. 소프트 제한은 하드 한계까지 일시적으로 초과할 수 있는 시스템 자원의 수준을 지정합니다. 소프트 제한은 하드 제한보다 작아야 합니다.

사용자가 하드 제한을 초과하여 자원 할당을 시도하면 작업이 중단됩니다. 이 경우 작업은 실패하며 `EDQUOT` 오류를 생성합니다.

사용자가 소프트 제한을 초과한 후에는 타이머가 시작되고 사용자는 유예 기간에 들어 가게 됩니다. 타이머가 움직이는 동안 사용자는 소프트 제한 이상으로 동작할 수 있습니다. 사용자가 소프트 제한 아래로 사용량을 줄이면 타이머는 재설정됩니다. 또한 소프트 제한 아래로 사용량을 줄이지 않아 유예 기간이 끝나고 타이머가 멈추면 소프트 제한은 하드 제한으로 대체됩니다.

예를 들어 사용자의 소프트 제한이 10,000블록이고 하드 제한이 12,000블록이라고 가정합니다. 사용자의 블록 사용량이 10,000블록을 초과하고 타이머가 유예 기간을 초과하는 경우, 이 사용자는 사용량이 10,000블록 소프트 제한 아래로 떨어지기 전까지 해당 파일 시스템에서 더 이상 디스크 블록을 할당할 수 없습니다.

관리자는 `quota(1M)` 명령을 사용하여 타이머 값을 볼 수 있습니다. `quota(1)` 명령은 `quota(1M)` 명령의 사용자 버전입니다. `quota(1)` 사용자 명령에는 사용자가 자신과 관련된 할당량에 대한 정보를 얻기 위해 지정할 수 있는 옵션이 포함되어 있습니다.

## 디스크 블록 및 파일 제한

사용자의 경우, 빈 파일을 만들어 블록을 전혀 사용하지 않으면서도 자신의 `inode` 할당량을 초과하는 것이 가능합니다. 또한 사용자 할당량의 모든 데이터 블록을 소비할 만큼 큰 파일을 만들어 오직 하나의 `inode`만 사용하면서도 해당 사용자의 블록 할당량을 초과하는 것도 가능합니다.

파일 시스템 할당량은 사용자가 할당할 수 있는 512바이트 블록의 수로 표현됩니다. 하지만 디스크 공간은 `DAU`로 사용자 파일에 할당됩니다. `DAU` 설정은 `sammkfs(1M)` 명령에 `-a allocation-unit` 옵션을 사용하여 지정됩니다. 블록 할당량을 파일 시스템 `DAU`의 배수로 설정하는 것이 좋습니다. 그렇지 않으면, 사용자는 가장 가까운 `DAU`에 해당하는 블록 수까지만 할당할 수 있게 됩니다. 블록 할당량 설정에 대한 지침은 [104페이지의 "기본 할당량 값 활성화"](#)를 참조하십시오.

## 할당량 활성화

시스템 파일 편집, 할당량 파일 작성 및 다양한 할당량 명령 입력을 포함하는 프로세스를 통해 할당량을 사용할 수 있습니다.

다음 하위 절에서는 할당량을 사용하기 위해 파일 시스템을 구성하는 방법과 할당량을 활성화하는 방법에 대해 더 자세히 설명합니다.

## 할당량 설정을 위한 지침

할당량을 활성화하기 전에 각 사용자에게 할당할 디스크 공간 및 inode 개수를 결정해야 합니다. 파일 시스템의 총 공간이 초과되지 않도록 하려면 파일 시스템의 총 크기를 사용자 수로 나누면 됩니다. 예를 들어 세 명의 사용자가 100메가바이트 슬라이스를 공유하고 동일한 디스크 공간을 필요로 한다면 각 사용자에게 33메가바이트를 할당할 수 있습니다. 모든 사용자가 자신의 한계에 도달하지 않는 환경에서는 파일 시스템의 총 크기보다 많이 개별 할당량을 설정할 수도 있습니다. 예를 들어 세 명의 사용자가 100메가바이트 슬라이스를 공유할 때 각 사용자에게 40메가바이트를 할당할 수 있습니다.

표시된 형식으로 다음 할당량 명령을 사용하여 할당량 정보를 표시할 수 있습니다.

- `squota(1)` 명령은 최종 사용자를 위한 명령입니다. 이 명령을 사용하면 사용자, 그룹 또는 관리자 세트별로 할당량 정보를 얻을 수 있습니다.
- `samquota(1M)` 명령은 시스템 관리자를 위한 명령입니다. 이 명령을 사용하여 할당량 정보를 얻거나 할당량을 설정할 수 있습니다. 사용자, 그룹 또는 관리자 세트에 대해 각각 `-U`, `-G` 및 `-A` 옵션을 사용하십시오. [코드 예 5-1](#)을 참조하십시오.

코드 예 5-1                    `samquota(1M)`를 사용하여 정보 검색

```
# 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
```

## ▼ 새로운 파일 시스템이 할당량을 사용하도록 구성

새로운 파일 시스템을 작성 중이고 현재 파일 시스템에 아무 파일도 없는 경우 이 절차를 사용하십시오. 기존의 파일 시스템이 할당량을 사용하도록 구성하려면 [101페이지의 "기존 파일 시스템이 할당량을 사용하도록 구성"](#)을 참조하십시오.

이 절차를 시작하기 전에 `mount` 또는 `/etc/vfstab` 파일에 지정된 `noquota` 마운트 옵션이 없는지 확인하십시오.

1. 슈퍼유저가 됩니다.

2. 파일 시스템을 만듭니다.

Sun StorageTek QFS 설치 및 업그레이드 안내서에 설명한 단계를 따르거나 [42페이지의 "구성 예제"](#)의 예제를 사용하여 `mcf` 파일을 만들고, 마운트 지점을 만들며, 파일 시스템을 초기화하는 등의 작업을 수행합니다.

3. `mount(1M)` 명령을 사용하여 파일 시스템을 마운트합니다.

예를 들어, 다음과 같습니다.

```
# mount /qfs1
```

4. `dd(1M)` 명령을 사용하여 할당량 파일을 만듭니다.

이 명령에 대한 인수는 다음과 같이 만들려는 할당량 유형에 따라 다릅니다.

■ 관리자 세트 할당량을 만들려면 다음 명령을 사용하십시오.

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

■ 그룹 할당량을 만들려면 다음 명령을 사용하십시오.

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

■ 사용자 할당량을 만들려면 다음 명령을 사용하십시오.

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

`dd(1M)` 명령에 대한 자세한 내용은 `dd(1M)` 매뉴얼 페이지를 참조하십시오.

5. **umount(1)** 명령을 사용하여 할당량 파일이 만들어진 파일 시스템을 마운트 해제하십시오.

예를 들어, 다음과 같습니다.

```
# umount /qfs1
```

파일 시스템을 마운트 해제해야 다시 마운트할 수 있으며 마운트 시에 할당량 파일을 읽게 됩니다. **umount(1M)** 명령에 대한 자세한 내용은 **umount(1M)** 매뉴얼 페이지를 참조하십시오.

6. **samfsck(1M)** 명령을 사용하여 파일 시스템 검사를 수행합니다.

다음 예제에서 **-F** 옵션은 할당량 파일에서 사용 중인 값을 재설정합니다.

```
# samfsck -F qfs1
```

7. **mount(1M)** 명령을 사용하여 파일 시스템을 다시 마운트합니다.

시스템에서 **/root** 디렉토리에 하나 이상의 할당량 파일이 있음을 감지할 경우 할당량을 사용할 수 있습니다.

---

주 - **/etc/vfstab** 또는 **samfs.cmd** 파일에 quota 마운트 옵션을 포함할 필요가 없습니다. quota 마운트 옵션은 **mount(1M)** 명령에서 기본적으로 사용 가능하며, 할당량은 시스템에서 할당량 파일의 존재를 감지할 때 자동으로 사용 가능합니다.

---

**mount(1M)** 명령에 대한 자세한 내용은 **mount\_samfs(1M)** 매뉴얼 페이지를 참조하십시오.

8. **samquota(1M)** 명령을 사용하여 사용자, 그룹 또는 관리자 세트에 대한 할당량을 설정합니다.

이 장의 하위 절에 이 프로세스에 대한 절차 및 예제가 있습니다. **samquota(1M)** 명령에 대한 자세한 내용은 **samquota(1M)** 매뉴얼 페이지를 참조하십시오.

## ▼ 기존 파일 시스템이 할당량을 사용하도록 구성

파일이 이미 있는 파일 시스템에 대해 할당량을 만드는 경우 이 절차를 사용하십시오. 새로운 파일 시스템이 할당량을 사용하도록 구성하려면 99페이지의 "새로운 파일 시스템이 할당량을 사용하도록 구성"을 참조하십시오.

이 절차를 시작하기 전에 `samfs.cmd` 또는 `/etc/vfstab` 파일에 지정된 `noquota` 마운트 옵션이 없는지 확인하십시오.

1. `su(1)` 명령을 사용하여 수퍼유저로 전환합니다.
2. `mount(1M)` 명령을 사용하여 `/etc/mnttab` 파일을 조사하고 파일 시스템이 마운트되었는지 확인합니다.

```
# mount
```

파일 시스템이 표시되는 마운트 목록에 나열되는지 확인합니다.

3. `cd(1)` 명령을 사용하여 할당량이 활성화될 파일 시스템의 루트 디렉토리로 변경합니다. 예를 들어, 다음과 같습니다.

```
# cd /oldfs1
```

4. `ls(1) -a` 명령을 사용하여 이 디렉토리의 파일 목록을 검색하고 할당량이 이미 파일 시스템에 존재하지 않는지 확인합니다.

`.quota_u`, `.quota_g`, `.quota_a`.

파일 시스템에 대해 임의의 할당량 유형이 설정된 경우 나중에 다른 할당량 유형을 설정할 수 있습니다. 새 할당량을 추가할 때 기존 할당량 파일을 수정하지 않도록 주의하십시오.

5. 적용하려는 할당량 유형에 대한 할당량 파일이 존재하지 않는 경우 **dd(1M)** 명령을 사용하여 할당량 파일을 작성합니다.

적용할 할당량 유형에서 가장 높은 기존 ID 번호를 결정하십시오. 이러한 ID에 대해 레코드를 포함할 수 있을 만큼 충분히 초기의 0 할당량 파일을 만드십시오. 각 할당량 파일 레코드는 128바이트가 필요합니다.

예를 들어 관리자 세트 할당량을 활성화하기 원하고 파일 시스템에서 사용 중인 가장 높은 관리자 세트 ID가 1024인 경우 다음과 같이 계산됩니다.

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

$$131200/4096 = 32.031...$$

다음 명령을 사용하십시오.

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

dd(1M) 명령에 대한 자세한 내용은 dd(1M) 매뉴얼 페이지를 참조하십시오.

6. **umount(1)** 명령을 사용하여 할당량 파일이 만들어진 파일 시스템을 마운트 해제하십시오.

예를 들어, 다음과 같습니다.

```
# umount /oldfs1
```

파일 시스템을 마운트 해제해야 다시 마운트할 수 있으며 마운트 시에 할당량 파일을 잃게 됩니다. 파일 시스템 마운트 해제에 대한 자세한 내용은 [62페이지의 "파일 시스템 마운트 해제"](#)를 참조하십시오.

7. **samfsck(1M) -F** 명령을 사용하여 파일 시스템 검사를 수행합니다.

이 명령은 할당량 파일에 할당된 레코드를 올바른 현재 사용량 정보로 업데이트합니다.

예를 들어, 다음과 같습니다.

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

8. **mount(1M)** 명령을 사용하여 할당량 파일이 만들어진 파일 시스템을 다시 마운트합니다.

시스템에서 /root 디렉토리에 하나 이상의 할당량 파일이 있음을 감지할 경우 할당량을 사용할 수 있습니다.

/etc/vfstab 또는 samfs.cmd 파일에 quota 마운트 옵션을 포함할 필요가 없습니다. quota 마운트 옵션은 mount(1M) 명령에서 기본적으로 사용 가능하며, 할당량은 시스템에서 할당량 파일의 존재를 감지할 때 자동으로 사용 가능합니다.

---

주 - 할당량 파일이 있는 경우 및 할당량이 활성화되지 않고 파일 시스템이 마운트된 경우, 블록 또는 파일이 할당되거나 할당 해제되면 할당량 레코드가 실제 사용량과 일치하지 않게 됩니다. 할당량을 갖는 파일 시스템이 마운트되고 할당량이 비활성화되어 실행되는 경우 할당량이 다시 활성화된 파일 시스템을 다시 마운트하기 전에 `samfsck(1M) -F` 명령을 실행하여 할당량 파일 사용량 계수를 업데이트하십시오.

---

`mount(1M)` 명령에 대한 자세한 내용은 `mount_samfs(1M)` 매뉴얼 페이지를 참조하십시오.

9. `samquota(1M)` 명령을 사용하여 사용자, 그룹 또는 관리자 세트에 대한 할당량을 설정합니다.

이 장의 하위 절에 이 프로세스에 대한 절차 및 예제가 있습니다. `samquota(1M)` 명령에 대한 자세한 내용은 `samquota(1M)` 매뉴얼 페이지를 참조하십시오.

## ▼ 디렉토리 및 파일에 관리자 세트 ID 할당

1. `su(1)` 명령을 사용하여 수퍼유저로 전환합니다.
2. 관리 ID를 설정합니다.

다음과 같이 `samchaid(1M)` 명령을 사용하여 디렉토리 또는 파일에 대한 관리자 세트 ID를 변경하십시오.

- 파일 또는 디렉토리에 대한 ID를 설정하려면 디렉토리 이름이나 경로를 지정하십시오. 예를 들어, 다음과 같습니다.

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

- 디렉토리 트리에 대한 ID를 설정하려면 `-R` 및 `-h` 옵션(필요한 경우)을 사용하십시오. `-R` 옵션은 순환 작업을 지정하고, `-h` 옵션은 대상이 아닌 링크를 변경합니다. 예를 들어, 다음과 같습니다.

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

`samchaid(1M)` 명령에 대한 자세한 내용은 `samchaid(1M)` 매뉴얼 페이지를 참조하십시오.

## 무한 할당량 설정

무한 할당량은 특별한 종류의 할당량입니다. 무한 할당량을 가진 사용자는 사용 가능한 파일 시스템 리소스에 대해 액세스가 거부되는 경우가 없습니다. 무한 할당량 값을 사용자, 그룹 또는 관리자 세트 ID 할당량 파일의 레코드 0에서 설정할 수 있고, 이 경우 이 레코드를 새로운 사용자, 그룹 또는 관리자 세트 ID에 대한 기본값으로 사용할 수 있습니다.

### ▼ 무한 할당량 설정

- 할당량 한계를 0으로 설정하려면 `samquota(1M)` 명령을 사용하십시오. 예를 들어, 다음과 같습니다.

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

`samquota(1M)` 명령을 사용하여 모든 하드 제한 및 소프트 제한의 값을 0으로 설정하여 특정 사용자, 그룹 또는 관리자 세트 ID에 대해 무한 할당량을 설정할 수 있습니다. 코드 예 5-2는 무한 할당량 설정 방법을 표시합니다.

코드 예 5-2

무한 할당량 설정

```
# 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.

## 기본 할당량 값 활성화

`samquota(1M)` 명령을 사용하여 사용자, 그룹 또는 관리자 세트에 대한 기본 할당량을 활성화할 수 있습니다. 이 작업은 0으로 설정된 사용자, 그룹 또는 관리자 세트의 기본 제한을 통해 수행됩니다.

## ▼ 사용자, 그룹 또는 관리자 세트에 대한 기본 할당량 값 활성화

- **samquota(1M)** 명령을 사용하십시오.

예를 들어 다음 **samquota(1M)** 명령은 모든 관리자 세트 ID에 대한 기본 할당량을 설정합니다.

```
# 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
```

위의 명령은 모든 사용자의 초기화되지 않은 관리자 세트 할당량 제한을 다음과 같이 설정하고 있습니다.

- 소프트 온라인 블록 제한(-b *limit:s*)은 12,000블록으로 설정됩니다.
- 하드 온라인 블록 제한(-b *limit:h*)은 15,000블록으로 설정됩니다.
- 총 소프트 블록 제한(-b *limit:s:t*)은 12기가블록으로 설정됩니다.
- 총 하드 블록 제한(-b *limit:h:t*)은 15기가블록으로 설정됩니다.
- 소프트 파일 제한(-f *limit:s*)은 1000파일로 설정됩니다.
- 하드 파일 제한(-f *limit:h*)은 1200파일로 설정됩니다.
- 유예 기간(-t *limit*)은 1주일로 설정됩니다.

---

주 - 할당량 레코드가 이미 있는 경우 기존 값이 계속 사용됩니다. 이것은 예를 들어 관리자 그룹이 이미 블록을 할당한 경우에 발생합니다.

---

-A 0 위치에 -U 0 또는 -G 0를 지정하여 사용자 또는 그룹에 대해 유사한 기본 할당량을 설정할 수 있습니다.

**samquota(1M)** 명령에 대한 자세한 내용은 **samquota(1M)** 매뉴얼 페이지를 참조하십시오.

## 제한 활성화

samquota(1M) 명령을 사용하여 특정 사용자, 그룹 또는 관리자 세트에 대한 여러 제한을 활성화할 수 있습니다.

### ▼ 사용자, 그룹 또는 관리자 세트에 대한 제한 활성화

- samquota(1M) 명령을 사용하십시오.

코드 예 5-3은 사용자, 그룹 및 관리자 세트 각각에 대한 제한을 활성화하는 명령을 표시합니다.

코드 예 5-3      할당량 명령

```
# 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) 명령에 대한 자세한 내용은 samquota(1M) 매뉴얼 페이지를 참조하십시오.

### ▼ 기존 할당량 파일을 사용하는 사용자, 그룹 또는 관리자 세트에 대한 제한 활성화 또는 변경하기

할당량이 설정된 후 또 다른 사용자, 그룹 또는 관리자 세트에 대해 제한할 경우 기존 할당량 파일을 템플릿으로 사용할 수 있습니다. 다음 절차에서 내용을 확인하십시오. 또한 이 절차를 사용하여 모든 할당량 설정을 변경할 수 있습니다.

1. samquota(1M) 명령을 사용하여 할당량 파일을 검색하고 해당 출력을 임시 파일로 보냅니다.

다음 추가 옵션 중 하나 이상과 -e 옵션을 사용하십시오. -U *userID*, -G *groupID* 또는 -A *adminsetID*

코드 예 5-4는 템플릿으로 사용할 quota.group 파일을 작성 및 검색하는 방법을 표시합니다.

---

주 - 그룹 할당량 항목을 사용자 할당량 항목을 작성하기 위한 템플릿으로 사용할 수 있습니다.

---

코드 예 5-4 quota.group 파일

```
# 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. 편집기를 사용하여 방금 작성한 임시 파일을 편집합니다.

코드 예 5-5는 1단계에서 생성한 파일을 vi(1) 편집기에서 연 것입니다. Group ID 101 이 102로 변경되었습니다. 이것은 그룹 101에 대한 할당량 세트를 그룹 102로 복사하는 명령을 생성합니다.

코드 예 5-5 편집 후 quota.group 파일

```
# Type ID
#           Online Limits                Total  Limits
#           soft          hard            soft          hard
# Files
# Blocks
# Grace Periods
#
samquota -G 102 \
-f      200:s:o -f      300:h:o          -f      200:s:t -f      300:h:t \
-b     40000:s:o -b     60000:h:o        -b 40000000:s:t -b 60000000:h:t \
-t      1d:o                               -t 1d:t /sam6
```

3. 파일을 저장하고 편집기를 종료합니다.

4. 편집기에서 수행한 변경사항을 적용하려면 셸을 사용하여 해당 파일을 실행합니다. 예를 들어, 다음과 같습니다.

```
# sh -x /tmp/quota.group
```

이 예제에서 -x 옵션은 실행하는 명령을 셸이 에코하도록 명령합니다. 필요한 경우 -x 옵션을 생략할 수 있습니다.

# 할당량 확인

디스크 및 inode 할당량을 활성화한 후 이러한 할당량을 확인할 수 있습니다. `samquota(1M)` 명령은 각 사용자, 그룹 또는 관리자 세트에 대한 할당량 보고서를 생성하는 관리자 명령입니다. `squota(1)` 명령은 사용자들이 자신의 개별 할당량을 확인할 수 있는 사용자 명령입니다.

## ▼ 초과된 할당량 확인

1. 슈퍼유저가 됩니다.
2. `samquota(1M)` 명령을 사용하여 마운트된 파일 시스템에 적용되는 할당량을 표시합니다.
  - 사용자 할당량을 표시하려면 다음 명령을 지정하십시오.

```
# samquota -U userID [ file ]
```

`userID`에 할당량을 검토할 사용자의 숫자로 된 사용자 ID 또는 사용자 이름을 지정합니다.

`file`에는 선택된 사용자, 그룹 또는 관리자 세트에 대한 파일 시스템을 지정합니다. `file` 인수는 파일 시스템에서 특정 파일의 이름이 될 수도 있습니다. 일반적으로 `file` 은 파일 시스템의 루트 디렉토리의 이름입니다.

예제 1. 코드 예 5-6은 서버에서 `sam6` 파일 시스템의 사용자 `hm1259`의 할당량 통계를 검색하고 이 사용자가 할당량을 초과하지 않고 있음을 나타내는 출력을 표시합니다.

코드 예 5-6

사용자 `hm1259`에 대해 초과된 할당량 확인

```
# samquota -U hm1259 /sam6
```

Online Limits						Total Limits		
Type	ID	In Use	Soft	Hard	In Use	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		

예제 2. 코드 예 5-7은 마운트된 모든 Sun StorageTek QFS 파일 시스템에서 사용자 memil의 할당량 통계를 검색하고 이 사용자가 할당량을 초과하고 있음을 나타내는 출력을 표시합니다. 위 출력에서 Blocks 행의 플러스 기호(+)를 주의하십시오. 소프트 할당량 제한이 파일에 대해 초과할 경우에는 이 플러스 기호가 Files 행에도 나타날 수 있습니다.

코드 예 5-7 사용자 memil에 대해 초과된 할당량 확인

```
# samquota -U memil

                Online Limits                Total Limits
                Soft    Hard                Soft    Hard
Type  ID      In Use
/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
```

하드 제한이 초과되거나 소프트 제한이 초과되고 유예 기간이 만료되면 위의 In Use 필드에 별표(\*)가 나타납니다. 할당량 레코드에 제한이 일관성이 없다고 판단되면(예: 소프트 제한이 하드 제한보다 큰 경우) 이 필드에 느낌표가 나타나고 모든 할당 작업이 금지됩니다.

- 그룹 할당량을 표시하려면 다음 명령을 지정하십시오.

```
# samquota -G groupID [ file ]
```

groupID에 할당량을 검토할 사용자 그룹의 숫자로 된 그룹 ID 또는 그룹 이름을 지정합니다. file에는 선택한 그룹에 대한 특정 파일 시스템을 지정합니다. file 인수는 파일 시스템에서 특정 파일의 이름이 될 수도 있습니다. 일반적으로 file은 파일 시스템의 루트 디렉토리의 이름입니다.

예를 들어 다음 명령은 qfs3 파일 시스템에서 turtles라는 그룹에 대한 사용자 할당량 통계를 검색합니다.

```
# samquota -G turtles /qfs3
```

- 관리자 세트 할당량을 표시하려면 다음 명령을 지정하십시오.

```
# samquota -A adminsetID [ file ]
```

*adminsetID*에 할당량을 검토할 사이트별 관리자 세트의 숫자로 된 관리자 세트 ID를 지정합니다. *file*에는 선택된 관리자 세트에 대한 파일 시스템을 지정합니다. *file* 인수는 파일 시스템에서 특정 파일의 이름이 될 수도 있습니다. 일반적으로 *file*은 파일 시스템의 루트 디렉토리의 이름입니다.

예를 들어 다음 명령은 마운트된 모든 Sun StorageTek QFS 파일 시스템에서 관리자 세트 457에 대한 사용자 할당량 통계를 검색합니다.

```
# samquota -A 457
```

---

## 할당량 변경 및 제거

할당량을 변경하여 사용자에게 할당된 디스크 공간 또는 inode의 수를 조정할 수 있습니다. 사용자 또는 전체 파일 시스템으로부터 할당량을 제거할 수도 있습니다. 다음 하위 절에서는 할당량을 변경하고 제거하는 방법에 대해 설명합니다.

- 110페이지의 "유예 기간 변경"
- 113페이지의 "유예 기간 만료 변경"
- 114페이지의 "추가 파일 시스템 자원 할당 방지"
- 117페이지의 "파일 시스템의 할당량 제거"
- 118페이지의 "할당량 정정"

### ▼ 유예 기간 변경

`samquota(1M)` 명령을 사용하여 소프트 시간 제한 유예 기간을 변경할 수 있습니다.

1. `samquota(1M)` 명령을 사용하여 사용자, 그룹 또는 관리자 세트에 대한 할당량 통계를 검색합니다.

지침에 대해서는 108페이지의 "초과된 할당량 확인"을 참조하십시오.

예제. 코드 예 5-8은 그룹 sam에 대한 정보를 검색하고 이 그룹이 해당 소프트 제한을 초과함을 표시합니다.

코드 예 5-8            소프트웨어 제한 초과

```
# samquota -G sam /sam6

                Online Limits                Total Limits
                Soft   Hard                Soft   Hard
/sam6
Files group    101      32      2000   2000      32      2000   2000
Blocks group   101     41888* 40000 60000000 43208 60000000 60000000
Grace period                   1w                          1w
--> Online soft limits under enforcement (since 30s ago)
```

2. samquota(1M) 명령 출력을 검토하고 새 제한의 값을 판별합니다.
3. samquota(1M) 명령을 사용하여 소프트 제한 유예 기간을 변경하십시오.

코드 예 5-9는 사용할 samquota(1M) 명령 옵션을 표시합니다.

코드 예 5-9            samquota(1M)를 사용한 소프트 시간 제한 유예 기간 변경

```
# samquota -U userID -t interval file
# samquota -G groupID -t interval file
# samquota -A adminID -t interval file
```

이들 명령에 대한 인수는 다음과 같습니다.

- *userID*는 할당량을 변경할 사용자의 숫자로 된 사용자 ID 또는 사용자 이름입니다.
- *groupID*는 할당량을 변경할 사용자 그룹의 숫자로 된 그룹 ID 또는 그룹 이름입니다.
- *adminID*는 할당량을 변경할 사이트별 관리자 세트의 숫자로 된 관리자 세트 ID입니다.
- *interval*은 유예 기간에 사용될 간격입니다. *interval*에 양을 나타내는 정수를 지정한 후, 필요한 경우 단위 승수를 지정하십시오. 기본 단위 승수는 초를 의미하는 s입니다. w(주), d(일), h(시간) 또는 m(분)을 지정할 수도 있습니다.
- *file*은 선택된 사용자, 그룹 또는 관리자 세트에 대한 특정 파일 시스템입니다. *file* 인수는 파일 시스템에서 특정 파일의 이름이 될 수도 있습니다. 일반적으로 *file*은 파일 시스템 루트 디렉토리의 이름입니다.

예를 들어 사용자 mem1에 대한 유예 기간을 변경하고자 했다고 가정합니다.  
**코드 예 5-10**은 할당량을 검증하는 데 사용하는 samquota(1M) 명령 및 해당 출력을 표시합니다.

코드 예 5-10 유예 기간 변경

```
# samquota -U mem1 /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			3d			0s		
---> Warning: online soft limits to be enforced in 2d23h59m7s								

다음 명령을 입력하면 유예 기간이 단축됩니다.

```
# samquota -U mem1 -t 1d /sam6
```

**코드 예 5-11**은 새 할당량을 확인하는 데 사용하는 samquota(1M) 명령을 표시합니다.

코드 예 5-11 새 할당량 확인

```
# samquota -U mem1 /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								

## 유예 기간 만료 변경

사용자가 소프트 할당량 제한을 초과한 경우, 유예 기간 자체를 변경해도 이미 시작된 유예 기간의 만료 타이머는 수정되지 않습니다. 유예 기간이 이미 시작된 경우에는 다음 중 하나의 방법으로 `samquota(1M)` 명령을 사용하여 유예 기간을 수정할 수 있습니다.

- 유예 기간 타이머 초기화 - 다음에 사용자가 파일 또는 블록을 할당하지만 여전히 소프트 제한을 넘으면 유예 기간 타이머가 재설정되고 유예 기간이 다시 시작됩니다.

**코드 예 5-12**는 타이머를 초기화하여 `sam` 그룹의 사용자가 다음에 `/sam6`에서 블록 또는 파일 할당을 시도할 때 카운트다운이 시작되도록 하는 명령을 표시합니다.

코드 예 5-12 타이머 초기화

```
# samquota -G sam -x clear /sam6
Setting Grace Timer: continue? y
# samquota -G sam /sam6
```

	Type	ID	In Use	Online Limits		In Use	Total Limits	
				Soft	Hard		Soft	Hard
/sam6								
Files	group	101	32	2000	2000	32	2000	2000
Blocks	group	101	41888+	40000	60000000	43208	60000000	60000000
Grace period				1w			1w	

---> Warning: online soft limits to be enforced in 6d23h59m56s

- 유예 기간 타이머 재시작 - 유예 기간이 재설정되면 타이머가 재설정되고 유예 기간이 재시작됩니다.

**코드 예 5-13**은 유예 기간을 재설정합니다.

코드 예 5-13 유예 기간 타이머 재설정

```
# samquota -G sam -x reset /sam6
Setting Grace Timer: continue? y
# samquota -G sam /sam6
```

	Type	ID	In Use	Online Limits		In Use	Total Limits	
				Soft	Hard		Soft	Hard
/sam6								
Files	group	101	32	2000	2000	32	2000	2000
Blocks	group	101	41888	40000	60000000	43208	60000000	60000000
Grace period				1w			1w	

---> Warning: online soft limits to be enforced in 6d23h59m52s

- 유예 기간을 값으로 설정 - 타이머가 값으로 설정되고 해당 값에서 즉시 카운트다운을 시작합니다. 이 값에는 제한이 없습니다. 이 값은 유예 기간보다 클 수 있습니다.

코드 예 5-14는 매우 긴 만료 기간을 설정합니다.

코드 예 5-14      매우 긴 유예 기간 설정

```
# samquota -G sam -x 52w /sam6
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 51w6d23h59m54s

- 유예 기간 타이머 만료 - 타이머가 즉시 만료하도록 설정됩니다.

코드 예 5-15는 유예 기간을 만료합니다.

코드 예 5-15      유예 기간 타이머 만료

```
# samquota -G sam -x expire /sam6
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	

---> Online soft limits under enforcement (since 6s ago)

## ▼ 추가 파일 시스템 자원 할당 방지

파일 시스템에서 할당량 값이 사용자, 그룹 또는 관리자 세트에 대해 일관되지 않음을 감지하는 경우 해당 사용자, 그룹 또는 관리자 세트가 더 이상의 시스템 자원을 사용하지 못하도록 합니다. 일치하지 않는 할당량 값을 만들어 파일 시스템 자원 할당을 방지할 수 있습니다. 예를 들어 하드 블록 또는 파일 제한이 소프트 블록 또는 파일 제한보다 낮은 경우 또는 사용자의 소프트 제한이 사용자의 하드 제한보다 큰 경우 더 이상의 할당을 금지할 수 있습니다.

파일 시스템은 일치하지 않는 할당량 설정을 특수 할당량으로 취급합니다. 일치하지 않는 할당량 값은 사용자, 그룹 또는 관리자 세트 ID 할당량 파일의 레코드 0에서 설정할 수 있고, 이 경우 이러한 값은 새로운 사용자, 그룹 또는 관리자 세트 ID에 대한 기본값이 됩니다.

다음 절차는 사용자, 그룹 또는 관리자 세트에 대한 더 이상의 시스템 자원 할당을 방지하는 방법입니다.

1. 슈퍼유저가 됩니다.
2. 현재의 할당량 정보를 검색, 저장 및 검토합니다.

**코드 예 5-16**은 `samquota(1M)` 명령을 사용하여 `sam` 그룹에 대한 현재의 그룹 할당량 정보를 검색하고 백업 파일에 쓰는 방법을 표시합니다.

코드 예 5-16          그룹 할당량 정보 검색

```
# 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
```

사용자 할당량에 대한 할당량 정보를 확인하려면 `-G` 옵션 위치에 `e -U userID` 옵션을 지정하십시오. 관리자 세트 할당량에 대한 할당량 정보를 확인하려면 `-G` 옵션 위치에 `-A adminID` 옵션을 지정하십시오.

3. `samquota(1M)` 명령을 사용하여 소프트 할당량을 0이 아닌 할당량으로, 하드 할당량을 0 할당량으로 설정합니다.

다음 명령은 `sam` 그룹에 대한 할당량을 일관되지 않게 설정합니다.

```
# samquota -Gsam -f 1:s -f 0:h -b 1:s -b 0:h /sam6
```

사용자 또는 관리자 세트에 대한 할당량을 일관되지 않게 하려면 `-G` 옵션 대신 `-U userID` 또는 `-A adminID` 옵션을 지정합니다.

4. **samquota(1M)** 명령을 사용하여 변경 사항을 확인합니다.

[코드 예 5-17](#)을 참조하십시오.

코드 예 5-17      변경된 할당량 확인

```
# samquota -G sam /sam6

                Online Limits                Total Limits
                Soft    Hard                Soft    Hard
Type    ID    In Use
/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.
```

위의 출력에서 0 할당량이 설정되어 있습니다. 출력에서 과다 할당량 조건을 표시하는 느낌표(!)를 주목하십시오.

5. **sh(1)** 및 **samquota(1M)** 명령을 사용하여 그룹의 할당량을 파일/블록 할당이 금지되기 전의 할당량으로 복원한 후 변경된 할당량을 확인합니다.

[코드 예 5-18](#)은 이러한 명령을 나타낸 것입니다.

코드 예 5-18      그룹 할당량 복원

```
# sh restore.quota.sam
Setting Grace Timer: continue? y
Setting Grace Timer: continue? y
# samquota -G sam /sam6

                Online Limits                Total Limits
                Soft    Hard                Soft    Hard
Type    ID    In Use
/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
```

사용자 할당량에 대해 이 작업을 수행하려면 -G 옵션 대신 -U *userID* 옵션을 지정하십시오. 관리자 세트 할당량에 대해 이 작업을 수행하려면 -G 옵션 대신 -A *adminID* 옵션을 지정하십시오.

## ▼ 파일 시스템의 할당량 제거

파일 시스템에 대한 할당량을 제거하거나 비활성화하려면 마운트 프로세스에서 할당량을 비활성화해야 합니다.

1. **su(1)** 명령을 사용하여 슈퍼유저로 전환합니다.
2. (선택 사항) 텍스트 편집기를 사용하여 `/etc/vfstab` 또는 `samfs.cmd` 파일에 `noquota` 마운트 옵션을 추가합니다.  
다른 방법으로는 나중에 마운트 명령을 실행할 때 `noquota`를 옵션으로 지정할 수 있습니다. 4단계를 참조하십시오.
3. 파일 시스템이 마운트된 경우에는 **umount(1M)** 명령을 사용하여 파일 시스템을 마운트 해제하십시오.  
예를 들어, 다음과 같습니다.

```
# umount /myfs
```

파일 시스템을 마운트 해제하는 데 문제가 있는 경우 [62페이지의 "파일 시스템 마운트 해제"](#)를 참조하십시오.

4. **mount(1M)** 명령을 사용하여 파일 시스템을 다시 마운트하십시오.  
[2단계](#)를 수행하지 않은 경우, `mount(1M)` 명령에 `noquota` 옵션을 포함하십시오.  
예를 들어, 다음과 같습니다.

```
# mount -o noquota /myfs
```

5. 다음 중 하나를 사용하여 할당량 파일을 불필요하게 합니다.
  - 할당량 기능을 나중에 다시 회복시킬 것으로 예상하고 따라서 할당량 파일을 파괴하지 않으려는 경우, 파일 시스템을 마운트 해제하고 파일 시스템에서 `samfsck(1M)` 명령을 `-F` 옵션과 함께 실행하고 `noquota` 마운트 옵션을 제거한 채로 다시 파일 시스템을 마운트하십시오.
  - 나중에 할당량 기능을 복원하지 않거나 할당량 파일에 의해 사용되는 공간을 활용하려면 `rm(1)` 명령을 사용하여 `.quota_u`, `.quota_g` 및 `.quota_a` 파일을 제거하십시오. 예를 들어, 다음과 같습니다.

```
# rm /myfs/.quota_[agu]
```

## ▼ 할당량 정정

1. 슈퍼유저가 됩니다.
2. 파일 시스템이 마운트된 경우에는 **umount(1M)** 명령을 사용하여 파일 시스템을 마운트 해제하십시오.

예를 들어, 다음과 같습니다.

```
# umount /myfs
```

파일 시스템을 마운트 해제하는 데 문제가 있는 경우 [62페이지의 "파일 시스템 마운트 해제"](#)를 참조하십시오.

3. **samfsck(1M) -F** 명령을 사용하여 파일 시스템 검사를 수행합니다.  
**samfsck(1M)** 명령은 정확한 현재의 사용량 정보로 할당량 파일에서 할당된 레코드를 업데이트합니다.

예를 들어, 다음과 같습니다.

```
# samfsck -F myfs
```

4. **mount(1M)** 명령을 사용하여 파일 시스템을 다시 마운트합니다.

예를 들어, 다음과 같습니다.

```
# mount /myfs
```

# Sun Cluster 환경의 Sun StorageTek QFS 구성

이 장에서는 Sun StorageTek QFS 소프트웨어가 Sun Cluster 환경에서 작동하는 방법에 대해 설명합니다. 또한 Sun Cluster 환경의 Sun StorageTek QFS 공유 파일 시스템 및 Sun Cluster 환경의 공유되지 않은 Sun StorageTek QFS 파일 시스템에 대한 구성 예제를 제공합니다.

본 장에는 다음 절이 포함되어 있습니다.

- 120페이지의 "시작하기 전에"
- 121페이지의 "제한사항"
- 122페이지의 "Sun Cluster 시스템과 Sun StorageTek QFS 소프트웨어의 상호 작용 방법"
- 123페이지의 "Sun Cluster용 Solaris Volume Manager의 Sun StorageTek QFS 지원"
- 129페이지의 "구성 예제"
- 131페이지의 "Sun Cluster 환경에서의 Sun StorageTek QFS 공유 파일 시스템 구성"
- 140페이지의 "Sun Cluster 환경에서 공유 해제된 파일 시스템 구성"
- 161페이지의 "클러스터 외부에 공유 클라이언트 구성"
- 172페이지의 "Sun StorageTek QFS 구성 변경"
- 178페이지의 "Sun Cluster를 사용한고가용성 Sun StorageTek SAM 구성"

# 시작하기 전에

Sun StorageTek QFS 소프트웨어의 버전 4U2 이상을 사용하여 Sun Cluster 환경에서 Sun StorageTek QFS 파일 시스템을 설치하고 고가용성으로 파일 시스템을 구성할 수 있습니다. 사용하는 구성 방법은 파일 시스템이 공유 또는 공유되지 않는지 여부에 따라서 다릅니다.

이 장은 사용자가 Sun StorageTek QFS 소프트웨어 및 Sun Cluster 환경 둘 다의 숙련된 사용자라고 가정합니다. 또한 다음 중 하나 또는 모두를 수행했다고 가정합니다.

- 파일 시스템을 고가용성으로 확장 가능하게 구성했거나 Sun Cluster 소프트웨어 제어하에 장애 조치 자원을 구성했습니다.
- Sun StorageTek QFS 독립형 및 Sun StorageTek QFS 공유 파일 시스템을 모두 설치하고 구성했습니다.

이 장을 계속하기 전에 다음 설명서를 읽는 것이 좋습니다.

- Sun StorageTek QFS 설치 및 업그레이드 안내서의 다음 장:
  - 2장, Sun Cluster 환경의 Sun StorageTek QFS 소프트웨어에 대한 하드웨어 및 소프트웨어 전제조건 정보를 제공합니다.
  - 4장, 고가용성을 위해 Sun Cluster 시스템에 Sun StorageTek QFS 파일 시스템을 설치하는 추가 작업에 대해 설명합니다.
- 다음 Sun Cluster 소프트웨어 설명서:
  - Sun Cluster Concepts Guide for Solaris OS - Sun Cluster 환경을 소개합니다. 특히 다음 절을 검토하십시오.

로컬 디스크

전역 장치

장치 ID(DID)

디스크 장치 그룹

디스크 장치 그룹 장애 복구

로컬 및 전역 이름공간

클러스터 파일 시스템

HAStoragePlus 자원 유형

볼륨 관리자

- Sun Cluster Software Installation Guide for Solaris OS, Sun Cluster 소프트웨어 설치 단계에 대해 설명합니다
- Sun Cluster Data Services Planning and Administration Guide for Solaris OS, 다양한 데이터 서비스를 포함하기 위해 Sun Cluster 환경을 계획하는 방법에 대해 설명합니다

- Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS, Oracle Real Application Clusters(RAC)용 Sun Cluster 데이터 서비스와 함께 Sun StorageTek QFS 공유 파일 시스템을 사용하는 방법에 대해 설명합니다.

---

주 - File System Manager 소프트웨어를 사용하여 Sun Cluster 환경의 파일 시스템을 제어할 수도 있습니다. 서버를 추가할 때 클러스터 노드를 인식 및 식별하며 다른 클러스터 노드를 추가하라는 메시지를 사용자에게 자동으로 표시합니다. 사용자에게 Sun Cluster 구성 내의 노드에 비아카이빙 HA(Highly Available) 공유 또는 독립형 Sun StorageTek QFS 파일 시스템을 만들 수 있는 옵션이 제공됩니다. 자세한 내용은 File System Manager 온라인 도움말을 참조하십시오.

---

## 제한사항

다음 제한 사항이 Sun Cluster 환경의 Sun StorageTek QFS 소프트웨어에 적용됩니다.

- 다음 제한 사항이 공유 파일 시스템에 적용됩니다.
  - SUNW.qfs 자원 유형에서만 구성할 수 있습니다. HAStoragePlus 자원 유형에서는 구성할 수 없습니다.
  - Oracle Real Application Clusters용 Sun Cluster 데이터 서비스에 의해서만 사용되는 확장 가능한 파일 시스템으로 구성할 수 있습니다.
- 다음 제한 사항이 공유 해제된 파일 시스템에 적용됩니다.
  - HAStoragePlus 자원 유형에서만 구성할 수 있습니다. SUNW.qfs 자원 유형에서는 구성할 수 없습니다.
  - Sun Cluster 소프트웨어에서 제공되는 모든 장애 복구 응용 프로그램이 공유되지 않은 Sun StorageTek QFS 소프트웨어에서 고가용성의 로컬 파일 시스템으로 지원됩니다. 지원이 특정 응용 프로그램으로 제한되지 않습니다.
  - 공유되지 않은 Sun StorageTek QFS 파일 시스템이 Sun Cluster 시스템의 기초적인 고유 파일 시스템일 수 없습니다.

---

주 - 페일백은 SUNW.qfs 에이전트의 기능으로 지원되지 않습니다.

---

---

주 - Sun StorageTek QFS 파일 시스템을 Sun Cluster 환경에 설치하면 신뢰성을 개선하고 예기치 못한 중단 시간을 줄이거나 제거할 수 있지만, 계획된 중단 시간을 없애지는 못합니다. 파일 시스템의 상태를 유지하기 위해때때로 Sun StorageTek QFS 소프트웨어를 중단하여 samfsck 프로세스를 실행해야 할 수 있습니다. 또한 소프트웨어 패치나 업데이트를 적용하기 위해 시스템을 종료해야 합니다.

---

# Sun Cluster 시스템과 Sun StorageTek QFS 소프트웨어의 상호 작용 방법

공유 파일 시스템은 Sun Cluster 디스크 ID(DID) 지원을 사용하여 Oracle Real Application Clusters용 Sun Cluster 데이터 서비스의 데이터 액세스를 활성화합니다. 공유 해제된 파일 시스템은 전역 장치 볼륨 지원과 볼륨 관리자 제어 볼륨 지원을 사용하여 Sun Cluster 시스템이 지원하는 장애 조치 응용 프로그램의 데이터 액세스를 활성화합니다.

## 공유 파일 시스템의 데이터 액세스

Sun Cluster 시스템의 제어 하에 있는 각 장치는 다중 경로를 갖는지 여부와 상관 없이 DID 지원을 사용하여 고유한 DID가 지정됩니다. 모든 고유한 DID 장치에 대해 해당되는 전역 장치가 있습니다. Sun StorageTek QFS 공유 파일 시스템은 DID 장치(/dev/did/\*) 만으로 구성되는 중복 저장소에 구성될 수 있습니다. 여기서 DID 장치는 호스트 버스 어댑터(HBA)를 통해 장치에 직접 연결되는 노드에서만 액세스할 수 있습니다.

Sun StorageTek QFS 공유 파일 시스템을 DID 장치에 구성하고 해당 파일 시스템과 함께 사용할 SUNW.qfs 자원 유형을 구성하면 파일 시스템의 공유 메타 데이터 서버가 높은 가용성을 갖게 됩니다. 그러면 Oracle Real Application Clusters용 Sun Cluster 데이터 서비스가 파일 시스템 안에서 데이터에 액세스할 수 있습니다. 또한 Sun StorageTek QFS Sun Cluster 에이전트가 필요할 때 자동으로 파일 시스템에 대한 메타 데이터 서버를 다시 찾을 수 있습니다.

---

주 - Sun StorageTek QFS 소프트웨어의 버전 4U6부터는 Sun Cluster 환경에서 클러스터 외부에 공유 클라이언트를 둘 수도 있습니다. 전체 구성 지침은 [161페이지의 "클러스터 외부에 공유 클라이언트 구성"](#)을 참조하십시오.

---

## 공유되지 않는 파일 시스템의 데이터 액세스

전역 장치는 DID 장치를 호스트하는 노드가 사용 가능하다고 가정할 때 Sun Cluster 시스템의 임의의 노드에서 기초 DID 장치에 액세스하기 위한 Sun Cluster 시스템의 메커니즘입니다. 전역 장치 및 볼륨 관리자 제어 볼륨은 Sun Cluster 시스템의 모든 노드에서 액세스할 수 있습니다. 공유되지 않는 Sun StorageTek QFS 파일 시스템은 원시 전역 장치(/dev/global/\*) 또는 볼륨 관리자 제어 볼륨 중 하나로 구성되는 중복 기억장치에 구성할 수 있습니다.

이러한 전역 장치 또는 볼륨 관리자 제어 장치에 공유되지 않는 파일 시스템을 구성하고 해당 파일 시스템과 함께 사용할 HASToragePlus 자원 유형을 구성하면 파일 시스템이 다른 노드로 장애 복구하는 기능으로 가용성이 높아집니다.

---

# Sun Cluster용 Solaris Volume Manager의 Sun StorageTek QFS 지원

Sun StorageTek QFS의 4U4 릴리스에는 Sun Cluster용 Solaris Volume Manager에 대한 지원이 추가되었습니다. 이는 Solaris 9 및 Solaris 10 OS 릴리스와 함께 제공되는 Solaris™ Volume Manager에 대한 확장입니다. Sun StorageTek QFS는 Solaris 10에서 Sun Cluster용 Solaris Volume Manager만 지원합니다.

Sun Cluster용 Solaris Volume Manager에 대한 Sun StorageTek QFS 지원은 공유 Sun StorageTek QFS 호스트 기반 미러링뿐만 아니라 Oracle의 Oracle RAC 기반 응용 프로그램을 위한 응용 프로그램 바이너리 복구(ABR) 및 직접 미러 읽기(DMR) 구현을 활용하기 위해 도입되었습니다.

Sun StorageTek QFS가 있는 Sun Cluster용 Solaris Volume Manager를 사용하려면 Sun Cluster 소프트웨어와 Sun Cluster 소프트웨어에 포함된 번들되지 않은 소프트웨어 패키지가 필요합니다.

Sun Cluster용 Solaris Volume Manager 지원이 추가되면서 네 가지 새로운 마운트 옵션이 소개되었습니다. 마운트 옵션은 Sun StorageTek QFS가 Sun Cluster용 Solaris Volume Manager에 해당 옵션이 구성되었음을 감지하는 경우에만 사용할 수 있습니다. 마운트 옵션은 다음과 같습니다.

- abr - 응용 프로그램 바이너리 복구 활성화
- dmr - 직접 미러 읽기 활성화
- noabr - 응용 프로그램 바이너리 복구 비활성화
- nodmr - 직접 미러 읽기 비활성화

다음은 Sun Cluster용 Solaris Volume Manager가 있는 Sun StorageTek QFS를 사용하기 위한 구성 예제입니다.

아래 예제에서 다음 구성이 완료된 것으로 가정합니다.

- Sun Cluster가 설치되었으며 설치 모드를 종료했습니다.
- Sun Cluster Oracle RAC 프레임워크를 활성화하는 추가 패키지가 설치되었습니다.
- Sun Cluster용 Solaris Volume Manager를 활성화하는 추가 Sun Cluster SUNWscmd 패키지가 설치되었습니다.
- Oracle RAC Framework Resource Group을 만들었으며 온라인입니다.

이 예제에는 세 개의 공유 Sun StorageTek QFS 파일 시스템이 있습니다.

- **CRS** - 이 파일 시스템은 Oracle RAC CRS(Cluster-ready Service)의 Oracle 10G 릴리스 구성에 필요합니다. 자세한 내용은 Sun Cluster Concepts를 참조하십시오.
- **Data** - 이 파일 시스템은 Oracle 관련 파일(Oracle 시스템 파일, Oracle 제어 파일 및 Oracle 데이터 파일)을 보관하는 데 사용됩니다.
- **Redo** - 이 파일 시스템은 Oracle Redo, Oracle Archive 및 Flash Back 파일 및 로그를 포함하는 데 사용됩니다.

## ▼ Sun Cluster용 Solaris Volume Manager가 있는 파일 시스템 구성

1. 각 노드에 `metadb`를 만듭니다.

예를 들어, 다음과 같습니다.

```
# metadb -a -f -c3 /dev/rdsk/c0t0d0s7
```

2. 하나의 노드에 디스크 그룹을 만듭니다.

예를 들어, 다음과 같습니다.

```
# metaset -s datadg -M -a -h scNode-A scNode-B
```

3. `sccdidadm`를 실행하여 하나의 노드에서 장치를 구합니다.

예를 들어, 다음과 같습니다.

```
scNode-A # sccdidadm -l
13 scNode-A:/dev/rdsk/c6t600C0FF000000000000332B62CF3A6B00d0 /dev/did/rdsk/d13
14 scNode-A:/dev/rdsk/c6t600C0FF000000000000876E950F1FD9600d0 /dev/did/rdsk/d14
15 scNode-A:/dev/rdsk/c6t600C0FF000000000000876E9124FAF9C00d0 /dev/did/rdsk/d15
16 scNode-A:/dev/rdsk/c6t600C0FF000000000000332B28488B5700d0 /dev/did/rdsk/d16
17 scNode-A:/dev/rdsk/c6t600C0FF00000000000086DB474EC5DE900d0 /dev/did/rdsk/d17
18 scNode-A:/dev/rdsk/c6t600C0FF000000000000876E975EDA6A000d0 /dev/did/rdsk/d18
19 scNode-A:/dev/rdsk/c6t600C0FF00000000000086DB47E331ACF00d0 /dev/did/rdsk/d19
20 scNode-A:/dev/rdsk/c6t600C0FF000000000000876E9780ECA8100d0 /dev/did/rdsk/d20
21 scNode-A:/dev/rdsk/c6t600C0FF00000000000004CAD5B68A7A100d0 /dev/did/rdsk/d21
22 scNode-A:/dev/rdsk/c6t600C0FF00000000000086DB43CF85DA800d0 /dev/did/rdsk/d22
23 scNode-A:/dev/rdsk/c6t600C0FF0000000000004CAD7CC3CDE500d0 /dev/did/rdsk/d23
24 scNode-A:/dev/rdsk/c6t600C0FF00000000000086DB4259B272300d0 /dev/did/rdsk/d24
25 scNode-A:/dev/rdsk/c6t600C0FF000000000000332B21D0B90000d0 /dev/did/rdsk/d25
26 scNode-A:/dev/rdsk/c6t600C0FF0000000000004CAD139A855500d0 /dev/did/rdsk/d26
27 scNode-A:/dev/rdsk/c6t600C0FF000000000000332B057D2FF100d0 /dev/did/rdsk/d27
28 scNode-A:/dev/rdsk/c6t600C0FF0000000000004CAD4C40941C00d0 /dev/did/rdsk/d28
```

미러링 체계는 다음과 같습니다.

21 <-> 13

14 <-> 17

23 <-> 16

15 <-> 19

4. 하나의 노드에 있는 세트에 장치를 추가합니다.

예를 들어, 다음과 같습니다.

```
# metaset -s datadg -a /dev/did/rdisk/d21 /dev/did/rdisk/d13
/dev/did/rdisk/d14 \
/dev/did/rdisk/d17 /dev/did/rdisk/d23 /dev/did/rdisk/d16
/dev/did/rdisk/d15 \
/dev/did/rdisk/d19
```

5. 하나의 노드에 미러를 만듭니다.

예를 들어, 다음과 같습니다.

```
metainit -s datadg d10 1 1 /dev/did/dsk/d21s0
metainit -s datadg d11 1 1 /dev/did/dsk/d13s0
metainit -s datadg d1 -m d10
metattach -s datadg d11 d1

metainit -s datadg d20 1 1 /dev/did/dsk/d14s0
metainit -s datadg d21 1 1 /dev/did/dsk/d17s0
metainit -s datadg d2 -m d20
metattach -s datadg d21 d2

metainit -s datadg d30 1 1 /dev/did/dsk/d23s0
metainit -s datadg d31 1 1 /dev/did/dsk/d16s0
metainit -s datadg d3 -m d30
metattach -s datadg d31 d3

metainit -s datadg d40 1 1 /dev/did/dsk/d15s0
metainit -s datadg d41 1 1 /dev/did/dsk/d19s0
metainit -s datadg d4 -m d40
metattach -s datadg d41 d4
```

```

metainit -s datadg d51 -p d1 10m
metainit -s datadg d52 -p d1 200m
metainit -s datadg d53 -p d1 800m

metainit -s datadg d61 -p d2 10m
metainit -s datadg d62 -p d2 200m
metainit -s datadg d63 -p d2 800m

metainit -s datadg d71 -p d1 500m
metainit -s datadg d72 -p d1 65g

metainit -s datadg d81 -p d2 500m
metainit -s datadg d82 -p d2 65g

```

6. 각 노드에서 **Sun StorageTek QFS** 설치를 수행합니다.  
 예를 들어, 다음과 같습니다.

```

pkgadd -d . SUNWqfsr SUNWqfsu

```

7. 각 노드에 `mcf` 파일을 만듭니다.  
 예를 들어, 다음과 같습니다.

```

/etc/opt/SUNWsamfs/mcf 파일
#
# File system Data
#
Data          2    ma  Data  on    shared
/dev/md/datadg/dsk/d53 20  mm  Data  on
/dev/md/datadg/dsk/d63 21  mm  Data  on
/dev/md/datadg/dsk/d3  22  mr  Data  on
/dev/md/datadg/dsk/d4  23  mr  Data  on
#
# File system Crs
#
Crs           4    ma  Crs   on    shared
/dev/md/datadg/dsk/d51 40  mm  Crs   on
/dev/md/datadg/dsk/d61 41  mm  Crs   on
/dev/md/datadg/dsk/d52 42  mr  Crs   on
/dev/md/datadg/dsk/d62 43  mr  Crs   on
#

```

```
# File system Redo
#
Redo          6    ma   Redo   on    shared
/dev/md/datadg/dsk/d71  60  mm   Redo   on
/dev/md/datadg/dsk/d81  61  mm   Redo   on
/dev/md/datadg/dsk/d72  62  mr   Redo   on
/dev/md/datadg/dsk/d82  63  mr   Redo   on
```

8. 파일 시스템 호스트 파일을 만듭니다.

예를 들어, 다음과 같습니다.

```
/etc/opt/SUNWsamfs/hosts.Data
/etc/opt/SUNWsamfs/hosts.Crs
/etc/opt/SUNWsamfs/hosts.Oracle

# scNode-A:root> /usr/cluster/bin/scconf -p |egrep "Cluster node
name:|Node private hostname:"
      Cluster node name:      scNode-A
      Node private hostname:  clusternode1-priv
      Cluster node name:      scNode-B
      Node private hostname:  clusternode2-priv

# Host          Host IP          Server  Not  MDS Server
# Name          Address          Priority Used  Host
#-----
scNode-A        clusternode1-priv  1      -    server
scNode-B        clusternode2-priv  2      -
```

9. /etc/opt/SUNWsamfs/samfs.cmd 파일을 만듭니다.

예를 들어, 다음과 같습니다.

```
fs = Data
  stripe=1
  sync_meta=1
  mh_write
  qwrite
  forcedirectio
  notrace
  rdlease=300
  wrlease=300
  aplease=300
```

```

fs = Crs
    stripe=1
    sync_meta=1
    mh_write
    qwrite
    forcedirectio
    notrace
    rdlease=300
    wrlease=300
    aplease=300

fs = Redo
    stripe=1
    sync_meta=1
    mh_write
    qwrite
    forcedirectio
    notrace
    rdlease=300
    wrlease=300
    aplease=300

```

10. **Sun StorageTek QFS** 파일 시스템을 만듭니다. 자세한 내용은 Sun StorageTek QFS 설치 및 업그레이드 안내서를 참조하십시오.

예를 들어, 다음과 같습니다.

```

/opt/SUNWsamfs/sbin/sammkfs -S <filesystem>

```

11. **Sun StorageTek QFS** 메타 데이터 서버의 장애 조치를 관리하도록 **Sun Cluster**에 자원 그룹을 구성합니다.

- a. /etc/vfstab 마운트 항목을 구축하고 추가합니다.

예를 들어, 다음과 같습니다.

```

#
#
# RAC on shared QFS
Data - /cluster/Data samfs - no shared,notrace
Redo - /cluster/Redo samfs - no shared,notrace
Crs - /cluster/Crs samfs - no shared,notrace

```

- b. 각 노드의 클러스터에 파일 시스템을 마운트합니다.

먼저 공유 Sun StorageTek QFS 파일 시스템을 현재 메타 데이터 서버에 마운트한 후 파일 시스템을 각 메타 데이터 클라이언트에 마운트합니다.

이 단계를 확인하려면

```
# df -h -F samfs를 입력합니다.
```

- c. 메타 데이터 서버를 관리할 Sun Cluster 자원 그룹을 만듭니다.

QFS 자원 유형 등록:

```
# scrgadm -a -t SUNW.qfs
```

Sun Cluster 및 공유 Sun StorageTek QFS 메타 데이터 노드가 있는 자원 그룹 추가:

```
# scrgadm -a -g sc-QFS-rg -h scNode-A,scNode-B -y  
RG_DEPENDENCIES="rac-framework-rg"
```

공유 Sun StorageTek QFS 파일 시스템 자원 및 SUNWqfs 자원 유형을 자원 그룹에 추가:

```
# scrgadm -a -g sc-QFS-rg -t SUNW.qfs -j sc-qfs-fs-rs -x  
QFSfileSystem=/cluster/Data, \  
/cluster/Redo,/cluster/Crs
```

자원 그룹을 온라인으로 전환:

```
# scswitch -Z -g sc-QFS-rg
```

공유 Sun StorageTek QFS 파일 시스템을 이제 사용할 수 있습니다.

---

## 구성 예제

이 장에서는 Sun Cluster 시스템에 있는 Sun StorageTek QFS 공유 파일 시스템 및 공유 해제된 Sun StorageTek QFS 파일 시스템에 대한 구성 예제를 제공합니다. 모든 구성 예제는 다음으로 구성되는 플랫폼을 기초로 합니다.

- 두 노드를 갖는 SunPlex™ 시스템(scnode-A 및 scnode-B)
- 고가용성 및 중복성을 위한 다중 호스트 액세스 가능 디스크 장치를 포함하는 DID 장치

이 장의 모든 구성은 또한 [코드 예 6-1](#)을 기초로 합니다. 이 코드 예제에서 `scdidadm(1M)` 명령은 DID 장치를 표시하고 `-L` 옵션은 Sun Cluster 시스템의 모든 노드에 대한 경로를 포함하여 DID 장치 경로를 나열합니다.

코드 예 6-1

DID 장치 및 해당 DID 장치 경로를 나열하는 명령

```
# 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
```

[코드 예 6-1](#)은 DID 장치 d4 - d8이 두 Sun Cluster 시스템(scnode-A 및 scnode-B) 모두에서 액세스할 수 있음을 표시합니다. Sun StorageTek QFS 파일 시스템 크기 요구 사항 및 원하는 응용 프로그램 및 구성에 대한 지식으로 파일 시스템에 대한 가장 적절한 장치 할당을 결정할 수 있습니다. `Solaris format(1M)` 명령을 사용하여 각 DID 장치의 크기 및 분할 영역 배치를 판별하고 필요한 경우 각 DID 장치의 분할 영역 크기를 변경할 수 있습니다. 사용 가능한 DID 장치가 주어질 때 자신의 크기 요구 사항에 따라서 파일 시스템을 포함할 복수 장치 및 그들의 연관된 분할 영역을 구성할 수도 있습니다.

---

# Sun Cluster 환경에서의 Sun StorageTek QFS 공유 파일 시스템 구성

Sun Cluster 환경에 Sun StorageTek QFS 공유 파일 시스템을 설치할 때 SUNW.qfs 자원 유형 하에 파일 시스템의 메타 데이터 서버를 구성합니다. 이 구성이 메타 데이터 서버의 가용성을 높이며 Sun StorageTek QFS 공유 파일 시스템이 Sun Cluster 환경에 구성된 모든 노드에서 전역으로 액세스할 수 있게 합니다.

Sun StorageTek QFS 공유 파일 시스템은 일반적으로 확장 가능한 응용 프로그램과 연관됩니다. Sun StorageTek QFS 공유 파일 시스템은 하나 이상의 Sun Cluster 노드에 마운트되고 확장 가능한 응용 프로그램이 그 위에서 활성화됩니다.

Sun Cluster 시스템의 한 노드가 실패하거나 사용자가 자원 그룹을 전환하는 경우 메타 데이터 서버 자원(Sun StorageTek QFS Sun Cluster 에이전트)이 필요할 때 자동으로 파일 시스템의 메타 데이터 서버를 다시 찾습니다. 이것은 공유 파일 시스템에 대한 다른 노드의 액세스가 영향을 받지 않게 합니다.

---

주 - Sun Cluster 시스템 제어 하에 있는 Sun StorageTek QFS 공유 파일 시스템에 대한 메타 데이터 서버를 수동으로 다시 찾으려면 Sun Cluster 관리 명령을 사용해야 합니다. 이러한 명령에 대한 자세한 내용은 Sun Cluster 문서를 참조하십시오.

---

## 메타 데이터 서버 자원 고려사항

Sun Cluster 시스템이 부트할 때 메타 데이터 서버 자원은 파일 시스템이 자원 그룹의 일부인 모든 노드에 마운트되도록 합니다. 그러나 해당 노드의 파일 시스템 마운트는 모니터되지 않습니다. 그러므로 실패하는 경우 메타 데이터 서버 자원이 온라인 상태에 있는 경우에도 특정 노드에서 파일 시스템이 사용 불가능할 수 있습니다.

Sun Cluster 관리 명령을 사용하여 메타 데이터 서버 자원 그룹을 오프라인으로 만드는 경우 메타 데이터 서버 자원 아래의 파일 시스템은 계속 노드에 마운트되어 있습니다. 파일 시스템을 마운트 해제하려면(시스템 종료된 노드는 예외) 적절한 Sun Cluster 관리 명령을 사용하여 메타 데이터 서버 자원 그룹을 관리되지 않은 상태로 만들어야 합니다.

나중에 파일 시스템을 다시 마운트하려면 자원 그룹을 관리 상태로 만든 후 온라인 상태로 만들어야 합니다.

## 구성 예제

이 절은 Oracle Real Application Clusters용 Sun Cluster 데이터 서비스로 원시 DID 장치에 설치된 Sun StorageTek QFS 공유 파일 시스템의 예제를 표시합니다. Oracle Real Application Clusters용 Sun Cluster 데이터 서비스와 함께 Sun StorageTek QFS 공유 파일 시스템을 사용하는 방법에 대한 자세한 정보는 Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS를 참조하십시오.

코드 예 6-1에 표시된 것처럼 DID 장치 d4 - d8은 높은 가용성을 가지며 제어기 기반 기억장치에 포함되어 있습니다. Sun Cluster 환경에 Sun StorageTek QFS 공유 파일 시스템을 구성하려면 제어기 기반 저장소가 RAID-1 또는 RAID-5를 사용하여 장치 중복성을 지원해야 합니다.

이 예제에서는 간단하게 다음 두 파일 시스템이 작성됩니다.

- qfs1 - 이 파일 시스템은 Oracle Real Application Clusters 공유 설치, 구성 및 로그 파일에 사용됩니다.
- qfs2 - 이 파일 시스템은 Oracle Real Application Clusters 소프트웨어가 공유하는 데이터베이스 파일에 사용됩니다.

또한 d4 장치가 Sun StorageTek QFS 메타 데이터에 사용됩니다. 이 장치는 2개의 50GB 슬라이스를 갖습니다. 나머지 장치인 d5부터 d8까지는 Sun StorageTek QFS 파일 데이터에 사용됩니다.

이 구성은 다음 부속절에서 자세히 설명하는 것처럼 5가지 기본 단계를 포함합니다.

1. Sun StorageTek QFS 파일 시스템 작성 준비
2. 파일 시스템 작성 및 Sun Cluster 노드 구성
3. 구성 검증
4. 네트워크 이름 서비스 구성
5. Oracle Real Application Clusters용 Sun Cluster 데이터 서비스 구성

## ▼ Sun StorageTek QFS 공유 파일 시스템 작성 준비

1. Sun Cluster 시스템의 한 노드에서 **format(1M)** 유틸리티를 사용하여 /dev/did/dsk/d4(코드 예 6-2)에 분할 영역을 배치합니다.  
이 예제에서는 해당 작업이 scnode-A 노드에서 수행됩니다.

코드 예 6-2            /dev/did/dsk/d4에 분할 영역 배치

```
# format /dev/did/rdisk/d4s2
# format> partition
[ output deleted ]
# 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

NOTE: Partition 2 (backup) will not be used and was created by format(1M) by
default.
```

분할 영역(또는 슬라이스) 0은 볼륨의 볼륨 목차(VTOC)를 건너뛰며 50GB 분할 영역으로 구성됩니다. 분할 영역 1은 분할 영역 0과 동일한 크기를 갖도록 구성됩니다.

2. 동일한 노드에서 **format(1M)** 유틸리티를 사용하여 /dev/did/dsk/d5(코드 예 6-3)에 분할 영역을 배치합니다.

```
# format /dev/did/rdisk/d5s2
# format> partition
[ output deleted ]
# 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

NOTE: Partition 2 (backup) will not be used and was created by format(1M) by
default.
```

3. 계속 동일한 노드에서 d5 분할을 d6에서 d8까지 복제합니다.

이 예제는 장치 d6에 대한 명령을 표시합니다.

```
# prtvtoc /dev/did/rdisk/d5s2 | fmthard -s - /dev/did/rdisk/d6s2
```

4. 파일 시스템의 가능한 호스트인 모든 노드에서 다음을 수행합니다.

- a. mcf 파일에 2개의 새로운 구성 항목(qfs1 및 qfs2)을 추가하여 6개 분할 영역을 2개의 **Sun StorageTek QFS** 공유 파일 시스템으로 구성합니다(코드 예 6-4).

코드 예 6-4

mcf 파일에 구성 항목 추가

```
# cat >> /etc/opt/SUNWsamfs/mcf <<EOF
#
# Sun StorageTek QFS file system configurations
#
# Equipment      Equipment      Equipment      Family      Device      Additional
# Identifier      Ordinal        Type           Set         State       Parameters
# -----
qfs1              100           ma            qfs1        -           shared
/dev/did/dsk/d4s0 101           mm            qfs1        -
/dev/did/dsk/d5s0 102           mr            qfs1        -
/dev/did/dsk/d6s0 103           mr            qfs1        -
qfs2              200           ma            qfs2        -           shared
/dev/did/dsk/d4s1 201           mm            qfs2        -
/dev/did/dsk/d7s0 202           mr            qfs2        -
/dev/did/dsk/d8s0 203           mr            qfs2        -
EOF
```

mcf 파일에 대한 자세한 내용은 34페이지의 "mcf 파일의 기능" 또는 Sun StorageTek QFS 설치 및 업그레이드 안내서를 참조하십시오.

- b. /etc/opt/SUNWsamfs/samfs.cmd 파일을 편집하여 **Oracle Real Application Clusters용 Sun Cluster** 데이터 서비스에 필요한 마운트 옵션을 추가합니다 (코드 예 6-5).

코드 예 6-5

samfs.cmd 파일 예제

```
fs = qfs2
  stripe = 1
  sync_meta = 1
  mh_write
  qwrite
  forcedirectio
  rdlease = 300
```

Oracle Real Application Clusters용 Sun Cluster 데이터 서비스에 필요한 마운트 옵션에 대한 자세한 내용은 Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS를 참조하십시오.

- c. 구성이 올바른지 검증합니다.

각 노드의 mcf 파일 및 samfs.cmd 파일을 구성한 후에 이 검증을 수행하십시오.

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

## ▼ Sun StorageTek QFS 공유 파일 시스템 작성 및 Sun Cluster 노드 구성

작성하려는 각 파일 시스템에 대해 이 절차를 수행하십시오. 이 예제는 qfs1 파일 시스템을 작성하는 방법에 대해 설명합니다.

1. 다음 명령을 사용하여 Sun Cluster 개인용 상호 연결 이름을 확보합니다.

```
# /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. 파일 시스템의 가능한 호스트인 각 노드에서 다음을 수행합니다.

- a. `samd(1M) config` 명령을 사용하는데, 이 명령은 Sun StorageTek QFS 데몬에 새로운 Sun StorageTek QFS 구성이 사용 가능하다고 신호로 알려줍니다.

```
# samd config
```

- b. 1단계에서 확보한 Sun Cluster 시스템의 개인용 상호 연결 이름을 기초로 파일 시스템에 대한 Sun StorageTek QFS 공유 호스트 파일 (`/etc/opt/SUNWsamfs/hosts.family-set-name`)을 작성합니다.

3. Sun Cluster 시스템의 상호 연결 이름으로 고유한 Sun StorageTek QFS 공유 파일 시스템의 호스트 구성 파일을 편집합니다(코드 예 6-6).

Sun Cluster 소프트웨어 장애 조치 및 펜싱(fencing) 작업을 위해 Sun StorageTek QFS 공유 파일 시스템이 Sun Cluster 시스템과 동일한 상호 연결 이름을 사용해야 합니다.

코드 예 6-6

각 파일 시스템의 호스트 구성 파일 편집

```
# 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 시스템의 한 노드에서 `sammkfs(1M)` `-S` 명령을 사용하여 Sun StorageTek QFS 공유 파일 시스템을 작성합니다.

```
# sammkfs -S qfs1 < /dev/null
```

5. 파일 시스템의 가능한 호스트인 각 노드에서 다음을 수행합니다.

- a. `mkdir(1M)` 명령을 사용하여 파일 시스템에 대한 전역 마운트 지점을 작성하고, `chmod(1M)` 명령을 사용하여 `root`를 마운트 지점의 소유자로 만들고, `chown(1M)` 명령을 사용하여 읽기/쓰기(755) 액세스 권한을 갖는 다른 사람이 해당 마운트 지점을 사용할 수 있게 합니다.

```
# mkdir /global/qfs1
# chmod 755 /global/qfs1
# chown root:other /global/qfs1
```

- b. `/etc/vfstab` 파일에 Sun StorageTek QFS 공유 파일 시스템 항목을 추가합니다.

```
# 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
```

## ▼ 구성 검증

작성하는 각 파일 시스템에 대해 이 절차를 수행하십시오. 이 예제는 `qfs1` 파일 시스템에 대한 구성 검증 방법에 대해 설명합니다.

1. 파일 시스템에 대한 메타 데이터 서버로 작용 중인 노드를 모르는 경우 `samsharefs(1M)` `-R` 명령을 사용하십시오.

코드 예 6-7에서 `qfs1`용 메타 데이터 서버는 `scnode-A`입니다.

코드 예 6-7                   어떤 노드가 메타 데이터 서버인지 판별

```
# samsharefs -R qfs1
#
# Host file for family set 'qfs1'
#
# Version: 4      Generation: 1      Count: 2
# Server = host 1/scnode-A, length = 165
#
scnode-A clusternode2-priv 1 - server
scnode-B clusternode2-priv 2 -
```

2. `mount(1M)` 명령을 사용하여 먼저 메타 데이터 서버에 파일 시스템을 마운트한 후 **Sun Cluster** 시스템의 각 노드에 마운트합니다.

---

주 - 먼저 메타 데이터 서버에 파일 시스템을 마운트해야 합니다.

---

```
# mount qfs1
# ls /global/qfs1
lost+found/
```

3. 노드 사이에 **Sun StorageTek QFS** 공유 파일 시스템을 변경하는 `samsharefs(1M)` `-s` 명령을 실행하여 자발적인 장애 조치를 검증합니다.

```
# samsharefs -s scnode-B qfs1
# ls /global/qfs1
lost+found/
# samsharefs -s scnode-A qfs1
# ls /global/qfs1
lost+found
```

4. 필수 **Sun Cluster** 자원 유형이 자원 구성에 추가되었는지 검증합니다.

```
# scrgadm -p | egrep "SUNW.qfs"
```

5. **Sun Cluster** 자원 유형을 찾을 수 없는 경우 `scrgadm(1M)` `-a -t` 명령을 사용하여 자원 구성에 해당 자원 유형을 추가합니다.

```
# scrgadm -a -t SUNW.qfs
```

6. `SUNW.qfs` 자원 유형을 등록하고 구성합니다.

```
# 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` 명령을 사용하여 자원 그룹을 온라인으로 만듭니다.

```
# scswitch -Z -g qfs-rg
```

8. 자원 그룹이 구성된 모든 노드에서 기능하는지 확인합니다.

```
# sscswitch -z -g qfs-rg -h scnode-B
# sscswitch -z -g qfs-rg -h scnode-A
```

## ▼ Oracle Real Application Clusters용 Sun Cluster 데이터 서비스 구성

이 절에서는 Sun StorageTek QFS 공유 파일 시스템과 함께 사용할 Oracle Real Application Clusters용 데이터 서비스를 구성하는 방법의 예제를 제공합니다. 자세한 내용은 Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS를 참조하십시오.

1. **Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS**의 설명에 따라 데이터 서비스를 설치합니다.
2. **Sun StorageTek QFS** 공유 파일 시스템을 마운트합니다.
3. **Oracle** 데이터베이스 작업이 성공하도록 파일 시스템에 대한 올바른 소유권과 권한을 설정합니다.

```
# chown oracle:dba /global/qfs1 /global/qfs2
# chmod 755 /global/qfs1 /global/qfs2
```

4. oracle 사용자로서 **Oracle Real Application Clusters** 설치에 필요한 하위 디렉토리 및 데이터베이스 파일을 작성합니다.

```
$ id
uid=120(oracle) gid=520(dba)
$ mkdir /global/qfs1/oracle_install
$ mkdir /global/qfs2/oracle_db
```

Oracle Real Application Clusters 설치하는 /global/qfs1/oracle\_install 디렉토리 경로를 Oracle 작업에서 사용되는 ORACLE\_HOME 환경 변수의 값으로 사용합니다. Oracle Real Application Clusters 데이터베이스 파일의 경로는 /global/qfs2/oracle\_db 디렉토리 경로가 접두어로 붙습니다.

5. **Oracle Real Application Clusters** 소프트웨어를 설치합니다.  
설치 중에 4단계에서 정의한 설치를 위한 경로를 제공합니다 (/global/qfs1/oracle\_install).
6. **Oracle Real Application Clusters** 데이터베이스를 작성합니다.  
데이터베이스 작성 중에 데이터베이스 파일이 qfs2 공유 파일 시스템에 위치되기 원함을 지정하십시오.

7. **Oracle Real Application Clusters** 데이터베이스 인스턴스의 시작 및 종료를 자동화하려는 경우 자원 그룹 및 자원에 대한 필수 종속성이 설정되게 합니다.

자세한 내용은 Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS를 참조하십시오.

---

주 - Oracle Real Application Clusters 데이터베이스 인스턴스의 시작 및 종료를 자동화할 경우 Sun Cluster 소프트웨어 버전 3.1 9/04 또는 호환 버전을 사용해야 합니다.

---

주 - Oracle RAC 10g 구성에 사용되는 공유 Sun StorageTek QFS 구성에서 Cluster Ready Services(CRS)용 Oracle 설치 프로그램이 사용자에게 root.sh를 실행하도록 요청할 경우 이 명령은 경우에 따라 실패합니다. 또는 Oracle Cluster Registry(OCR) 파일이 root.sh에 의해 생성되는 경우에는 CRS 레지스트리가 불안정하게 됩니다.

이 문제를 해결하려면 700416보다 크게 되도록 OCR 파일을 사전 할당합니다. 예를 들어, 여기에 설명된 대로 root.sh를 실행하기 전에 사용자 oracle로서 1메가바이트 크기의 파일을 사전에 할당합니다.

```
$ dd if=/dev/zero of=<OCR file path> bs=1024k count=1M
```

---

## Sun Cluster 환경에서 공유 해제된 파일 시스템 구성

Sun Cluster 시스템에 공유 해제된 Sun StorageTek QFS 파일 시스템을 설치할 때 Sun Cluster HAStoragePlus 자원 유형에 고가용성(HA)을 위한 파일 시스템을 구성합니다. Sun Cluster 시스템에서 공유 해제된 Sun StorageTek QFS 파일 시스템은 일반적으로 고가용성 네트워크 파일 서버(HA-NFS) 또는 고가용성 ORACLE(HA-ORACLE)과 같은 하나 이상의 장애 조치 응용 프로그램과 관련이 있습니다. 공유되지 않은 Sun StorageTek QFS 파일 시스템과 장애 복구 응용 프로그램이 둘 다 하나의 자원 그룹에서 활성화됩니다. 즉, 자원 그룹은 한 번에 하나의 Sun Cluster 노드에서 활성화됩니다.

공유되지 않은 Sun StorageTek QFS 파일 시스템은 임의의 주어진 시간에 단일 노드에 마운트됩니다. Sun Cluster 결함 모니터가 오류를 발견하는 경우나 사용자가 자원 그룹을 전환하는 경우, 자원 그룹이 이전에 구성된 방법에 따라서 공유되지 않은 Sun StorageTek QFS 파일 시스템 및 그의 연관된 HA 응용 프로그램이 다른 노드로 장애 복구됩니다.

Sun Cluster 전역 장치 그룹에 들어있는 모든 파일 시스템(/dev/global/\*)은 HAStoragePlus 자원 유형과 함께 사용할 수 있습니다. 파일 시스템이 HAStoragePlus 자원 유형으로 구성될 때 Sun Cluster 자원 그룹의 일부가 되고 Sun Cluster RGM(Resource Group Manager) 제어 하의 파일 시스템이 자원 그룹이 활성화된 노드에 로컬로 마운트됩니다. RGM이 자원 그룹 전환을 유발하거나 다른 구성된 Sun Cluster 노드로 장애 복구할 때 공유되지 않은 Sun StorageTek QFS 파일 시스템이 현재 노드에서 마운트 해제되고 새 노드에 다시 마운트됩니다.

공유되지 않은 각 Sun StorageTek QFS 파일 시스템은 최소한 2개의 원시 디스크 분할 영역이나 볼륨 관리자 제어 볼륨(Solstice DiskSuite™/Solaris Volume Manager 또는 VERITAS Volume Manager)이 필요하며, 하나는 Sun StorageTek QFS 메타 데이터(inode)용이고 하나는 Sun StorageTek QFS 파일 데이터용입니다. 복수 데이터 경로를 통해 복수 디스크 사이에 복수 분할 영역이나 볼륨을 구성하면 공유되지 않은 Sun StorageTek QFS 파일 시스템 성능이 증가됩니다. 메타 데이터 및 파일 데이터 분할 영역 크기 조정에 대한 내용은 7페이지의 "디자인의 기본"을 참조하십시오.

이 절에서는 공유 해제된 Sun StorageTek QFS 파일 시스템을 사용하여 Sun Cluster 시스템 구성의 세 가지 예제를 제공합니다. 이러한 예제에서 파일 시스템이 다음의 HA-NFS 파일 마운트 지점과 결합하여 구성됩니다.

- 예제 1의 원시 전역 장치
- 예제 2의 Solstice DiskSuite/Solaris Volume Manager 제어 볼륨
- 예제 3의 VERITAS Volume Manager(VxVM) 제어 볼륨

이러한 모든 구성에서 단순화할 수 있도록 각 파일 시스템의 10%가 Sun StorageTek QFS 메타 데이터에 사용되고 나머지 공간은 Sun StorageTek QFS 파일 데이터에 사용됩니다. 크기 및 디스크 배치 고려사항에 대한 내용은 Sun StorageTek QFS 설치 및 업그레이드 안내서를 참조하십시오.

## 예제 1: 원시 전역 장치의 HA-NFS

이 예제에서는 원시 전역 장치에 HA-NFS와 함께 공유 해제된 Sun StorageTek QFS 파일 시스템을 구성하는 방법을 나타냅니다. 이 구성의 경우 원시 전역 장치가 제어기 기반 기억장치에 포함되어야 합니다. 이 제어기 기반 저장소는 RAID-1 또는 RAID-5를 통해 장치 중복성을 지원해야 합니다.

코드 예 6-1에 표시된 것처럼, 이 예제에서 사용된 DID 장치 d4 - d7이 높은 가용성을 가지며 제어기 기반 저장소에 포함되어 있습니다. HAStoragePlus 자원 유형은 전역 장치의 사용을 필요로하므로, 각 DID 장치(/dev/did/dsk/dx)는 다음 구문을 사용하여 전역 장치로서 액세스할 수 있습니다. /dev/global/dsk/dx.

이 예제의 기본 절차는 다음과 같습니다.

1. 공유되지 않는 파일 시스템 작성을 준비합니다.
2. 파일 시스템을 작성하고 Sun Cluster 노드를 구성합니다.
3. 네트워크 이름 서비스 및 IPMP(IP Measurement Protocol) 검증 테스트를 구성합니다.
4. HA-NFS를 구성하고 고가용성을 위해 파일 시스템을 구성합니다.

## ▼ 공유되지 않는 Sun StorageTek QFS 파일 시스템 작성 준비

1. format(1M) 유틸리티를 사용하여 /dev/global/dsk/d4에 분할 영역을 배치합니다.

```
# format /dev/global/rdsk/d4s2
# format> partition
[ output deleted ]
# 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
```

NOTE: Partition 2 (backup) will not be used and was created by format(1m) by default.

분할 영역(또는 슬라이스) 0은 볼륨의 볼륨 목차(VTOC)를 건너뛰며 20GB 분할 영역으로 구성됩니다. 나머지 공간은 분할 영역 1로 구성됩니다.

2. 전역 장치 d4 분할 영역을 전역 장치 d5 - d7에 복제합니다.

이 예제는 전역 장치 d5에 대한 명령을 나타냅니다.

```
# prtvtoc /dev/global/rdsk/d4s2 | fmthard \
-s - /dev/global/rdsk/d5s2
```

3. 파일 시스템의 가능한 호스트인 모든 노드에서 다음을 수행합니다.

- a. mcf 파일에 새로운 파일 시스템 항목을 추가하여 **Sun StorageTek QFS** 파일 시스템에 8개 분할 영역(각각 2개의 분할 영역을 갖는 4개의 전역 장치)을 구성합니다.

```
# cat >> /etc/opt/SUNWsamfs/mcf <<EOF
#
# Sun StorageTek QFS file system configurations
#
# Equipment      Equipment      Equipment      Family      Device      Additional
# Identifier      Ordinal        Type           Set         State       Parameters
# -----
qfsnfs1          100            ma             qfsnfs1     on
/dev/global/dsk/d4s0 101            mm             qfsnfs1
/dev/global/dsk/d5s0 102            mm             qfsnfs1
/dev/global/dsk/d6s0 103            mm             qfsnfs1
/dev/global/dsk/d7s0 104            mm             qfsnfs1
/dev/global/dsk/d4s1 105            mr             qfsnfs1
/dev/global/dsk/d5s1 106            mr             qfsnfs1
/dev/global/dsk/d6s1 107            mr             qfsnfs1
/dev/global/dsk/d7s1 108            mr             qfsnfs1
EOF
```

mcf 파일에 대한 내용은 34페이지의 "mcf 파일의 기능"을 참조하십시오.

- b. mcf 파일에 추가한 구성 정보가 올바른지 검증하고 작업을 진행하기 전에 mcf 파일의 오류를 수정합니다.

HASStoragePlus 자원 유형에 Sun StorageTek QFS 파일 시스템을 구성하기 전에 이 단계를 완료하는 것이 중요합니다.

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

### ▼ Sun StorageTek QFS 파일 시스템 작성 및 Sun Cluster 노드 구성

1. 파일 시스템의 가능한 호스트인 각 노드에서 **samd(1M)** config 명령을 실행합니다. 이 명령은 Sun StorageTek QFS 때문에 새로운 Sun StorageTek QFS 구성이 사용 가능함을 신호합니다.

```
# samd config
```

2. Sun Cluster 시스템의 한 노드에서 `sammkfs(1M)` 명령을 사용하여 파일 시스템을 작성합니다.

```
# sammkfs qfsnfs1 < /dev/null
```

3. 파일 시스템의 가능한 호스트인 각 노드에서 다음을 수행합니다.

- a. `mkdir(1M)` 명령을 사용하여 파일 시스템에 대한 전역 마운트 지점을 작성하고, `chmod(1M)` 명령을 사용하여 `root`를 마운트 지점의 소유자로 만들고, `chown(1M)` 명령을 사용하여 읽기/쓰기(755) 액세스 권한을 갖는 다른 사람이 해당 마운트 지점을 사용할 수 있게 합니다.

```
# mkdir /global/qfsnfs1
# chmod 755 /global/qfsnfs1
# chown root:other /global/qfsnfs1
```

- b. `/etc/vfstab` 파일에 Sun StorageTek QFS 파일 시스템 항목을 추가합니다. 마운트 옵션 필드에 `sync_meta=1` 값이 있습니다.

```
# 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. 파일 시스템을 마운트 및 마운트 해제하여 구성을 검증합니다.

```
# mount qfsnfs1
# ls /global/qfsnfs1
lost+found/
# umount qfsnfs1
```

4. `scrgadm(1M) -p | egrep` 명령을 사용하여 필수 Sun Cluster 자원 유형이 자원 구성에 추가되었는지 검증합니다.

```
# scrgadm -p | egrep "SUNW.HAStoragePlus|SUNW.LogicalHostname|SUNW.nfs"
```

- 필수 **Sun Cluster** 자원 유형을 찾을 수 없는 경우 `scrgadm(1M) -a -t` 명령을 사용하여 구성에 해당 자원 유형을 추가합니다.

```
# scrgadm -a -t SUNW.HAStoragePlus
# scrgadm -a -t SUNW.LogicalHostname
# scrgadm -a -t SUNW.nfs
```

## ▼ 네트워크 이름 서비스 및 IPMP 검증 테스트 구성

이 절에서는 Sun Cluster 노드를 위한 네트워크 이름 서비스 및 IPMP 검증 테스트를 구성하는 방법의 예제를 제공합니다. 자세한 내용은 Sun Cluster Software Installation Guide for Solaris OS, System Administration Guide: IP Services 및 System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)를 참조하십시오.

- `vi` 또는 다른 텍스트 편집기를 사용하거나 `/etc/nsswitch.conf` 파일을 사용하여 노드 이름에 대해 **Sun Cluster** 시스템 및 파일을 조사합니다.

Network Information Name service(NIS) 서버를 구성하기 전에 이 단계를 수행합니다.

```
# 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
[remainder of file content not shown]
```

2. /etc/nsswitch.conf 파일에 수행한 변경이 올바른지 확인합니다.

```
# grep '^hosts:' /etc/nsswitch.conf
hosts:    cluster files nis [NOTFOUND=return]
#
```

3. 사용 가능한 네트워크 어댑터를 사용하여 **IPMP** 검증 테스트를 설정합니다.
  - 어댑터 qfe2 및 qfe3이 예제로 사용됩니다.
  - a. 각 어댑터에 대해 정적으로 **IPMP** 테스트 주소를 구성합니다.

```
#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. 동적으로 **IPMP** 어댑터를 구성합니다.

```
# 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. 구성을 확인합니다.

```
# 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
```

## ▼ 고가용성을 위한 HA-NFS 및 Sun StorageTek QFS 파일 시스템 구성

이 절에서는 HA-NFS 구성 방법의 예제를 제공합니다. HA-NFS에 대한 자세한 내용은 Sun Cluster Data Service for Network File System (NFS) Guide for Solaris OS 및 NFS 문서를 참조하십시오.

1. Sun StorageTek QFS 파일 시스템을 위한 NFS 공유 지점을 작성합니다.

공유 지점은 Sun StorageTek QFS 파일 시스템이 아니라 /global 파일 시스템에 들어 있다는 점을 기억하십시오.

```
# mkdir -p /global/nfs/SUNW.nfs
# echo "share -F nfs -o rw /global/qfsnfs1" > \
/global/nfs/SUNW.nfs/dfstab.nfs1-res
```

2. NFS 자원 그룹을 작성합니다.

```
# scrgadm -a -g nfs-rg -y PathPrefix=/global/nfs
```

3. 사용자 사이트의 주소를 사용하여 /etc/hosts 표에 NFS 논리 호스트를 추가합니다.

```
# cat >> /etc/hosts << EOF
#
# IP Addresses for LogicalHostnames
#
192.168.2.10      lh-qfs1
EOF
```

4. scrgadm(1M) -a -L -g 명령을 사용하여 NFS 자원 그룹에 논리 호스트를 추가합니다.

```
# scrgadm -a -L -g nfs-rg -l lh-nfs1
```

5. scrgadm(1M) -c -g 명령을 사용하여 HAStoragePlus 자원 유형을 구성합니다.

```
# 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. 자원 그룹을 온라인으로 만듭니다.

```
# scswitch -Z -g nfs-rg
```

7. NFS 자원 유형을 구성하고 HAStoragePlus 자원에 대한 종속성을 설정합니다.

```
# scrgadm -a -g nfs-rg -j nfs1-res -t SUNW.nfs -y \  
Resource_dependencies=qfsnfs1-res
```

8. NFS 자원을 온라인으로 만듭니다.

```
# scswitch -e -j nfs1-res
```

NFS 자원 /net/lh-nfs1/global/qfsnfs1이 이제 완전히 구성되었으며 또한 높은 가용성을 갖습니다.

9. Sun StorageTek QFS 파일 시스템에서 고가용성 NFS 파일 시스템의 가용성을 발표하기 전에 자원 그룹이 오류 없이 모든 구성된 노드 사이에 전환될 수 있고 온라인 및 오프라인이 될 수 있는지 확인하도록 테스트합니다.

```
# 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
```

## 예제 2: Solstice DiskSuite/Solaris Volume Manager 가 제어하는 볼륨의 HA-NFS

이 예제는 Solstice DiskSuite/Solaris Volume Manager 소프트웨어가 제어하는 볼륨에서 HA-NFS를 갖는 공유 해제된 Sun StorageTek QFS 파일 시스템을 구성하는 방법을 나타냅니다. 이 구성에서 DID 장치가 RAID-1 또는 RAID-5 볼륨을 사용하여 중복 제어기 기반 기억장치에 포함되는지 여부를 선택할 수 있습니다. 일반적으로 Solaris Volume Manager는 기초 제어기 기반 기억장치가 중복되지 않을 때만 사용됩니다.

**코드 예 6-1**에 표시된 것처럼, 이 예제에서 사용된 DID 장치 d4 - d7이 높은 가용성을 가지며 제어기 기반 저장소에 포함되어 있습니다. Solaris Volume Manager는 DID 장치가 Solaris Volume Manager가 볼륨을 구성할 수 있는 원시 장치를 채우는 데 사용될 것을 필요로 합니다. Solaris Volume Manager는 전역으로 액세스 가능한 디스크 그룹을 작성하며, HAStoragePlus 자원 그룹이 이 그룹을 Sun StorageTek QFS 파일 시스템 작성에 사용할 수 있습니다.

이 예제는 다음 단계를 따릅니다.

1. Solstice DiskSuite/Solaris Volume Manager 소프트웨어를 준비합니다.
2. 공유되지 않는 파일 시스템 작성을 준비합니다.
3. 파일 시스템을 작성하고 Sun Cluster 노드를 구성합니다.
4. 네트워크 이름 서비스 및 IPMP 검증 테스트를 구성합니다.
5. HA-NFS를 구성하고 고가용성을 위해 파일 시스템을 구성합니다.

## ▼ Solstice DiskSuite/Solaris Volume Manager 소프트웨어 준비

1. **Solaris Volume Manager** 메타 데이터베이스(`metadb`)가 **Sun StorageTek QFS** 파일 시스템의 가능한 호스트인 각 노드에 구성되어 있는지 여부를 판별합니다.

```
# 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)` 명령이 메타 데이터베이스 구성을 반환하지 않는 경우, 각 노드에서 하나 이상의 로컬 디스크에 셋 이상의 데이터베이스 복제를 작성합니다. 각 복제의 크기는 최소한 16MB여야 합니다. 메타 데이터베이스 구성 작성에 대한 자세한 내용을 보려면 Sun Cluster Software Installation Guide for Solaris OS를 참조하십시오.

2. 이 **Sun StorageTek QFS** 파일 시스템을 위한 모든 **Solaris Volume Manager** 볼륨을 포함할 **HA-NFS** 디스크 그룹을 작성합니다.

```
# metaset -s nfsdg -a -h scnode-A scnode-B
```

3. **Solaris Volume Manager**가 볼륨을 작성할 수 있는 원시 장치 풀에 **DID** 장치 `d4 - d7`을 추가합니다.

```
# metaset -s nfsdg -a /dev/did/dsk/d4 /dev/did/dsk/d5 \  
/dev/did/dsk/d6 /dev/did/dsk/d7
```

## ▼ Sun StorageTek QFS 파일 시스템 준비

1. `format(1M)` 유틸리티를 사용하여 `/dev/global/dsk/d4`에 분할 영역을 배치합니다.

```
# format /dev/global/rdsk/d4s2
# format> partition
[ output deleted ]
# 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

NOTE: Partition 2 (backup) will not be used and was created by format(1m) by
default.
```

이 예제는 분할 영역 또는 슬라이스 0이 볼륨의 볼륨 목차(VTOC)를 건너뛰며 20GB 분할 영역으로 구성됨을 보여줍니다. 나머지 공간은 분할 영역 1로 구성됩니다.

2. DID 장치 d4의 분할 영역을 DID 장치 d5 - d7에 복제합니다.

이 예제는 장치 d5에 대한 명령을 표시합니다.

```
# prtvtoc /dev/global/rdsk/d4s2 | fmthard \
-s - /dev/global/rdsk/d5s2
```

3. 8개 분할 영역(4개의 DID 장치, 각각 2개의 분할 영역)을 2개의 RAID-1(미러) Sun StorageTek QFS 메타 데이터 볼륨과 2개의 RAID-5(패리티 스트라이프) Sun StorageTek QFS 파일 데이터 볼륨으로 구성합니다.

- a. 이러한 4개 드라이브의 분할 영역(슬라이스) 0을 2개의 RAID-1 세트로 결합합니다.

```
# metainit -s nfsdg -f d1 1 1 /dev/did/dsk/d4s0
# metainit -s nfsdg -f d2 1 1 /dev/did/dsk/d5s0
# metainit -s nfsdg d10 -m d1 d2
# metainit -s nfsdg -f d3 1 1 /dev/did/dsk/d6s0
# metainit -s nfsdg -f d4 1 1 /dev/did/dsk/d7s0
# metainit -s nfsdg d11 -m d3 d4
```

b. 이러한 4개 드라이브의 분할 영역 1을 2개의 RAID-5 세트로 결합합니다.

```
# metainit -s nfsdg d20 -p /dev/did/dsk/d4s1 205848574b
# metainit -s nfsdg d21 -p /dev/did/dsk/d5s1 205848574b
# metainit -s nfsdg d22 -p /dev/did/dsk/d6s1 205848574b
# metainit -s nfsdg d23 -p /dev/did/dsk/d7s1 205848574b
# metainit -s nfsdg d30 -r d20 d21 d22 d23
```

c. 파일 시스템의 가능한 호스트인 각 노드에서 Sun StorageTek QFS 파일 시스템 항목을 mcf 파일에 추가합니다.

```
# cat >> /etc/opt/SUNWsamfs/mcf <<EOF

# Sun StorageTek QFS file system configurations
#
# Equipment      Equipment      Equipment      Family      Device      Additional
# Identifier      Ordinal        Type           Set         State       Parameters
# -----
qfsnfs1          100            ma             qfsnfs1    on
/dev/md/nfsdg/dsk/d10  101            mm             qfsnfs1
/dev/md/nfsdg/dsk/d11  102            mm             qfsnfs1
/dev/md/nfsdg/dsk/d30  103            mr             qfsnfs1
EOF
```

mcf 파일에 대한 자세한 내용은 34페이지의 "mcf 파일의 기능"을 참조하십시오.

4. mcf(4) 구성이 각 노드에서 올바른지 검증하고 작업을 진행하기 전에 mcf 파일의 모든 오류를 수정합니다.

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

## ▼ Sun StorageTek QFS 파일 시스템 작성 및 Sun Cluster 노드 구성

1. 파일 시스템의 가능한 호스트인 각 노드에서 **samd(1M)** config 명령을 사용합니다. 이 명령은 Sun StorageTek QFS 때문에 새로운 Sun StorageTek QFS 구성이 사용 가능함을 신호합니다.

```
# samd config
```

2. 디스크 그룹의 **Solaris Volume Manager** 중재 감지를 활성화하는데, 이는 드라이브 오류를 감지할 때 **Sun Cluster** 시스템에게 도움을 줍니다.

```
# metaset -s nfsdg -a -m scnode-A
# metaset -s nfsdg -a -m scnode-B
```

3. 파일 시스템의 가능한 호스트인 각 노드에서 **NFS** 디스크 그룹이 존재하는지 확인합니다.

```
# metaset -s nfsdg -t
```

4. **Sun Cluster** 시스템의 한 노드에서 **sammkfs(1M)** 명령을 사용하여 파일 시스템을 작성합니다.

```
# sammkfs qfsnfs1 < /dev/null
```

5. 파일 시스템의 가능한 호스트인 각 노드에서 다음을 수행합니다.

- a. **mkdir(1M)** 명령을 사용하여 파일 시스템에 대한 전역 마운트 지점을 작성하고, **chmod(1M)** 명령을 사용하여 **root**를 마운트 지점의 소유자로 만들고, **chown(1M)** 명령을 사용하여 읽기/쓰기(755) 액세스 권한을 갖는 다른 사람이 해당 마운트 지점을 사용할 수 있게 합니다.

```
# mkdir /global/qfsnfs1
# chmod 755 /global/qfsnfs1
# chown root:other /global/qfsnfs1
```

- b. **/etc/vfstab** 파일에 **Sun StorageTek QFS** 파일 시스템 항목을 추가합니다. 마운트 옵션 필드에 **sync\_meta=1** 값이 있습니다.

```
# 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. 파일 시스템을 마운트 및 마운트 해제하여 구성을 검증합니다.

한 번에 한 노드에서 이 단계를 수행하십시오. 이 예제에서는 qfsnfs1 파일 시스템이 한 노드에서 마운트 및 마운트 해제됩니다.

```
# mount qfsnfs1
# ls /global/qfsnfs1
lost+found/
# umount qfsnfs1
```

---

주 - 마운트 지점을 테스트할 때 `metaset -r (release)` 및 `-t (take)` 명령을 사용하여 Sun Cluster 노드 사이에 `nfsdg` 디스크 그룹을 이동하십시오. 그런 다음 `samd(1M) config` 명령을 사용하여 데몬에 구성 변경을 경고하십시오.

---

6. `scrgadm(1M) -p | egrep` 명령을 사용하여 필수 Sun Cluster 자원 유형이 자원 구성에 추가되었는지 검증합니다.

```
# scrgadm -p | egrep "SUNW.HAStoragePlus|SUNW.LogicalHostname|SUNW.nfs"
```

7. 필수 Sun Cluster 자원 유형을 찾을 수 없는 경우 다음 명령의 하나 이상으로 해당 유형을 추가하십시오.

```
# scrgadm -a -t SUNW.HAStoragePlus
# scrgadm -a -t SUNW.LogicalHostname
# scrgadm -a -t SUNW.nfs
```

## ▼ 네트워크 이름 서비스 및 IPMP 검증 테스트 구성

네트워크 이름 서비스 및 IPMP 유효성 테스트를 구성하려면 다음의 지침을 따릅니다. [145페이지의 "네트워크 이름 서비스 및 IPMP 검증 테스트 구성"](#)

## ▼고가용성을 위한 HA-NFS 및 Sun StorageTek QFS 파일 시스템 구성

이 절에서는 HA-NFS 구성 방법의 예제를 제공합니다. HA-NFS에 대한 자세한 내용은 Sun Cluster Data Service for Network File System (NFS) Guide for Solaris OS 및 NFS 문서를 참조하십시오.

1. Sun StorageTek QFS 파일 시스템을 위한 NFS 공유 지점을 작성합니다.

공유 지점은 Sun StorageTek QFS 파일 시스템이 아니라 /global 파일 시스템에 포함 되어 있습니다.

```
# mkdir -p /global/nfs/SUNW.nfs
# echo "share -F nfs -o rw /global/qfsnfs1" > \
/global/nfs/SUNW.nfs/dfstab.nfs1-res
```

2. NFS 자원 그룹을 작성합니다.

```
# scrgadm -a -g nfs-rg -y PathPrefix=/global/nfs
```

3. 로컬 호스트를 NFS 자원 그룹에 추가합니다.

```
# scrgadm -a -L -g nfs-rg -l lh-nfs1
```

4. HAStoragePlus 자원 유형을 구성합니다.

```
# 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. 자원 그룹을 온라인으로 만듭니다.

```
# scswitch -Z -g nfs-rg
```

6. NFS 자원 유형을 구성하고 HAStoragePlus 자원에 대한 종속성을 설정합니다.

```
# scrgadm -a -g nfs-rg -j nfs1-res -t SUNW.nfs -y \
Resource_dependencies=qfsnfs1-res
```

7. scswitch(1M) -e -j 명령을 사용하여 NFS 자원을 온라인으로 만듭니다.

```
# scswitch -e -j nfs1-res
```

NFS 자원 /net/lh-nfs1/global/qfsnfs1이 완전히 구성되었으며 높은 가용성을 갖습니다.

8. **Sun StorageTek QFS** 파일 시스템에서 고가용성 **NFS** 파일 시스템의 가용성을 발표하기 전에 자원 그룹이 오류 없이 모든 구성된 노드 사이에 전환될 수 있고 온라인 및 오프라인이 될 수 있는지 확인하도록 테스트합니다.

```
# 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
```

### 예제 3: VxVM 볼륨의 HA-NFS

이 예제는 VERITAS Volume Manager 제어 볼륨(VxVM 볼륨)에서 HA-NFS를 갖는 공유 해제된 Sun StorageTek QFS 파일 시스템을 구성하는 방법을 보여줍니다. 이 구성으로 DID 장치가 RAID-1 또는 RAID-5를 사용하는 중복 제어기 기반 기억장치에 포함되는지 여부를 선택할 수 있습니다. 일반적으로 VxVM은 기초 기억장치가 중복되지 않을 때만 사용됩니다.

**코드 예 6-1**에 표시된 것처럼, 이 예제에서 사용된 DID 장치 d4 - d7이 높은 가용성을 가지며 제어기 기반 저장소에 포함되어 있습니다. VxVM은 공유 DID 장치가 VxVM이 볼륨을 구성하는 원시 장치를 채우는 데 사용되도록 요구합니다. VxVM은 디스크 그룹을 Sun Cluster 장치 그룹으로 등록하여 고가용성 디스크 그룹을 작성합니다. 이들 디스크 그룹은 전역으로 액세스할 수 없지만, 최소한 한 노드에 액세스 가능하게 하여 장애 복구될 수 있습니다. 디스크 그룹은 HAStoragePlus 자원 유형에서 사용될 수 있습니다.

---

주 - VxVM 패키지는 설치, 패치 및 라이선스되어야 하는 별도의 추가 패키지입니다. VxVM 설치에 대해서는 VxVM Volume Manager 문서를 참조하십시오.

---

VxVM과 함께 Sun StorageTek QFS 소프트웨어를 사용하려면 다음 VxVM 패키지를 설치해야 합니다.

- VRTSvlic
- VRTSvmdoc
- VRTSvmmman
- VRTSvmpro
- VRTSvxvm
- VRTSob 및 VRTSobgui(선택 사항 브라우저 인터페이스 패키지)

이 예제는 다음 단계를 따릅니다.

1. VxVM 소프트웨어를 구성합니다.
2. 공유되지 않는 파일 시스템 작성을 준비합니다.

3. 파일 시스템을 작성하고 Sun Cluster 노드를 구성합니다.
4. 구성을 검증합니다.
5. 네트워크 이름 서비스 및 IPMP 검증 테스트를 구성합니다.
6. HA-NFS를 구성하고 고가용성을 위해 파일 시스템을 구성합니다.

## ▼ VxVM 소프트웨어 구성

이 절에서는 Sun StorageTek QFS 소프트웨어와 함께 사용하도록 VxVM 소프트웨어를 구성하는 방법의 예제를 제공합니다. VxVM 소프트웨어에 대한 자세한 내용은 VxVM 문서를 참조하십시오.

1. VERITAS에 대한 DMP(동적 다중 경로 지정)의 상태를 판별합니다.

```
# vxddmpadm listctlr all
```

2. `scdidadm(1M)` 유틸리티를 사용하여 VxVM이 사용할 실제 장치의 HBA 제어기 번호를 판별합니다.

다음 예제에 표시된 것처럼 다중 노드 액세스 가능 저장소는 HBA 제어기 c6을 사용하여 `scnode-A`로부터 사용 가능하고 제어기 c7을 사용하여 노드 `scnode-B`로부터 사용 가능합니다.

```
# scdidadm -L
[ some output deleted]
4   scnode-A:/dev/dsk/c6t60020F20000037D13E26595500062F06d0 /dev/did/dsk/d4
4   scnode-B:/dev/dsk/c7t60020F20000037D13E26595500062F06d0 /dev/did/dsk/d4
```

3. VxVM을 사용하여 c6 제어기를 통해 보이는 대로 모든 사용 가능한 저장소를 구성합니다.

```
# vxddmpadm getsubpaths ctlr=c6
```

4. 이 제어기의 모든 장치를 VxVM 제어 하에 둡니다.

```
# vxddiskadd fabric_
```

5. 디스크 그룹을 작성하고, 볼륨을 작성한 후 새 디스크 그룹을 시작합니다.

```
# /usr/sbin/vxdg init qfs-dg qfs-dg00=disk0 \
qfsdg01=disk1 qfsdg02=disk2 qfsdg03=disk3
```

6. 이전에 시작한 디스크 그룹이 이 시스템에서 활성화인지 확인하십시오.

```
# vxvg import nfsdg
# vxvg free
```

7. Sun StorageTek QFS 메타 데이터용 2개의 미리 볼륨과 Sun StorageTek QFS 파일 데이터 볼륨용 2개의 볼륨을 구성합니다.

이러한 미러링 작업은 백그라운드 프로세스로서 수행되어 완료하기 위한 충분한 시간을 제공합니다.

```
# vxassist -g nfsdg make m1 10607001b
# vxassist -g nfsdg mirror m1&
# vxassist -g nfsdg make m2 10607001b
# vxassist -g nfsdg mirror m2&
# vxassist -g nfsdg make m10 201529000b
# vxassist -g nfsdg mirror m10&
# vxassist -g nfsdg make m11 201529000b
# vxassist -g nfsdg mirror m11&
```

8. 이전에 작성된 VxVM 디스크 그룹을 Sun Cluster 제어 디스크 그룹으로 구성합니다.

```
# scconf -a -D type=vxvm,name=nfsdg,nodelist=scnode-A:scnode-B
```

## ▼ Sun StorageTek QFS 파일 시스템 작성 준비

파일 시스템의 가능한 호스트인 각 노드에서 이 절차를 수행하십시오.

1. Sun StorageTek QFS 파일 시스템 항목을 mcf 파일에 추가합니다.

코드 예 6-8            mcf 파일에 파일 시스템 추가

```
# cat >> /etc/opt/SUNWsamfs/mcf <<EOF
# Sun StorageTek QFS file system configurations
#
# Equipment                    Equipment    Equipment    Family        Device        Additional
# Identifier                    Ordinal      Type          Set           State          Parameters
# -----                        -----      -----        -----        -----
qfsnfs1                        100          ma            qfsnfs1       on
/dev/vx/dsk/nfsdg/m1           101          mm            qfsnfs1
/dev/vx/dsk/nfsdg/m2           102          mm            qfsnfs1
/dev/vx/dsk/nfsdg/m10          103          mr            qfsnfs1
/dev/vx/dsk/nfsdg/m11          104          mr            qfsnds1
EOF
```

mcf 파일에 대한 자세한 내용은 34페이지의 "mcf 파일의 기능"을 참조하십시오.

2. mcf(4) 구성이 올바른지 검증하고 작업을 진행하기 전에 mcf 파일의 모든 오류를 수정합니다.

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

## ▼ Sun StorageTek QFS 파일 시스템 작성 및 Sun Cluster 노드 구성

1. 파일 시스템의 가능한 호스트인 각 노드에서 **samd(1M)** config 명령을 사용합니다. 이 명령은 Sun StorageTek QFS 데몬에 새로운 Sun StorageTek QFS 구성이 사용 가능함을 신호합니다.

```
# samd config
```

2. Sun Cluster 시스템의 한 노드에서 **sammkfs(1M)** 명령을 사용하여 파일 시스템을 작성합니다.

```
# sammkfs qfsnfs1 < /dev/null
```

3. 파일 시스템의 가능한 호스트인 각 노드에서 다음을 수행합니다.
  - a. **mkdir(1M)** 명령을 사용하여 파일 시스템에 대한 전역 마운트 지점을 작성하고, **chmod(1M)** 명령을 사용하여 root를 마운트 지점의 소유자로 만들고, **chown(1M)** 명령을 사용하여 읽기/쓰기(755) 액세스 권한을 갖는 다른 사람이 해당 마운트 지점을 사용할 수 있게 합니다.

```
# mkdir /global/qfsnfs1
# chmod 755 /global/qfsnfs1
# chown root:other /global/qfsnfs1
```

- b. `/etc/vfstab` 파일에 Sun StorageTek QFS 파일 시스템 항목을 추가합니다. 마운트 옵션 필드에 `sync_meta=1` 값이 들어있습니다.

```
# 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
```

## ▼ 구성 검증

1. 파일 시스템의 가능한 호스트인 모든 노드가 올바르게 구성되었는지 확인합니다.

이렇게 하려면 [157페이지의 "VxVM 소프트웨어 구성"](#)에서 작성한 디스크 그룹을 노드로 이동하고, 파일 시스템을 마운트한 후 마운트 해제하십시오. 한 번에 한 노드에서 이 검증을 수행하십시오.

```
# scswitch -z -D nfsdg -h scnode-B
# mount qfsnfs1
# ls /global/qfsnfs1
lost+found/
# umount qfsnfs1
```

2. 필수 Sun Cluster 자원 유형이 자원 구성에 추가되었는지 확인합니다.

```
# scrgadm -p | egrep "SUNW.HAStoragePlus|SUNW.LogicalHostname|SUNW.nfs"
```

3. 필수 Sun Cluster 자원 유형을 찾을 수 없는 경우 다음 명령의 하나 이상으로 해당 유형을 추가하십시오.

```
# scrgadm -a -t SUNW.HAStoragePlus
# scrgadm -a -t SUNW.LogicalHostname
# scrgadm -a -t SUNW.nfs
```

## ▼ 네트워크 이름 서비스 및 IPMP 검증 테스트 구성

네트워크 이름 서비스 및 IPMP 유효성 테스트를 구성하려면 다음의 지침을 따릅니다. [145페이지의 "네트워크 이름 서비스 및 IPMP 검증 테스트 구성"](#)

## ▼ 고가용성을 위한 HA-NFS 및 Sun StorageTek QFS 파일 시스템 구성

HA-NFS 및 파일 시스템을 고가용성으로 구성하려면 [154페이지의 "고가용성을 위한 HA-NFS 및 Sun StorageTek QFS 파일 시스템 구성"](#)의 지침을 따릅니다.

---

# 클러스터 외부에 공유 클라이언트 구성

Sun Cluster 환경을 구성하고 공유 클라이언트가 클러스터 외부에 있도록 하려면 다음 구성을 수행합니다.

아래 예제는 두 개의 노드를 가진 메타 데이터 서버 클러스터 구성을 기초로 하고 있습니다.

## 구성 전제조건

다음 항목은 클러스터 외부에 공유 클라이언트를 설치하기 위해 구성되거나 검증되어야 합니다.

- 161페이지의 "Sun StorageTek QFS 메타 데이터 서버 Sun Cluster 노드"
- 162페이지의 "Sun StorageTek QFS 메타 데이터 클라이언트 노드"
- 162페이지의 "Sun Cluster 장치 구성"
- 162페이지의 "클러스터 외부의 클라이언트 구성을 위한 요구 사항"
- 163페이지의 "최소 소프트웨어 릴리스 레벨"
- 163페이지의 "지원되는 하드웨어 아키텍처"
- 163페이지의 "저장소 요구 사항"

## Sun StorageTek QFS 메타 데이터 서버 Sun Cluster 노드

Sun StorageTek QFS 메타 데이터 서버 Sun Cluster 노드의 경우 다음 요구 사항이 충족되어야 합니다.

- mm 장치로 식별된 Sun StorageTek QFS 메타 데이터 장치는 데이터 장치와 동일한 물리적 LUN에 존재할 수 없습니다.
- Sun Cluster `scconf localonly` 플래그는 공유 Sun StorageTek QFS 구성에서 mm 장치로 정의된 메타 데이터 장치를 위해 활성화될 수 없습니다.
- mr 장치로 식별된 Sun StorageTek QFS 데이터 장치는 메타 데이터 장치와 동일한 물리적 LUN에 존재할 수 없습니다.

## Sun StorageTek QFS 메타 데이터 클라이언트 노드

Sun StorageTek QFS 메타 데이터 클라이언트 노드는 다음 요구 사항을 만족해야 합니다.

- mm 장치로 식별된 Sun StorageTek QFS 메타 데이터 장치는 공유 Sun StorageTek QFS 구성에서 nodev 장치로 식별되어야 합니다.
- Sun StorageTek QFS 클라이언트에서 mr로 식별된 데이터 장치는 클러스터된 노드에 사용된 /dev/did/dsk 구문 대신 /dev/dsk 구문을 따라야 합니다.

## Sun Cluster 장치 구성

localonly 플래그는 모든 데이터 장치에 설정되어야 합니다.

/etc/opt/SUNWsamfs/mcf 파일을 통해 데이터 장치에 대한 로컬 모드를 설정하여 Sun StorageTek QFS 데이터 장치로 사용되는 장치를 확인합니다.

Sun Cluster를 실행하는 임의의 노드에서 root로서 다음을 수행합니다.

```
scconf -r -D name=dsk/dX,nodelist=node2
```

```
scconf -c -D name=dsk/dX,localonly=true
```

## 클러스터 외부의 클라이언트 구성을 위한 요구 사항

Sun Cluster 및 공유 Sun StorageTek QFS 클라이언트를 포함하는 구성의 복잡성 때문에 개별 사설 네트워크는 Sun StorageTek QFS 메타 데이터 트래픽에 필수입니다. 또한 다음 사항이 일치해야 합니다.

- 메타 데이터 서버(MDS) 서버에 사용된 Sun Cluster 노드는 IPMP 그룹을 사용해야 합니다.
- Sun Cluster Sun StorageTek QFS 자원 그룹은 MDS와 클러스터 외부에 존재하는 클라이언트 사이의 메타 데이터 통신을 위해 구성된 논리적 호스트 이름을 가지고 있어야 합니다.
- Sun Cluster Sun StorageTek QFS 자원 그룹은 논리적 호스트 이름 자원과 Sun StorageTek QFS 파일 시스템 자원 사이의 종속성 등록 정보를 사용해야 합니다. 예를 들어, Sun StorageTek QFS 파일 시스템 자원은 논리적 호스트 이름 자원에 종속적입니다.
- ma, mr 및 gx 장치만 지원됩니다.
- 이러한 구성을 위해 지원되는 소프트웨어 볼륨 관리자는 없습니다.
- 기타 Sun Cluster 에이전트는 지원되지 않습니다.

## 최소 소프트웨어 릴리스 레벨

다음 최소 소프트웨어 릴리스 레벨이 요구됩니다.

- Sun Cluster 3.1 업데이트 3, 업데이트 4 또는 Sun Cluster 3.2(SPARC 및 AMD)
- 패치 03이 포함된 공유 Sun StorageTek QFS 4.5 또는 Sun StorageTek QFS 4.6(SUNWqfsr, SUNWqfsu 패키지)
- Solaris 9 업데이트 8(SPARC 및 X86 클라이언트 전용) 또는 Solaris 10(SPARC 및 AMD). 기타 운영 체제는 지원되지 않습니다.

## 지원되는 하드웨어 아키텍처

다음 하드웨어 아키텍처가 지원됩니다.

---

주 - 혼합 아키텍처는 지원되지 않습니다.

---

- SPARC
- AMD64

## 저장소 요구 사항

공유 저장소 구성은 RAID5 지원과 함께 하드웨어 수준의 미러링을 포함해야 합니다. 서버 및 클라이언트는 Sun StorageTek Traffic Manager(MPxIO) 구성을 사용해야 하고 공유 저장소만 지원됩니다.

## 구성 지침

다음 예제는 SPARC Sun Cluster 노드를 이루는 구성을 보여줍니다.

```
ctelab30    MDS #SPARC Sun Cluster Node
ctelab31    MDS #SPARC Sun Cluster Node
ctelab32    MDC #SPARC QFS Client Node
```

### ▼ /etc/hosts 파일 편집

- 운영 체제 설치 후에 각 노드의 /etc/hosts 파일을 편집하여 노드를 준비합니다.

예를 들어, 다음과 같습니다.

```
### SC Cluster Nodes ###
129.152.4.57 ctelab30 # Cluster Node
129.152.4.58 ctelab31 # Cluster Node
129.152.4.59 ctelab32 # QFS Client Node

### SC Logical ###
192.168.4.100 sc-qfs1

### QFS NET ### ##
## ctelab30
192.168.4.20 ctelab30-4
192.168.4.160 ctelab30-qfe1-test
192.168.4.210 ctelab30-qfe2-test

## ctelab31
192.168.4.21 ctelab31-4
192.168.4.161 ctelab31-qfe1-test
192.168.4.211 ctelab31-qfe2-test

## ctelab32
192.168.4.22 ctelab32-qfs
```

## ▼ 메타 데이터 서버 네트워크 구성

다음 예제는 서버 네트워크의 설정 프로세스를 설명합니다. 이 예제는 다음 설정을 적용합니다.

- Sun Cluster 개인용 상호 연결에 사용된 ce0, ce1
- 공용 네트워크에 사용된 qfe0(129.152.4.0)
- Sun StorageTek QFS 개인용 네트워크와 함께 사용할 IPMP 그룹에 사용된 qfe1, qfe2

이 예제에서 /etc/hosts, /etc/netmasks, /etc/nsswitch.conf, /etc/hostname.qfe1 및 /etc/hostname.qfe2 파일은 다음과 같이 각 서버 클러스터 노트에서 수정되어야 합니다.

1. /etc/nsswitch.conf 파일을 확인합니다.

예를 들어, 다음과 같습니다.

```
hosts: cluster files dns nis
```

2. /etc/netmasks 파일에 다음을 추가합니다.

```
192.168.4.0 255.255.255.0
```

3. 다음 내용을 포함하도록 /etc/hostname.qfe1 파일을 편집합니다.
 

```
ctelab30-4 netmask + broadcast + group qfs_ipmp1 up addif
ctelab30-qfe1-test deprecated -failover netmask + broadcast +
up
```
4. 다음 내용을 포함하도록 /etc/hostname.qfe2 파일을 편집합니다.
 

```
ctelab30-qfe2-test netmask + broadcast + deprecated group
qfs_ipmp1 -failover standby up
```

## ▼ 메타 데이터 클라이언트 네트워크 구성

다음 예제는 클라이언트 네트워크의 설정 프로세스를 설명합니다. 이 예제는 다음 설정을 적용합니다.

- 공용 네트워크에 사용된 qfe0(129.152.4.0)
- Sun StorageTek QFS 사설 네트워크에 사용된 qfe1. (192.168.4.0)

이 예제에서 /etc/hosts, /etc/netmasks, /etc/nsswitch.conf, /etc/hostname.qfe1 및 /etc/hostname.qfe2 파일은 다음과 같이 각 메타 데이터 제어기(MDC) 노드에서 수정되어야 합니다.

1. /etc/nsswitch.conf 파일을 확인하고 다음과 같이 수정합니다.
 

```
hosts: files dns nis
```
2. /etc/netmasks 파일에 다음을 추가합니다.
 

```
192.168.4.0 255.255.255.0
```
3. 다음 내용을 포함하도록 /etc/hostname.qfe1 파일을 편집합니다.
 

```
ctelab32-4
```

## ▼ Sun Cluster 설치 및 구성

운영 체제가 준비되고 노드에서 MPxIO 다중 경로 지정 소프트웨어가 활성화되면 다음과 같이 Sun Cluster 소프트웨어를 설치하고 구성할 수 있습니다.

1. **Sun Cluster** 설명서에 따라 **Sun Cluster** 소프트웨어를 설치합니다.
2. 퀵럼 장치로 사용할 공유 저장 장치를 확인합니다.
 

예를 들어, 다음과 같습니다.

```
sccdidadm -L
scconf -a -q globaldev=dx
scconf -c -q reset
```

## ▼ Sun StorageTek QFS 메타 데이터 서버 구성

Sun Cluster 소프트웨어가 설치되고 클러스터 구성이 검증되면 다음과 같이 Sun StorageTek QFS MDS를 설치하고 구성할 수 있습니다.

1. Sun StorageTek QFS 설치 및 업그레이드 안내서의 지침에 따라 **Sun StorageTek QFS** 소프트웨어를 설치합니다.

예를 들어, 다음과 같습니다.

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

2. **Sun Cluster** 명령 `scdidadm -L`을 사용하여 **Sun StorageTek QFS** 구성에 사용될 장치를 확인합니다.

3. 파일 시스템 장치를 반영하도록 `mcf` 파일을 편집합니다.

예를 들어, 다음과 같습니다.

```
#
# File system Qfs1 #
Qfs1 2 maQfs1onshared
/dev/did/dsk/d7s0 20 mmQfs1on
/dev/did/dsk/d8s0 21 mmQfs1on
/dev/did/dsk/d16s0 22 mrQfs1on
/dev/did/dsk/d10s0 23 mrQfs1on
/dev/did/dsk/d13s0 24 mrQfs1on
#
# File system Qfs2 #
Qfs2 5 maQfs2onshared
/dev/did/dsk/d9s0 50 mmQfs2on
/dev/did/dsk/d11s0 51 mmQfs2on
/dev/did/dsk/d17s0 52 mrQfs2on
/dev/did/dsk/d12s0 53 mrQfs2on
/dev/did/dsk/d14s0 54 mrQfs2on
/dev/did/dsk/d15s0 55 mrQfs2on
/dev/did/dsk/d18s0 56 mrQfs2on
```

4. **MDS Sun StorageTek QFS** 데이터 장치에 로컬 모드를 설정합니다.

예를 들어, 위에 정의된 `Qfs1` 파일 시스템의 경우 `mr` 장치로 정의된 장치에 대해 다음 작업을 수행합니다.

```
#!/usr/cluster/bin/scconf -r -D name=dsk/d16,nodelist=ctelab31
#!/usr/cluster/bin/scconf -c -D name=dsk/d16,localonly=true
#!/usr/cluster/bin/scconf -r -D name=dsk/d10,nodelist=ctelab31
#!/usr/cluster/bin/scconf -c -D name=dsk/d10,localonly=true
#!/usr/cluster/bin/scconf -r -D name=dsk/d13,nodelist=ctelab31
#!/usr/cluster/bin/scconf -c -D name=dsk/d13,localonly=true
```

5. /etc/opt/SUNWsamfs/defaults.conf 파일을 편집합니다.  
예를 들어, 다음과 같습니다.

```
trace
all = on
sam-fsd.size = 10M
sam-sharefsd.size = 10M
endtrace
```

6. Sun StorageTek QFS 파일 시스템 호스트 파일을 작성합니다.

호스트 파일에 대한 정보는 Sun StorageTek QFS 설치 및 업그레이드 안내서 및 [57페이지의 "공유 호스트 파일 변경"](#)을 참조하십시오.

---

주 – 클러스터 외부의 MDC와 통신하기 때문에 네트워크 상에 Sun StorageTek QFS 메타 데이터 트래픽을 설정해야 합니다. MDC는 Sun Clustere 구성의 일부가 아니므로 이 트래픽을 위해 논리 호스트를 사용합니다. 이 예제 구성에서 sc-qfs1은 이 호스트의 이름입니다.

---

MDS에 공유 호스트 테이블을 작성하려면 다음 작업을 수행하십시오.

- a. Sun Cluster scconf 명령을 사용하여 호스트 순서에 대한 정보를 얻으십시오. 예를 들어, 다음과 같습니다.

```
# /usr/cluster/bin/scconf -p | egrep Cluster node name: |Node private
hostname: |Node ID:
```

- b. scconf 명령 결과를 참조하십시오. 예를 들어, 다음과 같습니다.

```
Cluster node name:   ctelab30
Node ID:            1
Node private hostname:  clusternode1-priv

Cluster node name:   ctelab31
Node ID:            2
Node private hostname:  clusternode2-priv
```

c. 공유 호스트 파일을 만듭니다.

예를 들어, /etc/opt/SUNWsamfs/hosts.Qfs1 파일에는 다음과 같은 내용이 포함됩니다.

```
#
# MDS
# Shared MDS Host file for family set 'Qfs1'
#
#
ctelab30 clusternode1-priv,sc-qfs11-server
ctelab31 clusternode2-priv,sc-qfs12-
ctelab32 ctelab32-4--
```

d. 로컬 호스트 파일을 만듭니다.

예를 들어, /etc/opt/SUNWsamfs/hosts.Qfs1.local 파일은 다음과 같은 내용이 포함됩니다.

```
#
# MDS
# Local MDS Host file for family set 'Qfs1'
ctelab30 clusternode1-priv
ctelab31 clusternode2-priv
```

7. sammkfs 명령을 사용하여 파일 시스템을 만듭니다.

예를 들어, 다음과 같습니다.

```
# /opt/SUNWsamfs/sbin/sammkfs -S Qfs1
```

8. 각 클러스터 노드에 마운트 지점을 준비합니다.

예를 들어, 다음과 같습니다.

```
# mkdir -p /cluster/qfs1 /cluster/qfs2
```

9. /etc/vfstab 파일에 파일 시스템 항목을 추가합니다.

예를 들어, 다음과 같습니다.

```
###
# QFS Filesystems
###
Qfs1 - /cluster/qfs1 samfs - no shared
Qfs2 - /cluster/qfs2 samfs - no shared
```

10. 파일 시스템을 마운트합니다.

예를 들어, 다음과 같습니다.

```
# mount Qfs1, mount Qfs2 no each cluster node
```

11. Sun Cluster MDS 자원 그룹을 만듭니다.

Sun Cluster에 MDS 자원 그룹을 만들려면 다음 단계를 수행합니다.

a. QFS 자원 유형을 추가합니다.

예를 들어, 다음과 같습니다.

```
# /usr/cluster/bin/scrgadm -a -t SUNW.qfs
```

b. MDS 자원 그룹을 만듭니다.

예를 들어, 다음과 같습니다.

```
# /usr/cluster/bin/scrgadm -a -g sc-qfs-rg -h ctelab30,ctelab31
# /usr/cluster/bin/scrgadm -c -g sc-qfs-rg -y RG_description= Metadata Server
+ MDC Clients
```

c. 자원 그룹에 논리적 호스트 이름을 추가합니다.

예를 들어, 다음과 같습니다.

```
# /usr/cluster/bin/scrgadm -a -L -g sc-qfs-rg -l sc-qfs1 -n
qfs_ipmp1@ctelab30,qfs_ipmp1@ctelab31
# /usr/cluster/bin/scrgadm -c -j sc-qfs1 -y RG_description= Logical Hostname
resource for sc-qfs1
```

d. MDS 자원 그룹에 Sun StorageTek QFS 파일 시스템 자원을 추가합니다.

예를 들어, 다음과 같습니다.

```
# /usr/cluster/bin/scrgadm -a -g sc-qfs-rg -t SUNW.qfs -j fs-qfs-rs -x \
# QFSFileSystem=/cluster/qfs1,/cluster/qfs2 -y Resource_dependencies=sc-qfs1
```

e. 자원 그룹을 온라인으로 만듭니다.

예를 들어, 다음과 같습니다.

```
# /usr/cluster/bin/scswitch -Z -g sc-qfs-rg
```

f. 상태를 점검합니다.

예를 들어, 다음과 같습니다.

```
# /usr/cluster/bin/scswitch
```

## ▼ Sun StorageTek QFS 메타 데이터 클라이언트 구성

운영 체제가 모든 메타 데이터 클라이언트에 설치된 후에 Sun StorageTek QFS 클라이언트 설치 및 구성을 진행할 수 있습니다.

이 지침을 수행하기 전에 MPxIO가 활성화되어 있고 클라이언트가 모든 디스크 장치에 액세스할 수 있는지 확인하십시오.

1. Sun StorageTek QFS 설치 및 업그레이드 안내서의 지침에 따라 **Sun StorageTek QFS** 소프트웨어를 설치합니다.

예를 들어, 다음과 같습니다.

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

2. MDC의 `format` 명령 및 MDS의 `Sun Cluster scdidadm -L` 명령을 사용하여 **Sun StorageTek QFS** 구성에 사용될 장치를 확인합니다.

3. 메타 데이터 클라이언트에 `mcf` 파일을 작성합니다.

예를 들어, 다음과 같습니다.

```
#
# File system Qfs1
#
Qfs1      2  ma Qfs1 on shared
nodev    20  mm Qfs1 on
nodev    21  mm Qfs1 on
/dev/dsk/c6t600C0FF00000000000332B21D0B90000d0s0 22  mr Qfs1 on
/dev/dsk/c6t600C0FF0000000000876E9124FAF9C00d0s0 23  mr Qfs1 on
/dev/dsk/c6t600C0FF000000000004CAD7CC3CDE500d0s0 24  mr Qfs1 on
#
# File system Qfs2
# Qfs2      5  ma Qfs2 on shared
nodev    50  mm Qfs2 on
nodev    51  mm Qfs2 on
/dev/dsk/c6t600C0FF00000000000332B057D2FF100d0s0 52  mr Qfs2 on
/dev/dsk/c6t600C0FF0000000000876E975EDA6A000d0s0 53  mr Qfs2 on
/dev/dsk/c6t600C0FF0000000000876E9780ECA8100d0s0 54  mr Qfs2 on
/dev/dsk/c6t600C0FF000000000004CAD139A855500d0s0 55  mr Qfs2 on
/dev/dsk/c6t600C0FF000000000004CAD4C40941C00d0s0 56  mr Qfs2 on
```

4. /etc/opt/SUNWsamfs/defaults.conf 파일을 편집합니다.  
예를 들어, 다음과 같습니다.

```
trace
all = on
sam-fsd.size = 10M
sam-sharefsd.size = 10M
endtrace
```

5. Sun StorageTek QFS 파일 시스템 호스트 파일을 작성합니다.  
MDS 호스트 파일의 정보를 사용하여 아래 예제를 수행하십시오.

---

주 - MDS와 MDC 사이의 메타 데이터 통신의 경우 클러스터의 구성원이 아닌 클라이언트는 논리 호스트 상에서 통신해야 합니다.

---

- a. 공유 호스트 파일을 만듭니다.

예를 들어, /etc/opt/SUNWsamfs/hosts.Qfs1 파일은 다음과 같은 사항이 포함됩니다.

```
#
# MDC
# Shared Client Host file for family set 'Qfs1'
ctelab30  sc-qfs1      1      -      server
ctelab31  sc-qfs1      2      -
ctelab32  ctelab32-4    -      -
```

- b. 로컬 호스트 파일을 만듭니다.

예를 들어, /etc/opt/SUNWsamfs/hosts.Qfs1.local 파일은 다음과 같은 내용이 포함됩니다.

```
#
# MDC
# Local Client Host file for family set 'Qfs1'
ctelab30  sc-qfs1@ctelab32-4
ctelab31  sc-qfs1@ctelab32-4
```

---

주 - /etc/opt/SUNWsamfs/hosts.QFS1.local 파일은 각 클라이언트에 따라 다릅니다. 이 예제에서 클라이언트는 ctelab32-4에 구성된 인터페이스를 사용하여 메타 데이터 트래픽을 위해 호스트 sc-qfs1에 바인드합니다.

---

6. 각 클러스터 노드에 마운트 지점을 만듭니다.

예를 들어, 다음과 같습니다.

```
# mkdir -p /cluster/qfs1 /cluster/qfs2
```

7. /etc/vfstab 파일을 편집합니다.

예를 들어, 다음과 같습니다.

```
###
# QFS Filesystems
###
Qfs1 - /cluster/qfs1 samfs - yes bg,shared
Qfs2 - /cluster/qfs2 samfs - yes bg,shared
```

8. 파일 시스템을 마운트합니다.

예를 들어, 다음과 같습니다.

```
# mount Qfs1, mount Qfs2 no each MDC node
```

---

## Sun StorageTek QFS 구성 변경

이 절은 Sun Cluster 환경에서 Sun StorageTek QFS 공유 또는 공유 해제된 파일 시스템 구성을 변경, 비활성화 또는 제거하는 방법을 보여줍니다. 다음 절이 들어 있습니다.

- 173페이지의 "공유 파일 시스템 구성 변경"
- 174페이지의 "원시 전역 장치를 사용하는 파일 시스템에서 HA-NFS 비활성화"
- 175페이지의 "Solaris Volume Manager 제어 볼륨을 사용하는 파일 시스템에서 HA-NFS 비활성화"
- 176페이지의 "VxVM 제어 볼륨을 사용하는 Sun StorageTek QFS 파일 시스템에서 HA-NFS 비활성화"

## ▼ 공유 파일 시스템 구성 변경

이 예제 절차는 [132페이지의 "구성 예제"](#)의 예제를 기초로 합니다.

1. 각 노드에 oracle 사용자로서 로그인하고 데이터베이스 인스턴스를 종료하고 수신기를 중지합니다.

```
$ sqlplus "as sysdba"  
SQL > shutdown immediate  
SQL > exit  
$ lsnrctl stop listener
```

2. superuser로서 메타 데이터 서버에 로그인하고 메타 데이터 서버 자원 그룹을 관리되지 않는 상태로 만듭니다.

```
# scswitch -F -g qfs-rg  
# scswitch -u -g qfs-rg
```

이 시점에서 공유 파일 시스템이 모든 노드에서 마운트 해제됩니다. 지금 파일 시스템의 구성, 마운트 옵션 등에 대한 모든 변경을 적용할 수 있습니다. 또한 필요한 경우 파일 시스템을 다시 작성할 수도 있습니다. 파일 시스템을 다시 만든 후 사용하려면 [132페이지의 "구성 예제"](#)의 단계를 따릅니다.

3. 메타 데이터 서버 자원 그룹 구성 또는 Sun StorageTek QFS 소프트웨어를 변경하려면, 자원, 자원 그룹 및 자원 유형을 제거하고, 모든 것이 제거되었는지 확인합니다.

예를 들어 새 패키지로 업그레이드해야 합니다.

```
# scswitch -n -j qfs-res  
# scswitch -r -j qfs-res  
# scrgadm -r -g qfs-rg  
# scrgadm -r -t SUNW.qfs  
# scstat
```

이 시점에서 다른 이름, 노드 목록 등을 정의하기 위해 자원 그룹을 재작성할 수 있습니다. 또한 필요한 경우 Sun StorageTek QFS 공유 소프트웨어를 제거하거나 업그레이드할 수 있습니다. 새 소프트웨어가 설치된 후 메타 데이터 자원 그룹 및 자원을 다시 작성하고 온라인으로 만들 수 있습니다.

## ▼ 원시 전역 장치를 사용하는 파일 시스템에서 HA-NFS 비활성화

원시 전역 장치를 사용 중인 공유 해제된 Sun StorageTek QFS 파일 시스템에서 HA-NFS를 비활성화하려면 이 일반 예제 절차를 사용합니다. 이 절차 예제는 141페이지의 "예제 1: 원시 전역 장치의 HA-NFS"을 기초로 합니다.

1. `scswitch(1M) -F -g` 명령을 사용하여 자원 그룹을 오프라인으로 만듭니다.

```
# scswitch -F -g nfs-rg
```

2. NFS, Sun StorageTek QFS 및 LogicalHost 자원 유형을 비활성화합니다.

```
# scswitch -n -j nfs1-res  
# scswitch -n -j qfsnfs1-res  
# scswitch -n -j lh-nfs1
```

3. 이전에 구성한 자원을 제거합니다.

```
# scrgadm -r -j nfs1-res  
# scrgadm -r -j qfsnfs1-res  
# scrgadm -r -j lh-nfs1
```

4. 이전에 구성한 자원 그룹을 제거합니다.

```
# scrgadm -r -g nfs-rg
```

5. NFS 구성 디렉토리를 정리합니다.

```
# rm -fr /global/nfs
```

6. 이전에 추가되었고 더 이상 필요하지 않은 경우 사용된 자원 유형을 비활성화합니다.

```
# scrgadm -r -t SUNW.HAStoragePlus  
# scrgadm -r -t SUNW.LogicalHostname  
# scrgadm -r -t SUNW.nfs
```

## ▼ Solaris Volume Manager 제어 볼륨을 사용하는 파일 시스템에서 HA-NFS 비활성화

Solstice DiskSuite/Solaris Volume Manager 제어 볼륨을 사용 중인 공유 해제된 Sun StorageTek QFS 파일 시스템에서 HA-NFS를 비활성화하려면 이 일반 예제 절차를 사용합니다. 이 절차 예제는 149페이지의 "예제 2: Solstice DiskSuite/Solaris Volume Manager가 제어하는 볼륨의 HA-NFS"을 기초로 합니다.

1. 자원 그룹을 오프라인으로 만듭니다.

```
# scswitch -F -g nfs-rg
```

2. NFS, Sun StorageTek QFS 및 LogicalHost 자원 유형을 비활성화합니다.

```
# scswitch -n -j nfs1-res  
# scswitch -n -j qfsnfs1-res  
# scswitch -n -j lh-nfs1
```

3. 이전에 구성한 자원을 제거합니다.

```
# scrgadm -r -j nfs1-res  
# scrgadm -r -j qfsnfs1-res  
# scrgadm -r -j lh-nfs1
```

4. 이전에 구성한 자원 그룹을 제거합니다.

```
# scrgadm -r -g nfs-rg
```

5. NFS 구성 디렉토리를 정리합니다.

```
# rm -fr /global/nfs
```

6. 이전에 추가되었고 더 이상 필요하지 않은 경우 사용된 자원 유형을 비활성화합니다.

```
# scrgadm -r -t SUNW.HAStoragePlus  
# scrgadm -r -t SUNW.LogicalHostname  
# scrgadm -r -t SUNW.nfs
```

7. RAID-5 및 RAID-1 세트를 삭제합니다.

```
# metaclear -s nfsdg -f d30 d20 d21 d22 d23 d11 d1 d2 d3 d4
```

8. 드라이브 오류의 중재 감지를 제거합니다.

```
# metaset -s nfsdg -d -m scnode-A  
# metaset -s nfsdg -d -m scnode-B
```

9. nfsdg 디스크 그룹에서 공유 DID 장치를 제거합니다.

```
# metaset -s nfsdg -d -f /dev/did/dsk/d4 /dev/did/dsk/d5 \  
/dev/did/dsk/d6 /dev/did/dsk/d7
```

10. Sun Cluster 시스템의 노드 사이에서 디스크 그룹 nfsdg의 구성을 제거합니다.

```
# metaset -s nfsdg -d -f -h scnode-A scnode-B
```

11. 더 이상 필요없는 경우 메타 데이터베이스를 삭제합니다.

```
# metadb -d -f /dev/dsk/c0t0d0s7  
# metadb -d -f /dev/dsk/c1t0d0s7  
# metadb -d -f /dev/dsk/c2t0d0s7
```

## ▼ VxVM 제어 볼륨을 사용하는 Sun StorageTek QFS 파일 시스템에서 HA-NFS 비활성화

VxVM 제어 볼륨을 사용 중인 공유 해제된 Sun StorageTek QFS 파일 시스템에서 HA-NFS를 비활성화하려면 이 일반 예제 절차를 사용합니다. 이 절차 예제는 [156페이지](#)의 "예제 3: VxVM 볼륨의 HA-NFS"을 기초로 합니다.

1. 자원 그룹을 오프라인으로 만듭니다.

```
# scswitch -F -g nfs-rg
```

2. NFS, Sun StorageTek QFS 및 LogicalHost 자원 유형을 비활성화합니다.

```
# scswitch -n -j nfs1-res
# scswitch -n -j qfsnfs1-res
# scswitch -n -j lh-nfs1
```

3. 이전에 구성한 자원을 제거합니다.

```
# scrgadm -r -j nfs1-res
# scrgadm -r -j qfsnfs1-res
# scrgadm -r -j lh-nfs1
```

4. 이전에 구성한 자원 그룹을 제거합니다.

```
# scrgadm -r -g nfs-rg
```

5. NFS 구성 디렉토리를 정리합니다.

```
# rm -fr /global/nfs
```

6. 이전에 추가되었고 더 이상 필요하지 않은 경우 사용된 자원 유형을 비활성화합니다.

```
# scrgadm -r -t SUNW.HAStoragePlus
# scrgadm -r -t SUNW.LogicalHostname
# scrgadm -r -t SUNW.nfs
```

7. 서브디스크를 삭제합니다.

```
# vxdg destroy nfsdg
```

8. VxVM 장치를 제거합니다.

```
# vxdisk rm fabric_0 fabric_1 fabric_2 fabric_3 fabric_4
```

---

# Sun Cluster를 사용한 고가용성 Sun StorageTek SAM 구성

Sun StorageTek SAM은 또한 Sun Cluster 소프트웨어를 사용하여 고가용성을 위한 시스템으로 구성할 수 있습니다. 기본 노드에 장애가 발생하면 클러스터의 다른 노드가 자동으로 아카이빙 작업 부하를 호스트할 수 있도록 함으로써 Sun Cluster 소프트웨어는 작동 중단 시간을 크게 줄이고 생산성을 높일 수 있습니다.

고가용성 SAM(HA-SAM)은 Sun StorageTek QFS Sun Cluster 에이전트에 따라 다르므로 이 구성은 Sun StorageTek QFS Sun Cluster 에이전트에 의해 마운트되고 관리되는 공유 Sun StorageTek QFS 파일 시스템과 함께 설치되어야 합니다.

자세한 내용은 Sun StorageTek Storage Archive Manager 아카이브 구성 및 관리 안내서를 참조하십시오.

# 고급 항목

---

이 장에서는 기본적인 시스템 관리 및 사용법 수준 이상의 고급 항목을 다룹니다. 본 장에는 다음 절이 포함되어 있습니다.

- 179페이지의 "데몬, 프로세스 및 추적 사용"
- 184페이지의 "setfa(1) 명령을 사용하여 파일 속성 설정"
- 186페이지의 "WORM-FS 파일 시스템 구성"
- 201페이지의 "용량이 큰 파일 작업"
- 202페이지의 "다중판독기 파일 시스템 구성"
- 204페이지의 "이기종 컴퓨팅 환경에서 SAN-QFS 파일 시스템 사용"
- 213페이지의 "I/O 유형 이해"
- 214페이지의 "대형 파일에 대한 파일 전송 성능 향상"
- 217페이지의 "Qwrite 기능 활성화"
- 218페이지의 "쓰기 스로틀 설정"
- 219페이지의 "Flush-Behind 속도 설정"
- 219페이지의 "inode 수 및 inode 해시 테이블 조정"

---

## 데몬, 프로세스 및 추적 사용

시스템 데몬 및 프로세스를 이해하고 있으면 디버그할 때 유용합니다. 이 절에서는 Sun StorageTek QFS 데몬 및 프로세스에 대해 설명합니다. 또한 데몬 추적에 대한 정보를 제공합니다.

## 데몬 및 프로세스

모든 Sun StorageTek QFS 데몬은 `sam-daemon_named` 형식으로 이름이 지정됩니다. 프로세스도 이와 유사한 방식으로 이름이 지정되지만 마지막에 소문자 `d`가 붙지 않습니다.

표 7-1은 시스템에서 실행할 수 있는 몇 가지 데몬 및 프로세스를 표시합니다. `sam-genericd` 및 `sam-catserved` 등의 다른 데몬 및 프로세스도 시스템 작업에 따라 실행 중일 수 있습니다.

표 7-1 데몬 및 프로세스

프로세스	설명
<code>sam-fsd</code>	마스터 데몬입니다.
<code>sam-sharefsd</code>	Sun StorageTek QFS 공유 파일 시스템 데몬을 호출합니다.
<code>sam-rpcd</code>	원격 프로시저 호출(RPC) 애플리케이션 프로그래밍 인터페이스(API) 서버 프로세스를 제어합니다.

Sun StorageTek QFS 소프트웨어를 실행할 때 `init`는 `/etc/inittab` 프로세스의 일부로 `sam-fsd` 데몬을 시작합니다. 이 데몬은 `init` 레벨 0, 2, 3, 4, 5 및 6에서 시작되며, 장애가 있는 경우 자동으로 재시작해야 합니다.

Sun StorageTek QFS 공유 파일 시스템에서 `sam-fsd` 데몬은 항상 활성화되어 있습니다. 또한 각 마운트된 공유 파일 시스템에 대해 하나의 `sam-sharefsd` 데몬이 활성화되어 있습니다.

Sun StorageTek QFS 공유 파일 시스템이 마운트되어 있을 경우 소프트웨어는 공유 파일 시스템 데몬(`sam-sharefsd`)을 시작합니다. 서버와 클라이언트 호스트 간의 통신을 위해 TCP 소켓이 사용됩니다. 메타 데이터 서버에 연결하는 모든 클라이언트는 호스트 파일에 대해 유효합니다.

---

주 - 호스트 파일에 대한 자세한 내용은 `hosts.fs(4)` 매뉴얼 페이지를 참조하십시오.

---

메타 데이터 서버의 `sam-sharedfsd` 데몬이 일명 `sam-qfs` 포트에 있는 수신 소켓을 엽니다. Sun StorageTek QFS 설치 프로세스 동안 `sam-qfs` 항목이 `/etc/services` 파일에 자동으로 추가됩니다. 이 항목을 제거하지 마십시오. 또한 공유 파일 시스템 포트가 `/etc/inet/services` 파일에 포트 번호 7105로 정의됩니다. 이 포트가 다른 서비스와 충돌하지 않는지 확인합니다.

---

주 - Sun StorageTek QFS 4U2 릴리스 이전의 릴리스에 대해서는 파일 시스템당 하나의 포트가 필요합니다. 파일에서 이 항목을 제거할 수 있습니다.

---

모든 메타 데이터 작업, 블록 할당 및 할당 해제 및 레코드 잠금은 메타 데이터 서버에서 수행됩니다. `sam-sharefsd` 데몬은 어떠한 정보도 보유하지 않습니다. 따라서 파일 시스템에 대한 일관성 문제를 일으키지 않고 중지하거나 다시 시작할 수 있습니다.

## 추적 파일

여러 Sun StorageTek QFS 프로세스는 추적 파일에 메시지를 쓸 수 있습니다. 이러한 메시지는 데몬에 의해 수행된 작업의 상태 및 진행 상황에 대한 정보를 포함하고 있습니다. 메시지는 주로 Sun Microsystems 직원이 성능을 향상시키고 문제를 진단하는 데 사용됩니다. 메시지 내용 및 형식은 릴리스마다 변경될 수 있습니다.

추적 파일은 디버깅에 사용될 수 있습니다. 기본적으로 추적 파일은 활성화되지 않습니다. `defaults.conf` 파일을 편집하여 추적 파일을 활성화할 수 있습니다. 모든 프로세스에 대해 추적을 활성화하거나 개별 프로세스에 대해 추적을 활성화할 수 있습니다. 추적할 수 있는 프로세스에 대한 정보는 `defaults.conf(4)` 매뉴얼 페이지를 참조하십시오.

기본적으로 추적 파일은 `/var/opt/SUNWsamfs/trace` 디렉토리에 기록됩니다. 해당 디렉토리에서 추적 파일이 프로세스(`archiver`, `catserver`, `fsd`, `ftpd`, `recycler`, `sharefsd` 및 `stager`)에 대해 명명됩니다. `defaults.conf` 구성 파일에서 명령을 지정하여 추적 파일의 이름을 변경할 수 있습니다. 또한 추적 파일의 크기에 대한 제한을 설정하고 추적 로그를 회전할 수 있습니다. 추적 제어에 대한 내용은 `defaults.conf(4)` 매뉴얼 페이지를 참조하십시오.

## 추적 파일 내용

추적 파일 메시지는 메시지의 시간 및 소스가 포함되어 있습니다. 메시지는 프로세스의 이벤트에 의해 생성됩니다. `defaults.conf` 파일의 명령을 사용하여 해당 이벤트를 선택할 수 있습니다.

기본 이벤트는 다음과 같습니다.

- 고객 통지 `syslog` 또는 `notify` 파일 메시지
- 치명적이지 않은 프로그램 오류
- 치명적인 `syslog` 메시지
- 프로세스 초기화 및 완료
- 기타 이벤트

또한 다음 이벤트를 추적할 수 있습니다.

- 메모리 할당
- 프로세스간 통신
- 파일 작업
- 운영자 메시지
- 변경될 때의 대기열 내용
- 기타 이벤트

기본 메시지 요소(프로그램 이름, 프로세스 ID(PID) 및 시간)는 항상 포함되며 제외시킬 수 없습니다. 선택적으로 메시지는 다음 요소도 포함할 수 있습니다.

- 날짜(시간은 항상 포함됨)
- 소스 파일 이름 및 행 번호
- 이벤트 유형

## 추적 파일 교환

추적 파일이 무한대로 커지는 것을 막기 위해 `sam-fsd` 데몬은 추적 파일의 크기를 모니터링하고 정기적으로 다음 명령을 실행합니다.

```
/opt/SUNWsamfs/sbin/trace_rotate
```

이 스크립트를 실행하면 추적 파일이 연속된 번호로 지정된 복사본으로 이동됩니다. 이 스크립트는 해당 작업에 적합하게 수정할 수 있습니다. `cron(1)` 또는 기타 장치를 사용하여 이러한 기능을 제공할 수도 있습니다.

## 추적되고 있는 프로세스 확인

현재 어떤 프로세스가 추적되고 있는지 확인하려면 명령줄에 `sam-fsd(1M)` 명령을 입력합니다. [코드 예 7-1](#)은 이 명령의 출력 결과를 표시합니다.

```

# sam-fsd
Trace file controls:
sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
              cust err fatal misc proc date
              size  0    age  0
sam-archiverd /var/opt/SUNWsamfs/trace/sam-archiverd
              cust err fatal misc proc date
              size  0    age  0
sam-catserverd /var/opt/SUNWsamfs/trace/sam-catserverd
              cust err fatal misc proc date
              size  0    age  0
sam-fsd       /var/opt/SUNWsamfs/trace/sam-fsd
              cust err fatal misc proc date
              size  0    age  0
sam-rftd      /var/opt/SUNWsamfs/trace/sam-rftd
              cust err fatal misc proc date
              size  0    age  0
sam-recycler  /var/opt/SUNWsamfs/trace/sam-recycler
              cust err fatal misc proc date
              size  0    age  0
sam-sharefsd  /var/opt/SUNWsamfs/trace/sam-sharefsd
              cust err fatal misc proc date
              size  0    age  0
sam-stagerd   /var/opt/SUNWsamfs/trace/sam-stagerd
              cust err fatal misc proc date
              size  0    age  0
sam-serverd   /var/opt/SUNWsamfs/trace/sam-serverd
              cust err fatal misc proc date
              size  0    age  0
sam-clientd   /var/opt/SUNWsamfs/trace/sam-clientd
              cust err fatal misc proc date
              size  0    age  0
sam-mgmt      /var/opt/SUNWsamfs/trace/sam-mgmt
              cust err fatal misc proc date
              size  0    age  0

```

추적 파일 활성화에 대한 자세한 내용은 defaults.conf(4) 매뉴얼 페이지 및 sam-fsd(1M) 매뉴얼 페이지를 참조하십시오.

## setfa(1) 명령을 사용하여 파일 속성 설정

Sun StorageTek QFS 파일 시스템을 사용하면 최종 사용자가 파일 및 디렉토리에 대한 성능 속성을 설정할 수 있습니다. 응용 프로그램은 파일 또는 디렉토리를 기본으로 이러한 성능 기능을 활성화할 수 있습니다. 다음 절에서는 응용 프로그램 프로그래머가 이러한 기능을 사용하여 파일 및 디렉토리에 대한 파일 속성 선택, 파일 공간 사전 할당, 파일에 대한 할당 방식 지정, 파일 스트라이프 너비 지정 등을 수행하는 방법에 대해 설명합니다.

다음 하위 절에 설명된 기능 구현에 대한 자세한 정보는 setfa(1) 매뉴얼 페이지를 참조하십시오.

### 파일 및 디렉토리에 대한 파일 속성 선택

setfa(1) 명령은 새 파일 또는 기존 파일에 대한 속성을 설정합니다. 파일이 없는 경우에는 파일이 새로 만들어집니다.

파일 뿐만 아니라 디렉토리에 대해서도 속성을 설정할 수 있습니다. 디렉토리에 대해 setfa(1)를 사용하는 경우, 해당 디렉토리 내에 만들어지는 파일 및 디렉토리는 원래의 디렉토리에서 설정된 속성을 상속 받습니다. 파일 또는 디렉토리의 속성을 기본값으로 재설정하려면 -d(기본) 옵션을 사용하십시오. -d 옵션이 사용되면 이 속성이 기본값으로 재설정된 후 다른 속성이 처리됩니다.

### 파일 공간 사전 할당

최종 사용자는 파일에 대한 공간을 미리 할당할 수 있습니다. 이 공간은 특정 파일과 연결되므로 파일 시스템의 다른 파일은 해당 파일에 할당된 디스크 주소를 사용할 수 없습니다. 사전 할당을 사용하면 주어진 파일에 대한 공간을 사용할 수 있으며 이것은 파일 시스템이 꼭 차는 것을 방지합니다. 사전 할당은 데이터가 실제로 디스크에 쓰여지는 시점이 아니라 요청 시점에 수행됩니다.

파일을 사전 할당하면 공간이 낭비될 수 있다는 점에 유의하십시오. 파일 크기가 할당량보다 작으면, 커널이 파일에 대해 현재의 파일 크기에서 할당량까지 공간을 할당합니다. 파일이 닫히면 할당량 아래의 공간은 사용 가능한 공간이 되지 않습니다.

-L 또는 -l(소문자 L) 옵션과 함께 setfa(1) 명령을 사용하여 파일에 대한 공간을 사전 할당할 수 있습니다. 두 옵션 모두 파일 길이를 인수로 허용합니다. 비어 있거나 데이터를 포함할 수 있는 기존 파일에 대해서는 -L 옵션을 사용하십시오. 아직 데이터가 없는 파일에 대해 -l 옵션을 사용합니다. -l 옵션을 사용하는 경우 파일은 사전 할당된 제한을 넘을 수 없습니다.

예를 들어 /qfs/file\_alloc이라는 1기가바이트 파일을 사전 할당하려면 다음과 같이 입력합니다.

```
# setfa -l 1g /qfs/file_alloc
```

파일에 대한 공간이 사전 할당된 후, 파일을 0 길이로 자르거나 파일을 제거하면 해당 파일에 할당된 모든 공간을 되찾을 수 있습니다. 파일 시스템에서 파일에 사전 할당된 공간 중 일부만을 되찾을 수 있는 방법은 없습니다. 또한 파일이 -l 옵션으로 사전 할당 되면 향후 작업에서 사전 할당된 크기를 초과하여 파일을 확장할 수 없습니다.

## 파일 할당 방식 및 스트라이프 너비 선택

기본적으로 파일은 마운트 시점에 지정된 할당 방식 및 스트라이프 너비를 사용합니다 (mount\_samfs(1M) 매뉴얼 페이지 참조). 그러나 최종 사용자가 파일 또는 디렉토리에 다른 할당 체계를 사용하고자 할 수 있습니다. 사용자는 setfa(1) 명령을 -s(스트라이프) 옵션과 함께 사용하여 이를 수행할 수 있습니다.

할당 방식은 라운드 로빈이나 스트라이프로 지정할 수 있습니다. -s 옵션은 표 7-2에 표시된 것처럼 할당 방식과 스트라이프 너비를 지정합니다.

표 7-2 파일 할당 및 스트라이프 너비

-s 옵션	할당 방식	스트라이프 너비	설명
0	라운드 로빈	해당 사항 없음	파일이 한 장치에 더 이상 공간이 없을 때까지 해당 장치에 할당됩니다.
1-255	스트라이프	1-255 DAU	파일은 각 디스크마다 이 DAU 수로 모든 디스크 장치에 걸쳐 스트라이프됩니다.

다음 예제는 라운드 로빈 할당 방법을 지정하여 파일을 명시적으로 만드는 방법을 표시합니다.

```
# setfa -s 0 /qfs/100MB.rrobin
```

다음 예제는 스트라이프 너비 64 DAU를 사용하여 스트라이프 할당 방법을 지정하여 파일을 명시적으로 만드는 방법을 표시합니다(사전 할당이 사용되지 않음).

```
# setfa -s 64 /qfs/file.stripe
```

## 스트라이프 그룹 장치 선택

스트라이프 그룹 장치는 Sun StorageTek QFS 파일 시스템에 대해서만 지원됩니다.

사용자는 파일이 특정 스트라이프 그룹에서 할당이 시작되도록 지정할 수 있습니다. 파일 할당 방식이 라운드 로빈인 경우, 파일은 지정된 스트라이프 그룹에서 할당됩니다.

**코드 예 7-2**는 file1 및 file2가 두 개의 다른 스트라이프 그룹에 독립적으로 분산되도록 지정하는 setfa(1) 명령을 나타냅니다.

**코드 예 7-2**                    스트라이프 그룹에 파일을 분산하는 setfa(1) 명령

```
# setfa -g0 -s0 file1
# setfa -g1 -s0 file2
```

이 기능은 원래의 장치 속도에 해당하는 성능 수준을 얻어야 하는 응용 프로그램에서 특히 중요합니다. 자세한 내용은 setfa(1) 매뉴얼 페이지를 참조하십시오.

---

## WORM-FS 파일 시스템 구성

WORM(한 번 쓰고 많이 읽음) 기술은 데이터의 무결성 및 이 기술을 사용하는 저장된 파일의 승인된 법적 허용성으로 인해 많은 응용 프로그램에서 사용됩니다. Sun StorageTek QFS 소프트웨어의 릴리스 4U3부터는 SUNwsamfswm이라는 부가 패키지로써 WORM-FS 기능을 사용할 수 있게 되었습니다. 4U4 소프트웨어 릴리스에서 WORM-FS 인터페이스는 새 Sun StorageTek 5310 NAS 장비와 호환되도록 수정되었습니다. ssum을 사용하는 기존 WORM-FS 인터페이스는 더 이상 지원되지 않습니다.

---

주 - WORM-FS 패키지(SUNwsamfswm)는 Sun StorageTek QFS 소프트웨어 패키지와 함께 제공되지만 pkgadd 명령을 사용하여 별도로 설치해야 합니다.

---

WORM-FS 기능은 WORM 설정의 기본값 및 사용자 정의 가능 파일 보유 기간, 데이터 및 경로 불변성 및 하위 디렉토리 상속을 제공합니다.

WORM-FS는 두 가지 모드 중 하나로 동작할 수 있습니다. 기본 모드인 Sun 표준 호환 모드(여기서는 간단히 표준 모드라고 함)와 Sun 애플리케이션 호환 모드(여기서는 애플리케이션 모드라고 함)로서, 이 모드는 Sun StorageTek 5320 NAS 장비의 애플리케이션 모드와 호환되도록 설계되었고 Network Appliance가 정의한 인터페이스와 유사합니다.

표준 모드와 에플레이션 모드 사이의 한 가지 차이점은 보유될 수 있는 파일의 속성에 대한 제한입니다. 특히, 표준 모드에서 UNIX 실행 권한을 가진 파일은 보유될 수 없습니다. 에플레이션 모드에는 이러한 제한이 없습니다. 표준 모드의 제한은 NFS 및 FTP에 대해 정의된 보유 트리거의 속성 때문입니다. 이러한 프로토콜의 경우 `setuid` 모드를 파일에 설정하도록 지정함으로써 보유가 요청됩니다. 파일이 보유되면 클라이언트는 `setuid` 모드 비트 세트를 볼 수 있습니다. 하지만 실행 파일에 대한 제한으로 `root` 사용자가 소유한 실행 파일을 WORM으로 만들어 제거할 수 없게 함으로써 발생할 수 있는 보안 허점을 예방합니다. 이러한 방법의 이점은 사용자나 응용 프로그램이 시스템에서 WORM 보호 파일을 보다 쉽게 확인할 수 있다는 것입니다.

## WORM-FS 기능 활성화

WORM-FS 기능을 활성화하는 데 사용할 수 있는 4개의 마운트 옵션이 있습니다.

- `worm_capable`  
표준 WORM 모드, WORM 트리거 명령 `chmod 4000 file-name/directory-name`은 파일이나 디렉토리에 WORM 비트를 설정하는 데 사용됩니다.
- `worm_lite`  
표준 WORM 모드 제한의 일부를 완화시킵니다. 시스템 관리자는 보유 만료 전에 파일을 삭제하거나 파일 보유 기간을 줄일 수 있습니다. 파일 데이터 및 경로 무결성은 변경되지 않고 유지됩니다. 자세한 내용은 [188페이지의 "WORM "Lite" 옵션](#)을 참조하십시오.
- `worm_emul`  
WORM 에플레이션 모드는 Sun StorageTek 5320 NAS 장비의 에플레이션 모드와 호환되도록 설계되었습니다. 이 모드는 서로 다른 WORM 트리거를 사용하여 표준 WORM 기능을 제공합니다. 디렉토리나 파일을 쓰기 가능에서 읽기 전용으로 변경하면 WORM 비트가 설정됩니다.
- `emul_lite`  
WORM 에플레이션 모드는 Sun StorageTek 5320 NAS 장비의 "lite" 버전과 호환되도록 설계되었습니다. 이 모드는 서로 다른 WORM 트리거와 함께 표준 WORM lite 기능을 제공합니다. 디렉토리나 파일을 쓰기 가능에서 읽기 전용으로 변경하면 WORM 비트가 설정됩니다. `worm_lite` 옵션을 사용하여 관리자는 파일에 대한 특수한 작업을 수행할 수 있습니다. 자세한 내용은 [188페이지의 "WORM "Lite" 옵션](#)을 참조하십시오.

이 4개의 마운트 옵션은 다소 배타적입니다. "lite"에서 표준 WORM 모드로 업그레이드할 수 있지만 표준 WORM 모드에서 에플레이션 모드 또는 에플레이션 모드에서 표준 모드로 변경할 수는 없습니다. 이 옵션은 파일 시스템이 마운트될 때 명령줄에 제공되거나 `/etc/vfstab`에 나열되거나 `/opt/SUNWsamfs/samfs.cmd`에 제공될 수 있습니다. 마운트 옵션에 대한 우선 순위 규칙이 적용됩니다.

WORM 속성은 마운트 테이블에 저장되며 WORM 파일이 파일 시스템의 임의 위치에 있는 디렉토리에 생성되도록 합니다.

---

주 - /etc/vfstab에 WORM 마운트 옵션을 설정하려면 시스템 관리 권한이 있어야 합니다.

---

**코드 예 7-3**은 WORM-FS 마운트 옵션의 예를 보여줍니다. /samfs1에 마운트된 samfs1 파일 시스템은 WORM을 작동할 수 있으며 파일에 대한 기본 보유 기간은 기본적으로 60분으로 설정되어 있습니다.

**코드 예 7-3** WORM-FS 마운트 옵션 사용

```
# cat /etc/vfstab
#device device mount FS fsckmountmount
#to mount to fsck pointtypepassat bootoptions
#
fd      -   /dev/fdfd-no-
/proc   -   /procproc-no-
/dev/dsk/c0t0d0s1- -swap-no-
samfs1  -   /samfs1samfs=yes worm_capable,def_retention=60
swap    -   /tmpmpfs=yes-
```

WORM-FS 기능이 활성화되고 최소한 하나의 WORM 파일이 파일 시스템에 상주하면 파일 시스템의 수퍼 블록이 업데이트되어 WORM 기능에 반영됩니다. worm\_lite 또는 emul\_lite 마운트 옵션을 사용하지 않을 경우 sammkfs를 통해 파일 시스템을 다시 구축할 수 없습니다.

## WORM "Lite" 옵션

worm\_lite 및 emul\_lite 마운트 옵션은 변경된 WORM 환경을 생성하여 WORM이 활성화된 볼륨과 보유된 파일에서 수행할 수 있는 작업에 대한 제한을 완화시킵니다. WORM lite 옵션은 데이터 보유에 대한 보장은 필요하지만 WORM이 시스템에 엄격한 제약을 가하지 않는 문서 관리 및 보유 정책을 가진 회사를 위한 솔루션이 될 수 있습니다. 일부 데이터 보유 결정을 변경하고 심지어 역으로 뒤집기 위한 메커니즘이 존재합니다.

WORM lite 옵션은 더 엄격한 표준 WORM 정책으로 업그레이드하기 전에 WORM 시스템 및 응용 프로그램을 테스트 및 구성하는 데 사용될 수도 있습니다.

WORM lite 환경은 표준 WORM 모드와 비슷하게 동작합니다. 파일 데이터 및 경로는 변하지 않고 유지되지만 시스템 관리자는 다음과 같은 특별한 작업을 수행할 수 있습니다.

- 보유 기간이 만료되기 전에 WORM 파일을 제거합니다.
- WORM 파일의 보유 기간을 단축합니다.
- WORM Lite 활성화된 볼륨을 삭제하거나 `sammkfs` 명령을 사용하여 다시 구축합니다.

## WORM 파일 생성

WORM 마운트 옵션을 사용하면 파일 시스템이 WORM 파일을 포함할 수 있지만, 자동으로 WORM 파일을 생성하지는 않습니다. WORM 파일을 생성하려면 먼저 디렉토리를 WORM 작동 가능하게 만들어야 합니다. 이렇게 하려면 일반 디렉토리를 생성한 다음 WORM 트리거 명령을 사용하여 WORM 비트를 해당 디렉토리에 설정합니다. 사용되는 마운트 옵션에 따라 다음 WORM 트리거 명령을 이용할 수 있습니다.

- `worm_capable` 또는 `worm_lite` 마운트 옵션을 사용하고 있는 경우 `chmod 4000 directory-name`을 사용하여 WORM 비트를 설정하십시오.
- `worm_emul` 또는 `emul_lite` 마운트 옵션을 사용하고 있는 경우 WORM 비트를 설정하려면 디렉토리에서 쓰기 권한을 제거하십시오.

디렉토리에 WORM 비트를 설정한 후, 해당 디렉토리에서 파일을 생성한 다음 적절한 WORM 트리거를 사용하여 보유하려는 파일에 WORM 비트를 설정할 수 있습니다. WORM 트리거는 파일 및 디렉토리에 대해 동일합니다.

다음은 4개의 마운트 옵션 각각에 대해 시스템 범위의 기본 보유 값을 이용한 WORM 트리거의 사용 예제입니다.

## 예제 1: WORM 트리거 chmod 4000

표준 WORM 기능을 사용한 WORM 트리거의 간단한 응용:

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 -          /samfs1 samfs  -          no          bg,worm_capable

[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod 4000 WORM
[root@ns-east-44]# sls -D

WORM:
mode: drwxr-xr-x  links:  2  owner: root      group: root
length: 4096  admin id:  0  inode:  1025.1
access: Jan 30 15:50  modification: Jan 30 15:50
changed: Jan 30 15:50  attributes:  Jan  1  1970
creation: Jan 30 15:50  residence:   Jan 30 15:50
worm-capable          retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod 4000 test
[root@ns-east-44]# sls -D

test:
mode: -r-Sr--r--  links:  1  owner: root      group: root
length: 0  admin id:  0  inode:  1026.3
access: Jan 30 15:51  modification: Jan 30 15:51:00
changed: Jan 30 15:51  retention-end: Mar  1 15:51 2007
creation: Jan 30 15:51  residence:   Jan 30 15:51:00
retention: active          retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
rm: test not removed: Read-only file system
[root@ns-east-44]# ls
test
```

## 예제 2: WORM 트리거 chmod 4000

표준 WORM lite 기능을 사용한 WORM 트리거의 간단한 응용:

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 -          /samfs1 samfs -          no          bg,worm_lite

[root@ns-east-44]# mount samfs1
[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod 4000 WORM
[root@ns-east-44]# sls -D

WORM:
mode: drwxr-xr-x  links:  2  owner: root      group: root
length:          4096  admin id:  0  inode:   1025.1
access:          Jan 30 16:12  modification: Jan 30 16:12
changed:        Jan 30 16:12  attributes:   Jan 1 1970
creation:       Jan 30 16:12  residence:    Jan 30 16:12
worm-capable    retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod 4000 test
[root@ns-east-44]# sls -D

test:
mode: -r-Sr--r--  links:  1  owner: root      group: root
length:          0  admin id:  0  inode:   1026.1
access:          Jan 30 16:13  modification: Jan 30 16:13
changed:        Jan 30 16:13  retention-end: Mar  1 16:13 2007
creation:       Jan 30 16:13  residence:    Jan 30 16:13
retention:      active          retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
[root@ns-east-44]# ls
[root@ns-east-44]#
```

### 예제 3: WORM 트리거 chmod -w

WORM 에뮬레이션 모드를 사용한 WORM 트리거의 간단한 응용:

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 -          /samfs1 samfs  -          no      bg,worm_emul

[root@ns-east-44]# mount samfs1
[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod -w WORM
[root@ns-east-44]# sls -D

WORM:
mode: drwxr-xr-x  links:  2  owner: root      group: root
length:      4096  admin id:    0  inode:    1025.1
access:      Jan 30 16:26  modification: Jan 30 16:26
changed:     Jan 30 16:26  attributes:   Jan  1 1970
creation:    Jan 30 16:26  residence:    Jan 30 16:26
worm-capable      retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod -w test
[root@ns-east-44]# sls -D

test:
mode: -r--r--r--  links:  1  owner: root      group: root
length:           0  admin id:    0  inode:    1026.1
access:          Jan 30 16:26  modification: Jan 30 16:26
changed:         Jan 30 16:26  retention-end: Mar  1 16:26 2007
creation:        Jan 30 16:26  residence:     Jan 30 16:26
retention:       active      retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
rm: test not removed: Read-only file system
[root@ns-east-44]# ls
test
```

## 예제 4: WORM 트리거 chmod -w

WORM 애플리케이션 lite 모드를 사용한 WORM 트리거의 간단한 응용:

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 -          /samfs1 samfs -          no          bg,emul_lite

[root@ns-east-44]# mount samfs1
[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod -w WORM
[root@ns-east-44]# sls -D

WORM:
mode: drwxr-xr-x  links:  2  owner: root      group: root
length:          4096  admin id:  0  inode:   1025.1
access:          Jan 30 16:36  modification: Jan 30 16:36
changed:        Jan 30 16:36  attributes:   Jan 1 1970
creation:       Jan 30 16:36  residence:    Jan 30 16:36
worm-capable    retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod -w test
[root@ns-east-44]# sls -D

test:
mode: -r--r--r--  links:  1  owner: root      group: root
length:           0  admin id:  0  inode:   1026.1
access:          Jan 30 16:36  modification: Jan 30 16:36
changed:        Jan 30 16:36  retention-end: Mar 1 16:36:00 2007
creation:       Jan 30 16:36  residence:    Jan 30 16:36
retention:      active          retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
[root@ns-east-44]# ls
[root@ns-east-44]#
```

주 - WORM 트리거를 적용할 때에는 주의하십시오. 파일에 WORM 기능이 적용된 후에는 파일 데이터 및 경로를 변경할 수 없습니다. 이 기능이 파일에 적용된 후에는 취소할 수 없습니다. 또한 WORM 트리거가 파일에 적용되면 볼륨도 WORM 볼륨이 되어 동일한 방식을 유지합니다. 볼륨은 볼륨 관리 또는 RAID 인터페이스를 통해서만 제거할 수 있습니다. WORM "lite" 옵션 중 하나가 볼륨을 생성하는 데 사용되었다면 이 볼륨은 sammkfs를 사용하여 다시 구축될 수 있습니다.

## 보유 기간

또한 WORM-FS 기능은 사용자 정의할 수 있는 파일 보유 기간을 포함합니다. 파일에 보유 기간을 지정하면 지정된 기간 동안 해당 파일의 WORM 기능이 유지됩니다.

---

주 - 처음에 Solaris/UNIX 유틸리티를 사용하여 기간을 지정하거나 연장한 경우 보유 기간은 01/18/2038 이상으로 늘릴 수 없습니다. 그 이유는 이 유틸리티가 부호 있는 32비트 숫자를 사용하여 시간을 초 단위로 나타내기 때문입니다. 1970년 1월 1일을 기준으로 측정합니다. 기준 시점부터 2\*\*31초는 01/18/2038 약 10:14 PM으로 연장됩니다. 그러나 기본 보유 기간을 사용하여 이 날짜를 초과할 수 있습니다. [196페이지의 "기본 보유 기간 설정"](#)을 참조하십시오.

---

파일에 대해 보유 기간을 설정하려면 다음 중 하나를 수행하십시오.

- touch 유틸리티를 사용하거나 libc 서브루틴 utimes()를 사용하는 프로그램으로 파일의 액세스 시간을 앞당깁니다. 파일의 보유 기간은 분으로 저장됩니다. 액세스 시간이 앞당겨진 후 적절한 WORM 트리거를 사용하여 WORM 비트를 설정하십시오.
- 파일에 대해 기본 보유 기간을 사용합니다. 이는 적절한 WORM 트리거를 적용하고 파일 시스템이 기본 보유 기간을 적용하도록 허용하면 됩니다. 자세한 내용은 [196페이지의 "기본 보유 기간 설정"](#)을 참조하십시오.

**코드 예 7-4**는 파일에 WORM 트리거를 사용하고(chmod 4000 명령으로) 파일의 WORM 기능을 표시하기 위해 s1s 명령을 사용하여 WORM 가능한 디렉토리에 파일을 생성하는 과정을 보여줍니다. 이 예제는 파일 시스템의 기본 보유 기간([코드 예 7-3](#)에 설정된 대로 60분)을 사용합니다.

**코드 예 7-4** WORM 가능 디렉토리 및 WORM 파일 생성

```
# cd WORM
# echo "This is a test file" >> test
# s1s -D
test:
  mode: -rw-r--r--  links: 1  owner: root      group: other
  length: 20  admin id: 0  inode: 1027.1
  access: Oct 30 02:50  modification: Oct 30 02:50
  changed: Oct 30 02:50  attributes: Oct 30 02:50
  creation: Oct 30 02:50  residence: Oct 30 02:50

  checksum: gen  no_use  not_val  algo: 0
```

```
# chmod 4000 test
# sfs -D
test:
  mode: -r--r--r--  links: 1  owner: root      group: other
  length: 20  admin id: 0  inode: 1027.1
  access: Oct 30 02:50  modification: Oct 30 02:50
  changed: Oct 30 02:50  retention-end: Oct 30 2005 03:50
  creation: Oct 30 02:50  residence: Oct 30 02:50
  retention: active retention-period: 0y, 0d, 1h, 0m
  checksum: gen no_use not_val algo: 0
```

WORM-FS 기능을 추가하면 Sun StorageTek QFS 파일 시스템의 파일에 대해 다음 세 가지 상태가 가능합니다.

- 정상(Normal)
- 보유(Retained)
- 만료(Expired)

정상 상태는 Sun StorageTek QFS 파일 시스템의 일반 파일 상태를 표시합니다. 보유 또는 활성 상태로의 변화는 WORM 비트가 파일에 설정될 때 발생합니다. 만료 또는 종료 상태는 파일의 보유 기간이 초과될 때 발생합니다.

보유 기간이 파일에 지정되고 WORM 트리거가 적용되면 파일의 경로 및 데이터는 변경되지 않습니다. 보유 기간이 만료되면 상태가 "만료"로 변경되지만 경로 및 데이터는 계속 변경되지 않습니다.

파일이 만료 상태에 있으면 다음 두 작업만 가능합니다.

- 보유 기간 연장(보유 기간은 WORM "lite" 옵션을 사용하지 않는 경우 단축될 수 없습니다).
- 파일 삭제

보유 기간이 연장될 경우 파일의 상태는 "활성"으로 돌아가고 새 종료 날짜 및 기간이 그에 따라서 설정됩니다.

파일에 대한 하드 및 소프트 링크를 둘 다 WORM-FS 기능과 함께 사용할 수 있습니다. 하드 링크는 WORM 가능 디렉토리에 상주하는 파일에만 설정할 수 있습니다. 하드 링크가 생성된 후에는 원래 파일과 동일한 WORM 특성을 갖습니다. 소프트 링크도 설정할 수 있지만, 소프트 링크는 WORM 기능을 사용할 수 없습니다. WORM 파일에 대한 소프트 링크는 Sun StorageTek QFS 파일 시스템의 임의의 디렉토리에 생성할 수 있습니다.

WORM-FS 기능의 또 다른 속성은 디렉토리 상속입니다. WORM 속성이 포함된 디렉토리 아래에 생성된 새 디렉토리는 상위 디렉토리로부터 이 속성을 상속 받습니다. 디렉토리가 기본 보유 기간으로 설정되면 이 보유 기간도 모든 새 하위 디렉토리에 의해 상속됩니다. WORM 가능한 상위 디렉토리 내의 모든 파일에 WORM 비트를 설정할 수 있습니다. 일반 사용자가 일반 UNIX 권한을 사용하여 소유하거나 액세스할 수 있는 디렉토리 및 파일에 WORM 기능을 설정할 수 있습니다.

---

주 - WORM 가능 디렉토리는 WORM 파일이 들어있지 않을 때에만 삭제할 수 있습니다.

---

## 기본 보유 기간 설정

파일 시스템에 대한 기본 보유 기간은 `/etc/vfstab` 파일에서 마운트 옵션으로 설정할 수 있습니다. 예를 들어, 다음과 같습니다.

```
samfs1 - /samfs1 samfs - no
bg, worm_emul, def_retention=1y60d
```

기본 보유 기간 설정 형식은 *MyNdOhPm*이며, 여기서 *M*, *N*, *O* 및 *P*는 음이 아닌 임의의 정수이고 *y*, *d*, *h* 및 *m*은 각각 연도, 일, 시간 및 분을 나타냅니다. 위 단위의 모든 조합을 사용할 수 있습니다. 예를 들어 `1y5d4h3m`은 1년, 5일, 4시간, 3분을 나타내고, `30d8h`는 30일, 8시간을 나타내며, `300m`은 300분을 나타냅니다. 이 형식은 보유 기간이 분으로 지정된 4U5 이전의 소프트웨어 버전과 역호환됩니다. 기간은 분으로 세밀히 표시되지만 기간의 정확성은 하루를 기준으로 한다는 점에 유의해야 합니다. 또한 일, 시간 및 분을 처리하는 함수는 보유 기간을 결정할 때 윤년을 고려하지 않습니다. 기본 보유 기간을 설정하는 데 이 중 하나(또는 모두)를 사용할 경우 이를 고려해야 합니다.

2038년 이후로 파일이나 디렉토리의 보유 기간을 설정하는 데 기본 보유 기간을 사용할 수도 있습니다. 이렇게 하려면 기본 보유 기간을 2038을 초과하는 값으로 설정하고 파일 시스템을 마운트합니다. 그런 다음 적절한 WORM 트리거를 사용하여 기본 보유 기간을 적용하십시오. 다음은 디렉토리 및 파일의 보유 기간을 2038년 이후로 설정하기 위해 기본 보유 기간을 사용한 예입니다.

```
[root@ns-east-44]# grep samfs1 /etc/vfstab
samfs1 -          /samfs1 samfs -          no
bg,worm_capable,def_retention=34y
[root@ns-east-44]# mount samfs1
[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod 4000 WORM
[root@ns-east-44]# sls -D
WORM:
mode: drwxr-xr-x  links:  2  owner: root      group: root
length:      4096  admin id:  0  inode:   1026.1
access:      Feb 20 14:24  modification: Feb 20 14:24
changed:     Feb 20 14:24  attributes:   Jul 26 1970
creation:    Feb 20 14:24  residence:    Feb 20 14:24
worm-capable      retention-period: 34y, 0d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod 4000 test
[root@ns-east-44]# sls -D
test:
mode: -r-Sr--r--  links:  1  owner: root      group: root
length:           0  admin id:  0  inode:   1027.1
access:          Feb 20 14:24  modification: Feb 20 14:25
changed:         Feb 20 14:25  retention-end: Feb 20 14:25 2041
creation:        Feb 20 14:24  residence:     Feb 20 14:24
retention:       active      retention-period: 34y, 0d, 0h, 0m
```

다음 절인 [197페이지](#)의 "[touch를 사용하여 보유 기간 설정](#)"에 설명된 대로 touch 유틸리티를 사용하여 디렉토리에 대한 기본 보유 기간을 설정할 수도 있습니다. 이 보유 기간은 파일 시스템의 기본 보유 기간에 우선하며 하위 디렉토리로 상속됩니다.

## touch를 사용하여 보유 기간 설정

touch 유틸리티를 사용하여 파일 또는 디렉토리의 보유 기간을 설정하거나 연장할 수 있습니다. touch를 사용하여 디렉토리(파일 제외)에 대한 기본 보유 기간을 단축할 수도 있습니다.

보유 기간을 설정하려면 먼저 touch를 사용하여 파일이나 디렉토리의 액세스 시간을 앞당긴 다음 chmod 명령을 사용하거나 쓰기 권한(해당 시점의 WORM 모드에 따라 다름)을 제거하여 WORM 트리거를 적용해야 합니다.

코드 예 7-6은 touch 유틸리티를 사용한 파일 보유 기간 설정 및 WORM 트리거 적용을 보여줍니다.

코드 예 7-6 touch 및 chmod를 사용하여 보유 기간 설정

```
# touch -a -t200508181125 test
# sls -D
test:
  mode: -rw-r--r--  links: 1  owner: root      group: root
  length:          0  admin id: 0  inode: 1027.1
  access:          Aug 18 2005  modification: Aug 18 11:19
  changed:         Aug 18 11:19  attributes:   Aug 18 11:19
  creation:        Aug 18 11:19  residence:   Aug 18 11:19

# chmod 4000 test
# sls -D
test:
  mode: -r-Sr--r--  links: 1  owner: root      group: root
  length:          0  admin id: 0  inode: 1027.1
  access:          Aug 18 2005  modification: Aug 18 11:19
  changed:         Aug 18 11:19  retention-end: Aug 18 2005 11:25
  creation:        Aug 18 11:19  residence:   Aug 18 11:19
  retention:       active      retention-period: 0y, 0d, 0h, 6m
```

touch에 대한 -a 옵션이 파일 또는 디렉토리의 액세스 시간을 변경하는 데 사용됩니다. -t 옵션은 액세스 시간 필드에 사용될 시간을 지정합니다. 시간 인수에 대한 형식은 다음과 같고 [[CC]YY]MMDDhhmm[.SS]입니다.

- [CC] - 연도의 처음 두 자리 숫자
- [YY] - 연도의 두 번째 두 자리 숫자
- MM - 1년 중 월(01-12)
- DD - 해당 월의 날짜(01-31)
- hh - 시간(00-23)
- mm - 분(00-59)
- [SS] - 초(00-61)

CC, YY 및 SS 필드는 옵션입니다. CC 및 YY가 제공되지 않으면 기본값은 현재 연도입니다. 이러한 옵션에 대한 자세한 내용은 touch 매뉴얼 페이지를 참조하십시오.

보유 기간을 영구 보유로 설정하려면 액세스 시간을 설정 가능한 다음 최대값으로 설정하십시오. 203801182214.07.

## 파일 보유 기간 연장

코드 예 7-7은 touch를 사용하여 파일의 보유 기간을 연장하는 예제를 나타낸 것입니다.

코드 예 7-7                    touch를 사용한 파일의 보유 기간 연장

```
# s1s -D test
test:
  mode: -r-Sr--r--  links: 1  owner: root      group: root
  length:          0  admin id: 0  inode: 1029.1
  access:          Aug 18 11:35  modification: Aug 18 11:33
  changed:         Aug 18 11:33  retention-end: Aug 18 2005 11:35:00
  creation:        Aug 18 11:33  residence:     Aug 18 11:33
  retention:       over          retention-period: 0y, 0d, 0h, 2m
# touch -a -t200508181159 test
# s1s -D
test:
  mode: -r-Sr--r--  links: 1  owner: root      group: root
  length:          0  admin id: 0  inode: 1029.1
  access:          Aug 18 11:35  modification: Aug 18 11:33
  changed:         Aug 18 11:33  retention-end: Aug 18 2005 11:59
  creation:        Aug 18 11:33  residence:     Aug 18 11:33
  retention:       active        retention-period: 0y, 0d, 0h, 26m
```

이 예제에서 보유 기간은 2005년 8월 18일 오전 11:59분으로 연장되었으며 이는 WORM 트리거가 처음 적용된 시간부터 26분 후입니다.

---

주 - 보유 기간 연장을 위한 touch의 사용은 활성 WORM 모드와 관련이 없습니다.

---

## s1s를 사용한 WORM-FS 파일 보기

WORM 파일 속성을 보려면 s1s 명령을 사용하십시오. -D 옵션은 디렉토리가 WORM 가능한지 여부를 표시합니다. 파일에서 이 옵션을 사용하여 보유 기간이 시작한 시기, 종료할 시기, 현재 보유 상태 및 명령줄에 지정된 지속기간을 표시할 수 있습니다.

보유 기간의 시작은 파일의 modified time 필드에 저장됩니다. 보유 기간의 종료는 파일의 attribute time 필드에 저장됩니다. 이 시간은 달력 날짜로 표시됩니다. s1s 출력 결과의 추가 행은 보유 기간 상태 및 지속기간을 표시합니다.

코드 예 7-8은 `sls -D`가 파일의 보유 상태를 표시하는 방식의 예를 보여줍니다.

코드 예 7-8                    `sls`를 사용하여 파일의 보유 상태 찾기

```
sls -D test
test:
  mode: -r-Sr--r--  links: 1  owner: root group: root
  length: 5  admin id: 0  inode: 1027.1
  access: Aug 18 2005  modification: Aug 18 11:19
  changed: Aug 18 11:19  retention-end: Aug 18 2005 11:25
  creation: Aug 18 11:19  residence: Aug 18 11:19
  retention: active  retention-period: 0y, 0d, 0h, 6m
```

이 예제에서 보유 상태는 `retention: active` 지정에 의해 표시되는 대로 활성화되며 파일에 WORM 비트가 설정되었음을 의미합니다. 보유 기간은 2005년 8월 18일 11:19분에 시작했으며 2005년 8월 18일 11:25분에 종료합니다. 보유 기간은 0년, 0일, 0시, 6분으로 지정되었습니다.

## `sfind`를 사용한 WORM-FS 파일 찾기

특정 보유 기간을 갖는 파일을 검색하려면 `sfind` 유틸리티를 사용하십시오. 옵션에 대한 자세한 내용은 `sfind(1)` 매뉴얼 페이지를 참조하십시오. 다음 옵션을 사용할 수 있습니다.

- `-ractive` - 보유 기간이 활성화된 파일을 찾습니다.
- `-rover` - 보유 기간이 만료된 파일을 찾습니다.
- `-rafter date` - 보유 기간이 지정된 날짜 후에 종료되는 파일을 찾습니다. 날짜는 `YYYYMMDDHHmm`으로 지정되며, 여기서 `YYYY`는 연도, `MM`은 월, `DD`는 일, `HH`는 시간 및 `mm`은 분을 의미합니다.

코드 예 7-9는 보유 기간이 2004년 12월 24일 오후 3시에 만료되는 파일을 찾는 명령의 예입니다.

코드 예 7-9                    `sfind`를 사용하여 특정 날짜 후에 만료되는 모든 WORM 파일 찾기

```
# sfind -rafter 200412241500
```

- `-rremain time` - 최소한 지정된 시간이 남아 있는 보유 기간을 갖는 파일을 찾습니다. 시간은 `MyNdOhPm`으로 지정되며, 여기서 `M`, `N`, `O` 및 `P`는 음이 아닌 임의의 정수이며 `y`, `d`, `h` 및 `m`은 각각 연도, 일, 시간 및 분을 나타냅니다.

예를 들어 **코드 예 7-10**은 만료 전에 1년 10일 5시간 10분 이상이 남은 파일을 찾는 명령을 나타냅니다.

코드 예 7-10 `sfind`를 사용하여 지정된 것보다 시간이 남는 모든 WORM 파일 검색

```
# sfind -rremain 1y10d5h10m
```

- `-rlonger time` - 지정된 시간보다 긴 보유 기간을 갖는 파일을 찾습니다. 시간은 `MyNdOhPm`으로 지정되며, 여기서 `M`, `N`, `O` 및 `P`는 음이 아닌 임의의 정수이며 `y`, `d`, `h` 및 `m`은 각각 연도, 일, 시간 및 분을 나타냅니다.

**코드 예 7-11**은 10일 이상의 보유 기간을 갖는 파일을 찾는 명령의 예입니다.

코드 예 7-11 `sfind`를 사용하여 지정된 것보다 긴 보유 기간을 갖는 모든 WORM 파일 검색

```
# sfind -rlonger 10d
```

- `-rpermanent` - 보유 기간이 영구적인 파일을 찾습니다.

## 용량이 큰 파일 작업

매우 큰 용량의 파일을 조작할 때에는 시스템에서 사용 가능한 디스크 캐시의 크기에 주의해야 합니다. 디스크 캐시보다 큰 파일을 쓰려는 경우 사용하는 파일 시스템의 유형에 따라 동작이 다릅니다.

- Sun StorageTek QFS 파일 시스템을 사용하는 경우 시스템이 `ENOSPC` 오류를 반환합니다.
- SAM-QFS 파일 시스템을 사용하는 경우 요청을 처리할 만한 디스크 공간이 없기 때문에 프로그램이 존재하지 않는 공간을 대기하며 멈추게 됩니다.

SAM-QFS 환경에서 작업하고 응용 프로그램에서 디스크 캐시보다 큰 파일을 기록해야 할 경우에는 `segment(1)` 명령을 사용하여 파일을 세그먼트로 나눌 수 있습니다. `segment(1)` 명령에 대한 자세한 내용은 `segment(1)` 매뉴얼 페이지를 참조하거나 Sun StorageTek Storage Archive Manager 아카이브 구성 및 관리 안내서를 참조하십시오.

---

## 다중판독기 파일 시스템 구성

다중 판독기 파일 시스템은 단일 작성기 호스트 및 다중 판독기 호스트로 구성됩니다. 다중 판독기 파일 시스템을 활성화하는 `writer` 및 `reader` 마운트 옵션은 Sun StorageTek QFS 파일 시스템에만 호환됩니다. 마운트 옵션은 이 절 및 `mount_samfs(1M)` 매뉴얼 페이지에 설명되어 있습니다.

`mount(1M)` 명령에서 `-o writer` 옵션을 지정하여 단일 작성기 호스트에 다중 판독기 파일 시스템을 마운트할 수 있습니다. `writer` 마운트 옵션으로 마운트된 호스트 시스템은 파일 시스템에 쓰기가 허용된 유일한 호스트 시스템입니다. `writer` 호스트 시스템은 파일 시스템을 업데이트합니다. 다중 판독기 파일 시스템에서 오직 하나의 호스트만 `writer` 마운트 옵션으로 마운트되어 활성화된 파일 시스템을 포함할 수 있습니다. `-o writer`가 지정되면 디렉토리는 각 변경 시점에 디스크에 쓰여지고 파일은 닫는 시점에 디스크에 쓰여집니다.



---

**주의** - 한 시점에 둘 이상의 작성기 호스트가 마운트된 파일 시스템이 있는 경우 다중 판독기 파일 시스템이 손상될 수 있습니다. 이러한 상황이 발생하지 않도록 하는 것이 사이트 관리자의 역할입니다.

---

`mount(1M)` 명령에 `-o reader` 옵션을 지정하여 하나 이상의 판독기 호스트에 다중 판독기 파일 시스템을 마운트할 수 있습니다. `reader` 옵션으로 마운트된 다중 판독기 파일 시스템을 가질 수 있는 호스트 시스템의 수에는 제한이 없습니다.

다중 판독기 파일 시스템과 Sun StorageTek QFS 공유 파일 시스템의 가장 큰 차이점은 다중 판독기 호스트는 디스크에서 메타 데이터를 읽고, Sun StorageTek QFS 공유 파일 시스템의 클라이언트 호스트는 네트워크를 통해 메타 데이터를 읽는다는 점입니다. Sun StorageTek QFS 공유 파일 시스템은 다중 판독기 호스트를 지원합니다. 이 구성에서 다중 공유 호스트는 다중 판독기 호스트가 내용을 분산시키는 동안 내용을 추가할 수 있습니다.

---

**주** - 파일 시스템을 Sun StorageTek QFS 공유 파일 시스템으로 마운트하려는 경우 호스트에 `writer` 옵션을 지정할 수 없습니다. 그러나 `reader` 옵션은 지정할 수 있습니다.

Sun StorageTek QFS 공유 파일 시스템 클라이언트 호스트를 읽기 전용 호스트로 만들 경우 `reader` 마운트 옵션을 사용하여 해당 호스트에 Sun StorageTek QFS 공유 파일 시스템을 마운트합니다. 추가로 Sun StorageTek QFS 공유 파일 시스템에서 `reader` 옵션을 사용하는 경우 `sync_meta` 마운트 옵션을 1로 설정합니다. Sun StorageTek QFS 공유 파일 시스템에 대한 자세한 정보는 67페이지의 "[Sun StorageTek QFS 공유 파일 시스템 구성](#)"을 참조하십시오. 마운트 옵션에 대한 정보는 `mount_samfs(1M)` 매뉴얼 페이지를 참조하십시오.

---

다중 관독기 파일 시스템의 모든 관독기는 `ma` 장치를 설명하는 장치 정의에 액세스할 수 있어야 합니다. 기본 메타 데이터 서버 호스트에 있는 `mcf` 파일의 행을 대체 메타 데이터 서버의 `mcf` 파일에 복사합니다. 행을 복사한 후 구성에 따라 디스크 분할 영역이 모든 호스트에 동일한 방법으로 나타나지 않을 수도 있기 때문에 디스크 제어기에 대한 정보를 업데이트해야 합니다.

다중 관독기 파일 시스템 환경에서 Sun StorageTek QFS 소프트웨어는 동일한 파일 시스템에 액세스하는 모든 서버가 언제나 현재 환경에 액세스할 수 있도록 합니다. 작성기가 파일을 닫으면 Sun StorageTek QFS 파일 시스템은 해당 파일에 대한 모든 정보를 즉시 디스크에 씁니다. `reader` 호스트는 작성기에 의해 파일이 닫힌 후 해당 파일에 액세스할 수 있습니다. `refresh_at_eof` 마운트 옵션을 지정하여 다중 관독기 파일 시스템의 호스트 시스템이 파일 시스템과 동기화되도록 하는 데 도움을 줄 수 있습니다.

기본적으로 `reader` 호스트에 있는 파일에 대한 메타 데이터 정보는 파일에 액세스할 때마다 무효화되고 새로 고쳐집니다. 데이터가 변경되면 무효화됩니다. `cat(1)`, `ls(1)`, `touch(1)`, `open(2)`을 통해서든, 다른 방법을 통해서든 여기에는 모든 유형의 액세스가 포함됩니다. 이 즉시 새로고침 간격은 새로고침이 수행될 때 올바른 데이터를 보장하지만 성능에 영향을 줄 수 있습니다. 사이트 기본 설정에 따라 `mount(1M)` 명령의 `-o invalid=n` 옵션을 사용하여 0초와 60초 사이로 새로 고침 간격을 지정할 수 있습니다. 새로 고침 간격을 작은 값으로 설정하면 Sun StorageTek QFS 파일 시스템이 디렉토리 및 기타 메타 데이터 정보를 마지막 새로 고침 후 `n` 초에 읽게 됩니다. 새로 고침을 자주 할수록 시스템에 대한 오버헤드가 많아지지만 `n`이 0이 아닌 경우 스테일 정보가 존재할 수 있습니다.



---

주의 - `reader` 호스트에서 읽을 수 있도록 파일이 열려 있는 경우, 해당 파일이 작성기에 의해 제거되거나 잘리는 것을 방지할 수 없습니다. 응용 프로그램 잠금과 같은 또 다른 메커니즘을 사용하여 부주의한 쓰기 작업으로부터 읽기를 보호해야 합니다.

---

---

# 이기종 컴퓨팅 환경에서 SAN-QFS 파일 시스템 사용

SAN-QFS 파일 시스템에서는 여러 호스트가 최대의 디스크 속도로 Sun StorageTek QFS 시스템에 저장된 데이터에 액세스할 수 있습니다. 이 기능은 특히 유형이 이기종 환경에서 고성능의 공유 디스크 액세스가 요구되는 데이터베이스, 데이터 스트리밍, 웹 페이지 서비스 또는 기타 응용 프로그램에서 유용하게 사용될 수 있습니다.

SAN-QFS 파일 시스템은 SAN(Storage Area Network)에서 광섬유로 연결된 장치와 함께 사용될 수 있습니다. SAN-QFS 파일 시스템은 Sun StorageTek QFS 소프트웨어 및 Tivoli SANergy 파일 공유 소프트웨어 같은 소프트웨어를 통해 데이터에 고속으로 액세스할 수 있습니다. SAN-QFS 파일 시스템을 사용하려면 SANergy(2.2.4 이상) 및 Sun StorageTek QFS 소프트웨어가 둘 다 있어야 합니다. 지원되는 Sun StorageTek QFS 및 SANergy 소프트웨어의 레벨에 대한 자세한 내용은 해당 Sun 영업 담당자에게 문의하십시오.

---

주 - Solaris OS 및 지원되는 Linux 운영 체제를 갖춘 환경에서 Solaris 호스트에 SAN-QFS 파일 시스템이 아닌 Sun StorageTek QFS 공유 파일 시스템을 사용합니다.

Sun StorageTek QFS 공유 파일 시스템에 대한 자세한 내용은 [67페이지의 "Sun StorageTek QFS 공유 파일 시스템 구성"](#)을 참조하십시오. SAN-QFS 파일 시스템과 Sun StorageTek QFS 공유 파일 시스템의 비교에 대해서는 [212페이지의 "SAN-QFS 공유 파일 시스템 및 Sun StorageTek QFS 공유 파일 시스템 비교"](#)를 참조하십시오.

---

**그림 7-1**은 Sun StorageTek QFS 소프트웨어와 SANergy 소프트웨어를 모두 사용하는 SAN-QFS 파일 시스템을 보여주며 클라이언트와 메타 데이터 제어기(MDC) 시스템이 LAN 전체에서 메타 데이터를 관리하는 것을 보여줍니다. 클라이언트가 저장 장치와의 I/O를 직접 수행합니다.

Solaris OS만을 실행 중인 클라이언트는 Sun StorageTek QFS 소프트웨어를 호스트하며 Solaris 이외의 OS를 실행 중인 모든 이기종 클라이언트는 SANergy 소프트웨어와 NFS 소프트웨어를 호스트하고 있습니다. SAN-QFS 파일 시스템의 메타 데이터 서버는 Sun StorageTek QFS 및 SANergy 소프트웨어를 둘 다 호스트합니다. 이 서버는 파일 시스템에 대한 메타 데이터 서버로서 뿐만 아니라 SANergy MDC로서 작동합니다.

---

주 - SANergy 소프트웨어는 x64 하드웨어 플랫폼에서 지원되지 않습니다.

---

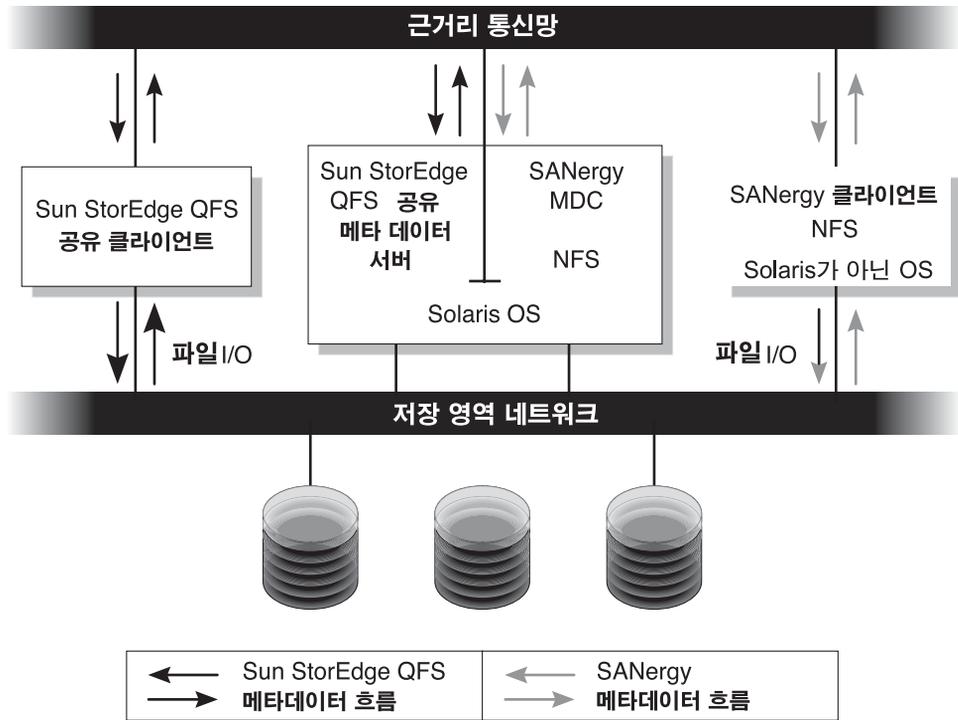


그림 7-1 Sun StorageTek QFS 소프트웨어 및 SANergy 소프트웨어를 사용하는 SAN-QFS 파일 시스템

이 절의 나머지 부분에서는 SAN-QFS 파일 시스템의 기타 측면에 대해 설명합니다.

- 206페이지의 "시작하기 전에"
- 206페이지의 "SAN-QFS 파일 시스템 활성화"
- 209페이지의 "SAN-QFS 파일 시스템 마운트 해제"
- 211페이지의 "문제 해결: SANergy 파일 보류를 갖는 SAN-QFS 파일 시스템 마운트 해제"
- 211페이지의 "SAN-QFS 파일 시스템의 블록 할당량"
- 211페이지의 "SAN-QFS 파일 시스템의 파일 데이터 및 파일 속성"
- 212페이지의 "samgrowfs(1M)을 사용한 SAN-QFS 파일 시스템 확장"
- 212페이지의 "SAN-QFS 공유 파일 시스템 및 Sun StorageTek QFS 공유 파일 시스템 비교"

## 시작하기 전에

SAN-QFS 파일 시스템을 사용하기 전에 다음 구성 고려 사항을 기억하고 그에 따라서 계획합니다.

- SAN-QFS 파일 시스템에서 사용하도록 구성되는 디스크는 볼륨 관리자의 제어를 받을 수 없습니다.
- SAN-QFS 환경에서 Sun StorageTek QFS 메타 데이터 서버를 활성화하거나 위치를 변경하려면 새 메타 데이터 서버 시스템을 SANergy 메타 데이터 제어기로 구성해야 합니다.
- SAN-QFS 파일 시스템은 세그먼트된 파일을 인식하지 않습니다. 따라서 SAN-QFS 환경에서 세그먼트된 파일을 사용하는 경우 예기치 않은 동작이 발생할 수 있습니다.
- Sun StorageTek QFS mcf 파일에서 ms 또는 md 장치로 분류된 장치는 SAN-QFS 파일 시스템에서 지원되지 않습니다.

---

주 - 이 안내서는 비 Solaris 클라이언트가 파일 시스템 공유를 위해 SANergy 소프트웨어와 NFS 소프트웨어를 호스트하고 있다고 가정합니다. 이 안내서의 텍스트와 예제가 이 구성을 반영합니다. 비 Solaris 클라이언트가 NFS 소프트웨어 대신 Samba 소프트웨어를 호스트하는 경우 Samba 문서를 참조하십시오.

---

## SAN-QFS 파일 시스템 활성화

다음 절차는 SAN-QFS 파일 시스템을 활성화하는 방법에 대해 설명합니다. 다음 절차를 표시된 순서대로 수행하십시오.

- 206페이지의 "[메타 데이터 제어기에서 SAN-QFS 파일 시스템 활성화](#)"
- 207페이지의 "[클라이언트에서 SAN-QFS 파일 시스템 활성화](#)"
- 208페이지의 "[클라이언트에 SANergy 소프트웨어 설치](#)"

### ▼ 메타 데이터 제어기에서 SAN-QFS 파일 시스템 활성화

SAN-QFS 파일 시스템을 사용할 때 사용자 환경에 있는 한 호스트 시스템이 SANergy 메타 데이터 제어기(MDC)의 역할을 수행합니다. 이것은 Sun StorageTek QFS 파일 시스템이 상주하는 호스트 시스템입니다.

1. Sun StorageTek QFS 파일 시스템이 상주하는 호스트에 로그인하고 슈퍼유저가 됩니다.
2. Sun StorageTek QFS 파일 시스템이 테스트되고 정확하게 동작하는지 확인합니다.
3. SANergy 소프트웨어를 설치하고 구성합니다.

지침에 대해서는 SANergy 문서를 참조하십시오.

4. `pkginfo(1)` 명령을 사용하여 **SANergy** 소프트웨어 릴리스 레벨을 확인합니다.

```
# pkginfo -l SANergy
```

5. 파일 시스템이 마운트되었는지 확인합니다.

`mount(1M)` 명령을 사용하여 마운트를 확인하거나 파일 시스템을 마운트합니다.

6. 다음 형식으로 `share(1M)` 명령을 사용하여 클라이언트 호스트에 대한 **NFS** 액세스를 활성화합니다.

```
MDC# share -F nfs -d qfs-file-system-name /mount-point
```

*qfs-file-system-name*의 경우 Sun StorageTek QFS 파일 시스템의 이름을 `qfs1`과 같이 지정합니다. `share(1M)` 명령에 대한 자세한 내용은 `share(1M)` 또는 `share_nfs(1M)` 매뉴얼 페이지를 참조하십시오.

*mount-point*에 대해 *qfs-file-system-name*의 마운트 지점을 지정합니다.

7. **Microsoft Windows** 클라이언트에 연결하려는 경우 **NFS**보다는 **Samba**를 구성하여 보안 및 이름공간 기능을 제공합니다.

이렇게 하려면 `/etc/init.d/sanergy` 파일에 `SANERGY_SMBPATH` 환경 변수를 추가하고 Samba 구성 파일의 위치를 가리키도록 합니다. 예를 들어 Samba 구성 파일의 이름이 `/etc/swf/smb.conf`인 경우 `/etc/init.d/sanergy` 파일의 시작 부분에 다음 행을 추가해야 합니다.

```
SANERGY_SMBPATH=/etc/swf/smb.conf
export SANERGY_SMBPATH
```

8. (선택 사항) **MDC**에서 파일 시스템 표(`/etc/dfs/dfstab`)를 편집하여 부트 시 액세스를 활성화합니다.

부트 시 이 액세스 기능을 자동으로 활성화하려는 경우에 이 단계를 수행하십시오.

## ▼ 클라이언트에서 SAN-QFS 파일 시스템 활성화

**MDC**에서 파일 시스템을 활성화하면 이를 클라이언트 호스트에서 활성화할 준비가 된 것입니다. **SAN-QFS** 파일 시스템은 **IRIX**, **Microsoft Windows**, **AIX** 및 **Linux** 호스트를 포함하여 여러 가지 클라이언트 호스트를 지원합니다. 지원되는 특정 클라이언트에 대한 자세한 내용은 Sun 영업 담당자에게 문의하십시오.

모든 클라이언트는 서로 다른 동작 특성을 갖습니다. 이 절차는 일반적인 용어를 사용하여 클라이언트에서 SAN-QFS 파일 시스템을 활성화하기 위해 취해야 하는 작업을 설명합니다. 클라이언트에 특정한 정보에 대해서는 클라이언트 호스트와 함께 제공되는 문서를 참조하십시오.

1. 각 클라이언트 호스트에 로그인합니다.
2. 각 클라이언트에서 파일 시스템 기본값 테이블을 편집하고 파일 시스템을 추가합니다. 예를 들어 Solaris OS에서 각 클라이언트의 `/etc/vfstab` 파일을 편집하고 다음과 같이 Sun StorageTek QFS 파일 시스템의 이름을 추가합니다.

```
server:/qfs1 - /qfs1 nfs - yes noac,hard,intr,timeo=1000
```

다른 운영 체제 플랫폼에서는 파일 시스템 기본값 테이블이 `/etc/vfstab`가 아닌 다른 파일에 상주할 수 있습니다. 예를 들어 Linux 시스템에서는 이 파일이 `/etc/fstab`입니다.

`/etc/vfstab` 파일 편집에 대한 자세한 내용은 Sun StorageTek QFS 설치 및 업그레이드 안내서를 참조하십시오. 필수 또는 제안된 NFS 클라이언트 마운트 옵션에 대한 자세한 내용은 SANergy 문서를 참조하십시오.

## ▼ 클라이언트에 SANergy 소프트웨어 설치

클라이언트 호스트에서 파일 시스템을 활성화한 후 클라이언트에 SANergy 소프트웨어를 설치할 수 있습니다. 다음 절차는 일반 용어로 SANergy 설치 과정을 설명합니다.

1. **SANergy** 소프트웨어를 설치하고 구성합니다.  
지침에 대해서는 SANergy 문서를 참조하십시오.
2. `mount(1M)` 명령을 사용하여 파일 시스템을 **NFS** 마운트합니다.  
예를 들어, 다음과 같습니다.

```
# mount host:/mount-point/ local-mount-point
```

`host`에 대해 MDC를 지정하십시오.

`mount-point`에 대해 MDC에서 Sun StorageTek QFS 파일 시스템의 마운트 지점을 지정합니다.

`local-mount-point`에 대해 SANergy 클라이언트에서 마운트 지점을 지정합니다.

### 3. SANergy fuse 명령을 사용하여 소프트웨어를 결합합니다.

```
# fuse |mount-point
```

*mount-point*에 대해 SANergy 클라이언트에서 마운트 지점을 지정합니다.

## SAN-QFS 파일 시스템 마운트 해제

다음 절차는 SANergy 소프트웨어를 사용 중인 SAN-QFS 파일 시스템을 마운트 해제하는 방법에 대해 설명합니다. 다음 절차를 표시된 순서대로 수행하십시오.

- 209페이지의 "SANergy 클라이언트에서 SAN-QFS 파일 시스템 마운트 해제"
- 210페이지의 "메타 데이터 제어기에서 SAN-QFS 파일 시스템 마운트 해제"
- 210페이지의 "Sun StorageTek QFS 클라이언트에서 SAN-QFS 파일 시스템 마운트 해제"
- 210페이지의 "Sun StorageTek QFS 서버에서 SAN-QFS 파일 시스템 마운트 해제"

### ▼ SANergy 클라이언트에서 SAN-QFS 파일 시스템 마운트 해제

SAN-QFS 파일 시스템을 마운트 해제하려는 각 클라이언트 호스트에 대해 다음 단계를 수행하십시오.

1. 클라이언트 호스트에 로그인하고 슈퍼유저가 됩니다.
2. SANergy unfuse 명령을 사용하여 소프트웨어에서 파일 시스템을 분리합니다.

```
# unfuse |mount-point
```

*mount-point*에 대해 SANergy 클라이언트에서 마운트 지점을 지정합니다.

3. `umount(1M)` 명령을 사용하여 NFS에서 파일 시스템을 마운트 해제합니다.

```
# umount host : /mount-point / local-mount-point
```

*host*에 대해 MDC를 지정하십시오.

*mount-point*에 대해 MDC에서 Sun StorageTek QFS 파일 시스템의 마운트 지점을 지정합니다.

*local-mount-point*에 대해 SANergy 클라이언트에서 마운트 지점을 지정합니다.

## ▼ 메타 데이터 제어기에서 SAN-QFS 파일 시스템 마운트 해제

1. **MDC** 시스템에 로그인하고 슈퍼유저가 됩니다.
2. **unshare(1M)** 명령을 사용하여 클라이언트 호스트에 대한 **NFS** 액세스를 비활성화합니다.

```
MDC# unshare qfs-file-system-name /mount-point
```

*qfs-file-system-name*에 대해 Sun StorageTek QFS 파일 시스템의 이름을 *qfs1*과 같이 지정합니다. **unshare(1M)** 명령에 대한 자세한 내용은 **unshare(1M)** 매뉴얼 페이지를 참조하십시오.

*mount-point*에 대해 *qfs-file-system-name*의 마운트 지점을 지정합니다.

## ▼ Sun StorageTek QFS 클라이언트에서 SAN-QFS 파일 시스템 마운트 해제

각 관련 클라이언트 호스트에 대해 다음 단계를 따르십시오.

1. **Sun StorageTek QFS** 클라이언트 호스트에 로그인하고 슈퍼유저가 됩니다.
2. **umount(1M)** 명령을 사용하여 파일 시스템을 마운트 해제합니다.  
예를 들어, 다음과 같습니다.

```
# umount /qfs1
```

## ▼ Sun StorageTek QFS 서버에서 SAN-QFS 파일 시스템 마운트 해제

1. **Sun StorageTek QFS** 파일 시스템이 상주하는 호스트 시스템에 로그인하고 슈퍼유저가 됩니다.
2. **umount(1M)** 명령을 사용하여 파일 시스템을 마운트 해제합니다.

## 문제 해결: SANergy 파일 보류를 갖는 SAN-QFS 파일 시스템 마운트 해제

SANergy 소프트웨어는 Sun StorageTek QFS 파일에 보류를 실행하여 가속화된 액세스를 위해 임시로 예약합니다. 보류가 적용되고 있을 때 SANergy가 충돌하는 경우 파일 시스템을 마운트 해제할 수 없게 됩니다. SAN-QFS 파일 시스템을 마운트 해제할 수 없는 경우 /var/adm/messages 파일을 조사하여 미해결 SANergy 보류를 설명하는 콘솔 메시지를 찾으십시오.

가능한 경우에는 언제나 SANergy 파일 공유 기능이 이 보류를 정리할 수 있도록 하지만, 긴급한 상황이나 SANergy 파일 공유 시스템 장애의 경우에는 다음 절차를 사용하여 재부트하지 않도록 합니다.

### ▼ SANergy 파일 보류가 존재하는 경우 파일 시스템 마운트 해제

1. `unshare(1M)` 명령을 사용하여 **NFS** 액세스를 비활성화합니다.
2. `samunhold(1M)` 명령을 사용하여 **SANergy** 파일 시스템 보류를 해제합니다.  
이 명령에 대한 자세한 내용은 `samunhold(1M)` 매뉴얼 페이지를 참조하십시오.
3. `umount(1M)` 명령을 사용하여 파일 시스템을 마운트 해제합니다.

## SAN-QFS 파일 시스템의 블록 할당량

SANergy 소프트웨어는 블록 할당량을 강제로 실행하지 않습니다. 그러므로 SANergy 소프트웨어로 파일을 쓸 때 블록 할당량을 초과할 수 있습니다. 할당량에 대한 자세한 내용을 보려면 [98페이지의 "할당량 활성화"](#)를 참조하십시오.

## SAN-QFS 파일 시스템의 파일 데이터 및 파일 속성

SANergy 소프트웨어는 메타 데이터 작업을 위해 NFS 소프트웨어를 사용하며, 이는 NFS close-to-open 일관성 모델이 파일 데이터 및 속성에 사용됨을 의미합니다. SANergy 클라이언트 사이의 파일 데이터와 속성은 열려있는 파일에 대해 POSIX 일관성 모델을 지원하지 않습니다.

## samgrowfs(1M)을 사용한 SAN-QFS 파일 시스템 확장

samgrowfs(1M) 명령을 사용하여 SAN-QFS 파일 시스템의 크기를 늘릴 수 있습니다. 이 작업을 수행하려면 63페이지의 "파일 시스템에 디스크 캐시 추가"에 설명된 절차를 따르십시오.



---

주의 - 이 절차를 수행할 때 mcf 파일의 행에 나타난 장치 순서는 파일 시스템 슈퍼블록에 나열된 장치의 순서와 일치해야 합니다.

---

samgrowfs(1M) 명령을 실행하면 이미 mcf 파일에 있는 장치는 슈퍼블록의 위치를 유지합니다. 새로운 장치는 해당하는 순서에 따라 하위 항목에 쓰여지게 됩니다.

새로운 순서가 슈퍼블록에 있는 순서와 일치하지 않으면 SAN-QFS 파일 시스템은 확장될 수 없습니다.

## SAN-QFS 공유 파일 시스템 및 Sun StorageTek QFS 공유 파일 시스템 비교

SAN-QFS 공유 파일 시스템과 Sun StorageTek QFS 공유 파일 시스템의 유사성은 다음과 같습니다.

- 파일을 스테이지할 수 있습니다.
- 기본 파일 시스템 호스트가 데이터 쓰기를 담당하지 않는 데이터 캡처 환경에서 유용합니다.
- 파일 쓰기에 대한 경합이 있는 환경에서 편리합니다.

표 7-3에서는 파일 시스템 간의 차이점을 설명합니다.

표 7-3 SAN-QFS 공유 파일 시스템 및 Sun StorageTek QFS 공유 파일 시스템

SAN-QFS 파일 시스템	Sun StorageTek QFS 공유 파일 시스템
메타 데이터에 대해 NFS 프로토콜을 사용합니다.	자연 메타 데이터를 사용합니다.
다른 유형의 컴퓨팅 환경(즉, 모든 호스트가 Sun 시스템이 아닌 경우)에 적합합니다.	동종 Solaris OS 환경에서 선호됩니다.
여러 이기종 호스트가 데이터를 쓸 수 있어야 하는 환경에 유용합니다.	여러 호스트가 동시에 동일한 파일에 써야 하는 환경에 적합합니다.

---

# I/O 유형 이해

Sun StorageTek QFS 파일 시스템은 페이지된 I/O, 직접 I/O 및 I/O 유형 간 전환을 지원합니다. 다음 절은 이러한 I/O 유형을 설명합니다.

## 페이지된 I/O

페이지된 I/O가 사용될 경우, 사용자 데이터는 가상 메모리 페이지에 캐시되며 커널은 데이터를 디스크에 기록합니다. 표준 Solaris OS 인터페이스는 페이지된 I/O를 관리합니다. 페이지된 I/O(버퍼 또는 캐시된 I/O라고도 함)가 기본적으로 선택됩니다.

## 직접 I/O

직접 I/O는 데이터가 사용자의 버퍼와 디스크 사이에 직접 전송되는 프로세스입니다. 따라서 시스템에서 소요되는 시간이 줄어듭니다. 높은 성능을 위해서는 대형 블록이 정렬된 연속적인 I/O에 대해서만 직접 I/O를 지정하십시오.

`setfa(1)` 명령 및 `sam_setfa(3)` 라이브러리 루틴에는 파일 또는 디렉토리에 대해 직접 I/O 속성을 설정하는 `-D` 옵션이 있습니다. 디렉토리에 적용된 경우 해당 디렉토리에 생성된 파일 및 디렉토리는 직접 I/O 속성을 방지합니다. `-D` 옵션이 설정되면 파일은 직접 I/O를 사용합니다.

Solaris OS `directio(3C)` 함수를 호출하여 파일에 대해 직접 I/O를 선택할 수도 있습니다. 함수 호출을 사용하여 직접 I/O를 활성화할 경우 해당 설정은 파일이 활성인 동안에만 지속됩니다.

파일 시스템 기반의 직접 I/O를 활성화하려면 다음 중 하나를 수행하십시오.

- `mount(1M)` 명령에 `-o forcedirectio` 옵션을 지정합니다.
- `/etc/vfstab` 파일의 마운트 옵션 열에 `forcedirectio` 키워드를 입력하거나 `samfs.cmd` 파일에서 명령으로 사용하십시오.

자세한 내용은 `setfa(1)`, `sam_setfa(3)`, `directio(3C)`, `samfs.cmd(4)` 및 `mount_samfs(1M)` 매뉴얼 페이지를 참조하십시오.

## I/O 전환

기본적으로 페이지된 I/O가 수행되고 I/O 전환은 비활성화됩니다. 그러나 Sun StorageTek QFS 파일 시스템은 시스템이 자동으로 직접 I/O로 전환하기 전에 사이트에서 정의된 양의 페이지된 I/O가 발생하는 프로세스인 자동 I/O 전환을 지원합니다.

I/O 전환은 대량 I/O 작업에서 페이지 캐시 사용량을 줄여야 합니다. I/O 전환을 활성화하려면 `samu(1M)`를 사용하거나 `dio_wr_consec` 및 `dio_rd_consec` 매개변수를 `samfs.cmd` 파일에서 명령으로 또는 `mount(1M)` 명령의 옵션으로 사용하십시오.

이러한 옵션에 대한 자세한 내용은 `mount_samfs(1M)` 또는 `samfs.cmd(4)` 매뉴얼 페이지를 참조하십시오.

---

## 대형 파일에 대한 파일 전송 성능 향상

Sun StorageTek QFS 파일 시스템을 조정하여 다양한 파일 크기를 처리하도록 할 수 있습니다. 파일 시스템 설정을 활성화하여 대형 파일에 대한 디스크 파일 전송 성능을 향상시킬 수 있습니다.

---

주 - Sun은 프로덕션 환경이 아닌 다른 환경에서 성능 조정을 테스트할 것을 권장합니다. 이러한 변수를 잘못 조정하면 전체 시스템에 예기치 않은 결과를 가져올 수 있습니다.

사이트가 Sun™ Enterprise Services(SES) 지원 계약을 체결한 경우 성능 조정 매개변수를 변경하려면 SES에 알려십시오.

---

### ▼ 파일의 전송 성능 향상

1. 장치 읽기/쓰기 명령을 최대로 설정합니다.

Solaris `/etc/system` 파일의 `maxphys` 매개변수는 장치 드라이버가 어느 한 시점에 읽거나 쓰는 최대 바이트 수를 제어합니다. `maxphys` 매개변수의 기본값은 사용 중인 Sun Solaris OS의 레벨에 따라 다를 수 있지만, 대개 128KB 정도입니다.

`/etc/system`에 다음 행을 추가하여 `maxphys`를 8KB로 설정합니다.

```
set maxphys = 0x800000
```

---

주 - `maxphys` 값은 반드시 2의 제곱으로 설정되어야 합니다.

---

## 2. SCSI 디스크의 최대 전송 매개변수를 설정합니다.

sd 드라이버는 /kernel/drv/sd.conf 파일에서 sd\_max\_xfer\_size 정의를 찾아 특정 파일에 대한 대량 전송을 활성화합니다. 이 정의가 존재하지 않으면 드라이버는 sd 장치 드라이버 정의 sd\_max\_xfer\_size에 정의된 값을 사용하며, 이 값은 1024 x 1024바이트입니다.

대량 전송을 활성화 및 사용하려면 /kernel/drv/sd.conf 파일 끝에 다음 행을 추가하십시오.

```
sd_max_xfer_size=0x800000;
```

## 3. 광섬유 디스크의 최대 전송 매개변수를 설정합니다.

ssd 드라이버는 /kernel/drv/ssd.conf 파일에서 ssd\_max\_xfer\_size 정의를 찾아 특정 파일에 대한 대량 전송을 가능하게 합니다. 이 정의가 존재하지 않으면 드라이버는 ssd 장치 드라이버 정의 ssd\_max\_xfer\_size에 정의된 값을 사용하며, 이 값은 1024 x 1024바이트입니다.

/kernel/drv/ssd.conf 파일 끝에 다음 행을 추가하십시오.

```
ssd_max_xfer_size=0x800000;
```

---

주 - Solaris 10 x86 플랫폼의 경우 /kernel/drv/sd.conf 파일에서 변경이 이루어집니다. 최대 전송 크기가 8메가바이트인 경우 다음 행이 추가됩니다.

```
sd_max_xfer_size=0x800000
```

---

## 4. 시스템을 재부트합니다.

## 5. writebehind 매개변수를 설정합니다.

이 단계는 페이지된 I/O에만 영향을 미칩니다.

writebehind 매개변수는 Sun StorageTek QFS 파일 시스템에서 페이지된 I/O를 수행할 때 파일 시스템에서 뒤에 쓰여지는 바이트 수를 지정합니다. writebehind 값이 RAID의 읽기-수정-쓰기 값에 대한 배수에 일치하면 성능이 향상될 수 있습니다.

이 매개변수는 KB 단위로 지정되고 8KB 배수로 잘립니다. 설정된 경우, 이 매개변수는 직접 I/O가 수행될 때 무시됩니다. 기본 writebehind 값은 512KB입니다. 이 값은 대형 블록의 연속 I/O에 적합합니다.

writebehind 크기를 하드웨어 및 소프트웨어 RAID 5 모두에 대해 RAID-5 스트라이프 크기의 배수로 설정합니다. RAID-5 스트라이프 크기는 구성된 스트라이프 너비를 곱한 데이터 디스크의 수입니다.

예를 들어 스트라이프 너비가 16KB인 3개의 데이터 디스크와 1개의 패리티 디스크 (3+1)로 RAID-5 장치를 구성했다고 가정합니다. 읽기-수정-쓰기 RAID-5 패리티 생성 부하를 막으려면 writebehind 값은 48KB, 96KB 또는 기타 배수여야 합니다.

또한, Sun StorageTek QFS 파일 시스템의 경우 DAU(sammkfs(1M) -a 명령)는 RAID-5 스트라이프 크기의 배수여야 합니다. 이러한 할당은 블록이 연속되도록 합니다. writebehind 크기를 재설정 한 후 시스템 성능을 테스트해야 합니다. 다음은 디스크 쓰기 시간을 테스트하는 예제입니다.

```
# timex dd if=/dev/zero of=/sam/myfile bs=256k count=2048
```

writebehind 매개변수를 마운트 옵션, samfs.cmd 파일 내부, /etc/vfstab 파일 내부 또는 samu(1M) 유틸리티 내의 명령에서 설정할 수 있습니다. 마운트 옵션에서 이를 활성화하는 방법에 대한 자세한 내용은 mount\_samfs(1M) 매뉴얼 페이지에서 -o writebehind=*n* 옵션을 참조하십시오. samfs.cmd 파일에서 활성화하는 자세한 내용은 samfs.cmd(4) 매뉴얼 페이지를 참조하십시오. samu(1M) 내에서 활성화하는 자세한 내용은 samu(1M) 매뉴얼 페이지를 참조하십시오.

## 6. readahead 매개변수를 설정합니다.

이 단계는 페이지된 I/O에만 영향을 미칩니다.

readahead 매개변수는 Sun StorageTek QFS 파일 시스템에서 페이지된 I/O가 수행될 때 파일 시스템에서 미리 읽는 바이트 수를 지정합니다. 이 매개변수는 KB 단위로 지정되고 8KB 배수로 잘립니다. 설정된 경우, 이 매개변수는 직접 I/O가 수행될 때 무시됩니다.

readahead 매개변수의 크기를 늘리면 대형 파일 전송의 성능은 높아지지만 한계가 있습니다. readahead 크기를 재설정 한 후, 전송 속도에 더 이상의 향상이 없을 때까지 시스템 성능을 테스트해야 합니다. 다음은 디스크 읽기 시간을 테스트하는 예제입니다.

```
# timex dd if=/sam/myfile of=/dev/null bs=256k
```

해당 환경에 대해 여러 가지 readahead 값을 테스트해야 합니다. readahead 매개변수는 페이지된 I/O에 대한 I/O 성능을 향상시키지만 성능에 그리 큰 영향을 주지 않는 크기로 설정되어야 합니다. readahead 값을 설정할 때에는 메모리 크기 및 동시 스트림의 수를 고려하는 것도 중요합니다. 스트림의 수를 곱한 readahead 값을 메모리보다 큰 값으로 설정하면 페이지 스래싱이 발생할 수 있습니다.

기본 readahead 값은 1024KB입니다. 이 값은 대형 블록의 연속 I/O에 적합합니다. 소형 블록, 변칙 I/O 응용 프로그램의 경우, readahead를 일반 요청 크기로 설정합니다. 데이터베이스 응용 프로그램에서는 고유의 미리 읽기를 수행하므로 이러한 응용 프로그램의 경우에는 readahead를 0으로 설정하십시오.

readahead 설정은 마운트 옵션, samfs.cmd 파일, /etc/vfstab 파일 또는 samu(1M) 유틸리티 내의 명령에서 활성화할 수 있습니다. 마운트 옵션에서 이를 활성화하는 방법에 대한 자세한 내용은 mount\_samfs(1M) 매뉴얼 페이지에서 -o readahead=*n* 옵션을 참조하십시오. samfs.cmd 파일에서 이 설정을 활성화하는 방법에 대한 자세한 내용은 samfs.cmd(4) 매뉴얼 페이지를 참조하십시오. samu(1M) 내에서 이 설정을 활성화하는 방법에 대한 자세한 내용은 samu(1M) 매뉴얼 페이지를 참조하십시오.

## 7. 스트라이프 너비를 설정합니다.

mount(1M) 명령의 `-o stripe=n` 옵션은 파일 시스템에 대한 스트라이프 너비를 지정합니다. 스트라이프 너비는 디스크 할당 단위(DAU) 크기를 기준으로 합니다.  $n$  인수는  $n \times$  DAU 바이트가 하나의 장치에 쓰여진 후에 쓰기가 다음 장치로 전환되도록 지정합니다. DAU 크기는 파일 시스템이 `sammkfs(1M) -a` 명령으로 초기화될 때 설정됩니다.

`-o stripe=0`이 설정되면 파일은 라운드 로빈 할당 방식을 사용하여 파일 시스템 장치에 할당됩니다. 이 방식에 따라 각 파일은 하나의 장치가 가득 찰 때까지 해당 장치에 완전히 할당됩니다. 라운드 로빈은 다중 스트림 환경에 적합한 설정입니다.

`-o stripe=n`이 0보다 큰 정수로 설정되면 파일은 스트라이프 방식을 사용하여 파일 시스템 장치에 할당됩니다. 적합한 `-o stripe=n` 설정을 결정하려면 여러 설정값을 적용하여 성능 수치를 확인해 보십시오. 스트라이프는 특정한 대역폭이 요구되는 터키(일괄 공급) 응용 프로그램에 적합한 설정입니다.

스트라이프 너비는 `/etc/vfstab` 파일 또는 `samfs.cmd` 파일에서 설정할 수도 있습니다.

mount(1M) 명령에 대한 자세한 내용은 `mount_samfs(1M)` 매뉴얼 페이지를 참조하십시오. `samfs.cmd` 파일에 대한 자세한 내용은 `samfs.cm(4)` 매뉴얼 페이지를 참조하십시오.

---

## Qwrite 기능 활성화

기본적으로 Sun StorageTek QFS 파일 시스템은 동일한 파일에 대한 동시 읽기 및 쓰기를 비활성화합니다. 이것은 UNIX vnode 인터페이스 표준으로 정의된 모드로서 오직 하나의 쓰기 작업에만 독점적인 액세스 권한이 부여되고 다른 생성기 및 판독기 호스트는 대기해야 합니다. Qwrite는 다른 스레드에서 동일한 파일에 대한 동시 읽기 및 쓰기를 가능하게 합니다.

Qwrite 기능은 동일한 파일에 대한 다중 동시 트랜잭션을 위해 데이터베이스 응용 프로그램에서 사용될 수 있습니다. 데이터베이스 응용 프로그램은 일반적으로 용량이 큰 파일을 관리하고 동일한 파일에 대한 동시 읽기 및 쓰기를 수행합니다. 하지만 파일에 대한 각 시스템 호출은 커널 내에서 읽기/쓰기 잠금을 설정하고 해제하게 됩니다. 이러한 잠금은 동일한 파일에 대한 동시 작업을 막습니다. 응용 프로그램 자체에서 파일 잠금 메커니즘을 구현할 경우 커널 잠금 메커니즘은 I/O를 불필요하게 직렬화하여 성능을 저하시킵니다.

Qwrite는 `/etc/vfstab` 파일, `samfs.cmd` 파일 및 마운트 옵션으로 활성화할 수 있습니다. mount(1M) 명령의 `-o qwrite` 옵션은 파일 시스템 잠금 메커니즘을 거치지 않고 ([NFS]를 통해 파일 시스템에 액세스하는 응용 프로그램 제외) 응용 프로그램에서 데이터 액세스를 제어하도록 합니다. qwrite가 지정되면 파일 시스템은 다른 스레드에서 동일한 파일에 대한 동시 읽기 및 쓰기가 가능하게 됩니다. 이 옵션은 드라이브 레벨에서 여러 요청을 대기열에 두어 I/O 성능을 향상시킵니다.

다음은 mount(1M) 명령을 사용하여 데이터베이스 파일 시스템에서 Qwrite를 활성화하는 예입니다.

```
# mount -F samfs -o qwrite /db
```

이 기능에 대한 자세한 내용은 samfs.cmd(4) 매뉴얼 페이지의 qwrite 명령 또는 mount\_samfs(1M) 매뉴얼 페이지의 -o qwrite 옵션을 참조하십시오.

---

## 쓰기 스로틀 설정

-o wr\_throttle=*n* 옵션은 한 파일에 대해 해결되지 않은 쓰기 KB의 수를 *n*으로 제한합니다. 기본적으로 Sun StorageTek QFS 파일 시스템에서 wr\_throttle은 16MB로 설정됩니다.

파일에 *n* 쓰기 KB가 미해결 상태인 경우, 해당 파일에 쓰기를 시도하는 응용 프로그램은 충분한 바이트로 I/O를 완료하여 응용 프로그램이 다시 시작할 수 있을 때까지 일시 중단됩니다.

파일 시스템에 액세스하는 수천 개의 NFS 공유 워크스테이션과 같이 사이트에 수 많은 스트림이 있는 경우, 한 번에 과도한 양의 메모리가 디스크로 플러시되는 것을 피하기 위해 -o wr\_throttle=*n* 옵션을 조정할 수 있습니다. 일반적으로, 스트림의 수를 -o wr\_throttle=*n* 옵션의 1024 x *n* 인수와 곱한 값이 호스트 시스템 메모리의 총 크기에서 Solaris OS가 필요로 하는 메모리의 크기를 뺀 값보다 작아야 합니다(다음 공식 참조).

```
number-of-streams x n x 1024 < total-memory - Solaris-OS-memory-needs
```

턴키(일괄 공급) 응용 프로그램의 경우 메모리에 더 많은 페이지를 보유하므로 기본값인 16,384KB보다 큰 값을 사용해야 할 수 있습니다.

---

## Flush-Behind 속도 설정

두 개의 마운트 매개변수가 연속으로 쓰여지는 페이지 및 스테이지 페이지에 대한 Flush-Behind 속도를 제어합니다. `flush_behind` 및 `stage_flush_behind` 마운트 매개변수는 `samfs.cmd` 파일, `/etc/vfstab` 파일 또는 `mount(1M)` 명령에서 지정됩니다.

`flush_behind=n` 마운트 매개변수는 최대 Flush-Behind 값을 설정합니다. 순차적으로 쓰여지면서 수정된 페이지는 디스크에 비동기적으로 쓰여져 Solaris™ Volume Manager(SVM) 레이어가 페이지를 깨끗하게 유지하도록 도와줍니다. 이 기능을 활성화하려면  $n$ 을 16 ~ 8192의 정수로 설정하십시오. 기본적으로  $n$ 은 0으로 설정되며 이 기능을 비활성화합니다.  $n$  인수는 KB 단위로 지정됩니다.

`stage_flush_behind=n` 마운트 매개변수는 최대 스테이지 Flush-Behind 값을 설정합니다. 스테이지되는 스테이지 페이지는 디스크에 비동기적으로 쓰여져 SVM 레이어가 페이지를 깨끗하게 유지하도록 도와줍니다. 이 기능을 활성화하려면  $n$ 을 16 ~ 8192의 정수로 설정하십시오. 기본적으로  $n$ 은 0으로 설정되며 이 기능을 비활성화합니다.  $n$  인수는 KB 단위로 지정됩니다.

이러한 마운트 매개변수에 대한 자세한 내용은 `mount_samfs(1M)` 매뉴얼 페이지 또는 `samfs.cmd(4)` 매뉴얼 페이지를 참조하십시오.

---

## inode 수 및 inode 해시 테이블 조정

Sun StorageTek QFS 파일 시스템을 사용하여 `/etc/system` 파일에서 다음 두 개의 조정 가능한 매개변수를 설정할 수 있습니다.

- `ninodes`
- `nhino`

이러한 매개변수에 대해 기본값이 아닌 설정을 활성화하려면 `/etc/system` 파일을 편집한 후 시스템을 재부트합니다.

다음 하위 절에서는 이러한 매개변수에 대해 자세히 설명합니다.

## ninodes 매개변수

ninodes 매개변수는 기본값 inode의 최대 수를 지정합니다. ninodes의 값은 응용 프로그램이 많은 inode를 사용하지 않을 때에도 Sun StorageTek QFS가 자신에게 할당되도록 유지하는 코어 내 inode의 수를 결정합니다.

/etc/system 파일에서 이 매개변수의 형식은 다음과 같습니다.

```
set samfs:ninodes = value
```

value의 범위는 16 ~ 2000000입니다. ninodes의 기본값은 다음 중 하나입니다.

- ncsize 설정과 동일한 값. ncsize 매개변수는 디렉토리 이름 조회 캐시(DNLC)의 항목 수를 지정하는 Solaris 조정 매개변수입니다. ncsize에 대한 자세한 정보는 Solaris Tunable Parameters Reference Manual을 참조하십시오.
- 2000. 파일 시스템은 ncsize 설정이 0이거나 범위를 벗어난 경우 ninodes를 2000으로 설정합니다.

## nhino 매개변수

nhino 매개변수는 코어 내의 inode 해시 테이블의 크기를 지정합니다.

/etc/system 파일에서 이 매개변수의 형식은 다음과 같습니다.

```
set samfs:nhino = value
```

value의 범위는 1 ~ 1048756입니다. value는 2의 0이 아닌 거듭제곱이어야 합니다. nhino의 기본값은 다음 중 하나입니다.

- 8로 나누고 필요한 경우 가장 가까운 2의 거듭제곱으로 반올림된 ninodes 값과 동일한 값. 예를 들어 /etc/system에 다음 행이 존재한다고 가정합니다.

```
set samfs:ninodes 8000
```

이 예제에서 nhino가 설정되지 않은 경우, 시스템은 8000을 8로 나눈 후 2의 거듭제곱 값 중 가장 가까운 값( $2^{10}$ )으로 반올림한 1024로 가정합니다.

- 512. 파일 시스템은 ninodes 설정이 범위를 벗어난 경우 nhino를 512로 설정합니다.

## ninodes 및 nhino 매개변수 설정 시기

숫자로 inode를 검색하면(디렉토리에서 inode 번호를 얻은 후 또는 NFS 파일 핸들에서 inode 번호를 추출한 후) Sun StorageTek QFS 파일 시스템은 코어 내 inode의 캐시를 검색합니다. 이 프로세스 속도를 높이기 위해 파일 시스템은 해시 테이블을 유지하여 확인해야 할 inode의 수를 줄입니다.

큰 해시 테이블은 적당량의 메모리를 사용하여 비교 및 검색의 횟수를 줄입니다. nhino 값이 너무 큰 경우, 전체 inode 목록(inode 동기화 및 마운트 해제)을 통해 제거하는 작업을 수행하면 시스템이 더 느려집니다. 많은 수의 파일을 조작하는 사이트 및 대량의 NFS I/O를 수행하는 사이트의 경우, 이러한 매개변수를 기본값보다 크게 설정하는 것이 이로울 수 있습니다.

사이트에 적은 수의 파일만 있는 파일 시스템이 있는 경우, 이러한 수를 기본값보다 작게 만드는 것이 이로울 수도 있습니다. 다른 파일 시스템을 백업하기 위해 큰 단일 파일인 tar(1) 파일을 기록하는 파일 시스템이 있는 경우를 예로 들 수 있습니다.



# 문제 해결 Sun StorageTek QFS

이 부록에서는 Sun StorageTek QFS 파일 시스템의 문제를 해결하는 데 사용할 수 있는 몇몇 도구와 절차에 대해 설명합니다. 특히 다음 항목에 대해 설명합니다.

- 223페이지의 "파일 시스템 무결성 확인 및 파일 시스템 복구"
- 225페이지의 "공유 파일 시스템에서 실패하거나 멈춘 `sammkfs(1M)` 또는 `mount(1M)` 명령 문제 해결"
- 240페이지의 "Linux 클라이언트 문제 해결"

## 파일 시스템 무결성 확인 및 파일 시스템 복구

Sun StorageTek QFS 파일 시스템은 파일 시스템 작업에 중요한 다음 레코드에 유효성 레코드를 씁니다. 디렉토리, 간접 블록 및 inode. 디렉토리 검색 중 파일 시스템이 손상을 감지하는 경우 EDOM 오류를 발행하고 해당 디렉토리는 처리되지 않습니다. 간접 블록이 유효하지 않은 경우 ENOCSI 오류를 발행하고 해당 파일은 처리되지 않습니다. 표 A-1에 이러한 오류 표시기가 요약되어 있습니다.

표 A-1 오류 표시기

오류	Solaris OS 의미	Sun StorageTek QFS 의미
EDOM	인수가 도메인 영역을 벗어납니다.	유효성 레코드의 값이 범위를 벗어납니다.
ENOCSI	CSI 구조를 사용할 수 없습니다.	구조 사이의 링크가 유효하지 않습니다.

또한 inode가 검증되고 디렉토리와 교차 검사됩니다.

다음 파일에서 오류 조건을 모니터해야 합니다.

- 표 A-1에 표시된 오류에 대해 /etc/syslog.conf에 지정된 로그 파일
- 장치 오류에 대한 /var/adm/messages 파일

불일치가 발견되면 파일 시스템을 마운트 해제하고 `samfsck(1M)` 명령을 사용하여 검사해야 합니다.

---

주 - `samfsck(1M)` 명령을 마운트된 파일 시스템에서 실행할 수 있지만 그 결과를 신뢰할 수 없습니다. 이 때문에 마운트되지 않은 파일 시스템에서만 명령을 실행하는 것이 바람직합니다.

---

## ▼ 파일 시스템 확인

- `samfsck(1M)` 명령을 사용하여 파일 시스템 검사를 수행합니다.  
이 명령을 다음 형식으로 사용합니다.

```
samfsck -V family-set-name
```

*family-set-name*에 대해 `mcf` 파일에서 지정된 파일 시스템 이름을 지정하십시오.

다음과 같이 `tee(1)` 명령과 함께 사용하여 `samfsck(1M)`의 수행 결과를 화면 및 파일 모두에 보낼 수 있습니다.

- C 셸:

```
# samfsck -V family-set-name |& tee file
```

- Bourne 셸:

```
# samfsck -V family-set-name 2>&1 | tee file
```

`samfsck(1M)`가 반환하는 치명적이지 않은 오류는 NOTICE로 시작합니다. 치명적이지 않은 오류는 블록 손실 및 Orphan입니다. NOTICE 오류가 반환되어도 파일 시스템은 여전히 일관성이 유지됩니다. 편리한 예약된 유지 보수 정전 중 이러한 비치명적인 오류를 복구할 수 있습니다.

치명적인 오류 앞에는 ALERT가 붙습니다. 이러한 오류에는 중복된 블록, 유효하지 않은 디렉토리, 유효하지 않은 간접 블록 등이 포함됩니다. 이러한 오류가 발생하면 파일 시스템은 일관성이 없게 됩니다. ALERT 오류가 하드웨어 오작동으로 발생한 것이 아닌 경우에는 Sun에 문의하십시오.

samfsck(1M) 명령이 파일 시스템 손상을 감지하고 ALERT 메시지를 반환하는 경우 손상 이유를 판별해야 합니다. 하드웨어에 결함이 있는 경우 파일 시스템을 복구하기 전에 문제를 해결하십시오.

samfsck(1M) 및 tee(1) 명령에 대한 자세한 내용은 samfsck(1M) 및 tee(1) 매뉴얼 페이지를 참조하십시오.

## ▼ 파일 시스템 복구

1. umount(1M) 명령을 사용하여 파일 시스템을 마운트 해제합니다.

파일 시스템이 마운트되지 않은 경우 samfsck(1M) 명령을 실행하십시오. 파일 시스템 마운트 해제에 대한 자세한 내용은 [62페이지의 "파일 시스템 마운트 해제"](#)를 참조하십시오.

2. samfsck(1M) 명령을 사용하여 파일 시스템 복구합니다. 공유 파일 시스템을 복구 중인 경우 메타 데이터 서버에서 명령을 실행하십시오.

다음 형식으로 samfsck(1M) 명령을 실행하여 파일 시스템을 복구할 수 있습니다.

```
# samfsck -F -V fsname
```

*fsname*에 대해 mcf 파일에서 정의된 파일 시스템 이름을 지정합니다.

---

## 공유 파일 시스템에서 실패하거나 멈춘 sammkfs(1M) 또는 mount(1M) 명령 문제 해결

다음 절에서는 공유 파일 시스템에서 sammkfs(1M) 또는 mount(1M) 명령이 실패하거나 mount(1M) 명령이 멈추는 경우 해결하는 방법에 대해 설명합니다.

이 절의 절차는 클라이언트 호스트 및 서버에서 수행할 수 있습니다. 메타 데이터 서버에서만 실행할 수 있는 명령 앞에는 server# 프롬프트가 나타납니다.

## 실패한 sammkfs(1M) 명령 복구

sammkfs(1M) 명령 수행 결과에 예상치 못한 장치가 초기화되어야 함을 나타내는 오류나 메시지를 반환하는 경우 다음 절차를 수행해야 합니다. 여기에는 mcf 파일 확인 및 mcf 파일 변경 사항을 시스템에 전파하는 단계가 포함되어 있습니다.

### ▼ mcf 파일 확인 및 mcf 파일 변경을 시스템으로 전파

1. sam-fsd(1M) 명령을 사용하여 mcf 파일을 확인합니다.

예를 들어, 다음과 같습니다.

```
# sam-fsd
```

sam-fsd(1M) 명령의 출력에서 검사하고 수정해야 할 오류가 있는지 확인합니다.

2. sam-fsd(1M) 명령의 출력에서 /etc/opt/SUNWsamfs/mcf 파일에 오류가 있는 것으로 표시되면 mcf 파일을 편집하여 이 문제를 해결합니다.
3. sam-fsd(1M) 명령을 다시 실행하여 mcf 파일을 확인합니다.

sam-fsd(1M) 명령 출력 결과에 mcf 파일이 올바르다고 나타날 때까지 위의 1단계, 2단계 및 3단계를 반복하십시오.

4. samd(1M) config 명령을 실행합니다.

sam-fsd 데몬에 구성 변경 사항을 알려 mcf 파일 변경 사항을 전파해야 합니다.

예를 들어, 다음과 같습니다.

```
# samd config
```

## 실패한 mount(1M) 명령 복구

mount(1M) 명령은 여러 가지 원인으로 실패할 수 있습니다. 이 절에서는 마운트 문제를 해결하는 데 사용할 수 있는 몇 가지 작업에 대해 설명합니다. mount(1M) 명령이 실패한 것이 아니라 멈춘 경우에는 232페이지의 "멈춘 mount(1M) 명령 복구"를 참조하십시오.

몇 가지 실패한 mount(1M) 유형 및 해당 해결 방법은 다음과 같습니다.

- 클라이언트에서 Shared server is not mounted라는 메시지가 생성되면서 mount(1M) 명령이 실패하는 경우, 서버 호스트를 결정하고 해당 파일 시스템을 메타 데이터 서버에 마운트하십시오.
- 파일 시스템과 mcf 파일 사이에 일치하지 않는 요소가 있다는 메시지가 나타나면 mount 명령이 실패할 경우 다음을 확인하십시오.
  - mcf 파일이 구문적으로 올바릅니다. 자세한 내용은 [226페이지의 "mcf 파일 확인 및 mcf 파일 변경을 시스템으로 전파"](#)를 참조하십시오.
  - mcf 파일에 대한 최근의 변경 사항이 올바르게 지정되었는지 확인합니다. 자세한 내용은 [226페이지의 "mcf 파일 확인 및 mcf 파일 변경을 시스템으로 전파"](#)를 참조하십시오.
  - mcf 파일이 클라이언트의 어떠한 차이로 인해 조정된 장치 이름이나 컨트롤러 번호에 대해 서버의 mcf 파일과 일치하는지 확인합니다. samfsconfig(1M) 명령을 사용하여 이러한 문제 중 일부를 진단할 수 있습니다. samfsconfig(1M) 명령 사용에 대한 자세한 내용은 [230페이지의 "samfsconfig\(1M\) 명령 사용"](#)을 참조하십시오.
- mount(1M) 명령이 기타 다른 이유로 실패한 경우에는 다음 절에 설명된 절차에 따라 mount(1M) 명령을 올바르게 수행하기 위해 갖추어야 할 시스템 요소를 확인하십시오. 이러한 절차는 다음과 같습니다.
  - [227페이지의 "파일 시스템의 마운트 가능 여부 확인"](#)
  - [229페이지의 "samfsinfo\(1M\) 및 samsharefs\(1M\) 명령 사용"](#)
  - [230페이지의 "samfsconfig\(1M\) 명령 사용"](#)

## ▼ 파일 시스템의 마운트 가능 여부 확인

이 절차에서 오류가 나타나지 않는 경우 [229페이지의 "samfsinfo\(1M\) 및 samsharefs\(1M\) 명령 사용"](#)을 수행하십시오. 파일 시스템이 작성되고 공유 호스트 파일이 올바르게 초기화되어 있는지 확인하는 데 도움이 됩니다.

다음 절차에서는 mount(1M) 명령이 실패할 경우 확인해야 할 사항에 대해 설명합니다.

### 1. 마운트 지점 디렉토리가 있는지 확인합니다.

확인할 수 있는 방법은 다양합니다. 예를 들어 다음 형식으로 ls(1) 명령을 실행할 수 있습니다.

```
ls -ld mountpoint
```

마운트지점에 대해 Sun StorageTek QFS 공유 파일 시스템의 마운트 지점의 이름을 지정합니다.

ls(1) 명령의 출력을 검사할 때 출력에는 액세스 모드 755가 있는 디렉토리가 나타나야 합니다. 즉, 해당 코드가 drwxr-xr-x로 표시되어야 합니다. 코드 예 A-1은 출력 예제를 표시합니다.

코드 예 A-1                    액세스 모드 값

```
# ls -ld /sharefs1
drwxr-xr-x  2 root      sys           512 Mar 19 10:46 /sharefs1
```

액세스 모드가 이 레벨이 아닌 경우에는 다음과 같이 chmod(1) 명령을 입력하십시오.

```
# chmod 755 mountpoint
```

마운트지점에 대해 Sun StorageTek QFS 공유 파일 시스템의 마운트 지점의 이름을 지정합니다.

2. /etc/vfstab 파일에 파일 시스템에 대한 항목이 있는지 확인합니다.

코드 예 A-2는 일명 sharefs1 공유 파일 시스템에 대한 항목을 표시합니다.

코드 예 A-2                    /etc/vfstab 파일 예제

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

/etc/vfstab 파일에서 공유 파일 시스템의 항목에 대한 Mount Parameters 필드에 shared 플래그가 있는지 확인합니다.

3. 마운트 지점 디렉토리가 NFS 사용에 대해 공유되어 있지 않은지 확인합니다.

마운트 지점이 공유되어 있는 경우 unshare(1M) 명령을 사용하여 공유를 해제하십시오. 예를 들어, 다음과 같습니다.

```
# unshare mountpoint
```

마운트지점에 대해 Sun StorageTek QFS 공유 파일 시스템의 마운트 지점의 이름을 지정합니다.

## ▼ samfsinfo(1M) 및 samsharefs(1M) 명령 사용

이 절차는 이러한 명령의 출력을 분석하는 방법을 표시합니다.

### 1. 서버에서 samfsinfo(1M) 명령을 입력합니다.

이 명령을 다음 형식으로 사용합니다.

```
samfsinfo filesystem
```

*filesystem*에 대해 mcf 파일에 지정된 대로 Sun StorageTek QFS 공유 파일 시스템의 이름을 지정합니다. 코드 예 A-3은 samfsinfo(1M) 명령 및 출력을 표시합니다.

코드 예 A-3 samfsinfo(1M) 명령 예제

```
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
```

코드 예 A-3의 출력 결과에서 아래 행에 shared 키워드가 있는 것을 볼 수 있습니다.

```
name:      sharefs1      version:      2      shared
```

파일 시스템 장치, 서수 및 장비 번호 목록이 아래 행 다음에 나타나 있습니다.

```
ord  eq  capacity      space  device
```

이러한 번호는 해당 파일 시스템의 mcf(4) 항목에 있는 장치와 일치해야 합니다.

2. 서버에서 **samsharefs(1M)** 명령을 입력합니다.

이 명령을 다음 형식으로 사용합니다.

```
samsharefs -R filesystem
```

*filesystem*에 대해 mcf 파일에 지정된 대로 Sun StorageTek QFS 공유 파일 시스템의 이름을 지정합니다. 코드 예 A-4는 samsharefs(1M) 명령 및 출력을 표시합니다.

코드 예 A-4 samsharefs(1M) 명령 예제

```
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 -
```

다음 정보는 samfsinfo(1M) 또는 samsharefs(1M) 명령의 진단 출력에 관련된 것입니다.

- 두 명령을 수행하여 진단 또는 오류 메시지가 나타나면 해당 문제를 해결하십시오. samfsinfo(1M) 명령의 수행 결과에 shared 키워드가 포함되어 있는지 확인하십시오.
- 이러한 명령은 대체 서버 호스트와 파일 시스템에 대한 호스트의 mcf(4) 항목에 nodev 장치가 없는 클라이언트 호스트에서 실행할 수 있습니다.

samfsinfo(1M) 및 samsharefs(1M) 명령을 수행하여 불규칙성이 나타나지 않으면 [230페이지의 "samfsconfig\(1M\) 명령 사용"](#)을 수행하십시오.

## ▼ samfsconfig(1M) 명령 사용

파일 시스템에 대한 mcf 파일에 nodev 장치 항목이 있는 클라이언트에서는 전체 파일 시스템에 액세스할 수 없고 공유 호스트 파일에도 직접 액세스할 수 없습니다.

samfsconfig(1M) 명령을 사용하여 공유 파일 시스템의 데이터 분할 영역에 대한 액세스 가능 여부를 결정할 수 있습니다.

- samfsconfig(1M) 명령을 실행합니다.

이 명령을 다음 형식으로 사용합니다.

```
samfsconfig list-of-devices
```

*list-of-devices*의 경우 mcf 파일의 파일 시스템 항목으로부터 장치 목록을 지정합니다. 공백을 사용하여 목록에 있는 여러 장치를 구분하십시오.

예제 1. 코드 예 A-5는 tethys 호스트에 대한 mcf 파일을 보여줍니다. 이 호스트는 mcf 파일에 nodev 항목이 없는 호스트입니다. 그런 다음 실행된 samfsconfig(1M) 명령이 표시됩니다.

코드 예 A-5                    nodev 항목이 없는 samfsconfig(1M) 명령 예제

```
tethys# cat /etc/opt/SUNWsamfs/mcf
sharefs1          10  ma  sharefs1  on  shared
/dev/dsk/c1t210002037E9C296d0s6 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/c1t210002037E9C296d0s6
/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/c1t210002037E9C296d0s6 11  mm  sharefs1  -
/dev/dsk/c3t50020F2300005D22d0s6 12  mr  sharefs1  -
/dev/dsk/c3t50020F2300006099d0s6 13  mr  sharefs1  -
/dev/dsk/c3t50020F230000651Cd0s6 14  mr  sharefs1  -
```

예제 2. 코드 예 A-6은 해당 mcf 파일의 nodev 항목이 있는 호스트에서 사용되는 samfsconfig(1M) 명령을 표시합니다.

코드 예 A-6                    nodev 항목이 있는 samfsconfig(1M) 명령 예제

```
dione# cat /etc/opt/SUNWsamfs/mcf
sharefs1          10  ma  sharefs1  on  shared
nodev             11  mm  sharefs1  -
/dev/dsk/c4t50020F23000055A8d0s3 12  mr  sharefs1  -
/dev/dsk/c4t50020F23000055A8d0s4 13  mr  sharefs1  -
/dev/dsk/c4t50020F23000055A8d0s5 14  mr  sharefs1  -
dione# samfsconfig /dev/dsk/c4t50020F23000055A8d0s3
/dev/dsk/c4t50020F23000055A8d0s4 /dev/dsk/c4t50020F23000055A8d0s5
# Family Set 'sharefs1' Created Mon Apr 29 15:12:18 2002
# Missing slices
# Ordinal 1
# /dev/dsk/c4t50020F23000055A8d0s3 12  mr  sharefs1  -
# /dev/dsk/c4t50020F23000055A8d0s4 13  mr  sharefs1  -
# /dev/dsk/c4t50020F23000055A8d0s5 14  mr  sharefs1  -
```



- b. 구성 파일 특히 /etc/opt/SUNWsamfs/mcf를 검사합니다.
  - c. 구성 파일을 검사하고 데몬이 활성화되어 있는지 확인한 후 이 절차를 다시 시작합니다.
4. samsharefs(1M) 명령을 입력하여 호스트 파일을 검사합니다.

코드 예 A-11은 samsharefs(1M) 명령 및 올바른 출력을 표시합니다.

코드 예 A-8 samsharefs(1M) -R 명령

```
titan-server# samsharefs -R sharefs1
#
# Host file for family set 'sharefs1'
#
# Version: 3      Generation: 50      Count: 4
# Server = host 0/titan, length = 216
#
titan 173.26.2.129 1 - server
tethys 173.26.2.130 2 -
dione dione 0 -
mimas mimas 0 -
```

시스템 출력에서 다음을 확인하십시오.

- 출력의 1 열에 호스트 이름이 있고 서버로 지정되어 있는지 확인합니다.
- 호스트 IP 주소는 2 열에 있습니다. 다중 IP 주소가 있는 경우 모두 유효해야 합니다.

5. 서버에서 netstat(1M) 명령을 입력합니다.

코드 예 A-9는 서버 titan에서 입력한 netstat(1M) 명령을 표시합니다.

코드 예 A-9 서버에서의 netstat(1M) 명령

```
titan-server# netstat -a | grep sam-qfs
*.sam-qfs *.*                0      0 24576  0 LISTEN
*.sam-qfs *.*                0      0 24576  0 LISTEN
titan.32834 titan.sam-qfs 32768  0 32768  0 ESTABLISHED
titan.sam-qfs titan.32891 32768  0 32768  0 ESTABLISHED
titan.sam-qfs tethys.32884 24820  0 24820  0 ESTABLISHED
titan.sam-qfs dione.35299 24820  0 24820  0 ESTABLISHED
*.sam-qfs *.*                0      0 24576  0 LISTEN
```

서버에서 netstat(1M) 명령의 출력에 다음이 포함되어 있는지 확인합니다.

- LISTEN 항목 세 개
- 호스트에 대한 ESTABLISHED 항목 두 개
- 구성되어 sam-fsd 데몬을 실행중인 각 클라이언트에 대한 ESTABLISHED 항목 하나

이 예제는 tethys 및 dione에 대한 ESTABLISHED 항목을 표시합니다. 마운트 여부에 상관 없이 구성 및 실행되고 있는 각 클라이언트에 대해 ESTABLISHED 항목이 하나 있어야 합니다.

## 6. 클라이언트에서 netstat(1M) 명령을 입력합니다.

[코드 예 A-10](#)은 클라이언트 dione에서 입력한 netstat(1M) 명령을 표시합니다.

코드 예 A-10                    클라이언트에서의 netstat(1M) 명령

```
dione-client# netstat -a | grep sam-qfs
*.sam-qfs      *.*                0      0 24576      0 LISTEN
*.sam-qfs      *.*                0      0 24576      0 LISTEN
dione.32831    titan.sam-qfs    24820   0 24820      0 ESTABLISHED
*.sam-qfs      *.*                0      0 24576      0 LISTEN
```

## 7. 해당 출력에 다음이 포함되어 있는지 확인하십시오.

- LISTEN 항목 세 개. 모든 항목은 sam-fsd 데몬용입니다.
- ESTABLISHED 항목 하나

이러한 행이 나타나면 네트워크 연결이 설정됩니다.

ESTABLISHED 연결이 보고되지 않으면 다음 중 하나 이상의 절차를 수행하십시오.

- [234페이지의 "클라이언트가 서버에 연결할 수 있는지 확인"](#)
- [237페이지의 "서버가 클라이언트에 연결할 수 있는지 확인"](#)
- [238페이지의 "sam-sharefsd 추적 로그 확인"](#)

## ▼ 클라이언트가 서버에 연결할 수 있는지 확인

[232페이지의 "네트워크 연결 확인"](#)에 설명된 절차를 사용해도 ESTABLISHED 연결이 나타나지 않으면 아래 단계를 수행하십시오.

### 1. samsharefs(1M) 명령을 사용하여 서버의 호스트 파일을 확인합니다.

samsharefs(1M) 명령은 대체 서버 호스트 및 파일 시스템에 대한 호스트의 mcf 항목에 nodev 장치가 없는 클라이언트 호스트에서 실행할 수 있습니다. 이 단계의 경우 이 명령을 다음 형식으로 사용합니다.

```
samsharefs -R filesystem
```

*filesystem*에 대해 mcf 파일에 지정된 대로 Sun StorageTek QFS 공유 파일 시스템의 이름을 지정합니다. 코드 예 A-11은 samsharefs(1M) -R 명령을 표시합니다.

코드 예 A-11                    samsharefs(1M) -R 명령

```
titan-server# samsharefs -R sharefs1
#
# Host file for family set 'sharefs1'
#
# Version: 3        Generation: 50        Count: 4
# Server = host 0/titan, length = 216
#
titan 173.26.2.129 1 - server
tethys 173.26.2.130 2 -
dione dione 0 -
mimas mimas 0 -
```

2. 이 출력 결과를 저장합니다.

이 절차의 단계가 실패할 경우, 다음 절차를 수행할 때 이 출력 결과가 필요합니다.

3. 출력 결과가 예상과 일치하는지 확인합니다.

명령이 실패할 경우 파일 시스템이 만들어졌는지 확인하십시오. 이런 경우, 다음 중 하나가 발생할 수 있습니다.

- mcf 파일이 올바르게 생성되지 않았습니다. samfsconfig(1M) 명령을 사용하여 mcf 파일이 올바른지 확인하십시오.
- 파일 시스템이 만들어지지 않았습니다.
- 초기 호스트 구성 파일이 만들어지지 않았습니다. 구성 프로세스에 기존의 mcf 파일 편집, 시스템의 나머지 부분에 mcf 파일 변경 사항 전파 및 호스트 파일 구성이 포함됩니다.

4. 첫 번째 열에서 서버 이름이 들어 있는 행을 찾습니다.

5. 클라이언트에서 samsharefs(1M) 출력 결과의 두 번째 열의 각 항목에 대해 ping(1M) 명령을 사용하여 서버에 연결할 수 있는지 확인합니다.

이 명령을 다음 형식으로 사용합니다.

```
ping servername
```

*servername*에 samsharefs(1M) 명령 출력 결과의 두 번째 열에 나타난 서버의 이름을 지정합니다.

코드 예 A-12는 ping(1M)의 출력을 표시합니다.

코드 예 A-12            samsharefs(1M) 출력 결과에 나타난 시스템 이름에 대해 ping(1M) 사용

```
dione-client# ping 173.26.2.129
ICMP Host Unreachable from gateway dione (131.116.7.218)
for icmp from dione (131.116.7.218) to 173.26.2.129
dione-client# ping titan
titan.foo.com is alive
```

6. ping(1M) 명령을 실행했을 때 연결할 수 없는 호스트가 표시되는 경우 클라이언트에서 hosts.filesystem.local 파일을 확인합니다.

samsharefs(1M) 출력의 두 번째 열에 두 개 이상의 항목이 있고 일부 항목에 연결할 수 없는 경우, 공유 파일 시스템이 사용할 항목에 대해 연결 가능한 항목만 나타나는지 확인합니다. 또한 필수 항목이 해당 호스트의 /etc/opt/SUNWsamfs/hosts.filesystem.local 파일 항목에 나타나는지 확인합니다. 연결할 수 없는 호스트는 여기에 입력되지 않아야 합니다.

sam-sharefsd 데몬이 연결할 수 없는 서버 인터페이스에 연결을 시도할 경우 설치, 재부팅 또는 파일 시스템 호스트 재구성 이후 서버에 연결하는 데 상당한 지연 시간이 발생할 수 있습니다. 사실 이것은 메타 데이터 서버 장애 조치 작업에 영향을 줍니다.

코드 예 A-13은 hosts.sharefs1.local 파일을 표시합니다.

코드 예 A-13            hosts.filesystem.local 파일 검사

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

7. ping(1M) 명령을 실행했을 때 연결할 수 있는 서버 인터페이스가 없는 것으로 표시되면 올바른 서버 인터페이스를 활성화시킵니다.

일반적인 작업을 위해 서버 네트워크 인터페이스를 구성 또는 초기화하거나 samsharefs(1M) 명령을 사용하여 호스트 파일에 있는 인터페이스 이름이 실제 이름과 일치하도록 해당 이름을 업데이트합니다.

## ▼ 서버가 클라이언트에 연결할 수 있는지 확인

232페이지의 "네트워크 연결 확인"에 설명된 절차를 사용해도 ESTABLISHED 연결이 나타나지 않으면 아래 단계를 수행하십시오.

1. `samsharefs(1M)` 명령의 수행 결과를 사용합니다.

234페이지의 "클라이언트가 서버에 연결할 수 있는지 확인"에서 생성된 출력을 사용하거나 해당 절차의 첫 번째 단계를 수행하여 다시 생성할 수 있습니다.

2. 첫 번째 열에서 클라이언트의 이름이 들어 있는 행을 찾습니다.

3. 클라이언트에서 `hostname(1M)` 명령을 실행하고 출력 결과가 `samsharefs(1M)` 출력 결과의 첫 번째 열에 있는 이름과 일치하는지 확인합니다.

코드 예 A-14는 `hostname(1M)` 명령 및 해당 출력을 표시합니다.

코드 예 A-14                    `hostname(1M)` 출력 결과

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

4. `hostname(1M)` 명령 출력이 `samsharefs(1M)` 출력의 두 번째 열에 있는 이름과 일치하는 경우 서버에서 `ping(1M)` 명령을 사용하여 클라이언트에 연결할 수 있는지 확인합니다.

코드 예 A-15는 `ping(1M)` 명령 및 해당 출력 결과를 표시합니다.

코드 예 A-15                    `ping(1M)` 출력 결과

```
titan-server# ping dione
dione is alive
```

코드 예 A-13의 2열에 있는 모든 항목이 연결 가능할 필요는 없지만, 모든 가능한 서버가 연결을 수용하도록 하려면 모든 인터페이스는 해당 열에 있어야 합니다. 해당 서버는 공유 호스트 파일에 선언되지 않은 인터페이스로부터의 연결을 거부합니다.

5. `ping(1M)` 명령을 실행했을 때 연결할 수 있는 클라이언트 인터페이스가 없는 것으로 표시되면 올바른 클라이언트 인터페이스를 활성화시킵니다.

일반적인 작업을 위해 클라이언트 네트워크 인터페이스를 구성 또는 초기화하거나 `samsharefs(1M)` 명령을 사용하여 호스트 파일에 있는 인터페이스 이름이 실제 이름과 일치하도록 해당 이름을 업데이트합니다.

## ▼ sam-sharefsd 추적 로그 확인

추적 로그 파일은 작업 중 sam-sharefsd(1M) 데몬으로 생성된 정보를 보관합니다. 추적 로그 파일에는 호스트 파일 변경 사항 및 메타 데이터 서버 변경 사항과 같은 다른 작업 외에도 연결 시도/수신/거부/거절 등에 대한 정보가 포함되어 있습니다.

로그 파일의 추적 문제에는 로그 파일을 사용하여 다른 호스트에서 작업 순서를 일치시키는 것이 포함됩니다. 호스트의 시계가 동기화된 경우, 로그 파일 해석은 매우 단순화됩니다. 설치 단계 중 하나를 사용하여 네트워크 시간 데몬 xntpd(1M)를 활성화할 수 있습니다. 이것은 Sun StorageTek QFS 공유 파일 시스템 작업 중 메타 데이터 서버 및 모든 클라이언트 호스트의 시계를 동기화합니다.

추적 로그는 초기 구성 설정 시 특히 유용합니다. 클라이언트 로그는 발신 연결 시도를 표시합니다. 서버 로그 파일의 해당 메시지는 Sun StorageTek QFS 공유 파일 시스템의 네트워크 및 구성 문제를 진단하는 데 가장 유용한 도구 중 일부입니다. 로그 파일에는 대부분의 일반 문제를 해결하기 위한 진단 정보가 포함되어 있습니다.

다음 절차를 수행하면 mount(1M) 문제를 해결할 수 있습니다.

- 232페이지의 "네트워크 연결 확인"
- 234페이지의 "클라이언트가 서버에 연결할 수 있는지 확인"
- 237페이지의 "서버가 클라이언트에 연결할 수 있는지 확인"

위의 절차 중 어느 것으로도 문제가 해결되지 않으면 이 절에 있는 단계를 수행하십시오. 서버 및 클라이언트 호스트 모두에서 이 단계를 수행할 수 있습니다.

1. /var/opt/SUNWsamfs/trace/sam-sharefsd 파일이 있는지 확인합니다.  
이 파일이 없거나 최근의 수정 사항이 반영되지 않은 것으로 나타나면 다음 단계를 진행하십시오.  
해당 파일이 있는 경우 tail(1) 또는 다른 명령을 사용하여 파일의 마지막 몇 행을 검사하십시오. 상태가 의심스러운 경우, 이 절의 나머지 절차 중 하나 이상을 사용하여 문제를 조사합니다.
2. 1단계에서 /var/opt/SUNWsamfs/trace/sam-sharefsd 파일이 없거나 해당 파일이 최근에 수정되지 않은 것으로 나타나는 경우  
/etc/opt/SUNWsamfs/defaults.conf 파일을 편집하고 sam-sharefsd 추적을 활성화하는 행을 추가합니다.
  - a. defaults.conf 파일이 /etc/opt/SUNWsamfs에 없는 경우  
/opt/SUNWsamfs/examples/defaults.conf에서 /etc/opt/SUNWsamfs로 defaults.conf 예제 파일을 복사합니다.

```
# cd /etc/opt/SUNWsamfs
# cp /opt/SUNWsamfs/examples/defaults.conf .
```

- b. vi(1) 또는 다른 편집기를 사용하여 /etc/opt/SUNWsamfs/defaults.conf 파일을 편집하고 추적을 활성화하기 위한 행을 추가합니다.

코드 예 A-16은 defaults.conf 파일에 추가하는 행을 표시합니다.

코드 예 A-16 defaults.conf에서 추적을 활성화하는 행

```
trace
sam-sharefsd = on
sam-sharefsd.options = all
endtrace
```

- c. samd(1M) config 명령을 실행하여 sam-fsd(1M) 데몬을 재구성하고 새 defaults.conf 파일을 인식하도록 합니다.

예를 들어, 다음과 같습니다.

```
# samd config
```

- d. sam-fsd(1M) 명령을 실행하여 구성 파일을 검사합니다.

코드 예 A-17은 sam-fsd(1M) 명령 출력 결과를 표시합니다.

코드 예 A-17 sam-fsd(1M) 명령 출력

```
# 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-sharefsd의 로그 파일을 검사하여 오류를 확인합니다.

```
# more /var/opt/SUNWsamfs/trace/sam-sharefsd
```

3. 추적 파일의 마지막 10여 개의 행을 확인하여 진단 정보를 알아냅니다.

코드 예 A-18은 일반적인 sam-sharefsd 클라이언트 로그 파일을 표시합니다. 이 예제에서 해당 서버는 titan이고 클라이언트는 dione입니다. 이 파일에는 패키지 설치 후 발생한 정상적인 로그 항목이 포함되며 마운트된 파일 시스템에서 정상적으로 작동하는 데몬으로 끝납니다.

코드 예 A-18 클라이언트 추적 파일

```
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: rdsoc dione/0 (buf=6c000).
2004-03-23 16:13:15 shf-shsam2[13837:1]: FS shsam2: Signal 1 received: Hangup
2004-03-23 16:13:15 shf-shsam2[13837:1]: FS shsam2: Wakened from AWAIT_WAKEUP
2004-03-23 16:13:15 shf-shsam2[13837:1]: FS shsam2: mount; flags=18889
2004-03-23 16:18:55 shf-shsam2[13837:1]: FS shsam2: Signal 1 received: Hangup
2004-03-23 16:18:55 shf-shsam2[13837:1]: FS shsam2: Wakened from AWAIT_WAKEUP
```

## Linux 클라이언트 문제 해결

Linux 클라이언트와 Solaris 클라이언트는 서로 다른 절차를 사용하여 시스템 정보를 찾고 Sun StorageTek QFS 문제를 진단합니다.

Linux 커널의 시스템 정보가 있는 파일은 /proc 파일 시스템에 있습니다. 예를 들어, /proc/cpuinfo 파일은 하드웨어 정보를 포함하고 있습니다. 표 A-2는 유용한 문제 해결 정보가 포함된 몇몇 파일을 보여줍니다.

표 A-2 /proc 파일

파일 이름	제공 정보
버전	실행 중인 커널 버전
cpuinfo	하드웨어 정보
uptime	부트 시간 이후의 시간(초 단위) 및 프로세스에 소요된 전체 시간
모듈	로드된 모듈에 대한 정보
cmdline	부트 시에 커널로 전달된 명령줄 매개변수
filesystems	기존 파일 시스템 구현
scsi/scsi	연결된 SCSI 장치
fs/samfs/<QFS file system>/fsid	파일 시스템 ID로 반드시 NFS의 공유 옵션에 포함되어야 함

Linux 커널 로그 메시지는 /var/log/messages 파일에 기록됩니다.

## 문제 해결 도구

Linux 커널에는 다수의 변형이 있으므로 문제 해결이 매우 어려울 수 있습니다. 디버깅에 도움이 되는 몇 가지 도구가 있습니다.

- lkcd 및 kgdb와 같은 프로젝트에는 커널 충돌 덤프 정보가 있습니다.
- kdb, kgdb 및 icebox와 같은 프로젝트는 커널 디버거입니다.

---

주 - Red Hat Linux 또는 SuSE에는 이 프로젝트가 기본적으로 제공되지 않습니다. 해당되는 RPM이나 SRPM을 구해야 하며, 커널을 재구성하여 이를 사용해야 합니다.

---

- strace 명령을 사용하여 시스템 호출과 신호를 추적합니다. 이는 Solaris truss 명령과 유사합니다.
- Sun StorageTek QFS samtrace 명령은 추적 버퍼의 덤프를 생성합니다.
- Sun StorageTek QFS samexplorer 명령은 Sun StorageTek QFS 진단 보고 스크립트를 생성합니다.

---

주 - 추적 파일은 Solaris 클라이언트와 마찬가지로 Linux 클라이언트의 /var/opt/SUNWsamfs/trace 디렉토리에 있습니다.

---

## FAQ

Linux 클라이언트에 대한 다음 질문은 Solaris 플랫폼에서 Sun StorageTek QFS에 익숙한 사용자들이 자주 묻는 내용입니다.

- Q:** Linux 설치 스크립트에 음의 점수가 있으며 소프트웨어를 설치할 수 없다고 표시됩니다. 소프트웨어를 계속 설치할 수 있는 방법이 있습니까?
- A:** `-force-custom` 및 `-force-build` 설치 옵션을 시도할 수 있습니다. 그러나 모듈을 설치할 때 시스템 패닉이 발생할 수 있습니다. `spinlock` 디버깅과 같은 일부 커널 해킹 옵션이 활성화된 상태로 커널이 구축된 경우에는 특히 더 위험할 수 있습니다.
- Q:** Linux에서 `vmstat`, `iostat`, `top` 및 `truss`와 같은 명령을 사용할 수 있습니까?
- A:** 다수의 Linux 설치에 `vmstat`, `top` 및 `iostat` 명령이 있습니다. 설치되어 있지 않은 경우 `sysstat` 및 `procps RPM`을 사용하여 추가할 수 있습니다. Linux에서 `truss`에 해당하는 명령은 `ltrace` 및 `strace`입니다.
- Q:** Sun StorageTek QFS Linux 클라이언트에서 Sun StorageTek Traffic Manager를 사용할 수 있습니까?
- A:** 예. 먼저 Sun StorageTek Traffic Manager 설명서에 나와 있는 대로 다중 경로를 지원하는 사용자 정의 커널을 만듭니다. 그런 다음 Linux 클라이언트 소프트웨어를 설치합니다.
- Q:** Sun StorageTek QFS Linux 클라이언트에서 EFI(Extensible Firmware Interface) 레이블을 사용할 수 있습니까?
- A:** 대부분의 Linux 커널은 GPT(GUID Partition Table) 분할 영역이 있는 EFI 레이블을 지원하도록 구축되지 않았습니다. 따라서 EFI 레이블을 사용하려면 `CONFIG_EFI_PARTITION` 옵션 세트로 커널을 다시 구축해야 합니다. 사용자 정의 커널 구축에 대한 자세한 내용은 배포 설명서를 참조하십시오.
- Q:** Sun StorageTek QFS Linux 클라이언트 소프트웨어에서 LVM(Logical Volume Management), EVMS(Enterprise Volume Management System) 또는 Device Mapper와 같은 다른 Linux 볼륨 관리자를 사용할 수 있습니까?
- A:** EVMS가 있는 파일 시스템을 사용하려면 해당 파일 시스템용 FSIM(File System Interface Module)이 있어야 합니다. Sun StorageTek QFS 제품에 대한 FSIM이 없습니다. LVM을 사용하려면 `fdisk`에 표시되는 분할 영역 유형이 반드시 LVM(8e)이어야 합니다. Sun StorageTek QFS가 사용하는 분할 영역은 SunOS여야 합니다.
- Q:** 2TB보다 큰 파일 시스템을 사용할 수 있습니까?
- A:** 예. 그러나 `df`와 같은 파일 시스템 정보를 제공하는 일부 유틸리티가 Linux에서 실행될 때 잘못된 정보를 반환할 수 있습니다. 또한 NFS 또는 Samba와 함께 파일 시스템을 공유하는 경우 문제가 있을 수 있습니다.

**Q:** Linux 클라이언트에서 지원하는 마운트 옵션과 Solaris 클라이언트가 지원하는 옵션 사이에 차이점이 있습니까?

**A:** Linux 클라이언트에서 지원되지 않는 `samfs` 마운트 옵션이 많습니다. 두 가지 주의할 옵션은 `nosuid`과 `forcedirectio`입니다. Linux 클라이언트에서 지원되는 마운트 옵션의 전체 목록은 Sun StorageTek QFS Linux 클라이언트 안내서를 참조하십시오.

---

주 - `mdadm`(복수 장치 관리) 패키지는 Sun StorageTek QFS Linux 클라이언트의 경로 장애 조치에 사용되지 않아야 합니다. `mdadm` 패키지는 사용하는 장치에 대한 수퍼 블록을 작성합니다. 그 결과 `mdadm`은 Solaris가 해당 장치에 쓴 데이터를 손상시킬 수 있습니다. 또한 Solaris는 `mdadm`이 장치에 쓴 수퍼 블록을 손상시킬 수 있습니다.

---



# Sun StorageTek QFS 공유 파일 시스템의 마운트 옵션

Sun StorageTek QFS 공유 파일 시스템은 몇 가지 마운트 옵션과 함께 마운트할 수 있습니다. 이 부록에서는 역할에 따라 몇 가지 해당 옵션을 설명합니다.

mount(1M) 명령을 사용하고 /etc/vfstab 파일에 입력하거나 samfs.cmd 파일에 입력하여 대부분의 마운트 옵션을 지정할 수 있습니다. 예를 들어, 다음 /etc/vfstab 파일에는 Sun StorageTek QFS 공유 파일 시스템에 대한 mount(1M) 옵션이 포함되어 있습니다.

```
sharefs1 - /sfs samfs - no shared,mh_write
```

samu(1M) 운영자 유틸리티를 사용하여 일부 마운트 옵션을 동적으로 변경할 수 있습니다. 이러한 옵션에 대한 자세한 내용은 [253페이지의 "samu\(1M\) 운영자 유틸리티 사용"](#)을 참조하십시오.

다음 절에서는 Sun StorageTek QFS 공유 파일 시스템에서 사용할 수 있는 마운트 옵션에 대해 요약 설명합니다. 이러한 옵션에 대한 자세한 내용은 mount\_samfs(1M) 매뉴얼 페이지 또는 각 옵션 설명에서 언급한 상호 참조를 참조하십시오.

- [246페이지의 "백그라운드에서 마운트: bg 옵션"](#)
- [246페이지의 "파일 시스템 마운트 재시도: retry 옵션"](#)
- [246페이지의 "Sun StorageTek QFS 공유 파일 시스템 선언: shared 옵션"](#)
- [247페이지의 "할당 크기 조정: minallopsz=n 및 maxallopsz=n 옵션"](#)
- [247페이지의 "Sun StorageTek QFS 공유 파일 시스템에서 임대 사용: rdlease=n, wrlease=n 및 aplease=n 옵션"](#)
- [248페이지의 "다중 호스트 읽기 및 쓰기 활성화: mh\\_write 옵션"](#)
- [250페이지의 "최소 동시 스레드 수 설정: min\\_pool=n 옵션"](#)
- [250페이지의 "캐시된 속성 유지: meta\\_timeo=n 옵션"](#)
- [251페이지의 "스트라이프 할당 지정: stripe 옵션"](#)

- 251페이지의 "메타 데이터가 기록되는 빈도 지정: `sync_meta=n` 옵션"
- 252페이지의 "WORM 기능 활성화: `worm_capable` 및 `def_retention` 옵션"

---

## 백그라운드에서 마운트: `bg` 옵션

`bg` 마운트 옵션은 첫 마운트 작업이 실패할 경우, 그 다음부터의 마운트 시도는 백그라운드에서 수행되도록 지정합니다. 기본적으로 `bg`는 유효하지 않으며, 마운트 시도는 포어그라운드에서 계속 진행됩니다.



---

주의 – Sun StorageTek QFS 공유 파일 시스템을 Sun Cluster 노드에 마운트하거나 클러스터 외부에 공유 Sun StorageTek QFS 클라이언트가 있는 경우 이 옵션을 사용하지 마십시오.

---

---

## 파일 시스템 마운트 재시도: `retry` 옵션

`retry` 마운트 옵션은 시스템이 파일 시스템 마운트를 시도할 횟수를 지정합니다. 기본 값은 10000입니다.

---

## Sun StorageTek QFS 공유 파일 시스템 선언: `shared` 옵션

`shared` 마운트 옵션은 파일 시스템이 Sun StorageTek QFS 공유 파일 시스템이 되도록 선언합니다. 파일 시스템이 Sun StorageTek QFS 공유 파일 시스템으로 마운트되도록 하려면 이 옵션을 `/etc/vfstab` 파일에 지정해야 합니다. `samfs.cmd` 파일 또는 `mount(1M)` 명령에 이 옵션을 사용할 경우 오류가 발생하지는 않지만, 파일 시스템이 Sun StorageTek QFS 공유 파일 시스템으로 마운트되지 않습니다.

---

## 할당 크기 조정: `minallocsz=n` 및 `maxallocsz=n` 옵션

`mount(1M)` 명령에 대한 `-o minallocsz=n` 및 `-o maxallocsz=n` 옵션은 스페이스의 크기를 KB로 지정합니다. 이것은 최소 블록 할당 크기입니다. 파일이 증가하는 경우, 메타 데이터 서버는 추가 임대가 허용될 때 블록을 할당합니다. `-o minallocsz=n` 옵션을 사용하여 이 할당의 초기 크기를 지정할 수 있습니다. 메타 데이터 서버는 애플리케이션의 액세스 패턴에 따라 최대 `-o maxallocsz=n` 옵션 설정값까지(초과할 수는 없음) 블록 할당의 크기를 늘릴 수 있습니다.

`/etc/vfstab` 파일 또는 `samfs.cmd` 파일의 `mount(1M)` 명령줄에서 이 `mount(1M)` 옵션들을 지정할 수 있습니다.

---

## Sun StorageTek QFS 공유 파일 시스템에서 임대 사용: `rdlease=n`, `wrlease=n` 및 `aplease=n` 옵션

임대는 임대가 유효한 동안 파일에 대해 작업을 수행할 수 있는 공유 호스트 권한을 부여합니다. 메타 데이터 서버는 자신을 포함한 각 공유 호스트에 대한 임대를 수행합니다. 파일 작업을 계속 수행할 수 있도록 필요에 따라 임대를 갱신할 수 있습니다. 가능한 파일 작업은 다음과 같습니다.

- 읽기 임대는 기존 파일 데이터를 읽을 수 있도록 합니다.
- 쓰기 임대는 기존 파일 데이터를 덮어쓸 수 있도록 합니다.
- 추가 임대는 파일의 크기를 확장하고 새로 할당된 블록을 쓸 수 있도록 합니다.

공유 호스트는 필요한 만큼 임대를 계속해서 업데이트할 수 있습니다. 임대는 최종 사용자에게 투명합니다. 표 B-1은 각 임대 유형의 지속 시간을 지정할 수 있는 마운트 옵션을 표시합니다.

표 B-1 임대 관련 `mount(1M)` 옵션

옵션	작업
<code>-o rdlease=n</code>	이 옵션은 읽기 임대에 대한 최대 시간을 초 단위로 지정합니다.
<code>-o wrlease=n</code>	이 옵션은 쓰기 임대에 대한 최대 시간을 초 단위로 지정합니다.
<code>-o aplease=n</code>	이 옵션은 추가 임대에 대한 최대 시간을 초 단위로 지정합니다.

이러한 모든 세 가지 임대 옵션은  $15 \leq n \leq 600$  범위의  $n$ 으로 지정할 수 있습니다. 각 임대의 기본 시간은 30초입니다. 임대가 유효한 동안에는 파일을 자를 수 없습니다. 이러한 임대 설정에 대한 자세한 내용은 mount\_samfs(1M) 매뉴얼 페이지를 참조하십시오.

현재 메타 데이터 서버가 다운되어 메타 데이터 서버를 변경하는 경우 대체 메타 데이터 서버가 제어할 수 있으려면 모든 임대가 만료되어야 하므로 임대 시간을 변경 조치 시간에 추가해야 합니다.

짧은 임대 시간을 설정하면 임대가 만료된 후 갱신되어야 하므로 클라이언트 호스트와 메타 데이터 서버 사이에 더 많은 트래픽이 발생할 수 있습니다. Sun Cluster 환경에서의 임대 시간에 대해서는 Sun StorageTek QFS 설치 및 업그레이드 안내서를 참조하십시오.

---

## 다중 호스트 읽기 및 쓰기 활성화: mh\_write 옵션

기본적으로 Sun StorageTek QFS 공유 파일 시스템에서는 여러 호스트가 동시에 동일한 파일을 읽을 수 있으며, 이러한 파일에 쓰고 있는 호스트가 없다면 I/O는 모든 호스트에서 페이지될 수 있습니다. 한 번에 오직 하나의 호스트만 파일에 추가하거나 쓸 수 있습니다.

mh\_write 옵션은 여러 호스트에서의 동일한 파일에 대한 쓰기 액세스를 제어합니다. mh\_write가 메타 데이터 서버 호스트에 마운트 옵션으로 지정되면 Sun StorageTek QFS 공유 파일 시스템은 여러 호스트에서 동시에 동일한 파일에 대해 읽기와 쓰기를 수행할 수 있습니다. 메타 데이터 서버 호스트에 mh\_write가 지정되지 않으면 오직 하나의 호스트만 한 시점에서 파일에 쓸 수 있습니다.

기본적으로 mh\_write는 비활성화되고, 오직 하나의 호스트만 한 시점에서 파일에 대한 쓰기 액세스 권한을 갖습니다. 이러한 쓰기 권한 보유 기간은 wrlease 마운트 옵션 기간으로 결정됩니다. Sun StorageTek QFS 공유 파일 시스템이 mh\_write 옵션이 활성화되어 있는 메타 데이터 서버에 마운트되면 여러 호스트에서 동일한 파일에 대한 동시 읽기 및 쓰기가 발생할 수 있습니다.

표 B-2는 메타 데이터 서버에서 `mh_write` 옵션의 활성화 여부에 따라 여러 호스트에서의 파일 액세스가 어떻게 영향을 받는지 설명한 것입니다.

표 B-2 `mh_write` 옵션에 따른 파일 액세스

메타 데이터 서버에서 <code>mh_write</code> 가 활성화되지 않은 경우	메타 데이터 서버에서 <code>mh_write</code> 가 활성화된 경우
여러 호스트에서 읽기가 허용됩니다. 페이지된 I/O를 사용할 수 있습니다.	여러 호스트에서 읽기가 허용됩니다. 페이지된 I/O를 사용할 수 있습니다.
오직 하나의 호스트만 쓰기가 허용됩니다. 페이지된 I/O를 사용할 수 있습니다. 다른 모든 호스트는 대기합니다.	여러 호스트에서 읽기 및/또는 쓰기가 허용됩니다. 쓰기 작업 중인 호스트가 있는 경우 모든 I/O는 직접 이루어집니다.
오직 하나의 추가 호스트만 존재합니다. 다른 모든 호스트는 대기합니다.	오직 하나의 호스트만 추가할 수 있습니다. 다른 모든 호스트는 읽기 및/또는 쓰기가 가능합니다. 쓰기 작업 중인 호스트가 있는 경우 모든 I/O는 직접 이루어집니다.

`mh_write` 옵션은 잠금 작업을 변경하지 않습니다. 파일 잠금은 `mh_write`를 사용하든 안 하든 동일하게 작동합니다. `mh_write` 옵션의 효과는 다음과 같습니다.

- `mh_write`를 사용 중일 때 모든 호스트가 동시에 동일한 파일을 읽고 쓸 수 있습니다.
- `mh_write`를 사용하지 않을 경우 하나의 호스트만 주어진 시간 간격 중에 주어진 파일에 쓸 수 있으며 해당 시간 간격 중에 파일로부터 읽을 수 있는 호스트는 없습니다.

Sun StorageTek QFS 공유 파일 시스템은 호스트 간에 일관성을 유지합니다. 호스트가 처음 읽기 또는 쓰기 시스템 호출을 실행할 때 약간의 시간 동안 파일을 읽거나 쓰는 것을 허용하는 임대가 됩니다. 해당 임대가 있으면 `mh_write` 없이 다른 호스트가 파일을 액세스하지 못합니다. 특히 임대는 얻을 수 있는 시스템 호출 기간 보다 길게 지속될 수 있습니다.

`mh_write`를 사용하지 않을 때 Sun StorageTek QFS 공유 파일 시스템은 데이터 읽기 및 쓰기에 POSIX에 가까운 작업을 제공합니다. 그러나 메타 데이터의 경우 액세스 시간 변경은 다른 호스트에 직접 나타나지 않을 수도 있습니다. 파일에 대한 변경 사항은 쓰기 임대의 마지막에서 디스크에 더해지며 읽기 임대를 한 경우 시스템은 새로 기록된 데이터를 볼 수 있도록 모든 스테일 캐시 페이지를 무효화합니다.

`mh_write`를 사용 중일 때 작업은 덜 일관적일 수도 있습니다. 동시 읽기 및 쓰기가 있는 경우 Sun StorageTek QFS 공유 파일 시스템은 파일에 액세스하는 모든 호스트들을 직접 I/O 모드로 전환합니다. 이것은 페이지가 정렬된 I/O가 다른 호스트에 즉시 보여져야 함을 의미합니다. 그러나 이것을 막는 정상적인 임대 체계를 사용할 수 없기 때문에 페이지가 정렬되지 않은 I/O는 볼 수 있거나 파일에 기록되는 스테일 데이터가 될 수 있습니다.

다중 호스트가 동일한 파일에 동시에 쓰야 하고 애플리케이션이 페이지가 정렬된 I/O를 수행할 때에만 `mh_write` 옵션을 지정해야 합니다. 다른 경우, 호스트 간의 통합을 위해 `flock()(mh_write와 작동)`를 사용하는 것은 일관성을 보장하지 않기 때문에 데이터가 일관되지 않을 위험이 있습니다.

`mh_write`에 대한 자세한 내용은 `mount_samfs(1M)` 매뉴얼 페이지를 참조하십시오.

---

## 최소 동시 스레드 수 설정: `min_pool=n` 옵션

`min_pool=n` 마운트 옵션은 Sun StorageTek QFS 공유 파일 시스템에 대한 최소 동시 스레드 수를 설정합니다. 기본적으로 Solaris 시스템은 `min_pool=64`이며 Linux 클라이언트는 8입니다. 이는 기본 설정값을 사용하면 Solaris에는 최소한 64개의 활성 스레드가 있으며 Linux에는 8개가 있다는 의미입니다. Sun StorageTek QFS 공유 파일 시스템의 활동에 따라 `min_pool=n` 마운트 옵션을 8과 2048 사이의 임의의 값으로 조정할 수 있습니다.

`min_pool` 마운트 옵션은 `samfs.cmd` 파일에서 설정해야 합니다. `/etc/vfstab` 파일이나 명령줄에서 설정하면 무시됩니다.

---

주 - `min_pool` 마운트 옵션은 이전 `nstreams` 마운트 옵션을 대체합니다. 이 소프트웨어의 4U6 버전에서 `nstreams` 옵션은 무시됩니다. 이 소프트웨어의 버전 5에서는 완전히 제거될 예정입니다.

---

---

## 캐시된 속성 유지: `meta_timeo=n` 옵션

`meta_timeo=n` 마운트 옵션은 메타 데이터 정보에 대한 확인 시점 사이에 시스템이 대기하는 시간을 결정합니다. 기본적으로 시스템은 3초마다 메타 데이터 정보를 새로 고칩니다. 예를 들어, 여러 개의 파일이 새로 생성된 Sun StorageTek QFS 공유 파일 시스템에 입력한 `ls(1)` 명령은 3초가 경과하기 전에는 모든 파일에 대한 정보를 반환하지 못할 수 있습니다. `n`에 대해  $0 \leq n \leq 60$  사이의 값을 지정할 수 있습니다.

---

## 스트라이프 할당 지정: stripe 옵션

기본적으로 Sun StorageTek QFS 공유 파일 시스템의 데이터 파일은 라운드 로빈 파일 할당 방식을 사용하여 할당됩니다. 파일 데이터가 디스크에 걸쳐 스트라이프되도록 지정하려면 메타 데이터 호스트 및 모든 가능한 메타 데이터 호스트에서 stripe 마운트 옵션을 지정합니다. 기본적으로 공유되지 않은 파일 시스템은 스트라이프 방식을 사용하여 파일 데이터를 할당합니다.

라운드 로빈 할당에서 파일은 라운드 로빈 방식으로 각 슬라이스 또는 스트라이프 그룹에 만들어집니다. 이 방식은 한 파일에 대해 슬라이스 또는 스트라이프 그룹 속도의 최대 성능을 보장합니다. 파일 할당 방식에 대한 자세한 내용은 7페이지의 "디자인의 기본"을 참조하십시오.

---

## 메타 데이터가 기록되는 빈도 지정: sync\_meta=*n* 옵션

sync\_meta=*n* 옵션을 sync\_meta=1 또는 sync\_meta=0으로 설정할 수 있습니다.

기본값인 sync\_meta=1의 경우, Sun StorageTek QFS 공유 파일 시스템은 메타 데이터가 변경될 때마다 디스크에 파일 메타 데이터를 씁니다. 이렇게 되면 데이터 성능은 낮아지지만 데이터의 일관성이 보장됩니다. 이것은 메타 데이터 서버를 변경하려는 경우에 적용되어야 하는 설정입니다.

sync\_meta=0으로 설정하는 경우 Sun StorageTek QFS 공유 파일 시스템은 메타 데이터를 버퍼에 쓴 후에 디스크에 씁니다. 이렇게 지연되는 쓰기는 더 높은 성능을 제공하지만 갑작스런 시스템 중단이 발생한 후에는 데이터 일관성이 떨어집니다.

---

## WORM 기능 활성화: worm\_capable 및 def\_retention 옵션

선택 사항인 WORM 패키지를 사용하는 경우 worm\_capable 마운트 옵션을 사용하여 파일 시스템이 WORM 파일을 지원하도록 할 수 있습니다. def\_retention 마운트 옵션을 사용하면 def\_retention=*MyNdOhPm* 형식으로 기본 보유 시간을 설정할 수 있습니다.

이 형식에서 *M*, *N*, *O* 및 *P*는 음이 아닌 정수이며 *y*, *d*, *h* 및 *m*은 각각 연도, 일, 시간 및 분을 나타냅니다. 위 단위의 모든 조합을 사용할 수 있습니다. 예를 들어 1y5d4h3m은 1년, 5일, 4시간, 3분을 나타내고, 30d8h는 30일, 8시간을 나타내며, 300m은 300분을 나타냅니다. 이 형식은 이전 소프트웨어 버전의 형식과 역호환됩니다. 여기서 보유 기간은 분 단위로 지정됩니다.

WORM 기능에 대한 더 자세한 내용은 [186페이지의 "WORM-FS 파일 시스템 구성"](#)을 참조하십시오.

# samu(1M) 운영자 유틸리티 사용

---

이 장은 samu(1M)를 사용하여 Sun StorageTek QFS 환경 내에 구성된 장치를 제어하는 방법을 설명합니다. 많은 samu(1M) 디스플레이는 저장소 및 아카이브 관리 체계를 사용하는 사이트에 대해서만 유용합니다. Sun StorageTek QFS 환경에서 samu(1M)를 사용하는 경우에는 일부 디스플레이가 적용되지 않습니다.

본 장에는 다음 절이 포함되어 있습니다.

- 253페이지의 "개요"
- 256페이지의 "운영자 디스플레이"
- 306페이지의 "운영자 명령"

samu(1M) 내에서 수행할 수 있는 작업은 samcmd(1M) 명령을 사용하여 수행할 수도 있습니다. samcmd(1M)에 대한 자세한 내용은 samcmd(1M) 매뉴얼 페이지를 참조하십시오.

---

## 개요

samu(1M) 운영자 유틸리티에는 80개의 글자 폭으로 최소 24행을 표시하는 디스플레이 터미널이 필요합니다. 이 유틸리티에는 다음과 같은 기능이 포함됩니다.

- Sun StorageTek QFS 및 SAM-QFS 장치 및 파일 시스템 작업을 모니터링할 수 있는 디스플레이
- 디스플레이 선택, 디스플레이 옵션 설정, 장치 액세스와 작업 제어 및 디스플레이 창의 스냅샷 만들기 등을 할 수 있는 명령
- 실행 중인 Sun StorageTek QFS 또는 SAM-QFS 파일 시스템을 조정할 수 있는 명령

이 장에 나와있는 디스플레이 창은 대표적인 예입니다. 터미널에 표시되는 정보의 정확한 형식과 양은 터미널 모델과 Sun StorageTek QFS 환경에 구성된 장치에 따라 달라질 수 있습니다.

다음 절에서는 samu(1M)의 시작 및 중지, 유틸리티와 상호 작용, 도움말 창 액세스 및 운영자 디스플레이 보기 등의 작업 방법에 대해 설명합니다.

## ▼ samu(1M)를 호출하려면

- samu(1M)를 시작하려면 명령줄에서 samu(1M)를 입력합니다.

```
# samu
```

시스템은 samu(1M)를 시작하고 도움말 디스플레이를 표시합니다. 이것이 기본 초기 디스플레이입니다. 다른 samu(1M) 디스플레이를 보려면 [254페이지의 "samu\(1M\) 화면 표시"](#)에 있는 단계를 수행하십시오.

samu(1M) 유틸리티를 사용하면 해당 초기 디스플레이를 선택할 수 있습니다. samu(1M) 명령줄 옵션에 대한 자세한 내용은 samu(1M) 매뉴얼 페이지를 참조하십시오.

---

주 - samu(1M)은 vi(1) 편집기와 마찬가지로 curses(3CURSES) 루틴을 기반으로 합니다. samu(1M) 실행에 문제가 있는 경우, 터미널 유형이 올바르게 정의되어 있는지 확인하십시오.

---

## ▼ samu(1M) 화면 표시

samu(1M) 명령은 다른 samu(1M) 화면 표시를 위해 해당 명령의 옵션을 허용합니다.

1. 콜론(:)을 입력하여 samu(1M) 프롬프트를 불러 냅니다.  
콜론을 입력하고 나면 왼쪽 하단에 다음이 나타납니다.

```
Command:
```

2. 보려는 디스플레이에 해당하는 문자를 입력하고 **Return**을 누릅니다.

예를 들어 v 디스플레이를 보려면 **v**를 입력하고 Command: 프롬프트 뒤에 **Return**을 누릅니다.

입력할 전체 문자 목록 및 대응하는 디스플레이에 대해서는 [267페이지의 "\(h\) - 도움말 디스플레이"](#)를 참조하십시오.

## ▼ samu(1M) 중지

● samu(1M)를 종료하려면 다음 중 하나를 입력합니다.

- q
- :q

samu(1M) 운영자 유틸리티가 있으며, 다시 명령 셸로 돌아갑니다.

## samu(1M)와 상호 작용

samu(1M)와 상호 작용하는 것은 앞뒤로 페이지 이동, 명령 입력, 화면 새로 고침 및 유틸리티 종료 측면에서 UNIX vi(1) 편집기와 상호 작용하는 것과 유사합니다.

각 디스플레이에는 이 장에 해당하는 고유 절이 있으며 각 디스플레이 절은 해당 디스플레이에서 검색하는 데 사용할 수 있는 제어 키 순서를 표시합니다. samu(1M) 메뉴얼 페이지에는 제어 키 탐색 순서가 요약되어 있습니다.

디스플레이 창의 마지막 행은 명령 및 디스플레이 오류 메시지를 표시합니다. 명령 오류가 발생하면 다음 운영자 작업이 있을 때까지 자동 디스플레이 새로 고침이 중지됩니다.

## 장치 입력

Sun StorageTek QFS 환경에 포함된 각 장치에는 mcf 파일의 Equipment Ordinal (예: 10)이 할당됩니다. 여러 samu(1M) 명령은 Equipment Ordinal을 사용하여 특정 장치를 참조합니다.

예제. :off 명령의 구문은 다음과 같습니다.

```
:off eq
```

eq에는 어드레스할 장치의 장비 서수를 입력합니다.

## 온라인 도움말 가져오기

samu(1M)를 시작하면 시스템에서 자동으로 첫 번째 도움말 화면을 표시합니다.

도움말(h) 디스플레이에 대한 자세한 정보는 [267페이지의 "\(h\) - 도움말 디스플레이"](#)를 참조하십시오.

### ▼ 디스플레이 화면에서 온라인 도움말 액세스하기

- :h를 입력합니다.

한 화면에서 다음 화면으로 이동하려면 다음과 같은 조합 키를 입력합니다.

- Ctrl-f를 눌러 한 페이지 앞으로 이동합니다.
- Ctrl-b를 눌러 한 페이지 뒤로, 즉 이전 페이지로 이동합니다.

h 키를 누르면 언제든지 도움말 디스플레이로 돌아갈 수 있습니다.

---

## 운영자 디스플레이

samu(1M) 운영자 디스플레이를 보려면 각 디스플레이에 해당하는 키를 눌러야 합니다. 소문자 키 a ~ w는 작동 정보를 표시합니다.

---

주 - 대문자 samu(1M) 디스플레이(C, D, F, I, J, K, L, M, N, P, R, S, T 및 U)는 기술 지원 직원의 도움을 받아야만 고객 사이트에서 사용할 수 있도록 설계되었습니다.

이 장에서는 이러한 대문자 디스플레이에 대해 소문자 디스플레이 만큼 자세하게 설명하지 않습니다.

---

디스플레이가 화면 영역을 넘는 경우에는 more라는 단어가 화면 디스플레이 하단에 나타나 디스플레이에 추가 정보가 있음을 나타냅니다. Ctrl-f를 사용하여 다음 페이지로 이동하여 추가 내용을 볼 수 있습니다.

**코드 예 C-1**을 보면 more라는 단어가 있습니다. 이는 후속 화면에 추가 정보가 있음을 나타냅니다.

**코드 예 C-1**            추가 텍스트가 있음을 나타내는 samu(1M) 화면

```
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)에서는 장치 및 해당 장비 서수를 입력하라는 메시지가 나타납니다. 구성 디스플레이(c)는 모든 제거 가능한 매체 장치에 대한 장비 서수를 표시합니다. 모든 디스플레이를 제어하려면 디스플레이에 대해 나열된 컨트롤 키를 사용하십시오.

다음 절에서는 알파벳 순으로 운영자 디스플레이에 대해 설명합니다. 예제뿐만 아니라 필요한 경우, 표시되는 필드를 설명하는 표와 함께 디스플레이를 제시하였습니다.

---

주 – Sun StorageTek SAM 소프트웨어 없이 Sun StorageTek QFS 소프트웨어가 설치된 경우 일부 연산자 디스플레이만 이용할 수 있습니다.

---

## (a) – 아카이버 상태 디스플레이

a 디스플레이는 아카이버 상태를 표시합니다.

보려는 것이 무엇인지에 따라 이 디스플레이를 다음과 같이 다르게 호출할 수 있습니다.

- 파일 시스템을 기본으로 아카이버 상태를 표시하는 아카이버 상태 요약 표시하려면 다음 형식을 사용하여 해당 명령을 입력하십시오.

```
Command:a
```

- 특정 파일 시스템에 대해 아카이브 정보를 표시하려면 다음 형식을 사용하여 해당 명령을 입력하십시오.

```
Command:a filesystem
```

*filesystem*에 파일 시스템의 이름을 지정합니다.

## 탐색

표 C-1은 a 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

표 C-1 a 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	이전 파일 시스템
Ctrl-f	다음 파일 시스템
Ctrl-d	<i>arcopies</i> 페이지 뒤로(하단 부분)
Ctrl-u	<i>arcopies</i> 페이지 앞으로(하단 부분)

표 C-2는 :a filesystem 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

표 C-2 :a filesystem 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	이전 파일 시스템
Ctrl-f	다음 파일 시스템

## 예제 디스플레이

코드 예 C-2는 요약 디스플레이의 단일 파일 시스템에 대한 작업과 통계를 보여줍니다.

코드 예 C-2 samu(1M) a 디스플레이

```
Archiver status                samu 4.6 07:44:02 August 8 2005
sam-archiverd:  Waiting for resources
sam-arfind:  samfs1 mounted at /sam1
Waiting until 2005-05-08 07:54:02 to scan .inodes
sam-arfind:  samfs2 mounted at /sam2
Waiting until 2005-05-08 07:52:57 to scan .inodes
sam-arfind:  qfs1 mounted at /qfs1
Waiting until 2005-05-08 07:44:33 to scan .inodes
sam-arfind:  qfs2 mounted at /qfs2
Waiting until 2005-05-08 07:53:21 to scan .inodes
sam-arfind:  qfs3 mounted at /qfs3
Waiting until 2005-05-08 07:44:11 to scan .inodes
```

```

sam-arfind: qfs4 mounted at /qfs4
Waiting until 2005-05-08 07:53:35 to scan .inodes

sam-arfind: shareqfs1 mounted at /shareqfs1
Shared file system client. Cannot archive.

sam-arfind: shareqfs2 mounted at /shareqfs2
Shared file system client. Cannot archive.

sam-arcopy: qfs4.arset5.1.83 dt.DAT001
Waiting for volume dt.DAT001

```

## 필드 설명

표 C-3에서는 상세 디스플레이의 필드를 보여줍니다.

표 C-3 samu(1M) a 디스플레이 필드 설명

필드	설명
samfs1 mounted at	마운트 지점
regular files	일반 파일의 수와 전체 크기
offline files	오프라인 파일의 수와 전체 크기
archdone files	archdone 파일의 수와 크기. 아카이버에서 프로세스를 완료하여 archdone 파일에 대한 추가 프로세스를 수행할 수 없음을 나타냅니다. archdone로 표시된 파일은 아카이브용으로 처리되었지만 반드시 아카이브된 것은 아닙니다.
copy1	아카이브 복사본 1의 파일의 수와 총 크기
copy2	아카이브 복사본 2의 파일의 수와 총 크기
copy3	아카이브 복사본 3의 파일의 수와 총 크기
copy4	아카이브 복사본 4의 파일의 수와 총 크기
디렉토리	디렉토리의 수와 총 크기
sleeping until	아카이버가 다시 실행되는 시기를 나타냅니다.

## (c) - 장치 구성 디스플레이

c 디스플레이는 구성의 연결을 표시합니다. 모든 장치 이름 및 장비 서수를 나열합니다. 장치 구성 디스플레이를 호출하려면 다음 형식을 사용하여 명령을 입력하십시오.

```
Command: c
```

### 탐색

표 C-4는 이 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

표 C-4 c 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	한 페이지 뒤로
Ctrl-d	반 페이지 앞으로
Ctrl-f	한 페이지 앞으로
Ctrl-u	반 페이지 뒤로

### 예제 디스플레이

코드 예 C-3에서는 장치 구성 디스플레이를 보여줍니다.

코드 예 C-3 samu(1M) c 디스플레이

```
Device configuration:          samu    4.6 07:48:11 Sept 8 2006
ty  eq state  device_name                fs family_set
sk  100 on    /etc/opt/SUNWsamfs/dcstkconf  100 dcL700
tp  120 off   /dev/rmt/1cbn                 100 dcL700
sg  130 on    /dev/rmt/4cbn                 100 dcL700
sg  140 on    /dev/rmt/5cbn                 100 dcL700
tp  150 off   /dev/rmt/3cbn                 100 dcL700
hy  151 on    historian                       151
```

## 필드 설명

표 C-5에서는 이 디스플레이에 대한 필드 설명을 보여줍니다.

표 C-5 samu(1M) c 디스플레이 필드 설명

필드	설명
ty	장치 유형
eq	장치의 장비 서수
state	장치의 현재 작동 상태 올바른 장치 상태는 다음과 같습니다. <ul style="list-style-type: none"><li>• on - 장치에 액세스할 수 있습니다.</li><li>• ro - 장치에 읽기 전용 액세스만 할 수 있습니다.</li><li>• off - 장치에 액세스할 수 없습니다.</li><li>• down - 장치에 유지 보수 액세스만 할 수 있습니다.</li><li>• idle - 장치에 새로운 연결을 할 수 없습니다. 진행 중인 작업은 완료될 때까지 계속됩니다.</li><li>• nalloc-nalloc 플래그가 설정되어 있어 해당 장치에 추가로 할당되지 않도록 합니다. 자세한 내용은 27페이지의 "논리 장치 번호(LUN)당 할당 제어"를 참조하십시오.</li></ul>
device_name	장치 경로
fs	패밀리 세트 장비 서수
family_set	장치가 속한 저장 패밀리 세트 또는 라이브러리의 이름

## (C) - 메모리 디스플레이

c 디스플레이는 지정된 메모리 주소의 내용을 표시합니다. 주소에 내용을 표시하려면 16진수로 주소를 입력하십시오.

이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command: c hex-address
```

hex\_address에 대해 메모리 위치의 주소를 16진수로 지정합니다. 예를 들어, 다음과 같습니다.

```
Command: c 0x1044a998
```

이 디스플레이는 디버그에 대해 구성되었으며, Sun Microsystems 지원 직원의 도움을 받아야 사용할 수 있습니다.

## 예제 디스플레이

코드 예 C-4는 메모리 디스플레이를 표시합니다. 결과는 이 매뉴얼에 포함될 수 있도록 일부가 생략되었습니다.

코드 예 C-4                    samu(1M) C 디스플레이

```
Memory base: 0x1234567                    samu 4.6 07:52:25 Sept 8 2006
00000000 80921000 137ff801 edd05e21 40853420 .....x.mP^!@.4
00000010 00a00420 018528b0 01a604e0 02840640 . . .(0.&.`...@
00000020 02d030a1 a0853420 0080a0a0 100a6fff .P0! .4 . . .o.
00000030 f6921000 13c65e23 582d0000 0ba01020 v....F^#X-... .
00000040 00c45e20 48c608e0 2fd05e21 40920080 .D^ HF.`/P^!@...
00000050 037ff801 fa941000 16c45e20 48a600a0 ..x.z....D^ H&.
00000060 80921000 137ff801 d5d05e21 40853420 .....x.UP^!@.4
00000070 00a00420 018528b0 01a604e0 02840640 . . .(0.&.`...@
00000080 02d030a1 c0853420 0080a0a0 100a6fff .P0!@.4 . . .o.
00000090 f6921000 13c65e23 58a01020 00c45e20 v....F^#X . .D^
000000a0 48c608e0 2fd05e21 40920080 037ff801 HF.`/P^!@....x.
000000b0 e39405a2 00c45e20 48a600a0 80921000 c..".D^ H&. ....
000000c0 137ff801 bed05e21 40853420 00a00420 ..x.>P^!@.4 . .
000000d0 018528b0 01a604e0 02840640 02d030a1 ..(0.&.`...@.P0!
000000e0 e0853420 0080a0a0 100a6fff f6921000 `.4 . . .o.v...
000000f0 13c65e23 58a01020 00c45e20 48c608e0 .F^#X . .D^ HF.`
```

## (d) - 데몬 추적 컨트롤 디스플레이

d 디스플레이에는 defaults.conf 파일에 지정된 방식으로 추적되는 이벤트가 표시 됩니다. 추적 파일 활성화에 대한 자세한 내용은 defaults.conf(4) 매뉴얼 페이지를 참조하십시오.

이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command:d
```

## 예제 디스플레이

코드 예 C-5에서는 추적 파일 정보를 보여줍니다. 여기에는 추적되는 데몬에 대한 정보, 추적 파일의 경로, 추적되는 이벤트 및 추적 파일의 크기와 나이에 대한 정보가 포함됩니다.

코드 예 C-5                    samu(1M) d 디스플레이

```
Daemon trace controls                    samu 4.6 07:56:38 Sept 8 2006
sam-amld                    /var/opt/SUNWsamfs/trace/sam-amld
                             cust err fatal misc proc debug date
                             size    0    age 0
sam-archiverd               /var/opt/SUNWsamfs/trace/sam-archiverd
                             cust err fatal misc proc debug date
                             size    0    age 0
sam-catserverd              /var/opt/SUNWsamfs/trace/sam-catserverd
                             cust err fatal misc proc debug date
                             size    0    age 0
sam-fsd                     /var/opt/SUNWsamfs/trace/sam-fsd
                             cust err fatal misc proc debug date
                             size    0    age 0
sam-rftd                    /var/opt/SUNWsamfs/trace/sam-rftd
                             cust err fatal misc proc debug date
                             size    0    age 0
sam-recycler                /var/opt/SUNWsamfs/trace/sam-recycler
                             cust err fatal misc proc debug date
                             size    0    age 0
sam-sharefsd                /var/opt/SUNWsamfs/trace/sam-sharefsd
                             cust err fatal misc proc debug date
                             size    0    age 0
sam-stagerd                 /var/opt/SUNWsamfs/trace/sam-stagerd
                             cust err fatal misc proc debug date
                             size    0    age 0
sam-serverd                 /var/opt/SUNWsamfs/trace/sam-serverd
                             cust err fatal misc proc debug date
                             size    0    age 0
sam-clientd                 /var/opt/SUNWsamfs/trace/sam-clientd
                             cust err fatal misc proc debug date
                             size    0    age 0
sam-mgmt                    /var/opt/SUNWsamfs/trace/sam-mgmt
                             cust err fatal misc proc debug date
                             size    0    age 0
```

## (D) - 디스크 볼륨 사전

D 디스플레이는 디스크 볼륨 사전을 표시하는데, 이는 `diskvols.conf` 파일에 정의된 디스크 아카이브를 위한 디스크 매체를 추적합니다. 사전에는 용량, 남은 공간 및 VSN의 상태를 표시하는 플래그를 포함하여 각 VSN에 대한 정보가 들어있습니다. 플래그에는 *unavailable*, *read only* 및 *bad media*가 포함됩니다.

이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command: D
```

### 예제 디스플레이

코드 예 C-6에서는 장치 구성 디스플레이를 보여줍니다.

코드 예 C-6            samu(1M) D 디스플레이

```
Disk volume dictionary samu      4.6 07:48:11 May 8 2007

header
version 460

volumes
magic 340322 version 9 nkeys 2 ndata 2
index  spacecapacityusedflagsvolume
   0  128824115201288778547210291200-----disk01
   1   6443827200 6443892736   70656-----disk02
clients
magic 340322 version 9 nkeys 1 ndata 1
```

### 플래그

표 C-6은 D 디스플레이에 대한 플래그를 표시합니다.

표 C-6            samu(1M) D 디스플레이의 플래그 필드

필드	설명
l----	볼륨에 레이블이 작성되었고 <code>seqnum</code> 파일이 작성되었습니다. 이 값은 소프트웨어가 새 <code>seqnum</code> 파일을 작성하지 못하도록 관리자에 의해 설정됩니다.
-r---	볼륨이 원격 호스트에 정의됩니다.

표 C-6 samu(1M) D 디스플레이의 플래그 필드(계속)

필드	설명
--U--	볼륨이 사용 불가능합니다.
---R-	볼륨이 읽기 전용입니다.
----E	매체 오류. 소프트웨어가 디스크 아카이브 사전에서 쓰기 오류를 감지할 때 설정됩니다.

diskvols samu(1M) 명령을 사용하여 디스크 볼륨 사전 플래그를 설정하거나 지울 수 있습니다. [316페이지](#)의 ":diskvols volume [+flag | -flag] 명령"을 참조하십시오.

## (f) - 파일 시스템 디스플레이

f 디스플레이에는 Sun StorageTek QFS 파일 시스템의 구성 요소가 표시됩니다.

이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command: f
```

## 예제 디스플레이

[코드 예 C-7](#)에서는 파일 시스템 디스플레이를 보여줍니다.

코드 예 C-7 samu(1M) f 디스플레이

```
File systems                                     samu 4.6 08:11:24 Sept 8 2006
```

ty	eq	state	device_name	status	high	low	mountpoint	server
ms	10	on	samfs1	m----	2----	d	90% 70%	/sam1
md	11	on	/dev/dsk/c5t8d0s3					
md	12	on	/dev/dsk/c5t8d0s4					
md	13	on	/dev/dsk/c5t8d0s5					
md	14	on	/dev/dsk/c5t8d0s6					
md	15	on	/dev/dsk/c5t8d0s7					
ms	20	on	samfs2	m----	2----	d	90% 70%	/sam2
md	21	on	/dev/dsk/c5t9d0s3					
md	22	on	/dev/dsk/c5t9d0s4					
md	23	on	/dev/dsk/c5t9d0s5					
md	24	on	/dev/dsk/c5t9d0s6					
md	25	on	/dev/dsk/c5t9d0s7					
ma	30	on	qfs1	m----	2----	d	90% 70%	/qfs1
mm	31	on	/dev/dsk/c5t10d0s0					
md	32	on	/dev/dsk/c5t10d0s1					

ma	40	on	qfs2	m----2----d	90%	70%	/qfs2
mm	41	on	/dev/dsk/c5t11d0s0				
md	42	on	/dev/dsk/c5t11d0s1				
ma	50	on	qfs3	m----2---r-	90%	70%	/qfs3
mm	51	on	/dev/dsk/c5t12d0s0				
mr	52	on	/dev/dsk/c5t12d0s1				
ma	60	on	qfs4	m----2---r-	90%	70%	/qfs4
mm	61	on	/dev/dsk/c5t13d0s0				
mr	62	on	/dev/dsk/c5t13d0s1				
ma	100	on	shareqfs1	m----2c--r-	80%	70%	/shareqfs1 spade
mm	101	on	/dev/dsk/c6t50020F2300004655d0s0				
mr	102	on	/dev/dsk/c6t50020F2300004655d0s1				
ma	110	on	shareqfs2	m----2c--r-	80%	70%	/shareqfs2 spade
mm	111	on	/dev/dsk/c6t50020F2300004655d0s6				
mr	112	on	/dev/dsk/c6t50020F2300004655d0s7				

## 필드 설명

표 C-7에서는 이 디스플레이에 대한 필드 설명을 보여줍니다.

표 C-7 samu(1M) f 디스플레이 필드 설명

필드	설명
ty	장치 유형
eq	장치의 장비 서수
state	장치의 현재 작동 상태 올바른 장치 상태는 다음과 같습니다. <ul style="list-style-type: none"> <li>• on - 장치에 액세스할 수 있습니다.</li> <li>• ro - 장치에 읽기 전용 액세스만 할 수 있습니다.</li> <li>• off - 장치에 액세스할 수 없습니다.</li> <li>• down - 장치에 유지 보수 액세스만 할 수 있습니다.</li> <li>• idle - 장치에 새로운 작업을 할 수 없습니다. 진행 중인 작업은 완료될 때까지 계속됩니다.</li> <li>• nalloc-nalloc 플래그가 설정되어 있어 해당 장치에 추가로 할당되지 않도록 합니다. 자세한 내용은 27페이지의 "논리 장치 번호(LUN)당 할당 제어"를 참조하십시오.</li> </ul>
device_name	파일 시스템의 이름 또는 장치의 경로
status	장치 상태. 상태 코드에 대한 설명은 301페이지의 "운영자 디스플레이 상태 코드"를 참조하십시오.
high	상한 디스크 사용률 임계값 퍼센트

표 C-7 samu(1M) f 디스플레이 필드 설명(계속)

필드	설명
low	하한 디스크 사용률 임계값 퍼센트
mountpoint	파일 시스템의 마운트 지점
서버	파일 시스템이 마운트되는 호스트 시스템의 이름

## (F) - 광 디스크 레이블 디스플레이

F 디스플레이에는 광 디스크의 레이블이 표시됩니다.

이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command: F
```

이 디스플레이는 디버그에 대해 구성되었으며, Sun Microsystems 지원 직원의 도움을 받아야 사용할 수 있습니다.

## (h) - 도움말 디스플레이

h 디스플레이에는 사용 가능한 samu(1M) 디스플레이의 요약이 표시됩니다. 기본적으로 이것은 명령줄에 samu(1M) 명령을 입력할 때 시스템이 표시하는 첫 번째 디스플레이입니다.

이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command: h
```

## 탐색

표 C-8은 이 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

표 C-8 h 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	한 페이지 뒤로
Ctrl-d	한 페이지 앞으로(상단 부분)

표 C-8 h 디스플레이에 대한 컨트롤 키(계속)

키	기능
Ctrl-f	한 페이지 앞으로
Ctrl-u	한 페이지 뒤로(상단 부분)
Ctrl-k	토글 경로 디스플레이

## 예제 디스플레이

도움말 화면은 여러 페이지로 구성되지만, 이 매뉴얼에서는 첫 번째 페이지만을 보여줍니다. 나머지 도움말 화면에서는 samu(1M) 명령을 보여줍니다.

코드 예 C-8은 Sun StorageTek SAM 파일 시스템에 대한 초기 도움말 화면을 표시합니다. Sun StorageTek QFS 파일 시스템에서 초기 도움말 화면에 모든 디스플레이가 나타나지는 않습니다. 예를 들어 Sun StorageTek QFS 시스템을 실행 중인 경우 제거 가능한 매체 디스플레이를 사용할 수 없습니다. SAM-QFS 구성을 사용 중인 경우 코드 예 C-8에 표시된 대로 도움말 화면이 나타납니다.

코드 예 C-8 Sun StorageTek SAM 시스템에 대한 samu(1M) 초기 도움말 화면

```

Help informationpage 1/15samu 4.6 08:18:13 Sept 8 2006
Displays:
  a  Archiver statuswPending stage queue
  c  Device configurationCMemory
  d  Daemon trace controlsDDisk volume dictionary
  f  File systemsFOptical disk label
  h  Help informationIInode
  l  Usage informationJPreview shared memory
  m  Mass storage statusKKernel statistics
  n  Staging statusLShared memory tables
  o  Optical disk statusMShared memory
  p  Removable media load requestsNFile system parameters
  r  Removable mediaPActive Services
  s  Device statusRSAM-Remote
  t  Tape drive statusSSector data
  u  Staging queueTSCSI sense data
  v  Robot catalogUDevice table

more (ctrl-f)

```

## (I) - Inode 디스플레이

I 디스플레이는 `inode`의 내용을 표시합니다.

보려는 것이 무엇인지에 따라 이 디스플레이를 다음과 같이 다르게 호출할 수 있습니다.

- 전체 파일 시스템에 대해 `inode`를 표시하려면 다음 형식을 사용하여 해당 명령을 입력하십시오.

```
Command: I filesystem
```

`filesystem`에 파일 시스템의 이름을 지정합니다.

- 특정 `inode`를 표시하려면 다음 형식을 사용하여 해당 명령을 입력하십시오.

```
Command: I inode-number
```

`inode-number`에 대해 16진수 또는 10진수로 `inode` 번호를 지정합니다.

### 탐색

표 C-9는 이 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

표 C-9 I 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	이전 <code>inode</code>
Ctrl-f	다음 <code>inode</code>
Ctrl-k	고급 디스플레이 형식

이 디스플레이는 디버그에 대해 구성되었으며, Sun Microsystems 지원 직원의 도움을 받아야 사용할 수 있습니다.

## 예제 디스플레이

코드 예 C-9는 inode 디스플레이를 표시합니다.

코드 예 C-9 samu(1M) I 디스플레이

```
Inode      0x1 (1) format: file          samu 4.6 08:27:14 Sept 8 2006
           incore: y

00008100 mode      -r-----          409cdf57 access_time
00000001 ino        (1)          1d32ea20
00000001 gen        (1)          4096b499 modify_time
00000002 parent.ino (2)          02588660
00000002 parent.gen (2)          4096b499 change_time
00000000 size_u          02588660
000c0000 size_l      (786432)     4096b443 creation_time
01000000 rm:media/flags          409a8a7c attribute_time
00000000 rm:file_offset          409c0ce6 residence_time
00000000 rm:mau          00000000 unit/cs/arch/flg
00000000 rm:position          00000000 ar_flags
00000000 ext_attrs  -----          00000000 stripe/stride/sg
00000000 ext.ino   (0)          00000000 media  -- --
00000000 ext.gen   (0)          00000000 media  -- --
00000000 uid       root          00000000 psize   (0)
00000000 gid       root          000000c0 blocks  (192)
00000001 nlink     (1)          00000600 free_ino (1536)
00011840 status -n-----  -----  ---

Extents (4k displayed as 1k):
00_ 000000d0.00 000000e0.00 000000f0.00 00000100.00 00000110.00 00000120.00
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) - 미리보기 공유 메모리 디스플레이

J 디스플레이는 미리보기 대기열에 대한 공유 메모리 세그먼트를 표시합니다.

이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command: J
```

## 탐색

표 C-10은 이 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

표 C-10 J 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	한 페이지 뒤로
Ctrl-d	반 페이지 앞으로
Ctrl-f	한 페이지 앞으로
Ctrl-u	반 페이지 뒤로

이 디스플레이는 디버그에 대해 구성되었으며, Sun Microsystems 지원 직원의 도움을 받아야 사용할 수 있습니다.

## 예제 디스플레이

코드 예 C-10은 미리보기 공유 메모리 디스플레이를 표시합니다. 이 결과는 이 메뉴얼에 포함될 수 있도록 일부가 생략되었습니다.

코드 예 C-10 samu(1M) J 디스플레이

```
Preview shared memory size: 155648 samu 4.6 08:30:05 Sept 8 2006

00000000 00040000 00014d58 00000000 00000000 .....MX.....
00000010 00000000 00000000 73616d66 73202d20 .....samfs -
00000020 70726576 69657720 6d656d6f 72792073 preview memory s
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) - 커널 통계 디스플레이

K 디스플레이에는 현재 메모리에 있는 inode의 수와 같은 커널 통계가 표시됩니다.

이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command:K
```

## 탐색

표 C-11은 이 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

표 C-11 K 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	한 페이지 뒤로
Ctrl-f	한 페이지 앞으로

이 디스플레이는 디버그에 대해 구성되었으며, Sun Microsystems 지원 직원의 도움을 받아야 사용할 수 있습니다.

## 예제 디스플레이

코드 예 C-11은 커널 통계 디스플레이를 표시합니다.

코드 예 C-11 samu(1M) K 디스플레이

```
Kernel statistics                               samu 4.6 08:33:19 Sept 8 2006

module: sam-qfs  name: general instance: 0 class: fs
version           4.4.sam-qfs, gumball 2004-05-07 12:12:04
configured file systems      8
mounted file systems        8
nhino                   16384
ninodes                 129526
inocount                129527
inofree                  128577
```

## (1) - 사용량 디스플레이

1 디스플레이는 각 라이브러리 및 파일 시스템에 사용된 용량 및 공간을 포함하여 파일 시스템에 대한 사용량 정보를 표시합니다.

이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command:1
```



## 예제 디스플레이

코드 예 C-13은 공유 메모리 표를 표시합니다.

코드 예 C-13 samu(1M) L 디스플레이

```
Shared memory tables          samu 4.6 08:38:31 May  8 2007

shm ptr tbl:                  defaults:
size          12000 (73728)    optical          mo
left          44c8 (17608)    tape            lt
scanner pid   1861           timeout         600
fifo path     01b0 /var/opt/SUNWsamfs/previews 100
dev_table     01cc           stages          1000
first_dev     0450           log_facility    184
scan_mess     cf50           dio minfilesize 100
preview_shmid 1              label barcode   FALSE
flags         0x20000000      barcodes low    FALSE
preview stages 55776         export unavail  FALSE
preview avail 100           attended        TRUE
preview count 0              start rpc       FALSE
preview sequence 445
age factor    1              vsn factor      1000
fs tbl ptr 0xd820           fs count        8
fseq 10 samfs1 state 0      0      0      0      0
fseq 20 samfs2 state 0      0      0      0      0
fseq 30 qfs1 state 0        0      0      0      0
fseq 40 qfs2 state 0        0      0      0      0
fseq 50 qfs3 state 0        0      0      0      0
fseq 60 qfs4 state 0        0      0      0      0
fseq 100 shareqfs1 state 0  0      0      0      0
fseq 110 shareqfs2 state 0  0      0      0      0
```

## (m) - 대량 저장 상태 디스플레이

m 디스플레이에는 대량 저장 파일 시스템과 구성원 드라이브의 상태가 표시됩니다. 이 디스플레이는 마운트된 파일 시스템만 표시합니다.

이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command:m
```

## 예제 디스플레이

코드 예 C-14는 m 디스플레이를 표시합니다. 멤버 드라이브가 한 공간 들어쓰기 되고 이러한 드라이브가 속한 파일 시스템 바로 아래에 나타납니다.

코드 예 C-14 samu(1M) m 디스플레이

```

Mass storage status                                samu 4.6 08:41:11 Sept 8 2006
  
```

ty	eq	status	use	state	ord	capacity	free	ra	part	high	low
ms	10	m----2----d	1%	on		68.354G	68.343G	1M	16	90%	70%
md	11		1%	on	0	13.669G	13.666G				
md	12		1%	on	1	13.669G	13.667G				
md	13		1%	on	2	13.669G	13.667G				
md	14		1%	on	3	13.674G	13.672G				
md	15		1%	on	4	13.674G	13.672G				
ms	20	m----2----d	1%	on		68.354G	68.344G	1M	16	90%	70%
md	21		1%	on	0	13.669G	13.667G				
md	22		1%	on	1	13.669G	13.667G				
md	23		1%	on	2	13.669G	13.667G				
md	24		1%	on	3	13.674G	13.672G				
md	25		1%	on	4	13.674G	13.672G				
ma	30	m----2----d	4%	on		64.351G	61.917G	1M	16	90%	70%
mm	31		1%	on	0	4.003G	3.988G				[8363840 inodes]
md	32		4%	on	1	64.351G	61.917G				
ma	40	m----2----d	1%	on		64.351G	64.333G	1M	16	90%	70%
mm	41		1%	on	0	4.003G	3.997G				[8382784 inodes]
md	42		1%	on	1	64.351G	64.333G				
ma	50	m----2---r-	1%	on		64.351G	64.333G	1M	16	90%	70%
mm	51		1%	on	0	4.003G	3.997G				[8382784 inodes]
mr	52		1%	on	1	64.351G	64.333G				
ma	60	m----2---r-	1%	on		64.351G	64.331G	1M	16	90%	70%
mm	61		1%	on	0	4.003G	3.997G				[8382784 inodes]
mr	62		1%	on	1	64.351G	64.331G				
ma	100	m----2c--r-	2%	on		270.672G	265.105G	1M	16	80%	70%
mm	101		1%	on	0	2.000G	1.988G				[4168992 inodes]
mr	102		2%	on	1	270.672G	265.469G				
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				

## 필드 설명

표 C-12에서는 이 디스플레이에 대한 필드 설명을 보여줍니다.

표 C-12 samu(1M) m 디스플레이 필드 설명

필드	설명
ty	장치 유형
eq	대량 저장 장치의 장비 서수
status	장치 상태. 상태 코드에 대한 설명은 301페이지의 "운영자 디스플레이 상태 코드"를 참조하십시오.
use	디스크 공간 사용 비율
state	대량 저장 장치의 현재 작동 상태
ord	저장소 패밀리 세트 내의 디스크 장치의 서수
용량	디스크에서 사용 가능한 공간의 1024바이트 블록 수
free	사용 가능한 디스크 공간의 1024바이트 블록 수
ra	Read-Ahead 크기(단위: KB)
part	부분 스테이지 크기(단위: KB)
high	상한 디스크 사용률 임계값 퍼센트
low	하한 디스크 사용률 임계값 퍼센트

## (M) - 공유 메모리 디스플레이

M 디스플레이에는 16진수로 원시 공유 메모리 세그먼트를 표시합니다. 이것은 장치 테이블입니다.

이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command: M
```

## 탐색

표 C-13은 이 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

표 C-13 M 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	한 페이지 뒤로
Ctrl-d	반 페이지 앞으로
Ctrl-f	한 페이지 앞으로
Ctrl-u	반 페이지 뒤로

이 디스플레이는 디버그에 대해 구성되었으며, Sun Microsystems 지원 직원의 도움을 받아야 사용할 수 있습니다.

## 예제 디스플레이

코드 예 C-15는 공유 메모리 디스플레이를 표시합니다. 결과는 이 매뉴얼에 포함될 수 있도록 일부가 생략되었습니다.

코드 예 C-15 samu(1M) M 디스플레이

```
Shared memory          size: 73728          samu 4.6 08:43:20 May  8 2007

00000000 00040000 00014d58 00000000 00000000  .....MX.....
00000010 00000000 00000000 73616d66 73202d20  .....samfs -
00000020 73686172 6564206d 656d6f72 79207365  shared memory se
00000030 676d656e 74000000 00012000 000044c8  gment..... ..DH
00000040 0000dd20 00000000 00000742 00000745  ..] .....B...E
00000050 00000001 00000000 00000000 c0000000  .....@...
00000060 00000001 0001534d 00000000 00000000  .....SM.....
00000070 00000000 00000000 00000000 00000000  .....

00000080 00000000 00000000 00000000 00000000  .....
00000090 20000000 000001b0 000001cc 00000450  .....0...L...P
000000a0 0000cf50 00000001 00000001 4c696365  ..OP.....Lice
000000b0 6e73653a 204c6963 656e7365 206e6576  nse: License nev
000000c0 65722065 78706972 65732e00 00000000  er expires.....
000000d0 00000000 00000000 00000000 00000000  .....
000000e0 00000000 00000000 00000000 00000000  .....
000000f0 00000000 00000000 00000000 00000000  .....
```



## (N) - 파일 시스템 매개변수 디스플레이

N 디스플레이에는 모든 마운트 지점 매개변수, 수퍼 블록 버전 및 기타 파일 시스템 정보가 표시됩니다.

이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command: N
```

### 탐색

표 C-14는 이 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

표 C-14 N 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	이전 파일 시스템
Ctrl-d	페이지 분할 영역 앞으로
Ctrl-f	다음 파일 시스템
Ctrl-i	자세한 상태 해석
Ctrl-u	페이지 분할 영역 뒤로

이 디스플레이는 디버그에 대해 구성되었으며, Sun Microsystems 지원 직원의 도움을 받아야 사용할 수 있습니다.

## 예제 디스플레이

코드 예 C-17에서는 파일 시스템 매개변수 디스플레이를 보여줍니다.

코드 예 C-17

samu(1M) N 디스플레이

```
File system parameters                samu 4.6 08:55:19 Sept 8 2006

mount_point      : /sam1                partial           : 16k
fs_type          : 6                    maxpartial       : 16k
server           :                      partial_stage    : 16384
filesystem name  : samfs1               flush_behind     : 0
eq_type         : 10 ms                 stage_flush_beh  : 0
state version   :      0      2         stage_n_window   : 262144
(fs,mm)_count   :      5      0         stage_retries    : 3
sync_meta       : 0                     stage timeout    : 0
stripe          : 0                     dio_consec r,w   :      0      0
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) - 광 디스크 상태 디스플레이

o 디스플레이에는 환경 내에 구성된 모든 광 디스크 드라이브의 상태가 표시됩니다. 이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command: o
```

### 탐색

표 C-15는 이 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

표 C-15      o 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	한 페이지 뒤로
Ctrl-f	한 페이지 앞으로
Ctrl-b	한 페이지 뒤로
Ctrl-d	반 페이지 앞으로
Ctrl-f	한 페이지 앞으로
Ctrl-k	선택(수동, 자동화된 라이브러리, 수동 및 자동화된 라이브러리, 우선 순위)
Ctrl-u	반 페이지 뒤로

### 예제 디스플레이

코드 예 C-18은 광 디스크 상태 디스플레이를 표시합니다.

코드 예 C-18      samu(1M) o 디스플레이

```
Optical disk status          samu    4.6 Thu Oct 11 13:15:40
ty eq status      act use state vsn
mo 35 --1---wo-r  1 29% ready oper2
```

## 필드 설명

표 C-16에서는 이 디스플레이에 대한 필드 설명을 보여줍니다.

표 C-16 samu(1M) o 디스플레이 필드 설명

필드	설명
ty	장치 유형
eq	광 디스크의 장비 서수
status	장치 상태. 상태 코드에 대한 설명은 301페이지의 "운영자 디스플레이 상태 코드"를 참조하십시오.
act	작업 카운트
use	사용 카트리지 공간의 비율
state	광 디스크의 현재 작동 상태 올바른 장치 상태는 다음과 같습니다. <ul style="list-style-type: none"><li>• ready - 장치가 켜져 있고 디스크가 전송에 로드되어 액세스할 수 있습니다.</li><li>• notrdy - 장치가 켜져있으나 디스크가 전송에 없습니다.</li><li>• idle - 장치에 새로운 연결을 할 수 없습니다. 진행 중인 작업은 완료될 때까지 계속됩니다.</li><li>• off - 장치에 액세스할 수 없습니다.</li><li>• down - 장치에 유지 보수 액세스만 할 수 있습니다.</li></ul>
vsn	광 디스크에 지정된 볼륨 시리얼 이름 또는 볼륨에 레이블이 지정되지 않은 경우에는 키워드 nolabel

## (p) - 제거 가능한 매체 로드 요청 디스플레이

p 디스플레이는 제거 가능한 매체에 대해 해결되지 않은 로드 요청에 대한 정보를 나열합니다. *mt* 인수를 사용하여 특정 매체 유형(예: DLT 테이프) 또는 매체 패밀리를 선택할 수 있습니다. 우선 순위 디스플레이에는 사용자 대기열이 아닌 미리보기 대기열의 우선 순위가 나열되고 항목이 우선 순위별로 정렬됩니다.

다음 형식으로 마운트 요청을 표시합니다.

- 사용자에 의한 수동 및 자동화된 라이브러리 요청 모두
- 우선 순위에 의한 수동 및 자동화된 라이브러리 요청 모두
- 수동 요청만
- 자동화된 라이브러리 요청만

보려는 것이 무엇인지에 따라 이 디스플레이를 다음과 같이 다르게 호출할 수 있습니다.

- 현재 선택된 모든 제거 가능한 매체에 대한 마운트 요청을 표시하려면 다음 형식을 사용하여 해당 명령을 입력하십시오.

```
Command: p
```

- 주어진 제거 가능한 매체 장치에 대한 마운트 요청을 표시하려면 다음 형식을 사용하여 해당 명령을 입력하십시오.

```
Command: p mt
```

*mt*에 대해 *mcf(4)* 매뉴얼 페이지에 표시된 매체 유형 중 하나를 지정합니다.

## 탐색

[표 C-17](#)은 이 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

**표 C-17** p 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	한 페이지 뒤로
Ctrl-d	반 페이지 앞으로
Ctrl-f	한 페이지 앞으로
Ctrl-k	다른 디스플레이 형식 사이를 토글합니다.
Ctrl-u	반 페이지 뒤로

## 예제 디스플레이

[코드 예 C-19](#)는 제거 가능한 매체 로드 요청 디스플레이를 표시합니다.

**코드 예 C-19** samu(1M) p 디스플레이

```
Removable media load requests all both samu 4.6 09:14:19 Sept 8 2006
count: 1

index type pid user rb flags wait count vsn
0 dt 15533 root 150 W--f--- 0:00 DAT001
```

## 필드 설명

표 C-18에서는 이 디스플레이에 대한 필드 설명을 보여줍니다.

표 C-18 samu(1M) p 디스플레이 필드 설명

필드	설명
index	미리보기 테이블의 인덱스 번호
type	제거 가능한 매체에 할당된 장치 유형 코드
pid	UNIX 프로세스 식별자. 프로세스 식별자 1은 NFS 액세스를 나타냅니다.
user	로드를 요청하는 사용자에게 할당된 이름
priority	요청 우선 순위
rb	요청된 VSN이 상주하는 자동화된 라이브러리의 장비 서수
flags	장치의 플래그. 표 C-19를 참조하십시오.
wait	마운트 요청을 받은 후 경과한 시간
count	스테이지인 경우 이 VSN에 대한 요청 횟수
vsn	볼륨의 VSN

## 플래그

표 C-19는 p 디스플레이에 대한 플래그를 표시합니다.

표 C-19 samu(1M) p 디스플레이의 플래그 필드

필드	설명
w-----	쓰기 액세스가 요청되었습니다.
-b-----	항목이 사용 중입니다.
--C----	VSN 지우기가 요청되었습니다.
---f---	파일 시스템이 요청되었습니다.
----N--	매체는 파일 시스템과 관계가 없습니다.
-----S-	배치면을 이미 마운트했습니다.
-----s	스테이지 요청 플래그

## (P) - 활성화 서비스 디스플레이

P 디스플레이는 Sun StorageTek QFS 단일 포트 다중 송신기로 등록된 서비스를 나열합니다.

이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command: P
```

### 탐색

표 C-20은 이 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

표 C-20 P 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	한 페이지 뒤로
Ctrl-f	한 페이지 앞으로

이 디스플레이는 디버그에 대해 구성되었으며, Sun Microsystems 지원 직원의 도움을 받아야 사용할 수 있습니다.

### 예제 디스플레이

코드 예 C-20은 활성화 서비스 디스플레이를 표시합니다.

코드 예 C-20 samu(1M) P 디스플레이

```
Active Services                    samu      4.6 09:08:33 Sept 8 2006

Registered services for host 'pup':
  sharedfs.qfs2
  sharedfs.qfs1
  2 services registered.
```

## (r) - 제거 가능한 매체 상태 디스플레이

r 디스플레이를 사용하면 테이프 드라이브와 같은 제거 가능한 매체 장치에 대한 작업을 모니터링할 수 있습니다. 특정 유형의 장치(예: 비디오 테이프) 또는 장치 패밀리(예: 모든 테이프 장치)를 모니터링할 수 있습니다.

보려는 것이 무엇인지에 따라 이 디스플레이를 다음과 같이 다르게 호출할 수 있습니다.

- 모든 제거 가능한 매체 장치에 대한 상태를 표시하려면 다음 형식을 사용하여 해당 명령을 입력하십시오.

```
Command:r
```

- 특정 장치에 대한 상태를 표시하려면 다음 형식을 사용하여 해당 명령을 입력하십시오.

```
Command:r eq
```

eq에는 해당 장치에 대한 장비 서수를 지정합니다.

## 예제 디스플레이

코드 예 C-21은 제거 가능한 매체 상태 디스플레이를 표시합니다.

코드 예 C-21 samu(1M) r 디스플레이

```
Removable media status: all          samu 4.6 09:11:27 Sept 8 2006
ty  eq  status      act  use  state  vsn
dt 150 --1-----r   0  63% ready  DAT001
```

## 필드 설명

표 C-21에서는 이 디스플레이에 대한 필드 설명을 보여줍니다.

표 C-21 samu(1M) r 디스플레이 필드 설명

필드	설명
ty	장치 유형
eq	드라이브의 장비 서수
status	장치 상태. 상태 코드에 대한 설명은 301페이지의 "운영자 디스플레이 상태 코드"를 참조하십시오.

표 C-21 samu(1M) r 디스플레이 필드 설명(계속)

필드	설명
act	작업 카운트
use	사용 카트리지 공간의 비율
state	제거 가능한 매체의 현재 작동 상태. 올바른 장치 상태는 다음과 같습니다. <ul style="list-style-type: none"> <li>• ready - 장치가 켜져 있고 디스크나 테이프가 전송에 로드되어 액세스할 수 있습니다.</li> <li>• notrdy - 장치가 켜져 있으나 디스크나 테이프가 전송에 없습니다.</li> <li>• idle - 장치에 새로운 연결을 할 수 없습니다. 진행 중인 작업은 완료될 때까지 계속됩니다.</li> <li>• off - 장치에 액세스할 수 없습니다.</li> <li>• down - 장치에 유지 보수 액세스만 할 수 있습니다.</li> <li>• nalloc-nalloc 플래그가 설정되어 있어 해당 장치에 추가로 할당되지 않도록 합니다. 자세한 내용은 27페이지의 "논리 장치 번호(LUN)당 할당 제어"를 참조하십시오.</li> </ul>
vsn	볼륨에 지정된 VSN 또는 볼륨에 레이블이 지정되지 않은 경우에는 키워드 nolabel. 볼륨이 전송에 없거나 장치가 꺼져있는 경우에는 비어 있습니다.

## (R) - Sun SAM-Remote 정보 디스플레이

R은 Sun SAM-Remote 구성에 대한 정보 및 상태를 표시합니다.

이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command: R
```

이 디스플레이는 디버그에 대해 구성되었으며, Sun Microsystems 지원 직원의 도움을 받아야 사용할 수 있습니다.

## (s) - 장치 상태 디스플레이

s 디스플레이에는 환경 내에서 구성된 모든 장치의 상태가 표시됩니다.

이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command: s
```

## 탐색

표 C-22는 이 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

표 C-22 s 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	한 페이지 뒤로
Ctrl-d	반 페이지 앞으로
Ctrl-f	한 페이지 앞으로
Ctrl-u	반 페이지 뒤로

## 예제 디스플레이

코드 예 C-22는 장치 상태 디스플레이를 표시합니다.

코드 예 C-22 samu(1M) s 디스플레이

```
Device status                    samu      4.6 09:14:05 Sept 8 2006
ty   eq state  device_name                fs status  pos
sk   100 on    /etc/opt/SUNWsamfs/dcstkconf    100 m-----r
      stk_dismount(2275) 0, volser 700073
sg   120 on    /dev/rmt/2cbn                    100 -----p
      empty
sg   130 on    /dev/rmt/5cbn                    100 --l----o-r
      Ready for data transfer
sg   140 on    /dev/rmt/6cbn                    100 -----p
      empty
sg   150 on    /dev/rmt/4cbn                    100 -----p
      empty
hy   151 on    historian                          151 -----
```

## 필드 설명

표 C-23에서는 이 디스플레이에 대한 필드 설명을 보여줍니다.

표 C-23 samu(1M) s 디스플레이 필드 설명

필드	설명
ty	장치 유형
eq	장치의 장비 서수
state	장치의 현재 작동 상태
device_name	장치 경로. 파일 시스템 장치의 경우에는 파일 시스템 이름
fs	장치가 속한 패밀리 세트의 장비 서수
status	장치 상태. 상태 코드에 대한 설명은 301페이지의 "운영자 디스플레이 상태 코드"를 참조하십시오.

## (S) - 섹터 데이터 디스플레이

s 디스플레이는 원시 장치 데이터를 표시합니다.

이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command: s
```

## 탐색

표 C-24는 이 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

표 C-24 s 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	이전 섹터
Ctrl-d	한 페이지 앞으로(상단 부분)
Ctrl-f	다음 섹터
Ctrl-k	고급 디스플레이 형식
Ctrl-u	한 페이지 뒤로(상단 부분)

이 디스플레이는 디버그에 대해 구성되었으며, Sun Microsystems 지원 직원의 도움을 받아야 사용할 수 있습니다.

## (t) - 테이프 드라이브 상태 디스플레이

t 디스플레이는 환경 내에 구성된 모든 테이프 드라이브의 상태를 표시합니다.

이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command:t
```

### 탐색

표 C-25는 이 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

표 C-25      t 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	한 페이지 뒤로
Ctrl-f	한 페이지 앞으로

### 예제 디스플레이

코드 예 C-23은 테이프 드라이브 상태 디스플레이를 표시합니다.

코드 예 C-23      samu(1M) t 디스플레이

```
Tape drive status                               samu      4.6 09:21:07 Sept 8 2006

ty  eq  status      act  use  state  vsn
sg 120 -----p    0   0%  notrdy
      empty
sg 130 -----p    0   0%  notrdy
      empty
sg 140 -----p    0   0%  notrdy
      empty
sg 150 --l-----r    0  41%  ready   700088
      idle
```

## 필드 설명

표 C-26에서는 이 디스플레이에 대한 필드 설명을 보여줍니다.

표 C-26 samu(1M) t 디스플레이 필드 설명

필드	설명
ty	장치 유형
eq	드라이브의 장비 서수
status	장치 상태. 상태 코드에 대한 설명은 301페이지의 "운영자 디스플레이 상태 코드"를 참조하십시오.
act	작업 카운트
use	사용 카트리지 공간의 비율
state	제거 가능한 매체의 현재 작동 상태. 올바른 장치 상태는 다음과 같습니다. <ul style="list-style-type: none"><li>• ready - 장치가 켜져있고 디스크나 테이프가 전송에 로드되어 액세스할 수 있습니다.</li><li>• notrdy - 장치가 켜져있으나 디스크나 테이프가 전송에 없습니다.</li><li>• idle - 장치에 새로운 연결을 할 수 없습니다. 진행 중인 작업은 완료될 때까지 계속됩니다.</li><li>• off - 장치에 액세스할 수 없습니다.</li><li>• down - 장치에 유지 보수 액세스만 할 수 있습니다.</li><li>• nalloc - nalloc 플래그가 설정되어 있어 해당 장치에 추가로 할당되지 않도록 합니다. 자세한 내용은 27페이지의 "논리 장치 번호(LUN)당 할당 제어"를 참조하십시오.</li></ul>
vsn	볼륨에 지정된 볼륨 일련 번호, 또는 볼륨에 레이블이 지정되지 않은 경우에는 키워드 nolabel. 볼륨이 전송에 없거나 장치가 꺼져있는 경우에는 비어 있습니다.

## (T) - SCSI 감지 데이터 디스플레이

T 디스플레이는 SCSI 장치의 SCSI 상태를 표시합니다.

이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command: T
```

### 탐색

표 C-27은 이 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

표 C-27 T 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	이전 장비
Ctrl-f	다음 장비

이 디스플레이는 디버그에 대해 구성되었으며, Sun Microsystems 지원 직원의 도움을 받아야 사용할 수 있습니다.

## (u) - 스테이지 대기열 디스플레이

u 디스플레이는 스테이징 대기열의 모든 파일을 나열합니다.

이 디스플레이를 호출하려면 다음 명령을 입력하십시오.

```
Command: u
```

## 탐색

표 C-28은 이 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

표 C-28 u 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	한 페이지 뒤로
Ctrl-d	반 페이지 앞으로
Ctrl-f	한 페이지 앞으로
Ctrl-k	각 항목의 두 번째 행에 경로를 표시합니다.
Ctrl-u	반 페이지 뒤로

## 예제 디스플레이

코드 예 C-24는 스테이지 대기열 디스플레이를 표시합니다.

코드 예 C-24 samu(1M) u 디스플레이

```
Staging queue by media type: all          samu 4.6 09:24:23 Sept 8 2006
volumes 1 files 22

  ty      length  fseq      ino  position  offset vsn
dt    451.611k   20      1030   207cc     473 DAT001
dt    341.676k   20      1031   207cc     7fc DAT001
dt    419.861k   20      1032   207cc     aa9 DAT001
dt    384.760k   20      1033   207cc     df2 DAT001
dt    263.475k   20      1034   207cc    10f5 DAT001
dt    452.901k   20      1035   207cc    1305 DAT001
dt    404.598k   20      1036   207cc    1690 DAT001
dt    292.454k   20      1037   207cc    19bb DAT001
dt    257.835k   20      1038   207cc    1c05 DAT001
dt    399.882k   20      1040   207cc    1e0b DAT001
dt    399.882k   40      1029   208d7      2 DAT001
dt    257.835k   40      1030   208d7    323 DAT001
dt    292.454k   40      1031   208d7    528 DAT001
dt    404.598k   40      1032   208d7    772 DAT001
dt    452.901k   40      1033   208d7    a9d DAT001
dt    263.475k   40      1034   208d7    e28 DAT001
dt    384.760k   40      1035   208d7   1038 DAT001
dt    419.861k   40      1036   208d7   133b DAT001
dt    341.676k   40      1037   208d7   1684 DAT001
```

dt	451.611k	40	1038	208d7	1931 DAT001
dt	161.326k	40	1039	208d7	1cba DAT001
dt	406.400k	40	1040	208d7	1dfe DAT001

## 필드 설명

표 C-29에서는 이 디스플레이에 대한 필드 설명을 보여줍니다.

표 C-29 samu(1M) u 디스플레이 필드 설명

필드	설명
ty	장치 유형
길이	파일 길이
fseq	파일 시스템 장비 번호
ino	inode 번호
위치	특정 매체에서 아카이브 파일의 위치
offset	특정 매체에서 아카이브 파일의 오프셋
vsn	볼륨의 VSN

## (U) - 장치 테이블 디스플레이

U 디스플레이는 사람이 판독할 수 있는 형식으로 장치 테이블을 표시합니다.

보려는 것이 무엇인지에 따라 이 디스플레이를 다음과 같이 다르게 호출할 수 있습니다.

- 모든 장치에 대한 장치 테이블을 표시하려면 다음 형식을 사용하여 해당 명령을 입력하십시오.

```
Command:U
```

- 특정 장치에 대한 장치 테이블을 표시하려면 다음 형식을 사용하여 해당 명령을 입력하십시오.

```
Command:U eq
```

eq에는 장치의 장비 서수를 지정합니다.

## 탐색

표 C-30은 이 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

표 C-30 U 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	이전 장비
Ctrl-f	다음 장비

이 디스플레이는 디버그에 대해 구성되었으며, Sun Microsystems 지원 직원의 도움을 받아야 사용할 수 있습니다.

## 예제 디스플레이

코드 예 C-25는 장치 테이블 디스플레이를 표시합니다.

코드 예 C-25 samu(1M) U 디스플레이

```
Device table: eq: 10      addr: 00000450  samu 4.6 09:28:40 Sept 8 2006

message:

0004000000014d58 0000000000000000      00000000 delay
0000000000000000 mutex                  00000000 unload_delay
00000aa8 next
73616d66 set:  samfs1
73310000
00000000
00000000
000a000a eq/fseq
08010801 type/equ_type
0000      state
00000000 st_rdev
00000000 ord/model
00000000 mode_sense
00000000 sense
00000000 space
00000000 capacity
00000000 active
00000000 open
00000000 sector_size
00000000 label_address
00000000 vsn:
00000000
00000000
```

```
00000000
00000000 status: -----
00000000 dt
73616d66 name: samfs1
```

## (v) - 자동화된 라이브러리 카탈로그 디스플레이

v 디스플레이에는 현재 자동화된 라이브러리에 분류된 모든 디스크 또는 테이프의 위치와 VSN이 표시됩니다.

보려는 것이 무엇인지에 따라 이 디스플레이를 다음과 같이 다르게 호출할 수 있습니다.

- 모든 장치에 대한 카탈로그를 표시하려면 다음 형식을 사용하여 해당 명령을 입력하십시오.

```
Command: v
```

- 특정 장치에 대한 카탈로그 정보를 표시하려면 다음 형식을 사용하여 해당 명령을 입력하십시오.

```
Command: v eq
```

eq에는 장치의 장비 서수를 지정합니다. 키워드 `historian`을 입력하여 기록자 카탈로그를 봅니다.

경우에 따라서는 다음과 같이 samu(1M)에서 장치를 입력하라는 메시지가 나타납니다.

```
Enter robot: eq
```

eq에 대해 장치의 장비 서수를 지정하거나 Return을 누릅니다. Return을 누르면 지정된 이전 장치에 대한 정보가 나타납니다.

모든 장치 이름 및 장비 서수의 목록은 260페이지의 "(c) - 장치 구성 디스플레이"를 참조하십시오.

## 탐색

표 C-31은 이 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

표 C-31 v 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	한 페이지 뒤로
Ctrl-d	다음 라이브러리 카탈로그
Ctrl-f	한 페이지 앞으로
Ctrl-i	상세, 2행 디스플레이 형식. Ctrl-i를 한 번 누르면 시간과 바코드가 나타납니다. Ctrl-i를 두 번 누르면 두 번째 행에 볼륨 예약이 나타납니다.
Ctrl-k	고급 정렬 키. Ctrl-k를 입력한 후 다음 중 하나를 입력하여 정렬 키를 선택할 수 있습니다. 1 - 슬롯별로 정렬 2 - 카운트별로 정렬 3 - 사용률별로 정렬 4 - VSN별로 정렬 5 - 액세스 시간별로 정렬 6 - 바코드별로 정렬 7 - 레이블 지정 시간별로 정렬
Ctrl-u	이전 자동화된 라이브러리 카탈로그
/	VSN 검색
%	바코드 검색
\$	슬롯 검색

## 예제 디스플레이

코드 예 C-26은 자동화된 라이브러리 카탈로그 디스플레이를 표시합니다.

코드 예 C-26 samu(1M) v 디스플레이

```
Robot VSN catalog by slot      : eq 100samu      4.6 09:30:25 Sept 8 2006
count 32
slot      access time count use flags      ty vsn
0      2004/05/08 08:35  64  0% -il-o-b----- sg 700071
1      2004/05/08 09:08  27 12% -il-o-b----- sg 700073
2      2004/05/08 09:12  26 12% -il-o-b----- sg 700077
```

3	2004/05/08 08:39	37	40%	-il-o-b-----	sg 700079
4	2004/05/08 09:16	24	6%	-il-o-b-----	sg 700084
5	2004/05/08 09:18	24	41%	-il-o-b-----	sg 700088
6	none	0	0%	-il-o-b-----	sg 700090
7	none	0	0%	-il-o-b-----	sg 700092
8	none	0	0%	-il-o-b-----	sg 000155
9	none	0	0%	-il-o-b-----	sg 000156
10	none	0	0%	-il-o-b-----	sg 000157
11	none	0	0%	-il-o-b-----	sg 000158
12	none	0	0%	-il-o-b-----	sg 000154
13	none	0	0%	-il-o-b-----	sg 000153
14	none	0	0%	-il-o-b-----	sg 000152

### 필드 설명

표 C-32에서는 이 디스플레이에 대한 필드 설명을 보여줍니다.

표 C-32 samu(1M) v 디스플레이 필드 설명

필드	설명
Robot VSN catalog	지정된 자동화된 라이브러리 이름과 디스플레이가 새로 고쳐진 시간
count	이 라이브러리의 카탈로그에 할당된 슬롯의 수
slot	지정된 라이브러리의 슬롯 번호
access time	블륨이 마지막으로 액세스된 시간
count	마지막 감사 작업 이후 이 블륨에 액세스한 횟수
use	블륨에 사용한 공간 비율
flags	장치의 플래그. 플래그에 대한 내용은 표 C-33을 참조하십시오.
ty	장치 유형
vsn	블륨의 VSN

## 플래그

표 C-33은 표 C-32에 있는 flags 필드의 플래그를 보여줍니다. 경우에 따라서는 2개 이상의 플래그가 하나의 필드에 발생할 수 있으며 하나의 플래그가 다른 플래그보다 우선합니다.

표 C-33 samu(1M) v 디스플레이의 플래그 필드

플래그	설명
A-----	볼륨에 감사가 필요합니다.
-i-----	슬롯이 사용 중입니다.
--l-----	레이블이 지정되었으며, N보다 우선합니다.
--N-----	레이블이 지정되지 않았습니다. 이 볼륨은 환경과 관계가 없습니다.
---E-----	매체 오류. 소프트웨어가 카트리지에서 쓰기 오류를 감지할 때 설정합니다.
----o-----	슬롯이 사용 중입니다.
-----C-----	볼륨이 클리닝 테이프입니다. p보다 우선합니다.
-----p-----	우선 순위 VSN
-----b-----	바코드를 감지했습니다.
-----W----	쓰기 보호. 카트리지에 대한 물리적인 쓰기 보호 메커니즘이 활성화되면 설정됩니다.
-----R----	읽기 전용
-----c--	리사이클
-----d-	중복된 VSN. U보다 우선합니다.
-----U-	볼륨 사용 불가
-----f	아카이버가 볼륨이 차 있음을 감지
-----X	내보내기 슬롯

## (w) - 보류 스테이지 대기열

w 디스플레이에는 볼륨이 아직 로드되지 않은 대기 스테이지 요청이 표시됩니다.

보려는 것이 무엇인지에 따라 이 디스플레이를 다음과 같이 다르게 호출할 수 있습니다.

- 모든 매체 장치에 대한 해결되지 않은 스테이지 대기열을 표시하려면 다음 형식을 사용하여 해당 명령을 입력하십시오.

Command:w

- 특정 매체 유형에 대한 해결되지 않은 스테이지 대기열을 표시하려면 다음 형식을 사용하여 해당 명령을 입력하십시오.

```
Command:w mt
```

*mt*에 대해 *mcf(4)* 매뉴얼 페이지에 표시된 매체 유형 중 하나를 지정합니다.

## 탐색

**표 C-34**는 이 디스플레이에서 사용할 수 있는 컨트롤 키를 표시합니다.

**표 C-34** w 디스플레이에 대한 컨트롤 키

키	기능
Ctrl-b	한 페이지 뒤로
Ctrl-d	반 페이지 앞으로
Ctrl-f	한 페이지 앞으로
Ctrl-k	각 항목의 두 번째 행에 경로를 표시합니다.
Ctrl-u	반 페이지 뒤로

## 예제 디스플레이

**코드 예 C-27**은 해결되지 않은 스테이지 대기열을 표시합니다.

**코드 예 C-27** samu(1M) w 디스플레이

```
Pending stage queue by media type: all      samu      4.6 Thu Oct 11 13:20:27
volumes 1 files 13
```

ty	length	fseq	ino	position	offset	vsn
at	1.383M	1	42	3a786	271b	000002
at	1.479M	1	56	3a786	5139	000002
at	1018.406k	1	60	3a786	6550	000002
at	1.000M	1	65	3a786	7475	000002
at	1.528M	1	80	3a786	99be	000002
at	1.763M	1	92	3a786	ce57	000002
at	1.749M	1	123	3a786	11ece	000002
at	556.559k	1	157	3a786	1532f	000002
at	658.970k	1	186	3a786	17705	000002
at	863.380k	1	251	3a786	1dd58	000002

at	1.268M	1	281	3a786	1f2b7	000002
at	1.797M	1	324	3a786	23dfa	000002
at	1.144M	1	401	3a786	2bb6d	000002

## 필드 설명

표 C-35에서는 이 디스플레이에 대한 필드 설명을 보여줍니다.

표 C-35 samu(1M) w 디스플레이 필드 설명

필드	설명
ty	장치 유형
길이	파일 길이
fseq	파일 시스템 장비 서수
ino	inode 번호
위치	특정 매체에서 아카이브 파일의 위치(십진수 형식)
offset	특정 매체에서 아카이브 파일의 오프셋
vsn	볼륨의 VSN

## 운영자 디스플레이 상태 코드

운영자 디스플레이에는 제거 가능한 매체 장치 디스플레이와 파일 시스템 디스플레이에 대한 여러 가지 상태 코드가 있습니다. 다음 섹션에서는 이들 상태 코드에 대해 설명합니다.

### 제거 가능한 매체 장치 디스플레이 상태 코드

o, r, s 및 t 운영자 디스플레이에는 제거 가능한 매체 장치에 대한 상태 코드가 표시됩니다. 상태 코드는 10가지 위치의 형식으로 표시되며 왼쪽(위치 1)에서 오른쪽(위치 10)으로 읽습니다.

이 절의 상태 코드는 samu(1M) f, m 및 v 디스플레이에 적용되지 않습니다. f 및 m 디스플레이의 상태 코드에 대한 내용은 [302페이지의 "파일 시스템 디스플레이 상태 코드"](#)를 참조하십시오. v 디스플레이의 상태 코드에 대한 내용은 [296페이지의 "\(v\) - 자동화된 라이브러리 카탈로그 디스플레이"](#)를 참조하십시오.

표 C-36에서는 각 위치에 대한 올바른 상태 코드를 정의합니다.

표 C-36 제거 가능한 매체 장치 디스플레이 상태 코드

상태 비트	장치에 대한 의미
s-----	매체가 스캔되고 있습니다.
m-----	자동화된 라이브러리는 선택 사항입니다.
M-----	유지 보수 모드
-E-----	스캔 도중 장치에 복구할 수 없는 오류가 발생했습니다.
-a-----	장치가 감사 모드입니다.
--l-----	매체에 레이블이 있습니다.
--N-----	외부 매체
--L-----	매체가 레이블되고 있습니다.
---I-----	장치 유틸 상태 대기 중입니다.
---A-----	운영자의 주의가 필요합니다.
----C-----	청소가 필요합니다.
----U-----	언로드가 요청되었습니다.
-----R-----	장치가 예약되었습니다.
-----w---	프로세스가 매체에 쓰는 중입니다.
-----o--	장치가 열려있습니다.
-----P-	장치가 위치를 지정하는 중입니다(테이프만 해당).
-----F-	자동화된 라이브러리의 경우, 모든 저장소 슬롯이 사용 중입니다. 테이프 및 광자기 드라이브의 경우, 매체가 가득 찼습니다.
-----R	장치가 준비되었고 매체가 읽기 전용입니다.
-----r	장치가 스핀 업 및 준비되었습니다.
-----p	장치가 있습니다.
-----W	장치가 쓰기 보호되어 있습니다.

## 파일 시스템 디스플레이 상태 코드

f 및 m 운영자 디스플레이에는 파일 시스템에 대한 상태 코드가 표시됩니다. 상태 코드는 11가지 위치의 형식으로 표시되며 왼쪽(위치 1)에서 오른쪽(위치 11)으로 읽습니다.

이 절의 상태 코드는 samu(1M) c, o, r, s, t 또는 v 디스플레이에 적용되지 않습니다. o, r, s 및 t 디스플레이의 상태 코드에 대한 정보는 301페이지의 "제거 가능한 매체 장치 디스플레이 상태 코드"를 참조하십시오. v 디스플레이의 상태 코드에 대한 내용은 296페이지의 "(v) - 자동화된 라이브러리 카탈로그 디스플레이"를 참조하십시오.

표 C-37에서는 각 위치에 대한 올바른 상태 코드를 정의합니다.

표 C-37 파일 시스템 디스플레이 상태 코드

상태 비트	파일 시스템에 대한 의미
m-----	파일 시스템이 현재 마운트되어 있습니다.
M-----	파일 시스템이 마운트되고 있습니다.
-u-----	파일 시스템이 마운트 해제되고 있습니다.
--A-----	파일 시스템 데이터가 아카이브되고 있습니다.
---R-----	파일 시스템 데이터가 릴리스되고 있습니다.
----S-----	파일 시스템 데이터가 스테이지되고 있습니다.
-----1-----	Sun StorageTek SAM 파일 시스템 버전 1
-----2-----	Sun StorageTek SAM 파일 시스템 버전 2
-----c-----	Sun StorageTek QFS 공유 파일 시스템
-----W---	단일 작성기
-----R--	다중 관독기
-----r-	mr 장치
-----d	md 장치

## 운영자 디스플레이 장치 상태

c, m, o, r, s 및 t 운영자 디스플레이는 장치 상태 코드를 표시합니다. 이러한 코드는 장치에 대한 현재 액세스 상태를 나타냅니다. 표 C-38은 유효한 상태 코드를 정의합니다.

표 C-38 운영자 디스플레이 장치 상태

장치 상태	설명
on	장치에 액세스할 수 있습니다. 특정 디스플레이에서는 이 상태가 ready 또는 notrdy 상태로 대체될 수도 있습니다.
ro	장치에 읽기 전용 액세스만 할 수 있습니다. 특정 디스플레이에서는 이 상태가 ready 또는 notrdy 상태로 대체될 수도 있습니다.

표 C-38 운영자 디스플레이 장치 상태(계속)

장치 상태	설명
off	장치에 액세스할 수 없습니다. 테이프 및 광 디스크 드라이브의 경우에는 다음과 같은 여러 가지 이유 때문에 장치가 off 상태가 될 수 있습니다. <ul style="list-style-type: none"> <li>• 클리닝이 요청되었으나 자동화된 라이브러리에 클리닝 카트리지가 없습니다.</li> <li>• 클리닝 카트리지를 드라이브에서 로드 또는 언로드할 수 없습니다.</li> <li>• 초기화 도중 드라이브 상태가 가득 찬 것을 알았으나 드라이브를 비울 수 없습니다.</li> <li>• 시스템은 드라이브에서 카트리지를 제거할 수 없습니다.</li> <li>• 스핀 업 도중 입출력용 드라이브를 열 수 없습니다.</li> <li>• 언로드를 위해 드라이브를 스핀 다운할 때 NOT READY 이외의 오류를 수신합니다.</li> <li>• 스핀 업 과정에서 드라이브의 표준 테이프 드라이버를 열 수 없습니다.</li> </ul>
down	장치에 유지 보수 액세스만 할 수 있습니다.
idle	장치에 새로운 연결을 할 수 없습니다. 진행 중인 작업은 완료될 때까지 계속됩니다.
ready	장치가 켜져있고 전송에 로드된 디스크나 테이프에 액세스할 수 있습니다.
notrdy	장치가 켜져있으나 디스크나 테이프가 전송에 없습니다.
unavail	장치에 액세스할 수 없으며 자동 작동에 사용할 수 없습니다. 장치가 unavail 상태에 있는 동안 매체 이동에 load(1M) 및 unload(1M) 명령을 계속 사용할 수 있습니다.
nalloc	nalloc 플래그가 설정되어 있어 해당 장치에 추가로 할당되지 않도록 합니다. 자세한 내용은 27페이지의 "논리 장치 번호(LUN)당 할당 제어"를 참조하십시오.

samu(1M) down, off 및 on 장치 상태 명령을 사용하여 장치 상태를 down, off 또는 on 으로 변경할 수 있습니다. 모든 samu(1M) 디스플레이로부터 이러한 명령을 입력할 수 있지만, c, m, o, r, s 또는 t 디스플레이로부터 명령을 입력하는 경우, 디스플레이에서 장치 상태 변경 사항을 볼 수 있습니다. 예를 들어 p 디스플레이 내에서 off로 장치 상태를 설정할 수 있지만 해당 디스플레이에 반영된 새로운 장치 상태를 볼 수 없습니다.

다음 절차는 down에서 on으로, 그리고 on에서 down으로 장치의 상태를 변경하기 위해 입력해야 하는 것을 표시합니다.

### ▼ down에서 on으로 드라이브 상태 변경하기

1. 드라이브 및 자동화된 라이브러리 장치 상태를 표시하는 samu(1M) 디스플레이를 불러옵니다.

다음 모든 samu(1M) 디스플레이는 장치 상태를 표시합니다. c, m, o, r, s 및 t.

2. 장치가 down 상태에 있는지 확인하려면 시각적으로 해당 디스플레이를 검사합니다.

3. `:off`를 입력합니다.

장치를 끄면 모든 작업이 중지되어 장치가 다음 단계에서 깨끗하게 시작될 수 있습니다.  
예를 들어, 다음과 같습니다.

```
Command:off eq
```

`eq`에는 장치의 장비 서수를 지정합니다.

4. `:on`을 입력합니다.

예를 들어, 다음과 같습니다.

```
Command:on eq
```

`eq`에는 장치의 장비 서수를 지정합니다.

## ▼ on에서 down으로 드라이브 상태 변경하기

1. 드라이브 및 자동화된 라이브러리 장치 상태를 표시하는 **samu(1M)** 디스플레이를 불러옵니다.

다음 모든 **samu(1M)** 디스플레이는 장치 상태를 표시합니다. `c`, `m`, `o`, `r`, `s` 및 `t`.

2. 장치가 `on` 상태에 있는지 확인하려면 시각적으로 해당 디스플레이를 검사합니다.

3. `:off`를 입력합니다.

장치를 끄면 모든 작업이 중지되어 장치가 다음 단계에서 완전히 정지될 수 있습니다.  
예를 들어, 다음과 같습니다.

```
Command:off eq
```

`eq`에는 장치의 장비 서수를 지정합니다.

4. `:down`을 입력합니다.

예를 들어, 다음과 같습니다.

```
Command:down eq
```

`eq`에는 장치의 장비 서수를 지정합니다.

## 운영자 명령

다음 절은 samu(1M) 운영자 유틸리티의 명령 인터페이스에서 입력할 수 있는 운영자 명령을 설명합니다. 모든 디스플레이로부터 명령을 입력할 수 있습니다.

다음 유형의 운영자 명령을 사용할 수 있습니다.

- 306페이지의 "장치 명령"
- 307페이지의 "파일 시스템 명령: I/O 관리"
- 309페이지의 "파일 시스템 명령: 직접 I/O 관리"
- 311페이지의 "파일 시스템 명령: Sun StorageTek QFS 공유 파일 시스템"
- 312페이지의 "파일 시스템 명령: 기타"
- 316페이지의 "기타 명령"

Solaris OS 명령줄의 운영자 명령을 입력하려면 이 명령을 samcmd(1M) 명령에 대한 인수로 사용해야 합니다. samcmd(1M) 명령에 대한 자세한 내용은 samcmd(1M) 매뉴얼 페이지를 참조하십시오.

다음 하위 절에서 핫 키 시리즈가 아니라 명령이 입력되도록 구성하기 위해 입력할 경우 각 samu(1M) 명령은 콜론(:)으로 시작됩니다.

## 장치 명령

표 C-39에서는 장치 명령 및 이러한 명령의 작업을 보여줍니다.

표 C-39      장치 명령 작업

명령	작업
down	장치 <i>eq</i> 에 대한 작업을 종료합니다.
idle	장치에 새로운 연결을 할 수 없도록 하여 장치 <i>eq</i> 에 대한 액세스를 제한합니다. 기존 작업은 완료될 때까지 계속됩니다.
off	장치 <i>eq</i> 를 논리적으로 끕니다.
on	장치 <i>eq</i> 를 논리적으로 켭니다.
unavail	<i>eq</i> 장치를 선택하여 파일 시스템과 함께 사용할 수 없도록 만듭니다. 예를 들어, 매체를 로드하여 파일 시스템을 복원하고 Sun StorageTek SAM 소프트웨어가 이 드라이브를 사용하지 않는 재난 복구 상태에서 드라이브 상태를 unavail로 설정할 수도 있습니다.

표 C-39 장치 명령 작업(계속)

명령	작업
unload	지정된 제거 가능한 매체 장치 <i>eq</i> 에 대해 마운트된 매체를 언로드합니다. 매거진 장치의 경우, unload 명령은 마운트된 카트리지를 언로드하고 매거진을 배출합니다.
nalloc	장치에 nalloc 플래그를 설정하여 해당 장치에 추가로 할당되지 않도록 합니다. 자세한 내용은 27페이지의 "논리 장치 번호(LUN)당 할당 제어"를 참조하십시오.
alloc	장치에서 nalloc 플래그를 제거합니다. nalloc 플래그는 해당 장치에 추가로 할당되지 않도록 합니다. 또한 on 명령은 플래그를 제거합니다. 자세한 내용은 27페이지의 "논리 장치 번호(LUN)당 할당 제어"를 참조하십시오.

이러한 모든 명령은 `:command eq` 형식으로 사용됩니다. *eq*에는 장치의 장비 서수를 지정합니다.

## 파일 시스템 명령: I/O 관리

다음 명령을 사용하면 I/O 특징을 동적으로 관리할 수 있습니다.

### `:flush_behind eq value` 명령

`flush_behind` 명령은 최대 `flush_behind value`를 설정합니다. 0보다 큰 값을 설정한 경우, 연속으로 쓰여지면서 수정된 페이지는 디스크에 비동기적으로 쓰여져 Solaris 커널 레이어가 페이지를 깨끗하게 유지하도록 도와줍니다. 이 옵션은 최대 `flush_behind` 값을 설정합니다.

*value*에 대해  $0 \leq \text{value} \leq 8192$ 에 해당하는 정수의 KB 수를 지정합니다. 기본적으로 *value*=0이며, 이는 `flush_behind`를 비활성화합니다.

*eq*에는 파일 시스템의 장비 서수를 지정합니다.

### `:force_nfs_async eq` 및 `:noforce_nfs_async eq` 명령

이러한 명령을 사용하면 NFS에서 데이터를 동시에 디스크에 기록하도록 요청하더라도 파일 시스템이 서버에 기록된 NFS 데이터를 캐시할지 여부를 제어할 수 있습니다.

`force_nfs_async` 명령은 NFS 데이터를 캐시합니다. 기본값인 `noforce_nfs_async` 명령은 디스크에 데이터를 동시에 씁니다.

`force_nfs_async` 명령은 파일 시스템이 NFS 서버로서 마운트되고 클라이언트가 `noac` NFS 마운트 옵션을 사용하여 마운트될 경우에만 효과적입니다. NFS 파일 시스템 마운트에 대한 자세한 정보는 `mount_nfs(1M)` 매뉴얼 페이지를 참조하십시오.

*eq*에는 파일 시스템의 장비 서수를 지정합니다.



---

주의 - `force_nfs_async` 옵션은 NFS 프로토콜에 위배됩니다. 이 명령을 주의하여 사용하십시오. 서버 방해가 있는 경우, 데이터가 손상될 수 있습니다. 다중 NFS 서버가 있는 경우 데이터가 NFS 서버에 캐시되어 모든 클라이언트가 즉시 볼 수 없습니다. 다중 NFS 서버는 Sun StorageTek QFS 공유 파일 시스템 내에서 사용할 수 있습니다. Sun StorageTek QFS 공유 파일 시스템에 대한 자세한 정보는 67페이지의 "[Sun StorageTek QFS 공유 파일 시스템 구성](#)"을 참조하십시오.

---

### `:readahead eq contig` 명령

`readahead` 명령은 파일 시스템이 미리 읽을 수 있는 최대 바이트 수를 지정합니다.

*eq*에는 파일 시스템의 장비 서수를 지정합니다.

*contig*에는 1KB 블록의 단위를 지정합니다. 이것은  $1 < contig < 8192$  범위의 정수여야 합니다. 지정된 *contig*는 8KB의 배수로 잘립니다. 기본 *contig*는 8(131072바이트)입니다.

예를 들어 다음 명령은 장비 서수가 3으로 정의된 파일 시스템에 대해 최대 연속 블록 크기를 262,144바이트로 설정합니다.

```
:readahead 3 256
```

이 값은 또한 `readahead` 명령을 지정해 `samfs.cmd` 파일에 구성할 수도 있습니다. 자세한 내용은 `samfs.cmd(4)` 매뉴얼 페이지를 참조하십시오.

### `:sw_raid eq` 및 `:nosw_raid eq` 명령

이러한 옵션은 파일 시스템이 `writebehind` 버퍼를 정렬하는지 여부를 지정합니다. Solstice DiskSuite™와 같은 패키지의 소프트웨어 RAID 기능이 이 파일 시스템에서도 사용되는 경우 `sw_raid`를 지정합니다. 기본 설정은 `nosw_raid`입니다.

*eq*에는 파일 시스템의 장비 서수를 지정합니다.

## `:writebehind eq contig` 명령

`writebehind` 명령은 파일 시스템이 나중에 쓸 수 있는 최대 바이트 수를 지정합니다. `eq`는 파일 시스템의 장비 서수를 지정합니다.

`contig`에는 1KB 블록의 단위를 지정합니다. 이것은  $1 < contig < 8192$  범위의 정수여야 합니다. 기본 `contig`는 8(131072바이트)입니다.

예를 들어 다음 명령은 장비 서수가 50으로 정의된 파일 시스템에 대해 최대 연속 블록 크기를 262,144바이트로 설정합니다.

```
:writebehind 50 256
```

이 값은 또한 `writebehind` 명령을 지정해 `samfs.cmd` 파일에 구성할 수도 있습니다. 자세한 내용은 `samfs.cmd(4)` 매뉴얼 페이지를 참조하십시오.

## `:wr_throttle eq value` 명령

`wr_throttle` 명령은 한 파일에 대한 해결되지 않은 쓰기 바이트 수를 `value` KB로 설정합니다.

`eq`는 파일 시스템의 장비 서수를 지정합니다.

`value`에 대해 정수의 KB 수를 지정합니다. `value=0`인 경우, 제한이 없습니다. 기본 값은 16384입니다.

## 파일 시스템 명령: 직접 I/O 관리

이 절의 명령은 Sun StorageTek QFS 파일 시스템에서 I/O를 제어합니다. 이러한 명령을 사용하면 I/O 크기 및 내역을 기반으로 한 개별 파일에 대해 I/O 유형을 변경할 수 있습니다. 예를 들어 `setfa(1)` 명령을 사용하여 파일에 직접 I/O가 지정된 경우, 이러한 옵션은 무시되며 가능하면 일반 파일에 대한 모든 I/O는 직접적입니다.

이러한 명령은 올바르게 정렬된 I/O 및 잘못 정렬된 I/O 모두를 참조합니다. 올바르게 정렬된 I/O는 파일 오프셋이 512바이트 경계에 떨어질 때 및 I/O 전송 길이가 적어도 512바이트일 때 발생합니다. 잘못 정렬된 I/O는 파일 오프셋이 512바이트 경계로 떨어지지 않고 전송 길이가 512바이트 미만일 때 발생합니다.

I/O 및 I/O 관리에 대한 자세한 내용은 [179페이지의 "고급 항목"](#)을 참조하십시오.

`:dio_rd_form_min eq value` 및  
`:dio_wr_form_min eq value` 명령

이러한 명령은 올바르게 정렬된 I/O에 대한 하한을 *value* 1024바이트 블록으로 설정합니다. `:dio_rd_form_min` 명령을 사용하여 읽기에 *value*를 설정하고 `:dio_wr_form_min` 명령을 사용하여 쓰기에 *value*를 설정합니다.

*eq*에는 파일 시스템의 장비 서수를 지정합니다.

*value*에 대해 하위 제한으로 사용할 1024바이트 블록의 정수를 지정합니다. 기본적으로 *value*=256입니다. *value*=0인 경우, 자동 I/O 전환을 사용할 수 없습니다.

`:dio_rd_ill_min eq value` 및  
`:dio_wr_ill_min eq value` 명령

이러한 명령은 잘못 정렬된 I/O에 대한 하한을 *value* 1024바이트 블록으로 설정합니다. `:dio_rd_ill_min` 명령을 사용하여 읽기에 *value*를 설정하고 `:dio_wr_ill_min` 명령을 사용하여 쓰기에 *value*를 설정합니다.

*eq*에는 파일 시스템의 장비 서수를 지정합니다.

*value*에 대해 하위 제한으로 사용할 1024바이트 블록의 정수를 지정합니다. 기본적으로 *value*=256입니다. *value*=0인 경우, 자동 I/O 전환을 사용할 수 없습니다.

`:dio_rd_consec eq value` 및 `:dio_wr_consec eq value` 명령

이러한 명령은 지정된 하위 제한보다 큰 버퍼 크기를 가지고 발생할 수 있는 연속적인 I/O 전송 횟수를 *value* 작업으로 설정합니다.

*eq*에는 파일 시스템의 장비 서수를 지정합니다.

*value*에 대해 지정된 하위 제한보다 큰 버퍼 크기를 가진 연속적인 I/O 전송 횟수를 지정합니다. 지정된 하위 제한은 정렬된 읽기의 경우 `:dio_rd_form_min` 또는 잘못 정렬된 읽기의 경우 `:dio_rd_ill_min`의 *value*입니다. 기본적으로 *value*=0이며, 이는 I/O 크기를 기초로 기본 직접 읽기가 발생하지 않음을 의미합니다.

자세한 정보는 다음 명령 또는 마운트 매개변수 중 하나 이상을 참조하십시오.

- 310페이지의 "`:dio_rd_form_min eq value` 및 `:dio_wr_form_min eq value` 명령"
- 310페이지의 "`:dio_rd_ill_min eq value` 및 `:dio_wr_ill_min eq value` 명령"

## `:dio_szero eq` 및 `:nodio_szero eq` 명령

이들 명령은 직접 I/O sparse zeroing 마운트 옵션을 설정하거나 제거합니다.

`dio_szero` 옵션은 직접 I/O로 씌여진 희박한 파일의 초기화되지 않은 영역이 액세스 될 때 0이 되게 합니다. 이는 희박한 파일 작동을 페이지된 I/O의 경우와 동일하게 합니다. 기본적으로 직접 I/O에 의해 씌여진 희박한 파일은 성능상의 이유로 0이 되는 초기화되지 않은 영역을 갖지 않습니다. 기본값은 `nodio_szero`입니다.

`eq`에는 파일 시스템의 장비 서수를 지정합니다.

## `:forcedirectio eq` 및 `:noforcedirectio eq` 명령

이러한 명령을 사용하면 직접 I/O가 기본 I/O 모드로 사용되는지 여부를 제어할 수 있습니다. 기본적으로 I/O 모드가 버퍼되고 페이지 캐시를 사용합니다. `forcedirectio` 명령을 사용하면 모든 전송에 직접 I/O를 사용할 수 있습니다. `noforcedirectio` 명령은 버퍼된 I/O가 있는 기본값을 활성화합니다.

`eq`에는 파일 시스템의 장비 서수를 지정합니다.

직접 I/O가 지정되면 시스템은 사용자의 버퍼 및 디스크 사이에 직접 데이터를 전송합니다. 파일 시스템이 크고 블록이 정렬된 순차적 I/O에 사용되는 경우에만 직접 I/O를 사용하십시오.

I/O에 대한 자세한 내용은 [179페이지의 "고급 항목"](#)을 참조하십시오.

## 파일 시스템 명령: Sun StorageTek QFS 공유 파일 시스템

다음 파일 시스템 명령은 Sun StorageTek QFS 공유 파일 시스템에서만 지원됩니다.

### `:meta_timeo eq interval` 명령

`metatimeo` 명령은 Sun StorageTek QFS 공유 파일 시스템 메타 데이터 캐시 시간 초과값을 설정합니다. 이 기능의 사용에 대한 자세한 내용은 [250페이지의 "캐시된 속성 유지 :meta\\_timeo=n 옵션"](#)을 참조하십시오.

`eq`에는 파일 시스템의 장비 서수를 지정합니다.

`interval`에는 초 단위 시간 간격을 지정합니다. `interval` 기본값은 3입니다. 이 시간 간격이 종료되면 클라이언트 호스트 시스템은 메타 데이터 서버 호스트에서 새로운 메타 데이터 정보 복사본을 가져옵니다.

## :mh\_write *eq* 및 :nomh\_write *eq* 명령

이러한 명령은 다중호스트 읽기 및 쓰기를 활성화 또는 비활성화합니다. 이 기능에 대한 정보는 248페이지의 "다중 호스트 읽기 및 쓰기 활성화: mh\_write 옵션"을 참조하십시오.

*eq*에는 파일 시스템의 장비 서수를 지정합니다.

## :minallocsz *eq value* 및 :maxallocsz *eq value* 명령

이러한 명령은 최소 및 최대 블록 할당 크기를 설정합니다.

*eq*에는 파일 시스템의 장비 서수를 지정합니다.

*value* 및 이 기능에 대한 자세한 정보는 247페이지의 "할당 크기 조정: minallocsz=*n* 및 maxallocsz=*n* 옵션"을 참조하십시오.

## :rdlease *eq interval*, :wrlease *eq interval* 및 :aplease *eq interval* 명령

이러한 명령은 읽기, 쓰기 및 추가 임대에 허용된 시간을 조정합니다. 이 기능에 대한 정보는 247페이지의 "Sun StorageTek QFS 공유 파일 시스템에서 임대 사용: rdlease=*n*, wrlease=*n* 및 aplease=*n* 옵션"을 참조하십시오.

*eq*에는 파일 시스템의 장비 서수를 지정합니다.

*interval*에 대해 정수 초를 지정합니다. 세 가지 모든 임대를 사용하면  $15 \leq \textit{interval} \leq 600$ 에 해당하는 *interval*을 지정할 수 있습니다. 기본 *interval*은 30입니다.

## 파일 시스템 명령: 기타

다음 명령을 사용하면 임대, 할당 크기 및 여러 다른 파일 시스템 특징을 제어할 수 있습니다.

### :abr *eq* 및 :noabr *eq* 명령

이들 명령은 ABR(Application Binary Recovery) 마운트 옵션을 설정하거나 지웁니다.

Sun StorageTek QFS asynchronous I/O(AIO)를 갖는 Oracle RAC 환경에서만 사용됩니다. 이들 마운트 옵션은 소프트웨어 미러의 ABR을 비활성화 또는 활성화합니다. ABR을 지원하는 Solaris 볼륨 관리자 미러 볼륨에 작성된 Sun StorageTek QFS 파일 시스템에만 적용됩니다.

*eq*에는 파일 시스템의 장비 서수를 지정합니다.

## :dmr eq 및 :nodmr eq 명령

이들 명령은 DMR(Direct Mirror Reads) 마운트 옵션을 설정하거나 지웁니다.

Oracle Real Application Cluster(RAC) 환경에서 Sun StorageTek QFS AIO와 전용으로 사용하기 위한 것입니다. 이들 마운트 옵션은 소프트웨어 미러의 DMR을 비활성화 또는 활성화합니다. DMR을 지원하는 Solaris 볼륨 관리자 미러 볼륨에 작성된 Sun StorageTek QFS 파일 시스템에만 적용됩니다.

eq에는 파일 시스템의 장비 서수를 지정합니다.

## :invalid eq interval 명령

invalid 명령은 파일이 수정된 후 적어도 *interval*초 동안 파일 시스템이 캐시된 속성을 유지하도록 지정합니다. 파일 시스템이 원래 reader 마운트 옵션을 사용하여 마운트된 경우에만 이 명령을 지정할 수 있습니다. 마운트 옵션에 대한 정보는 mount\_samfs(1M) 매뉴얼 페이지를 참조하십시오.

eq에는 파일 시스템의 장비 서수를 지정합니다.

*interval*에 대해 파일 수정 후 해당 속성을 유지하도록 초 수를 지정합니다. 예를 들어, *interval*=30이라고 가정합니다. 해당 파일 시스템에서 1s(1) 명령을 실행하는 경우 해당 작성기 호스트에 작성된 후 30초 동안 새로 작성된 파일이 출력에 나타나지 않을 수도 있습니다.

## :mm\_stripe eq value 명령

mm\_stripe 명령은 파일 시스템에 대한 메타 데이터 스트라이프 너비를 *value* 16KB 디스크 할당 단위(DAU)로 설정합니다.

eq에는 파일 시스템의 장비 서수를 지정합니다.

*value*에 대해 0 또는 1을 지정합니다. 기본값 *value*=1인 경우, 파일 시스템은 다른 LUN으로 전환하기 전에 메타 데이터의 DAU 하나를 DAU 하나에 씁니다. *value*=0인 경우, 메타 데이터는 모든 사용 가능한 메타 데이터 LUN에 대해 라운드 로빈됩니다.

## :qwrite *eq* 및 :noqwrite *eq* 명령

qwrite 및 noqwrite 명령은 다른 스레드로부터 동일한 파일로 동시 읽기 및 쓰기를 수행하는 기능을 제어합니다. 파일 시스템 사용자가 동일한 파일로 다중 동시 전송을 처리하는 경우에만 qwrite를 지정합니다. 예를 들어 이것은 데이터베이스 애플리케이션에서 유용합니다. qwrite 기능은 드라이브 레벨에서 여러 요청을 대기열에 두어 I/O 성능을 향상시킵니다. qwrite 지정은 파일 시스템의 NFS 읽기 또는 쓰기에 사용할 수 없습니다.

기본 설정은 noqwrite이며 파일 시스템이 동일한 파일로 동시 읽기 및 쓰기를 비활성화합니다. 이것은 UNIX vnode 인터페이스 표준으로 정의된 모드로서 오직 하나의 쓰기 작업에만 독점적인 액세스 권한이 부여되고 다른 작성기 및 판독기 호스트는 대기해야 합니다.

*eq*에는 파일 시스템의 장비 서수를 지정합니다.

## :refresh\_at\_eof *eq* 및 :norefresh\_at\_eof *eq* 명령

refresh\_at\_eof 및 norefresh\_at\_eof 명령은 다중 판독기 파일 시스템의 reader 마운트 옵션을 사용하여 마운트된 Sun StorageTek QFS 호스트로 빠르게 업데이트하는 데 사용될 수 있습니다. 이 옵션을 사용하면 읽기 버퍼가 파일의 끝을 초과할 경우 파일 시스템이 현재 파일 크기를 새로 고칠 수 있습니다. 예를 들어 작성기 호스트 시스템이 파일에 추가되고 판독기가 -f 옵션이 있는 tail(1) 명령을 실행하는 경우, 이 옵션을 사용할 수 있습니다. 기본값은 norefresh\_at\_eof입니다.

*eq*에는 파일 시스템의 장비 서수를 지정합니다.

## :suid *eq* 및 :nosuid *eq* 명령

suid 및 nosuid 명령은 실행 중인 프로그램이 소유자 ID를 자동으로 변경하도록 허용되는지 여부를 제어합니다. 이러한 마운트 옵션 사용의 의미에 대한 자세한 정보는 mount\_ufs(1M) 매뉴얼 페이지의 suid 및 nosuid 마운트 옵션을 참조하고 suid(2) 매뉴얼 페이지를 참조하십시오.

*eq*에는 파일 시스템의 장비 서수를 지정합니다.

## :stripe eq value 명령

stripe 명령은 파일 시스템에 대한 스트라이프 너비를 *value* 디스크 할당 단위(DAU)로 설정합니다. 스트라이프 너비는 다음 LUN으로 전환하기 전에 DAU 바이트를 곱한 *value*가 LUN 하나에 기록되도록 지정합니다. `sammkfs(1M) -a` 명령을 사용하여 초기화될 때 파일 시스템에 DAU 크기를 설정할 수 있습니다.

*eq*에는 파일 시스템의 장비 서수를 지정합니다.

*value*에 대해  $0 < value < 255$ 에 해당하는 정수를 지정합니다. *value*=0인 경우 각 슬라이스에서 파일이 라운드 로빈됩니다. ms 장비 유형이 있는 파일 시스템 및 스트라이프 그룹(gXXX) 구성요소 없이 ma 장비 유형이 있는 파일 시스템의 기본값 *value*는 다음과 같습니다.

- DAU 당 128KB/DAU < 128KB
- DAU 당 1 > 128KB

기본적으로 Sun StorageTek QFS 공유 파일 시스템에서 *value*=0입니다.

기본적으로 모든 스트라이프 그룹(gXXX) 구성요소와 ma 장비 유형이 있는 파일 시스템에서 *value*=0입니다.

일치하지 않는 스트라이프 그룹이 있는 경우 시스템은 *value*=0을 설정합니다.

파일 시스템 유형에 대한 자세한 내용은 [7페이지의 "디자인의 기본"](#) 및 [29페이지의 "파일 시스템 구성"](#)을 참조하십시오.

## :sync\_meta eq value 명령

sync\_meta 명령은 메타 데이터가 변경될 때마다 디스크에 기록되는지 여부를 결정합니다. Sun StorageTek QFS 공유 파일 시스템에서 이 명령을 사용하려는 경우 또한 [251페이지의 "메타 데이터가 기록되는 빈도 지정: sync\\_meta=n 옵션"](#)을 참조하십시오.

*eq*에는 파일 시스템의 장비 서수를 지정합니다.

*value*에 대해 다음과 같이 0 또는 1을 지정합니다.

- *value*가 0인 경우 메타 데이터는 변경된 후 버퍼에 유지됩니다. 더 높은 성능이 요구되는 공유 해제된 Sun StorageTek QFS 파일 시스템의 경우, *value*를 0으로 설정할 수 있습니다. 이런 경우 시스템은 메타 데이터를 디스크에 쓰기 전에 버퍼에 유지하는 지연되는 쓰기를 수행합니다. 이것은 공유되지 않는 파일 시스템 및 다중 관독기 파일 시스템으로 마운트되지 않은 파일 시스템에 대한 기본값입니다.
- *value*가 1인 경우 메타 데이터는 변경될 때마다 디스크에 기록됩니다. 이렇게 되면 성능은 낮아지지만 데이터의 일관성은 증가합니다. 이것은 다중 관독기 파일 시스템 또는 공유 파일 시스템으로 마운트된 Sun StorageTek QFS 파일 시스템에 대한 기본값입니다. Sun StorageTek QFS 공유 파일 시스템에 대해 장애 조치 기능이 필요한 경우 *value*를 1로 설정해야 합니다.

## :trace eq 및 :notrace eq 명령

trace 명령은 파일 시스템에 대한 추적을 활성화합니다. notrace 명령은 추적을 비활성화합니다. 이들은 모든 작업에 영향을 주는 전역 명령입니다. 파일 시스템 추적에 대한 자세한 내용은 defaults.conf(4) 매뉴얼 페이지를 참조하십시오.

eq에는 파일 시스템의 장비 서수를 지정합니다.

## 기타 명령

다음 명령을 사용하여 추적을 제어하고 디스크 장치에 대한 액세스를 열고 여러 다른 기타 작업을 수행할 수 있습니다.

### :clear vsn [ index ] 명령

clear 명령은 제거 가능한 매체 마운트 요청 디스플레이에서 지정된 VSN을 지웁니다. 자세한 내용은 282페이지의 "(p) - 제거 가능한 매체 로드 요청 디스플레이"를 참조하십시오.

vsn에는 마운트할 볼륨을 지정합니다. VSN 마운트 대기 프로세스는 취소됩니다.

index에 대해, 제거 가능한 매체 디스플레이에서 VSN의 십진 서수를 지정합니다.

### :devlog eq [ option ] 명령

devlog 명령은 하나 이상의 이벤트가 기록되도록 설정합니다.

eq에는 장치의 장비 서수를 지정합니다.

option에 대해 하나 이상의 이벤트 유형을 지정합니다. 가능한 이벤트 유형은 다음과 같습니다. all, date, default, detail, err, event, label, mig, module, msg, none, retry, stage, syserr 및 time. 이러한 옵션에 대한 내용은 defaults.conf(4) 매뉴얼 페이지를 참조하십시오. option이 지정되지 않은 경우 시스템은 지정된 eq에 대해 로그되는 현재 이벤트를 변경하지 않습니다.

### :diskvols volume [+flag | -flag] 명령

diskvols 명령은 디스크 볼륨 사전의 플래그를 설정하거나 제거합니다.

volume에 대해 디스크 볼륨 사전의 볼륨을 지정합니다.

flag에 대해 D samu(1M) 디스플레이의 5개 플래그 중 하나를 지정합니다. 디스크 볼륨 사전 및 플래그에 대한 내용은 264페이지의 "(D) - 디스크 볼륨 사전" 또는 samu(1M) 매뉴얼 페이지를 참조하십시오.

## :dtrace 명령

dtrace 명령은 다음과 같습니다.

- :dtrace *daemon\_name* on
- :dtrace *daemon\_name* off
- :dtrace *daemon\_name.variable value*

dtrace 명령은 다양한 추적 옵션을 지정합니다. 표 C-40은 추적 제어 명령 인수를 표시합니다.

표 C-40 추적 명령 인수

인수	설명
<i>daemon_name</i>	키워드 <code>all</code> 또는 프로세스 이름을 지정합니다. 키워드 <code>all</code> 이 지정된 경우에는 추적 명령이 모든 데몬에 영향을 줍니다. 다음 프로세스 이름 중 하나가 지정된 경우, 추적 명령은 해당 프로세스에만 영향을 줍니다. <code>sam-archiverd</code> , <code>sam-catserverd</code> , <code>sam-fsd</code> , <code>sam-rftd</code> , <code>sam-recycler</code> , <code>sam-sharefsd</code> 및 <code>sam-stagerd</code> . 키워드 <code>on</code> 또는 <code>off</code> 중 하나를 프로세스 이름 다음에 지정할 수 있습니다. <code>on</code> 또는 <code>off</code> 가 지정된 경우에는 지정된 모든 프로세스에 대해 추적이 활성화 또는 비활성화됩니다.
<i>variable value</i>	여러 가지 <i>variable</i> 및 <i>value</i> 인수를 지정할 수 있습니다. <code>defaults.conf(4)</code> 매뉴얼 페이지에는 이러한 인수에 대한 자세한 내용이 나와있습니다. 다음 <i>variable</i> 및 <i>value</i> 조합 중 하나를 지정합니다. <ul style="list-style-type: none"><li>• <i>file value.value</i>에는 추적 파일을 쓸 수 있는 파일 이름을 지정합니다. 전체 경로 이름을 지정할 수 있습니다.</li><li>• <i>options value.value</i>에는 공백으로 구분된 추적 옵션 목록을 지정합니다.</li><li>• <i>age value.age</i>에는 추적 파일 순환 나이를 지정합니다. 참고: 이 값을 2분 이하로 설정하지 마십시오. 그렇게 하면 순환이 절대 발생하지 않습니다.</li><li>• <i>size value.value</i>에는 순환을 시작할 추적 파일의 크기를 지정합니다.</li></ul>

## :fs *fsname* 명령

`fs` 명령은 파일 시스템이 N 디스플레이를 통해 표시되도록 설정합니다.

*fsname*에 대해 검사할 파일 시스템의 이름을 지정합니다.

## :mount *mntpt* 명령

`mount` 명령은 Sun StorageTek QFS 파일 시스템을 선택합니다. *mntpt*에는 파일 시스템의 마운트 지점을 지정합니다.

### :open *eq* 명령

open 명령은 지정된 디스크 장치에 대한 액세스를 활성화합니다. 이 명령을 실행해야 read 명령, 디스크 섹터 디스플레이(S) 또는 파일 레이블 디스플레이(F)를 사용할 수 있습니다.

*eq*에는 장치의 장비 서수를 지정합니다.

### :read *addr* 명령

read 명령은 현재 열린 디스크 장치에서 지정된 섹터를 읽습니다. 이 섹터를 읽으려면 먼저 장치를 열어야 합니다.

*addr*에는 16진수 섹터 주소를 지정합니다.

### :refresh *i* 명령

refresh 명령은 samu(1M) 화면 새로고침 사이의 시간을 결정합니다.

*i*에는 초 단위로 시간을 지정합니다.

### :snap [ *filename* ] 명령

snap 명령은 디스플레이 창의 스냅샷을 디스플레이 정보를 받는 파일의 이름인 *filename*으로 보냅니다.

문제 보고를 용이하게 하도록 모든 samu(1M) 유틸리티 디스플레이의 스냅샷을 만들 수 있습니다. 각각의 새로운 스냅샷은 스냅샷 파일에 추가됩니다. 기본 파일은 현재 실행 중인 디렉토리의 snapshots입니다. 파일은 인쇄하거나 vi(1)를 사용해 검사하거나 Sun Microsystems 고객 지원부에 팩스로 전송할 수 있습니다.

### :! *shell\_command* 명령

! 명령을 사용하면 samu(1M) 운영자 유틸리티 내에서도 셸 명령을 실행할 수 있습니다.

# 용어 해설

---

## 자모

- DAU** 디스크 할당 단위. 온라인 저장소의 기본 단위. 블록 크기라고도 합니다.
- FDDI** FDDI(Fiber-distributed Data Interface)는 최고 200km(124마일)까지 범위를 확장할 수 있는 근거리 통신망의 데이터 전송 표준입니다. FDDI 프로토콜은 토큰 링 프로토콜을 기초로 합니다.
- Fibre Channel** 장치간에 고속의 직렬 통신을 지정하는 ANSI 표준. Fibre Channel은 SCSI-3에서 버스 아키텍처 중 하나로 사용됩니다.
- FTP** FTP(File Transfer Protocol). TCP/IP 네트워크를 통해 두 호스트 사이에 파일을 전송하기 위한 인터넷 프로트콜
- hosts 파일** 호스트 파일은 공유 파일 시스템에 있는 모든 호스트 목록을 포함하고 있습니다. 파일 시스템을 Sun StorageTek QFS 공유 파일 시스템으로 초기화하고 있는 경우, 호스트 파일 /etc/opt/SUNWsamfs/hosts.*fs-name*은 파일 시스템이 생성된 후에 만들어야 합니다. **sammkfs(1M)** 명령은 파일 시스템 생성 시에 호스트 파일을 사용합니다. **samsharefs(1M)** 명령을 사용하여 나중에 호스트 파일의 내용을 바꾸거나 업데이트할 수 있습니다.
- inode** 인덱스 노드(index node). 파일을 기술하기 위해 파일 시스템에 의해 사용되는 데이터 구조. **inode**는 이름 이외의 파일과 관련된 모든 속성을 기술합니다. 속성은 소유권, 액세스, 권한, 크기 및 디스크 시스템에서 파일 위치가 포함됩니다.
- inode 파일** 파일 시스템에 상주하는 모든 파일에 대한 **inode** 구조를 포함하는 특수한 파일 (.inodes). **Inode** 길이는 512바이트입니다. **Inode** 파일은 파일 시스템의 파일 데이터에서 분리된 메타 데이터 파일입니다.
- LAN** 근거리 통신망(Local Area Network)
- LUN** 논리 장치 번호(Logical Unit Number)

<b>mcf</b>	마스터 구성 파일. 파일 시스템 환경 내에서 장치 사이의 관계(토폴로지)를 정의하는 초기화 시 읽게 되는 파일
<b>NFS</b>	네트워크 파일 시스템(Network File System). 이기종 네트워크 환경에서 원격 파일 시스템에 대한 투명한 액세스를 제공하는 Sun의 분산 파일 시스템
<b>NIS</b>	SunOS 4.0(최소) Network Information Service. 네트워크에서 시스템 및 사용자에 대한 주요 정보를 포함하고 있는 분산 네트워크 데이터베이스. NIS 데이터베이스는 주 서버(master server) 및 모든 종 서버(slave server)에 저장됩니다.
<b>RAID</b>	독립된 디스크의 중복 배열(Redundant Array of Inexpensive/Independent Disks). 파일을 안정적으로 저장하기 위해 여러 독립 디스크를 사용하는 디스크 기술. 단일 디스크 장애로 인한 데이터 손실로부터 보호하고, 결함을 해결하는 디스크 환경을 제공하며, 개별 디스크보다 더 높은 처리량을 제공합니다.
<b>RPC</b>	원격 절차 호출(Remote Procedure Call). 사용자 정의 네트워크 데이터 서버를 구현하기 위해 NFS에 의해 사용되는 기본 데이터 교환 메커니즘
<b>SAM-QFS</b>	Sun StorageTek SAM 소프트웨어를 Sun StorageTek QFS 파일 시스템과 결합하는 구성. SAM-QFS는 저장소 및 아카이브 관리 유틸리티와 함께 사용자 및 관리자에게 고속의 표준 UNIX 파일 시스템 인터페이스를 제공합니다. 표준 UNIX 파일 시스템 명령 뿐만 아니라 Sun StorageTek SAM 명령 세트에서 사용할 수 있는 많은 명령을 사용합니다.
<b>samfsdump</b>	컨트롤 구조 덤프를 만들고 해당하는 파일 그룹에 대한 모든 컨트롤 구조 정보를 복사하는 프로그램. UNIX tar(1) 유틸리티와 유사하지만, 일반적으로 파일 데이터를 복사하지는 않습니다. samfsrestore도 참조하십시오.
<b>samfsrestore</b>	컨트롤 구조 덤프로부터 inode 및 디렉토리 정보를 복원하는 프로그램. samfsdump도 참조하십시오.
<b>SCSI</b>	소형 컴퓨터 시스템 인터페이스(Small Computer System Interface). 디스크 및 테이프 드라이브, 자동화된 라이브러리 등과 같은 주변 장치에 대해 일반적으로 사용되는 전기 통신 사양
<b>Sun SAM-Remote</b>	
서버	전기능 Sun StorageTek SAM 저장소 관리 서버 및 Sun SAM-Remote 클라이언트 사이에 공유될 라이브러리를 정의하는 Sun SAM-Remote 서버 데몬
<b>Sun SAM-Remote</b>	
클라이언트	많은 가상 장치를 포함하며 자체 라이브러리 장치도 가질 수 있는 클라이언트 데몬을 갖는 Sun StorageTek SAM 시스템. 클라이언트는 하나 이상의 아카이브 복사본을 위해 아카이브 매체의 Sun SAM-Remote 서버에 의존합니다.
<b>tar</b>	테이프 아카이브(tape archive). 아카이브 이미지에 사용되는 표준 파일 및 데이터 기록 형식
<b>TCP/IP</b>	전송 컨트롤 프로토콜/인터넷 프로토콜(Transmission Control Protocol/Internet Protocol). 호스트간 주소 지정 및 라우팅, 패킷 전달(IP) 및 애플리케이션 지점간의 데이터 전달(TCP)을 담당하는 인터넷 프로토콜

- vfstab 파일** vfstab 파일은 파일 시스템에 대한 마운트 옵션을 포함하고 있습니다. 명령줄에 지정된 마운트 옵션은 /etc/vfstab 파일에 지정된 내용보다 우선하지만 /etc/vfstab 파일에 지정된 마운트 옵션은 samfs.cmd 파일에 지정된 내용보다 우선합니다.
- VSN** 볼륨 시리얼 이름(Volume Serial Name). 제거 가능한 매체 카트리지에 아카이브하는 경우, VSN은 볼륨 레이블에 쓰여지는 자기 테이프 및 광 디스크에 대한 논리적 식별자입니다. 디스크 캐시에 아카이브하는 경우, VSN은 디스크 아카이브 세트에 대한 고유한 이름입니다.
- WORM** 한 번 쓰기, 여러 번 읽기(Write Once Read Many). 한 번만 쓸 수 있지만 여러 번 읽을 수 있는 매체에 대한 저장소 유형

## 7

- 가상 장치** 연결된 하드웨어가 없는 소프트웨어 하위 시스템 또는 드라이버
- 간접 블록** 저장소 블록의 목록을 포함하는 디스크 블록. 파일 시스템에는 최고 세 레벨의 간접 블록이 있습니다. 첫 번째 레벨 간접 블록은 데이터 저장에 사용되는 블록 목록을 포함합니다. 두 번째 레벨 간접 블록은 첫 번째 레벨 간접 블록 목록을 포함합니다. 세 번째 레벨 간접 블록은 두 번째 레벨 간접 블록 목록을 포함합니다.
- 감사(전체)** VSN을 확인하기 위해 카트리지를 로드하는 프로세스. 광자기 카트리지의 경우, 용량 및 공간 정보가 파악되고 자동화된 라이브러리의 카탈로그에 입력됩니다. VSN도 참조하십시오.
- 공유 호스트 파일** 공유 파일 시스템을 생성할 때 시스템은 호스트 파일에서 메타 데이터 서버의 공유 호스트 파일로 정보를 복사합니다. samsharefs(1M) -u 명령을 사용할 때 이 정보를 업데이트합니다.
- 근거리 저장소** 액세스하기 위해 로봇에 의한 마운트를 필요로 하는 제거 가능한 매체 저장소. 근거리 저장소는 일반적으로 온라인 저장소보다 가격이 저렴하지만, 더 많은 액세스 시간을 필요로 합니다.

## 8

네트워크로 연결된  
자동화된 라이브러리

StorageTek, ADIC/Grau, IBM, Sony 등 벤더에서 제공한 소프트웨어 패키지를 사용하여 제어되는 라이브러리. Sun StorageTek SAM 파일 시스템은 자동화된 라이브러리로 특별히 디자인된 Sun StorageTek SAM 매체 교환기 데몬을 사용하는 업체 소프트웨어와 인터페이스합니다.

---

## ㄷ

다중 판독기 파일 시스템	다중 호스트에 마운트될 수 있는 파일 시스템을 지정할 수 있는 단일 작성기, 다중 판독기 기능. 여러 호스트가 파일 시스템을 읽을 수 있지만, 하나의 호스트만 파일 시스템에 쓸 수 있습니다. 다중 판독기는 <code>mount(1M)</code> 명령에서 <code>-o reader</code> 옵션으로 지정됩니다. 단일 작성기 호스트는 <code>mount(1M)</code> 명령에서 <code>-o writer</code> 옵션으로 지정됩니다. <code>mount(1M)</code> 명령에 대한 자세한 내용은 <code>mount_samfs(1M)</code> 매뉴얼 페이지를 참조하십시오.
데이터 장치	파일 시스템에서, 파일 데이터가 저장되는 장치 또는 장치 그룹
드라이브	제거 가능한 미디어 볼륨 사이에 데이터를 전송하기 위한 메커니즘
디렉토리	파일 시스템 내에서 다른 파일 및 디렉토리를 가리키는 파일 데이터 구조
디스크 공간 임계값	관리자가 정의하는 디스크 캐시 사용률의 최대 또는 최소 수준. 릴리서는 이와 같이 미리 정의된 디스크 공간 임계값을 기준으로 디스크 캐시 사용량을 제어합니다.
디스크 버퍼	<code>Sun SAM-Remote</code> 구성에서, 클라이언트에서 서버로 데이터 아카이브에 사용되는 서버 시스템의 버퍼
디스크 스트라이프	여러 디스크에 걸쳐 파일을 기록하는 프로세스로, 액세스 성능이 높아지고 전체적인 저장 용량이 증가합니다. 스트라이프도 참조하십시오.
디스크 캐시	온라인 디스크 캐시와 아카이브 매체 사이에서 데이터 파일 작성 및 관리에 사용되는 파일 시스템 소프트웨어의 디스크 상주 부분. 개별 디스크 분할 영역 또는 전체 디스크가 디스크 캐시로 사용될 수 있습니다.
디스크 할당 단위	<code>DAU</code> 를 참조하십시오.

---

## ㄴ

라운드 로빈	전체 파일이 순차적으로 논리 디스크에 쓰여지는 데이터 액세스 방식. 단일 파일이 디스크에 쓰여질 때 전체 파일이 첫 번째 논리 디스크에 쓰여집니다. 두 번째 파일은 그 다음 논리 디스크에 쓰여지는 방식으로 수행됩니다. 각 파일의 크기는 <code>I/O</code> 의 크기를 결정합니다.  디스크 스트라이프 및 스트라이프도 참조하십시오.
라이브러리	자동화된 라이브러리를 참조하십시오.
라이브러리 카탈로그	카탈로그를 참조하십시오.

로봇	저장소 슬롯과 드라이브 사이에 카트리지를 옮기는 자동화된 라이브러리의 일부. 전송 장치라고도 합니다.
로컬 파일 시스템	Sun Cluster 시스템의 한 노드에 설치되고 또 다른 노드에 크게 사용할 수 없는 파일 시스템. 또한 서버에 설치된 파일 시스템
리사이클러	만료된 아카이브 복사본이 차지하는 카트리지의 공간을 재생하는 Sun StorageTek SAM 유틸리티
릴리서	아카이브된 파일을 식별하고 해당 디스크 캐시 복사본을 릴리스해서 더 많은 디스크 캐시 공간을 사용할 수 있게 하는 Sun StorageTek SAM 구성요소 릴리서는 온라인 디스크 저장소의 양을 상한 및 하한 임계값으로 자동 조절합니다.
릴리스 우선 순위	파일 시스템의 파일이 아카이브된 후 해제되는 우선순위. 릴리스 우선순위는 파일 등록 정보의 다양한 가중치를 곱한 후 결과를 더하여 계산됩니다.

## □

마운트 지점	파일 시스템이 마운트되는 디렉토리
매체	테이프 또는 광 디스크 카트리지
매체 리사이클	적게 사용하는 아카이브 매체 리사이클 또는 재사용 프로세스
메타 데이터	데이터에 대한 데이터. 메타 데이터는 디스크에서 파일의 정확한 데이터 위치를 찾는 데 사용되는 인덱스 정보입니다. 파일, 디렉토리, 액세스 제어 목록, 심볼 링크, 제거 가능한 매체, 세그먼트된 파일 및 세그먼트된 파일의 인덱스에 대한 정보로 구성됩니다.
메타 데이터 장치	파일 시스템 메타 데이터가 저장되는 장치(예: 독립적으로 작동하는 디스크 또는 미러 장치). 파일 데이터 및 메타 데이터를 별도의 장치에 보관하면 성능이 향상될 수 있습니다. mcf 파일에서 메타 데이터 장치는 ma 파일 시스템 내에서 mm장치로 선언됩니다.
미러 쓰기	단일 디스크 장애 발생 시 데이터 손실을 막기 위해 별도의 디스크 세트에 두 개의 파일 복사본을 유지하는 프로세스

---

## ㄷ

백업 저장소	예기치 않은 손실을 대비하기 위한 파일 모음의 스냅샷. 백업에는 파일의 속성 및 관련 데이터가 모두 포함됩니다.
볼륨	데이터 공유를 위한 카트리지에서 이름이 지정된 영역. 카트리지는 하나 이상의 볼륨을 포함할 수 있습니다. 두 면이 있는 카트리지는 각 면에 하나씩 두 개의 볼륨을 가집니다.
볼륨 오버플로	시스템이 여러 볼륨에 걸쳐 단일 파일을 분산시킬 수 있는 용량. 볼륨 오버플로는 개별 카트리지의 용량을 초과하는 매우 큰 용량의 파일을 사용하는 사이트에서 유용합니다.
분할 영역	장치의 일부 또는 광자기 카트리지의 한 면
블록 크기	DAU를 참조하십시오.
블록 할당 맵	디스크에서 사용 가능한 저장소의 블록 및 해당 블록이 사용 중이거나 사용 가능한지의 여부를 나타내는 비트맵

---

## ㄸ

사전 할당	파일에 쓰기 위해 디스크 캐시에서 연속되는 공간을 확보하는 프로세스. 사전 할당은 크기가 0인 파일의 경우에만 지정할 수 있습니다. 자세한 내용은 <code>setfa(1)</code> 매뉴얼 페이지를 참조하십시오.
소프트 제한	디스크 할당량에서 일시적으로 초과할 수 있는 파일 시스템 자원(블록 및 inode)에 대한 임계값 한도. 소프트 제한이 초과되면 타이머가 시작됩니다. 지정된 시간 동안 소프트 제한을 초과한 경우에는 소프트 제한보다 낮게 파일 시스템 사용량을 줄일 때까지 더 이상 시스템 자원을 할당할 수 없습니다.

### 소형 컴퓨터 시스템 인터페이스(Small Computer System Interface)

SCSI를 참조하십시오.	
수퍼 블록	파일 시스템의 기본적인 매개 변수를 정의하는 파일 시스템의 데이터 구조. 수퍼 블록은 저장소 패밀리 세트의 모든 분할 영역에 쓰여지고 이러한 세트에서 분할 영역의 구성원을 식별합니다.
스테이징	근거리 파일 또는 오프라인 파일을 아카이브 저장소에서 다시 온라인 저장소로 복사하는 프로세스

스트라이프	파일이 인터레이스 방식으로 논리 디스크에 동시에 쓰여지는 데이터 액세스 방식. SAM-QFS 파일 시스템은 스트라이프 그룹을 사용하는 "하드 스트라이프"와 stripe=x 마운트 매개변수를 사용하는 "소프트 스트라이프"의 두 유형의 스트라이프를 제공합니다. 하드 스트라이프는 파일 시스템이 설정될 때 활성화되며 mcf 파일 내에 스트라이프 그룹이 정의되어야 합니다. 소프트 스트라이프는 stripe=x 마운트 매개변수를 통해 활성화되며 파일 시스템 또는 개별 파일에 대해 변경될 수 있습니다. stripe=0을 설정하면 비활성화됩니다. 파일 시스템이 동일한 수의 요소를 갖는 다중 스트라이프 그룹으로 구성되는 경우 하드 및 소프트 스트라이프를 둘 다 사용할 수 있습니다. 라운드 로빈도 참조하십시오.
스트라이프 그룹	mcf(4) 파일에서 하나 이상의 gXXX 장치로 정의되는 파일 시스템 내의 장치 모음. 스트라이프 그룹은 하나의 논리적 장치로 취급되고 언제나 디스크 할당 단위(DAU)와 동일한 크기로 스트라이프됩니다.
스트라이프 크기	쓰기가 다음 스트라이프 장치로 이동하기 전에 할당할 디스크 할당 단위(DAU)의 수. stripe=0 마운트 옵션이 사용되는 경우, 파일 시스템은 스트라이프 액세스가 아닌 라운드 로빈 액세스를 사용합니다.



아카이버	파일 복사를 제거 가능한 카트리지로 자동 제어하는 아카이브 프로그램
아카이브 매체	아카이브 파일이 쓰여지는 매체. 아카이브 매체는 라이브러리에서 제거 가능한 테이프 또는 광자기 카트리지가 될 수 있습니다. 또한 아카이브 매체는 다른 시스템에서 마운트 지점이 될 수 있습니다.
아카이브 저장소	아카이브 매체에 만들어진 파일 데이터의 복사본
연결	안정적인 스트림 전달 서비스를 제공하는 두 개의 프로토콜 모듈 사이의 경로. TCP 연결은 한 컴퓨터의 TCP 모듈에서 다른 컴퓨터의 TCP 모듈로 확장됩니다.
오프라인 저장소	로드를 위해 운영자의 간섭이 필요한 저장소
온라인 저장소	즉시 사용이 가능한 저장소(예: 디스크 캐시 저장소)
외부 사이트 저장소	서버와 떨어져 있고, 재난 복구를 위해 사용되는 저장소
원격 절차 호출	RPC를 참조하십시오.
위치 배열	파일에 지정되는 각 데이터 블록의 디스크 위치를 정의하는 파일의 inode 내의 배열
유예 기간	디스크 할당량의 경우 사용자가 파일을 작성하고 소프트 한계에 도달한 후 저장소를 할당하도록 허용되는 시간
이더넷	근거리, 패킷 스위칭 네트워크 기술. 원래는 동축 케이블용으로 개발되었으며, 현재는 STP(shielded twisted-pair) 케이블을 통해 사용되고 있습니다. 이더넷은 초당 10 또는 100메가바이트 LAN입니다.

이름 공간	파일, 해당 속성 및 해당 저장 위치를 식별하는 파일 모음의 메타 데이터 부분
임대	지정된 시간 동안 파일에 조작을 수행할 클라이언트 호스트 권한을 부여하는 기능. 메타 데이터 서버는 각 클라이언트 호스트에게 임대를 부여합니다. 파일 작업을 계속 수행할 수 있도록 필요에 따라 임대를 갱신할 수 있습니다.

## 入

자동화된 라이브러리	운영자의 간섭 없이 제거 가능한 매체 카트리지를 자동으로 로드 및 언로드하기 위해 설계된 로봇 제어 장치. 자동화된 라이브러리에는 하나 이상의 드라이브와 카트리지를 저장소 슬롯 및 드라이브로 이동하는 전송 장치가 포함됩니다.
장치 로그	장치 문제 분석에 사용되는 장치별 오류 정보를 제공하는 구성 가능한 기능
장치 스캐너	수동으로 마운트되는 모든 제거 가능한 장치의 존재 유무를 정기적으로 모니터링하고 사용자 또는 기타 프로세스에 의해 요구될 수 있는 마운트된 카트리지의 존재 유무를 감지하는 소프트웨어
저장소 슬롯	카트리가 드라이브에서 사용되지 않을 때 저장되는 자동화된 라이브러리 내부의 위치
저장소 패밀리 세트	단일 디스크 패밀리 장치로 집합적으로 표현되는 디스크 세트
전역 명령	모든 파일 시스템에 적용되고 첫 번째 fs = 행 앞에 나타나는 아카이브 및 릴리스 명령
제거 가능한 매체 파일	자기 테이프 또는 광 디스크 카트리지와 같은 제거 가능한 매체 카트리지에서 직접 액세스할 수 있는 특수한 유형의 사용자 파일. 아카이브 및 스테이지 파일 데이터를 쓰는 데도 사용됩니다.
지정 가능한 저장소	Sun StorageTek QFS 또는 Sun StorageTek SAM 파일 시스템을 통해 사용자가 참조하는 온라인, 근거리, 오프사이트 및 오프라인 저장소를 포함한 저장 공간
직접 I/O	대형 블록이 정렬된 순차적 I/O에 사용된 속성. setfa(1) 명령의 -D 옵션은 직접 I/O 옵션입니다. 이 옵션은 파일 또는 디렉토리에 대해 직접 I/O 속성을 설정합니다. 디렉토리에 적용되면 직접 I/O 속성이 상속됩니다.
직접 액세스	근거리 파일을 디스크 캐시로 가져올 필요 없이 아카이브 매체에서 바로 액세스할 수 있는 파일 속성(전혀 스테이지되지 않음)
직접 연결된 라이브러리	SCSI 인터페이스를 사용하여 서버에 직접 연결된 자동화된 라이브러리. SCSI 연결 라이브러리는 Sun StorageTek SAM 소프트웨어에 의해 직접 제어됩니다.

---

## ㄱ

- 카탈로그** 자동화된 라이브러리에서 VSN 레코드. 각 자동화된 라이브러리에는 하나의 카탈로그가 있고, 사이트에는 모든 자동화된 라이브러리에 대한 하나의 기록자가 있습니다. VSN도 참조하십시오.
- 카트리지** 테이프나 광 디스크와 같이 데이터 기록용 매체가 들어있는 물리적 엔티티. 하나의 매체, 볼륨 또는 미디어라고도 합니다.
- 커널** 기본적인 시스템 장치를 제공하는 중앙 제어 프로그램. UNIX 커널은 프로세스 작성 및 관리, 파일 시스템 액세스 기능 제공, 일반적인 보안 제공, 통신 장치 제공 등을 수행합니다.
- 클라이언트-서버** 한 사이트의 프로그램에서 다른 사이트의 프로그램에 요청을 보내고 응답을 기다리는 분산 시스템의 상호 작용 모델. 요청하는 프로그램을 클라이언트라고 합니다. 응답을 제공하는 프로그램을 서버라고 합니다.

---

## ㄷ

- 타이머** 사용자가 소프트웨어 제한에 도달하는 시간부터 사용자에게 부여된 하드 제한 사이에 경과된 시간을 추적하는 할당량 소프트웨어

---

## ㄴ

- 파일 시스템** 파일 및 디렉토리의 계층적 모음
- 파일 시스템별 명령** archiver.cmd 파일에서 전역 명령 다음에 오는 아카이버 및 릴리셔 명령은 특정 파일 시스템에 따라 다르고 fs =로 시작됩니다. 파일 시스템별 명령은 다음 fs = 명령줄이 오거나 파일의 끝에 도달할 때까지 적용됩니다. 여러 명령이 파일 시스템에 영향을 미칠 경우, 파일 시스템별 명령은 전역 명령보다 우선합니다.
- 패밀리 세트** 디스크 모음이나 자동화된 라이브러리 내의 드라이브와 같이 독립적인 물리적 장치의 그룹으로 표현되는 저장 장치. 또한 저장소 패밀리 세트를 참조하십시오.
- 패밀리 장치 세트** 패밀리 세트를 참조하십시오.

---

## ㅎ

하드 제한	디스크 할당량에서 사용자가 초과할 수 없는 파일 시스템 자원, 블록 및 inode에 대한 최대 한도
할당량	사용자가 사용할 수 있는 시스템 자원의 양

# 색인

---

## 심볼

! samu(1M) 명령, 318  
.inode 파일, 3

## 숫자

0 할당량, 104  
64비트 주소지정, 2

## A

abr samu(1M) 명령, 312  
abr 마운트 옵션, 123  
Additional parameters 필드, 38  
alloc 명령, 27  
API 루틴, 6, 50  
aplease  
    samu(1M) 명령, 312  
    마운트 옵션, 247  
archive(1) 명령, 49

## B

bg 마운트 옵션, 246

## C

chmod 4000 WORM 트리거, 190  
chmod -w WORM trigger, 192  
clear samu(1M) 명령, 316

## D

DAU, 6  
    md 장치, 10  
    mr 또는 gXXXX, 10  
    개요, 8  
    단일 할당 체계, 10  
    설정, 8  
    이중 할당 체계, 9  
    할당량 및 DAU, 97  
dd(1M) 명령, 99, 102  
def\_retention 마운트 옵션, 252  
defaults.conf 파일  
    변경 사항 전과, 54  
    추적 활성화, 238  
device state 필드, 38  
devlog samu(1M) 명령, 316  
DID 지원, 122  
dio\_rd\_consec  
    samu(1M) 명령, 310  
    마운트 옵션, 214  
dio\_rd\_form\_min samu(1M) 명령, 310  
dio\_rd\_ill\_min samu(1M) 명령, 310

dio\_szero samu(1M) 명령, 311  
dio\_wr\_consec  
  samu(1M) 명령, 310  
  마운트 옵션, 214  
dio\_wr\_form samu(1M) 명령, 310  
dio\_wr\_ill\_min samu(1M) 명령, 310  
directio(3C) 함수 호출, 2, 213  
diskvols samu(1M) 명령, 316  
diskvols.conf 파일  
  samu(1M) 디스플레이, 264  
dmr samu(1M) 명령, 313  
dmr 마운트 옵션, 123  
down samu(1M) 명령, 306  
dtrace samu(1M) 명령, 317

## E

EDOM 오류, 223  
EDQUOT 오류, 97  
EFI 레이블  
  Linux 클라이언트, 242  
emul\_lite 마운트 옵션(WORM), 187  
ENOCSE 오류, 223  
EVMS(Enterprise Volume Management System), 242

## F

File System Manager  
  개요, 29  
  계정 작성, 30  
  관리 서버 추가, 33  
  권한 수준 지정, 31  
  사용, 30  
  세션 시간 초과 설정, 34  
flush\_behind  
  samu(1M) 명령, 307  
  마운트 옵션, 219  
force\_nfs\_async samu(1M) 명령, 307  
forcedirectio  
  samu(1M) 명령, 311  
  마운트 옵션, 213

fs samu(1M) 명령, 317  
fsck(1M) 명령, 또한 samfsck(1M) 명령 참조, 3  
fsmadm(1M) 명령, 33

## G

gXXX 장치, 37

## H

HA-NFS  
  구성, 148, 160  
  비활성화, 174, 175, 176  
HA-SAM, 178  
HAStoragePlus 자원 유형, 140, 148  
hostname(1M) 명령, 237  
hosts 파일, 71, 76  
  Sun Cluster 사용, 136  
  예제, 86  
  호스트 삭제, 81  
hosts.*fsname*.local file, 72, 85, 86

## I

I/O  
  전환, 214  
  조정, 213  
  직접, 2, 213  
  직접 I/O samu(1M) 명령, 309  
  직접 I/O 파일 속성, 50  
  페이지된, 2, 215, 216  
idle samu(1M) 명령, 306  
inode  
  samu(1M) 디스플레이, 269  
  속성, 49  
  파일 내용, 7  
inodes 파일, 3  
inodes 해시 테이블, 조정, 219  
invalid samu(1M) 명령, 313  
IPMP  
  어댑터, 147  
  유효성 검사, 145

## L

- Linux 클라이언트, 5
  - FAQ, 242
  - 로그 파일, 240
  - 마운트 옵션, 243
  - 문제 해결, 240
  - 지원되는 OS 버전, xxv
- LogicalHost 자원 유형, 174
- ls(1) 명령, 또한 sfs(1) 명령 참조, 51
- LUN
  - LUN별 할당 제어, 27
- LUN당 할당 제어, 27
- LVM(Logical Volume Management), 242

## M

- ma 파일 시스템, 9, 36
- maxallocsz
  - samu(1M) 명령, 312
  - 마운트 옵션, 247
- maxphys 매개변수, 214
- mcf 파일
  - /dev/dsk 항목, 35
  - /dev/rmt 항목, 36
  - /dev/samst entry, 35
  - Additional parameters 필드, 38
  - device state 필드, 38
  - Equipment Identifier 필드, 35
  - Equipment Ordinal 필드, 36
  - Equipment Type 필드, 36
  - shared 키워드, 70
  - Sun Cluster 사용, 158
  - Sun Cluster 예제, 152
  - Sun Cluster에서의 공유 파일 시스템, 134
  - Sun Cluster에서의 공유 해제된 파일 시스템, 143
  - 공유 파일 시스템, 82
    - 구성, 34
    - 변경 사항 전과, 54
    - 업데이트, 82
    - 예제, 38, 42
    - 오류 검사, 65
    - 파일 시스템 크기 늘리기, 63
    - 패밀리 세트 필드, 37

- 필드, 34
- 확인, 226
- md 장치, 36
- mdadm, Linux 클라이언트에서 사용되지 않음, 243
- meta\_timeo
  - samu(1M) 명령, 311
  - 마운트 옵션, 250
- metadb(1M) 명령, 150
- metaset 명령, 153
- mh\_write
  - samu(1M) 명령, 312
  - 마운트 옵션, 248
- min\_pool 마운트 옵션, 250
- minallocsz
  - samu(1M) 명령, 312
  - 마운트 옵션, 247
- mm 장치, 36
- mm\_stripe samu(1M) 명령, 313
- mount samu(1M) 명령, 317
- mount(1M) 명령, 40, 59, 60
  - 공유 파일 시스템, 68
  - 문제 해결, 93, 225, 232
  - 실패, 226
- mr 장치, 36
- ms 파일 시스템, 9, 36

## N

- nalloc 명령, 27
- netstat(1M) 명령, 232, 233
- Network Appliance 인터페이스 에뮬레이션, 186
- NFS
  - 공유 지점, 148
  - 자원 그룹, 155
- nhino 매개변수, 219, 220
- ninodes 매개변수, 219, 221
- noabr samu(1M) 명령, 312
- nodev 키워드, 35
- nodio\_szero samu(1M) 명령, 311
- nodmr samu(1M) 명령, 313
- noforce\_nfs\_async samu(1M) 명령, 307

noforcedirectio samu(1M) 명령, 311  
nomh\_write samu(1M) 명령, 312  
noquota 마운트 옵션, 117  
noqwrite samu(1M) 명령, 314  
norefresh\_at\_eof samu(1M) 명령, 314  
nosetuid samu(1M) 명령, 314  
nosw\_raid samu(1M) 명령, 308  
notrace samu(1M) 명령, 316  
nsswitch.conf 파일, 146  
nstreams 마운트 옵션, 250

## O

OCR 파일, 140  
off samu(1M) 명령, 306  
on samu(1M) 명령, 306  
open samu(1M) 명령, 318  
Oracle Cluster Registry(OCR) 파일, 140  
Oracle RAC, 135  
    Sun Cluster 사용, 139

## Q

qfsrestore(1M) 명령, 66  
quota 마운트 옵션, 96  
quota.group 파일, 106  
Qwrite, 217  
qwrite  
    samu(1M) 명령, 314  
    마운트 옵션, 218

## R

rdlease  
    samu(1M) 명령, 312  
    마운트 옵션, 247  
read samu(1M) 명령, 318  
readahead  
    samu(1M) 명령, 308  
    마운트 옵션, 216

Red Hat Linux, 지원되는 배포, 5  
refresh samu(1M) 명령, 318  
refresh\_at\_eof samu(1M) 명령, 314  
release(1) 명령, 49  
retry 마운트 옵션, 246

## S

sam\_archive(3) API 루틴, 50  
sam\_release(3) API 루틴, 50  
sam\_segment(3) API 루틴, 50  
sam\_setfa(3) API 루틴, 50, 213  
sam\_ssum(3) API 루틴, 50  
sam\_stage(3) API 루틴, 50  
Samba, 207  
sam-catservd 데몬, 180  
samchaid(1M) 명령, 103  
samcmd(1M) aridle 명령, 56  
samcmd(1M) 명령, 59  
samd(1M) config 명령, 71  
samd(1M) 명령, 56, 64, 65  
samfs.cmd 파일, 40, 59, 214  
    Sun Cluster 사용, 135  
    예제, 61  
samfsck(1M) 명령, 37, 70, 74, 224, 225  
samfsconfig(1M) 명령, 230  
    예제 출력, 83  
sam-fsd 데몬, 78, 180  
sam-fsd(1M) 명령, 64, 65, 182  
samfsinfo(1M) 명령, 41  
    예제 출력, 229  
samfsrestore(1M) 명령, 66  
sam-genericd 데몬, 180  
samgrowfs(1M) 명령, 63, 64, 212  
sammkfs(1M) 명령, 26, 37, 41, 65  
    문제 해결, 225  
SAM-QFS, xxv  
samquota(1M) 명령, 98  
    인수, 111  
SAM-Remote, samu(1M) 디스플레이, 287

sam-rpcd 데몬, 180  
 sam-sharedfsd 데몬, 180  
 samsharefs(1M) 명령, 57, 71, 76  
   예제 출력, 230  
 sam-sharefsd 데몬, 78, 180  
   문제 해결, 93  
 sam-sharefsd 추적 로그, 238  
 samu(1M) 유틸리티  
   mount 명령, 59  
   nalloc 및 alloc 명령, 27  
   개요, 253  
   공유 파일 시스템 명령, 311  
   도움말 디스플레이, 267  
   디스플레이 키, 255  
   상태 코드, 301  
   운영자 디스플레이, 256  
   운영자 명령, 306  
   인터페이스, 255  
   장치 명령, 306  
   직접 I/O 명령, 309  
   파일 시스템 명령, 307, 312  
   호출, 253  
 samu(1M)의 장치 명령, 306  
 samunhold(1M) 명령, 211  
 SANergy, 206  
   설치, 208  
   파일 보류, 211  
 SANERGY\_SMBPATH 환경 변수, 207  
 SAN-QFS 파일 시스템  
   clients, 207  
   Sun StorageTek QFS 공유 파일 시스템과 비교, 212  
   개요, 204  
   마운트 해제, 209  
   메타 데이터 제어기, 206  
   문제 해결, 211  
   제한 사항, 206  
   확장, 212  
   활성화, 206  
 sconfig 명령, 158, 162  
 scdidadm(1M) 명령, 124, 157  
 scrgadm(1M) 명령, 129, 144, 148, 154  
 SCSI 디스크, 전송, 215  
 SCSI 상태, samu(1M) 디스플레이, 292  
 scswitch(1M) 명령, 129, 138, 155  
 sd.conf 파일, 215  
 sd\_max\_xfer\_size 정의, 215  
 segment(1) 명령, 50, 201  
 setfa 파일 속성, 50  
 setfa(1) 명령, 2, 26, 50, 184, 213  
 setuid samu(1M) 명령, 314  
 sfind(1) 명령  
   WORM 옵션, 200  
 share(1M) 명령, 207  
 shared 마운트 옵션, 246  
 sls(1) 명령, 49, 51  
   WORM 옵션, 199  
   출력 설명, 51  
 snap samu(1M) 명령, 318  
 Solstice DiskSuite/Solaris Volume Manager, 149  
 squash(1) 명령, 98  
 ssd.conf 파일, 215  
 ssd\_max\_xfer\_size 정의, 215  
 ssum(1) 명령, 49  
 stage(1) 명령, 50  
 stage\_flush\_behind  
   마운트 옵션, 219  
 Sun Cluster  
   DID 지원, 122  
   HA-NFS 구성, 148, 154, 160  
   HA-NFS 비활성화, 174, 175, 176  
   HA-SAM 구성, 178  
   HAStoragePlus 자원 유형, 140, 148  
   IPMP 유효성 검사, 145, 154, 160  
   NFS 자원 그룹, 148  
   Oracle RAC, 135, 139  
   Resource Group Manager, 141  
   Solstice DiskSuite/Solaris Volume Manager  
     사용, 149  
   Sun Cluster용 Solaris Volume Manager 사용, 123  
   VERITAS Clustered Volume Manager 사용, 156  
   개요, 120  
   개인용 상호 연결 이름, 136  
   공유 또는 공유 해제된 파일 시스템 사용, 122  
   공유 파일 시스템, 131

- 공유 파일 시스템 예제, 132
- 공유 해제된 파일 시스템 구성, 140
- 공유 해제된 파일 시스템 예제, 141
- 구성 검증, 137
- 구성 변경, 172
- 구성 예제, 129
- 네트워크 이름 서비스, 145, 154, 160
- 메타 데이터 서버 자원, 131
- 설명서, 120
- 제한 사항, 121
- 클러스터 외부에 클라이언트 구성, 161
- 클러스터 외부의 클라이언트, 161
- Sun Cluster-용 Solaris Volume Manager, 123
- SUNW.qfs 자원 유형, 129, 138
- SuSE Linux, 지원되는 배포, 5
- sw\_raid samu(1M) 명령, 308
- sync\_meta
  - samu(1M) 명령, 315
  - 마운트 옵션, 251
- syslog.conf 파일, 224

## T

- touch 유틸리티, WORM-FS 보유 기간 설정, 197
- trace samu(1M) 명령, 316
- trace\_rotate(1M) 명령, 182

## U

- unavail samu(1M) 명령, 306
- unfuse 명령, 209
- unload samu(1M) 명령, 307
- unmount(1M) 명령, 68
  - 공유 파일 시스템, 80
- unshare(1M) 명령
  - SAN-QFS 파일 시스템, 210

## V

- VERITAS Clustered Volume Manager, 156
- VFS, vnode 인터페이스

- vfstab 파일, 40, 59
  - shared 키워드, 70
  - Sun Cluster 사용, 144, 153
  - 공유 파일 시스템, 79
  - 보유 기간, WORM-FS, 196
  - 필드, 60
- Vnode 인터페이스
  - VFS 참조
- vrtdiskadd 명령, 157
- vrtdmpadm 명령, 157
- vxvg 명령, 158
- VxVM, 156
  - 구성, 157
  - 장치, 177
  - 패키지, 156

## W

- WORM 보유, 53
- worm\_capable 마운트 옵션, 187, 252
- worm\_emul 마운트 옵션, 187
- worm\_lite 마운트 옵션, 187
- WORM-FS
  - "lite" 옵션, 188
  - touch를 사용하여 보유 기간 설정, 197
  - WORM 파일 생성, 189, 194
  - WORM 파일 속성 보기, 199
  - 개요, 186
  - 검색, 200
  - 기본 보유 기간 설정, 196
  - 마운트 옵션, 188, 252
  - 에플리케이션 모드, 186
  - 옵션, 194
- wr\_throttle
  - samu(1M) 명령, 309
  - 마운트 옵션, 218
- writebehind
  - samu(1M) 명령, 309
  - 마운트 옵션, 215
- wrlease
  - samu(1M) 명령, 312
  - 마운트 옵션, 247

## X

xntpd(1M) 데몬, 238

### ㄱ

강제 마운트 해제, 62

고가용성 SAM 구성, 178

공유 메모리, samu(1M) 디스플레이, 270, 273, 276

공유 파일 시스템, 4

  aplease 마운트 옵션, 247

  hosts 파일, 71, 76

  I/O, 93

  Linux 클라이언트, 5

  maxallocsz 마운트 옵션, 247

  mcf 파일, 82

  meta\_timeo 마운트 옵션, 250

  mh\_write 마운트 옵션, 248

  min\_pool 마운트 옵션, 250

  minallocsz 마운트 옵션, 247

  nodev 키워드, 35

  sam-fsd 데몬, 78

  sam-sharesfsd 데몬, 78

  SAN-QFS 공유 파일 시스템과 비교, 212

  stripe 마운트 옵션, 251

  Sun Cluster 사용, 131, 136

  Sun Cluster 사용 예제, 132

  Sun Cluster 외부의 클라이언트, 161

  sync\_meta 마운트 옵션, 251

  wrelease 마운트 옵션, 247

  공유 해제로 변환, 73

  공유 해제에서 공유로 변환, 69

  구성, 67

  데몬, 180

  로컬 호스트 파일, 72, 78

  마운트, 68

  마운트 옵션, 245

  마운트 지점, 4

  마운트 해제, 80

  메타 데이터 서버 변경, 88, 91

  명령, samu(1M) 사용, 311

  입대, 247

  장애 조치, 88, 89, 91

  캐시된 속성, 250

  클라이언트 호스트 제거, 80

  클라이언트 호스트 추가, 76

  클라이언트-서버 통신, 93

  파일 잠금, 5

  공유 파일 시스템의 스레드, 250

  공유 호스트 파일

    변경, 57, 58

  관리자 세트, 할당량, 95, 103

  광 디스크 레이블, samu(1M) 디스플레이, 267

  광 디스크 상태

    samu(1M) 디스플레이, 281

  구성

    mcf 파일 만들기, 34

    samu(1M) 디스플레이, 260

    디스크 사용, 8

    변경사항, 전파, 53

    파일 할당, 13

  구성 변경사항 전파, 53

  기술 지원, xxx

### ㄴ

  네트워크 이름 서비스, 145

### ㄷ

  다중 관독기 파일 시스템, 202

  단일 할당 체계, 10

  대량 저장, samu(1M) 디스플레이, 274

  데몬

    sam-catserverd, 180

    sam-fsd, 78, 180

    sam-genericd, 180

    sam-rpcd, 180

    sam-sharedfsd, 180

    sam-sharesfsd, 78, 180

    samu(1M) 디스플레이, 262

    xntpd(1M), 238

  개요, 180

  공유 파일 시스템, 180

  추적, 181

  추적, samu(1M) 명령, 317

데이터 맞춤, 11  
도움말  
  samu(1M) 디스플레이, 267  
디렉토리 속성, 184  
디렉토리 이름 검색 캐시(DNLC), 7, 93  
디스크  
  추가 또는 업그레이드, 64  
디스크 볼륨 사전  
  samu(1M) 디스플레이, 264  
디스크 캐시  
  초과, 201  
  추가, 63  
디스크 할당 단위, DAU 참조

ㄹ  
라운드 로빈 할당, 14, 36  
  사용자 지정, 185  
  예제, 42  
라이브러리 카탈로그, samu(1M) 디스플레이, 296  
라이센싱  
  일반 정보, xxx  
로그 파일  
  Linux 클라이언트, 240  
  sam-sharefsd 추적 로그, 238, 239  
  파일 시스템 문제 해결, 224  
로컬 호스트 파일, 72, 78  
  작성, 85

ㄱ  
마운트 매개변수, 29  
마운트 옵션  
  abr, 123  
  bg, 246  
  dmr, 123  
  flush\_behind, 219  
  Linux 클라이언트, 243  
  meta\_timeo, 250  
  mh\_write, 248  
  min\_pool, 250  
  minallocsz 및 maxallocsz, 247

  rdlease, wrlease 및 aplease, 247  
  retry, 246  
  shared, 246  
  stage\_flush\_behind, 219  
  sync\_meta, 251  
  worm\_capable 및 def\_retention, 252  
  WORM-FS, 188  
  wr\_throttle, 218  
  공유 파일 시스템, 245  
  스트라이프, 251  
  할당량에 대한, 96  
마운트 지점, 확인, 227  
마운트 해제  
  SAN-QFS 파일 시스템, 209  
  파일 시스템, 62  
매체 언로드, samu(1M) 사용, 307  
메모리, samu(1M) 디스플레이, 261  
메시지 파일, 63, 224  
메타 데이터, 7  
  mcf의 장치, 36  
  개요, 3  
  내용, 7  
  분리, 7  
  서버, 4  
  할당, 14  
메타 데이터 서버, 공유 파일 시스템 참조  
무한 할당량, 104

ㅂ  
버퍼된 I/O  
  I/O, 페이지된  
보류 스테이지 대기열, samu(1M) 디스플레이, 299  
보유, 53  
  touch를 사용하여 기간 설정, 197  
  기간, WORM-FS, 194  
  기본 기간, WORM-FS, 196  
  마운트 옵션, 252  
볼륨 관리, 2  
분할 영역(추가, 변경, 삭제), 64  
블록 할당량, SAN-QFS 파일 시스템, 211

人

상태 코드

- samu(1M)를 사용하여 보기, 301, 302
- 서비스, samu(1M) 디스플레이, 285
- 설명서, xxviii
- 섹터 데이터, samu(1M) 디스플레이, 289
- 소프트 제한, 할당량, 97

소프트웨어

- 라이센싱, xxx
- 설명서, xxviii
- 운영 체제 요구 사항, xxv
- 수퍼 블록, 버전 1 및 2, 10, 41

스테이지

- 보류 스테이지 대기열 samu(1M) 디스플레이, 299
- 상태, samu(1M) 디스플레이, 278, 292

스트라이프

- samu(1M) 명령, 315
- 마운트 옵션, 217, 251
- 소개, 17

스트라이프 그룹, 6, 19, 37

- 예제, 45
- 일치하지 않음, 23
- 장치, 186
- 파일 속성, 50

스트라이프 너비

- 데이터 디스크, 11
- 메타 데이터 디스크, 13
- 스트라이프 할당, 17, 36, 251
- 사용자 지정, 185
- 스트라이프 너비, 11, 185, 217
- 예제, 44

시스템 파일, 214

쓰기 스토틀, 218

○

아카이버

- samu(1M) 디스플레이, 257

아카이브

- 복사본, 52

업그레이드

- 디스크, 64

에플리케이션 모드, WORM, 186

운영 체제, xxv

운영자 유틸리티, samu(1M) 참조

윤년, WORM 보유 기간, 196

이중 할당 체계, 9

임대, 247

ㄹ

자동화된 라이브러리

- samu(1M) 디스플레이, 282, 296

작성기 마운트 옵션, 202

장비

- Identifier 필드, 35

- Ordinal 필드, 36

- Type 필드, 36

장치 상태 코드, samu(1M)로 보기, 303

장치 상태, samu(1M) 디스플레이, 287

장치 테이블, samu(1M) 디스플레이, 294

제거 가능 매체

- samu(1M) 디스플레이, 282

- 상태 코드, samu(1M), 301

- 상태, samu(1M) 디스플레이, 286

ㄷ

추적 로그 파일, 238

추적 파일, 181

- 교환, 182

- 디렉토리, 181

- 이벤트, 181

ㅋ

캐시된

- I/O, I/O, 페이지된 참조

커널 통계, samu(1M) 디스플레이, 271

클라이언트-서버 통신, 93

클러스터 외부의 클라이언트, 161

## ㅅ

테이프 드라이브

samu(1M) 디스플레이, 290

## ㅇ

파일

대용량 파일 작업, 201

대형 파일 전송, 214

파일 공간의 사전 할당, 6, 184

파일 백업, 65

파일 보유, 53

파일 속성, 7, 49, 184

파일 시스템

ma, 36

ms, 36

ms 및 ma, 9

samu(1M) 디스플레이, 265

공유, 공유 파일 시스템 참조

공유로 변환, 69

기능, 2

기본 작업, 49

디스크 캐시 추가, 63

디자인, 7

마운트, 59

마운트 해제, 62

매개변수, samu(1M) 디스플레이, 279

메타 데이터, 3

명령, samu(1M) 사용, 307, 312

백업, 65

복구, 3, 225

볼륨 관리, 2

사용량, samu(1M) 디스플레이, 272

상태 코드, samu(1M)로 보기, 302

용량, 2

이름 바꾸기, 63

재작성, 64

지원되는 파일 수, 3

직접 I/O 명령, samu(1M) 사용, 309

캐시 초과, 201

파일 정보, 51

파일 할당, 13

할당량, 할당량 참조

파일 시스템 복구, 225

파일 시스템 이름 바꾸기, 63

파일 시스템의 복구, 3

파일 할당

라운드 로빈, 13, 14

방법, 185

사전 할당 공간, 184

스트라이프, 13, 17

스트라이프 그룹, 19

일치하지 않는 스트라이프 그룹, 23

관독기 마운트 옵션, 202

패밀리 세트 필드, 37

페이지된 I/O, I/O 참조

프로세스, 데몬 참조

## ㅎ

하드 제한, 할당량, 97

할당

nalloc 및 alloc 명령, 27

할당 크기 조정, 247

할당량

0, 104

default, 104

SAN-QFS에서의 블록 할당량, 211

개요, 95

관리자 세트, 95, 103

구성, 99, 101

그룹 할당량 보기, 115

디스크 블록 및 파일 제한, 97

명령, 98

무한, 104

변경, 110

소프트 제한, 97

수정, 118

유예 기간, 110, 113

자원 할당 방지, 115

제거, 110, 117

제한 변경, 106

제한 활성화, 106

하드 제한, 97

할당량 파일, 96

할당량이 있는 DAU, 97

확인, 108  
활성화, 98  
할당량 제거, 117  
할당량에 대한 유예 기간, 110, 113  
해시 테이블, 221

