



Sun StorEdge™ SAM-FS 저장 및 아카이브 관리 안내서

릴리스 4.1

Sun Microsystems, Inc.
www.sun.com

부품 번호: 817-7389-10
2004년 6월, 개정판 A

이 문서에 대한 의견은 다음 주소로 보내 주십시오. <http://www.sun.com/hwdocs/feedback>

Copyright 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc.는 이 설명서에 설명된 기술과 관련된 지적 재산권을 보유하고 있습니다. 특히 이러한 지적 재산권에는 <http://www.sun.com/patents>에 나열된 하나 이상의 미국 특허와 미국 및 기타 국가에서의 하나 이상의 추가 특허 또는 출원 중인 제품이 포함될 수 있습니다.

본 설명서 및 관련 제품은 사용, 복사, 배포 및 역컴파일을 제한하는 라이선스 하에서 배포됩니다. 본 제품 또는 설명서의 어떠한 부분도 Sun 및 Sun 소속 라이선스 부여자(있는 경우)의 사전 서면 승인 없이는 어떠한 형태나 수단으로도 재생산할 수 없습니다.

글꼴 기술과 같은 협력업체 소프트웨어는 Sun 제공업체로부터 저작권을 얻거나 사용 허가받은 것입니다.

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

Sun, Sun Microsystems, Sun 로고, AnswerBook2, docs.sun.com, Solaris 및 StorEdge는 미국 및 기타 국가에서 Sun Microsystems, Inc.의 상표 또는 등록 상표입니다.

모든 SPARC 상표는 라이선스 하에서 사용되며 미국 및 기타 국가에서 SPARC International, Inc.의 상표 또는 등록 상표입니다. SPARC 상표가 있는 제품은 Sun Microsystems, Inc.가 개발한 구조에 기초합니다.

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

OPEN LOOK 및 Sun™ Graphical User Interface는 해당 사용자 및 라이선스 피부여자를 위해 Sun Microsystems, Inc.가 개발했습니다. Sun은 컴퓨터 산업에서 시각적 또는 그래픽 사용자 인터페이스의 개념을 연구하고 개발하는데 있어 Xerox의 선구자적 업적을 인정합니다. Sun은 Xerox Graphical User Interface에 대한 Xerox의 비독점적 라이선스를 보유하고 있으며 이 라이선스는 OPEN LOOK GUI를 구현하거나 그 외의 경우 Sun의 서면 라이선스 계약을 준수하는 Sun의 라이선스 피부여자를 포괄합니다.

미국 정부 권리 - 상업용 목적으로 사용. 정부 기관 사용자는 Sun Microsystems, Inc.의 표준 계약 동의서의 적용을 받으며 FAR 및 추가 조항의 적용을 받습니다.

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



재활용
가능



Adobe PostScript

목차

머리말	xiii
본 설명서의 구성	xiv
UNIX 명령 사용	xiv
셸 프롬프트	xv
표기 규칙	xv
관련 설명서	xvi
온라인 Sun 설명서 액세스	xvi
제 3자 웹 사이트	xvii
Sun 기술 지원 연락	xvii
라이센스	xviii
진단	xviii
설치 지원	xviii
고객 의견	xviii

1. 개요 1

기능	1
아카이브	2
릴리스	2
스테이지	3
리사이클	3

저장 장치	3
명령	4
사용자 명령	5
일반 시스템 관리자 명령	6
파일 시스템 명령	7
자동화 라이브러리 명령	8
아카이버 명령	9
특수 유지 보수 명령	9
사용자 정의 가능 사이트 스크립트	10
애플리케이션 프로그래머 인터페이스	11
작동 유틸리티	11

2. Sun StorEdge SAM-FS 환경에서 자동화 라이브러리 및 수동으로 로드된 드라이브 사용하기 13

규칙	14
명령 인수	14
용어	15
자동화 라이브러리 작업	15
▼ 제거 가능한 매체 작업을 중지하려면	16
▼ 제거 가능한 매체 작업을 시작하려면	17
▼ 자동화 라이브러리를 실행하려면	17
▼ 자동화 라이브러리를 해제하려면	18
▼ 자동화 라이브러리에 카트리지를 로드하려면	18
▼ 드라이브에서 카트리지를 언로드하려면	19
카트리지에 레이블 지정	19
▼ 테이프에 레이블 지정 또는 변경하려면	20
▼ 광 디스크에 레이블을 지정 또는 변경하려면	21
▼ 볼륨을 감사하려면	21
▼ 자동화 라이브러리를 감사하려면 (직접 연결 전용)	22

클리닝 카트리지를 사용 23

- ▼ 클리닝 주기 회수를 재설정하려면 23
- ▼ 바코드가 있는 클리닝 카트리지를 사용하기 24
- ▼ 바코드가 없는 클리닝 카트리지를 사용하기 24
- ▼ 테이프 드라이브를 클리닝하려면 25

▼ 매체 오류를 지우려면 26

▼ 드라이브에서 고정된 카트리지를 제거하려면 27

카탈로그 작업, 카트리지를 가져오기 및 카트리지를 내보내기 29

내보낸 매체 추적 — 기록자 30

자동화 라이브러리로부터의 가져오기 또는 내보내기 30

▼ 편지함을 사용하여 라이브러리로부터 카트리지를 가져오기 31

▼ 편지함을 사용하여 라이브러리로부터 카트리지를 내보내기 31

▼ 편지함을 사용하지 않고 라이브러리로부터 카트리지를 가져오기 32

▼ 편지함을 사용하지 않고 라이브러리로부터 카트리지를 내보내기 33

▼ 로드 통지 사용하기 33

수동으로 로드된 드라이브 작업 34

▼ 카트리지를 로드하려면 34

▼ 카트리지를 언로드하려면 34

▼ 라이브러리 카탈로그를 보려면 35

3. 작동 절차가 업체별로 고유한 라이브러리의 기본 작업 37

ADIC/Grau 자동화 라이브러리 37

▼ 카트리지를 가져오려면 38

▼ 카트리지를 내보내려면 38

Fujitsu LMF 자동화 라이브러리 39

▼ 카트리지를 가져오려면 39

▼ 카트리지를 내보내려면 40

IBM 3584 UltraScalable 테이프 라이브러리 41

카트리지를 가져오기 41

드라이브 클리닝 41

파티션 41

▼ 카트리지를 언로드 하려면 41

IBM 3494 라이브러리 42

▼ 카트리지를 가져오려면 42

▼ 카트리지를 내보내려면 42

Sony 직접 연결 8400 PetaSite 자동화 라이브러리 43

▼ 테이프를 가져오려면 43

테이프 내보내기 44

▼ 편지함 슬롯을 저장소 슬롯으로 사용하지 않고 테이프를 내보내려면 44

▼ 편지함 슬롯을 저장소 슬롯으로 사용해 테이프를 내보내려면 45

▼ 카트리지를 다른 슬롯으로 이동하려면 45

Sony 네트워크 연결 자동화 라이브러리 46

▼ 카트리지를 가져오려면 46

▼ 카트리지를 내보내려면 47

StorageTek ACSLS 연결 자동화 라이브러리 47

▼ 테이프를 가져오려면 48

▼ 편지함을 사용해 테이프를 내보내려면 48

4. 아카이브 51

아카이버 - 작동 원리 51

아카이브 세트 52

아카이브 작업 53

1 단계: 아카이브로 파일 식별하기 54

2 단계: 아카이브 요청 구성 56

3 단계: 아카이브 요청 예약 57

4 단계: 아카이브 요청의 파일 아카이브 59

예제 기본 수행 결과 60

아카이버 데몬	60
아카이브 로그 파일 및 이벤트 로깅	61
archiver.cmd 파일	63
▼ archiver.cmd 파일 작성 또는 수정하기 및 변경내용 전파하기	63
archiver.cmd 파일	64
archiver.cmd 파일 예제	65
archiver.cmd 명령	67
전역 아카이브 명령	67
archivemeta 명령: 메타데이터의 아카이브 여부 제어	67
archmax 명령: 아카이브 파일의 크기 제어	68
bufsize 명령: 아카이버 버퍼 크기 설정	69
drives 명령: 아카이브에 사용된 드라이브 수 제어	70
examine 명령: 아카이브 스캔 제어	70
interval 명령: 아카이브 간격 지정	71
logfile 명령: 아카이버 로그 파일 지정	72
▼ 아카이버 로그 파일 백업하기	72
notify 명령: 이벤트 통지 스크립트 이름 바꾸기	73
ovflmin 명령: 볼륨 오버플로 제어	73
wait 명령: 아카이버 시작 연기	75
기타 파일 시스템 명령	75
fs 명령: 파일 시스템 지정	76
기타 파일 시스템 명령	76
아카이브 세트 할당 명령	76
아카이브 세트 할당	77
파일 크기 <i>search_criteria</i> : -access	78
파일 크기 <i>search_criteria</i> : -minsize 및 -maxsize	79
소유자 및 그룹 <i>search_criteria</i> : -user 및 -group	79
패턴 일치를 사용한 파일 이름 <i>search_criteria</i> : -name <i>regex</i>	80

릴리스 및 스테이지 파일속성: -release 및 -stage	82
아카이브 세트 구성원 충돌	83
아카이브 복사 명령	84
아카이브 이후 디스크 공간 릴리스 -release	84
디스크 공간 릴리스 지연: -norelease	85
아카이브 나이 설정	85
자동 아카이브 해제	86
메타 데이터에 대해 2개 이상의 복사본 지정	86
아카이브 설정 복사 매개변수	87
아카이브 파일의 크기 제어: -archmax	88
아카이버 버퍼 크기 설정: -bufsize	88
아카이브 요청에 대한 드라이브 수 지정: -drivemax, -drivemin 및 -drives	88
볼륨 공간 최소화하기: -fillvsns	90
아카이브 버퍼 잠금 지정 -lock	91
오프라인 파일의 아카이브 복사본 만들기: -offline_copy	92
리사이클 지정	92
연관 아카이브 -join	93
아카이브 해제 제어	94
아카이브 파일의 작성 방식 제어 -tapenonstop	95
볼륨 예약 -reserve	95
아카이브 우선 순위 설정 -priority	99
아카이브 예약: -startage, -startcount 및 -startsize	100
VSN 연결 명령	101
VSN 풀 명령	103
디스크 아카이브	105
구성 지침	106
디스크 아카이브 명령	106
▼ 아카이브 디스크 활성화 방법	107

디스크 아카이브 예제	109
예제 1	109
예제 2	110
예제 3	111
아카이버 예제	112
예제 1	112
예제 2	114
예제 3	116
예제 4	120
아카이버 지침	124
아카이버 문제 해결	125
파일이 아카이브되지 않는 원인	126
추가 아카이버 진단	127
파일이 릴리스되지 않는 원인	127

5. 릴리스 129

릴리서 개요	130
작동 원리	130
용어 정의	131
나이	131
대상	131
우선 순위	132
가중치	132
부분 릴리스	132
부분 릴리스 및 부분 스테이지	132
시스템 관리자 옵션 요약	134
사용자 옵션 요약	134
releaser.cmd 파일	135

나이 및 크기 관련 릴리스 우선 순위 명령 지정 `weight_age`,
`weight_age_access`, `weight_age_modification` 및
`weight_age_residence` 136

파일 나이 136

파일 크기 137

개별 파일 시스템의 명령 지정 `fs` 138

디버그 명령 지정 `no_release` 및 `display_all_candidates` 139

최소 상주 시간 지정 `min_residence_age` 139

로그 파일 지정 `logfile` 139

재아카이브된 파일의 릴리스 방지 `rearch_no_release` 141

릴리서 대상 목록의 크기 조정: `list_size` 142

릴리스에서 `archiver.cmd` 파일의 역할 142

릴리서 구성 143

릴리서 수동 실행 144

릴리서 문제 해결 144

6. 스테이지 147

`stager.cmd` 파일 147

- ▼ `stager.cmd` 파일 작성 또는 수정하기 및 변경내용 전파하기 149
 - 드라이브 수 지정 149
 - 스테이지 버퍼 크기 지정 150
 - 로그 파일 지정 151
 - 스테이지 요청 횟수 지정 153
 - 예제 `stager.cmd` 파일 153

스테이지에서 `archiver.cmd` 파일의 역할 154

`preview.cmd` 파일을 사용한 미리보기 요청 우선 순위 지정 154

- VSN 및 나이 명령(전역) 155
- 위터마크 명령(전역 또는 파일 시스템 전용) 156

총 미리보기 요청 우선 순위 계산 158

미리보기 요청 우선 순위 방식 설정 방법 158

- 예제 1: 스테이지 요청 실행 159
- 예제 2. 아카이브 요청 실행 159
- 예제 3. 매체별 우선 순위 지정 160
- 예제 4. 복합 우선 순위 지정 160

7. 리사이클 163

리사이클러:개요 163

리사이클 명령 165

로그 파일 지정 logfile 명령 165

리사이클 방지 no_recycle 명령 166

전체 자동화 라이브러리에 리사이클 지정 라이브러리 명령 166

리사이클러 구성 168

▼ 1 단계: recycler.cmd 파일 작성(선택적) 169

예제 recycler.cmd 파일 169

▼ 2 단계: archiver.cmd 파일 편집 (선택적) 171

▼ 3 단계: 리사이클러 실행 173

▼ 4 단계: 리사이클러용 crontab 파일 만들기(선택 사항) 174

▼ 5 단계: -recycle_ignore 및 ignore 매개변수 제거 175

▼ 6 단계: recycler.sh 파일 작성(선택적) 175

리사이클러 문제 해결 175

8. 사용중인 환경에서 하드웨어 업그레이드하기 177

자동화 라이브러리에 슬롯 추가 178

▼ 라이브러리에 슬롯을 추가하려면 178

라이브러리 업그레이드 또는 교체하기 179

▼ 라이브러리를 교체하거나 업그레이드하려면 179

DLT 테이프 드라이브 업그레이드하기 182

▼ 테이프 드라이브를 업그레이드하려면 182

9. 고급 항목 185

장치 로깅 185

장치 로그의 사용 시점 186

장치 로그 활성화 186

▼ samset(1M) 명령을 사용하여 장치 로그 활성화하기 187

▼ defaults.conf 파일을 편집하여 장치 로그 활성화하기 187

제거 가능한 매체 파일 188

▼ 제거 가능한 매체 또는 볼륨 오버플로 파일 작성하기 189

세그먼트된 파일 190

아카이브 190

재해 복구 191

시스템 오류 기능 보고 191

▼ SEF 보고를 활성화하려면 192

SEF 보고 결과 192

▼ SEF 출력 생성하기 193

SEF 로그 파일 관리 196

용어 해설 197

색인 217

머리말

본 설명서 *Sun StorEdge™ SAM-FS 저장 및 아카이브 관리 안내서*는 Sun StorEdge SAM-FS 4.1 릴리스에서 지원된 저장 및 아카이브 관리 소프트웨어를 설명합니다. Sun StorEdge SAM-FS 소프트웨어에서 온라인 디스크의 파일은 자동으로 아카이브 매체로 복사합니다. 아카이브 매체는 온라인 디스크나 제거 가능한 매체 카트리지로 구성됩니다.

Sun StorEdge SAM-FS 4.1 릴리스는 다음 Sun Solaris™ 운영 체제(OS) 플랫폼에서 지원됩니다.

- Solaris 8, 업데이트 5, 7/01
- Solaris 9, 업데이트 3, 4/03

본 설명서는 Sun StorEdge SAM-FS 소프트웨어의 구성 및 관리 책임이 있는 시스템 관리자용입니다. 이 설명서를 읽는 사용자는 시스템 관리자가 이미 계정 작성, 시스템 백업 수행, 기타 기본적인 Solaris 시스템 관리 업무를 포함한 Solaris 업무 절차에 해박한 지식을 보유한 상태임을 가정합니다.

참고 – Sun StorEdge SAM-FS 소프트웨어에 있는 저장 및 아카이브 관리자를 사용하여 Sun StorEdge QFS 파일 시스템을 실행하는 Sun StorEdge QFS 및 Sun StorEdge SAM-FS 소프트웨어 모두에 대해 라이선스를 구입할 수 있습니다. 이런 시스템을 *Sun SAM-QFS*라고 합니다.

본 설명서는 반드시 필요한 경우가 아니라면 Sun SAM-QFS 구성을 언급하지 않습니다. 본 설명서에서 저장 및 아카이브 관리에 대해 설명할 때 Sun StorEdge SAM-FS에 대한 참조를 Sun SAM-QFS 구성에도 적용한다고 가정할 수 있습니다. 또한 시스템 디자인 및 기능에 대해 설명할 때 Sun StorEdge QFS에 대한 참조를 Sun SAM-QFS 구성에도 적용한다고 가정할 수 있습니다.

본 설명서의 구성

본 설명서는 다음 장으로 구성되어 있습니다.

- 1장에서는 개요 정보를 제공합니다.
- 2장은 기본 작동을 설명합니다. 이 장에 포함된 정보는 대부분의 자동화 라이브러리와 수동으로 로드한 장치에 적용됩니다.
- 3장은 해당 라이브러리 종류와 관련된 운영 지침을 가진 라이브러리의 카트리지를 관리 방법을 설명합니다. 이 장은 이러한 라이브러리와 해당 라이브러리를 기본 작동 절차를 설명합니다.
- 4장은 아카이브 프로세스를 설명합니다.
- 5장은 릴리스 프로세스를 설명합니다.
- 6장은 스테이지 프로세스를 설명합니다.
- 7장은 리사이클 프로세스를 설명합니다.
- 8장은 Sun StorEdge SAM-FS 환경에 특정한 업그레이드 절차를 설명합니다.
- 9장은 Sun StorEdge SAM-FS 옵션의 고급 항목을 설명합니다.

용어집은 본 설명서 및 기타 Sun StorEdge QFS 및 Sun StorEdge SAM-FS 설명서에 사용된 용어들을 설명합니다.

UNIX 명령 사용

본 설명서에는 시스템 종료, 시스템 부팅 및 장치 구성과 같은 기본적인 UNIX® 명령에 대한 정보가 없습니다. 이러한 정보는 다음을 참조하십시오.

- 시스템과 함께 제공되는 소프트웨어 문서.
- 다음 URL의 Solaris™ 운영 체제 설명서:

<http://docs.sun.com>

셸 프롬프트

표 P-1은 본 설명서에서 사용되는 셸 프롬프트를 나타낸 것입니다.

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	#

표기 규칙

표 P-2는 본 설명서에서 사용된 표기 규칙을 나열한 것입니다.

표 P-2 표기 규칙

서체 또는 기호	의미	예
<i>AaBbCc123</i>	명령, 파일, 디렉토리 이름 또는 컴퓨터 화면 출력.	<code>.login</code> 파일을 편집하십시오. <code>ls -a</code> 를 사용하여 모든 파일을 나열합니다. <code>% You have mail.</code>
<i>AaBbCc123</i>	컴퓨터 화면 출력에서 사용자가 직접 입력하는 내용.	<code>% su</code> <code>Password:</code>
<i>AaBbCc123</i>	설명서 제목; 새로운 단어 또는 용어; 강조할 단어; 실제 이름 또는 값으로 대체될 명령행 변수.	<i>사용 설명서</i> 에서 6장을 참조하십시오. 이것을 <i>클래스</i> 옵션이라고 합니다. 이 작업을 수행하려면 루트 권한이 <i>반드시</i> 있어야 합니다. 파일을 삭제하려면 <code>rm filename</code> 을 입력하십시오.

표 P-2 표기 규칙 (계속)

서체 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 { p_호스트 s_호스트 }
\	명령행 끝의 백슬래시(\)는 명령이 다음 행에서 계속됨을 나타냅니다.	atm90 /dev/md/rdisk/d5 \ /dev/md/rdisk/d1

관련 설명서

본 설명서는 Sun StorEdge QFS 및 Sun StorEdge SAM-FS 소프트웨어의 조작을 설명하는 문서 세트의 일부입니다. 표 P-3은 이 제품의 릴리스 4.1 전체 문서 세트를 표시합니다.

표 P-3 관련 설명서

제목	제품 번호
<i>Sun SAM-Remote Administration Guide</i>	816-2094-11
<i>Sun QFS, Sun SAM-FS 및 Sun SAM-QFS 재난 복구 안내서</i>	816-7679-10
<i>Sun StorEdge QFS 및 Sun StorEdge SAM-FS 파일 시스템 관리 안내서</i>	817-7384-10
<i>Sun StorEdge QFS 및 Sun StorEdge SAM-FS 소프트웨어 설치 및 구성 안내서</i>	817-7394-10
<i>Sun StorEdge SAM-FS 저장 및 아카이브 관리 안내서</i>	817-7389-10
<i>Sun StorEdge QFS 및 Sun StorEdge SAM-FS 릴리스 노트</i>	817-7399-10

온라인 Sun 설명서 액세스

Sun StorEdge QFS 및 Sun StorEdge SAM-FS 소프트웨어 배포에는 이 제품에 대한 설명서 PDF 파일이 포함됩니다. 이러한 PDF 파일은 다음 위치에서 볼 수 있습니다.

- Sun 네트워크 저장소에 있는 설명서 웹 사이트.

이 웹 사이트에는 여러 저장 소프트웨어 제품에 대한 설명서가 있습니다.

- a. 이 웹 사이트에 액세스하려면 다음 URL로 이동하십시오.

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

Storage Software 페이지가 표시됩니다.

- b. 다음 목록에서 해당 링크를 클릭하십시오.

- Sun StorEdge QFS 소프트웨어
- Sun StorEdge SAM-FS 소프트웨어
- docs.sun.com.

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

- a. 이 웹 사이트에 액세스하려면 다음 URL로 이동하십시오.

docs.sun.com

docs.sun.com 페이지가 표시됩니다.

- b. 검색 상자에서 다음 중 하나를 검색하여 해당 제품 설명서를 찾으십시오.

- Sun StorEdge QFS
- Sun StorEdge SAM-FS

제 3자 웹 사이트

Sun은 이 설명서에 언급된 제 3자 웹 사이트의 기능에 대해 책임지지 않습니다. Sun은 해당 사이트 또는 자원을 통해 사용 가능한 내용, 광고, 제품 또는 기타 자료에 대해 보증하거나 책임지지 않습니다. Sun은 해당 사이트 또는 자원을 통해 사용 가능한 어떤 내용, 상품 또는 서비스에 사용 또는 신뢰성과 관련하여 발생하는 실제 또는 가정의 손해 또는 손상에 대한 책임을 지지 않습니다.

Sun 기술 지원 연락

이 설명서에 없는 이 제품에 대한 기술적인 질문이 있는 경우 다음 웹 사이트를 방문하십시오.

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

라이센스

Sun StorEdge QFS 및 Sun StorEdge SAM-FS 소프트웨어에 대한 라이선스를 구할 수 있는 정보는 Sun 판매 담당자나 공인 서비스 제공자(ASP)에게 문의하십시오.

진단

Sun StorEdge QFS 및 Sun StorEdge SAM-FS 소프트웨어에는 samexplorer(1M) 스크립트가 포함되어 있습니다. 이 진단 스크립트는 시스템 관리자 및 Sun 고객 지원 담당자가 매우 유용하게 사용할 수 있습니다. 이 스크립트는 서버 구성에 대한 진단 보고서를 작성하고 로그 정보를 수집합니다. 소프트웨어를 설치한 후, 이 스크립트에 대한 자세한 내용을 보려면 samexplorer(1M) 매뉴얼 페이지를 참조하십시오.

설치 지원

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

고객 의견

Sun은 본 설명서의 개선을 위해 항상 노력하고 있으며, 고객의 의견 및 제안을 언제나 환영합니다. 다음 웹 사이트를 방문하여 의견을 제출할 수 있습니다.

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

피드백과 함께 설명서의 제목 및 부품 번호를 적어 주십시오(*Sun StorEdge SAM-FS 저장 및 아카이브 관리 안내서*, 부품 번호 817-7389-10).

개요

Sun StorEdge SAM-FS 환경은 구성 가능한 파일 시스템에 저장, 아카이브 관리 및 검색 기능을 제공합니다. Sun StorEdge SAM-FS 소프트웨어는 온라인 디스크 캐시에서 아카이브 매체로 파일을 복사하여 파일을 저장합니다. 아카이브 매체는 다른 파일 시스템의 디스크나 자동 또는 수동으로 로드된 스토리지 장치의 제거 가능한 테이프 또는 광자기 카트리지로 구성 가능합니다. 추가로 Sun StorEdge SAM-FS 소프트웨어는 사이트에 지정된 사용 임계값으로 온라인 디스크 공간을 자동으로 유지합니다. 이러한 제품으로 아카이브한 파일 데이터와 관련된 디스크 공간을 릴리스하고 필요시 해당 파일을 온라인 디스크에 복원합니다.

이 장은 Sun StorEdge SAM-FS 구성요소에 대한 기술적 개요를 제공합니다. 다음 주제를 다룹니다.

- 1 페이지의 "기능"
- 3 페이지의 "저장 장치"
- 5 페이지의 "명령"

기능

Sun StorEdge SAM-FS 환경에는 파일 시스템과 저장 및 아카이브 관리 소프트웨어가 포함됩니다. Sun SAM-QFS 환경에는 Sun StorEdge QFS 파일 시스템도 포함됩니다. 두 파일 시스템은 모두 서버 디스크 캐시에 상주하는 고성능 UNIX 파일 시스템입니다. 이러한 파일 시스템 간의 주요 차이점은 Sun SAM-QFS 파일 시스템에서는 분산 공유 파일 시스템을 포함하므로 더욱 고급 기능이 제공된다는 점입니다. 파일 시스템에 대한 자세한 내용을 보려면 *Sun StorEdge QFS* 및 *Sun StorEdge SAM-FS 파일 시스템 관리 안내서*를 참조하십시오.

Sun StorEdge SAM-FS 환경에 있는 기타 구성요소들은 다음과 같습니다.

- 아카이버 — 온라인 디스크 캐시 파일을 자동으로 아카이브 매체에 복사합니다. 아카이브 매체는 온라인 디스크 파일이나 제거 가능한 매체 카트리지로 구성할 수 있습니다.

- 릴리서 — 해당 아카이브 파일이 차지하는 디스크 블록을 비워서 자동으로 온라인 디스크 캐시를 사이트에 지정된 사용 임계값 수준으로 유지합니다.
- 스테이저 — 파일 데이터를 디스크 캐시에 복원합니다. 사용자나 프로세스가 디스크 캐시에서 해제된 파일 데이터를 요청할 경우 스테이저는 자동으로 파일 데이터를 온라인 디스크 캐시에 복사합니다.
- 리사이클러 — 만료된 아카이브 복사본이 있는 아카이브 볼륨을 지우고 재사용 가능한 볼륨을 만듭니다.

다음 절에서 이상의 각 기능을 간략하게 설명합니다. 이 기능에 대한 자세한 정보는 다음 장들에서 찾을 수 있습니다.

아카이브

기본적으로 아카이버는 Sun StorEdge SAM-FS 파일 시스템의 모든 파일에 대해 하나의 아카이브 복사본을 자동으로 작성하고 아카이브 매체에 아카이브 복사본을 기록합니다. 다양한 아카이브 매체에 최고 4개의 아카이브 복사본을 작성하도록 아카이브를 구성할 수 있습니다. 파일이 *세그먼트화된* 경우 각 세그먼트는 하나의 파일로 취급되어 각각 아카이브됩니다. 아카이브 프로세스는 디스크 기반 파일을 사이트 정의 가능한 선택 기준 집합과 일치시킨 다음 시작됩니다.

아카이브에 대한 자세한 내용을 보려면 53 페이지의 "아카이브"를 참조하십시오. 세그먼트화된 파일에 대한 자세한 내용을 보려면 198 페이지의 "세그먼트된 파일"을 참조하십시오.

릴리스

*릴리스*는 아카이브된 파일 데이터가 사용하는 기본 저장 장치(디스크)를 비우는 프로세스입니다. 전체 디스크 공간에 대한 백분율로 표현되는 2개의 임계값은 온라인 디스크 캐시의 빈 공간 관리에 사용됩니다. 이러한 임계값은 최고 워터마크와 최저 워터마크입니다. 온라인 디스크 소비량이 최고 워터마크를 초과한 경우 시스템에서는 자동으로 해당 아카이브 파일이 차지하는 디스크 공간을 릴리스하기 시작합니다. 아카이브된 파일 데이터가 차지하는 디스크 공간은 최저 워터마크에 다다를 때까지 릴리스됩니다. 파일은 파일 크기와 나이에 따라 릴리스에 선택됩니다. 선택적으로 파일의 첫 부분은 빠른 액세스와 스테이징 지연을 막기 위해 디스크에 계속 유지됩니다. 파일이 세그먼트로 아카이브된 경우 해당 파일의 부분은 개별적으로 해제 가능합니다. 릴리서에 대한 자세한 내용을 보려면 133 페이지의 "릴리스"를 참조하십시오.

스테이지

데이터 블록을 해제한 파일에 액세스할 경우 스테이지는 자동으로 파일이나 파일 세그먼트 데이터를 온라인 디스크 캐시로 자동 스테이지합니다. 읽기 작동은 스테이지 작동을 바로 추적하므로 전체 파일이 완전히 스테이지되기 전에 애플리케이션에 파일을 즉시 사용할 수 있습니다.

Sun StorEdge SAM-FS 소프트웨어는 스테이지 요청 오류를 자동으로 처리합니다. 스테이지 오류가 나타나면 시스템은 파일의 다음 사용 가능한 아카이브 복사본을 찾습니다. 자동으로 처리 가능한 스테이지 오류에는 매체 오류, 매체 사용 불가, 자동화 라이브러리 사용 불가 등이 있습니다. 스테이지에 대한 자세한 내용을 보려면 [이를 참조하십시오](#).

리사이클

파일을 수정하면 이 파일의 기존 버전과 관련된 아카이브 복사본은 아카이브 매체에서 *expired*된 것으로 간주됩니다. 이 복사본들은 더 이상 필요하지 않기 때문에 시스템에서 제거할 수 있습니다. 리사이클러는 만료된 아카이브 복사본의 가장 큰 부분을 아카이브 볼륨으로 식별하고 만료되지 않은 복사본을 개별 볼륨으로 이동하여 보존합니다.

제거 가능한 매체 볼륨에 만료된 복사본만 있는 경우 다음 작업 중 하나를 수행할 수 있습니다.

- 임시로 재사용하기 위해 볼륨 레이블을 변경할 수 있습니다.
- 파일 변경 내역의 기록으로 볼륨을 오프사이트 저장소로 내보낼 수 있습니다. 표준 UNIX 유틸리티를 사용하여 만료된 아카이브 복사본으로부터 이전 버전의 파일을 복원할 수 있습니다.

리사이클링 프로세스는 데이터 파일에 관련된 경우 최종 사용자에게 투명합니다. 리사이클에 대한 자세한 내용을 보려면 167 페이지의 "리사이클"을 참조하십시오.

저장 장치

Sun StorEdge SAM-FS 환경은 다양한 테이프 저장 및 광자기 장치를 지원합니다. Sun StorEdge SAM-FS가 지원하는 자동화 라이브러리를 환경에 연결된 방식에 따라 다음 그룹으로 나눌 수 있습니다.

- 직접 연결. 직접 연결 라이브러리는 SCSI(소형 컴퓨터 시스템 인터페이스)를 사용해 호스트 시스템에 직접 연결됩니다. 이는 직접 연결이거나 광섬유 채널 연결입니다. 예를 들어 Sun StorEdge 라이브러리는 직접 연결을 사용합니다. Sun StorEdge SAM-FS 시스템은 자동화 라이브러리용 SCSI 표준을 사용하여 직접 제어합니다.
- 네트워크 연결. Sun StorEdge SAM-FS 소프트웨어는 라이브러리의 호스트 시스템 클라이언트로 구성될 수 있습니다. 네트워크 연결 라이브러리에는 StorageTek, ADIC/Grau, IBM 및 Sony 라이브러리 등이 있습니다. 이러한 라이브러리에서는 해당 업체가 제공하는 소프트웨어 패키지를 사용합니다. 이런 경우, Sun StorEdge SAM-FS 소프트웨어는 자동화 라이브러리로 특별히 디자인한 데몬을 사용하여 해당 업체 소프트웨어와 인터페이스합니다.

표 1-1에서는 다양한 자동화 라이브러리별 데몬을 보여줍니다.

표 1-1 자동화 라이브러리 데몬

데몬	설명
sam-robotd	로봇 제어 데몬의 실행을 감시합니다. sam-initd 데몬이 sam-robotd 데몬을 자동으로 시작합니다.
sam-genericd	직접 연결 라이브러리와 매체 교환기를 제어합니다. 또한 DAS 인터페이스를 통해 ADIC 라이브러리를 제어합니다.
sam-stkd	ACSAPI 인터페이스를 통해 StorageTek 매체 교환기를 제어합니다.
sam-ibm3494d	lmcpd 인터페이스를 통해 IBM 3494 테이프 라이브러리를 제어합니다.
sam-sonyd	DZC-8000S 인터페이스를 통해 Sony 네트워크 연결 자동화 라이브러리를 제어합니다.

지원되는 저장 장치 목록을 알아보려면 Sun Microsystems 판매 대리점이나 인증된 서비스 제공업체(ASP)에 문의하십시오.

Sun StorEdge SAM-FS 환경 내에서 관리되는 장치 사이의 관계는 마스터 구성 파일 /etc/opt/SUNWSamfs/mcf에 정의됩니다. mcf 파일은 Sun StorEdge SAM-FS 환경에 포함된 제거 가능한 매체 장치, 라이브러리 및 파일 시스템을 지정합니다. mcf 파일에 각 장치에 대한 고유한 장치 식별자가 지정됩니다. 또한 mcf의 항목에는 마운트된 아카이브 장치와 자동화 라이브러리 카탈로그 파일이 수동으로 정의됩니다.

시스템은 가능하면 표준 Solaris 디스크와 테이프 장치 드라이버를 사용합니다. 특정 라이브러리 및 광 디스크 장치와 같이 Solaris 운영 체제(OS)에서 직접 지원되지 않는 장치에 대한 특수 장치 드라이버는 Sun StorEdge SAM-FS 소프트웨어 패키지에 포함되어 있습니다.

명령

Sun StorEdge QFS 및 Sun StorEdge SAM-FS 환경은 파일 시스템, 데몬, 프로세스, 다양한 명령 유형(사용자, 관리자, 등), 도구로 구성됩니다. 이 절은 Sun StorEdge QFS 및 Sun StorEdge SAM-FS 소프트웨어 배포에 포함되어 있는 명령을 설명합니다.

Sun StorEdge QFS 및 Sun StorEdge SAM-FS 명령은 표준 UNIX 파일 시스템 명령과 함께 사용할 수 있습니다. 일부 명령들은 한 제품에만 특정합니다. 모든 명령은 UNIX 매뉴얼 (1) 페이지에 설명되어 있습니다.

*Sun StorEdge QFS 및 Sun StorEdge SAM-FS 파일 시스템 관리 안내서*에 데몬에 대한 개요 정보가 있지만 개별적인 데몬들은 적절한 문서 세트에서 설명됩니다.

이 절은 명령을 소개하고 Sun StorEdge QFS 또는 Sun StorEdge SAM-FS 파일 시스템 내에서 사용할 수 있는 명령의 종류를 표시합니다. 자세한 정보는 소프트웨어 배포에 포함되어 있는 매뉴얼 페이지를 참조하십시오.

본 절에서는 다음 주제를 다룹니다.

- 6 페이지의 "사용자 명령"
- 7 페이지의 "일반 시스템 관리자 명령"
- 7 페이지의 "파일 시스템 명령"
- 9 페이지의 "자동화 라이브러리 명령"
- 10 페이지의 "아카이버 명령"
- 10 페이지의 "특수 유지 보수 명령"
- 11 페이지의 "사용자 정의 가능 사이트 스크립트"
- 12 페이지의 "애플리케이션 프로그래머 인터페이스"
- 12 페이지의 "작동 유틸리티"

사용자 명령

기본적으로 파일 시스템은 최종 사용자에게 공개되어 있습니다. 그러나 사이트 관행에 따라 사이트에서 사용자가 일부 명령어를 사용하여 특정 작동의 미세 조정을 허용할 수 있습니다. 표 1-2에 이러한 명령의 요약이 나옵니다.

표 1-2 사용자 명령

명령	설명	사용 제품
archive(1)	파일을 아카이브하고 파일의 아카이브 속성을 설정합니다.	Sun StorEdge SAM-FS
release(1)	디스크 공간을 해제하고 파일의 릴리스 속성을 설정합니다.	Sun StorEdge SAM-FS
request(1)	제거 가능한 매체 파일을 작성합니다.	Sun StorEdge SAM-FS
sdu(1)	디스크 사용을 요약합니다. sdu(1) 명령은 du(1) 명령의 GNU 버전에 기반합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
segment(1)	세그먼트화된 파일 속성을 설정합니다.	Sun StorEdge SAM-FS
setfa(1)	파일 속성을 설정합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
sfind(1)	디렉토리 계층 구조에서 파일을 검색합니다. sfind(1) 명령은 find(1) 명령의 GNU 버전을 기반으로 하며 Sun StorEdge QFS 및 Sun StorEdge SAM-FS 파일 속성을 기반으로 한 검색용 옵션을 포함합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
sls(1)	디렉토리 내용을 열거합니다. sls(1) 명령은 ls(1) 명령의 GNU 버전에 기반하며 파일 시스템 속성과 정보 표시용 옵션을 포함합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
squota(1)	할당 정보를 보고합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
ssum(1)	파일의 체크섬 속성을 설정합니다.	Sun StorEdge SAM-FS
stage(1)	파일의 스테이지 속성을 설정하고 오프라인 파일을 디스크로 복사합니다.	Sun StorEdge SAM-FS

일반 시스템 관리자 명령

표 1-3은 시스템을 유지 및 관리하기 위해 사용할 수 있는 명령을 요약합니다.

표 1-3 일반 시스템 관리자 명령

명령	설명	사용 제품
samcmd(1M)	하나의 samu(1M) 운영자 인터페이스 유틸리티를 실행합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
samd(1M)	제거 가능한 로봇식 매체 데몬을 시작하거나 중지합니다.	Sun StorEdge SAM-FS
samcmd(1M)	Sun StorEdge QFS 또는 Sun StorEdge SAM-FS 진단 보고 스크립트를 생성합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
samset(1M)	Sun StorEdge SAM-FS 설정을 변경합니다.	Sun StorEdge SAM-FS
samu(1M)	전체 화면의 텍스트 기반 운영자 인터페이스를 호출합니다. 이 인터페이스는 curses(3X) 소프트웨어 라이브러리에 기반합니다. samu 유틸리티는 장치 상태를 표시하며 운영자의 자동화된 파일 제어를 허용합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS

파일 시스템 명령

표 1-4는 파일 시스템을 유지하기 위해 사용할 수 있는 명령을 요약합니다.

표 1-4 파일 시스템 명령

명령	설명	사용 제품
mount(1M)	파일 시스템을 마운트합니다. 이 명령어에 대한 메뉴얼 페이지 이름은 mount_samfs(1M)입니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
qfsdump(1M) qfsrestore(1M)	Sun StorEdge QFS Sun QFS 파일 시스템과 연관된 파일 데이터와 메타 데이터를 포함하는 덤프 파일을 작성 또는 복원합니다.	Sun StorEdge QFS
sambcheck(1M)	파일 시스템용 블록 사용을 열거합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
samchaid(1M)	파일 관리 설정 ID 속성을 변경합니다. 할당과 함께 사용합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
samfsck(1M)	파일 시스템의 메타 데이터 불일치를 확인하고 고치며, 할당되었으나 사용되지 않고 있던 디스크 공간을 사용합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS

표 1-4 파일 시스템 명령 (계속)

명령	설명	사용 제품
samfsconfig(1M)	구성 정보를 표시합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
samfsdump(1M) samfsrestore(1M)	Sun StorEdge SAM-FS 파일 시스템에 관련된 메타데이터 덤프 파일을 작성 또는 복원합니다.	Sun StorEdge SAM-FS Sun SAM-QFS
samfsinfo(1M)	Sun StorEdge QFS 또는 Sun StorEdge SAM-FS 파일 시스템의 레이아웃에 대한 정보를 표시합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
samfstyp(1M)	Sun StorEdge QFS 또는 Sun StorEdge SAM-FS 파일 시스템 유형을 결정합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
samgrowfs(1M)	디스크 장치를 추가하여 파일 시스템을 확장합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
sammkfs(1M)	디스크 장치에서 새로운 파일 시스템을 초기화합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
samncheck(1M)	마운트 지점과 inode 번호에 제공된 전체 디렉토리 경로명을 반환합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
samquota(1M)	할당 정보를 보고, 설정 또는 재설정합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
samquotastat(1M)	활성 및 비활성 파일 시스템 할당을 보고합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
samsharefs(1M)	Sun StorEdge QFS 공유 파일 시스템 구성 정보를 조작합니다.	Sun StorEdge QFS
samtrace(1M)	추적 버퍼를 덤프합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
samunhold(1M)	SANergy 파일 보류를 해제합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS
trace_rotate(1M)	추적 파일을 회전합니다.	Sun StorEdge QFS Sun StorEdge SAM-FS

자동화 라이브러리 명령

표 1-5는 Sun StorEdge SAM-FS 환경 내의 자동화 라이브러리 및 장치를 구성, 초기화 및 유지하는 데 사용할 수 있는 자동화 라이브러리 명령을 요약합니다.

표 1-5 자동화 라이브러리 명령

명령	설명
auditslot(1M)	지정된 자동화 라이브러리에서 단일 매체 카트리지를 슬롯을 감사합니다.
build_cat(1M)	자동화 라이브러리를 매체 카탈로그 파일을 구성합니다. 선택적으로 카탈로그 파일을 상주시킬 수 있습니다.
chmed(1M)	특정 카트리지의 라이브러리 카탈로그 플래그와 값을 설정하거나 지울 수 있습니다.
cleandrive(1M)	테이프 드라이브에 클리닝 테이프 로드를 요청합니다.
dump_cat(1M)	다양한 ASCII 형식의 바이너리 카탈로그 파일 콘텐츠를 표시합니다.
import(1M) samexport(1M)	우편함에 카트리지를 배치하여 라이브러리에서 카트리지를 가져오거나 내보낼 수 있습니다. 네트워크 첨부 라이브러리의 경우 이 명령은 라이브러리 카탈로그를 업데이트하지만 물리적으로 카트리지를 이동시키지는 않습니다.
samload(1M) unload(1M)	지정된 장치의 카트리지를 로드 또는 언로드합니다.
move(1M)	하나의 슬롯에서 다른 슬롯으로 카트리지를 이동합니다.
odlabel(1M)	Sun StorEdge SAM-FS 시스템과 사용하기 위해 광 디스크를 레이블합니다.
samdev(1M)	논리적 장치 항목인 /dev/samst를 추가합니다. 자동화 라이브러리, 광 디스크, 테이프 드라이브 정보와 통신하기 위해 사용합니다.
tplabel(1M)	Sun StorEdge SAM-FS 시스템과 사용하기 위해 테이프를 레이블합니다.

아카이버 명령

표 1-6은 Sun StorEdge SAM-FS 환경 내에서 아카이버의 작업을 제어하는 명령을 요약합니다.

표 1-6 아카이버 명령

명령	설명
archiver(1M)	아카이버 명령 파일에서 구문의 완성도와 의미의 정확성을 평가합니다.
archiver.sh(1M)	예외적인 아카이버 이벤트를 로그합니다.
showqueue(1M)	아카이버 대기 파일의 콘텐츠를 표시합니다.
reserve(1M)	블록을 저장 및 저장 해제합니다.
unreserve(1M)	

특수 유지 보수 명령

표 1-7은 Sun StorEdge SAM-FS 환경에서 사용할 수 있는 다양한 유지 보수 명령을 요약합니다.

표 1-7 특수 유지 보수 명령

명령	설명
archive_audit(1M)	각 카트리지에 모든 아카이브된 파일의 보고서를 생성합니다.
dmpshm(1M)	공유 메모리 세그먼트를 덤프합니다.
exarchive(1M)	아카이브 복사본을 조작(교환)합니다.
itemize(1M)	광디스크 목록을 작성합니다.
rearch(1M) unrearch(1M)	다시 아카이브된 아카이브 항목을 표시 또는 표시 해제합니다.
sam-recycler(1M)	아카이브 매체에서 만료된 아카이브 복사본이 사용하던 공간을 사용합니다.
sam-releaser(1M)	온라인 디스크 캐시 파일 시스템에서 디스크 공간을 해제합니다.
samdev(1M)	Sun StorEdge SAM-FS 파일 시스템에 의해 사용되는 실제 장치를 지정하는 /dev/samst 디렉토리의 심볼릭 링크를 작성합니다. 이 명령은 UNIX의 makedev(1M) 명령과 기능이 유사합니다.
samset(1M)	Sun StorEdge SAM-FS 작동에 사용된 변수를 변경 또는 표시합니다.
set_admin(1M)	관리자 명령어를 실행할 관리자 그룹용 승인을 추가 또는 제거해야 합니다.
set_state(1M)	Sun StorEdge SAM-FS 장치의 상태를 설정합니다.

표 1-7 특수 유지 보수 명령 (계속)

명령	설명
stageback.sh(1M)	Sun StorEdge SAM-FS 또는 Sun SAM-QFS 아카이브 테이프로부터 파일을 스테이지합니다.
star(1M)	테이프 아카이브를 작성하고 파일을 추가 또는 추출합니다. 이것은 tar(1) 명령의 GNU 버전이며 Sun StorEdge SAM-FS 파일 시스템과 사용하도록 확장되었습니다. 아카이브 테이프로부터 데이터를 읽어야 할 경우 이 명령을 재난 복구 상황에 사용할 수 있습니다.
tapealert(1M)	TapeAlert 이벤트를 해독합니다.
unarchive(1M)	하나 이상의 파일에서 아카이브 항목을 삭제합니다.
undamage(1M)	하나 이상의 파일이나 디렉토리용 아카이브 항목을 손상되지 않은 것으로 표시합니다.

사용자 정의 가능 사이트 스크립트

표 1-8은 Sun StorEdge SAM-FS 환경을 모니터 및 제어하는 데 사용할 수 있는 사용자 정의 가능 사이트 스크립트를 요약합니다. 기본적으로 소프트웨어는 /opt/SUNWsamfs/examples의 스크립트들을 설치합니다. /opt/SUNWsamfs/examples에서 /etc/opt/SUNWsamfs/scripts로 스크립트를 이동하고 사이트에 원하는 작업을 수행하도록 수정할 수 있습니다. 이러한 스크립트에 대한 자세한 내용은 각각의 매뉴얼 페이지를 참조하십시오.

표 1-8 사용자 정의 가능 사이트 스크립트

스크립트	설명
dev_down.sh(1M)	장치를 따로 적어두거나 표시를 해둔 경우 root로 전자우편을 보냅니다.
load_notify.sh(1M)	Sun StorEdge SAM-FS 소프트웨어가 라이브러리 밖에 있는 카트리지를 요청할 때 운영자에게 통지합니다.
log_rotate.sh(1M)	로그 파일을 회전합니다.
recover.sh(1M)	최신 samfsdump(1M)을 처리한 후 아카이브된 파일을 복구합니다.
restore.sh(1M)	온라인 또는 부분적인 온라인 상태로 파일을 복원합니다.
stageback.sh(1M)	아카이브 매체로부터 파일을 스테이지합니다.
tarback.sh(1M)	아카이브 매체로부터 파일을 재로드합니다.

애플리케이션 프로그래머 인터페이스

사용자 애플리케이션에 소 파일 시스템 요청을 하기 위해 애플리케이션 프로그래머 인터페이스(API)를 사용할 수 있습니다. 파일 시스템이 실행되는 시스템으로 지역 또는 원격 요청을 작성할 수 있습니다. API는 `libsam` 및 `libsamrpc` 라이브러리로 구성됩니다. 이러한 라이브러리에는 파일 상태 얻기, 파일에 대한 아카이브, 릴리스 및 스테이지 속성 설정, 자동화 라이브러리의 라이브러리 카탈로그 조작을 위한 라이브러리 루틴이 포함되어 있습니다. `sam-rpcd` 원격 절차 호출 데몬은 원격 요청을 다룹니다. `sam-rpcd` 데몬을 자동으로 시작하려면 `defaults.conf` 파일의 `samrpc=on`을 설정합니다.

API에 대한 자세한 내용을 보려면 `intro_libsam(3)` 매뉴얼 페이지를 참조하십시오. 이 매뉴얼 페이지는 `libsam` 및 `libsamrpc` 라이브러리 루틴을 사용하기 위한 개략적 정보를 제공합니다.

작동 유틸리티

Sun StorEdge SAM-FS 환경 내에서 `samu(1M)` 운영자 유틸리티 및 SAM-QFS Manager를 사용하여 기본 작동을 수행할 수 있습니다. 표 1-9는 작동 도구를 요약합니다.

표 1-9 작동 도구

GUI 도구	설명
SAM-QFS Manager	Sun StorEdge QFS 및 Sun StorEdge SAM-FS 소프트웨어에 웹 기반 그래픽 사용자 인터페이스를 제공합니다. Sun StorEdge QFS 및 Sun StorEdge SAM-FS 환경의 구성요소를 구성, 제어, 모니터 및 재구성하는 데 이 인터페이스를 사용할 수 있습니다. SAM-QFS Manager 설치에 대한 정보는 <i>Sun StorEdge QFS 및 Sun StorEdge SAM-FS 소프트웨어 설치 및 구성 안내서</i> 를 참조하십시오. SAM-QFS Manager 사용에 대한 정보는 해당 온라인 도움말을 참조하십시오.
<code>samu(1M)</code>	<code>samu(1M)</code> 운영자 유틸리티 액세스용 시작점을 제공합니다.

Sun StorEdge SAM-FS 환경에서 자동화 라이브러리 및 수동으로 로드된 드라이브 사용하기

자동화 라이브러리란 제거 가능한 카트리지를 운영자가 조작하지 않아도 로드 및 언로드할 수 있는 로봇 제어 장치를 말합니다. 카트리지는 라이브러리에서 가져오거나 라이브러리로 내보냅니다. 카트리지는 자동으로 로드 및 언로드됩니다. 아카이브 및 스테이지 프로세스에서는 사이트 정의 방식을 통해 사용할 드라이브의 수를 할당합니다. 자동화 라이브러리를 매체 교환기, 주크박스, 로봇, 라이브러리 또는 매체 라이브러리라고도 합니다.

다음 절에서는 Sun StorEdge SAM-FS 환경에서 라이브러리를 사용하는 여러 가지 측면에 대해 설명합니다. *Sun StorEdge QFS 및 Sun StorEdge SAM-FS 소프트웨어 설치 및 구성 안내서*에는 초기 구성 지침이 나와있고, 이 장에서는 자동화 라이브러리와 수동 로드 드라이브의 작동 지침에 대해 설명합니다. 이 장에서는 또한 요청된 볼륨이 라이브러리에 없는 경우에 이를 운영자에게 알리는 운영자 중심의 로드 통지 기능에 대해 설명합니다.

참고 – Sun StorEdge SAM-FS 소프트웨어는 많은 제조업체의 자동화 라이브러리와 상호 작동합니다. 라이브러리 모델 번호, 펌웨어 레벨 및 기타 호환성 정보에 관련된 정보에 대해서는 Sun 고객 지원에 문의하십시오.

일부 자동화 라이브러리의 경우, 특정 작업에 이 장에서 설명하는 작업과 다른 기능이 있습니다. Sun StorEdge SAM-FS 환경에서 사용할 때 자동화 라이브러리에 추가의 업체별 작동 지침이 있는지 결정하려면 37 페이지의 "작동 절차가 업체별로 고유한 라이브러리의 기본 작업"을 확인하십시오.

본 절에서는 다음 주제를 다룹니다.

- 14 페이지의 "규칙"
- 16 페이지의 "자동화 라이브러리 작업"
- 35 페이지의 "수동으로 로드된 드라이브 작업"

규칙

일반적으로 이 장에 설명된 기본 작업 수행 절차는 `samcmd(1M)` 명령, `samu(1M)` 운영자 유틸리티 및 다음 명령의 사용 방법을 표시합니다.

- `tplabel(1M)`
- `odlabel(1M)`
- `auditslot(1M)`
- `cleandrive(1M)`
- `chmed(1M)`
- `import(1M)`
- `set_state(1M)`
- `samexport(1M)`

그러나 작업을 수행하는 방법은 대부분 한 가지 이상입니다. 웹 기반 그래픽 사용자 인터페이스(GUI)인 SAM-QFS Manager 내에서부터 Sun StorEdge QFS 및 Sun StorEdge SAM-FS 소프트웨어까지 이러한 많은 작업들을 수행할 수 있습니다. Sun StorEdge QFS 및 Sun StorEdge SAM-FS 환경의 구성요소를 구성, 제어, 모니터 및 재구성하는 데 이 인터페이스를 사용할 수 있습니다. SAM-QFS Manager 설치에 대한 정보는 *Sun StorEdge QFS 및 Sun StorEdge SAM-FS 소프트웨어 설치 및 구성 안내서*를 참조하십시오. SAM-QFS Manager 사용에 대한 정보는 해당 온라인 도움말을 참조하십시오.

명령 인수

대부분의 명령은 공통된 인수 집합을 허용합니다. 표 2-1은 이러한 인수를 표시합니다.

표 2-1 명령 인수

인수	의미
<code>eq</code>	mcf 파일의 정의에 따라 어드레스되는 장치의 장비 서수 식별된 장치는 자동화 라이브러리, 드라이브 또는 파일 시스템일 수 있습니다.
<code>slot</code>	라이브러리 카탈로그에서 인식된 자동화 라이브러리의 저장소 슬롯 수.
<code>partition</code>	광자기 디스크의 한 면. 파티션은 1개나 2개여야 합니다.
<code>media_type</code>	매체 유형. 올바른 매체 유형의 목록에 대해서는 <code>mcf(4)</code> 매뉴얼 페이지를 참조하십시오.
<code>vsn</code>	볼륨에 할당된 볼륨 시리얼 이름

일부 명령은 현재의 환경에 따라 다양한 인수 조합을 허용합니다. 예를 들어 samu(IM) 운영자 유틸리티에서 load 명령은 다음과 같은 두 가지 형식을 갖습니다.

```
:load eq:slot
:load 매체유형.vsn
```

다음을 유의하십시오.

- 첫 번째 형식에서는 콜론(:)을 사용해 *eq*와 *slot*를 구분합니다.
- 두 번째 형식에서는 마침표(.)를 사용해 *media_type*과 *vsn*을 구분합니다.

용어

이 장에서는 일부 새로운 용어들이 사용되었습니다. 표 2-2는 가장 일반적으로 사용된 용어 및 그 의미를 표시합니다.

표 2-2 용어

용어	의미
자동화 라이브러리	테이프 및 광 카트리지를 저장하기 위한 자동화 장치.
카트리지	테이프 또는 광자기 카트리지. 광자기 카트리지는 하나 이상의 볼륨이나 파티션을 포함할 수 있습니다.
파티션	전체 테이프 또는 광자기 디스크의 한 면. 파티션에는 볼륨이 하나만 포함될 수 있습니다.
볼륨	데이터 저장용 카트리지의 명명된 영역. 카트리지는 하나 이상의 볼륨을 포함할 수 있습니다. 두 면이 있는 카트리지는 각 면에 하나씩 두 개의 볼륨을 가집니다. 볼륨 시리얼 이름(VSN)으로 볼륨을 식별합니다.

자동화 라이브러리 작업

몇 가지 기본 작업은 모든 자동화 라이브러리에 대해 동일합니다. 이 섹션에서는 다음과 같은 기본 작업을 설명합니다.

- 17 페이지의 "제거 가능한 매체 작업을 시작하려면"
- 16 페이지의 "제거 가능한 매체 작업을 중지하려면"
- 17 페이지의 "자동화 라이브러리를 실행하려면"
- 18 페이지의 "자동화 라이브러리를 해제하려면"
- 18 페이지의 "자동화 라이브러리에 카트리지를 로드하려면"
- 19 페이지의 "드라이브에서 카트리지를 언로드하려면"
- 20 페이지의 "카트리지에 레이블 지정"
- 22 페이지의 "볼륨을 감사하려면"
- 23 페이지의 "자동화 라이브러리를 감사하려면 (직접 연결 전용)"
- 23 페이지의 "클리닝 카트리지 사용"
- 26 페이지의 "테이프 드라이브를 클리닝하려면"
- 26 페이지의 "매체 오류를 지우려면"
- 28 페이지의 "드라이브에서 고정된 카트리지를 제거하려면"
- 29 페이지의 "카탈로그 작업, 카트리지 가져오기 및 카트리지 내보내기"
- 34 페이지의 "로드 통지 사용하기"

▼ 제거 가능한 매체 작업을 중지하려면

제거 가능한 매체 작업을 중지하고 Sun StorEdge SAM-FS 시스템을 마운트 상태로 들 수 있습니다. 예를 들어 라이브러리에서 카트리지를 수동으로 조작하는 경우에 이렇게 할 수 있습니다. 작업이 재개되면 보류 스테이지가 재실행되고 아카이브가 재개됩니다.

- **제거 가능한 매체 작업을 중지하려면 `samcmd(1M) idle` 및 `samd(1M) stop` 명령을 사용하십시오.**

이 명령을 다음 양식으로 사용합니다.

```
samcmd idle eq  
samd stop
```

*eq*에 대해, *mcf* 파일의 정의에 따라 어드레스되는 장비의 장비 서수를 입력합니다. 드라이브를 유틸리티 상태로 만들려면 *mcf* 파일에 구성된 각 *eq*에 대해 `samcmd idle eq` 명령을 입력합니다.

또한 `samu(1M)` 운영자 유틸리티 또는 SAM-QFS Manager를 사용하여 드라이브를 유틸리티 상태로 둘 수 있습니다.

참고 – Sun StorEdge SAM-FS 환경의 드라이브는 `samd(1M) stop` 명령을 실행하기 전에 유틸리티 상태여야 합니다. 이렇게 해야 아카이브, 스테이지 및 기타 프로세스가 현재 작업을 완료할 수 있습니다. `samd(1M) stop` 명령을 실행하지 않으면 아카이브, 스테이지 및 기타 작업을 재개할 때 예상치 못한 결과를 초래할 수 있습니다.

▼ 제거 가능한 매체 작업을 시작하려면

일반적으로 제거 가능한 매체 작업은 Sun StorEdge SAM-FS 파일 시스템이 마운트될 때 시작합니다.

- 파일 시스템을 마운트하지 않고 제거 가능한 매체 작업을 수동으로 시작하려면 다음과 같은 `samd(1M)` 명령을 입력합니다.

이 명령을 다음 양식으로 사용합니다.

```
# samd start
```

이 명령을 입력할 때 제거 가능한 매체 작업이 이미 실행 중인 경우에는 다음과 같은 메시지가 나타납니다.

```
SAM-FS sam-amld daemon already running
```

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

▼ 자동화 라이브러리를 실행하려면

라이브러리가 on 상태에 있는 경우, 이는 Sun StorEdge SAM-FS 시스템의 제어 하에 있으며 일반적인 작업을 사용하여 진행할 수 있습니다. 라이브러리를 켜 경우 Sun StorEdge SAM-FS 소프트웨어가 다음 작업을 수행합니다.

- 내부 상태에 관련된 장치를 질의합니다. 이를 통해 테이프의 위치, 바코드 사용 여부 등을 알 수 있습니다.
- 카탈로그 및 기타 내부 구조를 업데이트 합니다.

- **samcmd(1M) on** 명령을 사용하여 자동화 라이브러리를 켭니다.

이 명령을 다음 양식으로 사용합니다.

```
samcmd on eq
```

*eq*에 대해, *mcf* 파일의 정의에 따라 어드레스되는 자동화 라이브러리의 장비 서수를 지정합니다.

또한 *samu(1M)* 또는 SAM-QFS Manager를 사용하여 이 작업을 수행할 수 있습니다.

▼ 자동화 라이브러리를 해제하려면

라이브러리를 *off* 상태로 두면 I/O 작업을 중지하고 Sun StorEdge SAM-FS 제어로부터 자동화 라이브러리를 제거합니다. 카트리지가 자동으로 이동하지 않습니다. 자동화 라이브러리의 드라이브는 *on* 상태로 유지됩니다. 자동화 라이브러리를 끄고 다음과 같은 작업을 수행할 수 있습니다.

- 이 자동화 라이브러리에 대해서만 Sun StorEdge SAM-FS 작업 중지하기.
- 자동화 라이브러리 종료하기.

- **samcmd(1M) off** 명령을 사용하여 자동화 라이브러리를 끕니다.

이 명령을 다음 양식으로 사용합니다.

```
samcmd off eq
```

*eq*에 대해, *mcf* 파일의 정의에 따라 어드레스되는 자동화 라이브러리의 장비 서수를 지정합니다.

또한 *samu(1M)* 또는 SAM-QFS Manager를 사용하여 이 작업을 수행할 수 있습니다.

▼ 자동화 라이브러리에 카트리지를 로드하려면

아카이브나 스테이지를 위해 VSN을 요청하면 드라이브에 카트리지가 자동으로 로드됩니다. 로드된 카트리지를 저장소 슬롯에서 드라이브로 이동하여 준비 상태로 만드는 것을 말합니다.

- **samcmd(1M) load 명령을 사용하여 수동으로 카트리지를 로드합니다.**

이 명령은 드라이브가 `unavail` 상태인 경우에도 사용할 수 있습니다. 이 명령은 다음과 같은 두 가지 형식을 가질 수 있습니다.

```
samcmd load eq:slot[:partition]
samcmd load media_type.vsn
```

표 2-3 samcmd(1M) load에 대한 인수

인수	의미
<i>eq</i>	mcf 파일의 정의에 따라 어드레스되는 드라이브의 장비 서수.
<i>slot</i>	라이브러리 카탈로그에서 인식된 자동화 라이브러리의 저장소 슬롯 수.
<i>media_type</i>	매체 유형. 올바른 매체 유형의 목록에 대해서는 mcf(4) 매뉴얼 페이지를 참조하십시오.
<i>partition</i>	광자기 디스크의 한 면. 파티션은 1 또는 2여야 합니다. 이 인수는 테이프 카트리지에 적용할 수 없습니다.
<i>vsn</i>	볼륨에 할당된 볼륨 시리얼 이름

또한 `samu(1M)` 또는 `SAM-QFS Manager`를 사용하여 이 작업을 수행할 수 있습니다.

수동으로 로드한 카트리는 일반적으로 라이브러리에서 다음으로 사용 가능한 드라이브에 로드됩니다. 이를 위해 드라이브를 사용 불가능한 상태로 설정하려면 `samu(1M)` 유틸리티의 `:unavail` 명령을 사용하거나 `SAM-QFS Manager`를 사용해 장치의 상태를 변경합니다. 이러한 작업은 재해 복구 작업 과정 또는 테이프 분석 등을 위해 실행할 수 있습니다.

▼ 드라이브에서 카트리지를 언로드하려면

더 이상 볼륨이 필요하지 않으면 카트리가 자동으로 언로드됩니다. 수동으로 드라이브를 언로드할 수도 있습니다. 언로드란 드라이브에서 카트리지를 제거하는 것을 말합니다.

- **samcmd(1M) unload 명령을 사용하여 수동으로 카트리지를 언로드합니다.**

이 명령은 드라이브가 `unavail` 상태인 경우에도 사용할 수 있습니다. 이 명령을 다음 양식으로 사용합니다.

```
samcmd unload eq
```

eq에 대해, mcf 파일의 정의에 따라 어드레스되는 자동화 라이브러리의 장비 서수를 지정합니다.

또한 samu(1M) 또는 SAM-QFS Manager를 사용하여 이 작업을 수행할 수 있습니다.

카트리지에 레이블 지정

카트리지에 레이블을 지정하는 절차는 그 대상(테이프 또는 광 카트리지)에 따라 다릅니다. 다음 두 섹션에서는 이러한 절차에 대해 설명합니다.



주의 - 카트리지의 레이블을 지정 또는 변경하면 카트리지의 현재 데이터가 모든 소프트웨어에 액세스할 수 있습니다. 카트리지에 저장된 데이터가 필요하지 않다고 확실하는 경우에만 카트리지의 레이블을 변경하십시오.

▼ 테이프에 레이블 지정 또는 변경하려면

다음 tplabel(1M) 명령 행 형식은 테이프에 레이블을 작성하거나 변경할 때 가장 자주 사용하는 옵션을 보여줍니다.

```
tplabel [ -new | -old vsn ] -vsn vsn eq:slot
```

표 2-4 tplabel(1M)에 대한 인수

인수	의미
<i>vsn</i>	볼륨 시리얼 이름 레이블을 변경하는 경우에는 새 VSN 이름이 기존의 VSN 이름과 같을 수 있습니다.
<i>eq</i>	mcf 파일의 정의에 따라 어드레스되는 자동화 라이브러리 또는 수동 로드 드라이브의 장비 서수.
<i>slot</i>	라이브러리 카탈로그에서 인식된 자동화 라이브러리의 저장소 슬롯 수 수동으로 로드된 드라이브에는 이 인수를 적용할 수 없습니다.

● 새 테이프에 레이블을 지정하려면 tplabel(1M) 명령을 사용합니다.

이 명령을 다음 양식으로 사용합니다.

```
tplabel -new -vsn vsn eq:slot
```

- 기존 테이프의 레이블을 변경하려면 **tplabel(1M)** 명령을 사용합니다.
이 명령을 다음 양식으로 사용합니다.

```
tplabel -old vsn -new -vsn vsn eq:slot
```

명령을 실행해 테이프에 레이블을 지정 또는 변경하면 테이프가 로드되고, 위치가 지정되며, 테이프 레이블이 작성됩니다. **tplabel(1M)** 명령에 대한 자세한 내용은 **tplabel(1M)** 매뉴얼 페이지를 참조하십시오.

또한 SAM-QFS Manager를 사용하여 이 작업을 수행할 수도 있습니다.

▼ 광 디스크에 레이블을 지정 또는 변경하려면

다음 **odlabel(1M)** 명령 행 형식은 광 디스크에 레이블을 작성하거나 변경할 때 가장 자주 사용하는 옵션을 보여줍니다.

```
odlabel [ -new | -old vsn ] -vsn vsn eq:slot:partition
```

표 2-5 odlabel(1M)에 대한 인수

인수	의미
<i>vsn</i>	볼륨 시리얼 이름 레이블을 변경하는 경우에는 새 VSN 이름이 기존의 VSN 이름과 같을 수 있습니다.
<i>eq</i>	mcf 파일의 정의에 따라 어드레스되는 자동화 라이브러리 또는 수동 로드 드라이브의 장비 서수.
<i>slot</i>	라이브러리 카탈로그에서 인식된 자동화 라이브러리의 저장소 슬롯 수 수동으로 로드된 드라이브에는 이 인수를 적용할 수 없습니다.
<i>파티션</i>	광자기 디스크의 한 면. 파티션은 1개나 2개여야 합니다. 이 인수는 테이프 카트리지에 적용할 수 없습니다.

- 새 광 디스크에 레이블을 지정하려면 **odlabel(1M)** 명령을 사용합니다.
이 명령을 다음 양식으로 사용합니다.

```
odlabel -new -vsn vsn eq:slot:partition
```

- 기존 광 디스크의 레이블을 변경하려면 **odlabel(1M)** 명령을 사용합니다.
이 명령을 다음 양식으로 사용합니다.

```
odlabel -old vsn -vsn vsn eq:slot:partition
```

명령을 실행하여 광 디스크에 레이블을 지정하거나 변경하면 광 디스크가 로드되고 위치가 지정되며 테이프 레이블이 작성됩니다. `odlabel(1M)` 명령에 대한 자세한 내용은 `odlabel(1M)` 매뉴얼 페이지를 참조하십시오.

또한 SAM-QFS Manager를 사용하여 이 작업을 수행할 수도 있습니다.

▼ 볼륨을 감사하려면

때때로 테이프나 광 카트리지에 남아있는 보고된 공간을 라이브러리 카탈로그에서 업데이트해야 합니다. `auditslot(1M)` 명령은 볼륨을 포함한 카트리지를 로드하고 레이블을 읽고 슬롯에 대한 라이브러리 카탈로그 항목을 업데이트합니다.

● `auditslot(1M)` 명령을 사용하여 볼륨을 감사합니다.

이 명령을 다음 양식으로 사용합니다.

```
auditslot [-e] eq:slot[:partition]
```

표 2-6 `auditslot(1M)`에 대한 인수

인수	의미
<code>-e</code>	<code>-e</code> 옵션이 지정되고 매체가 테이프인 경우, 남아 있는 공간이 업데이트됩니다. 그렇지 않으면 나머지 공간이 변경되지 않습니다.
<code>eq</code>	<code>mcf</code> 파일의 정의에 따라 어드레스되는 자동화 라이브러리 또는 수동 로드 드라이브의 장비 서수.
<code>slot</code>	라이브러리 카탈로그에서 인식된 자동화 라이브러리의 저장소 슬롯 수 수동으로 로드된 드라이브에는 이 인수를 적용할 수 없습니다.
<code>partition</code>	광자기 디스크의 한 면. 파티션은 1개나 2개여야 합니다. 이 인수는 테이프 카트리지에 적용할 수 없습니다.

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

또한 `samu(1M)` 유틸리티의 `:audit` 명령 또는 SAM-QFS Manager를 사용하여 이 작업을 수행할 수 있습니다.

▼ 자동화 라이브러리를 감사하려면 (직접 연결 전용)

참고 - 네트워크 연결 자동화 라이브러리에 대해서는 이 작업을 수행할 수 없습니다.

전체 감사는 각 카트리지를 드라이브에 로드하고 레이블을 읽고 라이브러리 카탈로그를 업데이트합니다. 다음과 같은 상황에서는 라이브러리를 감사해야 합니다.

- Sun StorEdge SAM-FS 명령을 사용하지 않고 자동화 라이브러리의 카트리지를 이동한 후.
 - 라이브러리 카탈로그의 상태를 모르는 상태에서 이를 업데이트하려는 경우(예를 들면, 정전 후).
 - 편지함이 없는 자동화 라이브러리에서 카트리지를 추가, 제거 또는 이동한 경우.
- **samcmd(1M) audit 명령을 사용하여 자동화 라이브러리에서 전체 감사를 수행합니다.** 이 명령을 다음 양식으로 사용합니다.

```
samcmd audit eq
```

*eq*에 대해, *mcf* 파일의 정의에 따라 어드레스되는 자동화 라이브러리의 장비 서수를 지정합니다.

또한 *samu(1M)* 유틸리티의 `:audit` 명령 또는 SAM-QFS Manager를 사용하여 이 작업을 수행할 수 있습니다.

클리닝 카트리지 사용

Sun StorEdge SAM-FS 시스템을 사용하면 클리닝 카트리지를 가져와 테이프 드라이브를 청소할 수 있습니다. 이 절차는 클리닝 카트리지에 바코드가 있는지 여부에 따라 달라집니다. 다음 섹션에서는 클리닝 카트리지를 사용하는 여러 가지 측면에 대해 설명합니다.

클리닝 방법은 제조업체마다 다릅니다. 이러한 작업에 문제가 있는 경우에는 37 페이지의 "작동 절차가 업체별로 고유한 라이브러리의 기본 작업"을 참조하여 자신의 장비에 대해 권장된 특수 절차가 있는지를 알아보십시오.

참고 - 네트워크 연결 자동화 라이브러리에 대해서는 이 작업을 수행할 수 없습니다.

▼ 클리닝 주기 횟수를 재설정하려면

클리닝 테이프는 제한된 횟수의 클리닝 주기에 대해서만 유용하게 사용할 수 있습니다. samu(1M) 유틸리티의 :v 화면을 사용하거나 SAM-QFS Manager 내에서 남아 있는 주기의 횟수를 볼 수 있습니다.

Sun StorEdge SAM-FS 시스템은 각 클리닝 테이프에 대해 사용되는 클리닝 주기의 횟수를 추적하여 남아있는 사이클이 0이 될 때 테이프를 배출합니다. 예를 들어 DLT 클리닝 테이프의 주기는 20이고 Exabyte 클리닝 테이프의 주기는 10입니다. 클리닝 테이프를 가져오면 클리닝 주기는 해당 테이프 유형의 최대 사이클 횟수로 재설정됩니다.

현재 시스템에서 자동 클리닝을 사용할 수 있으나 자동화 라이브러리의 모든 클리닝 테이프의 카운트가 0이면 드라이브는 Off로 설정되고 Sun StorEdge SAM-FS 로그에 메시지가 발생합니다.

● chmed(1M) 명령을 사용하여 0의 계수로 클리닝 테이프를 재설정합니다.

이 명령을 다음 양식으로 사용합니다.

```
chmed -count count media_type.vsn
```

표 2-7 chmed(1M)에 대한 인수

인수	의미
<i>count</i>	클리닝 테이프를 재설정하려는 클리닝 주기 횟수.
<i>media_type</i>	매체 유형. 올바른 매체 유형의 목록에 대해서는 mcf(4) 매뉴얼 페이지를 참조하십시오.
<i>vsn</i>	볼륨에 할당된 볼륨 시리얼 이름

▼ 바코드가 있는 클리닝 카트리지를 사용하기

클리닝 카트리지에 바코드가 있으면 import(1M) 명령을 사용해 이를 가져올 수 있습니다.

1. 클리닝 카트리지에 CLEAN 바코드가 있는지 또는 CLN 문자로 시작하는지 확인하십시오.

2. import(1M) 명령을 사용해 카트리지를 가져옵니다.

이 명령을 다음 양식으로 사용합니다.

```
import eq
```

*eq*에 대해, mcf 파일의 정의에 따라 어드레스되는 자동화 라이브러리의 장비 서수를 지정합니다.

Sun StorEdge SAM-FS 시스템은 편지함에서 저장소 슬롯으로 카트리지를 이동하고 각 카트리지에 대해 라이브러리 카탈로그를 업데이트합니다. 이 명령을 실행하면 클리닝 매체 플래그가 설정되고, 액세스 카운트가 매체 유형에 따라 알맞은 클리닝 주기 횟수로 설정됩니다. 매체를 사용해 드라이브를 청소할 때마다 액세스 카운트가 감소합니다.

예를 들어 다음과 같은 명령을 사용하면 `mcf` 파일에서 50이라는 번호가 붙은 자동화 라이브러리로 클리닝 테이프를 가져올 수 있습니다.

```
# import 50
```

또한 `samu(1M)` 또는 `SAM-QFS Manager`를 사용하여 이 작업을 수행할 수 있습니다.

▼ 바코드가 없는 클리닝 카트리지 사용하기

카트리지에 바코드가 없으면 먼저 카트리지를 가져와야 합니다. 이 카트리는 클리닝 카트리지로 표시되지 않습니다. 다음과 같은 작업을 수행하십시오.

1. `import(1M)` 명령을 사용해 카트리지를 가져옵니다.

이 명령을 다음 양식으로 사용합니다.

```
import eq
```

`eq`에 대해, `mcf` 파일의 정의에 따라 어드레스되는 자동화 라이브러리의 장비 서수를 지정합니다.

2. `chmed(1M)` 명령을 사용해 카트리지 유형을 클리닝 카트리지로 변경합니다.

자동화 라이브러리의 장비 서수와 클리닝 카트리지가 로드되는 슬롯을 알아야 합니다. 다음 예의 명령 행에서 자동화 라이브러리의 장비 서수는 50이고 클리닝 카트리지 슬롯은 77입니다.

```
# chmed +C 50:77
```

이 명령을 실행하면 카트리지 유형이 클리닝 카트리지로 변경됩니다.

3. `chmed(1M)`을 다시 사용해 클리닝 주기 횟수를 설정합니다.

다음 예의 명령은 앞의 작업에서 사용한 카트리지에 대해 카운트를 설정합니다.

```
# chmed -count 20 50:77
```

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

▼ 테이프 드라이브를 클리닝하려면

참고 - Sun StorEdge SAM-FS 시스템은 네트워크 연결 라이브러리에서 자동 클리닝을 지원하지 않습니다. 자동 클리닝하려면 업체의 라이브러리 관리 소프트웨어를 사용해야 합니다.

Sun StorEdge SAM-FS 환경은 하드웨어가 클리닝 테이프를 지원하는 경우 클리닝 테이프의 사용을 지원합니다. 테이프 드라이브에서 클리닝을 요청하면 시스템은 자동으로 클리닝 테이프를 로드합니다.

시스템에서 바코드가 있는 레이블을 사용하는 경우, 클리닝 테이프의 VSN은 CLEAN이거나 바코드 레이블에서 CLN으로 시작해야 합니다. 또는 `chmed(1M)` 명령을 사용해 VSN을 클리닝 테이프로 표시하고 횟수를 설정할 수도 있습니다. 하나의 시스템에서 여러 클리닝 테이프를 사용할 수 있습니다.

참고 - 특정 드라이브 오류는 모든 클리닝 주기를 사용할 때까지 클리닝 카트리지를 반복해서 로드할 수도 있습니다. `chmed(1M)` 명령을 사용해 클리닝 카트리지에 대한 클리닝 주기 횟수를 제한하면 이러한 상황을 방지할 수 있습니다. 예를 들어, 다음과 같습니다.

```
# chmed -count 20 50:77
```

자동 클리닝을 사용할 수 없고 시스템이 바코드를 사용하는 경우 다음 절차를 수행하여 드라이브를 청소하도록 수동으로 요청하십시오.

- **cleandrive(1M) 명령을 사용합니다.**

이 명령을 다음 양식으로 사용합니다.

```
cleandrive eq
```

`eq`에 대해, `mcf` 파일의 정의에 따라 어드레스되는 자동화 라이브러리의 장비 서수를 지정합니다. 이는 클리닝 카트리지와 함께 로드되는 드라이브입니다.

▼ 매체 오류를 지우려면

카트리지에 하드웨어 또는 소프트웨어 오류가 발생하면 Sun StorEdge SAM-FS 시스템에서는 VSN 카탈로그에 `media error` 플래그를 설정합니다. `media error` 신호가 발생한 카트리지에서는 `chmed(1M)` 명령을 사용해 오류를 지우고 카트리지 사용을 시도할 수 있습니다. `media error` 플래그는 `samu(1M)` 유틸리티의 `v` 화면 및 SAM-QFS Manager에 표시됩니다.

1. **chmed(1M) 명령을 실행해 media error 플래그를 지웁니다.**

다음 양식으로 이 명령을 사용하여 media error 플래그를 지웁니다.

```
chmed -E media_type.vsn
```

표 2-8 chmed(1M)에 대한 인수

인수	의미
<i>media_type</i>	매체 유형. 올바른 매체 유형의 목록에 대해서는 mcf(4) 매뉴얼 페이지를 참조하십시오.
<i>vsn</i>	볼륨에 할당된 볼륨 시리얼 이름.

2. **auditslot(1M) 명령을 실행하여 남아 있는 정보를 업데이트합니다.**

이 명령을 다음 양식으로 사용합니다.

```
auditslot -e eq:slot[:partition]
```

표 2-9 auditslot(1M)에 대한 인수

인수	의미
-e	-e 옵션이 지정되고 매체가 테이프인 경우, 남아 있는 공간이 업데이트됩니다. 그렇지 않으면 나머지 공간이 변경되지 않습니다.
<i>eq</i>	mcf 파일의 정의에 따라 어드레스되는 자동화 라이브러리 또는 수동 로드 드라이브의 장비 서수
<i>slot</i>	라이브러리 카탈로그에서 인식된 자동화 라이브러리의 저장소 슬롯 수 수동으로 로드된 드라이브에는 이 인수를 적용할 수 없습니다.
<i>partition</i>	광자기 디스크의 한 면. 파티션은 1개나 2개여야 합니다. 이 인수는 테이프 카트리지에 적용할 수 없습니다.

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

또한 samu(1M) 유틸리티의 :audit 명령 또는 SAM-QFS Manager를 사용하여 이 작업을 수행할 수 있습니다.

▼ 드라이브에서 고정된 카트리지를 제거하려면

카트리지가 드라이브에 고정되면 다음과 같은 작업을 수행해야 합니다.

1. **samcmd(1M) off** 명령을 사용하여 자동화 라이브러리의 드라이브를 끕니다.
이 명령을 다음 양식으로 사용합니다.

```
samcmd off eq
```

*eq*에 대해, *mcf* 파일의 정의에 따라 어드레스되는 드라이브의 장비 서수를 지정합니다.
또한 *samu(1M)* 또는 SAM-QFS Manager를 사용하여 이 작업을 수행할 수 있습니다.

2. **samcmd(1M) off** 명령을 사용하여 자동화 라이브러리를 끕니다.
이 명령을 다음 양식으로 사용합니다.

```
samcmd off eq
```

*eq*에 대해, *mcf* 파일의 정의에 따라 어드레스되는 라이브러리의 장비 서수를 지정합니다.

또한 *samu(1M)* 또는 SAM-QFS Manager를 사용하여 이 작업을 수행할 수 있습니다.

3. 드라이브에서 카트리지를 물리적으로 제거합니다.

카트리지가나 드라이브를 손상시키지 않도록 주의하십시오.

4. **samcmd(1M) on** 명령을 사용하여 자동화 라이브러리 및 드라이브를 켭니다.

이 명령을 드라이브에 한 번 그리고 라이브러리에 한 번 실행합니다. 이 명령을 다음 양식으로 사용합니다.

```
samcmd on eq
```

*eq*에 대해, *mcf* 파일의 정의에 따라 어드레스되는 라이브러리 또는 드라이브의 장비 서수를 지정합니다.

자동화 라이브러리를 켜고 감사를 수행하면 작업이 완료된 것입니다. 그렇게 않으면 다음 작업을 수행합니다.

5. 카트리지를 다시 저장소 슬롯에 넣은 경우에는 **chmed(1M)** 명령으로 라이브러리 카탈로그를 조정하여 손상된 테이프에 사용 중인 플래그를 설정합니다.

이 명령을 다음 양식으로 사용합니다.

```
chmed +o eq:slot
```

표 2-10 chmed(1M)에 대한 인수

인수	의미
eq	mcf 파일의 정의에 따라 어드레스되는 자동화 라이브러리 또는 드라이브의 장비 서수.
slot	라이브러리 카탈로그에서 인식된 라이브러리의 저장소 슬롯 수. 수동으로 로드된 드라이브에는 이 인수를 적용할 수 없습니다.

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

나와있는 카트리지를 나중에 다시 집어넣으려면 카트리지를 자동화 라이브러리로 가져와야 합니다.

카탈로그 작업, 카트리지 가져오기 및 카트리지 내 보내기

자동화 라이브러리로부터 물리적 추가(가져오기) 및 제거(내보내기)를 하면 다음을 포함한 여러 기능을 수행할 수 있습니다.

- 카트리지를 교체할 수 있습니다.
- 재난 복구를 목적으로 나중에 사용하도록 오프 사이트 저장소에 카트리지를 재배치할 수 있습니다. 이 작업 중인 경우, chmed(1M) 명령의 -I 옵션을 사용하여 카트리지 저장 위치와 같은 추가 정보를 지정할 수 있습니다.

카트리지를 가져오고 내보낼 때 라이브러리 카트리지도 업데이트합니다. Sun StorEdge SAM-FS 시스템 내에서 import(1M) 및 samexport(1M) 명령을 사용하여 이 작업을 완성합니다. 또한 SAM-QFS Manager를 사용하여 이 작업을 수행할 수도 있습니다.

라이브러리 카탈로그는 Sun StorEdge SAM-FS 환경이 자동화 라이브러리에서 카트리지를 찾기 위해 필요로 하는 모든 정보의 중앙 리포지토리입니다. 라이브러리 카탈로그 파일은 자동화 라이브러리의 각 슬롯 관련 정보가 들어있는 이진 UFS 상주 파일입니다. 이 파일에 들어있는 정보에는 슬롯에 저장된 카트리지와 연결된 하나 이상의 볼륨 시리얼 이름(VSN), 해당 카트리지에 남아있는 용량과 공간 및 읽기 전용, 쓰기 보호, 리사이클 및 기타 카트리지에 대한 상태 정보를 나타내는 플래그 등이 포함됩니다.

Sun StorEdge SAM-FS 환경은 다음과 같이 자동화 라이브러리가 서버에 연결되는 방법에 따라 카탈로그를 다르게 취급합니다.

- 자동화 라이브러리가 직접 연결된 경우, 라이브러리 카탈로그는 라이브러리 카탈로그 항목과 자동화 라이브러리의 물리적 슬롯 사이에 일대일 매핑되어 있습니다. 라이브러리 카탈로그의 첫 번째 항목은 자동화 라이브러리의 첫 번째 슬롯에 해당합니다. 카탈로그가 필요하다면 시스템에서 라이브러리 카탈로그를 참조해 VSN이 포함된 슬롯을 찾고 명령을 실행해 해당 슬롯에서 드라이브로 카트리지를 로드합니다.
- 자동화 라이브러리가 네트워크 연결된 경우, 라이브러리 카탈로그는 슬롯에 직접 매핑되어 있지 않습니다. 이 경우 라이브러리 카탈로그는 자동화 라이브러리에 있는 VSN 목록입니다. 카트리지가 요청되면 시스템에서 업체의 소프트웨어에 요청하여 VSN을 드라이브에 로드합니다. 업체의 소프트웨어가 VSN의 저장소 슬롯을 찾습니다.

자동화 라이브러리 각각은 시스템 특성과 업체 제공 소프트웨어 때문에 카트리지를 가져오기/내보내기를 다른 방식으로 처리합니다. 예를 들어 ACL 4/52 라이브러리에서는 move 명령을 실행해 카트리지를 가져오기/내보내기 장치로 이동한 다음, 자동화 라이브러리에서 카트리지를 내보내야 합니다.

참고 - 네트워크 연결 자동화 라이브러리가 고유 유틸리티를 사용하여 카트리지를 가져오고 내보내서 import(1M) 및 samexport(1M) 명령이 Sun StorEdge SAM-FS 시스템에서 사용된 라이브러리 카탈로그 항목만 업데이트합니다. 네트워크 연결 라이브러리의 카트리지를 내보내기/가져오기에 관한 내용에 대해서는 37 페이지의 "작동 절차가 업체별로 고유한 라이브러리의 기본 작업"을 참조하십시오.

내보낸 매체 추적 — 기록자

Sun StorEdge SAM-FS 기록자는 자동화 라이브러리 또는 수동으로 마운트된 장치로부터 내보내진 카트리지를 추적합니다. 기록자는 가상 라이브러리와 같은 기능을 하나 정의된 하드웨어 장치는 없습니다. 기록자는 자동화 라이브러리와 mcf 파일에 구성되며, 자신과 연결된 모든 카트리지에 대한 항목을 기록하는 카탈로그가 있으며 카트리지를 가져오고 내보낼 수 있고, SAM-QFS Manager에는 또 다른 자동화 라이브러리로 나타납니다.

장치 유형 hy를 사용해 mcf 파일의 기록자를 구성될 수 있습니다. mcf 파일의 기록자를 구성하지 않는 경우 다음과 같이 작성됩니다.

```
historian n+l hy - on /var/opt/SUNWsamfs/catalog/historian
```

이러한 항목에서 $n+1$ 은 mcf 파일의 최신 장비 서수 더하기 1입니다. 카탈로그에 대해 다른 장비 서수 또는 경로 이름을 사용하려는 경우 mcf의 기록자만 정의하면 됩니다.

기록자 라이브러리 카탈로그는 기록자를 처음 시작할 때 32개 항목으로 초기화됩니다. 카탈로그가 상주하는 파일 시스템의 크기가 전체 카탈로그를 보유할 수 있을 만큼 커야 합니다. 사이트가 라이브러리에서 내보내진 기존 Sun StorEdge SAM-FS 카트리지를 추적할 수도 있습니다. 이러한 경우에는 build_cat(1M) 매뉴얼 페이지에서 설명한 것처럼 기존 카트리지를 기록자 카탈로그를 구축해야 합니다.

defaults.conf 파일의 다음 2가지 구성 명령이 기록자 행위에 영향을 줍니다.

- `exported_media = unavailable` 명령이 나타나면 자동화 라이브러리에서 내보낸 카트리는 기록자에게 사용할 수 없는 것으로 플래그됩니다. 사용할 수 없는 것으로 플래그된 카트리를 요청하면 EIO 오류가 발생합니다.
- `attended = no` 명령이 설정되면 기록자에게 로드 요청을 처리할 수 있는 운영자가 없다고 선언합니다. 기록자에게 알려졌으나 이미 로드되지 않은 카트리를 로드하라고 요청하면 EIO 오류가 발생합니다.

보다 자세한 구성 정보는 historian(7) 및 defaults.conf(4) 매뉴얼 페이지를 참조하십시오.

자동화 라이브러리로부터 가져오기 또는 내보내기

mailbox(편지함) 자동화 라이브러리에서 카트리를 추가 또는 제거하기 위한 자동화 라이브러리 안의 영역을 말합니다. `import(1M)` 명령을 사용하면 카트리를 편지함에서 저장소 슬롯으로 이동할 수 있습니다. `samexport(1M)` 명령을 사용하면 카트리를 저장소 슬롯에서 편지함으로 이동할 수 있습니다. 대부분의 라이브러리에서는 Sun StorEdge SAM-FS 소프트웨어가 시작될 때 편지함에 카트리가 있으면 소프트웨어 시작과 동시에 카트리를 자동으로 가져옵니다.

가져오기 및 내보내기 방법은 제조업체마다 다릅니다. 이러한 작업에 문제가 있는 경우에는 37 페이지의 "작동 절차가 업체별로 고유한 라이브러리의 기본 작업"을 참조하여 자신의 장비에 대해 권장된 특수 절차가 있는지를 알아보십시오.

다음 절은 카트리지를 가져오기 및 내보내기를 설명합니다.

- 32 페이지의 "편지함을 사용하여 라이브러리로 카트리를 가져오기"
- 32 페이지의 "편지함을 사용하여 라이브러리로부터 카트리를 내보내기"
- 33 페이지의 "편지함을 사용하지 않고 라이브러리로 카트리를 가져오기"
- 34 페이지의 "편지함을 사용하지 않고 라이브러리로부터 카트리를 내보내기"

▼ 편지함을 사용하여 라이브러리로 카트리지를 가져오기

편지함을 사용해 자동화 라이브러리로 카트리지를 가져오려면 다음과 같은 작업을 수행해야 합니다.

1. 제조업체의 권장 작업 방법에 따라 편지함을 엽니다.

대개는 편지함 주변에 단추가 있습니다. 경우에 따라서 편지함은 업체의 설명서에서 *mail slot*(메일 슬롯)이라고 하는 단일 슬롯 편지함을 의미합니다.

2. 편지함에 카트리지를 수동으로 설치합니다.

3. 편지함을 닫습니다.

4. `import(1M)` 명령을 사용해 카트리지를 가져옵니다.

이 명령을 다음 양식으로 사용합니다.

```
import eq
```

*eq*에 대해, *mcf* 파일의 정의에 따라 어드레스되는 라이브러리의 장비 서수를 지정합니다.

시스템은 편지함에서 저장소 슬롯으로 카트리지를 이동하고 각 카트리지에 대해 라이브러리 카탈로그를 업데이트합니다.

또한 `samu(1M)` 또는 `SAM-QFS Manager`를 사용하여 이 작업을 수행할 수 있습니다.

▼ 편지함을 사용하여 라이브러리로부터 카트리지 내보내기

이 절차에서는 저장소 슬롯에서 편지함이나 메일 슬롯으로 카트리지를 이동합니다. 편지함을 사용하는 라이브러리에서 카트리지를 내보내기(배출)하려면 다음과 같은 작업을 수행합니다.

1. **samexport(1M) 명령을 사용하여 카트리지를 저장소 슬롯에서 편지함으로 이동할 수 있습니다.**

이 명령을 다음 양식 중 하나로 사용합니다.

```
samexport eq:slot
samexport media_type.vsn
```

표 2-11 samexport(1M)에 대한 인수

인수	의미
<i>eq</i>	mcf 파일의 정의에 따라 어드레스되는 자동화 라이브러리의 장비 서수.
<i>slot</i>	라이브러리 카탈로그에서 인식된 자동화 라이브러리의 저장소 슬롯 수.
<i>media_type</i>	매체 유형. 올바른 매체 유형의 목록에 대해서는 mcf(4) 매뉴얼 페이지를 참조하십시오.
<i>vsn</i>	볼륨에 할당된 볼륨 시리얼 이름.

또한 samu(1M) 또는 SAM-QFS Manager를 사용하여 이 단계를 수행할 수 있습니다.

2. **제조업체의 권장 방법에 따라 편지함 또는 메일 슬롯을 엽니다.**

대개는 편지함 주변에 단추가 있습니다.

▼ 편지함을 사용하지 않고 라이브러리로 카트리지를 가져오기

1. **samcmd(1M) unload 명령을 사용합니다.**

이 명령을 다음 양식으로 사용합니다.

```
samcmd unload eq
```

*eq*에 대해, mcf 파일의 정의에 따라 어드레스되는 라이브러리의 장비 서수를 지정합니다.

시스템에서 현재 작업을 완료하고 off 상태로 설정한 다음, 현재 활성화된 카탈로그를 기록자에게 전송할 때까지 대기합니다.

2. **자동화 라이브러리 문의 잠금을 해제하고 엽니다.**
3. **사용 가능한 슬롯에 카트리지를 로드하십시오.**
4. **자동화 라이브러리 문을 닫고 잠급니다.**

자동화 라이브러리는 카트리지를 재초기화 및 스캔합니다. Sun StorEdge SAM-FS 소프트웨어는 가져온 카트리지의 VSN을 카탈로그에 추가하여 라이브러리 카탈로그를 업데이트합니다. 자동화 라이브러리 상태는 on으로 설정됩니다.

▼ 편지함을 사용하지 않고 라이브러리로부터 카트리지를 내보내기

1. samcmd(1M) unload 명령을 사용합니다.

이 명령을 다음 양식으로 사용합니다.

```
samcmd unload eq
```

*eq*에 대해, *mcf* 파일의 정의에 따라 어드레스되는 라이브러리의 장비 서수를 지정합니다.

시스템에서 현재 작업을 완료하고 off 상태로 설정한 다음, 현재 활성화된 카탈로그를 기록자에게 전송할 때까지 대기합니다.

2. 자동화 라이브러리 문의 잠금을 해제하고 엽니다.

3. 각각의 슬롯에서 카트리지를 제거합니다.

4. 자동화 라이브러리 문을 닫고 잠급니다.

자동화 라이브러리는 카트리지를 재초기화 및 스캔합니다. 시스템은 현재 라이브러리 슬롯에 있는 카트리지의 VSN을 사용해 라이브러리 카탈로그를 업데이트합니다. 제거된 카트리지의 VSN은 라이브러리 카탈로그에서 제거되어 기록자 파일에만 기록됩니다. 자동화 라이브러리 상태는 on으로 설정됩니다.

▼ 로드 통지 사용하기

Sun StorEdge SAM-FS 소프트웨어는 카트리지가 아카이빙 및 스테이징 조건을 충족시키도록 정기적인 로드를 요청합니다. 요청 대상이 라이브러리 내부에 상주하는 카트리지가 아닌 경우 자동으로 요청을 처리합니다. 요청 대상이 라이브러리 외부에 상주하는 카트리지가 아닌 경우 운영자가 조치해야 합니다. 실행시 라이브러리 외부에서 카트리지를 가져와야 하는 경우에는 *load_notify.sh*(1M) 스크립트가 email을 보냅니다.

1. 슈퍼유저가 됩니다.

2. cp(1) 명령을 사용하여 설치된 위치에서 작동 가능한 위치로 로드 통지 스크립트를 복사합니다.

예를 들어, 다음과 같습니다.

```
# cp /opt/SUNWsamfs/examples/load_notify.sh  
/etc/opt/SUNWsamfs/scripts/load_notify.sh
```

3. **more(1) 또는 또 다른 명령을 사용하여 defaults.conf 파일을 관찰합니다.**

다음 명령이 파일에 있고 다음과 같이 나타나는지 확인하십시오.

■ `exported_media=available`

■ `attended=yes`

이러한 명령은 기본으로 설정되어 있습니다. 로드 통지 기능을 사용하려면 이러한 명령이 변경되지 않았는지 확인해야 합니다.

4. **load_notify.sh 스크립트를 수정하여 운영자에게 통지를 보냅니다.**

기본 설정에 따라 스크립트에서 root에 전자우편을 보내지만, 다른 사람에게 전자우편을 보내거나 호출기에 연락하거나 아니면 다른 통지 방법을 제공하기 위해 스크립트를 편집할 수 있습니다.

수동으로 로드된 드라이브 작업

이 섹션에서는 자동화 라이브러리가 아니라 수동으로 로드한 독립형 드라이브인 경우에 달라지는 작업에 대해 설명합니다. 수동으로 로드한 드라이브에는 단일 슬롯 라이브러리 카탈로그가 각각 있습니다.

▼ 카트리지를 로드하려면

- 수동 로드 장치에 카트리지를 로드하려면 제조업체의 지시사항에 따라 드라이브에 카트리지를 설치해야 합니다.

Sun StorEdge SAM-FS 시스템은 카트리가 로드되었는지 인식하고 레이블을 읽으며 설명서, 단일 슬롯 카탈로그를 업데이트합니다. 더 이상의 작업은 필요하지 않습니다.

▼ 카트리지를 언로드 하려면

- **samcmd(1M) idle 명령을 사용하여 드라이브를 유틸리티 상태로 둡니다.**

이 명령을 사용하면 아카이브나 스테이지 프로세스가 활성화되지 않습니다. 이 명령을 다음 양식으로 사용합니다.

```
# samcmd idle eq
```

*eq*에 대해, *mcf* 파일의 정의에 따라 어드레스되는 드라이브의 장비 서수를 지정합니다. 모든 입출력 작업이 끝나면 드라이브가 *idle*에서 *off*로 전환되고 테이프가 배출됩니다.

테이프인 경우에는 되감기되어 카트리는 제거 준비 상태가 됩니다. 광 카트리는 자동으로 배출됩니다. 특정 카트리지 제거 방법에 대해서는 제조업체의 설명서를 참조하십시오.

또한 *samu(1M)* 또는 *SAM-QFS Manager*를 사용하여 이 작업을 수행할 수 있습니다.

▼ 라이브러리 카탈로그를 보려면

- **samu(1M) 유틸리티의 :v 명령을 사용합니다.**

이 명령을 다음 양식으로 사용합니다.

```
:v eq
```

*eq*에 대해, *mcf* 파일의 정의에 따라 어드레스되는 라이브러리의 장비 서수를 지정합니다.

작동 절차가 업체별로 고유한 라이브러리의 기본 작업

Sun StorEdge SAM-FS 환경에 많은 다른 제조업체의 라이브러리를 포함할 수 있습니다. 대부분의 라이브러리에 대해 13 페이지의 "Sun StorEdge SAM-FS 환경에서 자동화 라이브러리 및 수동으로 로드된 드라이브 사용하기"에 설명된 작동 절차를 사용해야 합니다. 그러나 일부 라이브러리는 작동 절차가 업체별로 다릅니다. 이 장에서는 이러한 라이브러리에 대해 설명합니다.

참고 – Sun StorEdge SAM-FS 소프트웨어는 많은 제조업체의 자동화 라이브러리와 호환됩니다. 라이브러리 모델 번호, 펌웨어 레벨 및 기타 호환성 정보에 관련된 정보는 Sun 판매 담당자 또는 인증된 서비스 제공업체에 문의하십시오.

이 장에서는 다음 자동화 라이브러리에 대해 설명합니다.

- 38 페이지의 "ADIC/Grau 자동화 라이브러리"
- 39 페이지의 "Fujitsu LMF 자동화 라이브러리"
- 41 페이지의 "IBM 3584 UltraScalable 테이프 라이브러리"
- 43 페이지의 "IBM 3494 라이브러리"
- 44 페이지의 "Sony 직접 연결 8400 PetaSite 자동화 라이브러리"
- 47 페이지의 "Sony 네트워크 연결 자동화 라이브러리"
- 49 페이지의 "StorageTek ACSLS 연결 자동화 라이브러리"

ADIC/Grau 자동화 라이브러리

ADIC/Grau 자동화 라이브러리가 있는 경우 카트리지를 내보내기 및 가져오기에 대해 이 절의 절차를 사용합니다. 이 절차는 13 페이지의 "Sun StorEdge SAM-FS 환경에서 자동화 라이브러리 및 수동으로 로드된 드라이브 사용하기"에 설명된 절차와 다릅니다.

ADIC/Grau 자동화 라이브러리의 카트리지를 물리적으로 추가 및 제거하기 위해 업체 제공 유틸리티를 사용하기 때문에 Sun StorEdge SAM-FS 인터페이스(import(1M), samexport(1M) 및 SAM-QFS Manager)는 라이브러리 카탈로그에만 영향을 줍니다.

▼ 카트리지를 가져오려면

카트리지를 가져오려면 다음 작업을 수행해야 합니다.

1. **ADIC/Grau 명령을 사용해 카트리지를 라이브러리에 물리적으로 가져옵니다.**
2. **Sun StorEdge SAM-FS import(1M) 명령을 사용하여 라이브러리 카탈로그를 업데이트합니다.**

이 명령을 다음 형식으로 사용합니다.

```
import -v volser eq
```

표 3-1 import(1M) 명령에 대한 인수

인수	의미
<i>volser</i>	추가할 <i>volser.grauaci</i> 인터페이스는 ADIC/Grau 자동화 라이브러리에 <i>volser</i> 정보가 있는지 확인한 후에 새 항목으로 라이브러리 카탈로그를 업데이트합니다.
<i>eq</i>	mcf 파일의 정의에 따라 어드레스되는 장치의 장비 서수.

▼ 카트리지를 내보내려면

카트리지를 내보내려면 다음 작업을 수행해야 합니다.

1. Sun StorEdge SAM-FS `samexport(1M)` 명령을 사용하여 라이브러리 카탈로그에서 항목을 제거합니다.

이 명령을 다음 형식 중 하나로 사용합니다.

```
samexport eq:slot  
samexport media_type.vsn
```

표 3-2 samexport(1M) 명령에 대한 인수

인수	의미
<code>eq</code>	mcf 파일의 정의에 따라 어드레스되는 장치의 장비 서수.
<code>slot</code>	라이브러리 카탈로그에서 인식된 자동화 라이브러리의 저장소 슬롯 수.
<code>media_type</code>	매체 유형. 올바른 매체 유형의 목록에 대해서는 mcf(4) 매뉴얼 페이지를 참조하십시오.
<code>vsn</code>	볼륨에 할당된 볼륨 시리얼 이름.

`samexport(1M)` 명령은 각 VSN을 내보낼 때마다 라이브러리 카탈로그를 업데이트하고, 각 VSN에 대한 라이브러리 카탈로그 항목을 라이브러리 카탈로그에서 기록자로 이동합니다.

2. ADIC/Grau 명령을 사용해 카트리지를 라이브러리에서 물리적으로 내보냅니다.

Fujitsu LMF 자동화 라이브러리

AFujitsu LMF 자동화 라이브러리가 있는 경우 카트리지 내보내기 및 가져오기에 대해 이 절의 절차를 사용합니다. 이 절차는 13 페이지의 "Sun StorEdge SAM-FS 환경에서 자동화 라이브러리 및 수동으로 로드된 드라이브 사용하기"에 설명된 절차와 다릅니다.

Fujitsu LMF 자동화 라이브러리의 카트리지를 물리적으로 추가 또는 제거하기 위해 업체 제공 유틸리티를 사용하기 때문에 Sun StorEdge SAM-FS 인터페이스(`import(1M)`, `samexport(1M)` 및 SAM-QFS Manager)는 라이브러리 카탈로그에만 영향을 줍니다.

▼ 카트리지를 가져오려면

카트리지를 가져오려면 다음 작업을 수행해야 합니다.

1. Fujitsu 명령을 사용해 카트리지를 라이브러리에 물리적으로 가져옵니다.
2. Sun StorEdge SAM-FS `import(1M)` 명령을 사용하여 라이브러리 카탈로그를 업데이트합니다.

이 명령을 다음 형식으로 사용합니다.

```
import -v volser eq
```

표 3-3 import(1M) 명령에 대한 인수

인수	의미
<i>volser</i>	추가할 <i>volser.fujitsulmf</i> 인터페이스는 LMF 자동화 라이브러리에 <i>volser</i> 정보가 있는지 확인한 후에 새 항목으로 라이브러리 카탈로그를 업데이트합니다.
<i>eq</i>	mcf 파일의 정의에 따라 어드레스되는 장치의 장비 서수.

▼ 카트리지를 내보내려면

카트리지를 내보내려면 다음 작업을 수행해야 합니다.

1. Sun StorEdge SAM-FS `samexport(1M)` 명령을 사용하여 라이브러리 카탈로그에서 항목을 제거합니다.

이 명령을 다음 형식 중 하나로 사용합니다.

```
samexport eq:slot
samexport media_type.vsn
```

표 3-4 samexport(1M) 명령에 대한 인수

인수	의미
<i>eq</i>	mcf 파일의 정의에 따라 어드레스되는 장치의 장비 서수.
<i>slot</i>	라이브러리 카탈로그에서 인식된 자동화 라이브러리의 저장소 슬롯 수.
<i>media_type</i>	매체 유형. 올바른 매체 유형의 목록에 대해서는 mcf(4) 매뉴얼 페이지를 참조하십시오.
<i>vsn</i>	볼륨에 할당된 볼륨 시리얼 이름.

`samexport(1M)` 명령은 각 VSN을 내보낼 때마다 라이브러리 카탈로그를 업데이트하고, 각 VSN에 대한 라이브러리 카탈로그 항목을 Sun StorEdge SAM-FS 라이브러리 카탈로그에서 Sun StorEdge SAM-FS 기록자로 이동합니다.

2. Fujitsu 명령을 사용해 카트리지를 라이브러리에서 물리적으로 내보냅니다.

IBM 3584 UltraScalable 테이프 라이브러리

IBM 3584 UltraScalable 테이프 라이브러리는 Sun StorEdge SAM-FS 환경에서 지원됩니다. 다음 절은 13 페이지의 "Sun StorEdge SAM-FS 환경에서 자동화 라이브러리 및 수동으로 로드된 드라이브 사용하기"에 설명된 절차와 다른 이 라이브러리 작동의 여러 가지 측면을 설명합니다.

카트리지 가져오기

Sun StorEdge SAM-FS 소프트웨어가 시작할 때 편지함에 있는 카트리지를 자동으로 가져오지 않습니다.

드라이브 클리닝

Sun StorEdge SAM-FS 환경에서 이 라이브러리를 사용하려면 자동 클리닝을 사용 해제하고 호스트된 클리닝을 사용합니다. 이 프로세스는 *IBM 3584 UltraScalable Tape Library Planning and Operator Guide*, IBM publication GA32-0408-01에 설명되어 있습니다. 또한 *ibm3584(7)* 매뉴얼 페이지에도 설명되어 있습니다.

파티션

이 라이브러리에는 몇 개의 테이프 드라이브가 포함됩니다. 여러 드라이브를 사용하는 경우에는 1개의 물리적 라이브러리를 2, 3 또는 4개의 논리 라이브러리로 나눌 수 있습니다. 라이브러리를 2개 이상의 논리 라이브러리로 나눈 경우에는 이러한 논리 라이브러리가 올바르게 작동하는지 확인한 후에 IBM 3584 라이브러리를 Sun StorEdge SAM-FS 환경에 추가해야 합니다.

파티션을 나눈 라이브러리에서 카트리지를 내보낼 때는 카트리지를 내보낸 논리 라이브러리만이 해당 드로어 슬롯에 액세스할 수 있습니다. 카트리지를 수동으로 제거 및 재삽입하는 경우에는 모든 논리 파티션에 액세스할 수 있습니다.

▼ 카트리지를 언로드 하려면

다음 작업에서는 이러한 상황에서 수행하는 제거 작업에 대해 설명합니다.

1. 문을 엽니다.
2. 카트리지를 제거합니다.
3. 문을 닫습니다.
4. 문이 잠긴 다음, 잠금이 해제될 때까지 대기합니다.
5. 문을 엽니다.
6. 카트리지를 교체합니다.
7. 문을 닫습니다.

Sun StorEdge SAM-FS 환경의 지역으로 파티션된 라이브러리로 이 라이브러리를 사용하는 것에 대한 자세한 정보는 IBM 설명서 또는 *ibm3584(7)* 매뉴얼 페이지를 참조하십시오.

IBM 3494 라이브러리

IBM 3494 라이브러리는 Sun StorEdge SAM-FS 환경에서 지원됩니다. 다음 절은 13 페이지의 "Sun StorEdge SAM-FS 환경에서 자동화 라이브러리 및 수동으로 로드된 드라이브 사용하기"에 설명된 절차와 다른 이 라이브러리 작동의 여러 가지 측면을 설명합니다.

▼ 카트리지를 가져오려면

카트리지를 가져오려면 다음 작업을 수행해야 합니다.

1. 새 매체를 I/O 슬롯에 넣습니다.

2. 문을 닫습니다.

라이브러리가 문을 잠그고 매체를 저장 영역으로 이동합니다. 한 번에 볼륨 100개만 가져올 수 있습니다.

라이브러리가 `access=private`으로 구성된 경우, 이것이 수행해야 할 마지막 단계입니다. 라이브러리는 매체가 이동할 때 데몬을 통지하고 매체는 카탈로그에 추가됩니다.

3. `import(1M)` 명령을 사용하여 매체를 카탈로그에 추가합니다. (선택 사항)

라이브러리가 `access=shared`로 구성된 경우에만 이 단계를 수행하십시오.

라이브러리가 `access=shared`로 구성된 경우, `import(1M)` 명령을 사용하여 매체를 카탈로그에 추가합니다.

▼ 카트리지를 내보내려면

1. `export(1M)` 명령을 사용하여 카트리지를 내보냅니다.

이 명령은 매체를 I/O 영역으로 이동하고 운영자 패널의 출력 모드 표시등을 켭니다.

2. I/O 영역에서 물리적으로 매체를 제거합니다.

Sony 직접 연결 8400 PetaSite 자동화 라이브러리

Sony 8400 PetaSite 시리즈 자동화 라이브러리는 가져오기 및 내보내기 편지함의 슬롯이 8개(슬롯 400-407)라는 점에서 다른 Sony 모델과 다릅니다. 이러한 이유 때문에 이 시스템에서 가져오기 및 내보내기 작업이 보다 간단합니다. 이 자동화 라이브러리는 바코드 판독기를 사용합니다.

편지함 슬롯이 저장 슬롯으로 사용될 수 있으므로 Sun StorEdge SAM-FS 라이브러리 카탈로그가 편지함 슬롯을 추적합니다.

참고 - 이 섹션의 내용은 Sony 직접 연결 8400 PetaSite 자동화 라이브러리에만 적용됩니다. 이 정보는 Sony 직접 연결 B9 및 B35 자동화 라이브러리와 관련이 없으며 47 페이지의 "Sony 네트워크 연결 자동화 라이브러리"에도 관련이 없습니다.

▼ 테이프를 가져오려면

테이프를 가져오려면 다음 작업을 수행해야 합니다.

1. 자동화 라이브러리의 전면 패널에 있는 열기/닫기 단추를 눌러 자동화 라이브러리의 문을 엽니다.
2. 편지함 슬롯에 카트리지를 로드합니다.
3. 자동화 라이브러리의 전면 패널에 있는 열기/닫기 단추를 누르고 편지함의 문을 수동으로 닫습니다.

문이 닫히면 자동화 라이브러리가 편지함 슬롯에서 카트리지 바코드를 확인합니다. 바코드에 문제가 있으면 해당 슬롯의 in 및 out 표시등이 모두 점멸합니다.

4. `import(1M)` 명령을 사용하여 가져온 카트리지를 인식하도록 Sun StorEdge SAM-FS 시스템을 사용합니다.

이 명령을 다음 형식으로 사용합니다.

```
import eq
```

`eq`에 대해, `mcf` 파일의 정의에 따라 어드레스되는 장치의 장비 서수를 지정합니다. 또한 SAM-QFS Manager를 사용하여 이 단계를 수행할 수도 있습니다.

테이프 내보내기

테이프 카트리지를 내보내는 절차는 편지함 슬롯을 저장소 슬롯으로 사용하고 있는지의 여부에 따라 달라집니다.

▼ 편지함 슬롯을 저장소 슬롯으로 사용하지 않고 테이프를 내보내려면

편지함 슬롯을 저장소 슬롯으로 사용하지 않는 경우에 카트리지를 내보내려면 다음 절차를 따라야 합니다.

1. `move(1M)` 명령을 실행해 카트리지를 편지함 슬롯(슬롯 400-407)으로 이동합니다. 이 명령을 다음 형식으로 사용합니다.

```
move source_slot destination_slot eq
```

표 3-5 move(1M) 명령에 대한 인수

인수	의미
<code>source_slot</code>	현재 카트리가 상주하는 슬롯의 번호.
<code>destination_slot</code>	카트리가 이동해야 하는 슬롯의 번호.
<code>eq</code>	<code>mcf</code> 파일의 정의에 따라 어드레스되는 장치의 장비 서수.

2. 자동화 라이브러리의 전면 패널에 있는 열기/닫기 단추를 누릅니다. 문이 열립니다.
3. 편지함 슬롯에서 카트리지를 제거합니다.

4. 자동화 라이브러리의 전면 패널에 있는 열기/닫기 단추를 누르고 편지함의 문을 수동으로 닫습니다.
5. **samexport(1M)** 명령을 실행하여 내보낸 카트리지를 인식하도록 Sun StorEdge SAM-FS 시스템을 사용합니다.

이 명령을 다음 형식으로 사용합니다.

```
samexport eq
```

*eq*에 대해, *mcf* 파일의 정의에 따라 어드레스되는 장치의 장비 서수를 지정합니다. 또한 SAM-QFS Manager를 사용하여 이 단계를 수행할 수도 있습니다.

▼ 편지함 슬롯을 저장소 슬롯으로 사용해 테이프를 내보내려면

편지함 슬롯을 저장소 슬롯으로 사용하고 있으며 내보낸 카트리가 편지함 슬롯 중 하나에 있는 경우에는 다음 절차에 따라 카트리를 내보냅니다.

1. 자동화 라이브러리의 전면 패널에 있는 열기/닫기 단추를 누릅니다.
문이 열립니다.
2. 편지함 슬롯에서 카트리를 제거합니다.
3. 자동화 라이브러리의 전면 패널에 있는 열기/닫기 단추를 누르고 편지함의 문을 수동으로 닫습니다.
4. **samexport(1M)** 명령을 실행하여 내보낸 카트리를 인식하도록 Sun StorEdge SAM-FS 시스템을 사용합니다.
이 명령을 다음 형식으로 사용합니다.

```
samexport eq
```

*eq*에 대해, *mcf* 파일의 정의에 따라 어드레스되는 장치의 장비 서수를 지정합니다. 또한 SAM-QFS Manager를 사용하여 이 단계를 수행할 수도 있습니다.

▼ 카트리를 다른 슬롯으로 이동하려면

카트리를 다른 슬롯으로 이동하려면 다음 작업을 수행해야 합니다.

1. 소스 슬롯이 사용 중이고 대상 슬롯이 비어있는지 확인합니다.

2. move(1M) 명령을 실행합니다.

이 명령을 다음 형식으로 사용합니다.

```
move eq:source_slot destination_slot
```

표 3-6 move(1M) 명령에 대한 인수

인수	의미
<i>eq</i>	mcf 파일의 정의에 따라 어드레스되는 장치의 장비 서수.
<i>source_slot</i>	현재 카트리지가 상주하는 슬롯의 번호.
<i>destination_slot</i>	카트리지가 이동해야 하는 슬롯의 번호.

또한 SAM-QFS Manager를 사용하여 이 단계를 수행할 수도 있습니다.

Sony 네트워크 연결 자동화 라이브러리

Sony 네트워크 연결 자동화 라이브러리가 있는 경우 카트리지 내보내기 및 가져오기에 대해 이 절의 절차를 사용합니다. 이 절차는 13 페이지의 "Sun StorEdge SAM-FS 환경에서 자동화 라이브러리 및 수동으로 로드된 드라이브 사용하기"에 설명된 절차와 다릅니다.

Sony 자동화 라이브러리의 카트리지를 물리적으로 추가 및 제거하기 위해 업체 제공 유틸리티를 사용하기 때문에 Sun StorEdge SAM-FS 인터페이스(import(1M), sameport(1M) 및 SAM-QFS Manager)는 라이브러리 카탈로그에만 영향을 줍니다.

▼ 카트리지를 가져오려면

카트리지를 가져오려면 다음 작업을 수행해야 합니다.

1. **Sony 명령을 사용해 카트리지를 라이브러리에 물리적으로 가져옵니다.**
2. **import(1M) 명령을 사용하여 라이브러리 카탈로그를 업데이트합니다.**
이 명령을 다음 형식으로 사용합니다.

```
import -v [ " ] volser [ " ] eq
```

표 3-7 import(1M) 명령에 대한 인수

인수	의미
" "	인용 부호. <i>volser</i> 는 공백이 포함될 경우 인용 부호로 묶어야 합니다.
<i>volser</i>	추가할 <i>volser</i> . PSC API 인터페이스는 Sony 자동화 라이브러리에 <i>volser</i> 정보가 있는지 확인한 후에 새 항목으로 라이브러리 카탈로그를 업데이트합니다. 카트리지가 라이브러리에 물리적으로 없으면 기록자 카탈로그에 항목을 둡니다.
<i>eq</i>	mcf 파일의 정의에 따라 어드레스되는 라이브러리의 장비 서수.

▼ 카트리지를 내보내려면

카트리지를 내보내려면 다음 작업을 수행해야 합니다.

1. samexport(1M) 명령을 사용하여 라이브러리 카탈로그에서 항목을 제거합니다.

이 명령을 다음 형식 중 하나로 사용합니다.

```
samexport eq:slot
samexport media_type.vsn
```

표 3-8 samexport(1M) 명령에 대한 인수

인수	의미
<i>eq</i>	mcf 파일의 정의에 따라 어드레스되는 장치의 장비 서수.
<i>slot</i>	라이브러리 카탈로그에서 인식된 자동화 라이브러리의 저장소 슬롯 수.
<i>media_type</i>	매체 유형. 올바른 매체 유형의 목록에 대해서는 mcf(4) 매뉴얼 페이지를 참조하십시오.
<i>vsn</i>	볼륨에 할당된 볼륨 시리얼 이름.

samexport(1M) 명령은 각 VSN을 내보낼 때마다 라이브러리 카탈로그를 업데이트하고, 각 VSN에 대한 라이브러리 카탈로그 항목을 라이브러리 카탈로그에서 기록자로 이동합니다.

2. Sony 명령을 사용해 라이브러리에서 카트리지를 물리적으로 내보냅니다.

StorageTek ACSLS 연결 자동화 라이브러리

StorageTek ACSLS 연결 자동화 라이브러리가 있는 경우 카트리지를 내보내기 및 가져오기에 대해 이 절의 절차를 사용합니다. 이 절차는 13 페이지의 "Sun StorEdge SAM-FS 환경에서 자동화 라이브러리 및 수동으로 로드된 드라이브 사용하기"에 설명된 절차와 다릅니다.

*mailbox*는 자동화 라이브러리에 카트리지를 가져오기 및 내보내기 하는 데 사용하는 영역입니다. 일부 StorageTek 자동화 라이브러리는 한 번에 1개의 카트리지만을 가져오기 및 내보내기 합니다. Sun StorEdge SAM-FS 환경 내에서 지원되는 편지함이 있는 StorageTek 자동화 라이브러리의 예제에는 StorageTek 9714 및 StorageTek 9710이 포함됩니다. StorageTek 9730은 편지함 슬롯을 사용합니다. StorageTek 설명서에서는 편지함과 편지함 슬롯을 *CAP*라고도 합니다.

ACSLS 연결 자동화 라이브러리로부터 카트리지를 가져오고 내보낼 경우, 다음을 기억하십시오.

- 카트리지를 가져올 때 Sun StorEdge SAM-FS 명령은 라이브러리 카탈로그에만 영향을 줍니다. `import(1M)` 명령은 자동화 라이브러리에 물리적으로 카트리지를 넣지 않습니다. ACSLS 명령을 사용해 카트리지를 물리적으로 가져와야 합니다.
- 카트리지를 내보낼 때 `samexport(1M)` 명령의 `-f` 옵션 또한 사용하지 않는 경우 Sun StorEdge SAM-FS 명령은 라이브러리 카탈로그에만 영향을 줍니다. `-f` 옵션을 사용하면 Sun StorEdge SAM-FS 시스템이 카트리지를 액세스 포트(CAP)에 볼륨을 두고 그에 따라 카탈로그를 업데이트하게 됩니다. `-f` 옵션을 지정하지 않은 경우, 카탈로그는 업데이트되지만 볼륨이 CAP에 없기 때문에 아직 ACSLS 명령을 사용하여 물리적으로 카트리지를 내보내야 합니다.

ACSLS 목록 및 Sun StorEdge SAM-FS 카탈로그를 일치시켜 유지하는 것이 사용자의 책임입니다.

또한 `samu(1M)` 또는 SAM-QFS Manager를 사용하여 가져오기 및 내보내기 절차를 수행할 수 있습니다.

▼ 테이프를 가져오려면

- 테이프 카트리지를 가져오려면 **import(1M)** 명령을 사용해야 합니다.
이 명령을 다음 형식으로 사용합니다.

```
import -v vsn eq
```

표 3-9 import(1M) 명령에 대한 인수

인수	의미
<i>vsn</i>	볼륨에 할당된 볼륨 시리얼 이름.
<i>eq</i>	mcf 파일의 정의에 따라 어드레스되는 장치의 장비 서수.

import(1M) 명령을 사용하면 라이브러리 카탈로그에 새 VSN이 나타납니다. VSN이 기록자에 있으면 Sun StorEdge SAM-FS 소프트웨어가 VSN 정보를 기록자에서 라이브러리 카탈로그로 이동합니다.

▼ 편지함을 사용해 테이프를 내보내려면

슬롯이나 VSN별로 테이프 카트리지를 내보낼 수 있습니다.

- 테이프 카트리지를 내보내려면 **samexport(1M)** 명령을 사용해야 합니다.
이 명령을 다음 형식 중 하나로 사용합니다.

```
samexport [-f] eq:slot
samexport [-f] media_type.vsn
```

표 3-10 samexport(1M) 명령에 대한 인수

인수	의미
-f	Sun StorEdge SAM-FS 시스템이 카트리지 액세스 포트(CAP)에 볼륨을 두고 그에 따라 카탈로그를 업데이트하게 됩니다.
eq	mcf 파일의 정의에 따라 어드레스되는 장치의 장비 서수.
slot	라이브러리 카탈로그에서 인식된 자동화 라이브러리의 저장소 슬롯 수.
media_type	매체 유형. 올바른 매체 유형의 목록에 대해서는 mcf(4) 매뉴얼 페이지를 참조하십시오.
vsn	볼륨에 할당된 볼륨 시리얼 이름.

samexport(1M) 명령은 각 VSN을 내보낼 때마다 라이브러리 카탈로그를 업데이트하고, 각 VSN에 대한 라이브러리 카탈로그 항목을 라이브러리 카탈로그에서 기록자로 이동합니다.

아카이브

*아카이브*란 Sun StorEdge SAM-FS 파일 시스템에서 제거 가능한 매체 카트리지가나 다른 파일 시스템의 디스크 파티션에 상주하는 볼륨으로 파일을 복사하는 프로세스를 말합니다. 이 장에서 사용하는 *아카이브 매체*라는 용어는 아카이브 볼륨이 쓰인 다양한 카트리지가나 디스크 슬라이스를 가리키는 말입니다. Sun StorEdge SAM-FS 아카이브 기능에는 파일이 즉시 아카이브되도록 지정하거나 파일이 아카이브되지 않도록 지정하거나 기타 작업을 수행할 때 사용할 수 있는 기능을 포함해 여러 가지 기능이 포함됩니다.

이 장에서는 아카이버의 작동 원리를 설명하고 사이트에 적합한 아카이브 정책 개발을 위한 일반 지침을 제공하며, `archiver.cmd` 파일을 만들어 정책을 실행하는 방법을 설명합니다.

다음 주제를 다룹니다.

- 53 페이지의 "아카이버 - 작동 원리"
- 65 페이지의 "archiver.cmd 파일"
- 70 페이지의 "archiver.cmd 명령"
- 109 페이지의 "디스크 아카이브"
- 116 페이지의 "아카이버 예제"
- 128 페이지의 "아카이버 지침"
- 129 페이지의 "아카이버 문제 해결"

아카이버 - 작동 원리

아카이버는 Sun StorEdge SAM-FS 파일을 아카이브 매체에 자동으로 기록합니다. 파일 아카이브 및 스테이지에 운영자 조치가 필요하지 않습니다. 파일은 아카이브 매체에 볼륨으로 아카이브되며, 각 볼륨은 *볼륨 시리얼 이름(VSN)*이라고 하는 고유 식별자에 의해 식별됩니다. 아카이브 매체에는 1개 이상의 볼륨이 포함될 수 있습니다. 개별 볼륨을 식별하려면 매체 유형과 VSN을 식별해야 합니다.

아카이버는 Sun StorEdge SAM-FS 파일 시스템이 마운트되면 자동으로 시작합니다. 다음 파일에 아카이브 명령을 삽입하면 아카이버 작동을 사이트에 적합하게 사용자 지정할 수 있습니다.

```
/etc/opt/SUNWsamfs/archiver.cmd
```

archiver.cmd 파일은 아카이브 실행을 위해 반드시 필요한 것은 아닙니다. 이 파일이 없으면 아카이버는 다음 기본값을 사용합니다.

- 모든 파일은 사용 가능한 볼륨으로 아카이브됩니다.
- 모든 파일의 *아카이브 나이*는 4분입니다. 아카이브 나이란 파일이 마지막으로 수정된 이후의 시간을 말합니다.
- *아카이브 간격*은 10분입니다. 아카이브 간격이란 완전한 아카이브 프로세스 간의 경과 시간을 말합니다.

다음 섹션에서는 아카이브 세트의 개념과 아카이브 프로세스 과정에서 수행하는 작업에 대해 설명합니다.

아카이브 세트

*아카이브 세트*는 아카이브할 파일의 그룹을 식별합니다. 아카이브 세트는 파일 시스템 그룹에 걸쳐 정의할 수 있습니다. 아카이브 세트의 파일은 크기, 소유권, 그룹 또는 디렉토리 위치에 관련된 공통된 기준을 공유합니다. 아카이브 세트는 아카이브 복사본의 대상, 복사본의 아카이브 유지 시간 및 데이터 복사전 대기 시간을 제어합니다. 아카이브 세트의 모든 파일은 해당 아카이브 세트와 연결된 볼륨에 복사됩니다. 파일 시스템의 파일은 한 아카이브 세트의 구성원 및 오직 1개의 아카이브 세트일 수 있습니다.

파일을 만들거나 수정하면 아카이버가 이를 아카이브 매체에 복사합니다. 아카이브 파일은 표준 UNIX tar(1) 형식과 호환됩니다. 이로써 Sun Solaris 운영 체제(OS) 및 기타 UNIX 시스템과의 데이터 호환성이 보장됩니다. 이 형식에는 파일 액세스 데이터(inode 정보)와 파일의 경로가 포함됩니다. Sun StorEdge SAM-FS 환경이 완전히 손실되는 경우, tar(1) 형식을 사용하면 표준 UNIX 도구 및 명령을 사용하여 파일을 복구할 수 있습니다. 또한 아카이브 프로세스는 Sun StorEdge SAM-FS 파일 시스템 동작에 필수적인 데이터를 복사합니다. 이 데이터는 디렉토리, 심볼 링크, 세그먼트된 파일의 인덱스 및 아카이브 매체 정보로 구성됩니다.

이 섹션의 나머지 부분에서 *파일*이라는 용어는 파일 데이터 및 메타 데이터를 모두 의미합니다. *파일 데이터* 및 *메타 데이터*라는 용어는 구분이 필요한 경우에만 사용합니다. *파일 시스템*이라는 용어는 마운트된 Sun StorEdge SAM-FS 파일 시스템을 말합니다.

아카이브 세트 이름은 관리자가 결정하며 다음과 같은 예외가 있기는 하지만 사실상 제한이 없습니다.

- 예약된 두 개의 아카이브 세트 이름, 즉 `no_archive` 및 `allsets`가 있습니다.
`no_archive` 아카이브 세트는 기본값으로 정의되어 있습니다. 이 아카이브 세트에 포함될 파일은 아카이브되지 않습니다. 임시 디렉토리(예: `/sam1/tmp`)의 파일은 `no_archive` 아카이브 세트에 포함될 수도 있습니다.
`allsets` 아카이브 세트는 모든 아카이브 세트에 적용되는 매개 변수를 정의하는 데 사용됩니다.
- 각 Sun StorEdge SAM-FS 파일 시스템에 명명된 아카이브 세트는 제어 구조 정보에 대해 예약됩니다. Sun StorEdge SAM-FS 파일 시스템은 각 파일 시스템에 기본 아카이브 세트를 제공합니다. 각 파일 시스템에는 메타 데이터와 데이터 파일이 모두 아카이브됩니다. 파일 시스템 아카이브 세트에는 디렉토리 및 링크 정보 및 다른 아카이브 세트에 포함되지 않는 파일이 포함됩니다. 기본 아카이브 세트에는 연결된 파일 시스템의 이름이 지정되며 이 아카이브 세트는 변경할 수 없습니다. 예를 들어 `samfs1`은 `samfs1`로 구성 및 명명된 파일 시스템의 아카이브 세트 이름입니다.
- 아카이브 세트 이름은 29개 문자로 제한됩니다. 문자는 26개의 대/소문자, 숫자(0~9) 및 밑줄(`_`) 문자로 제한됩니다.

아카이브 작업

기본적으로 아카이버는 각 아카이브 세트의 복사본을 1개씩 만들지만, 각 아카이버 세트에 대해 아카이브 복사본을 최대 4개씩 만들도록 요청할 수 있습니다. 아카이브 세트와 복사본 수는 볼륨 모음의 동의어가 됩니다. 아카이브 복사본은 별도의 볼륨에 대해 파일의 복제를 제공합니다.

파일이 완전한 상태에서 아카이브될 수 있도록 아카이버는 아카이브에 앞서 파일이 수정된 후 지정된 시간동안 대기합니다. 앞서 언급한 것처럼 이 시간을 *아카이브 나이*라고 합니다.

파일을 아카이브 또는 재아카이브의 대상으로 간주하려면 먼저 해당 파일의 데이터를 수정해야 합니다. 파일에 액세스만하면 해당 파일은 아카이브되지 않습니다. 예를 들어 파일에 대해 `touch(1)` 또는 `mv(1)` 명령을 실행하면 파일은 아카이브 또는 재아카이브되지 않습니다. `mv(1)` 명령을 실행하면 파일 이름은 변경되나 파일 데이터는 변경되지 않으며, `tar(1)` 파일에서 복원하는 경우에는 재해 복구 상황에서 여러 가지 문제가 발생할 수 있습니다. 재해 복구에 대한 자세한 내용은 *Sun QFS*, *Sun SAM-FS* 및 *Sun SAM-QFS 재난 복구 안내서*를 참조하십시오.

아카이브할 파일은 아카이브 나이에 따라 선택합니다. 아카이브 나이는 각 아카이브 복사본에 대해 정의할 수 있습니다.

touch(1) 명령을 사용하면 파일의 기본 시간 참조를 과거나 미래의 값으로 변경할 수 있습니다. 그러나 이렇게 하면 예상치 못한 아카이브 결과가 발생할 수 있습니다. 이러한 문제를 예방하기 위해 아카이버는 참조의 범위가 항상 다음과 같도록 참조를 조정합니다.

```
creation_time < time_ref < time_now
```

다음 섹션에서는 초기 파일 스캔에서 파일 복사 프로세스에 이르는 아카이버 수행 작업에 대해 설명합니다.

1 단계: 아카이브로 파일 식별하기

마운트된 파일 시스템 각각에는 별도의 sam-arfind 프로세스가 있습니다. sam-arfind 프로세스는 각 파일 시스템을 모니터링하여 아카이브할 파일을 결정합니다. 파일 시스템은 파일이 해당 아카이브 상태에 영향을 주는 방식으로 변경될 때마다 sam-arfind 프로세스에 통지합니다. 이러한 변경내용의 예는 파일 수정, 재아카이브, 아카이브 해제 및 이름 바꾸기입니다. 통지될 때 sam-arfind 프로세스는 파일을 관찰하여 필요한 아카이브 작업을 결정합니다.

sam-arfind 프로세스는 파일 등록 정보 설명을 사용해 파일이 속하는 아카이브 세트를 결정합니다. 파일의 아카이브 세트를 결정하는 데 사용되는 특성에는 다음 내용들이 포함됩니다.

- 파일 이름의 디렉토리 경로 부분 및 선택적으로 일반 표현식을 사용한 전체 파일 이름
- 파일 소유자의 사용자 이름
- 파일 소유자의 그룹 이름
- 최소 파일 크기
- 최대 파일 크기

1개 이상의 복사본에 대한 파일의 아카이브 나이가 부합 또는 초과된 경우에는 sam-arfind가 아카이브 세트에 대한 하나 이상의 아카이브 요청에 파일을 추가합니다. *아카이브 요청*은 모두 동일한 아카이브 세트에 속하는 파일의 모음입니다. 개별 아카이브는 재아카이브될 파일에 대해 사용됩니다. 이 아카이브를 사용하면 아직 아카이브되지 않은 파일 및 재아카이브될 파일들에 대해 독립적으로 예약을 제어할 수 있습니다. 아카이브 요청은 다음 디렉토리에 상주하는 파일입니다.

```
/var/opt/SUNWsamfs/archiver/file_sys/ArchReq
```

이러한 파일은 이진 파일이며, showqueue(1M) 명령을 사용하면 이를 표시할 수 있습니다.

아카이브 요청은 경우에 따라 *ArchReq*라고도 합니다.

하나 이상의 복사본에 대해 파일의 아카이브 나이가 채워지지 않은 경우, 파일이 상주하는 디렉토리 및 아카이브 나이가 도달되는 시간이 스캔 목록에 추가됩니다. 디렉토리는 스캔 목록 시간에 도달할 때 스캔됩니다. 해당 아카이브 나이에 도달한 파일은 아카이브 요청에 추가됩니다.

파일이 오프라인인 경우, `sam-arfind` 프로세스는 아카이브 복사본의 소스로 사용할 볼륨을 선택합니다. 파일 복사본이 재아카이브되는 경우, `sam-arfind` 프로세스는 재아카이브되는 아카이브 복사본이 포함된 볼륨을 선택합니다.

파일이 세그먼트된 경우에는 변경된 세그먼트만이 아카이브를 위해 선택됩니다. 세그먼트된 파일의 인덱스는 사용자 데이터를 포함하지 않으므로 파일 시스템 아카이브 세트의 구성원으로 취급되며 별도로 아카이브됩니다.

아카이브 우선 순위는 파일 등록 정보 특성 및 아카이브 세트와 연결된 파일 등록 정보 승수로부터 계산됩니다. 기본적으로 아카이브 우선 순위 계산은 다음과 같습니다.

$archive_priority = \text{the sum of } (file_property_value * property_multiplier)$

대부분의 `file_property_value` 번호는 1 또는 0이고, 등록 정보는 TRUE 또는 FALSE입니다. 예를 들어 아카이브 복사본 1을 만드는 경우, 등록 정보 복사본 1의 값은 1입니다. 따라서, 복사본 2, 복사본 3 및 복사본 4의 값은 0입니다.

아카이브 나이 및 파일 크기 등의 값은 0 또는 1이외의 값입니다.

`property_multiplier` 값은 아카이브 세트의 `-priority` 매개 변수로부터 결정합니다. 나이 또는 크기 등 파일의 다양한 측면의 값을 지정할 수 있기 때문에 사이트에서 아카이브 요청의 우선 순위를 변경할 수 있습니다. `-priority` 매개 변수에 대한 자세한 내용은 `archiver.cmd(4)` 매뉴얼 페이지를 참조하십시오.

`archive_priority` 및 등록 정보 승수는 부동 소수점 수입니다. 모든 등록 정보 승수의 기본 값은 0.0입니다. 아카이브 요청은 아카이브 요청의 최고 파일 우선 순위로 설정됩니다.

아카이브용으로 파일을 표시하는 방법은 연속 아카이브 및 스캔의 두 가지가 있습니다. 연속 아카이브의 경우, 아카이버는 아카이브가 필요한 파일을 결정하기 위해 파일 시스템과 함께 작동합니다. 스캔의 경우, 아카이버는 파일 시스템을 정기적으로 검사하고 아카이브용 파일을 선택합니다. 다음 섹션에서는 이러한 방법에 대해 설명합니다.

연속 아카이브

연속 아카이브는 기본 아카이브 방법입니다(`examine=noscan`). 연속 아카이브의 경우, `-startage`, `-startcount` 및 `-startsize` 매개변수를 사용하여 아카이브 세트에 대해 예약 시작 조건을 지정할 수 있습니다. 이 조건을 사용하여 아카이브 적시 대 아카이브 작업 완료를 최적화할 수 있습니다.

- 예제 1. 함께 아카이브해야 할 파일들을 작성하는 데 한 시간이 소요되는 경우, `-startage` 매개변수를 1 시간으로 설정(`-startage 1h`)하면 모든 파일이 아카이브 요청을 예약하기 전에 작성됩니다.
- 예제 2. `-startsize`를 150 기가바이트로 지정하면 아카이브는 150 기가바이트의 데이터가 아카이브 준비될 때까지 기다리게 됩니다.
- 예제 3. 3000개의 파일들을 아카이브용으로 생성할 경우 `-startcount 3000`를 지정하여 해당 파일들이 함께 아카이브 되도록 합니다.

모든 예약 시작 조건이 충족되면 `sam-arfind` 프로세스는 각 아카이브 요청을 아카이버 데몬 `sam-archiverd`에 보내서 아카이브 매체에 파일 복사를 예약합니다.

스캔된 아카이브

연속 아카이브에 대한 대안으로 `examine=scan`을 지정하고 `sam-arfind`가 스캔을 사용하여 아카이브용 파일을 관찰하도록 지시합니다. 아카이브해야 할 파일이 아카이브 요청에 포함됩니다. `sam-arfind` 프로세스는 각 파일 시스템을 정기적으로 스캔하여 아카이브할 파일을 결정합니다. `sam-arfind`가 수행하는 첫 번째 스캔은 디렉토리 스캔입니다. 이 스캔 과정에서 `sam-arfind`는 디렉토리 트리에서 하위 폴더를 포함해 내림차순으로 스캔합니다. 각 파일을 검사한 후, 파일이 아카이브할 필요가 없으면 파일 상태 플래그 `archdone`이 설정됩니다. 연속 스캔 과정에서 `.inodes` 파일이 스캔됩니다. `archdone` 플래그가 설정되지 않은 `inode`만 검사합니다.

파일 시스템 스캔이 완료되면 `sam-arfind` 프로세스는 아카이버 데몬 `sam-archiverd`에 각 아카이브 요청을 보내 아카이브 매체에 파일 복사 예약을 합니다. 그런 다음 `sam-arfind` 프로세스는 `interval=time` 명령에 의해 지정된 기간 동안 휴면 상태가 됩니다. 간격이 끝나면 `sam-arfind` 프로세스는 스캔을 다시 시작합니다.

2 단계: 아카이브 요청 구성

`sam-archiverd` 데몬에서 아카이브 요청을 받으면 이 요청은 *컴포지션*됩니다. 이 단계에서는 컴포지션 프로세스에 대해 설명합니다.

아카이브 요청의 모든 파일이 한번에 아카이브되지 않을 수도 있습니다. 그 원인은 아카이브 매체의 용량 또는 아카이버 명령 파일에 지정된 컨트롤 때문입니다. *컴포지션*이란 아카이브 요청에서 한번에 아카이브할 파일을 선택하는 프로세스를 말합니다. 아카이브 요청에 대한 아카이브 복사 작업이 완료되면 아카이브할 파일이 있는 경우 아카이브 요청이 다시 컴포지션됩니다.

`sam-archiverd` 데몬은 일정한 기본 및 사이트별 기준에 따라 아카이브 요청의 파일 순서를 정합니다. 기본 작업은 아카이브 요청의 모든 파일을 파일 시스템 스캔 과정에서 발견된 순서대로 동일한 아카이브 볼륨에 아카이브하는 것입니다. 사이트별 기준을 사용하면 파일 아카이브 순서와 파일이 볼륨에 분산되는 방식을 제어할 수 있습니다. 이러한 기준을 *아카이브 세트 매개 변수*라고 하며 이를 평가하는 순서는 `-reserve`, `-join`, `-sort`, `-rsort` (역방향 정렬을 수행함) 및 `-drives`입니다. 이러한 매개 변수에 대한 자세한 내용은 `archiver.cmd(4)` 매뉴얼 페이지를 참조하십시오.

아카이브 요청이 `-reserve` 소유자가 지정된 아카이브 세트에 속하는 경우, `sam-archiverd` 데몬은 파일의 디렉토리 경로, 사용자 이름 또는 그룹 이름에 따라 아카이브 요청의 파일 순서를 정합니다. 아카이브 세트의 `-reserve` 매개 변수가 이러한 작업을 제어합니다. 첫 번째 소유자에 속하는 파일을 아카이브로 선택합니다. 나머지 파일은 나중에 아카이브됩니다.

아카이브 요청이 `-join method`가 지정된 아카이브 세트에 속하는 경우 `sam-archiverd` 데몬은 지정된 `-join method`에 따라 파일을 분류합니다. 또한 `-sort` 또는 `-rsort` 방법이 지정된 경우 `sam-archiverd` 데몬이 `-sort` 또는 `-rsort` 방법에 따라 각 그룹 내의 파일들을 정렬합니다. 아카이브 요청은 결합되어 정렬됩니다.

결합된 파일의 각 그룹은 남은 컴포지션 및 예약 프로세스 과정에서 하나의 파일인 것처럼 취급됩니다.

아카이브 요청이 `-sort` 또는 `-rsort` 방법이 지정되어 있는 아카이브 세트에 속한 경우, `sam-archiverd` 데몬은 `-sort` 또는 `-rsort` 매개 변수에 지정된 정렬 방법에 따라 파일을 정렬합니다. 정렬 방법에 따라 `sam-archiverd` 데몬은 정렬 방법, 나이 크기 또는 디렉토리 위치를 기준으로 파일을 분류합니다. 기본적으로 아카이브 요청은 정렬되지 않기 때문에 파일은 파일 시스템 스캔 과정에서 발견되는 순서대로 아카이브됩니다.

`sam-archiverd` 데몬은 파일이 온라인인지 오프라인인지를 결정합니다. 온라인 및 오프라인 파일이 모두 아카이브 요청에 있으면 먼저 온라인 파일이 아카이브를 위해 선택됩니다.

아카이브 요청이 정렬 방법에 의해 합쳐지거나 정렬될 필요가 없는 경우, 오프라인 파일은 아카이브 복사본이 상주하는 볼륨에 의해 순서가 정해집니다. 이렇게 되면 각 아카이브 세트 내에서 동일한 볼륨의 모든 파일이 매체에 저장된 순서에 따라 동시에 스테이지됩니다. 오프라인 파일의 아카이브 복사본이 2개 이상 작성되면 오프라인 파일은 필요한 복사본이 모두 작성되고 난 후에 릴리스됩니다. 동일한 볼륨에서 첫 번째 파일로 스테이지되는 모든 파일이 아카이브 선택됩니다.

`-join`, `-sort` 또는 `-rsort` 매개 변수를 사용하면 오프라인 파일의 아카이브 성능에 좋지 않은 영향을 줍니다. 파일의 아카이브 순서가 오프라인 파일에 필요한 볼륨의 순서와 일치하지 않을 수 있기 때문입니다. 첫 번째 아카이브 복사본 작성 시에만 `-join`, `-sort` 또는 `-rsort` 매개 변수를 사용하는 것이 바람직합니다. 복사본이 시작될 때 사용할 수 있는 아카이브 매체가 충분한 경우에 다른 복사본은 첫 번째 복사본의 순서를 유지합니다.

아카이브 요청은 `sam-archiverd` 데몬의 예약 대기열에 입력됩니다.

3 단계: 아카이브 요청 예약

sam-archiverd 데몬의 스케줄러는 다음과 같은 조건에서 요청이 있으면 실행됩니다.

- 아카이브 요청이 예약 대기열에 입력됩니다.
- 아카이브 요청에 대한 아카이브가 완료되었습니다.
- 카탈로그 서버에서 매체 상태 변경이 수신됩니다.
- 아카이브 상태를 변경하는 메시지가 수신됩니다.

예약 대기열의 아카이브 요청의 순서가 우선 순위에 따라 정해집니다. 스케줄러가 실행될 때마다 모든 아카이브 요청을 검사하여 sam-arcopy 프로세스에 할당해 아카이브 매체에 파일을 복사할 수 있는지 확인합니다.

파일 복사에 사용할 드라이브가 있어야 합니다. 아카이브 세트에서 사용할 수 있고 아카이브 요청의 파일을 보유할 충분한 공간이 있는 볼륨이 있어야 합니다.

드라이브

아카이브 세트에 -drives 매개 변수가 지정되어 있으면 sam-archiverd 데몬은 아카이브 요청의 선택된 파일을 여러 드라이브에 나눕니다. 이 때, 사용 가능한 드라이브의 수가 -drives 매개 변수가 지정한 수보다 적으면 적은 수가 적용됩니다.

아카이브 요청의 파일의 총 크기가 -drivemin 값보다 적으면 1개의 드라이브만 사용합니다. -drivemin 값은 -drivemin 매개 변수에 의해 지정된 값이거나 archmax 값입니다.

archmax 값은 -archmax 매개 변수에 의해 지정된 값이거나 매체에 대해 정의된 값입니다. -archmax 매개 변수 및 archmax= 명령에 대한 자세한 내용은 archiver.cmd(4) 매뉴얼 페이지를 참조하십시오.

아카이브 요청의 파일의 총 크기가 -drivemin 값보다 크면 다음 값이 계산됩니다. $drive_count = total_size / drivemin$. drive_count이 -drives 매개변수가 지정한 드라이브 수보다 적으면 drive_count은 사용할 드라이브의 수가 됩니다.

드라이브는 아카이브 파일에 또다른 시간을 소요할 수 있습니다. -drivemax 매개변수를 사용하여 더 나은 드라이브를 이용할 수 있습니다. -drivemax 매개변수를 사용하면 더 많은 데이터에 대해 해당 드라이브를 재예약하기 전에 드라이브에 기록될 최대 바이트 수를 지정해야 합니다.

볼륨

아카이브 요청에 적어도 일부 파일을 보유할 공간이 충분한 볼륨이 있어야 합니다. 공간이 충분하면 가장 최근에 아카이브 세트에 사용한 볼륨을 사용합니다. 또한 아카이버가 해당 볼륨을 사용하지 않아야 합니다.

아카이버 세트에 사용할 수 있는 볼륨이 현재 사용 중인 경우에는 다른 볼륨을 선택합니다. 이는 `-fillvsns` 매개 변수가 지정되지 않은 경우에 가능합니다. 이 경우에는 아카이버 요청을 예약할 수 없습니다.

아카이버 요청이 너무 커서 1개의 볼륨에 맞지 않은 경우에는 볼륨에 맞는 파일을 선택해 해당 볼륨에 아카이브 합니다. 아카이브 요청에 1개의 볼륨에 맞지 않는 크기의 파일이 포함되고, 해당 아카이브 요청에 대한 볼륨 오버플로가 선택되지 않은 경우에는 파일을 아카이브할 수 없습니다. 이 조건에 적합한 메시지를 로그로 보냅니다.

아카이브 세트(`-ovflmin` 매개 변수 사용) 또는 매체(`ovflmin=` 명령 사용)에 대한 볼륨 오버플로를 지정할 수 있습니다. `-ovflmin` 매개 변수 및 `ovflmin=` 명령에 대한 자세한 내용은 `archiver.cmd(4)` 매뉴얼 페이지를 참조하십시오. 이 지정 `ovflmin`은 매체를 오버플로하는 최소 크기를 결정합니다. 아카이브 세트에 대해 지정된 `ovflmin`은 매체 정의 `ovflmin`보다 우선 순위가 높습니다. 파일의 크기가 `ovflmin`보다 작으면 파일을 아카이브할 수 없습니다. 이 조건에 적합한 메시지를 로그로 보냅니다.

파일의 크기가 `ovflmin`보다 크면 추가 볼륨을 필요한 만큼 지정할 수 있습니다. 추가 볼륨은 파일에 필요한 볼륨의 수를 최소화하기 위해 크기가 큰 순서부터 선택합니다.

아카이브 요청에 대해 사용 가능한 볼륨이 없으면, 아카이브 요청은 대기합니다.

파일이 온라인 또는 오프라인인지 여부와 같은 일부 등록 정보는 특정 아카이브 요청에 대한 예약 우선 순위를 결정할 때 단계 1에서 계산되는 아카이브 우선 순위와 함께 사용됩니다. 등록 정보 승수의 사용자 지정에 대한 자세한 내용은 `archiver.cmd(4)` 매뉴얼 페이지에서 설명한 `-priority` 매개 변수를 참조하십시오.

각 아카이브 요청에 대해 `sam-archiverd` 데몬은 아카이브 우선 순위를 다양한 시스템 리소스 등록 정보와 연결된 승수에 추가하여 예약 우선 순위를 계산합니다. 이러한 등록 정보는 아카이브 프로세스에서 사용할 첫 번째 볼륨의 드라이브 로드 여부에 관계없이 아카이브 요청이 대기 상태로 머문 시간(초)과 연결됩니다.

조정된 우선 순위를 사용하는 `sam-archiverd` 데몬은 준비된 각 아카이브 요청이 복사되도록 지정합니다.

4 단계: 아카이브 요청의 파일 아카이브

아카이브 요청이 아카이브될 준비가 되면 `sam-archiverd` 데몬은 각 아카이브 요청 단계를 거쳐 아카이브 파일(tarball) 경계에 표시를 하기 때문에 각 아카이브 파일의 크기는 `-archmax target_size` 지정 크기보다 작습니다. 1개의 파일이 `target_size`보다 크면 이 파일은 아카이브 파일의 유일한 파일이 됩니다.

사용할 아카이브 요청과 드라이브 각각에 대해 `sam-archiverd` 데몬은 `sam-arcopy` 프로세스에 아카이브 요청을 지정해 아카이브 매체에 파일을 복사합니다. 1개의 파일이 `target_size`보다 크면 이 파일은 아카이브 파일의 유일한 파일이 됩니다. 아카이브 정보는 `inode`에 입력됩니다.

아카이브 로깅이 활성화되면 아카이브 로그 항목이 생성됩니다.

파일이 스테이지 된 경우에는 디스크 공간이 릴리스됩니다. 이 프로세스는 목록의 모든 파일이 아카이브될 때까지 계속됩니다.

다양한 오류와 파일 상태 변경 때문에 파일이 제대로 복사되지 않을 수 있습니다. 여기에는 캐시 디스크의 읽기 오류 및 볼륨에 쓰기 오류 등이 포함될 수 있습니다. 상태 변경에는 선택후 수정, 쓰기용 파일 열기 및 제거된 파일 등이 포함됩니다.

`sam-arcopy` 프로세스가 있으면 `sam-archiverd` 데몬은 아카이브 요청을 검사합니다. 임의의 파일이 아카이브되면 아카이브 요청이 다시 컴포지션됩니다.

예제 기본 수행 결과

코드 예 4-1은 `archiver(1M) -1` 명령을 실행하여 나타난 출력 샘플입니다.

코드 예 4-1 `archiver(1M) -1` 명령 출력

```
# archiver

Archive media:
default:mo
media:mo archmax:5000000
media:lt archmax:50000000
Archive devices:
device:mo20 drives_available:1 archive_drives:1
device:lt30 drives_available:1 archive_drives:1
Archive file selections:
Filesystem samfs1:
samfs1 Metadata
  copy:1 arch_age:240
big path:. minsize:512000
  copy:1 arch_age:240
all path:
  copy:1 arch_age:30
```

```

Archive sets:
all
    copy:1 media:mo
big
    copy:1 media:lt
samfs1
    copy:1 media:mo

```

아카이버 데몬

sam-archiverd 데몬은 아카이브 작업을 예약합니다. sam-arfind 프로세스는 아카이브할 파일을 아카이브 세트에 할당합니다. sam-arcopy 프로세스는 아카이브할 파일을 선택한 볼륨에 복사합니다.

sam-archiverd 데몬은 Sun StorEdge SAM-FS 작업이 시작할 때 sam-fsd에 의해 시작됩니다. sam-archiver 데몬은 archiver(1M) 명령을 실행해 archiver.cmd 파일을 읽고 아카이브 제어에 필요한 테이블을 구축합니다. 마운트된 파일 시스템 각각에 대해 sam-arfind 프로세스가 시작되고, 이와 마찬가지로 파일 시스템이 마운트 해제 되면 연결된 sam-arfind 프로세스가 정지됩니다. 그런 다음, sam-archiverd 프로세스는 sam-arfind를 모니터링하고 운영자 또는 다른 프로세스의 신호를 처리합니다.

아카이브 로그 파일 및 이벤트 로깅

sam-arfind 및 sam-arcopy 프로세스는 아카이브되거나 자동으로 아카이브 해제되는 각 파일에 관한 정보가 포함된 로그 파일을 만듭니다. 로그 파일은 아카이브 작업의 연속 레코드입니다. 로그 파일을 사용해 일반적인 백업 목적으로 파일의 이전 복사본을 찾을 수 있습니다.

이 파일은 기본적으로 생성되지 않습니다. archiver.cmd 파일의 logfile= 명령을 사용해 로그 파일 생성과 로그 파일의 이름을 지정할 수 있습니다. 이 파일의 이름을 결정합니다. 로그 파일에 대한 자세한 내용은 이 장의 70 페이지의 "archiver.cmd 명령" 및 archiver.cmd(4) 매뉴얼 페이지를 참조하십시오.

아카이버는 syslog 기능 및 archiver.sh를 사용해 로그 파일에 경고 및 알람 메시지를 로그합니다

코드 예 4-2는 각 필드에 대한 정의가 있는 아카이버 로그 파일의 예제 행입니다.

코드 예 4-2 아카이버 로그 파일 행

```
A 2001/03/23 18:42:06 mo 0004A arset0.1 9a089.1329 samfs1
118.51162514 t0/fdn f 0 56
A 2001/03/23 18:42:10 mo 0004A arset0.1 9aac2.1 samfs1 189.53
1515016 t0/fae f 0 56
A 2001/03/23 18:42:10 mo 0004A arset0.1 9aac2.b92 samfs1 125.53
867101 t0/fai f 0 56
A 2001/03/23 19:13:09 lt SLOT22 arset0.2 798.1 samfs1 71531.14
1841087 t0/fhh f 0 51
A 2001/03/23 19:13:10 lt SLOT22 arset0.2 798.e0e samfs1 71532.12
543390 t0/fhg f 0 51
A 2003/10/23 13:30:24 dk DISK01/d8/d16/f216 arset4.1 810d8.1 qfs2
119571.301 1136048 t1/fileem f 0 0
A 2003/10/23 13:30:25 dk DISK01/d8/d16/f216 arset4.1 810d8.8ad
qfs2 119573.295 1849474 t1/fileud f 0 0
A 2003/10/23 13:30:25 dk DISK01/d8/d16/f216 arset4.1 810d8.16cb
qfs2 119576.301 644930 t1/fileen f 0 0
A 2003/10/23 13:30:25 dk DISK01/d8/d16/f216 arset4.1 810d8.1bb8
qfs2 119577.301 1322899 t1/fileeo f 0 0
```

왼쪽에서 오른쪽으로 읽는 이전 목록의 필드의 내용은 표 4-1에 나와있습니다.

표 4-1 아카이버 로그 파일 필드

필드	내용
1	아카이브 작업 <ul style="list-style-type: none"> • A(아카이브된 경우) • R(재아카이브된 경우) • U(아카이브되지 않은 경우)
2	yyyy/mm/dd 형식의 아카이브 작업의 날짜.
3	hh:mm:ss 형식의 아카이브 작업 시간.
4	아카이브 매체 유형. 매체 유형에 대한 자세한 내용은 mcf(4) 매뉴얼 페이지를 참조하십시오.
5	VSN 제거 가능한 매체 카트리지의 경우 이것은 볼륨 시리얼 이름입니다. 디스크 아카이브의 경우, 이것은 디스크 볼륨 이름 및 아카이브 tar(1) 파일 경로입니다.
6	아카이브 세트 및 복사본 수.
7	매체(tar(1) 파일)에서 아카이브 파일의 실제 시작 위치와 아카이브 파일의 16진수 파일 오프셋.
8	파일 시스템 이름.

표 4-1 아카이버 로그 파일 필드 (계속)

필드	내용
9	Inode 번호 및 생성 번호. 생성 번호란 inode 번호가 재사용되므로 고유성을 위해 inode 번호 이외에 추가로 사용하는 추가 번호를 말합니다.
10	파일이 1개의 볼륨에만 쓰인 경우 파일의 길이. 파일이 여러 볼륨에 쓰인 경우 섹션의 길이.
11	파일 시스템의 마운트 지점을 기준으로 파일의 경로 및 이름.
12	파일의 유형: <ul style="list-style-type: none"> • d(디렉토리). • f(일반 파일). • l(심볼 링크). • R(제거 가능한 매체 파일). • I(세그먼트 인덱스). • S(데이터 세그먼트).
13	오버플로된 파일이나 세그먼트의 섹션. 해당 파일이 오버플로된 파일인 경우 값은 0 이 아닙니다. 해당 값은 모든 다른 파일 유형에 대해 0입니다.
14	파일이 아카이브된 드라이브의 장비 서수.

archiver.cmd 파일

archiver.cmd 파일은 아카이버의 작업을 제어합니다. 기본적으로 아카이버는 sam-fsd가 시작하고 Sun StorEdge SAM-FS 파일 시스템이 마운트될 때마다 실행합니다. 아카이버의 기본 작업은 다음과 같습니다.

- 사용 가능한 모든 볼륨에 모든 파일을 아카이브합니다.
- 모든 파일의 아카이브 나이는 4분입니다.
- 아카이브 간격은 10분입니다.

사이트의 아카이브 요구 사항에 부합하도록 아카이버의 작업을 사용자 지정할 수 있습니다. 이러한 작업은 아카이버 명령 파일(archiver.cmd)에 있는 명령에 의해 제어됩니다.

▼ archiver.cmd 파일 작성 또는 수정하기 및 변경 내용 전파하기

1. archiver.cmd 파일을 편집할지 아니면 임시 archiver.cmd 파일을 편집할지 결정합니다. (선택 사항)

/etc/opt/SUNWsamfs/archiver.cmd 파일 및 시스템이 이미 파일을 아카이브하는 중이면 이 단계를 수행하십시오. archiver.cmd 파일을 프로덕션 환경에 두기 전에 편집 및 테스트할 수 있는 임시 장소에 복사하십시오.

2. vi(1) 또는 또 다른 편집기를 사용하여 archiver.cmd 파일 및 임시 파일을 편집합니다.

사이트에서 아카이브를 제어하기 위해 필요한 명령을 추가합니다. 이 파일에 포함할 수 있는 명령에 대한 정보는 70 페이지의 "archiver.cmd 명령" 및 109 페이지의 "디스크 아카이브"를 참조하십시오.

3. archiver.cmd 파일 또는 임시 파일을 저장하고 닫습니다.

4. archiver(1M) -lv 명령을 사용하여 올바른 파일인지 확인합니다.

archiver.cmd 파일로 변경하는 경우에는 반드시 archiver(1M) 명령을 사용해 구분 오류를 확인해야 합니다. 다음처럼 archiver(1M) 명령을 지정하면 현재 Sun StorEdge SAM-FS 시스템에 대해 archiver.cmd 파일을 검사할 수 있습니다.

```
# archiver -lv
```

위의 명령은 모든 옵션을 나열하고 archiver.cmd 파일의 목록, 볼륨, 파일 시스템 내용 및 오류의 목록을 표준 출력 파일(stdout)에 작성합니다. 오류가 발생하면 아카이버가 실행되지 않습니다.

기본적으로 archiver(1M) 명령은 /etc/opt/SUNWsamfs/archiver.cmd 파일의 오류 여부를 검사합니다. 프로덕션 환경에 두기 전에 임시 archiver.cmd 파일을 작업하려는 경우 archiver(1M) 명령의 -c 옵션을 사용하고 이 임시 파일의 이름을 제공할 수 있습니다.

5. 파일의 오류가 없어질 때까지 2 단계, 3 단계 및 4 단계를 반복합니다.

다음 단계로 이동하기 전에 모든 오류를 수정해야 합니다. 아카이버는 archiver.cmd 파일에 오류가 있으면 어떤 파일도 아카이브하지 않습니다.

6. 임시 파일을 /etc/opt/SUNWsamfs/archiver.cmd 파일로 이동합니다. (선택 사항)

임시 파일을 작업할 경우에만 이 단계를 수행합니다.

7. archiver.cmd 파일을 저장하고 닫습니다.

8. `samd(1M) config` 명령을 사용하여 파일의 변경 내용을 전파하고 시스템을 재시작합니다.

```
# samd config
```

archiver.cmd 파일

archiver.cmd 파일은 다음과 같은 유형의 명령으로 구성됩니다.

- 일반 명령
- 아카이브 세트 할당 명령
- 아카이브 세트 명령
- VSN 풀 명령
- VSN 연결 명령

명령은 archiver.cmd 파일에서 읽은 텍스트 행으로 구성됩니다. 각 명령 행에는 공백이나 탭으로 구분된 필드가 하나 이상 포함됩니다. 우물정자(#) 이후에 나타나는 텍스트는 주석으로 취급되어 검사되지 않습니다. 행은 끝에 백슬래시(\)를 붙여 다음 행에서 계속할 수 있습니다.

archiver.cmd 파일의 특정 명령은 시간 단위나 바이트 단위를 지정해야 합니다. 이러한 단위를 지정하려면 67 페이지의 표 4-2, “archiver.cmd 파일 명령 단위,”에 나와있는 문자 중 하나를 해당 단위를 나타내는 숫자의 접미사로 사용해야 합니다.

표 4-2 archiver.cmd 파일 명령 단위

장치 접미사	의미
시간 접미사	
s	초.
m	분. 60 초.
h	시. 3,600 초.
d	일 수 86,400 초.
w	주. 604,800 초.
y	년. 31,536,000 초.
크기 접미사	
b	바이트.
k	킬로바이트. 2**10, 즉 1,024 바이트.
M	메가바이트. 2**20, 즉 1,048,576 바이트.

표 4-2 archiver.cmd 파일 명령 단위

장치 접미사	의미
G	기가바이트. 2**30, 즉 1,073,741,824 바이트.
T	테라바이트. 2**40, 즉 1,099,511,627,776 바이트.
P	페타바이트. 2**50, 즉 1,125,899,906,842,624 바이트.
E	엑사바이트. 2**60, 즉 1,152,921,504,606,846,976 바이트.

archiver.cmd 파일 예제

코드 예 4-3에서는 예제 archiver.cmd 파일을 보여줍니다. 오른쪽의 주석은 다양한 유형의 명령을 나타냅니다.

```

interval = 30m                # General directives
logfile = /var/opt/SUNWsamfs/archiver/archiver.log

fs = samfs1                    # Archive Set Assignments
no_archive tmp
work work
    1 1h
    2 3h
images images -minsize 100m
    1 1d
    2 1w
samfs1_all .
    1 1h
    2 1h

fs = samfs2                    # Archive Set Assignments
no_archive tmp
system . -group sysadmin
    1 30m
    2 1h
samfs2_all .
    1 10m
    2 2h

params                          # Archive Set Directives
allsets -drives 2
images.1 -join path -sort size
endparams

vsns                              # VSN Associations
samfs1.1    mo    optic-2A
samfs1.2    lt    TAPE01
work.1      mo    optic-[3-9][A-Z]
work.2      lt    .*
images.1    lt    TAPE2[0-9]
images.2    lt    TAPE3[0-9]
samfs1_all.1 mo.*
samfs1_all.2 lt.*
samfs2.1    mo    optic-2A
samfs2.2    lt    TAPE01
system.1    mo    optic08a optic08b
system.2    lt    ^TAPE4[0-1]
samfs2_all.1 mo.*
samfs2_all.2 lt.*
endvsns

```

archiver.cmd 명령

다음 섹션에서는 archiver.cmd 명령에 대해 설명합니다. 이러한 명령은 다음과 같습니다.

- 70 페이지의 "전역 아카이브 명령"
- 79 페이지의 "기타 파일 시스템 명령"
- 80 페이지의 "아카이브 세트 할당 명령"
- 87 페이지의 "아카이브 복사 명령"
- 90 페이지의 "아카이브 설정 복사 매개변수"
- 105 페이지의 "VSN 연결 명령"
- 107 페이지의 "VSN 풀 명령"

전역 아카이브 명령

일반적인 명령은 전체 아카이버 작업을 제어합니다. archiver.cmd 파일의 일반 명령은 두 번째 필드의 등호(=)나 추가 필드의 부재에 의해 식별할 수 있습니다. 이러한 명령을 사용하면 아카이버 작업을 사이트 구성에 맞게 최적화할 수 있습니다.

전역 명령은 archiver.cmd 파일의 fs= 명령에 앞서 지정해야 합니다. fs= 명령은 특정 파일 시스템에 속하는 명령입니다. 아카이버가 fs= 명령 이후에 전역 명령을 발견하면 메시지를 생성합니다.

archivemeta 명령: 메타데이터의 아카이브 여부 제어

archivemeta 명령은 파일 시스템 메타데이터가 아카이브되는지 여부를 제어합니다. 파일이 자주 이동하며 파일 시스템의 디렉토리 구조에 일반적으로 변경 내용이 많은 경우 메타데이터를 아카이브할 수 있습니다. 그러나 디렉토리 구조가 아주 안정적인 경우 메타데이터 아카이브를 비활성화하고 카트리지가 메타데이터를 아카이브하기 위해 로드되고 언로드될 때 제거 가능한 매체 장치에 의해 수행된 작업을 줄일 수 있습니다. 기본적으로 메타데이터는 아카이브됩니다.

이 명령의 형식은 다음과 같습니다.

```
archivemeta = state
```

state에 대해 on 또는 off를 지정합니다. 기본값은 on입니다.

메타데이터 아카이브는 다음과 같이 버전 1 또는 버전 2 수퍼 블록 사용 여부에 따라 다릅니다.

- 버전 1 파일 시스템의 경우 아카이버는 디렉토리, 제거 가능한 매체 파일, 세그먼트 인덱스 `inode` 및 심볼릭 링크를 메타데이터로 아카이브합니다.
- 버전 2 파일 시스템의 경우, 제거 가능한 매체 파일 및 심볼릭 링크는 데이터 블록이 아니라 `inode`에 저장됩니다. 이들은 아카이브되지 않습니다. 디렉토리 및 세그먼트 인덱스 `inode`만 메타데이터로 아카이브됩니다. 심볼릭 링크는 데이터로 아카이브됩니다.

archmax 명령: 아카이브 파일의 크기 제어

`archmax` 명령은 아카이브 파일의 최대 크기를 지정합니다. 사용자 파일이 결합해 아카이브 파일이 됩니다. `target_size`이 충족된 후에는 아카이브 파일에 사용자 파일이 추가되지 않습니다. 대형 사용자 파일은 1개의 아카이브 파일에 쓰여집니다.

기본값을 변경하려면 다음 명령을 사용합니다.

```
archmax=media target_size
```

표 4-3 archmax 명령에 대한 인수

인수	의미
<i>media</i>	매체 유형. 올바른 매체 유형의 목록에 대해서는 <code>mcf(4)</code> 매뉴얼 페이지를 참조하십시오.
<i>target_size</i>	아카이브 파일의 최대 크기를 지정합니다. 아카이브 파일의 최대 크기는 매체에 따라 다릅니다. 기본적으로 광디스크에 작성되는 아카이브 파일은 5 메가바이트 이하입니다. 테이프용 최대 아카이브 파일의 기본 크기는 512 메가바이트입니다.

아카이브 파일에 대해 크거나 작은 크기를 설정하는 데는 장단점이 있습니다. 예를 들어 테이프에 아카이브하고 `archmax`를 큰 크기로 설정하면 테이프 드라이브가 멈췄다가 시작되는 경우가 발생할 수 있습니다. 그러나 대형 아카이브 파일을 쓰는 경우에는 테이프의 끝에 너무 일찍 도달하면 테이프의 대부분을 낭비할 수 있습니다. 일반적으로 `archmax`는 매체 용량의 5 퍼센트 이하로 설정하는 것이 바람직합니다. 예를 들어, 20기가바이트 테이프에 대해 다음 `archmax` 명령을 사용할 수 있습니다.

```
archmax=sg 1G
```

또한 `archmax` 명령은 각각의 아카이브 세트에 대해 설정할 수 있습니다.

bufsize 명령: 아카이버 버퍼 크기 설정

기본적으로 아카이브되는 파일은 메모리 버퍼를 사용하는 아카이브 매체에 복사됩니다. bufsize 명령을 사용하면 기본 값이 아닌 버퍼 크기를 지정할 수 있으며 버퍼를 잠글 수도 있습니다. 이렇게 하면 성능을 향상시킬 수 있을뿐만 아니라 다양한 *buffer_size* 값을 시험해 볼 수 있습니다.

이 명령의 형식은 다음과 같습니다.

```
bufsize=매체 버퍼크기 [ lock ]
```

표 4-4 bufsize 명령에 대한 인수

인수	의미
<i>media</i>	매체 유형. 올바른 매체 유형의 목록에 대해서는 mcf(4) 매뉴얼 페이지를 참조하십시오.
<i>buffer_size</i>	2부터 32까지 숫자를 지정합니다. 기본값은 4입니다. 이 값에 매체 유형에 대한 <i>dev_blksize</i> 값이 곱해지고 해당 결과 버퍼 크기가 사용됩니다. <i>dev_blksize</i> 는 <i>defaults.conf</i> 파일에 지정할 수 있습니다. 이 파일에 대한 자세한 내용은 <i>defaults.conf</i> (4) 매뉴얼 페이지를 참조하십시오.
<i>lock</i>	<p><i>lock</i> 인수는 아카이버가 아카이브 복사본을 만들 때 잠긴 버퍼를 사용하는지 여부를 나타냅니다. <i>lock</i>이 지정되면 아카이버가 <i>sam-arcopy</i>(1M) 작업 도중 메모리의 아카이브 버퍼에 파일 잠금을 설정합니다. 이렇게 하면 각 입출력 요청에 대해 버퍼의 잠금 및 잠금 해제 오버헤드를 피할 수 있고 시스템 CPU 시간을 줄일 수 있습니다.</p> <p><i>lock</i> 인수는 메모리 용량이 큰 대형 시스템인 경우에만 지정해야 합니다. 메모리가 충분하지 않으면 메모리 부족 현상이 발생할 수 있습니다.</p> <p><i>lock</i> 인수는 아카이브되는 파일에 대해 직접 입출력을 사용하는 경우에만 효과적으로 사용할 수 있습니다. 기본적으로 <i>lock</i>은 지정되지 않고 파일 시스템은 아카이브 대상을 포함한 모든 직접 입출력 버퍼에 대해 잠금을 설정합니다. 직접 입출력 사용에 대한 자세한 내용은 <i>setfa</i>(1) 매뉴얼 페이지, <i>sam_setfa</i>(3) 라이브러리 루틴 매뉴얼 페이지 또는 <i>mount_samfs</i>(1M) 매뉴얼 페이지의 <i>-o forcedirectio</i> 옵션을 참조하십시오.</p>

예를 들어 이 명령은 *archiver.cmd* 파일에 다음과 같은 행처럼 지정할 수 있습니다.

```
bufsize=od 7 lock
```

-bufsize 및 *-lock* 아카이브 세트 복사 매개변수를 사용하여 아카이브 세트를 기본으로 버퍼 크기 및 잠금을 지정할 수 있습니다. 자세한 내용은 90 페이지의 "아카이브 설정 복사 매개변수"를 참조하십시오.

drives 명령: 아카이브에 사용된 드라이브 수 제어

기본적으로 아카이버는 아카이브를 위해 자동화 라이브러리의 드라이브 모두를 사용합니다. 아카이버가 사용하는 자동화 라이브러리의 수를 제한하려면 드라이브 명령을 사용해야 합니다.

이 명령의 형식은 다음과 같습니다.

```
drives=auto_lib count
```

표 4-5 drives 명령에 대한 인수

인수	의미
<i>auto_lib</i>	mcf 파일에 정의된 자동화 라이브러리의 패밀러 세트 이름
<i>count</i>	아카이브 작업에 사용할 드라이브의 수

또한 92 페이지의 "아카이브 요청에 대한 드라이브 수 지정: -drivemax, -drivemin 및 -drives"에 설명된 -drivemax, -drivemin 및 -drives 아카이브 세트 복사 매개변수를 참조하십시오.

examine 명령: 아카이브 스캔 제어

새 파일 및 변경된 파일들은 아카이브 대상입니다. 아카이버는 다음 방법 중 하나를 구현하여 해당 파일들을 찾습니다.

- 연속 아카이브. 연속 아카이브가 구현되면 아카이버는 발생 후 즉시 파일 변경 내용을 감지하도록 파일 시스템과 함께 작동합니다.
- 스캔 기반 아카이브. 스캔 기반 아카이브의 경우, 아카이버는 아카이브가 필요한 파일을 찾아 정기적으로 파일 시스템을 스캔합니다.

examine 명령은 다음과 같이 아카이버가 연속 또는 스캔 기반 아카이브를 수행하는지 여부를 제어합니다.

```
examine=방법
```

*method*에 대해 표 4-6에 표시된 키워드 중 하나를 지정합니다.

표 4-6 examine 명령의 *method* 인수에 대한 값

<i>method</i> 값	의미
noscan	연속 아카이브를 지정합니다. 초기 스캔 후 디렉토리는 내용이 변경되거나 아카이브가 요청될 때에만 스캔됩니다. 디렉토리 및 inode 정보는 스캔되지 않습니다. 이 아카이브 방법의 성능이 특히 1,000,000개 이상의 파일이 있는 파일 시스템의 경우 스캔 기반 아카이브보다 더 좋습니다. 기본값입니다.
scan	스캔 기반 아카이브를 지정합니다. 초기 파일 시스템 스캔은 디렉토리 스캔입니다 후속 스캔은 inode 스캔입니다.
scandirs	디렉토리에 있는 스캔 기반 아카이브만 지정합니다. 지정되면, 아카이버가 no_archive 속성 세트를 사용하여 디렉토리를 찾는 경우, 해당 디렉토리는 스캔되지 않습니다. 변경내용이 없는 파일은 해당 디렉토리에 둘 수 있으며 그러면 아카이브 스캔에 소요된 많은 시간을 극적으로 줄일 수 있습니다.
scaninodes	inode에 있는 스캔 기반 아카이브만 지정합니다.

interval 명령: 아카이브 간격 지정

아카이버는 마운트된 모든 Sun StorEdge SAM-FS 파일 시스템의 상태를 관찰하도록 정기적으로 실행합니다. 타이밍은 아카이브 간격에 의해 제어됩니다. *아카이브 간격*이란 각 파일 시스템에 대한 스캔 작업 간의 시간을 말합니다. 이 시간을 변경하려면 interval 명령을 사용해야 합니다.

참고 - interval 명령은 examine=scan 명령도 archiver.cmd 파일에 지정된 경우에만 효력을 발생합니다.

이 명령의 형식은 다음과 같습니다.

```
interval=time
```

*time*의 경우, 파일 시스템의 스캔 동작 사이의 시간을 초로 지정합니다. 기본적으로 *time*은 초로 해석됩니다. 기본적으로 interval=600은 10분입니다. 분, 시간 등과 같은 시간의 단위를 지정할 수 있습니다. 시간 단위 지정에 대한 정보는 67 페이지의 표 4-2, “archiver.cmd 파일 명령 단위, ”를 참조하십시오.

아카이버는 samu(1M) 유틸리티의 :arrun 명령을 받는 즉시 모든 파일 시스템을 스캔하기 시작합니다. 또한 examine=scan 명령이 archiver.cmd 파일에 지정된 경우 :arrun 또는 :arscan이 실행된 후 스캔이 수행됩니다.

파일 시스템에 대해 `hwm_archive` 마운트 옵션이 설정된 경우에는 아카이브 간격을 자동으로 단축할 수 있습니다. 이 마운트 옵션은 파일 시스템이 채워지고 최고 워터마크가 초과되면 아카이버가 스캔을 시작하도록 지정합니다. `high=percent` 마운트 옵션은 파일 시스템에 대해 최고 워터마크를 설정합니다.

아카이브 간격 지정에 대한 자세한 내용은 `archiver.cmd(4)` 매뉴얼 페이지를 참조하십시오. 사용 가능한 마운트 매개 변수에 대한 자세한 내용은 `mount_samfs(1M)` 매뉴얼 페이지를 참조하십시오.

logfile 명령: 아카이버 로그 파일 지정

아카이버는 아카이브, 재아카이브 또는 자동으로 아카이브 해제되는 각 파일 관한 정보가 포함된 로그 파일을 생성합니다. 로그 파일은 아카이브 작업의 연속 레코드입니다. 로그 파일을 지정하려면 `logfile` 명령을 사용해야 합니다. 이 명령의 형식은 다음과 같습니다.

```
logfile=pathname
```

`pathname`에 대해 로그 파일의 절대 경로 및 이름을 지정합니다. 기본적으로 이 파일은 생성되지 않습니다.

예제. 전날의 로그 파일을 대체 위치에 복사하여 아카이버 로그 파일을 매일 백업한다고 가정합니다. 이를 위해서는 반드시 아카이버 로그 파일을 닫은 상태에서 복사 작업을 수행해야 합니다. 즉, 쓰기 작업을 위해 아카이버 로그 파일을 열어놓은 상태에서는 복사 작업을 수행해서는 안 됩니다.

▼ 아카이버 로그 파일 백업하기

다음과 같은 작업을 수행해야 합니다.

1. `mv(1)` 명령을 사용해 아카이버 로그 파일을 UFS 안으로 이동합니다.
이렇게 하면 아카이버 로그 파일에 쓰기 작업을 마칠 수 있는 `sam-arfind(1M)` 또는 `sam-arcopy(1M)` 작업 시간이 생깁니다.
2. `mv(1)` 명령을 사용하여 전날의 아카이버 로그 파일을 Sun StorEdge SAM-FS 파일 시스템으로 이동합니다.
`logfile` 명령은 또한 각각의 파일 시스템에 대해 설정할 수 있습니다.

notify 명령: 이벤트 통지 스크립트 이름 바꾸기

notify 명령은 *filename*에 아카이버의 이벤트 통지 스크립트 파일 이름을 설정합니다. 이 명령의 형식은 다음과 같습니다.

```
notify=filename
```

*filename*에 대해 아카이버 이벤트 통지 스크립트 또는 이 파일에 대한 전체 경로를 포함하는 파일의 이름을 지정합니다.

기본 파일 이름은 다음과 같습니다.

```
/etc/opt/SUNWsamfs/scripts/archiver.sh
```

아카이버는 이 스크립트를 실행하여 사이트에 고유한 방식으로 다양한 이벤트를 처리합니다. 스크립트는 첫 번째 인수에 대한 키워드와 함께 호출됩니다. 다음 키워드에 대해 설명합니다. emerg, alert, crit, err, warning, notice, info 및 debug.

추가 인수에 대해서는 기본 스크립트에 설명되어 있습니다. 자세한 내용은 archiver.sh(1M) 매뉴얼 페이지를 참조하십시오.

ovflmin 명령: 볼륨 오버플로 제어

*볼륨 오버플로*란 아카이브된 파일이 여러 볼륨을 스캔할 수 있도록 하는 프로세스를 말합니다. archiver.cmd 파일에서 ovflmin 명령을 사용하면 볼륨 오버플로가 활성화됩니다. 파일 크기가 ovflmin 명령의 *minimum_file_size* 인수를 초과할 경우 필요에 따라 아카이버는 이 파일의 다른 부분을 유형이 동일한 다른 볼륨에 씁니다. 각 볼륨에 쓰여지는 파일의 부분을 *절*이라고 합니다.

참고 - 볼륨 오버플로를 사용하려면 먼저 그 개념을 이해해야 합니다. 볼륨 오버플로는 사이트에 미칠 수 있는 결과를 철저히 분석한 후에 주의해서 사용해야 합니다. 재해 복구 및 리사이클은 볼륨을 스캔하는 파일보다 훨씬 어렵습니다.

아카이버는 ovflmin 명령을 통해 볼륨 오버플로를 제어합니다. ovflmin 명령은 볼륨을 오버플로할 수 있는 최소 크기 파일을 지정합니다. 기본적으로 볼륨 오버플로는 비활성화되어 있습니다.

이 명령의 형식은 다음과 같습니다.

```
ovflmin = media minimum_file_size
```

표 4-7 ovflmin 명령에 대한 인수

인수	의미
<i>media</i>	매체 유형. 올바른 매체 유형의 목록에 대해서는 mcf(4) 매뉴얼 페이지를 참조하십시오.
<i>minimum_file_size</i>	오버플로할 파일의 최소 크기를 지정합니다.

예제 1. 예를 들어 길이가 mo 매체 카트리지의 상당 부분(가령 25 퍼센트)을 차지하는 파일이 여러 개 있다고 가정합니다. 이러한 파일은 볼륨을 부분적으로 채우고 사용하지 않은 공간은 각 볼륨에 남겨둡니다. 볼륨을 좀 더 효과적으로 압축하려면 mo 매체의 ovflmin을 최소 파일의 크기보다 약간 작은 크기로 설정해야 합니다. 다음 명령은 이를 150 메가바이트로 설정합니다.

```
ovflmin=mo 150m
```

이 예제의 볼륨 오버플로를 활성화해도 아카이브 및 스테이지 하기 위해 2개의 볼륨이 파일을 로드합니다.

ovflmin 명령은 또한 각각의 아카이브 세트에 대해 설정할 수 있습니다.

예제 2. sls(1) 명령은 각 VSN에 있는 파일의 각 절을 표시하는 아카이브 사본을 나열합니다. 코드 예 4-4는 아카이버 로그 파일 및 여러 볼륨을 스캔하는 일명 file50이라는 큰 파일에 대한 sls -D 명령 출력을 표시합니다.

코드 예 4-4 아카이버 로그 파일 예제

```
A 97/01/13 16:03:29 lt DLT000 big.1 7eed4.1 samfs1 13.7
477609472 00 big/file50 0 0

A 97/01/13 16:03:29 lt DLT001 big.1 7fb80.0 samfs1 13.7
516407296 01 big/file50 0 1

A 97/01/13 16:03:29 lt DLT005 big.1 7eb05.0 samfs1 13.7
505983404 02 big/file50 0 2
```

코드 예 4-4는 file50가 DLT000, DLT001 및 DLT005의 VSN를 사용하여 세 볼륨을 스캔하는 것을 표시합니다. 각 섹션의 볼륨과 크기의 위치는 각각 일곱 번째 및 열 번째 필드에 표시되며, 함께 표시되는 sls -D 결과와 일치합니다. 아카이버 로그 항목에 대한 자세한 설명은 archiver(1M) 매뉴얼 페이지를 참조하십시오.

코드 예 4-5는 `sls -D` 명령 및 출력을 표시합니다.

코드 예 4-5 `sls(1M) -D` 명령 및 출력

```
# sls -D file50
file50:
  mode: -rw-rw----  links: 1  owner: gmm    group: sam
  length: 1500000172  admin id: 7  inode: 1407.5
  offline;  archdone;  stage -n
  copy1 ---- Jan 13 15:55          lt
    section 0: 477609472    7eed4.1    DLT000
    section 1: 516407296    7fb80.0    DLT001
    section 2: 505983404    7eb05.0    DLT005
  access:      Jan 13 17:08  modification: Jan 10 18:03
  changed:     Jan 10 18:12  attributes:    Jan 13 16:34
  creation:    Jan 10 18:03  residence:     Jan 13 17:08
```

볼륨 오버플로 파일은 체크섬을 생성하지 않습니다. 체크섬 사용에 관한 자세한 내용은 `ssum(1)` 매뉴얼 페이지를 참조하십시오.

참고 – 볼륨 오버플로 기능을 사용할 때 재해 발생시 볼륨 오버플로 데이터를 검색하기 어렵습니다. 해당 파일을 검색하는 방법에 대한 정보는 *Sun QFS, Sun SAM-FS 및 Sun SAM-QFS 재난 복구 안내서*의 예제를 참조하십시오. 자세한 내용은 `request(1)` 매뉴얼 페이지를 참조하십시오.

wait 명령: 아카이버 시작 연기

`wait` 명령을 사용하면 아카이버가 `samu(1M)` 또는 `SAM-QFS Manager`에서 시작 신호를 기다립니다. 신호가 수신되면 일반적인 아카이버 작업이 시작됩니다. 기본적으로 아카이버는 `sam-fsd(1M)`에 의해 시작될 때 아카이브를 시작합니다. 아카이브를 지연하려면 `wait` 명령을 사용합니다. 이 명령의 형식은 다음과 같습니다.

```
wait
```

또한 `wait` 명령은 또한 각각의 파일 시스템에 대해 설정할 수 있습니다.

기타 파일 시스템 명령

`fs=` 명령을 사용하여 일반적인 명령 후 `archiver.cmd` 파일의 특정 파일 시스템에 고유한 명령을 포함할 수 있습니다. `fs=` 명령이 나타난 후 아카이버는 모든 후속 명령이 개별 파일 시스템에 대해서만 처리되는 작업을 지정한다고 가정합니다.

`fs` 명령: 파일 시스템 지정

기본적으로 아카이브는 모든 파일 시스템에 적용됩니다. 그러나 일부 컨트롤을 개별 파일 시스템으로 한정할 수 있습니다. 개별 파일 시스템을 지정하려면 `fs` 명령을 사용해야 합니다. 이 명령의 형식은 다음과 같습니다.

```
fs=fsname
```

*fsname*에 대해 `mcf` 파일에 정의된 파일 시스템 이름을 지정합니다.

이러한 명령 다음에 오는 일반 명령 및 아카이브 세트 연결 명령은 다른 `fs=` 명령이 나타날 때까지 지정된 파일 시스템에만 적용됩니다. 예를 들면 이 명령을 사용해 각 파일 시스템에 대해 다른 로그 파일을 지정할 수 있습니다.

기타 파일 시스템 명령

몇몇 명령은 모든 파일 시스템에 대한 전역 명령 및 1개의 파일 시스템에 대한 고유 명령으로 동시에 지정될 수 있습니다. 이러한 명령의 결과는 지정된 위치에 관계없이 동일합니다. 이러한 명령은 다음과 같습니다.

- `interval` 명령. 자세한 내용은 74 페이지의 "interval 명령: 아카이브 간격 지정"을 참조하십시오.
- `logfile` 명령. 자세한 내용은 75 페이지의 "logfile 명령: 아카이버 로그 파일 지정"을 참조하십시오.
- `wait` 명령. 자세한 내용은 78 페이지의 "wait 명령: 아카이버 시작 연기"를 참조하십시오.

아카이브 세트 할당 명령

기본적으로 파일은 파일 시스템에 대해 지정된 아카이브 세트의 일부로 아카이브됩니다. SAM-QFS Manager에서 *archive policy*는 아카이브 세트를 정의합니다. 그러나 유사한 특성을 공유하는 파일이 포함된 아카이브 세트를 지정할 수 있습니다. 파일이 지정된 아카이브 세트 중 하나와 일치하지 않는 경우, 해당 파일은 파일 시스템에 지정된 기본 아카이브 세트의 일부로 아카이브됩니다.

아카이브 세트 할당

아카이브 세트 구성원 명령은 아카이브 세트에 유사한 특성의 파일을 할당합니다. 이러한 명령의 구문은 `find(1)` 명령에 따라 설정됩니다. 각 아카이브 세트 할당 명령의 형식은 다음과 같습니다.

```
archive_set_name path [search_criteria1 search_criteria2 ... ] [file_attributes]
```

표 4-8 아카이브 세트 할당 명령에 대한 인수

인수	의미
<i>archive_set_name</i>	아카이브 세트의 사이트 정의 이름. 아카이브 세트 할당 명령에서 첫 번째 필드여야 합니다. 아카이브 세트 이름은 대개 아카이브 세트에 속하는 파일의 특성을 나타냅니다. 아카이브 세트 이름은 알파벳, 숫자 및 밑줄 문자(_)로 제한됩니다. 기타 특수 문자 또는 공백은 사용할 수 없습니다. 아카이브 세트 이름의 첫 번째 문자는 알파벳이어야 합니다. 다양한 파일의 아카이브를 방지하려면 <code>no_archive</code> 를 <i>archive_set_name</i> 로 지정해야 합니다.
<i>path</i>	파일 시스템의 마운트 지점을 기준으로 한 경로. 이 경로를 사용하면 아카이브 세트 구성원 명령을 여러 Sun StorEdge SAM-FS 파일 시스템에 적용할 수 있습니다. 경로에 파일 시스템의 모든 파일을 포함하려면, 경로 필드에 마침표(.)를 사용해야 합니다. 경로 앞에는 슬래시(/)를 사용할 수 없습니다. <i>path</i> 에 의해 지정된 디렉토리 및 하위 디렉토리의 파일은 이 아카이브 세트에 포함되는 것으로 간주합니다.
<i>search_criteria1</i> <i>search_criteria2</i>	0개, 1개 또는 그 이상의 <i>search_criteria</i> 인수를 지정할 수 있습니다. 검색 기준을 지정해 파일 크기, 파일 소유권 및 기타 요소에 따라 아카이브 세트를 제한할 수 있습니다. 가능한 <i>search_criteria</i> 인수에 대한 내용은 다음 섹션을 참조하십시오.
<i>file_attributes</i>	0개, 1개 또는 그 이상의 <i>file_attributes</i> 를 지정할 수 있습니다. 이러한 파일 속성은 <code>sam-arfind</code> 프로세스가 아카이브 도중 파일 시스템을 스캔하면 파일에 대해 설정됩니다.

예제 1. 코드 예 4-6은 일반적인 아카이브 세트 구성원 명령을 표시합니다.

코드 예 4-6 아카이브 세트 구성원 명령

```
hmk_files    net/home/hmk      -user hmk
datafiles    xray_group/data  -size 1M
system       .
```

예제 2. 일명 `no_archive` 아카이브 세트에 파일을 포함시켜 아카이버를 억제할 수 있습니다. 코드 예 4-7은 모든 레벨의 `tmp` 디렉토리 및 해당 파일 시스템 내에서 `tmp` 디렉토리가 상주하는 디렉토리에 상관없이 파일의 아카이브를 막는 행을 표시합니다.

코드 예 4-7 아카이브를 막는 아카이브 명령

```
fs = samfs1
no_archive tmp
no_archive . -name */tmp/
```

다음 섹션에서는 지정할 수 있는 `search_criteria`에 대해 설명합니다.

파일 크기 `search_criteria`: `-access`

`-access age` 특성을 사용하여 파일의 나이가 아카이브 세트 구성원을 결정하는 데 사용되도록 지정할 수 있습니다. 이 `search_criteria`를 사용할 때 `age`보다 많은 액세스 시간을 가진 파일이 다른 매체에 재아카이브됩니다. `age`에 대해 표 4-9에 표시된 접미사 중 하나가 이어지는 정수를 지정합니다.

표 4-9 `-access age` 접미사

문자	의미
s	초
m	분
h	시
d	일 수
w	주
y	년

예를 들어 이 명령을 사용하여 오랜 시간 액세스하지 않던 파일이 더 저렴한 매체에 재아카이브 되도록 지정합니다.

파일 크기 *search_criteria*: -minsize 및 -maxsize

파일의 크기는 -minsize 크기 및 -maxsize 크기 특성을 사용하여 아카이브 세트 구성원을 결정하는 데 사용될 수 있습니다. size에는 표 4-10에 나와있는 문자 중 하나가 뒤에 붙는 정수를 지정합니다.

표 4-10 -minsize 및 -maxsize 크기 접미사

문자	의미
b	바이트
k	킬로바이트
M	메가바이트
G	기가바이트
T	테라바이트
P	페타바이트
E	엑사바이트

예제. 이 예제의 행은 500 킬로바이트 이상, 100 메가바이트 미만의 모든 파일이 아카이브 세트 big_files에 속하도록 지정합니다. 100 메가바이트보다 큰 파일은 아카이브 세트 huge_files에 속합니다. 코드 예 4-8에 행이 나옵니다.

코드 예 4-8 -minsize 및 -maxsize 명령 예제 사용

```
big_files . -minsize 500k -maxsize 100M
huge_files . -minsize 100M
```

소유자 및 그룹 *search_criteria*: -user 및 -group

소유권 및 그룹 정보는 -user name 및 -group name 특성을 사용하여 아카이브 세트 구성원을 결정하는 데 사용될 수 있습니다. 코드 예 4-9는 이 명령의 예제를 표시합니다.

코드 예 4-9 -user 및 -group 명령 사용 예제

```
adm_set . -user sysadmin
mktng_set . -group marketing
```

사용자 sysadmin에 속하는 모든 파일은 아카이브 세트 adm_set에 속하고, 그룹 이름이 marketing인 모든 파일은 아카이브 세트 mktng_set에 속합니다.

패턴 일치를 사용한 파일 이름 *search_criteria*: `-name regex`

일반 표현식을 사용하면 아카이브 세트에 포함시킬 파일 이름을 지정할 수 있습니다. `-name regex`를 *search_criteria*로 지정하면 일반 표현식 *regex*와 일치하는 전체 경로 이름이 아카이브 세트의 구성원이 되도록 지정됩니다.

regex 인수는 `regexp(5)` 매뉴얼 페이지에서 설명한 규칙을 따릅니다. 일반 표현식은 UNIX 와일드카드와 동일한 규칙을 따르지 않습니다.

내부에서는 선택된 디렉토리 아래의 모든 파일이 나열되어(지정 경로의 기준은 파일 시스템의 마운트 지점) 패턴이 일치되도록 위해 전달됩니다. 이렇게 되면 `-name regex` 필드에 패턴을 만들어 파일 이름과 경로 이름을 모두 일치시킬 수 있습니다.

예)

1. 다음 명령은 아카이브 세트 `images`의 파일을 `.gif`로 끝나는 파일로 제한합니다.

```
images . -name \.gif$
```

2. 다음 명령은 문자 `GEO`로 시작하는 파일을 선택합니다.

```
satellite . -name /GEO
```

3. `no_archive` 아카이브 세트를 가지고 일반 표현식을 사용할 수 있습니다. 다음과 같이 지정하면 `.o`로 끝나는 파일이 아카이브되지 않습니다.

```
no_archive . -name \.o$
```

4. `archiver.cmd` 파일에 코드 예 4-10에 표시된 행이 포함된다고 가정합니다.

코드 예 4-10 일반 표현식 예제

```
# File selections.
fs = samfs1
    1 ls
    2 ls
no_archive share/marketing -name fred\.
```

이 `archiver.cmd` 파일에 대해 아카이버는 사용자 디렉토리 또는 하위 디렉토리에 `fred.*`를 아카이브하지 않습니다. 파일의 아카이브는 다음과 같습니다.

- 코드 예 4-11은 코드 예 4-10에 표시된 명령을 지정한 경우 아카이브되지 않은 파일을 표시합니다.

코드 예 4-11 아카이브되지 않은 파일(코드 예 4-10에 표시된 명령을 가정함)

```
/saml/share/marketing/fred.anything
/saml/share/marketing/first_user/fred.anything
/saml/share/marketing/first_user/first_user_sub/fred.anything
```

- 코드 예 4-12는 코드 예 4-10에 표시된 명령을 지정한 경우 아카이브된 파일을 표시합니다.

코드 예 4-12 아카이브된 파일(코드 예 4-10에 표시된 명령을 가정함)

```
/saml/fred.anything
/saml/share/fred.anything
/saml/testdir/fred.anything
/saml/testdir/share/fred.anything
/saml/testdir/share/marketing/fred.anything
/saml/testdir/share/marketing/second_user/fred.anything
```

5. archiver.cmd 파일에 코드 예 4-13에 표시된 행이 포함된다고 가정합니다.

코드 예 4-13 예제 archiver.cmd 파일

```
# File selections.
fs = samfs1
   1 ls
   2 ls
no_archive share/marketing -name ^share/marketing/[^/]*fred\.
```

코드 예 4-13의 archiver.cmd 파일은 사용자 홈 디렉토리에 fred.*를 아카이브하지 않습니다. 이 파일은 사용자 하위 디렉토리와 디렉토리 share/marketing에 fred.*를 아카이브합니다. 이 경우 사용자 홈 디렉토리는 first_user입니다. 이 예제에서는 다음 슬래시(/)까지 share/marketing/에서 어느 것이나 사용자의 홈 디렉토리로 사용합니다. 파일의 아카이브는 다음과 같습니다.

- 다음 파일은 아카이브되지 않습니다.

```
/saml/share/marketing/first_user/fred.anything
```

- 코드 예 4-14는 코드 예 4-13에 표시된 명령을 지정한 경우 아카이브된 파일을 표시합니다.

코드 예 4-14 아카이브된 파일(코드 예 4-13에 표시된 명령을 가정함)

```

/saml/share/fred.anything
/saml/share/marketing/fred.anything
/saml/share/marketing/first_user/first_user_sub/fred.anything
/saml/fred.anything
/saml/testdir/fred.anything
/saml/testdir/share/fred.anything
/saml/testdir/share/marketing/fred.anything
/saml/testdir/share/marketing/second_user/fred.anything
/saml/testdir/share/marketing/second_user/sec_user_sub/fred.any

```

릴리스 및 스테이지 파일속성: -release 및 -stage

각각 -release 및 -stage 옵션을 사용하여 아카이브 세트 내에서 파일과 관련된 릴리스 및 스테이지 속성을 설정할 수 있습니다. 이 설정은 모두 이전에 설정한 스테이지 또는 릴리스 속성보다 우선합니다.

-release 옵션의 형식은 다음과 같습니다.

```
-release attributes
```

-release 명령의 *attributes*는 `releaserelease(1)` 명령 및 표 4-11의 내용과 동일한 규칙을 따릅니다.

표 4-11 -release 옵션

속성	의미
a	첫 번째 아카이브 복사 완료 후에 파일을 릴리스합니다.
d	기본값으로 재설정합니다.
n	파일을 릴리스하지 않습니다.
p	파일의 디스크 공간을 부분적으로 릴리스합니다.

-stage 옵션의 형식은 다음과 같습니다.

```
-stage attributes
```

-stage 명령의 *attributes*는 stage(1) 명령 및 표 4-12의 내용과 동일한 규칙을 따릅니다.

표 4-12 -stage 명령의 *attributes*

속성	의미
a	이 아카이브 세트의 파일을 연관 스테이지합니다.
d	기본값으로 재설정합니다.
n	이 아카이브 세트의 파일을 절대 스테이지하지 않습니다.

다음 예제에서는 파일 이름 지정과 파일 속성을 사용해 Macintosh 리소스 디렉토리를 부분적으로 릴리스하는 방법을 보여줍니다.

```
MACS . -name .*/\.\rscs/ -release p
```

아카이브 세트 구성원 충돌

경우에 따라서는 아카이브 세트의 파일에 포함시킬 경로 및 기타 파일 특성을 선택하면 아카이브 세트 구성원이 모호해질 수 있습니다. 이러한 경우에는 다음과 같은 방법으로 문제를 해결합니다.

1. 아카이브 세트의 첫 번째 구성원 정의를 선택합니다.
2. 전역으로 정의된 정의에 앞서 파일 시스템에 국한된 구성원 정의를 선택합니다.
3. 이전 정의와 정확히 일치하는 구성원 정의는 오류로 표시됩니다.

이러한 규칙의 결과는 제한된 구성원 정의는 명령 파일의 앞에 두어야 한다는 것입니다.

(*fs=fsname* 명령을 사용하여) 특정 파일 시스템에 대해 아카이브를 제어할 때 아카이버는 전역 명령을 검사하기 전에 파일 시스템 특정 명령을 검사합니다. 그러므로 파일은 전역 아카이브에 할당되지 않고 로컬 아카이브 세트(*no_archive* 아카이브 세트 포함)에 할당될 수 있습니다. 이는 *no_archive*와 같은 전역 아카이브 세트 할당을 설정할 때 의미가 있습니다.

코드 예 4-15는 *archiver.cmd* 파일을 표시합니다.

코드 예 4-15 구성원 충돌 가능성이 있는 *archiver.cmd* 파일

```
no_archive . -name .*\.\o$
fs = samfs1
    allfiles .
fs = samfs2
    allfiles .
```

읽기 코드 예 4-15에서 관리자는 두 파일 시스템에서 .o 파일을 아카이브할 생각이 없었습니다. 그러나 로컬 아카이브 세트 할당 allfiles이 전역 아카이브 세트 할당 no_archive에 앞서 실행되기 때문에 samfs1 및 samfs2 파일 시스템의 .o 파일은 아카이브됩니다.

코드 예 4-16에 표시된 명령을 사용하면 두 파일 시스템 모두에 아카이브된 .o 파일이 없음을 알 수 있습니다.

코드 예 4-16 수정된 archiver.cmd 파일

```
fs = samfs1
    no_archive . -name .*\.o$
    allfiles .
fs = samfs2
    no_archive . -name .*\.o$
    allfiles .
```

아카이브 복사 명령

아카이브 복사본을 지정하지 않으면 아카이버는 아카이브 세트의 파일에 대해 1개의 복사본을 작성합니다. 기본적으로 이 복사본은 파일의 아카이브 나이가 4분이 될 때 생성됩니다. 아카이브 복사본이 2개 이상 필요한 경우에는 아카이브 복사 명령을 사용해 첫 번째 복사본을 포함한 모든 복사본을 지정해야 합니다.

아카이브 복사 명령은 정수인 *copy_number*로 시작합니다. 이 숫자(1, 2, 3 또는 4)는 복사본 번호입니다. 숫자 다음에는 해당 복사본의 아카이브 특성을 지정하는 1개 이상의 인수가 옵니다.

아카이브 복사 명령은 자신이 속한 아카이브 세트 할당 명령 바로 다음에 나타나야 합니다. 각 아카이브 복사 명령의 형식은 다음과 같습니다.

```
copy_number [ -release | -norelease ] [archive_age] [unarchive_age]
```

다음 섹션에서는 아카이브 복사 명령 인수에 대해 설명합니다.

아카이브 이후 디스크 공간 릴리스 -release

복사본 번호 다음에 -release 명령을 사용해 아카이브 복사본을 만든 후에 파일의 디스크 공간이 자동으로 릴리스되도록 지정할 수 있습니다. 이 옵션의 형식은 다음과 같습니다.

```
-release
```

코드 예 4-17에서 그룹이 images인 파일은 아카이브 나이가 10분이 되면 아카이브됩니다. 아카이브 복사본 1이 생성되면 디스크 캐시 공간이 릴리스 됩니다.

코드 예 4-17 -release 명령을 사용하는 archiver.cmd 파일

```
ex_set . -group images
      1 -release 10m
```

디스크 공간 릴리스 지연: -norelease

여러 아카이브 복사본이 완료될 때까지 디스크 공간 릴리스를 지연할 수 있습니다. -norelease 옵션을 사용하면 -norelease가 표시된 복사본이 모두 생성될 때까지 디스크 캐시가 자동으로 릴리스 되지 않습니다. 이 옵션의 형식은 다음과 같습니다.

```
-norelease
```

코드 예 4-18은 일명 vault_tapes 아카이브 세트를 지정합니다. 2개의 복사본이 생성되지만, 이 아카이브 세트와 연결된 디스크 캐시는 2개의 복사본이 모두 생성된 후에 릴리스됩니다. 이 방법은 외부 저장소 볼륨을 만들기 전에 파일에 온라인 액세스해야 하는 사이트에서 사용할 수 있습니다.

코드 예 4-18 -norelease 명령을 사용하는 archiver.cmd 파일

```
vault_tapes
      1 -norelease 10m
      2 -norelease 30d
```

파일은 아카이브 복사본이 적어도 1개 이상 있어야 릴리스될 수 있기 때문에 1개의 복사본에 -norelease를 지정해도 자동 릴리스에 영향을 주지 않습니다. 또한 -norelease 및 -release 지정은 함께 사용할 수 없습니다.

아카이브 나이 설정

명령의 다음 필드로 아카이브 나이를 지정하면 파일에 아카이브 나이를 설정할 수 있습니다. 아카이브 나이는 시간의 경우 n 또는 분의 경우 m와 같은 접미사를 사용하여 지정될 수 있습니다. 67 페이지의 표 4-2, “archiver.cmd 파일 명령 단위,”는 전체 접미사 및 해당 의미 목록을 표시합니다.

코드 예 4-19에서는 아카이브 나이가 1시간이 될 때 디렉토리 data의 파일이 아카이브됩니다.

코드 예 4-19 아카이브 나이를 지정하는 archiver.cmd 파일

```
ex_set data
  1 1h
```

자동 아카이브 해제

파일의 아카이브 복사본을 2개 이상 지정하면 복사본 중 하나를 제외한 모두를 자동으로 아카이브 해제할 수 있습니다. 이러한 기능은 다양한 아카이브 나이를 사용해 다양한 매체에 파일을 아카이브하는 경우에 사용할 수도 있습니다.

코드 예 4-20은 아카이브 해제 나이를 지정하는 명령을 표시합니다.

코드 예 4-20 아카이브 해제 나이를 지정하는 archiver.cmd 파일

```
ex_set home/users
  1 6m 10w
  2 10w
  3 10w
```

home/users에 있는 파일의 첫 번째 복사본은 수정 후 6분이 지나면 아카이브됩니다. 파일의 나이가 10주가 되면 두 번째 및 세 번째 아카이브 복사본이 생성됩니다. 그리고 나서, 첫 번째 복사본이 아카이브 해제됩니다.

아카이브 해제를 제어하는 추가적인 방법에 대해서는 98 페이지의 "아카이브 해제 제어"를 참조하십시오.

메타 데이터에 대해 2개 이상의 복사본 지정

메타데이터의 복사본이 둘 이상 필요한 경우 fs= 명령 후 즉시 명령 파일에 복사본 정의를 둘 수 있습니다.

코드 예 4-21은 여러 메타데이터 복사본을 지정하는 archiver.cmd 파일을 표시합니다.

코드 예 4-21 여러 메타데이터 복사본을 지정하는 archiver.cmd 파일

```
fs = samfs7
  1 4h
  2 12h
```

이 예제에서는 samfs7 파일 시스템에 대한 메타 데이터의 복사본 1이 4시간 후에 생성되고 두 번째 복사본이 12 시간 후에 생성됩니다.

파일 시스템 메타 데이터에는 파일 시스템의 경로 이름 변경이 포함됩니다. 이러한 이유 때문에 디렉토리를 자주 변경하면 새 아카이브 복사본이 생성됩니다. 이렇게 되면 메타 데이터에 지정된 볼륨이 자주 로드됩니다.

아카이브 설정 복사 매개변수

archiver.cmd 파일의 아카이브 설정 매개변수 절은 params 명령으로 시작하여 endparams 명령으로 끝납니다. 코드 예 4-22는 아카이브 세트에 대한 명령 형식을 표시합니다.

코드 예 4-22 아카이브 설정 복사 매개변수 형식

```
Params
archive_set_name.copy_number[R] [ -param1 -param2 ...]
.
.
.
endparams
```

표 4-13 아카이브 세트 복사 매개변수에 대한 인수

인수	의미
<i>archive_set_name</i>	아카이브 세트의 사이트 정의 이름. 대개 아카이브 세트에 속하는 파일의 특성을 표시합니다. allsets일 수 있습니다. 아카이브 세트 이름은 알파벳, 숫자 및 밑줄 문자(_)로 제한됩니다. 기타 특수 문자 또는 공백은 사용할 수 없습니다. 아카이브 세트 이름의 첫 번째 문자는 알파벳이어야 합니다.
.	마침표(.). copy_number에서 archive_set_name을 분리하는 데 사용됩니다.

표 4-13 아카이브 세트 복사 매개변수에 대한 인수 (계속)

인수	의미
<i>copy_number</i>	아카이브 복사본 번호를 정의하는 정수. 1, 2, 3 또는 4일 수 있습니다.
R	정의되는 매개변수를 이 아카이브 세트의 재아카이브된 복사본에 대한 것으로 지정합니다. 예를 들어, R을 사용하고 <i>-param1</i> 매개변수의 VSN를 지정하여 특정 볼륨에 재아카이브된 복사본을 지시할 수 있습니다.
<i>-param1</i> <i>-param2</i>	하나 이상의 매개변수. 다음 하위 절은 <i>params</i> 및 <i>endparams</i> 명령 사이에서 지정될 수 있는 매개변수를 설명합니다.

가상 아카이브 세트 *allsets*는 모든 아카이브 세트에 대해 기본 아카이브 세트 명령을 설정하는 방법을 제공합니다. 모든 *allsets* 명령은 실제 아카이브 세트 복사본의 명령보다 선행해야 합니다. 각각의 아카이브 세트 복사본에 대해 설정된 매개 변수는 *allsets* 명령에 의해 설정된 매개 변수보다 우선합니다. *allsets* 아카이브 세트에 대한 자세한 내용은 *archiver.cmd(4)* 매뉴얼 페이지를 참조하십시오.

-disk_archive 매개 변수를 제외한 모든 아카이브 세트 처리 매개 변수에 대해서는 이 섹션에서 설명합니다. *-disk_archive* 매개 변수에 대한 자세한 내용은 109 페이지의 "디스크 아카이브"를 참조하십시오.

아카이브 파일의 크기 제어: *-archmax*

-archmax 명령은 아카이브 세트에 대해 최대 파일 크기를 설정합니다. 형식은 다음과 같습니다.

```
-archmax target_size
```

이 명령은 *archmax* 전역 명령과 아주 유사합니다. 이 명령 및 *target_size*에 입력할 값에 대한 정보는 71 페이지의 "archmax 명령: 아카이브 파일의 크기 제어"를 참조하십시오.

아카이버 버퍼 크기 설정: *-bufsize*

기본적으로 아카이브되는 파일은 아카이브 매체에 파일을 쓰기 전에 버퍼의 메모리에 저장됩니다. *-bufsize* 매개 변수를 사용하면 기본값이 아닌 버퍼 크기를 지정할 수 있습니다. 이렇게 하면 성능을 향상시킬 수 있을뿐만 아니라 다양한 *buffer_size* 값을 시험해 볼 수 있습니다.

이 매개 변수의 형식은 다음과 같습니다.

```
-bufsize=buffer_size
```

*buffer_size*에 대해 2부터 32까지 숫자를 지정합니다. 기본값은 4입니다. 이 값에 매체 유형에 대한 *dev_blksize* 값이 곱해지고 해당 결과 버퍼 크기가 사용됩니다. *dev_blksize*는 `defaults.conf` 파일에 지정됩니다. 이 파일에 대한 자세한 내용은 `defaults.conf(4)` 매뉴얼 페이지를 참조하십시오.

예를 들어 이 매개 변수는 `archiver.cmd` 파일에 다음과 같은 행으로 지정할 수 있습니다.

```
mysset.1 -bufsize=6
```

또한 `bufsize=media buffer_size` 명령을 지정해도 이 명령과 같은 명령을 전역으로 지정할 수 있습니다. 이 주제에 대한 자세한 내용은 72 페이지의 "bufsize 명령: 아카이브 버퍼 크기 설정"을 참조하십시오.

아카이브 요청에 대한 드라이브 수 지정: `-drivemax`, `-drivemin` 및 `-drives`

기본적으로 아카이버는 아카이브 세트 파일 하나를 아카이브하기 위해 매체 드라이브를 1개만 사용합니다. 아카이브 세트의 파일이 많거나 용량이 큰 경우에는 드라이브를 2개 이상 사용하는 것이 유리할 수 있습니다. 추가로 자동화 라이브러리의 드라이브들이 서로 다른 속도로 동작하는 경우 이 명령을 사용하여 아카이브 효율성을 향상시킬 수 있습니다.

코드 예 4-23은 드라이브에 대해 아카이브 요청을 분할하고 테이프 드라이브 전송 속도의 변화가 균형을 유지하는 데 사용할 수 있는 매개변수를 표시합니다.

코드 예 4-23 `-drivemax`, `-drivemin` 및 `-drives` 명령에 대한 형식

```
-drivemax max_size  
-drivemin min_size  
-drives number
```

표 4-14 -drivemax, -drivemin 및 -drives 매개변수에 대한 인수

인수	의미
<i>maxsize</i>	드라이브 하나를 사용하여 아카이브된 최대량의 데이터.
<i>minsize</i>	드라이브 하나를 사용하여 아카이브된 최소량의 데이터 기본값은 <code>-archmax target_size</code> 값(지정된 경우) 또는 매체 유형에 대한 기본값입니다. -drivemin <i>minsize</i> 를 지정한 경우, Sun StorEdge SAM-FS는 보증할 작업이 충분한 경우에만 여러 드라이브를 사용합니다. 지침으로, 카트리지 변경 시간(로드, 위치, 언로드)보다 전송 시간이 훨씬 더 길어지도록 크게 <i>최소크기</i> 를 설정할 수 있습니다.
<i>number</i>	이 아카이브 세트를 아카이브할 때 사용되는 드라이브의 수. 기본 값은 1입니다.

아카이브 요청은 다음과 같이 지정된 매개변수에 대해 검사됩니다.

- 아카이브 요청이 *min_size* 보다 작으면 1개의 드라이브만을 사용해 아카이브 요청을 작성합니다.
- 아카이브 요청이 *min_size* 보다 크면 아카이브 요청은 *min_size*에 대해 실행되고 적정 드라이브 수는 지정된 전체 드라이브 수로 예약됩니다.
- *min_size*가 0이면 지정된 전체 수의 드라이브 간에 분할을 시도합니다.

-drives 매개 변수를 사용하면 *min_size* 보다 큰 데이터가 한 번에 아카이브되는 경우에만 여러 드라이브를 사용합니다. 동시에 사용할 드라이브 수는 *arch_req_total_size/min_size* 와 -drives 매개 변수에 의해 지정된 드라이브 수 중에 작은 쪽입니다.

이 매개변수를 설정하려면 사용자는 작성 속도, 드라이브의 수, 드라이브 로드 및 언로드에 소요되는 시간 및 드라이브 전송 속도를 고려해야 합니다.

예제 1. 드라이브 간에 아카이브 요청을 분할하고자 하나 모든 드라이브를 작은 아카이브 요청으로 묶지 않으려는 경우에 -drivemin 및 -drives 매개 변수를 사용할 수 있습니다. 이러한 방법은 매우 큰 파일을 사용하는 작업에 적용할 수 있습니다.

예제 2. 예를 들어 `big_files`로 명명된 아카이브 세트를 5개의 드라이브에서 분할한다고 가정합니다. 이 아카이브 세트는 그 크기에 따라 표 4-15에서와 같이 분할될 수 있습니다.

표 4-15 아카이브 세트 예제 분할

아카이브 세트 크기	드라이브 수
< 20 기가바이트	1
≥ 20 기가바이트에서 < 30 기가바이트	2
≥ 30 기가바이트에서 < 40 기가바이트	3
≥ 40 기가바이트에서 < 50 기가바이트	4
≥ 50 기가바이트	5

코드 예 4-24는 여러 드라이브에 대해 아카이브 요청을 분할하기 위해 `archiver.cmd` 파일에서 사용하는 행을 표시합니다.

코드 예 4-24 여러 드라이브에 대해 아카이브 요청을 분할하기 위해 사용되는 명령

```
Params
bigfiles.1 -drives 5 -drivemin 10G
endparams
```

예제 3. 다음 행은 `archiver.cmd` 파일에 지정됩니다.

```
huge_files.2 -drives 2
```

아카이브 세트 `huge_files.2`에 있는 파일의 총 크기가 매체에 대한 `drivemin`의 2배 이상이면 파일을 아카이브하는 데 2개의 드라이브를 사용합니다.

예제 4. 아카이브 요청은 300 기가바이트입니다. 다음 행은 5 개의 각 드라이브에서 한번에 10 기가바이트를 아카이브하도록 `archiver.cmd` 파일에 지정됩니다.

```
-drives 5 -drivemax 10G
```

볼륨 공간 최소화하기: `-fillvsns`

기본적으로 아카이버는 아카이브 복사본을 기록할 때 아카이브 세트에 할당된 모든 볼륨을 사용합니다. 아카이브 복사본을 기록할 때 아카이버는 모든 파일에 대해 공간이 충분한 볼륨을 선택합니다. 이 작업으로 볼륨이 용량을 채우지 못할 수 있습니다. `-fillvsns`이 지정되면 아카이버는 아카이브 요청을 더 작은 그룹으로 나눕니다.

아카이브 버퍼 잠금 지정 -lock

기본적으로 아카이브되는 파일은 아카이브 매체에 파일을 쓰기 전에 버퍼의 메모리에 저장됩니다. 직접 입출력을 사용하는 경우에는 -lock 매개 변수를 사용해 이 버퍼를 잠글 수 있습니다. 이렇게 하면 성능을 향상시킬 수 있을뿐만 아니라 이 매개 변수를 시험해 볼 수 있습니다.

이 매개 변수의 형식은 다음과 같습니다.

```
-lock
```

-lock 매개 변수는 아카이버가 아카이브 복사본을 만들 때 잠긴 버퍼를 사용하는지 여부를 나타냅니다. -lock이 지정되면 아카이버가 sam-arcopy(1M) 작업 과정에서 메모리의 아카이브 버퍼에 파일 잠금을 설정합니다. 이렇게 하면 버퍼의 페이지 이동이 일어나지 않을뿐만 아니라 성능을 향상시킬 수 있습니다.

-lock 매개 변수는 메모리 용량이 큰 대형 시스템에 대해서만 지정해야 합니다. 메모리가 충분하지 않으면 메모리 부족 현상이 발생할 수 있습니다.

-lock 매개 변수는 아카이브되는 파일에 대해 직접 입출력을 사용하는 경우에만 효과적으로 사용할 수 있습니다. 기본적으로 -lock은 지정되지 않고 파일 시스템은 아카이브 대상을 포함한 모든 직접 입출력 버퍼에 대해 잠금을 설정합니다. 직접 입출력 사용에 대한 자세한 내용은 setfa(1) 매뉴얼 페이지, sam_setfa(3) 라이브러리 루틴 매뉴얼 페이지 또는 mount_samfs(1M) 매뉴얼 페이지의 -O forcedirectio 옵션을 참조하십시오.

예를 들어 이 매개 변수는 archiver.cmd 파일에 다음과 같은 행으로 지정할 수 있습니다.

```
yourset.3 -lock
```

bufsize=*media buffer_size* [lock] 명령에 lock 인수를 지정해도 이 매개 변수와 같은 매개 변수를 지정할 수 있습니다. 이 주제에 대한 자세한 내용은 72 페이지의 "bufsize 명령: 아카이버 버퍼 크기 설정"을 참조하십시오.

오프라인 파일의 아카이브 복사본 만들기: `-offline_copy`

파일은 아카이브 복사본 하나가 작성된 후 릴리스되는 대상입니다. 아카이브 복사본 모두가 작성되기 전에 파일이 릴리스되고 오프라인으로 나가는 경우, 아카이버는 이 매개 변수를 사용하여 다른 아카이브 복사본을 만들 때 사용할 방법을 결정합니다. 사용할 *method*를 선택할 때 Sun SAM-FS 시스템 및 사용 가능한 많은 디스크 캐시에 사용할 수 있는 드라이브의 수를 고려하십시오. 이 매개 변수의 형식은 다음과 같습니다.

`-offline_copy method`

*method*에 대해 표시된 키워드 중 하나를 지정합니다.

표 4-16 `-offline_copy` 명령의 *method* 인수에 대한 값

<i>method</i>	의미
none	아카이브 볼륨에 복사하기 전에 각 파일에 대해 필요한 경우 파일을 스테이지합니다. 기본값입니다.
direct	오프라인 볼륨에서 아카이브 볼륨으로 캐시를 사용하지 않고 직접 파일을 복사합니다. 이 방법은 원본 볼륨 및 대상 볼륨이 다른 볼륨이며 드라이브 두 개를 사용할 수 있다고 가정합니다. 이 방법이 지정되면 <code>stage_n_window</code> 마운트 옵션 값을 기본값 256 KB보다 높게 올립니다. 사용 가능한 마운트 매개 변수에 대한 자세한 내용은 <code>mount_samfs(1M)</code> 매뉴얼 페이지를 참조하십시오.
stageahead	또 다른 파일을 아카이브하는 동안 파일 하나를 스테이지합니다. 지정되면 파일을 대상에 기록하는 동안 시스템이 다음 아카이브 파일을 스테이지합니다.
stageall	아카이브하기 전에 디스크 캐시에 모든 파일을 스테이지합니다. 이 방법은 드라이브를 하나만 사용하며 디스크 캐시에 모든 파일에 대해 사용할 수 있는 공간이 있다고 가정합니다.

리사이클 지정

리사이클 프로세스를 사용하면 종료된 아카이브 이미지가 차지하는 아카이브 볼륨의 공간을 회수할 수 있습니다. 기본적으로 리사이클이 발생하지 않습니다.

리사이클하려는 경우, `archiver.cmd` 및 `recycler.cmd` 두 파일 모두에 명령을 지정할 수 있습니다. `archiver.cmd` 파일에서 사용할 수 있는 리사이클 명령에 대한 자세한 내용은 167 페이지의 "리사이클"을 참조하십시오.

연관 아카이브 -join

아카이버는 `-join path` 매개변수를 지정하는 경우 연관 아카이브를 사용합니다. 연관 아카이브는 전체 디렉토리를 1개의 볼륨에 아카이브하려는 경우와 아카이브 파일이 1개의 볼륨에 물리적으로 상주할 수 있는 경우에 유용하게 사용할 수 있습니다. 그렇지 않고 디렉토리를 함께 유지하려는 경우에는 `-sort path` 또는 `-rsort path` 매개 변수를 사용해 파일이 이어지도록 만듭니다. `-rsort`는 역방향 정렬을 수행합니다.

아카이버가 아카이브 파일을 볼륨에 기록할 경우 해당 볼륨에 사용자 파일을 효과적으로 압축합니다. 이후 동일한 디렉토리에서 파일에 액세스하면 스테이지 프로세스가 다음 파일을 읽기 위해 볼륨에서 이동할 때 시간이 지연될 수 있습니다. 지연 시간을 줄이기 위해 아카이브 파일 내의 동일한 디렉토리 경로에서 파일을 연속적으로 아카이브할 수 있습니다. 연관 아카이브의 프로세스는 동일한 디렉토리에서 파일을 함께 아카이브하는 공간 효율 알고리즘 보다 우선합니다. `-join path` 매개 변수를 사용하면 이러한 파일을 아카이브 세트 복사본 내에서 연속적으로 아카이브할 수 있습니다.

연관 아카이브는 파일 내용은 바뀌지 않지만 파일의 그룹에 항상 동시에 액세스하고자 할 때 유용하게 사용할 수 있습니다. 예를 들면 병원에서 의료용 이미지에 액세스하기 위해 연관 아카이브를 사용할 수 있습니다. 동일한 환자에 관련된 이미지를 1개의 디렉토리에 유지할 수 있으며, 의사는 이러한 이미지에 동시에 액세스할 수 있습니다. 이러한 정적 이미지는 디렉토리 위치에 따라 연속적으로 아카이브하면 보다 효과적으로 액세스할 수 있습니다. 예를 들어, 다음과 같습니다.

```
patient_images.1 -join path
```

참고 - `-join path` 매개 변수는 동일한 디렉토리의 데이터 파일을 동일한 아카이브 파일에 작성합니다. 몇 개의 작은 파일이 포함된 디렉토리가 여러 개 있으면 아카이버는 작은 아카이브 파일을 여러 개 만듭니다. 이렇게 작고 분리된 아카이브 파일은 각 아카이브 파일에 대한 `tar(1)` 헤더에 비해 데이터 파일이 상대적으로 작기 때문에 시스템의 쓰기 성능을 저하시킵니다. 이로 인해 고속 테이프 드라이브에 기록할 때 성능이 저하될 수 있습니다.

또한 `-join path` 매개변수가 동일한 디렉토리의 모든 파일이 단일 볼륨에 아카이브 되도록 지정하기 때문에 파일 그룹이 사용 가능한 볼륨에 적합하지 않을 수도 있습니다. 이러한 경우에는 더 많은 볼륨이 아카이브 세트에 할당된 후에야 파일이 아카이브됩니다. 또한 아카이브할 파일 그룹이 너무 커서 1개의 볼륨에 맞지 않을 수도 있습니다. 이러한 경우 파일이 아카이브되지 않습니다.

대부분의 애플리케이션에 대해, `-join path`의 동작을 더 제한할 필요가 없는 경우 `-sort path` 또는 `-join path` 매개변수 중 하나를 사용하는 것이 좋습니다.

또한 아카이브 세트 복사본 내의 파일을 나이, 크기 또는 경로별로 정렬할 수도 있습니다. age 및 size 인수는 함께 사용할 수 없습니다. 코드 예 4-25는 age 또는 size 인수와 -sort 매개변수를 함께 사용하여 아카이브 세트를 정렬하는 방법을 표시합니다.

코드 예 4-25 아카이브 세트 정렬 명령

```
cardiac.1 -sort path
cardiac.2 -sort age
catscans.3 -sort size
```

첫 번째 행은 아카이버가 아카이브 요청을 경로 이름별로 정렬하도록 만듭니다. 두 번째 행은 아카이버가 cardiac.2라고 하는 아카이브 세트 복사본을 파일의 나이별로 적은 나이부터 정렬하도록 만듭니다. 세 번째 행은 catscans라고 하는 아카이브 세트 복사본을 큰 파일부터 크기별로 정렬하도록 만듭니다. 역방향 정렬을 원하는 경우 -sort 자리에 -rsort를 지정해야 합니다.

아카이브 해제 제어

*언아카이브*는 파일 또는 디렉토리에 대한 아카이브 항목이 삭제되는 프로세스입니다. 기본적으로 파일은 아카이브 해제되지 않습니다. 파일은 마지막 액세스 후의 시간을 기준으로 아카이브 해제됩니다. 자주 액세스하는 데이터는 디스크와 같은 고속 매체에 저장하고, 드물게 액세스하는 데이터는 테이프에 저장할 수 있습니다.

예제 1. 코드 예 4-26은 archiver.cmd 파일을 표시합니다.

코드 예 4-26 아카이브 해제를 제어하는 명령

```
arset1 dir1
  1 10m   60d
  2 10m
  3 10m
vsns
arset1.1 mo OPT00[0-9]
arset1.2 lt DLTA0[0-9]
arset1.3 lt DLTB0[0-9]
```

코드 예 4-26에 표시된 archiver.cmd 파일에 의해 제어되는 파일은 자주 액세스하면 나이가 60일 보다 많더라도 항상 디스크에 상주합니다. 복사본 1 정보는 파일에 60일간 액세스하지 않은 경우에만 제거됩니다.

복사본 1 정보는 60일간 파일에 액세스되지 않아서 제거되고 누군가가 복사본 2에서 파일을 스테이지하면 테이프에서 읽기를 수행합니다. 파일이 다시 온라인이되면 아카이버는 새로운 복사본 1을 디스크에 만들고 60일 간의 액세스 주기는 처음부터 다시 시작됩니다. Sun StorEdge SAM-FS 아카이버는 파일에 다시 액세스하면 새 복사본 1을 다시 생성합니다.

예제 2

한 환자가 4주간 병원에 입원하고 있다고 가정합니다. 이 시간 동안 이 환자의 파일은 모두 고속 매체에 저장됩니다(복사본 1=mo). 환자는 4주 후에 퇴원합니다. 환자를 릴리스한 후에 60일 동안 이 환자에 대한 데이터에 액세스하지 않은 경우, inode의 복사본 1 항목은 아카이브되지 않고 복사본 2와 3 항목만 사용할 수 있습니다. 이제 볼륨을 리사이클해 디스크 라이브러리를 늘릴 필요없이 더 많은 환자를 위한 공간을 만들 수 있습니다. 환자가 6개월 후에 건강 진단을 위해 다시 병원을 찾은 경우, 데이터의 첫 번째 액세스는 테이프(복사본 2)에서 시작합니다. 이제 아카이버는 몇 일 또는 몇 개월이 소요될 수 있는 건강 진단 과정에서 데이터가 다시 고속 매체에 저장되도록 디스크에 자동으로 새 복사본 1을 만듭니다.

아카이브 파일의 작성 방식 제어 -tapenonstop

기본적으로 아카이버는 테이프 마크와 EOF 레이블을 작성하고 아카이브 파일 사이에 테이프 마크를 2개 더 작성합니다. 다음 아카이브 파일이 시작되면 드라이버는 첫 번째 테이프 마크 이후의 위치로 백업해 성능이 떨어집니다. -tapenonstop 매개 변수는 아카이버가 초기 테이프 마크만 작성하도록 지시합니다. 이외에도 -tapenonstop 매개 변수가 지정된 경우, 아카이버는 복사 작업의 끝에 아카이브 정보를 입력합니다.

-tapenonstop 매개 변수에 대한 자세한 내용은 archiver.cmd(4) 매뉴얼 페이지를 참조하십시오.

볼륨 예약 -reserve

기본적으로 아카이버는 archiver.cmd 파일의 볼륨 연결 섹션에서 설명한 것처럼 일반 표현식에 의해 지정된 볼륨에 아카이브 세트 복사본을 작성합니다. 그러나 경우에 따라서는 아카이브 세트 볼륨에 1개의 아카이브 세트의 파일만 포함되는 것이 바람직합니다. 볼륨 예약 프로세스를 사용하면 이 데이터 저장 요구 사항에 부합할 수 있습니다.

참고 - -reserve 매개 변수는 1개의 아카이브 세트만이 사용하는 볼륨을 예약합니다. 예약된 볼륨을 사용하는 사이트는 더 많은 카트리지 로드 및 언로드를 야기할 수 있습니다.

-reserve 매개 변수는 아카이브 세트에 볼륨을 예약합니다. -reserve 매개 변수가 설정되고 볼륨이 아카이브 세트 복사본에 할당되면 볼륨 식별자는 일반 표현식과 일치하지 않더라도 다른 아카이브 세트 복사본에 할당되지 않습니다.

아카이브 세트가 사용할 볼륨이 선택되면 볼륨에는 예약된 이름이 지정됩니다. 예약된 이름은 아카이브 세트와 볼륨을 연결하는 고유 식별자입니다.

-reserve 매개 변수의 형식은 다음과 같습니다.

```
-reserve 키워드
```

지정되는 키워드는 사용하는 형식에 따라 다릅니다. 다음과 같이 아카이브 세트 형식, 소유자 형식 및 파일 시스템 형식 등을 사용할 수 있습니다.

- 아카이브 세트 형식. 이 형식은 다음과 같이 설정 키워드를 사용합니다. -reserve set.
- 소유자 형식. 이 형식은 다음 키워드 중 하나를 사용합니다. dir, user, 또는 group. 코드 예 4-27은 이러한 명령에 대한 형식을 표시합니다.

코드 예 4-27 -reserve 매개변수에 대한 소유자 형식

```
-reserve dir  
-reserve user  
-reserve group
```

코드 예 4-27에 표시된 3가지 소유자 형식은 함께 사용할 수 없습니다. 즉, 3가지 소유자 형식 중 1가지만 아카이브 세트 및 복사본에 사용할 수 있습니다.

- 파일 시스템 형식. 이 형식은 다음과 같이 fs 키워드를 사용합니다. -reserve fs.

archiver.cmd 파일에서는 1가지, 2가지 또는 3가지 형식에 대한 -reserve 매개 변수를 지정할 수 있습니다. 3가지 형식은 합쳐서 아카이브 세트 매개 변수 정의에서 함께 사용할 수 있습니다.

예를 들어, 코드 예 4-28은 archiver.cmd 파일의 일부를 표시합니다. arset.1로 시작하는 행은 아카이브 세트, 그룹 및 파일 시스템을 기본으로 한 예약된 이름을 작성합니다.

코드 예 4-28 예약된 볼륨이 있는 archiver.cmd 파일

```
Params  
arset.1 -reserve set -reserve group -reserve fs  
endparams
```

예약된 볼륨에 관한 정보는 라이브러리 카탈로그에 저장됩니다. 라이브러리 카탈로그 내의 행에는 매체 유형, VSN, 예약 정보 및 예약 날짜와 시간이 포함됩니다. 예약 정보에는 슬래시(/)로 구분된 아카이브 세트 구성 요소, 경로 이름 구성 요소 및 파일 시스템 구성 요소 등이 포함됩니다.

이러한 슬래시는 경로 이름을 나타내지 *않으며* 단지 예약된 이름의 3가지 구성 요소를 표시하기 위한 구분 기호에 지나지 않습니다. 코드 예 4-29에 표시된 대로 예약된 볼륨을 설명하는 행은 라이브러리 카탈로그에서 #R 문자로 시작합니다.

코드 예 4-29 예약된 볼륨을 표시하는 라이브러리 카탈로그

```

6 00071 00071 lt 0xe8fe 12 9971464 1352412 0x6a000000 131072 0x
# -il-o-b----- 05/24/00 13:50:02 12/31/69 18:00:00 07/13/01 14:03:00
#R lt 00071 arset0.3// 2001/03/19 18:27:31
10 ST0001 NO_BAR_CODE lt 0x2741 9 9968052 8537448 0x68000000 1310
# -il-o----- 05/07/00 15:30:29 12/31/69 18:00:00 04/13/01 13:46:54
#R lt ST0001 hgm1.1// 2001/03/20 17:53:06
16 SLOT22 NO_BAR_CODE lt 0x76ba 6 9972252 9972252 0x68000000 1310
# -il-o----- 06/06/00 16:03:05 12/31/69 18:00:00 07/12/01 11:02:05
#R lt SLOT22 arset0.2// 2001/03/02 12:11:25

```

코드 예 4-29에서 일부 행은 지면 관계상 생략하였습니다.

1개 이상의 예약 정보 필드는 archiver.cmd 파일에 정의된 옵션에 따라 비울 수 있습니다. 날짜와 시간은 예약이 이루어진 시간을 나타냅니다. 예약 행은 아카이브 과정에서 아카이브 세트에 예약된 각 볼륨의 파일에 추가됩니다.

samu(1M) 유틸리티의 v 디스플레이를 사용하거나 코드 예 4-30에 표시된 형식 중 하나에서 archiver(1M) 또는 dump_cat(1M) 명령을 사용하면 예약 정보를 표시할 수 있습니다.

코드 예 4-30 예약 정보를 표시하기 위해 사용하는 명령

```

archiver -lv
dump_cat -V catalog_name

```

다음 형식은 볼륨에 지정된 예약된 이름의 매개 변수, 키워드 및 예제를 보여주는 각 형식을 설명합니다.

- 아카이브 세트 형식. 표 4-17에 표시된 대로 set 키워드는 예약된 이름에서 아카이브 세트 구성요소를 활성화합니다.

표 4-17 아카이브 세트 형식 예제

명령 및 키워드	예약된 이름 예제
-reserve set	users.1// Data.1//

예를 들어 코드 예 4-31에서 archiver.cmd 파일 단편에서 allsets 아카이브 세트 이름으로 시작하는 행은 모든 아카이브 세트에 대해 아카이브 세트별로 예약을 설정합니다.

코드 예 4-31 아카이브 세트로 볼륨 예약

```
Params
allsets -reserve set
endparams
```

- 소유자 형식. dir, user 및 group 키워드는 예약된 이름에서 소유자 구성 요소를 실행합니다. dir, user 및 group 키워드는 함께 사용할 수 없습니다. dir 키워드는 아카이브 세트 정의의 경로를 지정한 직후에 디렉토리 경로 구성 요소를 사용합니다. user 및 group 키워드는 설명이 없어도 알 수 있습니다. 표 4-18에서 예제를 보여줍니다.

표 4-18 소유자 세트 형식 예제

명령 및 키워드	예약된 이름 예제
-reserve dir	proj.1/p105/ proj.1/p104/
-reserve user	users.1/user5/ users.1/user4/
-reserve group	data.1/engineering/

참고 - -reserve 매개 변수는 1개의 아카이브 세트가 단독으로 사용하기 위한 볼륨을 예약합니다. 작은 파일이 몇 개 있는 디렉토리가 많아 예약된 각 볼륨에 작은 아카이브 파일 여러 개가 작성됩니다. 이러한 작고 분리된 아카이브 파일은 각 아카이브 파일에 대한 tar(1) 헤더에 비해 데이터 파일이 상대적으로 작기 때문에 시스템의 성능을 저하시킵니다.

- 파일 시스템 형식. fs 키워드는 예약된 이름의 파일 시스템 구성 요소를 실행합니다. 표 4-19에 예제가 나옵니다.

표 4-19 파일 시스템 형식 예제

명령 및 키워드	예약된 이름 예제
-reserve fs	proj.1/p103/samfs1 proj.1/p104/samfs1

124 페이지의 "예제 4"는 예약된 볼륨을 사용하여 전체 아카이브 예제를 표시합니다.

아카이버는 라이브러리 카탈로그 파일에 볼륨 예약을 기록합니다. 볼륨은 아카이브 데이터가 효과적으로 삭제되었기 때문에 레이블을 다시 지정할 때 자동으로 예약이 취소됩니다.

`reserve(1M)` 및 `unreserve(1M)` 명령을 사용하여 볼륨을 예약 및 예약 취소할 수도 있습니다. 이러한 명령에 대한 자세한 내용은 `reserve(1M)` 및 `unreserve(1M)` 매뉴얼 페이지를 참조하십시오.

아카이브 우선 순위 설정 `-priority`

Sun StorEdge SAM-FS 파일 시스템은 파일 아카이브를 위해 구성 가능한 우선 순위 시스템을 제공합니다. 각 파일에는 `archiver.cmd` 파일의 각 아카이브 세트에 대해 설정할 수 있는 파일 및 우선 순위 승수의 등록 정보에서 계산된 우선 순위가 지정됩니다. 등록 정보에는 온라인/오프라인, 나이, 만든 복사본의 수 및 크기 등이 포함됩니다.

기본적으로 아카이브 요청의 파일은 정렬되지 않으며 모든 등록 정보 승수는 0입니다. 이로 인해 파일은 처음 발견되고 처음 아카이브된 순서에 따라 아카이브됩니다. 우선 순위에 대한 자세한 내용은 `archiver(1M)` 및 `archiver.cmd(4)` 매뉴얼 페이지를 참조하십시오.

우선 순위와 정렬 방법을 설정하면 파일이 아카이브되는 순서를 제어할 수 있습니다. 다음은 설정 가능한 우선 순위의 예제입니다.

- `priority` 정렬 방법을 선택해 아카이브 요청 내의 파일을 우선 순위에 따라 아카이브합니다.
- `archive_loaded` 우선 순위를 변경해 매체 로드를 줄입니다.
- `offline` 우선 순위를 변경해 온라인 파일이 오프라인 파일보다 먼저 아카이브되도록 합니다.
- `copy#` 우선 순위를 변경해 복사 순서에 따라 아카이브 복사본을 만듭니다.

표 4-20에서는 아카이브 우선 순위에 대해 설명합니다.

표 4-20 아카이브 우선 순위

아카이브 우선 순위	정의
<code>-priority age value</code>	아카이브 나이 등록 정보 승수
<code>-priority archive_immediate value</code>	아카이브 즉각 등록 정보 승수
<code>-priority archive_overflow value</code>	여러 아카이브 볼륨 등록 정보 승수
<code>-priority archive_loaded</code>	아카이브 볼륨 로드 등록 정보 승수
<code>-priority copy1</code>	복사본 1 등록 정보 승수
<code>-priority copy2 value</code>	복사본 2 등록 정보 승수
<code>-priority copy3 value</code>	복사본 3 등록 정보 승수
<code>-priority copy4 value</code>	복사본 4 등록 정보 승수

표 4-20 아카이브 우선 순위 (계속)

아카이브 우선 순위	정의
-priority copies <i>value</i>	복사본 생성 등록 정보 승수
-priority offline <i>value</i>	파일 오프라인 등록 정보 승수
-priority queuewait <i>value</i>	대기열 대기 등록 정보 승수
-priority rearchive <i>value</i>	재아카이브 등록 정보 승수
-priority reqrelease <i>value</i>	재릴리스 등록 정보 승수
-priority size <i>value</i>	파일 크기 등록 정보 승수
-priority stage_loaded <i>value</i>	스테이지 볼륨 로드 등록 정보 승수
-priority stage_overflow <i>value</i>	여러 스테이지 볼륨 등록 정보 승수

*value*에 대해 다음 범위에 있는 부동 소수점 수를 지정합니다.

$$-3.400000000E+38 \leq \textit{value} \leq 3.402823466E+38$$

아카이브 예약: -startage, -startcount 및 -startsize

아카이버가 파일 시스템을 스캔할 때 아카이브될 파일을 식별합니다. 아카이브 대상으로 인식되는 파일은 *아카이브 요청*으로 알려진 목록에 놓입니다. 파일 시스템 스캔이 끝날 때 시스템은 아카이브를 위해 아카이브 요청을 예약합니다. -startage, -startcount 및 -startsize 아카이브 세트 매개변수는 아카이브 작업 부하를 제어하고 적시에 파일을 아카이브합니다. 표 4-21은 이러한 매개변수의 형식을 제어합니다.

표 4-21 -startage, -startcount 및 -startsize 명령 형식

명령	의미
-startage <i>time</i>	아카이브 요청 및 아카이브 시작에 포함되도록 표시되는 스캔의 첫 번째 파일 사이에 경과할 수 있는 <i>time</i> 을 지정합니다. <i>time</i> 에 대해 89 페이지의 "아카이브 나이 설정"에서 사용된 형식으로 시간을 지정합니다.
-startcount <i>count</i>	아카이브 요청에 포함되는 파일의 수를 지정합니다. 아카이브 요청의 파일 수가 <i>count</i> 에 도달하면 아카이브를 시작합니다. <i>count</i> 에 대해 정수를 지정합니다. 기본적으로 <i>count</i> 은 설정되어 있지 않습니다.
-startsize <i>size</i>	아카이브 요청에 아카이브되는 모든 파일의 최소 전체 크기를 바이트로 지정합니다. 파일의 전체 크기가 <i>size</i> 에 도달할 때 아카이브 작업이 누적되어 아카이브를 시작합니다. 기본적으로 <i>size</i> 는 설정되어 있지 않습니다.

`examine=method` 명령 및 `interval=time` 명령은 `-startage`, `-startcount` 및 `-startsize` 명령과 상호 작동하는 전역 명령입니다. `-startage`, `-startcount` 및 `-startsize` 명령은 아카이브 적시 대 아카이브 작업 완료율을 최적화합니다. 이러한 값은 `examine=method` 사양보다 우선합니다(있는 경우). 자세한 내용은 73 페이지의 "examine 명령: 아카이브 스캔 제어"를 참조하십시오. `interval`에 대한 자세한 내용은 74 페이지의 "interval 명령: 아카이브 간격 지정"을 참조하십시오.

`-startage`, `-startcount` 및 `-startsize` 명령 중 하나만 `archiver.cmd` 파일에 지정될 수 있습니다. 이 명령 중 두 개 이상이 지정되면 첫 번째 발생된 조건이 아카이브 동작을 시작합니다. `-startage`, `-startcount` 또는 `-startsize` 중 어느 것도 지정되지 않은 경우 아카이버 요청은 다음과 같이 `examine=method` 명령을 기본으로 예약됩니다.

- `examine=noscan`의 경우, 아카이브 요청이 첫 번째 파일이 아카이브 요청에 입력된 후 `interval=time` 명령의 사양에 따라 예약됩니다. 이것이 연속 아카이브입니다. 기본적으로 `examine=noscan`입니다.
- `examine=scan` | `scaninodes` | `scandirs`의 경우, 아카이브 요청이 파일 시스템 스캔 후 아카이브를 위해 예약됩니다.

`archiver.cmd(4)` 매뉴얼 페이지에는 이러한 명령을 사용하는 방법을 표시하는 예제가 있습니다.

VSN 연결 명령

`archiver.cmd` 파일의 VSN 연결 절은 아카이브 세트에 볼륨을 할당합니다. 이 섹션은 `vsns` 명령으로 시작해 `endvsns` 명령으로 끝납니다.

볼륨 모음은 다음 형식의 명령에 의해 아카이브 세트에 할당됩니다.

```
archive_set_name .copy_num media_type vsn_expr ... [ -pool vsn_pool_name ... ]
```

표 4-22 VSN 연결 명령에 대한 인수

인수	의미
<i>archive_set_name</i>	아카이브 세트의 사이트 정의 이름. 아카이브 세트 할당 명령에서 첫 번째 필드여야 합니다. 아카이브 세트 이름은 대개 아카이브 세트에 속하는 파일의 특성을 나타냅니다. 아카이브 세트 이름은 알파벳, 숫자 및 밑줄 문자(_)로 제한됩니다. 기타 특수 문자 또는 공백은 사용할 수 없습니다. 아카이브 세트 이름의 첫 번째 문자는 알파벳이어야 합니다.
<i>copy_num</i>	복사본의 아카이브 특성을 지정하는 1개 이상의 인수 앞에 오는 숫자. 아카이브 복사 명령은 숫자로 시작됩니다. 이 숫자(1, 2, 3 또는 4)는 복사본 번호입니다.
<i>media_type</i>	매체 유형. 올바른 매체 유형의 목록에 대해서는 mcf(4) 매뉴얼 페이지를 참조하십시오.
<i>vsn_expr</i>	일반 표현식. regexp(5) 매뉴얼 페이지를 참조하십시오.
<i>-pool vsn_pool_name</i>	VSN이 지정된 모음.

연결에는 최소 세 개의 필드 즉, *archive_set_name*, *copy_number* 및 *media_type*과 최소 하나의 불륨이 필요합니다. *archive_set_name* 및 *copy_number*는 마침표(.)로 연결됩니다.

다음 예제에서는 동일한 VSN을 여러 가지 방식으로 지정합니다.

예제 1. 코드 예 4-32는 VSN 사양의 2개 행을 보여줍니다.

코드 예 4-32 VSN 사양 — 예제 1

```
vsns
set.1 lt VSN001 VSN002 VSN003 VSN004 VSN005
set.1 lt VSN006 VSN007 VSN008 VSN009 VSN010
endvsns
```

예제 2. 코드 예 4-33은 백슬래시(\)를 사용해 다음 라인으로 진행되는 VSN 지정을 보여줍니다.

코드 예 4-33 VSN 사양 — 예제 2

```
vsns
set.1 lt VSN001 VSN002 VSN003 VSN004 VSN005 \
VSN006 VSN007 VSN008 VSN009 VSN010
endvsns
```

예제 3. 코드 예 4-34에서는 속기법으로 일반 표현식을 사용하는 VSN을 지정합니다.

코드 예 4-34 VSN 사양 — 예제 3

```
vsns
set.1 lt VSN0[1-9] VSN10
endvsns
```

볼륨은 `regexp(5)` 매뉴얼 페이지에서 설명한 일반 표현식인 1개 이상의 `vsns_expression` 키워드로 표시합니다. 이러한 일반 표현식은 와일드카드와 동일한 규칙을 따르지 않습니다. 일반 표현식 이외에도 볼륨을 선택하는 VSN 풀을 지정할 수 있습니다. 풀은 VSN 이 연결된 `-pool vsns_pool_name` 명령을 사용해 표현합니다.

아카이버가 아카이브 세트에 대해 볼륨이 필요한 경우, 모든 자동화 라이브러리 및 수동으로 마운트한 장치에서 선택한 매체 유형의 볼륨을 각각 검사하여 VSN 표현식을 만족시키는지 여부를 알아봅니다. 아카이브 복사 작업을 위해 충분한 공간이 포함된 표현식에 적합한 첫 번째 볼륨을 선택합니다. 예를 들어, 다음과 같습니다.

- 다음 명령은 복사본 1의 아카이브 세트 `ex_set`에 속하는 파일이 `optic20 ~ optic39` 볼륨 20개 중 임의의 볼륨을 사용하는 매체 유형 `mo`에 복사되도록 지정합니다.

```
ex_set.1 mo optic[2-3][0-9]
```

- 다음 명령은 복사본 2의 아카이브 세트 `ex_set`에 속하는 파일을 볼륨이 `TAPE`으로 시작하는 매체 유형 `lt`에 복사합니다.

```
ex_set.2 lt ^TAPE
```

Sun StorEdge SAM-FS 환경이 아카이브 세트별로 리사이클 되도록 구성된 경우에는 2개 이상의 아카이브 세트에 VSN을 지정해서는 안 됩니다.

참고 – `archiver.cmd` 파일을 설정할 때는 메타 데이터의 아카이브 세트에 볼륨을 지정해야 합니다. 각 파일 시스템에는 해당 파일 시스템과 이름이 같은 아카이브 세트가 있습니다. 메타 데이터 보존에 대한 자세한 내용은 `samfsdump(1M)` 매뉴얼 페이지 또는 *Sun QFS, Sun SAM-FS 및 Sun SAM-QFS 재난 복구 안내서*를 참조하십시오.

VSN 풀 명령

`archiver.cmd` 파일의 VSN 풀 질은 `vsnpools` 명령으로 시작해서 `endvsnpools` 명령 또는 `archiver.cmd` 파일의 마지막 부분으로 끝납니다. 이 섹션에서는 볼륨 모음을 지정합니다.

VSN 풀은 지정된 볼륨 모음입니다. VSN 풀은 아카이브 세트에 사용할 수 있는 볼륨을 정의할 때 유용하게 사용할 수 있습니다. VSN 풀은 아카이브 세트에 대한 볼륨 지정 및 예약에 유용한 버퍼를 제공합니다.

VSN 풀을 사용하면 조직 내 부서, 그룹 내 사용자, 데이터 유형 및 기타 그룹에서 사용할 수 있는 별도의 볼륨 그룹을 정의할 수 있습니다. 풀에는 이름, 매체 유형 및 볼륨 세트가 지정됩니다. *스크래치 풀*은 VSN 연결의 특정 볼륨이나 다른 VSN 풀을 모두 소모했을 때 사용하는 볼륨 세트입니다. VSN 연결에 대한 자세한 내용은 105 페이지의 "VSN 연결 명령"을 참조하십시오.

볼륨이 예약된 경우에는 이 볼륨이 원래 속했던 풀에서 더 이상 이를 사용할 수 없습니다. 따라서 지정된 풀 내의 볼륨 수는 볼륨을 사용함에 따라 바뀝니다. 다음 형식으로 archiver(1M) 명령을 입력하면 VSN 풀을 볼 수 있습니다.

```
# archiver -lv | more
```

VSN 풀 정의에는 흰 공백으로 분리되는 최소 세 개의 필드 즉, 풀 이름, 매체 유형 및 최소 하나의 VSN이 필요합니다. 구문은 다음과 같습니다.

```
vsn_pool_name media_type vsn_expression
```

표 4-23 VSN 풀 명령에 대한 인수

인수	의미
<i>vsn_pool_name</i>	VSN 풀을 지정합니다.
<i>media_type</i>	2개 문자의 매체 유형. 올바른 매체 유형의 목록에 대해서는 mcf(4) 매뉴얼 페이지를 참조하십시오.
<i>vsn_expression</i>	일반 표현 하나 이상의 <i>vsn_expression</i> 인수가 있을 수 있습니다. regcmp(3G) 매뉴얼 페이지를 참조하십시오.

다음 예제는 네 개의 VSN 풀을 사용합니다. *users_pool*, *data_pool*, *proj_pool* 및 *scratch_pool*. 세 개의 특정 풀 중 하나가 볼륨을 벗어나는 경우 아카이버는 스크래치 풀 VSN을 선택합니다. 코드 예 4-35는 네 개의 VSN 풀을 사용하는 *archiver.cmd* 파일을 표시합니다.

코드 예 4-35 VSN 풀 표시 예제

```
vsnpools
users_pool mo ^MO[0-9][0-9]
data_pool mo ^DA.*
scratch_pool mo ^SC[5-9][0-9]
proj_pool mo ^PR.*
endvsnpools
vsns
users.1 mo -pool users_pool -pool scratch_pool
```

data.1	mo	-pool data_pool	-pool scratch_pool
proj.1	mo	-pool proj_pool	-pool scratch_pool
endvsns			

디스크 아카이브

아카이브란 온라인 디스크에서 아카이브 매체로 파일을 복사하는 프로세스를 말합니다. 경우에 따라 아카이브 복사본은 자동화 라이브러리의 광자기 또는 테이프 카트리지의 볼륨에 작성되나 디스크 아카이브의 경우에는 파일 시스템의 온라인 디스크를 아카이브 매체로 사용합니다.

1개의 Sun StorEdge SAM-FS 파일 시스템에서 호스트 컴퓨터 시스템이 동일한 다른 파일 시스템으로 파일이 아카이브되도록 디스크 아카이브를 실행할 수 있습니다. 또한 디스크 아카이브는 다른 Sun Solaris 시스템의 파일 시스템에 소스 파일이 아카이브되도록 구현할 수 있습니다. 2개의 호스트 시스템을 사용해 디스크 아카이브를 실행하면 해당 시스템은 클라이언트와 서버로 작동합니다. *클라이언트 시스템*은 소스 파일을 호스트하는 시스템입니다. *서버 시스템*은 아카이브 복사본을 호스트하는 대상 시스템입니다.

아카이브 파일을 작성하는 파일 시스템은 UNIX 파일 시스템이 될 수 있습니다. Sun StorEdge SAM-FS 파일 시스템일 필요는 없습니다. 디스크 아카이브 복사본이 다른 호스트에 기록되는 경우 호스트에는 최소 하나의 Sun StorEdge SAM-FS 파일 시스템이 설치되어 있어야 합니다.

아카이버는 라이브러리의 볼륨에 아카이브된 파일을 취급하는 방식과 동일하게 디스크 볼륨에 아카이브된 파일을 취급합니다. 그럼에도 1개, 2개, 3개 또는 4개의 아카이브 복사본을 만들 수 있습니다. 여러 개의 아카이브 복사본을 만드는 경우에는 아카이브 복사본 중 하나가 디스크 볼륨에 작성되고 나머지 복사본은 제거 가능한 매체 볼륨에 작성될 수 있습니다. 이밖에 Sun StorEdge SAM-FS 파일 시스템의 디스크 볼륨에 아카이브하는 경우에는 아카이브 파일 복사본이 해당 파일 시스템의 archiver.cmd 파일 규칙에 따라 스스로 아카이브됩니다.

다음 목록에서는 온라인 디스크에 대한 아카이브와 제거 가능한 매체에 대한 아카이브의 유사점과 차이점 일부를 설명합니다.

- 광자기 디스크나 테이프에 작성하는 아카이브 복사본과는 달리 디스크에 작성하는 아카이브 복사본은 카탈로그에 기록되지 않습니다. 또한 디스크 볼륨의 아카이브 파일은 기록자에 나타나지 않습니다.
- 제거 가능한 매체 볼륨에 아카이브하는 경우에는 archiver.cmd 파일의 기본값을 변경하지 않고서도 파일 시스템을 마운트한 후에 아카이브를 시작할 수 있습니다. 그러나 볼륨 디스크에 아카이브하는 경우에는 파일 시스템을 마운트하기 전에 archiver.cmd 파일을 편집하고 디스크 아카이브 세트를 정의해야 합니다.

- 디스크 아카이브는 mcf(4) 파일 항목의 영향을 받지 않습니다. archiver.cmd 파일에 -disk_archive 매개 변수를 정의하고 /etc/opt/SUNWsamfs/diskvols.conf에 디스크 볼륨을 정의해야 합니다. 이는 추가 구성 파일이며, 제거 가능한 볼륨에만 아카이브하는 경우에는 필요하지 않습니다.

diskvols.conf 파일을 소스 파일이 상주하는 시스템에 만들어야 합니다. 아카이브 복사본이 작성되는 위치에 따라 이 파일에는 또한 다음과 같은 정보가 포함될 수 있습니다.

- 동일한 호스트 시스템에서 아카이브 복사본을 파일 시스템에 작성하는 경우, diskvols.conf 파일은 VSN 및 각 VSN의 경로를 정의합니다.
- 아카이브 복사본을 다른 Sun Solaris 시스템에 작성하는 경우, diskvols.conf 파일에는 해당 서버 시스템의 호스트 이름이 포함됩니다. 이러한 경우에는 또한 해당 시스템에 대한 쓰기 권한이 있는 클라이언트를 정의하는 서버 시스템에 diskvols.conf 파일이 있어야 합니다. 이 클라이언트/서버 관계를 작성하려는 경우 111 페이지의 "아카이브 디스크 활성화 방법"에 나온 절차를 시작하기 전에 서버 역할을 하는 호스트에 최소 하나의 Sun StorEdge SAM-FS 파일 시스템이 설치되어 있어야 합니다.

구성 지침

디스크 아카이브 볼륨은 어느 위치에나 상주할 수 있으나 원본 파일이 상주하는 위치 이외의 디스크에 상주하는 것이 바람직합니다. 가급적이면 클라이언트 시스템의 아카이브 복사본은 서버 시스템의 디스크 볼륨에 작성하는 것이 좋습니다. 아카이브 복사본을 2개 이상 만들어 2가지 이상의 아카이브 매체 유형에 작성하는 것이 바람직합니다. 예를 들어 복사본 1은 디스크 볼륨에, 복사본 2는 테이프에, 복사본 3은 광자기 디스크에 아카이브할 수 있습니다.

서버 시스템의 파일 시스템에 파일을 아카이브하는 경우, 아카이브 파일 자체는 대상 서버에 추가된 라이브러리의 제거 가능한 매체 카트리지에 아카이브될 수 있습니다.

디스크 아카이브 명령

온라인 디스크에 아카이브하는 경우 아카이버는 대부분의 archiver.cmd 명령을 인식합니다. 아카이버가 인식하는 명령은 아카이브 세트를 정의하고 리사이클을 구성합니다. 소리없이 무시된 명령은 특히 제거 가능한 매체 카트리지와 함께 작동하는 것과 관련되기 때문에 디스크 아카이브 환경에서 의미가 없습니다. 특히 시스템은 디스크 아카이브 세트에 대해 다음 명령을 인식합니다.

- 다음을 제외한 90 페이지의 "아카이브 설정 복사 매개변수"의 모든 리사이클 명령
 - -fillvsns
 - -ovflmin *min_size*
 - -reserve *method*
 - -tapenonstop

- 다음을 제외한 175 페이지의 "2 단계: archiver.cmd 파일 편집 (선택적)"의 모든 명령
 - `-recycle_dataquantity size`
 - `-recycle_vsncount count`
- `-disk_archive` 매개변수. 이것은 매개변수를 처리하는 아카이브 세트입니다. `archiver.cmd` 파일의 `-disk_archive` 매개변수를 지정하여 디스크 아카이브 세트를 정의해야 합니다. 아카이브 디스크의 마운트 지점에 디스크 아카이브 세트를 작성하는 동안 아카이버는 이 매개 변수를 사용해 데이터의 파일 시스템 계층을 관리합니다. 매개변수를 처리하는 모든 아카이브 세트처럼 `params` 및 `endparams` 명령 사이에 지정되어야 합니다. 코드 예 4-36은 이 명령의 형식을 표시합니다.

코드 예 4-36 `-disk_archive` 매개변수에 대한 형식

```
Params
archive_set.copy_number -disk_archive VSN_Name
endparams
```

`VSN_Name`에는 `diskvols.conf` 파일에 정의된 `VSN`을 지정합니다.

- `clients` 및 `endclients` 명령 클라이언트 호스트에서 서버 호스트로 원본 파일을 아카이브하기 위해 디스크 아카이브를 구현하는 경우 서버 호스트에 `diskvols.conf` 파일을 구성해야 합니다. 서버 시스템의 `diskvols.conf` 파일은 클라이언트 시스템의 이름을 포함해야 합니다. 이 명령의 형식은 다음과 같습니다.

코드 예 4-37 `clients` 및 `endclients` 명령에 대한 형식

```
clients
client_system1
client_system2
...
endclients
```

`client_system`에 대해 원본 파일이 있는 클라이언트 시스템의 호스트 이름을 지정합니다.

디스크 아카이브 명령에 대한 자세한 내용은 `archiver.cmd(4)` 매뉴얼 페이지를 참조하십시오.

▼ 아카이브 디스크 활성화 방법

언제라도 디스크 아카이브를 활성화할 수 있습니다. 이 절은 아카이브가 제 위치에 있고 디스크 아카이브를 사용 중인 환경에 추가하는 중이라는 가정하에서 절차를 설명합니다. 초기 설치의 일부로 디스크 아카이브를 활성화하려는 경우 해당 정보는 *Sun StorEdge QFS 및 Sun StorEdge SAM-FS 소프트웨어 설치 및 구성 안내서*를 참조하십시오. 설치할 때 디스크 아카이브를 추가하려는 경우 필요하지 않은 단계가 포함되어 있으므로 이 절차를 사용하지 마십시오.

1. 디스크 아카이브 복사본을 기록하려는 호스트에 최소 한 개의 Sun StorEdge SAM-FS 파일 시스템이 설치되어 있어야 합니다.
2. 아카이브할 파일이 있는 호스트 시스템에서 슈퍼유저가 됩니다.
3. 디스크 아카이브를 활성화하려면 *Sun StorEdge QFS* 및 *Sun StorEdge SAM-FS* 소프트웨어 설치 및 구성 안내서의 절차를 따릅니다.
Sun StorEdge SAM-FS 초기 설치 절차에는 일명 *디스크 아카이브 활성화* 단계가 포함됩니다. 해당 단계는 두 절차로 나뉩니다.
4. 아카이브할 파일이 있는 호스트에서 슈퍼유저가 됩니다.
5. 아카이브할 파일이 있는 호스트에서 `samd(1M) config` 명령을 사용하여 구성 파일 변경 내용을 전파하고 시스템을 재시작합니다.
예를 들어, 다음과 같습니다.

```
# samd config
```

6. 아카이브 복사본이 기록되는 호스트 시스템에서 슈퍼유저가 됩니다. (선택 사항)
다른 호스트의 디스크에 아카이브하려는 경우에만 이 단계를 수행합니다.
7. 아카이브 복사본이 기록될 호스트에서 `samd(1M) config` 명령을 사용하여 구성 파일 변경 내용을 전파하고 시스템을 재시작합니다. (선택 사항)
다른 호스트의 디스크에 아카이브하려는 경우에만 이 단계를 수행합니다.
예를 들어, 다음과 같습니다.

```
# samd config
```

디스크 아카이브 예제

예제 1

코드 예 4-38은 클라이언트 시스템 pluto에 상주하는 diskvols.conf 파일을 표시합니다.

코드 예 4-38 pluto의 diskvols.conf 파일

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on pluto
# VSN Name [Host Name:]Path
#
disk01 /sam_arch1
disk02 /sam_arch2/proj_1
disk03 mars:/sam_arch3/proj_3
```

위의 diskvols.conf 파일에서 disk01 및 disk02로 식별된 VNS는 원본 소스 파일이 상주하는 호스트 시스템에 작성됩니다. VSN disk03은 서버 시스템 mars의 VSN에 작성됩니다.

코드 예 4-39는 서버 시스템 mars의 diskvols.conf 파일을 표시합니다.

코드 예 4-39 mars의 diskvols.conf 파일

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on mars
#
clients
pluto
endclients
```

코드 예 4-40은 pluto의 archiver.cmd 파일 일부를 표시합니다.

코드 예 4-40 pluto의 archiver.cmd 파일

```
Params
arset1.2 -disk_archive disk01
arset2.2 -disk_archive disk02
arset3.2 -disk_archive disk03
endparams
```

예제 2

이 예제에서 /sam1/testdir0/filea 파일은 arset0.1에 대한 아카이브 세트이고 아카이버는 /sam1/testdir0/filea의 내용을 일명 /sam_arch1이라는 대상 경로에 복사합니다. 코드 예 4-41는 diskvols.conf 파일을 표시합니다.

코드 예 4-41 diskvols.conf 파일

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf
#
# VSN Name    [Host Name:]Path
#
disk01                /sam_arch1
disk02                /sam_arch12/proj_1
```

코드 예 4-42는 디스크 아카이브에 관련된 archiver.cmd 파일 행을 표시합니다.

코드 예 4-42 디스크 아카이브에 관련된 archiver.cmd 파일의 명령

```
.
.
.
Params
arset0.1 -disk_archive disk01
endparams
.
.
.
```

다음은 디스크에 아카이브된 파일 filea의 sls(1) 명령을 보여줍니다. 코드 예 4-43에서 다음을 유의하십시오.

- dk는 디스크 아카이브 매체에 대한 매체 유형입니다.
- disk01는 VSN입니다.
- f192는 디스크 아카이브 tar(1) 파일에 대한 경로입니다.

코드 예 4-43 sls(1M) 출력

```
# sls -D /sam1/testdir0/filea
/sam1/testdir0/filea:
mode: ---R----- 1 owner: root      group: other
length: 797904 admin id: 0 inode: 3134.49
archdone;
copy 1: ---- Dec 16 14:03          c0.1354 dk disk01 f192
access:      Dec 19 10:29 modification: Dec 16 13:56
changed:     Dec 16 13:56 attributes:   Dec 19 10:29
creation:    Dec 16 13:56 residence:    Dec 19 10:32
```

예제 3

이 예제에서 파일 /sam2/my_proj/fileb는 아카이브 세트 arset0.1의 클라이언트 호스트 snickers에 있으며, 아카이버는 이 파일의 내용을 서버 호스트 mars의 대상 경로 /sam_arch1에 복사합니다.

코드 예 4-44는 snickers의 diskvols.conf 파일을 표시합니다.

코드 예 4-44 snickers의 diskvols.conf 파일

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on snickers
#
# VSN Name    [Host Name:]Path
#
disk01       mars:/sam_arch1
```

코드 예 4-45는 mars의 diskvols.conf 파일을 표시합니다.

코드 예 4-45 mars의 diskvols.conf 파일

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on mars
#
clients
snickers
endclients
```

코드 예 4-46은 이 예제와 관련된 archiver.cmd 파일의 명령을 표시합니다.

코드 예 4-46 디스크 아카이브에 관련된 archiver.cmd 파일의 명령

```
.
.
.
Params
arset0.1 -disk_archive disk01
endparams
.
.
.
```

아카이버 예제

표 4-24는 이 절의 모든 예제가 사용하는 디렉토리 구조를 표시합니다.

표 4-24 디렉토리 구조 예제

최상위 디렉토리	첫 번째 레벨 하위 디렉토리	두 번째 레벨 하위 디렉토리	세 번째 레벨 하위 디렉토리
/sam	/projs	/proj_1	/katie
/sam	/projs	/proj_1	/sara
/sam	/projs	/proj_1	/wendy
/sam	/projs	/proj_2	/joe
/sam	/projs	/proj_2	/katie
/sam	/users	/bob	
/sam	/users	/joe	
/sam	/users	/katie	
/sam	/users	/sara	
/sam	/users	/wendy	
/sam	/data		
/sam	/tmp		

예제 1

이 예제에서는 archiver.cmd 파일을 사용하지 않는 경우 아카이버의 작업에 대해 설명합니다. 이 예제에서 Sun StorEdge SAM-FS 환경에는 1개의 파일 시스템, 드라이브가 2개인 광자동화 라이브러리 및 6개의 카트리지가 포함됩니다.

코드 예 4-47은 archiver(1M) -lv 명령을 실행하여 나타난 출력입니다. 이것은 아카이버에서 선택한 기본 매체가 유형 mo임을 보여줍니다. mo 매체만 사용할 수 있습니다.

코드 예 4-47 archiver(1M) -lv 출력 예제 1 부

```
# archiver -lv
Notify file: /etc/opt/SUNWsamfs/archiver.cmd

Archive media:
media:lt archmax: 512.0M Volume overflow not selected
media:mo archmax: 4.8M Volume overflow not selected
```

코드 예 4-48에서는 아카이버가 2개의 드라이브를 사용함을 보여줍니다. 12 개의 볼륨, 저장 용량 및 사용 가능한 공간을 나열합니다.

코드 예 4-48 archiver(1M) -lv 출력 예제 2 부

```
Archive libraries:
Device:hp30 drives_available:2 archive_drives:2
Catalog:
mo.optic00          capacity: 1.2G space: 939.7M -il-o-----
mo.optic01          capacity: 1.2G space: 934.2M -il-o-----
mo.optic02          capacity: 1.2G space: 781.7M -il-o-----
mo.optic03          capacity: 1.2G space: 1.1G -il-o-----
mo.optic10          capacity: 1.2G space: 85.5M -il-o-----
mo.optic11          capacity: 1.2G space: 0 -il-o-----
mo.optic12          capacity: 1.2G space: 618.9k -il-o-----
mo.optic13          capacity: 1.2G space: 981.3M -il-o-----
mo.optic20          capacity: 1.2G space: 1.1G -il-o-----
mo.optic21          capacity: 1.2G space: 1.1G -il-o-----
mo.optic22          capacity: 1.2G space: 244.9k -il-o-----
mo.optic23          capacity: 1.2G space: 1.1G -il-o-----
```

코드 예 4-49에서는 메타 데이터 및 데이터 파일 모두 아카이브 세트 samfs에 포함됨을 보여줍니다. 아카이버는 아카이브 나이가 기본 4분(240초)에 도달하면 파일 복사본을 1 개 만듭니다.

코드 예 4-49 archiver(1M) -lv 출력 예제 3 부

```
Archive file selections:
Filesystem samfs Logfile:
samfs Metadata
copy:1 arch_age:240
samfs1 path:.
copy:1 arch_age:240
```

코드 예 4-50에서는 지정된 순서에 따라 볼륨에 아카이브되는 아카이브 세트의 파일을 보여줍니다.

코드 예 4-50 archiver(1M) -lv 출력 예제 4 부

```
Archive sets:
allsets
samfs.1
media mo (by default)
Volumes:
optic00
optic01
optic02
optic03
optic10
```

코드 예 4-50

archiver(IM) -lv 출력 예제 4 부 (계속)

```
optic12
optic13
optic20
optic21
optic22
optic23
Total space available: 8.1G
```

예제 2

이 예제에서는 데이터 파일을 메타 데이터와 분리된 2개의 아카이브 세트로 분리하는 방법을 보여줍니다. 114 페이지의 "예제 2"의 광자동화 라이브러리 이외에도 수동으로 마운트한 DLT 테이프 드라이브가 있습니다. 큰 파일은 테이프에 아카이브되고 작은 파일은 광카트리지에 아카이브됩니다.

코드 예 4-51은 archiver.cmd 파일의 내용을 표시합니다.

코드 예 4-51

archiver.cmd 파일을 표시하는 archiver(IM) -lv 출력 1 부

```
# archiver -lv -c example2.cmd
Reading archiver command file "example2.cmd"
1: # Example 2 archiver command file
2: # Simple selections based on size
3:
4: logfile = /var/opt/SUNWsamfs/archiver/log
5: interval = 5m
6:
7: # File selections.
8: big . -minsize 500k
9: all .
10: 1 30s
11:
12: vsns
13: samfs.1 mo .*[0-2] # Metadata to optic00 - optic02
14: all.1 mo .*[3-9] .*[1-2][0-9] # All others for files
15: big.1 lt .*
16: endvsns
```

코드 예 4-52는 추가 DLT 및 해당 기본값이 아니라 사용되는 매체 및 드라이브를 표시합니다.

코드 예 4-52

매체 및 드라이브를 표시하는 archiver(IM) -lv 출력 2 부

```
Notify file: /etc/opt/SUNWsamfs/scripts/archiver.sh
Archive media:
media:lt archmax: 512.0M Volume overflow not selected
```

코드 예 4-52

매체 및 드라이브를 표시하는 archiver(IM) -lv 출력 2 부 (계속)

```

media:mo archmax: 4.8M Volume overflow not selected
Archive libraries:
Device:hp30 drives_available:0 archive_drives:0
Catalog:
mo.optic00      capacity: 1.2G space: 939.7M -il-o-----
mo.optic01      capacity: 1.2G space: 934.2M -il-o-----
mo.optic02      capacity: 1.2G space: 781.7M -il-o-----
mo.optic03      capacity: 1.2G space: 1.1G -il-o-----
mo.optic04      capacity: 1.2G space: 983.2M -il-o-----
mo.optic10      capacity: 1.2G space: 85.5M -il-o-----
mo.optic11      capacity: 1.2G space: 0 -il-o-----
mo.optic12      capacity: 1.2G space: 618.9k -il-o-----
mo.optic13      capacity: 1.2G space: 981.3M -il-o-----
mo.optic20      capacity: 1.2G space: 1.1G -il-o-----
mo.optic21      capacity: 1.2G space: 1.1G -il-o-----
mo.optic22      capacity: 1.2G space: 244.9k -il-o-----
mo.optic23      capacity: 1.2G space: 1.1G -il-o-----
Device:lt40 drives_available:0 archive_drives:0
Catalog:
lt.TAPE01      capacity: 9.5G space: 8.5G -il-o-----
lt.TAPE02      capacity: 9.5G space: 6.2G -il-o-----
lt.TAPE03      capacity: 9.5G space: 3.6G -il-o-----
lt.TAPE04      capacity: 9.5G space: 8.5G -il-o-----
lt.TAPE05      capacity: 9.5G space: 8.5G -il-o-----
lt.TAPE06      capacity: 9.5G space: 7.4G -il-o-----

```

코드 예 4-53에 파일 시스템의 구성이 나와 있습니다. 512000 바이트(500 킬로바이트)보다 큰 파일이 4분 후에 아카이브되며, 나머지 파일은 30초 후에 모두 아카이브됩니다.

코드 예 4-53

파일 시스템 구성을 표시하는 archiver(IM) -lv 출력 3 부

```

Archive file selections:
Filesystem samfs Logfile: /var/opt/SUNWsamfs/archiver/log
samfs Metadata
copy:1 arch_age:240
big path:. minsize:502.0k
copy:1 arch_age:240
all path:.
copy:1 arch_age:30

```

코드 예 4-54에서 아카이브의 분리가 다음과 같은 결과로 제거 가능한 매체에 설정됨을 보여 줍니다.

코드 예 4-54 아카이브 세트 및 제거 가능한 매체를 표시하는 archiver(1M) -lv 출력 3 부

```
Archive sets:
allsets
all.1
  media: mo
Volumes:
  optic03
  optic04
  optic10
  optic12
  optic13
  optic20
  optic21
  optic22
  optic23
  Total space available: 6.3G
big.1
  media: lt
Volumes:
  TAPE01
  TAPE02
  TAPE03
  TAPE04
  TAPE05
  TAPE06
  Total space available: 42.8G
samfs.1
  media: mo
Volumes:
  optic00
  optic01
  optic02
  Total space available: 2.6G
```

예제 3

이 예제에서는 사용자 파일과 프로젝트 데이터 파일이 다양한 매체에 아카이브됩니다. 디렉토리 데이터의 파일은 크기별로 광매체와 테이프 매체에 분리됩니다. 그룹 ID pict에 할당된 파일은 다른 세트의 볼륨에 할당됩니다. 디렉토리 tmp 및 users/bob의 파일은 아카이브되지 않습니다. 아카이브는 15분 간격으로 수행되며, 아카이브 레코드는 보유됩니다.

코드 예 4-55는 이 예제를 표시합니다.

코드 예 4-55 archiver(1M) -lv 명령 출력

```
# archiver -lv -c example3.cmd
Reading archiver command file "example3.cmd"
1: # Example 3 archiver command file
2: # Segregation of users and data
3:
4: interval = 30s
5: logfile = /var/opt/SUNWsamfs/archiver/log
6:
7: no_archive tmp
8:
9: fs = samfs
10: no_archive users/bob
11: prod_big data -minsize 50k
12:   1 1m 30d
13:   2 3m
14: prod data
15:   1 1m
16: proj_1 projs/proj_1
17:   1 1m
18:   2 1m
19: joe/ -user joe
20:   1 1m
21:   2 1m
22: pict . -group pict
23:   1 1m
24:   2 1m
25:
26: Params
27: prod_big.1 -drives 2
28: prod_big.2 -drives 2
29: endparams
30:
31: vsns
32: samfs.1 mo optic0[0-1]$
33: joe.1 mo optic01$
34: pict.1 mo optic02$
35: pict.2 mo optic03$
36: proj_1.1 mo optic1[0-1]$
37: proj_1.2 mo optic1[2-3]$
38: prod.1 mo optic2.$
39: joe.2 lt 0[1-2]$
40: prod_big.1 lt 0[3-4]$
41: prod_big.2 lt 0[5-6]$
42: endvsns
```

```

Notify file: /etc/opt/SUNWsamfs/scripts/archiver.sh

Archive media:
media:lt archmax: 512.0M Volume overflow not selected
media:mo archmax: 4.8M Volume overflow not selected

Archive libraries:
Device:hp30 drives_available:0 archive_drives:0
Catalog:
mo.optic00      capacity: 1.2G space: 939.7M -il-o-----
mo.optic01      capacity: 1.2G space: 934.2M -il-o-----
mo.optic02      capacity: 1.2G space: 781.7M -il-o-----
mo.optic03      capacity: 1.2G space: 1.1G -il-o-----
mo.optic04      capacity: 1.2G space: 983.2M -il-o-----
mo.optic10      capacity: 1.2G space: 85.5M -il-o-----
mo.optic11      capacity: 1.2G space: 0 -il-o-----
mo.optic12      capacity: 1.2G space: 618.9k -il-o-----
mo.optic13      capacity: 1.2G space: 981.3M -il-o-----
mo.optic20      capacity: 1.2G space: 1.1G -il-o-----
mo.optic21      capacity: 1.2G space: 1.1G -il-o-----
mo.optic22      capacity: 1.2G space: 244.9k -il-o-----
mo.optic23      capacity: 1.2G space: 1.1G -il-o-----

Device:lt40 drives_available:0 archive_drives:0
Catalog:
lt.TAPE01      capacity: 9.5G space: 8.5G -il-o-----
lt.TAPE02      capacity: 9.5G space: 6.2G -il-o-----
lt.TAPE03      capacity: 9.5G space: 3.6G -il-o-----
lt.TAPE04      capacity: 9.5G space: 8.5G -il-o-----
lt.TAPE05      capacity: 9.5G space: 8.5G -il-o-----
lt.TAPE06      capacity: 9.5G space: 7.4G -il-o-----

Archive file selections:
Filesystem samfs Logfile: /var/opt/SUNWsamfs/archiver/log
samfs Metadata
  copy:1 arch_age:240
no_archive Noarchive path:users/bob
prod_big path:data minsize:50.2k
  copy:1 arch_age:60 unarch_age:2592000
  copy:2 arch_age:180
prod path:data
  copy:1 arch_age:60
proj_1 path:projs/proj_1
  copy:1 arch_age:60
  copy:2 arch_age:60
joe path:. uid:10006
  copy:1 arch_age:60

```

코드 예 4-55

archiver(1M) -lv 명령 출력 (계속)

```
    copy:2  arch_age:60
pict path:. gid:8005
    copy:1  arch_age:60
    copy:2  arch_age:60
no_archive Noarchive path:tmp
samfs path:.
    copy:1  arch_age:240

Archive sets:
allsets

joe.1
media: mo
Volumes:
    optic01
Total space available: 934.2M

joe.2
media: lt
Volumes:
    TAPE01
    TAPE02
Total space available: 14.7G

pict.1
media: mo
Volumes:
    optic02
Total space available: 781.7M

pict.2
media: mo
Volumes:
    optic03
Total space available: 1.1G

prod.1
media: mo
Volumes:
    optic20
    optic21
    optic22
    optic23
Total space available: 3.3G

prod_big.1
media: lt drives:2
```

```

Volumes:
  TAPE03
  TAPE04
Total space available: 12.1G

prod_big.2
media: lt drives:2
Volumes:
  TAPE05
  TAPE06
Total space available: 16.0G

proj_1.1
media: mo
Volumes:
  optic10
Total space available: 85.5M

proj_1.2
media: mo
Volumes:
  optic12
  optic13
Total space available: 981.9M

samfs.1
media: mo
Volumes:
  optic00
  optic01
Total space available: 1.8G

```

예제 4

이 예제에서는 사용자 파일과 프로젝트 데이터 파일이 광 매체에 아카이브됩니다. 코드 예 4-56에서는 표 4-24에 나와있는 디렉토리 구조를 사용하지 않습니다.

4개의 VSN 풀이 정의됩니다. 이 중 3개 풀은 사용자, 데이터 및 프로젝트에 사용하고, 1개 풀은 스크래치 풀입니다. proj_pool에 매체가 없으면 scratch_pool에 따라 볼륨을 예약합니다. 이 예제에서는 세트 구성 요소, 소유자 구성 요소 및 파일 시스템 구성 요소에 따라 각 아카이브 세트에 대해 볼륨을 예약하는 방법을 보여줍니다. 아카이브는 10분 간격으로 수행되고 아카이브 로그가 보유됩니다.

코드 예 4-56에서는 archiver.cmd 파일 및 아카이버 결과를 보여줍니다.

코드 예 4-56 archiver.cmd 파일 및 아카이버 결과

```
Reading archiver command file "example4.cmd"
1: # Example 4 archiver command file
2: # Using 4 VSN pools
3:
4: interval = 30s
5: logfile = /var/opt/SUNWsamfs/archiver/log
6:
7: fs = samfs
8: users users
9:     1 10m
10:
11: data data
12:     1 10m
13:
14: proj projects
15:     1 10m
16:
17: Params
18: users.1 -reserve user
19: data.1 -reserve group
20: proj.1 -reserve dir -reserve fs
21: endparams
22:
23: vsnpools
24: users_pool mo optic0[1-3]$
25: data_pool mo optic1[0-1]$
26: proj_pool mo optic1[2-3]$
27: scratch_pool mo optic2.$
28: endvsnpools
29:
30: vsn
31: samfs.1 mo optic00
32: users.1 mo -pool users_pool -pool scratch_pool
33: data.1 mo -pool data_pool -pool scratch_pool
34: proj.1 mo -pool proj_pool -pool scratch_pool
35: endvsns

Notify file: /etc/opt/SUNWsamfs/scripts/archiver.sh

Archive media:
media:mo archmax: 4.8M Volume overflow not selected

Archive libraries:
Device:hp30 drives_available:0 archive_drives:0
Catalog:
```

```

Reading archiver command file "example4.cmd"
mo.optic00      capacity: 1.2G space: 939.7M -il-o-----
mo.optic01      capacity: 1.2G space: 934.2M -il-o-----
mo.optic02      capacity: 1.2G space: 781.7M -il-o-----
mo.optic03      capacity: 1.2G space: 1.1G -il-o-----
mo.optic04      capacity: 1.2G space: 983.2M -il-o-----
mo.optic10      capacity: 1.2G space: 85.5M -il-o-----
mo.optic11      capacity: 1.2G space: 0 -il-o-----
mo.optic12      capacity: 1.2G space: 618.9k -il-o-----
mo.optic13      capacity: 1.2G space: 981.3M -il-o-----
mo.optic20      capacity: 1.2G space: 1.1G -il-o-----
mo.optic21      capacity: 1.2G space: 1.1G -il-o-----
mo.optic22      capacity: 1.2G space: 244.9k -il-o-----
mo.optic23      capacity: 1.2G space: 1.1G -il-o-----

```

Archive file selections:

```

Filesystem samfs Logfile: /var/opt/SUNWsamfs/archiver/log
samfs Metadata
  copy:1 arch_age:240
users path:users
  copy:1 arch_age:600
data path:data
  copy:1 arch_age:600
proj path:projects
  copy:1 arch_age:600
samfs path:.
  copy:1 arch_age:240

```

VSN pools:

```

data_pool media: mo Volumes:
  optic10
Total space available: 85.5M

```

```

proj_pool media: mo Volumes:
  optic12
  optic13
Total space available: 981.9M

```

```

scratch_pool media: mo Volumes:
  optic20
  optic21
  optic22
  optic23
Total space available: 3.3G

```

```

users_pool media: mo Volumes:

```

```
Reading archiver command file "example4.cmd"
  optic01
  optic02
  optic03
Total space available: 2.7G

Archive sets:
allsets

data.1
  reserve:/group/
  media: mo
  Volumes:
    optic10
    optic20
    optic21
    optic22
    optic23
Total space available: 3.4G

proj.1
  reserve:/dir/fs
  media: mo
  Volumes:
    optic12
    optic13
    optic20
    optic21
    optic22
    optic23
Total space available: 4.2G

samfs.1
  media: mo
  Volumes:
    optic00
Total space available: 939.7M

users.1
  reserve:/user/
  media: mo
  Volumes:
    optic01
    optic02
    optic03
    optic20
    optic21
```

```

Reading archiver command file "example4.cmd"
  optic22
  optic23
Total space available:    6.0G

```

아카이버 지침

아카이버는 archiver.cmd 파일을 사용해 저장 관리 작업을 자동화합니다. 이 파일을 작성하려면 먼저 Sun StorEdge SAM-FS 파일 시스템 및 아카이버의 성능을 향상시킬 수 있는 몇 가지 일반적 지침을 검토하는 것이 바람직합니다. 이렇게 해야 데이터를 최대한 안전하게 저장할 수 있습니다.

각 사이트는 연산 애플리케이션, 데이터 저장 하드웨어 및 소프트웨어가 각기 다릅니다. 다음 권장 사항은 Sun Microsystems사의 경험을 바탕으로 한 것입니다. 사이트에 대해 archiver.cmd 파일을 작성할 때는 다음과 같은 측면을 고려해 사이트에 데이터 저장 요구 사항을 반영해야 합니다.

1. 아카이브 로그를 저장합니다. 아카이브 로그는 Sun StorEdge SAM-FS 소프트웨어를 사용할 수 없을 때에도 데이터 복구에 필수적인 정보를 제공합니다. Sun StorEdge SAM-FS 소프트웨어를 사용할 수 없는 기간에 심각한 재난이 발생할 수 있으므로 이에 대비해 이러한 로그를 안전한 곳에 보관하는 것이 바람직합니다.
2. 볼륨에 대해 일반 표현식을 사용합니다. 시스템이 여러 볼륨에 파일을 분산시킬 수 있도록 허용하여 자동으로 작동될 수 있도록 합니다. 시스템은 일반 표현식을 사용해 지정된 볼륨 범위에 따라 연속적으로 실행될 수 있습니다. 아카이브 세트 복사본에 대해 특정 볼륨 이름을 사용하면 볼륨을 빨리 채울 수 있기 때문에 매체를 제거 및 교체할 때 잘못된 워크플로 문제가 발생합니다.
3. 아카이브 간격은 파일 생성 및 수정 주기와 수정 복사본을 모두 저장할지 여부에 따라 결정해야 합니다. 아카이브 간격은 파일 시스템 스캔 간의 시간입니다. 아카이브 간격이 너무 짧으면 아카이버가 거의 연속해서 스캔할 수 없게 됩니다.
4. 사용 중인 파일 시스템의 수를 고려합니다. 여러 Sun StorEdge SAM-FS 파일 시스템은 대개 단일 Sun StorEdge SAM-FS 파일 시스템에 비하여 아카이브의 성능을 향상 시킵니다. 아카이버는 각 파일 시스템에 대해 별도의 프로세스를 사용합니다. 1개의 파일 시스템보다 훨씬 짧은 시간에 여러 파일 시스템을 스캔할 수 있습니다.
5. 디렉토리 구조를 사용하여 UNIX 파일 시스템처럼 Sun StorEdge SAM-FS 파일 시스템 내의 파일을 구성합니다. Sun Microsystems사에서는 성능을 고려해 디렉토리에 파일을 10,000개 이상 넣지 않도록 권장하고 있습니다.
6. 별도의 볼륨 2곳에 최소 2개의 파일 복사본을 만듭니다. 데이터를 1가지 매체 유형에 저장하면 매체에 물리적인 문제가 발생하는 경우 데이터를 잃을 수 있습니다. 가끔 적으면 1개의 아카이브 복사본에 의존해서는 안됩니다.

7. 정기적으로 `samfsdump(1M)`를 사용해 메타 데이터를 덤프해야 합니다. 디렉토리 구조, 파일 이름 등의 메타 데이터는 파일 시스템과 이름이 같은 아카이브 세트에 저장됩니다. 재해가 발생한 경우에는 이 정보를 이용해 파일 시스템을 복구할 수 있습니다. 이를 예방하려면 이 아카이브 세트를 존재하지 않는 VSN에 할당하여 이 데이터가 아카이브되지 않도록 할 수 있습니다. 메타 데이터 보존에 대한 자세한 내용은 *Sun QFS, Sun SAM-FS 및 Sun SAM-QFS 재난 복구 안내서* 또는 *Sun StorEdge QFS 및 Sun StorEdge SAM-FS 소프트웨어 설치 및 구성 안내서*를 참조하십시오.

아카이버 문제 해결

초기 설정한 경우에는 아카이버가 예정된 작업을 수행하지 않을 수도 있습니다. 다음과 같은 도구를 사용해 시스템의 아카이브 작업을 모니터링해야 합니다.

- `samu(1M)` 유틸리티의 `a` 디스플레이. 이 디스플레이는 각 파일 시스템에 대한 아카이버 작업을 보여줍니다. 또한 다음과 같은 아카이버 오류 및 경고 메시지를 표시합니다.

Errors in archiver commands - no archiving will be done

`samu(1M)` 유틸리티의 `a` 화면에는 각 파일 시스템에 대한 메시지가 포함됩니다. 아카이버가 `.inodes` 파일 및 현재 아카이브되고 있는 파일을 스캔할 때를 표시합니다.

- 아카이브 로그. `archiver.cmd` 파일에 이 로그를 정의할 수 있고 파일이 볼륨에 아카이브되는지 정기적으로 모니터링합니다. 아카이브 로그는 지나치게 커질 수 있으므로 수동으로나 `cron(1)` 작업을 통해 정기적으로 줄여야 합니다. 이러한 로그 파일은 정보가 데이터 복구를 활성화하므로 안전한 보관을 위해 아카이브해야 합니다.
- `sfind(1)`. 이 명령을 사용해 아카이브되지 않은 파일을 정기적으로 확인합니다. 파일을 아카이브하지 않은 경우에는 그 이유를 알고 있어야 합니다.
- `sls(1)`. 파일은 유효한 아카이브 복사본이 있어야만 릴리스할 수 있습니다. `sls -D` 명령은 복사 정보를 포함해 파일에 대한 `inode` 정보를 표시합니다.

참고 - `sls -D` 명령의 수행 결과에는 파일에 대해 `archdone`라는 말이 표시될 수도 있습니다. 이는 파일에 아카이브 복사본이 있음을 나타내는 것은 아닙니다. 단지, 아카이버가 파일을 스캔했으며 아카이버와 관련된 모든 작업이 완료되었음을 나타내는 것입니다. 아카이브 복사본은 `sls(1)` 명령에 의해 표시되는 복사 정보를 볼 수 있을 때만 존재합니다.

경우에 따라서는 아카이버에 카트리지의 공간이 없거나 카트리지가 없음을 나타내는 메시지가 나타나기도 합니다. 이러한 메시지는 다음과 같습니다.

- 아카이버에 아카이브 세트에 할당된 카트리지가 없으면 아카이버는 다음과 같은 메시지를 생성합니다.

```
No volumes available for Archive Set setname
```

- 아카이버에 아카이브 세트에 할당된 카트리지에 공간이 없으면 아카이버는 다음과 같은 메시지를 생성합니다.

```
No space available on Archive Set setname
```

파일이 아카이브되지 않는 원인

다음 체크리스트에는 Sun StorEdge SAM-FS 환경이 파일을 아카이브하지 않는 이유가 포함됩니다.

1. `archiver.cmd` 파일에 구문 오류가 있습니다. `archiver -lv` 명령을 실행해 오류를 식별하고 플래그된 행을 수정합니다.
2. `archiver.cmd` 파일에 `wait` 명령이 있습니다. `wait` 명령을 제거하거나 `samu(1M)` 유틸리티의 `:arrun` 명령으로 이를 무시합니다.
3. 사용할 수 있는 볼륨이 없습니다. `archiver(1M) -lv` 명령 출력에서 이 볼륨을 볼 수 있습니다. 필요한 볼륨을 추가합니다. 경우에 따라서는 기존의 카트리지를 내보내 자동화 라이브러리의 슬롯을 비워야 합니다.
4. 아카이브 세트용 볼륨이 가득 찼습니다. 카트리지를 내보내고 새 카트리지로 교체하거나(카트리지에는 반드시 레이블을 지정), 카트리지를 리사이클할 수 있습니다. 리사이클에 대한 자세한 내용을 보려면 167 페이지의 "리사이클"을 참조하십시오.
5. `archiver.cmd` 파일의 VSN 섹션이 올바른 매체를 명시하지 않습니다. 일반 표현식과 VSN 풀이 올바르게 정의되었는지 확인합니다.

6. 사용 가능한 볼륨에 파일을 아카이브할 공간이 충분하지 않습니다. 더 큰 파일이 있고 볼륨이 거의 찬 것 같으면 카트리지는 Sun StorEdge SAM-FS 환경이 허용하는 최대치까지 찰 수도 있습니다. 이러한 경우에는 카트리지를 추가하거나 리사이클해야 합니다.

-join path 매개 변수를 지정했고 디렉토리의 모든 파일을 볼륨에 아카이브할 충분한 공간이 없으면 아카이브가 실행되지 않습니다. 카트리지를 추가하거나 리사이클하거나 다음 매개 변수 중 하나를 사용해야 합니다. -sort path or -rsort path. 이러한 매개 변수에 대한 자세한 내용은 97 페이지의 "연관 아카이브 -join"을 참조하십시오.
7. archiver.cmd 파일에 큰 파일이 포함된 디렉토리나 파일 시스템에 대해 설정된 no_archive 명령이 있습니다.
8. archive(1) -n(아카이브하지 않음) 명령으로 너무 많은 명령을 설정해 파일이 아카이브되지 않습니다.
9. 큰 파일이 사용 중입니다. 큰 파일이 아카이브 나이에 도달하지 않아 아카이브되지 않습니다.
10. 자동화 라이브러리에 하드웨어 또는 구성 문제가 있습니다.
11. 클라이언트와 서버 사이에 네트워크 연결 문제가 있습니다. 클라이언트와 서버 사이의 연결을 확인합니다.

추가 아카이버 진단

위 목록의 항목에 대한 검사 이외에도 아카이버 문제 해결시 다음 사항을 확인해야 합니다.

1. syslog 파일(기본적으로 /var/adm/sam-log). 이 파일에는 문제의 원인을 나타낼 수 있는 아카이버 메시지가 포함될 수 있습니다.
2. 볼륨 용량. 필요한 볼륨이 모두 있고 아카이브에 대해 볼륨의 공간이 충분한지 확인합니다.
3. 아카이버가 과도하고 알 수 없는 카트리지 작업을 야기하거나 실행되지 않는 것처럼 보이면 추적 기능을 사용해 추적 파일을 검사합니다. 추적 파일에 대한 내용은 defaults.conf(4) 매뉴얼 페이지를 참조하십시오.
4. 아카이버 프로세스(sam-archiverd)에 대해 truss(1) -p pid 명령을 사용해 반응하지 않는 시스템 호출을 파악할 수 있습니다. truss(1) 명령에 대한 자세한 내용은 truss(1) 매뉴얼 페이지를 참조하십시오.
5. showqueue(1M) 명령은 아카이버 대기열 파일의 내용을 표시합니다. 이 명령을 사용하여 예약 또는 아카이브되는 아카이버 요청 상태를 알 수 있습니다. 예약할 수 없는 아카이브 요청은 원인을 나타내는 메시지를 생성합니다. 이 명령은 또한 아카이브의 진행 상황을 표시합니다.

파일이 릴리스되지 않는 원인

아카이버 및 릴리서는 함께 실행되어 디스크 캐스에 사용할 수 있는 데이터 양의 균형을 유지합니다. 파일이 디스크 캐스에서 자동으로 릴리스되지 않는 주요 원인은 파일이 아직 아카이브되지 않았기 때문입니다.

파일이 릴리스되지 않는 원인에 대한 자세한 내용은 150 페이지의 "릴리서 문제 해결"을 참조하십시오.

릴리스

*릴리스*란 릴리서에서 아카이브된 파일을 식별하고 파일의 디스크 캐시 복사본을 릴리스하여 사용할 수 있는 디스크 캐시 공간을 확보하는 프로세스를 말합니다. 릴리스를 통해 만들어진 공간에는 아카이브 매체에서 다른 파일을 만들거나 스테이지할 수 있습니다. 릴리서는 아카이브된 파일만을 릴리스할 수 있습니다. 파일을 릴리스하면 디스크 캐시에 데이터가 없는 파일이 생성됩니다.

Sun StorEdge SAM-FS 파일 시스템은 사이트에 고유한 디스크 임계값에 도달하면 릴리서 프로세스를 실행합니다. 이와는 달리 `release(1)` 명령을 사용하면 파일의 디스크 공간을 즉시 릴리스하거나 파일의 릴리스 매개 변수를 설정할 수 있습니다. 릴리서 프로세스에 대한 자세한 내용은 `sam-releaser(1M)` 매뉴얼 페이지를 참조하십시오.

릴리서에는 파일을 아카이브 직후에 릴리스하거나 릴리스하지 않거나 부분적으로 릴리스하도록 지정할 수 있는 기능이 있습니다. `filemgr(1)`와 같은 애플리케이션은 파일의 시작 부분만을 읽기 때문에 부분 릴리스 기능을 사용하는 것이 유용합니다. 부분 릴리스 기능을 사용하면 파일의 일부는 디스크 캐시에 남고 나머지 부분은 릴리스 됩니다. 디스크 캐시에 남아있는 파일의 처음 부분을 읽는다고 해서 반드시 아카이브 매체에서 파일의 나머지 부분을 다시 디스크 캐시로 스테이지하는 것은 아닙니다. 이 장에서는 이러한 기능과 더불어 기타 여러 가지 다른 기능에 대해 설명합니다.

본 절에서는 다음 주제를 다룹니다.

- 134 페이지의 "릴리서 개요"
- 134 페이지의 "작동 원리"
- 135 페이지의 "용어 정의"
- 136 페이지의 "부분 릴리스 및 부분 스테이지"
- 139 페이지의 "`releaser.cmd` 파일"
- 147 페이지의 "릴리스에서 `archiver.cmd` 파일의 역할"
- 148 페이지의 "릴리서 구성"
- 149 페이지의 "릴리서 수동 실행"
- 150 페이지의 "릴리서 문제 해결"

릴리서 개요

파일 시스템 사용률이 구성된 최저 워터마크를 초과하면 파일 시스템 관리 소프트웨어에서 릴리서를 호출합니다. 릴리서는 먼저 `releaser.cmd` 파일을 읽고 릴리스 프로세스를 제어하는 명령을 수집합니다. 그 다음에는 파일 시스템을 스캔하고 각 파일에 관한 정보를 수집합니다. 마지막으로 릴리서는 전체 파일 시스템을 스캔한 후에 우선 순위에 따라 파일 릴리스를 시작합니다.

릴리서는 구성된 최저 워터마크 이상 파일 시스템이 남아있으면 파일 릴리스 계속합니다. 일반적으로 릴리서는 충분한 여유 공간을 남겨두어 파일 시스템이 최저 워터마크 이하로 떨어질 수 있도록 합니다. 릴리서가 릴리스할 파일을 찾지 못하면 종료됩니다. 릴리서는 나중에 릴리스할 파일이 더 있을 때 실행됩니다. 파일 시스템은 최고 워터마크 이상인 경우 매 1분마다 릴리서를 시작합니다.

최고 및 최저 워터마크는 `high=percent` 및 `low=percent` 파일 시스템 마운트 옵션을 사용하여 설정됩니다. 사용 가능한 마운트 매개 변수에 대한 자세한 내용은 `mount_samfs(1M)` 매뉴얼 페이지를 참조하십시오.

작동 원리

파일 시스템에는 수천개의 파일이 들어있을 수 있습니다. 몇 개의 큰 파일을 릴리스해야 파일 시스템이 최저 워터마크로 돌아올 수 있기 때문에 모든 파일의 릴리스 우선 순위를 추적하는 것은 비경제적입니다. 그러나 릴리서는 각 파일의 우선 순위를 검사해야 합니다. 그렇지 않으면, 최상의 대상을 찾지 못할 수도 있습니다. 이에 대비해 릴리서는 처음 10,000개의 대상만을 식별합니다.

처음 10,000개의 대상을 식별하여 이 가운데 최저 우선 순위 대상보다 높은 우선 순위의 대상이 없으면 이후 대상은 무시합니다.

릴리서는 처음 10,000개의 대상의 우선 순위를 결정한 후에 릴리스 우선 순위가 가장 높은 파일을 선택합니다. 릴리서는 각 파일을 릴리스한 다음, 파일 시스템 캐시 사용률이 최저 워터마크 이하인지 검사합니다. 최저 워터마크 이하이면 릴리서는 파일 릴리스를 중지합니다. 그렇지 않으면 릴리서는 우선 순위에 따라 파일 릴리스를 계속합니다.

릴리서에서 10,000개의 대상을 모두 릴리스했으나 파일 시스템이 최저 워터마크 이상인 경우에는 10,000개의 새로운 대상을 처음부터 다시 식별합니다.

릴리서는 실행 가능한 대상을 찾지 못할 경우 종료됩니다. 예를 들어 파일에 아카이브 복사본이 없는 경우에도 릴리서가 종료될 수 있습니다. Sun StorEdge SAM-FS 파일 시스템은 1분 경과 후 릴리서를 다시 시작합니다.

용어 정의

이 절에서는 이 장에서 사용된 용어에 대해 설명합니다.

나이

age 개념은 주어진 이벤트로부터 지금까지 경과한 시간을 의미합니다. 파일의 *inode*는 릴리스가 사용하는 다음 시간을 추적합니다.

- 상주 변경 시간
- 수정된 날짜 시간
- 액세스한 날짜 시간

-D 옵션과 함께 `sls(1)` 명령을 사용하면 이러한 시간을 볼 수 있습니다. 각 시간에는 이에 상응하는 나이가 있습니다. 예를 들어 현재 시각이 10:15 a.m.인 경우, 변경 시각이 10:10 a.m.인 파일의 데이터 변경 나이는 5분입니다. `sls(1)` 옵션에 대한 자세한 내용은 `sls(1)` 매뉴얼 페이지를 참조하십시오.

대상

*candidate*는 릴리스할 수 있는 파일입니다. 파일이 대상이 될 수 없는 이유는 다음과 같습니다.

- 파일이 이미 오프라인 상태입니다.
- 파일이 아카이브되지 않았습니다.
- `archiver.cmd` 명령 파일에서 해당 파일에 대해 `-norelease` 속성을 지정하여 필요한 복사본이 아직 만들어지지 않았습니다.
- 파일이 손상된 것으로 표시되어 있습니다.
- 파일이 일반 파일이 아닙니다. 디렉토리, 블록, 문자 특수 파일 또는 파이프입니다.
- 아카이버가 추가 복사본을 만들기 위해 파일을 스테이지하고 있습니다. 먼저 아카이버가 파일을 스테이지 한 후에 파일을 릴리스할 수 있습니다.
- 파일의 나이가 음수입니다. 이러한 현상은 시간 설정이 잘못된 NFS 클라이언트에서 흔히 발생합니다.
- 파일은 릴리스할 수 없음으로 표시됩니다. `release(1) -n` 명령을 사용하여 이를 지정할 수 있습니다.
- 파일이 스테이지된 시간이 설정된 최소 상주 시간 이하입니다. 자세한 내용은 144 페이지의 "최소 상주 시간 지정 `min_residence_age`"를 참조하십시오.

- 파일이 부분 릴리스를 위해 `release(1)` 명령의 `-p` 옵션을 사용해 플래그되었고 이미 부분적으로 릴리스되었습니다.
- 파일이 너무 작습니다.

우선 순위

우선 순위란 해당 대상의 숫자 속성에 적용되는 사용자 제공 가중치에 따른 대상 파일의 순위를 나타내는 수치를 말합니다. 전체 우선 순위는 나이 우선 순위 및 크기 우선 순위 두 가지입니다.

우선 순위가 높은 대상 파일부터 차례로 릴리스됩니다.

가중치

가중치는 우선 순위를 계산할 때 필요한 파일 속성은 포함시키고 필요하지 않은 파일 속성은 제외하도록 조정하는 수치. 예를 들어 크기 가중치가 0으로 설정되면 파일의 크기 속성은 우선 순위 계산에서 제외됩니다. 가중치는 0.0에서 1.0까지의 부동 소수점 값입니다.

부분 릴리스

파일의 시작 부분만 디스크 캐시에 남아있고 나머지 부분은 릴리스되도록 지정하면 파일을 *부분적으로 릴리스*할 수 있습니다. 예를 들어 파일의 시작 부분을 읽는 `filemgr(1)`과 같은 유틸리티를 사용하는 경우에 부분 릴리스를 유용하게 사용할 수 있습니다.

부분 릴리스 및 부분 스테이지

릴리스 및 스테이지는 상호 보완적인 프로세스입니다. 파일을 아카이브한 후에 온라인 디스크 캐시에서 완전히 릴리스하거나 파일의 시작 부분(*stub*)은 디스크 캐시에 남아있고 나머지 부분이 릴리스되도록 사이트에서 지정할 수 있습니다. 파일을 부분적으로 릴리스할 수 있으면 파일을 스테이지하지 않고도 파일의 `stub`에 있는 데이터에 바로 액세스할 수 있습니다.

시스템 관리자는 파일 시스템이 마운트될 때 온라인에 남아있는 기본적인 부분 릴리스 크기와 `stub`의 최대 크기를 모두 지정할 수 있습니다. 시스템 관리자는 다음과 같이 이를 `mount(1M)` 명령에 설정할 수 있습니다.

- `-o partial=n` 옵션을 지정하여 파일 stub의 기본 크기(n)를 온라인 상태로 설정합니다. `-o partial=n` 설정은 `-o maxpartial=n` 설정보다 작거나 동일해야 합니다. 최소 설정은 `-o partial=8`킬로바이트입니다. 기본 설정은 `-o partial=16`킬로바이트입니다.
- `-o maxpartial=n` 옵션을 지정하여 파일 stub의 최대 크기(n)를 온라인 상태로 설정합니다. 온라인에 남아있는 파일 stub의 크기를 제한하려면 `-o maxpartial=n` 옵션을 사용하여 온라인에 남아있는 최대 stub과 같은 크기를 지정합니다. 부분 릴리스 기능을 비활성화하려면 `-o maxpartial=0`을 지정합니다.

사용자는 `release(1)` 명령에 `-p` 옵션을 지정하거나 `sam_release(3)` 라이브러리 루틴에 `p` 옵션을 지정하여 파일의 기본 stub 크기를 지정할 수 있습니다. 다양한 유형의 파일이나 애플리케이션에 대해 다양한 크기의 파일 stub을 지정하려면 사용자는 `release(1)` 명령에 `-s` 옵션을 지정하거나 `sam_release(3)` 라이브러리 루틴에 `s` 옵션을 지정할 수 있습니다. `-s` 및 `s` 값은 파일 시스템이 마운트될 때 `mount(1M)` 명령에서 사용하는 `-o maxpartial` 값보다 작아야 합니다.

또 다른 마운트 옵션인 `-o partial_stage=n`을 사용하면 시스템 관리자가 파일의 나머지 부분을 스테이지하기 전에 읽어야 하는 부분 릴리스 stub의 크기를 결정할 수 있습니다. 즉, `-o partial_stage=n` 크기 지정 이상을 읽으면 파일 스테이지가 시작됩니다.

기본적으로 `-o partial_stage=n` 옵션은 부분 릴리스 stub의 크기로 설정됩니다. 이 값은 구성할 수 있으나 다음과 같이 파일 스테이지에 영향을 줍니다.

- `-o partial_stage=n` 옵션을 부분 릴리스 stub의 크기로 설정하면 기본 작동에 따라 애플리케이션이 부분 릴리스 stub의 끝에 도달해야 파일이 스테이지됩니다. stub의 끝에 도달할 때까지 기다리면 파일의 나머지 부분에 대한 액세스가 지연됩니다.
- `-o partial_stage=n` 옵션이 부분 릴리스 stub보다 작은 값으로 설정되면 다음과 같은 상황이 발생합니다. 애플리케이션이 `-o partial_stage=n` 옵션이 설정한 임계값을 지난 후에 파일의 나머지 부분이 스테이지됩니다. 이렇게 되면 파일 데이터의 나머지 부분에 대한 액세스 지연 가능성이 낮아집니다.

예제. 다음 옵션을 사용한다고 가정합니다.

- `-o partial_stage=16` (즉, 16킬로바이트)
- `-o partial=2097152` (즉, 2기가바이트)
- `-o maxpartial=2097152` (즉, 2기가바이트)

`filemgr(1)` 프로그램을 사용하고, 이 프로그램에서는 파일의 처음 8킬로바이트를 읽습니다. 파일은 스테이지되지 않습니다. VOD(video-on-demand) 프로그램에서 동일한 파일을 읽고 처음 16킬로바이트 이상을 읽으면 파일이 스테이지됩니다. 아카이브 테이프가 마운트되고 위치가 지정되는 동안 애플리케이션은 디스크 데이터의 2기가바이트를 계속 읽습니다. VOD 프로그램에서 파일 데이터를 2킬로바이트 이상 읽으면 애플리케이션은 스테이지 작업 직후 읽기를 시작합니다. 애플리케이션이 부분 파일 데이터를 읽는 동안 테이프 마운트와 위치 지정이 완료되기 때문에 애플리케이션은 대기하지 않습니다.

몇 가지 명령행 옵션에 따라 파일의 부분 릴리스 표시 여부가 결정됩니다. 일부 옵션은 시스템 관리자가 활성화하고, 일부 옵션은 개별 사용자가 활성화할 수 있습니다. 다음 섹션에서는 다양한 유형의 사용자가 설정할 수 있는 릴리스 특성을 설명합니다.

시스템 관리자 옵션 요약

시스템 관리자는 파일 시스템이 마운트되면 부분 릴리스의 최대 값과 기본 값을 변경할 수 있습니다. 표 5-1의 mount(1M) 옵션은 부분 릴리스에 영향을 줍니다. mount(1M) 명령에 대한 자세한 내용은 mount_samfs(1M) 매뉴얼 페이지를 참조하십시오.

표 5-1 부분 릴리스 마운트 옵션

mount(1M) 옵션	결과
-o maxpartial= <i>n</i>	파일에 부분 릴리스 표시가 된 경우에 온라인 디스크 캐시에 남아있을 수 있는 최대 공간(단위: 킬로바이트)을 결정합니다. 최대 값은 2,097,152 킬로바이트, 즉 2킬로바이트입니다. 최소 값은 0으로 이 값에서는 파일이 부분 릴리스되지 않습니다. -o maxpartial=0으로 지정하면 부분 릴리스 기능은 비활성화되고, 릴리스된 파일은 완전 릴리스되며, 디스크 캐시에 남아있는 파일의 부분이 없습니다. 파일 시스템을 마운트한 후에는 이 옵션에 지정된 값을 변경할 수 없습니다. 기본적으로 <i>n</i> 인수는 16으로 설정됩니다. 이 설정을 사용하면 디스크의 최대량이 16 KB인 부분 릴리스에 파일을 표시할 수 있습니다.
-o partial= <i>n</i>	사용자가 release(1) 명령의 -p 옵션을 사용해 파일에 부분 릴리스 표시를 하는 경우 디스크 캐시 잔여 기본 공간(단위: 킬로바이트)을 설정합니다. <i>n</i> 인수는 8 이상이어야 하지만 -o maxpartial= <i>n</i> 옵션에 지정된 값이 될 수도 있습니다. 일부 애플리케이션에서는 전체 파일에 액세스하지 않아도 작업을 완료할 수 있기 때문에 이 옵션을 사용하면 필요한 파일의 시작 부분을 애플리케이션에서 사용할 수 있습니다. 이밖에도 이 옵션을 사용하면 파일이 불필요하게 스테이지되지 않습니다. 기본 값은 -o partial=16입니다.
-o partial_stage= <i>n</i>	부분 릴리스된 파일에 액세스할 때 파일의 <i>n</i> 바이트를 읽은 후에 아카이브 매체에서 전체 파일을 스테이지하도록 지정합니다. 이 값은 대개 -o partial 설정 값 이하로 설정됩니다. <i>n</i> 에는 0부터 -o maxpartial 지정까지의 정수 값을 지정합니다. 기본적으로 이 값은 16으로 설정되거나 -o partial 옵션에 지정된 값입니다.
-o stage_n_window= <i>n</i>	한번에 <i>n</i> 으로 스테이지되는 데이터의 양을 지정합니다. <i>n</i> 에 대해 64에서 2,048,000 까지 정수를 지정합니다. 기본값은 256 KB입니다. 이 옵션은 stage -n 속성이 설정된 파일에만 적용됩니다.

사용자 옵션 요약

시스템 관리자는 파일 릴리스 후에 디스크 캐시에 남아있을 수 있는 파일 크기의 최대 값과 기본 값을 설정합니다. 시스템 관리자는 또한 특정 파일 시스템에 대한 부분 릴리스 기능의 활성화 여부를 결정합니다.

그러나 사용자는 `release(1)` 명령과 `sam_release(3)` 라이브러리 루틴을 사용해 다른 릴리스 속성을 설정하고 부분 릴리스 표시가 되는 파일을 지정할 수 있습니다. 부분 릴리스 속성을 결정하는 명령과 라이브러리 옵션은 표 5-3에서 설명합니다.

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

표 5-2 사용자 릴리스 옵션

옵션	결과
<code>release(1)</code> 명령 및 <code>-p</code> 옵션 또는 <code>sam_release(3)</code> 라이브러리 루틴 및 <code>p</code> 옵션	<code>-p</code> 및 <code>p</code> 옵션을 사용하면 부분 릴리스 지정 파일에 표시를 할 수 있습니다. 이러한 옵션을 사용하면 파일이 릴리스된 후에 온라인 디스크 캐시에 남아있는 파일의 양은 파일이 상주하는 파일 시스템이 마운트될 때 설정된 <code>-o partial=n</code> 옵션의 값에 따라 달라집니다. 온라인에 남아있는 바이트 수를 지정할 때는 이러한 옵션을 사용할 수 없습니다.
<code>release(1)</code> 명령 및 <code>-s</code> <code>partial_size</code> 옵션 또는 <code>sam_release(3)</code> 라이브러리 루틴 및 <code>s</code> 옵션	<code>-s</code> 및 <code>s</code> 옵션을 사용하면 부분 릴리스 지정 파일에 표시를 하고 온라인 디스크 캐시에 남아있는 파일의 양을 지정할 수 있습니다. <code>-s</code> 또는 <code>s</code> 옵션에 대한 인수는 온라인에 남아있는 양을 킬로바이트 단위로 지정합니다. 파일 시스템을 마운트할 때 온라인에 남아있는 파일의 양을 <code>-o maxpartial=n</code> 값에 지정된 양보다 많도록 지정할 수 없습니다. 사용자가 지정한 값이 파일 시스템의 값보다 큰 경우에는 파일 시스템의 값을 사용하고 사용자 지정 사양은 무시됩니다.

releaser.cmd 파일

`/etc/opt/SUNWsamfs/releaser.cmd` 파일은 사이트별 릴리스 작업을 지정하는 명령행으로 구성됩니다. `releaser.cmd` 파일에는 릴리스 우선 순위 설정 명령, 로그 파일 지정 명령 및 기타 작업에 대한 명령이 포함될 수 있습니다.

다음 섹션에서는 `releaser.cmd` 명령에 대해 설명합니다.

- 140 페이지의 "나이 및 크기 관련 릴리스 우선 순위 명령 지정 `weight_age`, `weight_age_access`, `weight_age_modification` 및 `weight_age_residence`"
- 143 페이지의 "개별 파일 시스템의 명령 지정 `fs`"
- 144 페이지의 "디버그 명령 지정 `no_release` 및 `display_all_candidates`"
- 144 페이지의 "최소 상주 시간 지정 `min_residence_age`"
- 145 페이지의 "로그 파일 지정 `logfile`"
- 147 페이지의 "재아카이브된 파일의 릴리스 방지 `rearch_no_release`"
- 147 페이지의 "릴리서 대상 목록의 크기 조정: `list_size`"

이러한 명령에 대한 자세한 사항은 `releaser.cmd(4)` 매뉴얼 페이지를 참조하십시오.

나이 및 크기 관련 릴리스 우선 순위 명령 지정 `weight_age`, `weight_age_access`, `weight_age_modification` 및 `weight_age_residence`

`releaser.cmd` 파일에 정의된 명령에 의해 결정된 우선 순위에 따라 파일 시스템에서 파일을 릴리스합니다. 파일 나이와 크기를 모두 고려합니다. 기본적으로 사이트에서는 나이와 크기가 큰 파일부터 차례로 릴리스하고 나머지는 디스크에 남겨둡니다. 다음 섹션에서는 파일 시스템에서 파일의 릴리스 우선 순위를 결정할 때 릴리서의 파일 나이 및 크기 고려 방식에 대해 설명합니다.

릴리서 명령에 대한 자세한 내용은 `releaser.cmd(4)` 매뉴얼 페이지를 참조하십시오.

파일 나이

릴리서에서는 나이와 관련된 파일 릴리스 우선 순위의 구성 요소를 결정할 때 다음과 같은 나이를 고려합니다.

- 파일이 마지막으로 액세스된 이후의 나이
- 파일이 마지막으로 수정된 이후의 나이
- 파일에서 디스크 캐시 상주를 변경한 이후의 나이

경우에 따라서는 파일의 수정 나이보다 액세스 나이를 우선 순위에 둘 수도 있습니다. 또는 가장 최근에 액세스한 시간, 수정 시간 및 상주 변경 시간에서 파생된 단순 나이를 우선 순위에 둘 수도 있습니다.

기본적으로 파일의 나이는 다음과 같은 세 가지 파일 나이 중에 가장 최근의 것으로 합니다.

- 파일 액세스 나이
- 파일 수정 나이
- 파일 상주 나이

명령을 통해 가중치를 둔 나이 우선 순위를 파일의 릴리스 우선 순위 계산에 사용할 수 있습니다.

코드 예 5-1은 나이 우선 순위 명령의 형식을 표시합니다.

코드 예 5-1 나이 우선 순위 명령 형식

```
weight_age = float
weight_age_access = float
weight_age_modification = float
weight_age_residence = float
```

- `weight_age` 명령은 파일의 기본 나이(파일의 액세스, 수정 또는 상주 나이 중 최소 값)를 가중치로 지정합니다. `float`에 대해 다음 범위에 있는 부동 소수점 수를 지정합니다. $0.0 \leq float \leq 1.0$. 기본적으로 `float = 1.0`입니다.

이 명령은 `weight_age_residence`, `weight_age_modify` 또는 `weight_age_access` 명령과 함께 지정할 수 없습니다.

- `weight_age_residence`, `weight_age_modify` 및 `weight_age_access` 명령은 파일의 세 가지 나이 중에서 한 가지, 두 가지 또는 세 가지 나이의 조합에 따라 나이를 결정하도록 지정합니다. `float`에 대해 다음 범위에 있는 부동 소수점 수를 지정합니다. $0.0 \leq \text{부동} \leq 1.0$. 기본적으로 `float = 1.0`입니다.

이러한 명령은 `weight_age` 명령과 함께 지정할 수 없습니다.

`weight_age_residence`, `weight_age_modify` 및 `weight_age_access` 명령을 사용하는 경우, 파일의 나이 관련 우선 순위는 세 가지 나이의 조합에 따라 계산됩니다. 먼저 각 파일의 가능 나이에 대한 파일 나이 데이터를 수집합니다. 다음으로 파일 나이 데이터에 `releaser.cmd` 파일에서 지정한 가중치를 곱합니다. 마지막으로 코드 예 5-2에 나와있는 것처럼 각 가중치를 곱한 나이 데이터의 결과를 합하여 파일의 나이 관련 우선 순위를 계산합니다.

코드 예 5-2 우선 순위 계산

```
파일 액세스 나이 * weight_age_access
+ 파일 수정 나이 * weight_age_modification
```

코드 예 5-2 우선 순위 계산 (계속)

```
+ 파일 상주 나이 * weight_age_residence
-----
= age_related_priority
```

예제. 코드 예 5-3에서 `releaser.cmd` 파일의 행에서는 파일의 릴리스 우선 순위를 계산할 때 파일의 상주 나이만을 고려하고 수정 나이와 액세스 나이는 무시하도록 지정함을 보여줍니다.

코드 예 5-3 `releaser.cmd` 파일 일부

```
weight_age_residence = 1.0
weight_age_modify = 0.0
weight_age_access = 0.0
```

파일의 나이 관련 우선 순위를 계산한 후에는 파일의 크기 관련 우선 순위를 곱합니다. 크기 관련 우선 순위는 다음 섹션에서 설명하는 방식으로 계산합니다.

파일 크기

릴리서는 파일 릴리스 우선 순위의 크기 관련 구성 요소를 결정할 때 파일의 크기를 고려합니다. 파일의 크기(단위: 4킬로바이트 블록)에 `weight_size` 명령에 지정된 가중치를 곱해 파일 릴리스 우선 순위의 크기 관련 구성 요소를 구합니다.

`weight_size` 명령의 형식은 다음과 같습니다.

```
weight_size = float
```

*float*에 대해 다음 범위에 있는 부동 소수점 수를 지정합니다. $0.0 \leq float \leq 1.0$. 기본적으로 *float* = 1.0입니다.

예제. 코드 예 5-4는 파일의 우선 순위 계산 시 파일의 크기가 `samfs1` 및 `samfs2` 파일 시스템의 모든 파일에 대해 무시되도록 지정하는 `releaser.cmd` 파일을 표시합니다.

코드 예 5-4 `releaser.cmd` 파일

```
# releaser.cmd file
logfile = /var/adm/default.releaser.log
weight_size = 0.0
#
fs = samfs1
weight_age = 1.0
logfile = /var/adm/samfs1.releaser.log
#
fs = samfs2
```

```
weight_age_modify = 0.3
weight_age_access = 0.03
weight_age_residence = 1.0
logfile = /var/adm/samfs2.releaser.log
```

개별 파일 시스템의 명령 지정 fs

releaser.cmd 파일의 fs = *family_set_name* 명령을 사용하여 fs = 명령을 따르는 명령들이 명명된 파일 시스템에만 적용됨을 표시할 수 있습니다. 이 명령의 형식은 다음과 같습니다.

```
fs = family_set_name
```

*family_set_name*에는 mcf 파일에 패밀리 세트의 이름을 지정합니다.

첫 번째 fs = 명령 이전의 명령은 전역 명령이므로 모든 파일 시스템에 적용됩니다. fs = 명령 이후의 명령은 전역 명령보다 우선합니다. 이 장에서 설명하는 명령은 전역 명령 또는 하나의 파일 시스템에 대한 명령으로 사용할 수 있습니다.

releaser.cmd(4) 매뉴얼 페이지에 fs = 명령의 예제가 나와있습니다.

디버그 명령 지정 no_release 및 display_all_candidates

no_release 및 display_all_candidates 명령은 릴리서를 조정하거나 디버그할 때 유용하게 사용할 수 있습니다. 이러한 명령은 다음과 같습니다.

- no_release 명령을 사용하면 파일이 온라인 디스크 캐시에서 제거되지 않습니다. 이 명령을 사용해 파일을 릴리스하지 않고도 releaser.cmd의 명령을 확인할 수 있습니다. 이 명령의 형식은 다음과 같습니다.

```
no_release
```

- display_all_candidates 명령을 사용하면 로그 파일에 모든 대상의 이름을 작성할 수 있습니다. 이 명령의 형식은 다음과 같습니다.

```
display_all_candidates
```

이러한 명령은 릴리서가 파일 시스템에서 대상을 실제로 릴리스하지 않고 대상의 이름을 로그 파일에 작성하기 때문에 디버그할 때 유용하게 사용할 수 있습니다.

최소 상주 시간 지정 min_residence_age

min_residence_age 명령을 사용하면 파일이 릴리스 대상 대상이 되기 전에 파일 시스템에 상주하는 최소 시간을 지정할 수 있습니다. 이 명령의 형식은 다음과 같습니다.

```
min_residence_age = time
```

time에는 초 단위로 시간을 지정합니다. 기본 시간은 600초, 즉 10분입니다. 실제로는 최소 또는 최대 time 설정이 없습니다.

로그 파일 지정 logfile

logfile 명령이 `releaser.cmd` 파일에 지정되면, 릴리서는 파일의 활동 로그를 지정된 파일 이름에 추가하거나 이름이 없는 경우에는 파일 이름을 만듭니다. 이 명령의 형식은 다음과 같습니다.

```
logfile = filename
```

*filename*에 로그 파일의 이름을 지정합니다.

코드 예 5-5는 예제 로그 파일을 나타냅니다 (참고: 일부 행은 지면 관계상 줄바꿈하였습니다).

코드 예 5-5 릴리서 로그 파일 예제

```
Releaser begins at Wed Apr 28 17:29:06 1999
inode pathname          /sam1/.inodes
low-water mark          24%
weight_size             1
weight_age              1
fs equipment ordinal    1
family-set name         samfs1
started by sam-amld?    yes
release files?          yes
display_all_candidates? no
---before scan---
blocks_now_free:        3481504
lwm_blocks:             3729362
---scanning---
10501 (R: Wed Apr 21 18:47:50 CDT 1999) 10001 min, 500 blks /sam1/testdir0/filevp
10500 (R: Wed Apr 21 18:48:10 CDT 1999) 10000 min, 500 blks /sam1/testdir0/filewq
...
---after scan---
blocks_now_free:        3730736
lwm_blocks:             3729362
archnodrop: 0
already_offline: 0
bad_inode_number: 0
damaged: 0
extension_inode: 0
negative_age: 0
nodrop: 1
not_regular: 9
number_in_list: 675
released_files: 202
too_new_residence_time: 0
too_small: 2
```

코드 예 5-5

릴리서 로그 파일 예제 (계속)

```

total_candidates: 675
total_inodes: 1376
wrong_inode_number: 0
zero_arch_status: 689
zero_inode_number: 0
zero_mode: 0
CPU time: 2 초.
Elapsed time: 10 초.
Releaser ends at Wed Apr 28 17:29:16 1999

```

releaser(1M) 매뉴얼 페이지에서는 이 로그 파일에 포함된 정보에 대해 설명합니다. 각 릴리서가 실행되면서 로그 크기가 증가하기 때문에 로그 크기를 줄이거나 logfile 키워드를 생략합니다.

코드 예 5-6은 ---after scan--- 행 아래의 통계에는 다음과 같은 수식 관계가 존재함을 나타냅니다.

코드 예 5-6

코드 예 5-5의 ---after scan---으로부터의 수식 관계

```

total_inodes = wrong_inode_number +
zero_inode_number +
zero_mode +
not_regular +
extension_inode +
zero_arch_status +
already_offline +
damaged +
nodrop +
archnodrop +
too_new_residence_time +
too_small +
negative_age +
total_candidates
released_files = total_candidates

```

재아카이브된 파일의 릴리스 방지

research_no_release

기본적으로 재아카이브 표시된 파일은 릴리스됩니다. `research_no_release` 명령이 `releaser.cmd(4)` 파일에 지정된 경우 릴리서는 재아카이브로 표시된 파일을 릴리스하지 않습니다. 이 명령의 형식은 다음과 같습니다.

```
research_no_release
```

릴리서 대상 목록의 크기 조정: list_size

`list_size` 명령을 사용하여 릴리서 대상의 수를 지정할 수 있습니다. 최저 워터 마크에 도달하는 데 요구되는 파일의 수를 릴리스하기 전에 릴리서가 여러 파일 시스템을 스캔하는 경우 기본값 10,000 보다 큰 레벨로 이 값을 올릴 수도 있습니다. 여러 작은 파일이 있는 파일 시스템에서 가능한 일입니다. 릴리서 로그 파일에서 릴리서 작업에 대한 정보를 얻을 수 있습니다. 이 명령의 형식은 다음과 같습니다.

```
list_size = number
```

수에 대해서 $10 \leq \text{number} \leq 2,147,483,648$ 과 같은 정수를 지정합니다.

릴리스에서 archiver.cmd 파일의 역할

`archiver.cmd` 파일의 명령 대부분은 아카이브에 영향을 주지만 아카이브 세트 할당 명령을 사용하면 아카이브 세트의 모든 파일에 적용되는 릴리스 속성을 지정할 수 있습니다.

아카이브 세트 할당 명령의 형식은 다음과 같습니다.

```
archive_set_name path [search_criteria ...] directives ...
```

표 5-3에서는 릴리스와 관련된 명령을 보여줍니다.

표 5-3 아카이브 세트 할당 명령

명령	결과
-release a	아카이브 세트의 파일이 첫 번째 아카이브 복사본을 만든 후에 릴리스되도록 지정합니다. 각 파일의 아카이브 복사본을 두 개 이상 만드는 경우에는 이 옵션을 사용하지 마십시오. 이러한 경우에는 첫 번째 복사본이 두 번째 복사본을 만들기 위해 스테이지됩니다.
-release n	아카이브 세트의 파일이 릴리스되지 않도록 지정합니다.
-release p	아카이브 세트의 파일이 아카이브 후에 부분적으로 릴리스되도록 지정합니다.

이러한 명령과 기타 archiver.cmd 명령에 대한 자세한 내용은 53 페이지의 "아카이브"를 참조하십시오.

릴리서 구성

캐시에 있는 파일의 특성이 현재 사이트에 적합한지 확인해야 합니다. 일부 KB만 스테이지하는 경우 테이프를 로드하는 것은 낭비이므로 시스템을 바이어스하여 캐시에 작은 파일을 유지할 수도 있습니다. 코드 예 5-7은 가장 큰 파일을 먼저 릴리스하는 데 releaser.cmd 파일에서 사용하는 명령을 표시합니다.

코드 예 5-7 가장 큰 파일 먼저 릴리스하는 명령

```
weight_size = 1.0  
weight_age = 0.0
```

아니면 최근에 수정된 파일이 곧 다시 수정될 수 있기 때문에 이러한 파일을 캐시에 보유해야 하는 경우가 있습니다. 이렇게 하면 수정을 활성화하기 위해 파일을 스테이지할 때 생성되는 오버헤드를 피할 수 있습니다. 이 경우, 나이 가중치의 두 번째 세트를 사용합니다. 코드 예 5-8은 가장 오래 전에 수정된 파일에서 시작하여 가장 최근에 수정된 파일까지의 엄격한 순서로 파일에 가중치를 가하도록 releaser.cmd 파일에서 사용하는 명령입니다.

코드 예 5-8 가장 오래 전에 수정된 파일 먼저 릴리스하는 명령

```
weight_size = 0.0  
weight_age_access = 0.0  
weight_age_modify = 1.0  
weight_age_residence = 0.0
```

그러나 다음 예에서와 같이 대부분의 경우에는 이렇게 간단하지 않습니다.

예제 1. 가장 큰 파일을 먼저 릴리스한다고 가정합니다. 크기가 같은 작은 파일이 수 백 개 있고 큰 파일이 몇 개 있습니다. 작은 파일의 크기를 합한 것이 큰 파일 하나의 크기보다 클 수도 있습니다. 결국 릴리서는 큰 파일을 모두 릴리스합니다. `weight_age = 0.0`가 지정된 경우에는 작은 파일의 크기와 릴리스 우선 순위가 모두 같기 때문에 릴리서는 무작위로 작은 파일을 릴리스합니다.

이 경우에는 `weight_age = 0.01`을 릴리스 기준으로 설정합니다. 릴리서는 크기가 같은 파일 두 개중에 더 오래된 것을 먼저 릴리스합니다.

예제 2. 이 예제에서는 보다 효과적으로 가장 큰 파일을 먼저 릴리스하도록 지정하는 방법을 보여줍니다.

`weight_size = 1.0` 및 `weight_age = 0.01`을 설정합니다.

이러한 명령을 사용하면 최근에 액세스한 큰 파일보다 오래전에 액세스한 작은 파일이 우선 대상이 되기 때문에 크기 우선 정책에 위배됩니다. `weight_age`를 `weight_size`보다 작게 만들면 이 결과를 바꿀 수 있습니다. 예를 들어 이전 설정 내용에 따라 100분 전에 스테이지된 4킬로바이트 파일과 방금 스테이지된 8킬로바이트 파일 모두 릴리스 우선 순위가 같습니다.

릴리서는 릴리스할 파일을 무작위로 선택합니다. 릴리서에서 4킬로바이트 파일을 선택한 경우에는 크기 우선 정책에 위배됩니다. `weight_age`를 훨씬 작게 설정(예: 0.001)하면 이 결과를 바꿀 수 있습니다. 1,000분 전에 스테이지된 4킬로바이트 파일의 경우, 방금 스테이지된 8킬로바이트 파일과 우선 순위가 같게 됩니다.

`no_release` 및 `display_all_candidates` 명령을 사용해 릴리서를 수동으로 실행하여 우선 순위 가중치를 조정할 때 사용할 우선 순위에 따른 대상 목록을 만들 수 있습니다.

릴리서 수동 실행

경우에 따라서는 릴리서를 수동으로 실행해야 하는 경우가 있습니다. 이러한 경우에는 릴리서가 도달할 수 있는 최저 워터마크와 파일 시스템의 마운트 지점을 알아야 합니다.

예를 들어 47%까지 찰 때까지 `/sam1` 파일 시스템의 파일을 릴리스하려면 루트로 로그인하여 다음을 입력합니다.

```
# /opt/SUNWsamfs/sbin/sam-releaser /sam1 47 1.0
```

최종 인수 `weight-size`는 `releaser.cmd` 파일의 `weight-size` 명령보다 우선 순위가 낮습니다. 릴리서 실행 시, `releaser.cmd` 파일에 지정된 경우 화면과 릴리서 로 그 파일에 정보를 기재합니다. 자세한 내용은 `sam-releaser(1M)` 매뉴얼 페이지를 참조하십시오.

릴리서 문제 해결

릴리서가 파일을 릴리스하지 않는다면 몇 가지 원인이 있을 수 있습니다. 다음과 같은 원인을 예로 들 수 있습니다.

- 먼저 파일을 아카이브해야 릴리스할 수 있습니다. 아카이브 복사본이 없을 수 있습니다. 자세한 내용은 130 페이지의 "파일이 아카이브되지 않는 원인"을 참조하십시오.
- 아카이버에서 파일이 릴리스되지 않도록 요청하였습니다. 이러한 현상은 다음과 같은 상황에서 발생할 수 있습니다.
 - 아카이버에서 오프라인 파일을 방금 스테이지하여 추가 복사본을 만들었습니다.
 - `archiver.cmd` 파일에 `-norelease` 명령이 설정되고 `-norelease`로 플래그된 복사본이 모두 아카이브되지 않았습니다. 릴리서 요약 결과에는 `archnodrop` 플래그가 설정된 파일의 총 수가 표시됩니다.
- 파일이 부분 릴리스로 설정되고, 파일 크기가 디스크 할당 단위(DAU) 크기(블록 크기)로 반올림된 부분 크기 이하입니다.
- 파일에서 마지막 `min_residence_age` 분에 상수를 변경하였습니다.
- 디렉토리나 파일이 릴리스되지 않도록 `release -n` 명령을 사용하였습니다.
- `archiver.cmd` 파일에서 과도한 디렉토리나 파일에 대해 `-release n` 옵션을 설정하였습니다.
- 릴리서 최고 워터마크가 지나치게 높게 설정되어 자동 릴리스가 너무 늦게 시작됩니다. `samu(1M)` 유틸리티의 `m` 디스플레이에서나 `SAM-QFS Manager`를 사용해 이를 확인하고 값을 낮춥니다.
- 릴리서 최저 워터마크가 지나치게 높게 설정되어 자동 릴리스가 너무 빨리 멈춥니다. `samu(1M)` 유틸리티의 `m` 디스플레이에서나 `SAM-QFS Manager`를 사용해 이를 확인하고 값을 낮춥니다.
- 큰 파일이 사용 중입니다. 이러한 파일은 아카이브 나이에 도달하지 않아 아카이브 및 릴리스되지 않습니다.

스태이지

스태이지란 니어라인 또는 오프라인 저장소에서 온라인 저장소로 파일 데이터를 복사하는 프로세스를 말합니다. 스테이지 기능을 사용하면 파일을 바로 스테이지하거나 스테이지하지 않을 수 있으며, 부분 스테이지 지정 및 기타 스테이지 작업을 지정할 수 있습니다. 예를 들면 큰 파일에서 작은 레코드에 무작위로 액세스하는 애플리케이션에서 스테이지 비활성화 기능을 사용할 수 있으며, 스테이지를 활성화 기능을 사용하면 파일을 온라인으로 스테이지하지 않고도 아카이브 매체에서 직접 데이터에 액세스할 수 있습니다.

이 장은 Sun StorEdge SAM-FS 파일 스테이지 기능을 설명합니다. 다음 항목에 대해 설명합니다.

- 151 페이지의 "stager.cmd 파일"
- 158 페이지의 "스테이지에서 archiver.cmd 파일의 역할"
- 158 페이지의 "preview.cmd 파일을 사용한 미리보기 요청 우선 순위 지정"
- 163 페이지의 "총 미리보기 요청 우선 순위 계산"
- 163 페이지의 "미리보기 요청 우선 순위 방식 설정 방법"

stager.cmd 파일

stager.cmd 파일을 사용하면 스테이지의 작동을 지정할 수 있습니다. 이 파일의 전체 경로 이름은 /etc/opt/SUNWsamfs/stager.cmd입니다. 기본적으로 스테이지는 다음과 같은 작업을 수행합니다.

- 스테이지는 라이브러리의 모든 드라이브를 사용해 파일 스테이지를 시도합니다.
- 스테이지 버퍼 크기는 매체 유형에 따라 다르며, 스테이지 버퍼를 잠그지 않습니다.
- 로그 파일을 작성하지 않습니다.
- 최대 1000개의 스테이지 요청을 한 번에 활성화할 수 있습니다.

stager.cmd 파일을 사용하면 명령을 지정해 기본 작동을 변경할 수 있습니다. 이 섹션에서는 스테이지 명령에 대해 설명합니다. 스테이지 명령에 대한 자세한 내용은 stager.cmd(4) 매뉴얼 페이지를 참조하십시오.

157 페이지의 "예제 stager.cmd 파일"에서는 모든 명령을 설정한 후에 완료된 stager.cmd 파일을 보여줍니다.

코드 예 6-1은 이 장의 예제에서 사용되는 mcf 파일 예제를 표시합니다.

코드 예 6-1 이 장의 예제에서 사용된 mcf 파일

```
#
# Sun StorEdge SAM-FS file system configuration example
#
# Equipment      Eq Eq Family Dev Additional
# Identifier     Or Tp Set   St Parameters
# -----
samfs1           60 ms samfs1
/dev/dsk/c1t1d0s6 61 md samfs1 on
/dev/dsk/c2t1d0s6 62 md samfs1 on
/dev/dsk/c3t1d0s6 63 md samfs1 on
/dev/dsk/c4t1d0s6 64 md samfs1 on
/dev/dsk/c5t1d0s6 65 md samfs1 on
#
samfs2           2 ms samfs2
/dev/dsk/c1t1d0s0 15 md samfs2 on
/dev/dsk/c1t0d0s1 16 md samfs2 on
#
/dev/samst/c0t2d0 20 od -      on
/dev/samst/c1t2u0 30 rb dog  on /var/opt/SUNWsamfs/catalog/dogcat
/dev/samst/c1t5u0 31 od dog  on
/dev/samst/c1t6u0 32 od dog  on
/dev/rmt/0cbn     40 od -      on
/dev/samst/c1t3u1 50 rb bird on /var/opt/SUNWsamfs/catalog/birdcat
/dev/rmt/2cbn     51 tp bird  on
```

▼ stager.cmd 파일 작성 또는 수정하기 및 변경내용 전파하기

1. vi(1) 또는 다른 편집기를 사용하여 stager.cmd 파일을 엽니다.

이 파일의 전체 경로는 다음과 같습니다.

```
/etc/opt/SUNWsamfs/stager.cmd
```

이 파일에 포함할 수 있는 명령에 대한 정보는 다음 하위 절을 참조하십시오.

- 153 페이지의 "드라이브 수 지정"
 - 154 페이지의 "스테이지 버퍼 크기 지정"
 - 155 페이지의 "로그 파일 지정"
 - 157 페이지의 "스테이지 요청 횟수 지정"
2. stager.cmd 파일을 저장하고 닫습니다.
 3. config 옵션이 있는 samd(1M) 명령을 사용하여 파일의 변경 내용을 전파하고 시스템을 재시작합니다.

```
# samd config
```

드라이브 수 지정

기본적으로 스테이저는 파일을 스테이지할 때 사용 가능한 모든 드라이브를 사용합니다. 스테이저가 모든 드라이브를 사용하고 있다면 아카이버의 작업이 방해받을 수 있습니다. drives 명령은 스테이저가 사용할 수 있는 드라이브 수를 지정합니다. 이 명령의 형식은 다음과 같습니다.

```
drives = library count
```

표 6-1 drives 명령에 대한 인수

인수	의미
library	Sun StorEdge SAM-FS mcf 파일에 나타난 라이브러리의 패밀리 세트 이름.
count	사용 가능한 드라이브의 최대 수. 기본적으로 이 수는 해당 라이브러리에 대해 mcf 파일에 구성된 드라이브 수입니다.

예를 들어 다음과 같은 명령 행은 `dog` 패밀리 세트의 라이브러리에서 하나의 드라이브만 파일 스테이지에 사용하도록 지정합니다.

```
drives = dog 1
```

`mcf` 파일 작성에 대한 자세한 내용은 `mcf(4)` 매뉴얼 페이지를 참조하십시오.

스테이지 버퍼 크기 지정

기본적으로 스테이지되는 파일은 버퍼의 메모리에서 읽은 후에 아카이브 매체에서 온 라인 디스크 캐시로 다시 복원됩니다. `bufsize` 명령을 사용하면 기본 값이 아닌 버퍼 크기를 지정할 수 있으며 버퍼를 잠글 수도 있습니다. 이렇게 하면 성능을 향상시킬 수 있을뿐만 아니라 다양한 `buffer_size` 값을 시험해 볼 수 있습니다. 이 명령의 형식은 다음과 같습니다.

```
bufsize=media buffer_size [ lock ]
```

표 6-2 `bufsize` 명령에 대한 인수

인수	의미
<code>media</code>	<code>mcf(4)</code> 매뉴얼 페이지의 목록에서 아카이브 매체 유형을 지정합니다.
<code>buffer_size</code>	2부터 32까지 숫자를 지정합니다. 기본값은 4입니다. 이 값에 매체 유형에 대한 <code>dev_blksize</code> 값이 곱해지고 해당 결과 버퍼 크기가 사용됩니다. <code>dev_blksize</code> 는 <code>defaults.conf</code> 파일에 지정할 수 있습니다. <code>buffer_size</code> 에 지정된 수가 클수록 사용하는 메모리의 용량도 많아집니다. 이 파일에 대한 자세한 내용은 <code>defaults.conf(4)</code> 매뉴얼 페이지를 참조하십시오.
<code>lock</code>	<p><code>lock</code> 인수는 아카이버가 아카이브 복사본을 스테이지할 때 잠긴 버퍼를 사용하는지 여부를 나타냅니다. <code>lock</code>가 지정되면 스테거는 복사 작업 과정에서 메모리의 아카이브 버퍼에 파일 잠금을 설정합니다. 이렇게 하면 각 입출력 요청에 대해 버퍼의 잠금 및 잠금 해제 오버헤드를 피할 수 있고 시스템 CPU 시간을 줄일 수 있습니다.</p> <p><code>lock</code> 인수는 메모리 용량이 큰 대형 시스템인 경우에만 지정해야 합니다. 메모리가 충분하지 않으면 메모리 부족 현상이 발생할 수 있습니다.</p> <p><code>lock</code> 인수는 스테이지되는 파일에 대해 직접 입출력을 사용하는 경우에만 효과적으로 사용할 수 있습니다. 기본적으로 <code>lock</code>은 지정되지 않고 파일 시스템은 아카이브 대상을 포함한 모든 직접 입출력 버퍼에 대해 잠금을 설정합니다. 직접 입출력 사용에 대한 자세한 내용은 <code>setfa(1)</code> 매뉴얼 페이지, <code>sam_setfa(3)</code> 라이브러리 루틴 매뉴얼 페이지 또는 <code>mount_samfs(1M)</code> 매뉴얼 페이지의 <code>-o forcedirectio</code> 옵션을 참조하십시오.</p>

예를 들어 이 명령은 `stager.cmd` 파일에 다음과 같은 행으로 지정할 수 있습니다.

```
bufsize=od 8 lock
```

로그 파일 지정

Sun StorEdge SAM-FS 파일 시스템이 파일 스테이지 이벤트 정보를 모으고 로그 파일에 기록하도록 요청할 수 있습니다. `logfile` 명령은 스테이지가 로깅 정보를 기록할 수 있는 로그 파일을 지정합니다. 이 명령의 형식은 다음과 같습니다.

```
logfile=filename [ event ]
```

*filename*에는 전체 경로 이름을 지정합니다.

*event*에 대해 하나 이상의 스테이지 이벤트를 지정합니다. 두 개 이상의 *event*를 지정한 경우 스페이스를 사용하여 각 *event*를 분리합니다. 기본적으로 사용 가능한 이벤트는 다음과 같습니다. `finish cancel error`. 가능한 *event* 사양은 다음과 같습니다.

표 6-3 *event* 인수에 대한 키워드

<i>event</i>	작업
<code>all</code>	모든 스테이지 이벤트를 로그합니다.
<code>start</code>	파일에 대한 스테이지가 시작할 때 로그합니다.
<code>finish</code>	파일에 대한 스테이지가 끝날 때 로그합니다. 기본적으로 사용 가능합니다.
<code>cancel</code>	운영자가 스테이지를 취소할 때 로그합니다. 기본적으로 사용 가능합니다.
<code>error</code>	스테이지 오류를 로그합니다. 기본적으로 사용 가능합니다.

로그 파일이 지정된 경우에는 스테이지는 스테이지된 각 파일에 대해 로그 파일에 하나 이상의 행을 작성합니다. 이 행에는 파일 이름, 스테이지 날짜와 시간 및 `VSN` 등과 같은 정보가 포함됩니다. 예를 들어 다음 명령 행은 파일 `/var/adm/stage.log`를 지정합니다.

```
logfile=/var/adm/stage.log
```

에서는 스테이지 로그 파일의 예제를 보여줍니다.

코드 예 6-2

스테이지 로그 파일 예제

```
S 2003/12/16 14:06:27 dk disk01 e.76d 2557.1759 1743132 /saml/testdir0/filebu 1
root other root 0
F 2003/12/16 14:06:27 dk disk01 e.76d 2557.1759 1743132 /saml/testdir0/filebu 1
root other root 0
S 2003/12/16 14:06:27 dk disk02 4.a68 1218.1387 519464 /saml/testdir1/fileaq 1
root other root 0
S 2003/12/16 14:06:43 dk disk01 13.ba5 3179.41 750880 /saml/testdir0/filecl 1
root other root 0
F 2003/12/16 14:06:43 dk disk01 13.ba5 3179.41 750880 /saml/testdir0/filecl 1
root other root 0
S 2003/12/16 14:06:59 dk disk01 17.167b 1155.1677 1354160 /saml/testdir0/filedb
1 root other root 0
F 2003/12/16 14:06:59 dk disk01 17.167b 1155.1677 1354160 /saml/testdir0/filedb
1 root other root 0
S 2003/12/16 14:06:59 dk disk02 f.f82 3501.115 1458848 /saml/testdir1/filecb 1
root other root 0
S 2003/12/16 14:07:15 dk disk01 1f.473 1368.1419 636473 /saml/testdir0/fileed 1
root other root 0
S 2003/12/16 14:07:15 dk disk02 16.f15 3362.45 1065457 /saml/testdir1/filecz 1
root other root 0
S 2003/12/16 14:07:31 dk disk01 23.201d 3005.1381 556807 /saml/testdir0/fileeq
1 root other root 0
S 2003/12/16 14:07:47 dk disk01 26.c4d 2831.1113 1428718 /saml/testdir0/fileez
1 root other root 0
S 2003/12/16 14:07:47 dk disk02 1b.835 3736.59 1787855 /saml/testdir1/filedp 1
root other root 0
```

표 6-4에 표시된 대로 스테이지 로그 파일은 아홉 개의 필드로 분리되는 정보 행으로 구성됩니다. 표 6-4는 스테이지 로그 파일 필드의 내용을 설명합니다.

표 6-4 스테이지 로그 파일 필드

필드	내용 설명
1	스테이지 작업. S는 시작을 나타냅니다. C는 취소된 경우를 나타냅니다. E는 오류를 나타냅니다. F는 마침을 나타냅니다.
2	yyyy/mm/dd 형식의 스테이지 작업 날짜.
3	hh:mm:ss 형식의 스테이지 작업 시간.
4	아카이브 매체 유형. 매체 유형에 대한 자세한 내용은 mcf(4) 매뉴얼 페이지를 참조하십시오.
5	VSN.
6	매체(tar(1) 파일)에서 아카이브 파일의 실제 시작 위치와 아카이브 파일의 16진수 파일 오프셋.

표 6-4 스테이지 로그 파일 필드 (계속)

필드	내용 설명
7	Inode 번호 및 생성 번호. 생성 번호란 inode 번호가 재사용되므로 고유성을 위해 inode 번호 이외에 추가로 사용하는 번호를 말합니다.
8	파일 길이.
9	파일 이름.
10	아카이브 복사본 번호.
11	파일의 사용자 ID.
12	파일의 그룹 ID.
13	요청자의 그룹 ID.
14	파일이 스테이징된 드라이브의 장비 서수.

스테이지 요청 횟수 지정

`maxactive` 명령을 사용해 한 번에 활성화될 수 있는 스테이지 요청 횟수를 지정할 수 있습니다. 이 명령의 형식은 다음과 같습니다.

```
maxactive=number
```

기본적으로 *number*는 4000입니다. 허용된 최소 수는 1입니다.

예를 들어 다음 명령 행은 대기열에 동시에 있을 수 있는 최대 스테이지 요청 횟수를 500으로 지정합니다.

```
maxactive=500
```

예제 stager.cmd 파일

코드 예 6-3는 `stager.cmd` 파일 예제를 표시합니다.

코드 예 6-3 예제 `stager.cmd` 파일

```
# This is stager.cmd file /etc/opt/SUNWsamfs/stager.cmd
drives=dog 1
bufsize=od 8 lock
logfile=/var/adm/stage.log
maxactive=500
```

스테이지에서 archiver.cmd 파일의 역할

archiver.cmd 파일의 명령 대부분은 아카이브에 영향을 주지만 아카이브 세트 할당 명령을 사용하면 아카이브 세트의 모든 파일에 적용되는 스테이지 속성을 지정할 수 있습니다. 아카이브 세트 할당 명령의 형식은 다음과 같습니다.

```
archive_set_name path [search_criteria ...] directives ... ]
```

일명 53 페이지의 "아카이브" 장은 아카이브 세트 할당 명령 및 해당 인수를 자세하게 설명합니다. 표 6-5는 아카이브 세트 할당 명령에 *directives*으로 나타날 수 있는 스테이지 명령을 표시합니다.

표 6-5 archiver.cmd 파일에 나타날 수 있는 스테이지 *directives*

명령	결과
-stage a	아카이브 세트의 파일이 연관 스테이지 되도록 지정합니다.
-stage n	아카이브 세트의 파일이 스테이지 되지 않도록 지정합니다.

이러한 명령과 기타 archiver.cmd 명령에 대한 자세한 내용은 53 페이지의 "아카이브"를 참조하십시오.

preview.cmd 파일을 사용한 미리보기 요청 우선 순위 지정

아카이버 및 스테이지 프로세스 모두 매체 로드 및 언로드를 요청할 수 있습니다. 요청 횟수가 매체 로드를 위해 사용 가능한 드라이브 수를 초과하면 초과 요청 횟수가 미리보기 대기열로 전송됩니다.

미리보기 대기열의 아카이브 및 스테이지 요청은 즉시 실행할 수 없습니다. 기본적으로 미리보기 요청은 FIFO(선입선출) 순서에 따라 실행됩니다.

미리보기 대기열에 있을 수 있는 항목 수는 defaults.conf 파일의 previews= 명령에 의해 결정됩니다. 이 명령의 값 변경에 대한 자세한 내용은 defaults.conf(4) 매뉴얼 페이지를 참조하십시오.

미리보기 요청에 대해 여러 가지 우선 순위를 지정할 수 있습니다. 다음과 같은 위치에 작성된 미리보기 명령 행에 명령을 입력하여 FIFO 기본 값을 변경할 수 있습니다.

```
/etc/opt/SUNWsamfs/preview.cmd
```

이 파일은 요청이 파일 스테이지를 위한 것이냐 아카이브를 위한 것이냐에 따라 미리보기 요청을 예약합니다. 또한 특정 VSN에 대한 우선 순위를 높일 수도 있습니다. 이밖에 `preview.cmd` 파일의 설정으로 최고 워터마크(HWM) 또는 최저 워터마크(LWM) 설정에 따라 전체 또는 특정 파일 시스템에 대한 미리보기 요청 우선 순위를 재지정할 수 있습니다.

`sam-amld` 데몬이 시작할 때 미리보기 명령을 읽습니다. 해당 하나의 명령을 지정해야 합니다. `sam-amld` 명령이 실행되는 동안 이 파일을 변경한 경우 `sam-amld` 데몬을 재시작하여 효력을 발생합니다. 주석 행은 우물정자(#)로 시작해 행의 끝까지 이어집니다. 이 파일에 대한 자세한 내용은 `preview.cmd(4)` 매뉴얼 페이지를 참조하십시오.

`preview.cmd` 파일에는 다음과 같은 2 가지 유형의 명령이 나타납니다.

- 모든 파일 시스템에 적용되는 전역 명령. 전역 명령은 처음 `fs =` 행 앞에 있어야 합니다.
- 특정 파일 시스템에 사용되는 명령. 이 명령은 전역 명령 뒤에 나타납니다. `archiver.cmd` 파일과 마찬가지로 `preview.cmd` 파일에도 개별 파일 시스템에 사용되는 명령을 포함할 수 있습니다. 개별 파일 시스템에 사용되는 명령은 파일에서 모든 전역 명령 이후에 나타나야 합니다.

파일 시스템 명령은 `fs = file_system_name` 명령으로 시작해야 합니다. 이 명령은 이후의 모든 명령이 속한 파일 시스템의 이름을 지정합니다. 파일 명령의 블록은 두 개 이상 파일에 나타날 수 있습니다. 파일 시스템 명령은 다음 `fs =` 행이 나타나거나 파일이 끝날 때까지 적용됩니다.

참고 - 여러 명령이 파일 시스템에 영향을 미치는 경우에는 특정 파일 시스템에 사용되는 명령이 전역 명령보다 우선합니다.

VSN 및 나이 명령(전역)

VSN 및 나이 우선 순위 명령은 전역 명령입니다. 이러한 명령이 `preview.cmd` 파일에 있는 경우에는 특정 파일 시스템에 사용되는 명령 이전에 나타나야 합니다. 즉, `fs =` 명령 이전에 나타나야 합니다. VSN 우선 순위 명령의 형식은 다음과 같습니다.

```
vsn_priority = value
```

이 명령은 정적 우선 순위 요소입니다. 이 명령은 우선 순위가 높은 VSN으로 플래그된 VSN에 대해 총 우선 순위의 증가 값을 나타냅니다. `vsn_priority`의 기본 값은 1000.0입니다. VSN은 이 값을 얻기 위해 미리보기 요청으로 예약될 때 우선 순위 플래그를 설정해야 합니다. `chmed(1M)` 명령을 사용해 `p` 옵션으로 우선 순위 플래그를 설정합니다 (예: `chmed +p lt.AAA123`). 이 플래그 설정은 이전에 미리보기 요청이 아닌, VSN에 대해 실행된 모든 요청에 적용됩니다. 나이 우선 순위 명령의 형식은 다음과 같습니다.

```
age_priority = factor
```

이 명령은 정적 우선 순위 요소입니다. 이 명령의 전체 결과는 동적입니다. `age_priority` 요소에 요청이 미리보기 요청인 초를 곱합니다. 이 결과에 요청의 전체 우선 순위를 더합니다. 요청의 실행 대기 시간이 길수록 나이 요소는 커집니다. 이 요소를 설정하면 기존의 요청은 우선 순위가 더 높은 다른 요소가 있는 새 요청으로 대체되지 않습니다.

이 요소가 1.0보다 크면 총 우선 순위를 계산할 때 시간 요소의 중요성이 증가합니다. 이 요소가 1.0보다 작으면 시간 요소의 중요성이 감소합니다. 이 요소를 0.0으로 설정하면 전체 우선 순위 계산에서 시간 요소가 제거됩니다.

우선 순위 플래그가 설정되지 않은 VSN은 대기열에 남아있는 시간에 따라 우선 순위가 증가합니다. 이 경우에는 우선 순위 플래그가 이미 설정되어 나중에 대기열에 들어오는 VSN보다 우선 순위가 더 높을 수 있습니다.

위터마크 명령(전역 또는 파일 시스템 전용)

위터마크 미리보기 요청 명령은 전역 명령 또는 파일 시스템 전용 명령으로 사용할 수 있습니다. 위터마크 우선 순위 명령은 미리보기 요청의 위터마크 우선 순위 (`wm_priority`)를 결정합니다. 코드 예 6-4는 `wm_priority` 요소가 여러 설정의 합계임을 표시합니다.

코드 예 6-4 `wm_priority` 계산

```
lwm_priority +  
lhwm_priority +  
hlwm_priority +  
hwm_priority  
-----  
= wm_priority
```

`wm_priority` 요소가 양수면 전체 우선 순위 계산 결과는 스테이지 요청보다 아카이브 요청의 우선 순위가 더 높아지게 됩니다. 그러나 `wm_priority` 요소는 음수가 될 수도 있습니다. 이 경우 아카이브 요청의 전체 우선 순위가 감소해 아카이브 요청보다 스테이지 요청의 우선 순위가 높아집니다. 설정이 0.0 즉, 지정된 명령 없으면 파일 시스템이 이러한 조건일 때 아카이브 요청에 대해 별다른 작업이 발생하지 않은 것입니다. 이에 대한 자세한 내용은 164 페이지의 "예제 1: 스테이지 요청 실행"의 예제를 참조하십시오.

표 6-6에서는 4 가지 워터마크 우선 순위 명령과 그 인수를 보여줍니다.

표 6-6 워터마크 우선 순위 명령

우선 순위 명령	인수
<code>lwm_priority = value</code>	<i>value</i> 에는 파일 시스템이 LWM 레벨 이하인 경우에 아카이브 요청에 대한 <code>wm_priority</code> 요소의 변화량을 지정합니다. 기본 값은 0.0입니다.
<code>lhwm_priority = value</code>	<i>value</i> 에는 파일 시스템이 LWM 이하에서 이상으로 증가했으나 HWM 레벨 이하인 경우에 아카이브 요청에 대한 <code>wm_priority</code> 요소의 변화량을 지정합니다. 이는 일반적으로 릴리서에서 채워지고 있다는 것을 나타냅니다. 기본 값은 0.0입니다.
<code>hlwm_priority = value</code>	<i>value</i> 에는 파일 시스템이 HWM 이상에서 이하로 증가했으나 LWM 레벨 이상인 경우에 아카이브 요청에 대한 <code>wm_priority</code> 요소의 변화량을 지정합니다. 이는 일반적으로 릴리서에서 파일 시스템이 LWM 이하가 되도록 충분한 디스크 공간을 남겨둘 수 없다는 것을 나타냅니다. 기본 값은 0.0입니다.
<code>hwm_priority = value</code>	<i>value</i> 에는 파일 시스템이 HWM 레벨 이상인 경우에 아카이브 요청에 대한 <code>wm_priority</code> 요소의 변화량을 지정합니다. 기본 값은 0.0입니다.

결국 4 가지 워터마크 설정은 파일 시스템이 얼마나 채워졌는지와 HWM 및 LWM이 설정된 레벨을 나타내는 퍼센트 값을 포함하는 동적 우선 순위 요소를 생성합니다. 미리 보기 요청에 할당된 값은 요소가 전역이나 파일 시스템 전용이나 아니면 설정되지 않았느냐에 따라 달라집니다.

파일 시스템의 조건이 바뀌면 해당 파일 시스템에 연결된 각 VSN의 우선 순위는 `chmed(1M)` 명령의 `p` 옵션의 사용 여부에 관계없이 적절한 워터마크 우선 순위에 따라 다시 계산됩니다.

워터마크 우선 순위는 아카이브에 대한 매체 요청을 계산할 때만 사용합니다. 스테이지에 대한 매체 요청을 계산할 때는 사용하지 않습니다.

다음 예제 명령은 파일 시스템이 HLWM에 있을 때 아카이브 요청에 대한 우선순위를 약간 높이는 방법을 표시합니다. 코드 예 6-5는 파일 시스템이 LWM 이하가 되도록 릴리서가 디스크 공간을 충분히 늘리는데 사용하는 설정을 표시합니다.

코드 예 6-5 LWM 이하로 가도록 설정

```
lhwm_priority = -200.0
hlwm_priority = 100.0
```

총 미리보기 요청 우선 순위 계산

미리보기 요청의 수적 우선 순위는 몇 가지 정적/동적 요소를 결합하여 결정할 수 있습니다. 숫자가 높을수록 우선 순위가 높습니다. 정적 우선 순위 요소는 요청이 발생할 때 설정됩니다. 정적 우선 순위 요소의 결과는 요청이 발생해 실행 대기 중인 상태가 끝나면 전체 우선 순위에 영향을 미치지 않습니다. 동적 우선 순위 요소는 요청이 실행 대기 중인 경우에 요청의 전체 우선 순위를 높이거나 낮춥니다.

미리보기 요청의 전체 우선 순위는 모든 우선 순위 요소의 합계입니다. 계산 방식은 다음과 같습니다.

```
total priority = vsn_priority + wm_priority + (age_priority *  
time_in_sec_as_preview_request)
```

미리보기 요청 우선 순위 방식 설정 방법

기본 미리보기 요청 FIFO 방식을 변경할 때는 타당한 시스템상의 이유가 있어야 합니다. 기본 미리보기 요청 FIFO 방식을 변경해야 하는 경우는 다음과 같습니다.

- 조건 1: 스테이지 요청이 아카이브 요청보다 먼저 처리해야 하는 경우
- 조건 2: 파일 시스템이 가득 차기 직전에 아카이브 요청에 가장 높은 우선 순위를 부여해야 하는 경우
- 조건 3: 특정 매체 그룹 사용 요청을 미리보기 요청 목록의 가장 높은 우선 순위로 높여야 하는 경우

사용자의 데이터 액세스가 가장 중요하거나 VSN 드라이브가 제한되거나 파일 아카이브가 배경 기능으로 수행되는 환경에서는 `preview.cmd` 파일을 사용해 저장 시스템 리소스가 스테이지 요청에 기여하는 방식을 조정할 수 있습니다. `preview.cmd` 파일의 설정을 사용자 정의하여 위 시나리오 모두를 지원하고 구성된 Sun StorEdge SAM-FS 환경에 영향을 줄 수 있습니다.

데이터는 이 파일의 설정에 의해 영향을 받지 않기 때문에 명령 설정을 시험 및 조정하여 각 미리보기 요청의 우선 순위에 대해 가중치를 부여할 때 아카이브와 스테이지 요청간에 적절한 균형을 잡는 것이 바람직합니다.

코드 예 6-6은 앞의 세 조건을 지정하는 `preview.cmd` 파일 예제를 표시합니다.

코드 예 6-6 `preview.cmd` 파일 예제

```
# condition 1  
lwm_priority = -200.0  
lhwm_priority = -200.0
```

코드 예 6-6 preview.cmd 파일 예제 (계속)

```
hlwm_priority = -200.0
# condition 2
hwm_priority = 500.0
# condition 3
age_priority = 1.0
```

예제 1: 스테이지 요청 실행

다음 예제 설정에서는 스테이지 요청의 우선 순위를 아카이브 요청보다 높이는 한 가지 방법을 보여줍니다. 이 예제에서는 다음과 같은 사항을 가정합니다.

- 몇몇 요청이 100초 동안 대기열에 있습니다.
- 기본 vsn_priority는 1000입니다.

표 6-7에서는 총 요청 우선 순위 계산 방법을 보여줍니다.

표 6-7 요청 우선 순위 예제

우선 순위	계산
우선 순위가 LWM인 아카이브 VSN:	$1000 + (-200) + (1 \times 100) = 900$
우선 순위가 LWM인 스테이지 VSN:	$1000 + 0 + (1 \times 100) = 1100$
우선 순위 LWM이 없는 스테이지 VSN:	$0 + 0 + (1 \times 100) = 100$

이 예제에서는 다른 요소가 같은 경우 wm_priority에 들어가는 음수로 인해 아카이브 요청보다 스테이지 요청의 우선 순위가 높게 됩니다.

예제 2: 아카이브 요청 실행

파일을 다시 사용자에게 스테이지하는 것과 매체에 아카이브된 새 파일을 가져오는 것 간에 균형이 잡힌 환경에서는 HWM의 초과 여부가 가장 중요합니다. 이러한 경우에 파일 시스템이 차있는 비율을 낮출 수 있을 만큼 아카이브 요구 사항에 부합하는 파일이 충분하지 않으면 보류 중인 아카이브 요청을 완료하는 것이 파일 시스템이 가득 차지 않도록 하는 차선책입니다.

이러한 경우에 preview.cmd 파일은 다음과 같이 단순할 수 있습니다.

```
hwm_priority = 500.0
```

예제 3: 매체별 우선 순위 지정

프로젝트 중심 환경에서는 특정 VSN을 사용하고 다른 사용자와 격리된 특정 사용자가 여러 파일 그룹 관련 작업을 하는 경우가 있습니다. 이러한 환경에서는 특정 프로젝트가 특정 기간에 높은 우선 순위를 갖습니다. 즉, 사용 가능한 시스템 저장 리소스에서 높은 우선 순위가 필요한 경우가 있습니다. 다음 명령을 사용해 `preview.cmd` 파일을 구성해 사용자와 매체에 알맞은 매체 드라이브 우선 순위를 지정할 수 있습니다.

```
hwm_priority = 5000.0
```

우선 순위 사용자 그룹의 모든 VSN에는 다음과 같은 정보를 입력합니다.

```
# chmed +p lt.AAA123 ## or whatever VSN is used
```

그 후에 VSN AAA123(또는 임의의 VSN 사용)이 필요한 모든 요청을 미리보기 대기열의 기타 보류 중인 마운트 요청보다 우선 순위로 둡니다.

나중에 사용자 매체의 우선 순위를 해제하려면 모든 VSN에 대해 다음과 같은 리버스 명령을 입력합니다.

```
# chmed -p lt.AAA123 ## or whatever media type is used
```

예제 4: 복합 우선 순위 지정

요구 사항이 다음과 같은 Sun StorEdge SAM-FS 파일 시스템이 2개 있다고 가정합니다.

- 요청이 대기열에 있는 시간이 너무 길어서는 안됩니다(`age_priority`).
- 파일 시스템이 LWM 이하이면 스테이지 요청에 우선 순위를 부여해야 합니다.
- 파일 시스템이 LWM 이상이나 HWM 이하인 경우에는 아카이브 또는 스테이지 요청에 우선 순위를 지정하지 않아도 됩니다. 코드 예 6-7에 영향 받은 명령이 나옵니다.

코드 예 6-7 명령

```
lwm_priority = -200.0  
lhwm_priority = 0.0  
hlwm_priority = 0.0
```

이 경우 다른 명령은 영향을 받지 않습니다.

파일 시스템이 HWM 이상이면 아카이브 요청에 우선 순위를 부여해야 합니다.

두 가지 파일 시스템이 모두 HWM 이상이면 두 번째 파일 시스템(예: samfs2)이 채워지지 않도록 하는 것이 우선입니다. 예를 들면, samfs1이 사용자 작업 파일 시스템이고 samfs2가 중요 시스템 파일 시스템인 경우를 말합니다.

chmed(1M) 명령의 p 플래그가 설정된 경우에는 무조건 미리보기 요청 대기열에서 VSN의 선택 그룹에 대한 요청에 우선 순위를 부여해야 합니다.

코드 예 6-8은 위 목록의 요구 사항에 따라 요청에 우선 순위를 지정하는 preview.cmd 파일을 표시합니다.

코드 예 6-8 preview.cmd 파일

```
age_priority = 100.0
vsn_priority = 20000.0
lhwm_priority = -200.0
hlwm_priority = -200.0
fs = samfs1
hwm_priority = 1000.0
fs = samfs2
hwm_priority = 5000.0
```

리사이클

*리사이클*이란 아카이브 볼륨의 공간을 회수하는 프로세스를 말합니다. 리사이클러는 아카이버와 함께 작동하여 사용하지 않는 아카이브 복사본이 차지하는 공간을 회수합니다. 파일을 수정하면 기존 버전에 연결된 아카이브 복사본을 시스템에서 제거할 수 있습니다. 리사이클러는 완료된 아카이브 복사본의 가장 큰 부분을 아카이브 볼륨으로 식별하고 완료되지 않은 사본을 개별 볼륨으로 이동할 것을 지시합니다. 완료된 사본만 제공된 볼륨에 존재할 경우 사이트가 정의한 작업이 수행됩니다. 예를 들어 이러한 볼륨은 즉시 재사용을 위해 레이블을 변경하거나 외부 저장소로 내보낼 수 있기 때문에 파일 변경 내역 기록을 별도로 보유할 수 있습니다. 리사이클 프로세스는 데이터 파일과 관련이 있기 때문에 사용자가 알 수 없습니다.

이 장에서는 다음 항목에 대해 설명합니다.

- 167 페이지의 "리사이클러:개요"
- 169 페이지의 "리사이클 명령"
- 172 페이지의 "리사이클러 구성"
- 180 페이지의 "리사이클러 문제 해결"

리사이클러:개요

리사이클러는 완료된 아카이브 복사본이 사용하는 공간을 사이트 지정 매개 변수가 지정한 최소치로 유지합니다. 특정 아카이브 볼륨의 공간은 항상 다음과 같은 요소로 구성됩니다.

- *현재 데이터*는 현재 활성화된 아카이브 이미지가 사용하는 공간입니다.
- *완료된 데이터*는 현재 더 이상 활성화되지 않은 아카이브 이미지가 사용하는 공간입니다.
- *여유 공간*은 현재 활성화되거나 완료된 아카이브 이미지가 사용하지 않는 공간입니다.

볼륨의 용량은 볼륨의 데이터를 위한 전체 공간입니다. 예를 들어 3GB를 쓴 10GB 테이프 볼륨은 용량이 10GB이고 여유 공간이 7GB입니다.

새 아카이브 매체 또는 레이블이 새로 지정된 아카이브 매체는 모든 용량이 여유 공간으로 시작합니다. 데이터가 매체에 아카이브됨에 따라 여유 공간이 줄고 현재 데이터가 늘어나게 됩니다.

파일 시스템에서 아카이브된 파일이 변경되거나 제거되면 아카이브 이미지가 만료되고 아카이브된 파일은 현재 데이터 분류에서 만료 데이터 분류로 이동합니다. 이러한 이미지가 사용하는 물리적 공간은 동일하며 단지 해당 공간을 가리키는 파일 시스템에 더 이상 파일이 없게됩니다.

이러한 만료 이미지, 즉 만료 데이터는 결국 모든 여유 공간을 사용합니다. 공간을 리사이클해야 이러한 이미지를 제거할 수 있고, 이미지가 차지하는 공간은 여유 공간이 됩니다. 리사이클러의 목적은 현재 데이터를 잃지 않고 만료된 데이터가 사용하는 공간을 여유 공간으로 변환하는 것입니다.

예를 들어 테이프와 같은 제거 가능한 매체 카트리지는 추가만 가능합니다. 제자리에 다시 쓸 수는 없습니다. 카트리지를 다시 사용할 수 있는 유일한 방법은 카트리지에서 현재 데이터를 모두 제거하고 카트리지에 레이블을 다시 지정한 다음, 처음부터 다시 사용하기 시작하는 것입니다.

`sam-recycler(1M)` 명령을 입력해 리사이클을 초기화해야 합니다. 리사이클 초기화는 수동으로 하거나 `cron(1)` 작업을 통해 할 수 있습니다. 표 7-1에 리사이클 방법이 나옵니다.

표 7-1 리사이클 방법 및 매체 테이프

리사이클 방법	매체 및 기록
자동화 라이브러리	제거 가능한 매체 카트리지. 라이브러리를 사용하여 아카이브할 때 <code>recycler.cmd</code> 파일에 리사이클 명령을 둡니다.
아카이브 세트	제거 가능한 매체 카트리지 및 디스크. 아카이브 세트 아카이브할 때 <code>recycler.cmd</code> 파일을 사용하지 마십시오. 모든 리사이클 명령을 <code>archiver.cmd</code> 파일에 둡니다.

라이브러리 또는 아카이브 세트 중 하나를 사용하여 리사이클할 수 있습니다. 디스크에 아카이브하려는 경우 표 7-1에 표시된 대로 아카이브 세트를 사용해서만 리사이클할 수 있습니다.

리사이클러 및 아카이버는 다음과 같이 함께 작동합니다.

1. 리사이클러는 `rearchive` 속성을 사용하여 볼륨에 있는 현재의 모든 (유효한) 아카이브 이미지를 표시합니다.
2. 제거 가능한 매체에 아카이브하려는 경우 리사이클러는 `recycle` 속성을 사용하여 선택된 아카이브 볼륨을 표시합니다. 이렇게 하면 아카이버가 더 이상의 아카이브 이미지를 볼륨에 기록하지 못합니다.

3. 아카이버는 모든 표시된 이미지를 또 다른 볼륨으로 이동시킵니다. 이러한 작업을 *재아카이브*라고 합니다. 아카이버가 기존 볼륨의 현재 아카이브 이미지를 새 볼륨으로 이동시킨 후 기존 볼륨에는 여유 공간 및 만료된 공간만 있게 됩니다. 제거 가능한 매체 카트리지에 아카이브하려는 경우 카트리지를 레이블 변경 후 재사용할 수 있습니다. 디스크에 아카이브하려는 경우 리사이클러는 만료된 아카이브 이미지가 있는 파일을 제거합니다.

리사이클러는 주기적으로 실행되도록 설계되었습니다. 리사이클러는 한번 호출될 때 마다 가능한 많은 작업을 수행합니다. 리사이클러는 아카이버가 파일을 재아카이브하기 전에 복사본에 재아카이브 표시를 마쳐야 합니다.

가끔 `rearchive` 속성 세트가 있는 만료된 아카이브 이미지는 매체에 남습니다. 이러한 현상은 다음과 같은 상황에서 발생할 수 있습니다.

- 아카이버는 리사이클러가 만료된 아카이브 이미지를 표시한 후에는 실행되지 않습니다.
- 매체는 만료되지 않은 아카이브 이미지를 이동시킬 때 사용하는 아카이버를 사용할 수 없습니다.
- 기타 다른 아카이버 변칙.

리사이클러는 각 실행 사이에 라이브러리 카탈로그 및 `inode`에 상태 정보를 보유합니다. 리사이클 프로세스 중 `s1s(1)` 명령 및 해당 `-D` 옵션을 사용하여 파일에 대한 정보를 표시할 수 있습니다. `s1s(1)` 명령의 출력은 파일이 재아카이브에 대해 예약이 되어 있는지의 여부를 표시합니다.

리사이클 명령

`recycler.cmd` 파일은 다음 섹션에서 설명한 명령을 허용합니다.

- 169 페이지의 "로그 파일 지정 `logfile` 명령"
- 170 페이지의 "리사이클 방지 `no_recycle` 명령"
- 170 페이지의 "전체 자동화 라이브러리에 리사이클 지정 라이브러리 명령"

로그 파일 지정 `logfile` 명령

`logfile` 명령은 리사이클러 로그 파일을 지정합니다. 이 명령의 형식은 다음과 같습니다.

```
logfile = filename
```

*filename*에 대해 로그 파일에 대한 경로를 지정합니다.

다음은 logfile= 명령 행의 예제입니다.

```
logfile = /var/adm/recycler.log
```

리사이클 방지 no_recycle 명령

no_recycle 명령을 사용하면 볼륨의 리사이클을 방지할 수 있습니다. VSN을 지정하려면 일반 표현식과 하나 이상의 특정 매체 유형을 사용합니다. 이 명령의 형식은 다음과 같습니다.

```
no_recycle media_type VSN_regex [ VSN_regex ... ]
```

표 7-2 no_recycle 명령에 대한 인수

인수	의미
<i>media_type</i>	mcf(4) 매뉴얼 페이지의 매체 유형을 지정합니다.
<i>VSN_regex</i>	공백으로 구분된 하나 이상의 일반 표현식을 지정해 볼륨을 나타냅니다. regex의 형식에 대한 자세한 내용은 regexp(5) 매뉴얼 페이지 또는 83 페이지의 "패턴 일치"를 사용한 파일 이름 search_criteria: -name regex"를 참조하십시오.

*media_type*을 지정하면 특정 유형의 매체에 저장된 볼륨의 리사이클을 방지할 수 있습니다. *VSN_regex*를 하나 이상 지정하면 일반 표현식을 사용해 리사이클에서 제외될 특정 카트리지를 식별할 수 있습니다.

예를 들어 다음과 같은 명령 행은 VSN 식별자가 DLT로 시작하는 테이프 볼륨을 리사이클에서 제외합니다.

```
no_recycle lt DLT.*
```

전체 자동화 라이브러리에 리사이클 지정 라이브러리 명령

라이브러리 명령을 사용하면 특정 라이브러리에 연결된 VSN에 대해 다양한 리사이클 매개 변수를 지정할 수 있습니다. 이 명령의 형식은 다음과 같습니다.

```
library parameter [ parameter ... ]
```

*library*는 mcf(4) 파일의 Family Set 필드에 지정된 라이브러리의 이름을 지정합니다. *parameter*는 표 7-3에서 공백으로 구분된 *parameter* 키워드를 하나 이상 지정합니다.

표 7-3 라이브러리 명령 *parameter* 값

<i>parameter</i>	작업
-dataquantity <i>size</i>	리사이클러가 유용한 데이터의 볼륨을 비우기 위해 재아카이브 예약할 수 있는 데이터의 양을 제한합니다. 기본 값은 1GB입니다.
-hwm <i>percent</i>	라이브러리 최고 워터마크. 기본 값은 95입니다.
-ignore	이 라이브러리의 볼륨이 리사이클되는 것을 방지합니다. 이 명령은 <i>recycler.cmd</i> 파일을 테스트할 때 유용합니다.
-mail [<i>email_address</i>]	지정된 <i>email_address</i> 로 전자우편을 보냅니다. 기본적으로 전자우편은 전송되지 않습니다. -mail에 인수가 지정되어 있지 않으면 전자우편을 root로 보냅니다.
-mingain <i>value</i>	최소 VSN 증가. 기본 값은 50입니다.
-vsncount <i>count</i>	리사이클할 볼륨의 수를 카운트로 제한합니다. 기본 값은 1입니다.

예를 들면 다음과 같은 명령 행을 고려할 수 있습니다.

```
gr47 -hwm 85 -ignore -mail root -mingain 40
```

이 명령행은 라이브러리 *gr47*에 대해 다음과 같은 사항을 지정합니다.

- 라이브러리는 볼륨이 85 퍼센트 차면 리사이클 대상이 됩니다.
- 최소 퍼센트 증가는 40 퍼센트입니다.
- 1GB 이상 재아카이브되지 않습니다. 이는 기본 값이므로 *recycler.cmd* 파일에 지정되지 않습니다.
- 하나의 볼륨만 리사이클됩니다. 이 역시 기본 설정입니다.
- 리사이클 전자우편 메시지를 root로 보냅니다.

리사이클러 구성

리사이클러를 구성하기 전 다음 사항을 참고합니다.

- `archiver.cmd` 파일의 명령은 아카이브 세트별로 리사이클을 제어합니다. `recycler.cmd` 파일의 명령은 라이브러리별로 리사이클을 제어합니다. 이밖에 `recycler.cmd` 파일은 일반 리사이클러 작동을 제어합니다. 리사이클러 명령에 대한 자세한 내용은 169 페이지의 "리사이클 명령"을 참조하십시오.
- 제거 가능한 매체 파일이 있는 볼륨을 리사이클하지 마십시오. `request(1)` 명령을 사용하여 제거 가능한 매체 파일을 작성합니다. 리사이클러는 `request(1)` 명령으로 만든 제거 가능한 매체 파일을 보유하지 않습니다. 제거 가능한 매체 파일이 있는 볼륨은 유출할 수 없습니다.
- Sun StorEdge SAM-FS 파일 시스템에서 유지 관리를 수행하는 동안 리사이클러를 실행하지 마십시오. 리사이클러는 `.inodes` 파일과 `mcf` 파일을 사용해 현재 또는 만료된 파일을 식별하고 파일 시스템에 연결된 장치를 식별합니다. 이러한 파일에 올바른 정보가 없으면 현재 아카이브된 데이터가 만료된 것으로 표시되어 리사이클될 수 있습니다.
- 모든 Sun StorEdge SAM-FS 파일 시스템은 리사이클러가 실행중일 때 마운트 되어야 합니다. 온라인 디스크에서 리사이클 하려면 디스크 볼륨이 포함된 파일 시스템을 마운트하고 호스트 시스템에 액세스할 수 있어야 합니다.

리사이클러는 기본적으로 활성화되어 있지 않습니다. `sam-recycler(1M)` 명령을 입력해 리사이클을 초기화해야 합니다. 리사이클러가 초기화되면 170 페이지의 "전체 자동화 라이브러리에 리사이클 지정 라이브러리 명령"에 지정된 기본 리사이클러 설정 사항이 적용됩니다. 리사이클러에 대한 자세한 내용은 `sam-recycler(1M)` 매뉴얼 페이지를 참조하십시오.

다음 섹션에서는 리사이클러를 구성하는 프로세스에 대해 설명합니다. 이 프로세스에는 다음 단계들이 포함됩니다.

- 173 페이지의 "1 단계: `recycler.cmd` 파일 작성(선택적)"
- 175 페이지의 "2 단계: `archiver.cmd` 파일 편집 (선택적)"
- 177 페이지의 "3 단계: 리사이클러 실행"
- 178 페이지의 "4 단계: 리사이클러용 `crontab` 파일 만들기(선택 사항)"
- 179 페이지의 "5 단계: `-recycle_ignore` 및 `ignore` 매개변수 제거"
- 179 페이지의 "6 단계: `recycler.sh` 파일 작성(선택적)"

라이브러리의 카트리지에 아카이브하는 경우, 이 프로세스는 `recycler.cmd` 파일을 만들고, 경우에 따라서는 `archiver.cmd` 파일을 편집하는 것으로 구성됩니다. 디스크에 아카이브하는 경우에는 아카이브 세트별로만 아카이브할 수 있기 때문에 이 디스크 볼륨의 리사이클을 활성화하려면 `archiver.cmd` 파일을 편집합니다. 다음 절차에서는 아카이브 매체에 대해 리사이클러를 구성하는 방법에 대해 설명합니다.

▼ 1 단계: `recycler.cmd` 파일 작성(선택적)

라이브러리에서 카트리지의 아카이브 복사본을 리사이클하는 경우에 이 단계를 수행합니다.

디스크 볼륨의 아카이브 복사본을 리사이클하는 경우에는 리사이클이 `archiver.cmd` 파일의 명령에 의해 제어되기 때문에 이 단계를 완료할 수 없습니다. `archiver.cmd` 파일의 리사이클 구성에 대한 자세한 내용은 175 페이지의 "2 단계: `archiver.cmd` 파일 편집 (선택적)"을 참조하십시오.

`recycler.cmd` 파일에는 일반 리사이클 명령이 포함되며 Sun StorEdge SAM-FS 환경의 각 라이브러리에 대한 명령이 포함될 수도 있습니다. 리사이클 명령에 대한 자세한 내용은 169 페이지의 "리사이클 명령"을 참조하십시오.

아카이브 세트별로 리사이클하는 경우에도 `recycler.cmd` 파일에 각 라이브러리를 구성해야 합니다. 이렇게 해야 필요한 경우에 아카이브 세트에 포함되지 않는 VSN을 리사이클할 수 있습니다.

일반적인 `recycler.cmd` 파일에는 다음과 같은 명령 행이 포함됩니다.

- 리사이클러 로그 파일을 지정하는 `logfile=` 명령 행. 시스템은 이 파일에 리사이클 메시지와 리사이클 보고를 작성합니다.
- 리사이클할 볼륨이 포함된 각 라이브러리에 대한 하나 이상의 명령 행. 이 명령 행에는 리사이클할 라이브러리에 대한 `mcf` 파일에서 가져온 패밀리 세트 이름이 포함됩니다. 이는 리사이클러에 대한 라이브러리를 식별합니다.

아직 `recycler.cmd` 행을 만들고 있고 테스트를 하지 않았기 때문에 `ignore` 키워드를 사용해야 합니다. 이 프로세스의 다음 단계에서 `ignore` 키워드를 제거합니다.

`recycler.cmd` 파일을 작성하려면 다음 단계를 수행합니다.

1. 슈퍼유저가 됩니다.
2. `vi(1)` 또는 또 다른 편집기를 사용하여 `/etc/opt/SUNWsamfs/recycler.cmd` 파일을 엽니다.
3. 이 장에 설명된 하나 이상의 명령을 추가하여 리사이클러 작업을 제어합니다.
4. 파일을 저장하고 닫습니다.

예제 `recycler.cmd` 파일

코드 예 7-1에서는 `recycler.cmd` 파일의 예제를 보여줍니다.

코드 예 7-1 A `recycler.cmd` `recycler.cmd` 파일 예제

```
logfile = /usr/tmp/recycler.log
stk30 -hwm 51 -mingain 60 -ignore -mail root
```

다음 섹션에서는 코드 예 7-1에 지정된 매개 변수에 대해 설명합니다.

-hwm 51 매개 변수

최고 워터마크를 지정하면 매체 사용 퍼센트를 설정해 설정 값 이하에서는 리사이클이 일어나지 않도록 할 수 있습니다. 이 퍼센트는 라이브러리에서 사용되는 공간과 총 용량의 비율입니다. 예를 들면 3개는 100 퍼센트 찼고 나머지 7개는 각각 30 퍼센트씩 차 있는 20GB의 테이프를 총 10개 보유한 라이브러리의 매체 사용률은 다음과 같습니다.

$$((3 * 1.00 + 7 * 0.30) * 20G) / (10 * 20G) * 100\% = 51\%$$

이 계산에서는 현재 데이터와 만료 데이터를 구분하지 않았습니다. 다만 사용되는 매체의 양만을 다루었습니다.

이 예에서 최고 워터마크가 51 퍼센트 이하라면 리사이클러는 리사이클할 자동화 라이브러리의 VSN을 자동으로 선택하지 않습니다.

참고 - 리사이클 플래그를 설정하기 위해 다음 명령을 사용하여 VSN을 강제 리사이클합니다.

```
# chmed +c lt.AAA123
```

+c 플래그가 설정되면 아카이버는 더 이상의 아카이브 이미지를 볼륨에 기록하지 않습니다. +c 플래그는 samu(1M) 유틸리티를 통해 볼 수 있습니다. 자세한 내용은 chmed(1M) 및 samu(1M) 매뉴얼 페이지를 참조하십시오.

-mingain 60 매개 변수

최소 VSN 증가 퍼센트는 카트리지를 리사이클하여 얻을 수 있는 공간의 최저 한계를 설정합니다. 예를 들어 자동화 라이브러리의 카트리지 95 퍼센트는 현재 데이터이고 5 퍼센트는 만료된 데이터인 경우, 카트리지 리사이클을 통해 얻은 증가는 5 퍼센트뿐입니다. 이 공간을 회수하기 위해 다른 95 퍼센트를 이동할 만한 가치가 없을 수도 있습니다. 최소 증가를 6 퍼센트 이상을 설정하면 리사이클러가 이 예제의 VSN을 자동으로 선택하지 못합니다.

이외에도 90 퍼센트가 만료된 데이터, 5 퍼센트가 현재 데이터, 5 퍼센트가 여유 공간인 카트리지를 예로 들 수 있습니다. 이 경우에는 리사이클을 통해 90 퍼센트가 증가하게 됩니다.

-ignore 매개 변수

-ignore 키워드를 사용하면 리사이클러가 특정 라이브러리를 리사이클하지 못하기 때문에 리사이클러를 구성할 때 사용해야 합니다.

-mail root 매개 변수

-mail 매개 변수는 특정 라이브러리에 대해 리사이클이 일어날 때 리사이클러에서 메일을 보내도록 지정합니다. 메일 메시지의 제목 행은 다음과 같습니다.

```
Robot robot-name recycle
```

표 7-2는 샘플 메시지 본문을 표시합니다.

코드 예 7-2 리사이클 메시지 샘플

```
I will recycle VSN vsn.
Cannot find any candidate VSN in this media changer.
Previously selected VSN vsn is not yet finished recycling.
Previously selected VSN vsn is now finished recycling. It will now
be post-recycled.
```

▼ 2 단계: archiver.cmd 파일 편집 (선택적)

이 단계는 아카이브 세트별로 리사이클하는 경우에 수행합니다. 디스크에 아카이브하는 경우에는 아카이브 세트별로 리사이클하는 것이 유일한 리사이클 방법이기 때문에, 디스크에 아카이브 하려면 리사이클을 위해서는 이 단계를 완료해야 합니다.

라이브러리별로 리사이클하려면 다음 단계를 진행할 수 있습니다.

- archiver.cmd 파일을 편집하려면 일명 66 페이지의 "archiver.cmd 파일 작성 또는 수정하기 및 변경내용 전파하기" 절차를 수행합니다.

아카이브 세트로 리사이클을 활성화하도록 archiver.cmd 파일에 추가하는 명령은 params 및 endparams 명령 사이에 나타나야 합니다. 표 7-4는 사용할 수 있는 아카이브 세트 리사이클 명령을 표시합니다.

표 7-4 아카이브 세트 리사이클 명령

명령	기능
-recycle_dataquantity <i>size</i>	리사이클러가 유용한 데이터의 볼륨을 비우기 위해 재아카이브 예약할 수 있는 데이터의 양을 제한합니다.
-recycle_hwm <i>percent</i>	최고 워터마크 퍼센트를 설정합니다.
-recycle_ignore	아카이브 설정이 리사이클되는 것을 방지합니다.
-recycle_mailaddr <i>mail_address</i>	리사이클러 메시지를 <i>mail_address</i> 에 보냅니다.
-recycle_mingain <i>percent</i>	여유 공간을 <i>percent</i> 이상 증가시키는 VSN으로 리사이클을 제한합니다.
-recycle_vsncount <i>count</i>	재아카이브할 볼륨의 수를 <i>count</i> 로 제한합니다.

위의 명령에 대한 자세한 내용은 53 페이지의 "아카이브"를 참조하거나 archiver.cmd(4) 매뉴얼 페이지를 참조하십시오.

코드 예 7-3에서는 디스크 아카이브 리사이클에 대한 archiver.cmd 예제를 보여줍니다.

코드 예 7-3 archiver.cmd 파일의 디스크 아카이브 지정

```

fs = samfs1
  1 2m

arset0 testdir0
  1 2m
  2 4m

arset1 testdir1
  1 2m
  2 4m

Params
arset0.1 -disk_archive disk01 -recycle_hwm 80 \
-recycle_mingain 20 -recycle_ignore
arset1.1 -disk_archive disk02 -recycle_hwm 80 \
-recycle_mingain 20 -recycle_ignore
endparams

```


코드 예 7-4

제거 가능한 매체 카트리지에 대한 리사이클러 로그 파일 예제 (계속)

```

----Status-----      Count      Bytes      Use Obsolete Free      Library:Type:VSN
no-data VSN            0           0           0    87      13      m160:at:000003
no-data VSN            0           0           0    33      67      m160:at:000004
no-data VSN            0           0           0    27      73      m160:at:000005
no-data VSN            0           0           0     8      92      m160:at:000044
no-data VSN            0           0           0     7      93      m160:at:000002
no-data VSN            0           0           0     5      95      m160:at:000033
empty VSN              0           0           0     0     100     m160:at:CLN005
empty VSN              0           0           0     0     100     m160:at:000001

Recycler finished.

===== Recycler ends at Wed Dec 12 14:05:32 2001 =====

```

코드 예 7-5에서는 디스크 아카이브 파일의 리사이클에 대한 예제 리사이클러 로그 파일을 보여줍니다.

코드 예 7-5

디스크 아카이브 파일에 대한 리사이클러 로그 파일 예제

```

---Archives---      -----Percent-----
----Status-----      Count      Bytes      Use Obsolete Free      Library:Type:VSN
new candidate        0           0           0    41      59      <none>:dk:disk01

677 files recycled from VSN disk01 (mars:/sam4/copy1)
0 directories recycled from VSN disk01 (mars:/sam4/copy1)

```

▼ 4 단계: 리사이클러용 crontab 파일 만들기(선택 사항)

시스템이 예상대로 실행되는 경우에는 슈퍼유저가 리사이클러를 주기적으로 실행할 수 있도록 crontab 항목을 만들 수 있습니다. 사이트의 상황에 따라 두 시간마다 한 번 이상 리사이클러를 실행할 수 있습니다.

● crontab 항목을 작성합니다.

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

다음과 같은 루트의 crontab 파일 예제 항목은 다음과 같이 cron 데몬이 홀수 시간의 5분마다 한 시간동안 리사이클러를 실행하도록 합니다.

```
5 1,3,5,7,9,11,13,15,17,19,21,23 * * * /opt/SUNWsamfs/sbin/sam-recycler
```

▼ 5 단계: `-recycle_ignore` 및 `ignore` 매개변수 제거

1. `vi(1)` 또는 또 다른 편집기를 사용하여 `archiver.cmd` 파일에서 `-recycle_ignore` 매개변수를 제거합니다.
2. `vi(1)` 또는 또 다른 편집기를 사용하여 `recycler.cmd` 파일에서 `ignore` 매개변수를 제거합니다.
이제 리사이클을 하게됩니다.

▼ 6 단계: `recycler.sh` 파일 작성(선택적)

제거 가능한 카트리지에서 아카이브 복사본을 리사이클하는 경우에 이 단계를 수행합니다. 디스크에만 아카이브하는 경우에는 이 단계를 수행하지 않습니다.

리사이클러는 VSN의 현재 이미지가 모두 다른 VSN에 재아카이브되면 `recycler.sh` 스크립트를 실행합니다. 예를 들어 `recycler.sh(1M)` 매뉴얼 페이지를 참조하십시오. `/opt/SUNWsamfs/examples/recycler.sh` 예제에서는 리사이클된 VSN에 레이블을 다시 지정하는 방법과 슈퍼유저에 메일을 보내는 방법을 보여줍니다.

리사이클러에서 인수가 다음과 같은 `/opt/SUNWsamfs/sbin/recycler.sh` 스크립트를 호출하였습니다.

```
Media type: $1  VSN: $2  Slot: $3  Eq: $4
```

알려진 활성 아카이브 복사본이 VSN에서 모두 유출된 것으로 리사이클러가 판단하면 `/opt/SUNWsamfs/sbin/recycler.sh` 스크립트가 호출됩니다. 리사이클된 카트리지가 필요없게 되는 사이트 요구 사항을 파악해야 합니다. 일부 사이트에서는 카트리지를 레이블 변경 및 재사용하고, 일부 사이트에서는 자동화 라이브러리에서 카트리지를 제거하여 나중에 내역 파일에 액세스하는데 사용합니다. 자세한 내용은 `recycler(1M)` 및 `recycler.sh(1M)` 매뉴얼 페이지를 참조하십시오.

리사이클러 문제 해결

리사이클러에 발생하는 가장 빈번한 문제점은 리사이클러가 실행될 때 다음과 유사한 메시지를 생성할 때 나타납니다.

```
Waiting for VSN mo:OPT000 to drain, it still has 123 active archive copies.
```

다음 조건 중 하나는 리사이클러가 이 메시지를 생성하도록 할 수 있습니다.

- 조건 1: 아카이버가 볼륨에서 123개의 아카이브 복사본을 재아카이브하지 않습니다.
- 조건 2: 123개의 아카이브 복사본이 파일 시스템의 파일에 적용되지 않습니다. 이러한 복사본은 123개의 메타 데이터 아카이브 복사본에 적용됩니다.

상황 1이 발생할 수 있는 원인은 다음과 같습니다.

- 재아카이브해야 하는 파일은 no_archive로 표시됩니다.
- 재아카이브해야 하는 파일은 no_archive 아카이브 세트에 있습니다.
- 파일은 사용 가능한 VSN이 없기 때문에 아카이브할 수 없습니다.
- archiver.cmd 파일에는 wait 명령이 포함됩니다.

어떠한 상황이 적용되는지 파악하려면 -v 옵션을 사용해 리사이클러를 실행합니다. 코드 예 7-6에 표시된 대로 이 옵션은 리사이클러 로그 파일의 아카이브 복사본 123개와 관련된 파일의 경로 이름을 표시합니다.

코드 예 7-6 리사이클러 메시지

```
Archive copy 2 of /sam/fast/testA resides on VSN LSDAT1
Archive copy 1 of /sam3/tmp/dir2/filex resides on VSN LSDAT1
Archive copy 1 of Cannot find pathname for file system /sam3
inum/gen 30/1 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gunk/tstfila00 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gunk/tstfilF82 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gunk/tstfilV03 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gink/tstfila06 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gink/tstfila33 resides on VSN LSDAT1
Waiting for VSN dt:LSDAT1 to drain, it still has 8 active archive
copies.
```

이 예제 결과에는 7개의 경로 이름이 포함된 메시지가 `Cannot find pathname...` 텍스트를 포함하는 하나의 메시지와 함께 표시됩니다. `LSDAT1`이 유출되지 않는 문제를 해결하려면 7개의 파일을 재아카이브할 수 없는 원인을 파악해야 합니다. 7개의 파일이 재아카이브되면 오직 하나의 아카이브 복사본이 파일에 연결되지 않습니다. 이러한 상황은 `.inodes` 파일을 부분적으로 손상시킨 시스템 충돌의 결과로만 발생합니다.

경로 이름을 찾는 문제를 해결하려면 `samfsck(1M)`를 실행해 원본에서 분리된 `inode`를 회수합니다. `samfsck(1M)`를 실행하지 않거나 파일 시스템의 마운트를 해제하여 `samfsck(1M)`를 실행할 수 없으면, `recycler -v` 수행 결과에 유효한 아카이브 복사본이 있는지 확인한 후에 수동으로 카트리지에 레이블을 다시 지정할 수 있습니다. 그러나 `.inodes` 파일에 남아있는 잘못된 `inode`가 리사이클러에서 계속 발생하기 때문에 다음에 `VSN`이 리사이클 캔디데이트가 될 때 동일한 문제가 다시 발생할 수도 있습니다.

리사이클러가 리사이클할 `VSN`을 선택하지 못하면 또 다른 리사이클러 문제가 발생합니다. 각 `VSN`이 거부된 이유를 파악하기 위해서 `-d` 옵션을 사용해 리사이클러를 실행할 수 있습니다. 여기에는 리사이클러가 리사이클할 `VSN`을 선택하는 방법에 대한 정보가 표시됩니다.

사용중인 환경에서 하드웨어 업그레이드하기

이 장에서는 기존 Sun StorEdge SAM-FS 환경 내의 하드웨어를 업그레이드하는 방법을 설명합니다. 다음 주제를 다룹니다.

- 184 페이지의 "자동화 라이브러리에 슬롯 추가"
- 185 페이지의 "라이브러리 업그레이드 또는 교체하기"
- 189 페이지의 "DLT 테이프 드라이브 업그레이드하기"

또한 Sun StorEdge SAM-FS 환경 내에서 다른 특정한 유형의 동작 및 업그레이드가 수행되어야 합니다. 다음 출판물에 이러한 동작들이 설명되어 있습니다.

- *Sun StorEdge QFS 및 Sun StorEdge SAM-FS 소프트웨어 설치 및 구성 안내서*는 Sun StorEdge QFS 및 Sun StorEdge SAM-FS 소프트웨어 업그레이드를 설명합니다.
- *Sun StorEdge QFS 및 Sun StorEdge SAM-FS 파일 시스템 관리 안내서*에서는 다음 유형의 작업과 업그레이드에 대해 설명합니다.
 - 파일 시스템 초기화
 - 시스템에 대한 구성 파일 변경 전파
 - 파일 시스템 마운트
 - 파일 시스템 마운트 해제
 - 파일 시스템 무결성 확인 및 파일 시스템 복구
 - 업그레이드를 위한 정보 보존
 - 하드웨어 장치 업그레이드 준비
 - 파일 시스템에 디스크 캐시 추가
 - 파일 시스템의 디스크 교체
 - 호스트 시스템 업그레이드
 - Solaris OS 업그레이드

자동화 라이브러리에 슬롯 추가

소프트웨어 라이선스는 Sun StorEdge SAM-FS에서 관리하는 카트리지 슬롯의 수를 제어합니다. 슬롯의 수를 늘리려면 이 섹션에서 설명하는 작업을 수행해야 합니다.

▼ 라이브러리에 슬롯을 추가하려면

1. 새로운 라이선스 키 세트를 공인 서비스 제공자(ASP)를 통해 구매야 할지 자신의 계정에 ASP가 지정되지 않은 경우 Sun Microsystems를 통해 구매야 할지를 결정합니다.

새 라이선스 키가 필요하지 않은 경우 2 단계를 진행합니다.

새 라이선스 키가 필요한 경우 다음 단계를 수행하십시오.

- a. 기존 라이선스 키를 새 라이선스 키로 교체합니다.

새 라이선스 키가 있으면 다음 작업을 수행합니다. 라이선스 키는 다음 파일의 1열에서 시작합니다.

```
/etc/opt/SUNWsamfs/LICENSE.4.1
```

다른 키워드, 호스트 ID 또는 기타 정보는 나타날 수 없습니다.

- b. `samd(1M) config` 명령을 실행하여 Sun StorEdge SAM-FS 소프트웨어가 새 라이선스 키를 인식하도록 합니다. (선택 사항)

새 라이선스 키가 있으면 다음 작업을 수행합니다. 예를 들어, 다음과 같습니다.

```
# samd config
```

2. 라이브러리 카탈로그를 언로드합니다.

sacmd(IM) unload 명령을 다음 형식으로 사용합니다.

```
samcmd unload eq
```

*eq*에 대해, *mcf* 파일의 정의에 따라 자동화 라이브러리의 장비 서수를 지정합니다. 이 명령은 라이브러리 카탈로그 항목을 기록자 카탈로그로 이동하고 각 카트리지에 대한 카탈로그 정보를 보존합니다.

이 명령이 실행된 후 *samu(IM) v* 화면을 사용하여 자동화 라이브러리의 *v* 화면이 비었는지와 기록자의 *v* 화면이 자동화 라이브러리에 있던 *VSN*으로 구성되어 있는지 확인합니다.

3. Sun StorEdge SAM-FS 시스템을 사용합니다.

이 작업의 수행 방법에 대한 자세한 내용은 13 페이지의 "Sun StorEdge SAM-FS 환경에서 자동화 라이브러리 및 수동으로 로드된 드라이브 사용하기"를 참조하십시오.

4. 제조업체의 권장 절차에 따라 호스트 시스템과 라이브러리를 종료합니다.

5. 라이브러리 하드웨어 엔지니어가 자동화 라이브러리에 슬롯을 추가합니다.

6. 정상적인 작동 개시 절차에 따라 호스트 시스템에 전원을 연결합니다.

7. Sun StorEdge SAM-FS 시스템을 시작합니다.

이 작업의 수행 방법에 대한 자세한 내용은 13 페이지의 "Sun StorEdge SAM-FS 환경에서 자동화 라이브러리 및 수동으로 로드된 드라이브 사용하기"를 참조하십시오. 새 라이센스 정보는 *samu(IM)* 유틸리티의 1 디스플레이에 나타납니다.

라이브러리 업그레이드 또는 교체하기

다른 자동화 라이브러리의 연결을 해제하고 설치하기 전에 *Sun StorEdge QFS* 및 *Sun StorEdge SAM-FS 파일 시스템 관리 안내서*의 "하드웨어 업그레이드 준비" 섹션에서 설명한 것처럼 업그레이드를 준비합니다.

▼ 라이브러리를 교체하거나 업그레이드하려면

1. 라이브러리 카탈로그를 언로드합니다.

samcmd(1M) unload 명령을 다음 형식으로 사용합니다.

```
samcmd unload eg
```

eg에 대해, mcf 파일의 정의에 따라 자동화 라이브러리의 장비 서수를 지정합니다. 이 명령은 라이브러리 카탈로그 항목을 기록자 카탈로그로 이동하고 각 카트리지에 대한 카탈로그 정보를 보존합니다.

이 명령이 실행된 후 samu(1M) v 화면을 사용하여 자동화 라이브러리의 v 화면이 비었는지와 기록자의 v 화면이 자동화 라이브러리에 있던 VSN으로 구성되어 있는지 확인합니다.

2. Update the /etc/opt/SUNWsamfs/inquiry.conf file. (선택 사항)

이 파일의 새 라이브러리는 업체, 자동화 라이브러리 모델 및 Sun StorEdge SAM-FS 내부 이름에 의해 식별됩니다.

예를 들어 릴리스된 inquiry.conf 파일에는 다음 행이 포함됩니다.

```
"HP", "C1710T", "hpoplib" # HP optical library
```

이 행은 시스템이 HP사가 제작한 C1710T 모델 SCSI 장치를 발견한 경우 이를 hpoplib로 구동한다는 의미입니다. 처음 2개의 필드(업체/제품)는 하드웨어 장치에서 반환됩니다. 마지막 필드인 hpoplib는 시스템에서 장치와 통신하는 방법을 결정하기 위해 내부적으로 사용하는 이름입니다. inquiry.conf 파일을 변경해야 하는 경우 변경 사항은 sam-amld 데몬을 재시작해야 적용됩니다.

3. 현재의 /etc/vfstab 파일을 /etc/vfstab.cur로 저장합니다.

4. /etc/vfstab 파일을 편집합니다.

모든 Sun StorEdge SAM-FS 마운트를 yes에서 no로 변경합니다.

5. /etc/opt/SUNWsamfs/archiver.cmd 파일을 archiver.cmd.cur로 저장합니다.

6. /etc/opt/SUNWsamfs/archiver.cmd 파일을 편집합니다.

wait 명령을 첫 번째 행으로 추가합니다.

7. 제조업체의 권장 절차에 따라 호스트 시스템과 주변 기기의 전원을 차단합니다.

8. 자동화 라이브러리의 연결을 해제합니다.

9. 연결 케이블을 새 자동화 라이브러리에 연결합니다.

- 권장 전원 연결 순서에 따라 주변 기기와 호스트 시스템에 전원을 연결합니다.
- 호스트 시스템에서 새 자동화 라이브러리를 식별하는지 확인합니다.
다음 명령을 입력합니다.

```
> probe-scsi-all
```

다음 단계를 진행하기 전에 새 자동화 라이브러리와 해당 드라이브가 표시되어야 합니다. 이러한 장치가 식별되지 않으면 자동화 라이브러리와 해당 드라이브는 연결상의 문제가 있는 것입니다.

- 시스템을 부팅합니다.
새 구성으로 부팅하려면 다음 명령을 입력합니다.

```
> boot -rv
```

- 드라이브 또는 자동화 라이브러리의 대상 번호가 바뀌거나 자동화 라이브러리의 드라이브 순서 또는 번호가 바뀐 경우에는 `/etc/opt/SUNWsamfs/mcf` 파일을 수정하여 새 구성을 반영합니다. (선택 사항)
이는 *Sun StorEdge QFS* 및 *Sun StorEdge SAM-FS* 소프트웨어 설치 및 구성 안내서에서 설명한 초기 설치와 유사합니다.
- 새 `/dev/samst` 항목을 만듭니다. (선택 사항)
새 장비를 추가하려면 이 작업을 수행합니다. 다음 명령을 입력합니다.

```
# samdev
```

15. Sun StorEdge SAM-FS 시스템을 초기화합니다.

파일 시스템을 마운트하거나 다음 명령을 입력하여 이 작업을 수행할 수 있습니다.

```
# samd start
```

시스템은 초기화되면 자동화 라이브러리의 슬롯 수가 변경되었음을 인식합니다. 시스템에서는 자동화 라이브러리에 대해 전체 감사를 실행하여 라이브러리 카탈로그를 업데이트합니다. 전체 감사는 아카이브가 재실행되기 전에 완료되어야 합니다.

감사에서 문제가 있는 경우 가장 흔한 원인은 자동화 라이브러리의 드라이브 순서가 /etc/opt/SUNWsamfs/mcf 파일의 순서와 일치하지 않기 때문입니다. 드라이브에 두 개의 속성 즉, 자동화 라이브러리의 SCSI 대상 ID 및 위치가 있음을 기억하십시오. 이러한 2가지 속성은 업그레이드 이전과 이후에 정확해야 합니다.

아무 문제없이 감사가 완료되면 다음 단계를 진행합니다.

16. /etc/vfstab 및 /etc/opt/SUNWsamfs/archiver.cmd 파일을 업그레이드 이전 버전으로 교체합니다.

저장된 /etc/vfstab.cur 및 /etc/opt/SUNWsamfs/archiver.cur 파일을 각각 사용합니다.

17. 시스템을 재부팅하여 구성에 오류가 없는지 확인합니다.

자동화 라이브러리에서는 위치 번호로 드라이브를 호출합니다. 시스템에서 카트리지를 드라이브에 로드하려면 가령 슬롯 123에서 드라이브 3으로 카트리지를 로드하도록 자동화 라이브러리에 명령을 보내야 합니다.

드라이브 3은 세 번째 mcf 항목에 따라 SCSI 대상 6이 될 수도 있습니다. 시스템에서는 SCSI 대상 6이 mcf 파일의 세 번째 드라이브 항목이기 때문에 SCSI 대상 6을 드라이브 3으로 인식합니다. 자동화 라이브러리에서는 SCSI 대상 6이 차지하는 물리적 위치에 SCSI 대상 6을 드라이브 3으로 인식합니다.

자동화 라이브러리가 요청되어 카트리지를 드라이브에 로드하면 시스템은 드라이브에 대해 장치 준비 상태를 테스트합니다. 여기에서 시스템은 mcf 파일의 /dev/samst/scsi-target 항목에서 정의된 SCSI 대상 ID를 사용합니다. 따라서 항목은 카트리지가 로드된 드라이브와 일치해야 합니다.

이 정보를 알 수 있는 좋은 방법은 없습니다. 대개는 제조업체가 드라이브에 SCSI ID가 오름차순으로 설정된 자동화 라이브러리를 판매하지만 이를 보증할 수는 없습니다. 이것은 결정하는 한 가지 방법은 samu(1M) 유틸리티의 :load 명령을 사용하여 카트리지를 로드한 후 samu(1M) 유틸리티의 s 화면을 관찰하여 t 화면의 상태 플래그에 p 보다는 r을 표시하는 드라이브를 찾는 것입니다.

DLT 테이프 드라이브 업그레이드하기

고밀도 고속 테이프 기술을 이용하려면 일반적으로 자동화 라이브러리의 DLT 테이프 드라이브나 독립형 테이프 드라이브를 업그레이드할 수 있습니다. 예를 들어 DLT 4000 드라이브에서 DLT 7000 드라이브로 업그레이드할 수 있습니다.

Sun StorEdge SAM-FS 환경에서 이것은 Sun StorEdge SAM-FS 환경을 시작하기 전에 필요한 경우 새 드라이브 추가, 새 구성 재시동 및 mcf 파일 업데이트의 문제입니다. 이외에도 여러 슬롯을 업그레이드하려면 해당 ASP 또는 Sun Microsystems에 문의하여 업그레이드된 라이선스가 필요한지 확인해야 합니다.

드라이브를 업그레이드하려면 먼저 다음 제한 사항과 일반 정보를 참고해야 합니다.

- Sun StorEdge SAM-FS 환경은 동일한 직접 연결 자동화 라이브러리 내의 혼합 DLT 테이프 드라이브를 지원하지 않습니다. 예를 들어 Sun StorEdge SAM-FS 시스템은 동일한 자동화 라이브러리에 있는 DLT 4000 테이프 드라이브와 DLT 7000 테이프 드라이브를 구분할 수 없습니다. 따라서 모든 DLT 드라이브를 동시에 새 드라이브로 교체해야 합니다.
- 저밀도 테이프는 고밀도 테이프 및 테이프 드라이브와 함께 사용할 수 있습니다. 고밀도 드라이브를 사용하면 저용량 테이프에 읽기 및 쓰기를 계속할 수 있습니다.
- 고밀도 DLT 테이프를 최대한 이용하려면 기존의 파일을 리사이클하여 이를 고밀도 테이프로 이동해야 합니다. 이를 위해서는 저밀도 테이프를 모두 읽기 전용으로 마크한 다음, 이를 리사이클용으로 마크합니다. 리사이클 테이프에 대한 자세한 내용을 보려면 167 페이지의 "리사이클"을 참조하십시오.
- 각 테이프에 레이블이 지정되면 테이프의 밀도는 라이브러리 카탈로그에 인식 및 기록됩니다.

▼ 테이프 드라이브를 업그레이드하려면

1. 현재 덤프 파일이 충분한지 여부를 알아봅니다. (선택 사항)

덤프 파일이 충분하지 않으면 파일 시스템에서 `samfsdump(1M)`를 수행한 후에 다음 단계를 진행합니다.

2. /kernel/drv/st.conf 파일을 업데이트하여 새 드라이브를 식별합니다.

테이프 드라이브는 이 파일에서 업체, 테이프 모델 및 Sun StorEdge SAM-FS 내부 이름에 의해 식별됩니다. 예를 들어 릴리스된 st.conf 파일에는 다음 행이 포함됩니다.

```
"QUANTUM DLT7000", "DLT 7000 tape drive", "dlt7-tape"
```

예제 파일은 /opt/SUNWsamfs/examples/st.conf_changes에 제공됩니다. /kernel/drv/st.conf에 대한 전체 파일에서 읽기를 하거나 필요한 변경 사항을 병합할 수 있습니다. st.conf 파일의 업데이트에 대한 자세한 내용은 *Sun StorEdge QFS 및 Sun StorEdge SAM-FS 소프트웨어 설치 및 구성 안내서*를 참조하십시오.

3. 제조업체의 권장 절차에 따라 호스트 시스템과 주변 기기의 전원을 차단합니다.

4. 테이프 드라이브를 새 드라이브로 교체합니다.

5. 권장 전원 연결 순서에 따라 주변 기기와 호스트 시스템에 전원을 연결합니다.

6. 호스트 시스템에서 새 드라이브를 식별하는지 확인합니다.

다음 명령을 입력합니다.

```
> probe-scsi-all
```

다음 단계를 진행하기 전에 자동화 라이브러리와 새 드라이브가 표시되어야 합니다. 이러한 장치가 표시되지 않으면 연결상의 문제가 있는 것이므로 해결해야 합니다. 이 명령에서 올바른 정보를 반환하면 다음 단계를 진행할 수 있습니다.

7. 시스템을 부팅합니다.

새 구성으로 부팅하려면 다음 명령을 입력합니다.

```
> boot -rv
```

8. /etc/opt/SUNWsamfs/mcf 파일을 수정하여 새 구성을 반영합니다. (선택 사항)

드라이브나 자동화 라이브러리의 대상 번호가 바뀌거나 자동화 라이브러리의 드라이브 순서 또는 번호가 바뀌면 이 단계를 수행합니다. 이는 *Sun StorEdge QFS 및 Sun StorEdge SAM-FS 소프트웨어 설치 및 구성 안내서*에서 설명한 초기 설치와 유사합니다.

9. 새 장비에 대해 새 /dev/samst 항목을 만듭니다. (선택 사항)

이러한 항목을 만들기 위해서는 다음 명령을 입력해야 합니다.

```
# samdev
```

10. Sun StorEdge SAM-FS 시스템을 시작합니다.

11. 파일 시스템을 마운트합니다.

기존 Sun StorEdge SAM-FS 테이프를 계속 사용할 수 있습니다.

고급 항목

이 장에서는 기본적인 시스템 관리 및 사용법 수준 이상의 고급 항목을 다룹니다.

다음 주제를 다룹니다.

- 193 페이지의 "장치 로깅"
- 196 페이지의 "제거 가능한 매체 파일"
- 198 페이지의 "세그먼트된 파일"
- 200 페이지의 "시스템 오류 기능 보고"

장치 로깅

장치 로깅 기능은 특정 유형의 장치 문제를 분석하는데 사용할 수 있는 장치별 오류 정보를 제공합니다. 장치 로깅 기능을 사용하면 자동화 라이브러리, 테이프 드라이브 또는 광 드라이브의 이벤트 실패 순서를 알 수 있습니다. 장치 로깅 기능은 소프트 매체 오류(예: 복원 가능한 읽기 오류)를 수집하지 않습니다.

장치 로깅 메시지는 개별 로그 파일에 작성됩니다. 자동화 라이브러리, 테이프 및 광 드라이브 장치 각각에 대한 로그 파일이 있고 기록자에 대해서는 하나의 로그 파일이 있습니다. 로그 파일의 위치는 `/var/opt/SUNWsamfs/devlog`입니다. 각 개별 로그 파일의 이름은 장비 서수와 동일합니다.

예제. Sun StorEdge SAM-FS 파일 시스템과 단일 Hewlett Packard 광 라이브러리의 광 드라이브가 2개라고 가정합니다.

코드 예 9-1에서는 mcf 파일을 보여줍니다.

코드 예 9-1 예제 mcf 파일

```
/dev/samst/clt5u0 40 hp hp40 - etc/opt/SUNWsamfs/hp40_cat
/dev/samst/clt4u0 41 mo hp40 -
/dev/samst/clt6u0 42 mo hp40 -
```

코드 예 9-2는 /var/opt/SUNWsamfs/devlog 파일을 표시합니다.

코드 예 9-2 devlog 파일

```
# pwd
/var/opt/SUNWsamfs/devlog
# ls
40      41      42      43
#
```

장치 43은 기록자입니다.

장치 로그의 사용 시점

장치 로그는 특히 모든 장치에 대한 로깅 옵션을 모두 사용하고 장치 활동이 많은 경우에 많은 로그 메시지를 쉽게 만들 수 있습니다. 처음에 장치 로그 설정은 다음 기본 값으로 설정됩니다.

```
err retry syserr date
```

Sun StorEdge SAM-FS 환경에 구성된 장치 중 하나에 문제가 있는 것으로 생각되면 해당 장치에 대해 추가 로깅 이벤트를 활성화해야 합니다. 또한 서비스 공급자의 권장에 따라 서비스 로깅을 활성화해야 하는 경우도 있습니다. 이러한 경우에는 이벤트를 detail로 설정해야 합니다. 경우에 따라서는 서비스 공급자의 권장에 따라 장치에 대한 이벤트를 all로 설정해야 합니다. 이렇게 하면 추가 로그 정보가 추가됩니다. 그러나 일반적으로 로깅이 과도한 상태에서는 시스템을 실행하지 않는 것이 바람직합니다.

장치 로그 정보는 samexplorer(1M) 명령이 실행되면 자동으로 수집됩니다. 이렇게 되면 파일 시스템 서비스를 통해 문제 분석 작업의 일환으로 가능한 장치 오류 정보를 검토할 수 있습니다.

장치 로그 활성화

2가지 방법을 사용해 장치 로그를 활성화할 수 있습니다.

방법 1과 방법 2 모두 다음 사항이 적용됩니다.

- *eq*는 *mcf* 파일에 정의된 장치의 Equipment Ordinal(장비 서수)이거나 모든 장치에 대한 키워드 *all*입니다.
- 장치 로깅 이벤트는 *samset(1M)* 매뉴얼 페이지에 나열되어 있습니다. 또한 예도 나열되어 있습니다. 장치 로그 메시지는 영문으로만 볼 수 있습니다. *이벤트*는 다음 목록에 나와있는 유형 중 하나 이상입니다.
 - *all*
 - *date*
 - *default*
 - *detail*
 - *err*
 - *event*
 - *label*
 - *mig*
 - *module*
 - *msg*
 - *none*
 - *retry*
 - *stage*
 - *stage_ck*
 - *syserr*
 - *tapealert*
 - *time*

두 가지 방법 중 하나로 장치 로그를 활성화할 수 있습니다. 이러한 절차는 다음과 같습니다.

- 195 페이지의 "*samset(1M)* 명령을 사용하여 장치 로그 활성화하기"
- 196 페이지의 "*defaults.conf* 파일을 편집하여 장치 로그 활성화하기"

▼ *samset(1M)* 명령을 사용하여 장치 로그 활성화하기

- *samset(1M)* 명령을 사용합니다.
예를 들어, 다음과 같습니다.

```
# samset devlog eq event
```

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

▼ defaults.conf 파일을 편집하여 장치 로그 활성화하기

1. 슈퍼유저가 됩니다.
2. vi(1) 또는 또 다른 편집기를 사용하여 /etc/opt/SUNWsamfs/defaults.conf 파일을 엽니다.
3. defaults.conf 파일의 devlog 명령을 추가합니다.
다음 명령을 추가합니다.

```
devlog eq event
```

eq에 대해 메시지를 로그하려는 장치의 장비 서수를 지정합니다.

이벤트에 대해 194 페이지의 "장치 로그 활성화"에서 설명된 하나 이상의 이벤트를 지정합니다. 두 개 이상의 이벤트를 지정한 경우 공백 문자를 사용하여 분리합니다.

Sun StorEdge SAM-FS 파일 시스템이 시작되면 자동으로 각 사용 가능한 장치에 대한 이벤트 유형을 default로 설정합니다. 또한 samset(1M) 명령을 사용하면 각 장치 로그에 대한 현재 설정을 결정할 수 있습니다.

4. defaults.conf 파일을 저장하고 닫습니다.
5. samd(1M) config 명령을 사용하여 defaults.conf 파일의 변경 내용을 전파합니다.

```
# samd config
```

제거 가능한 매체 파일

request(1) 명령을 사용하면 데이터 버퍼링 시 디스크 캐시를 사용하지 않는 파일을 수동으로 만들고, 쓰고, 읽을 수 있습니다. 이러한 방식으로 만든 파일을 제거 가능한 매체 파일이라고 합니다.

제거 가능한 매체 파일은 권한, 사용자 이름, 그룹 이름 및 크기 특징이 있다는 점에서 전형적인 Sun StorEdge SAM-FS 파일처럼 보입니다. 그러나 데이터는 디스크 캐시에 상주하지 않습니다. 그래서 디스크 캐시보다 큰 파일을 작성하고 제거 가능한 매체 카트리지에 기록할 수 있습니다. request(1) 명령에서 지정한 파일에 대해 시스템은 .inodes 파일의 inode 항목을 작성합니다. 사용자는 제거 가능한 매체에서 파일이 시작하는 위치를 몰라도 됩니다. (디스크 캐시의 데이터가 있는 파일에 대해 동일합니다.) Sun StorEdge SAM-FS 파일 시스템은 inode 항목의 해당 정보를 읽습니다. 제거 가능한 매체 파일 여러 개가 동일한 볼륨에 상주할 수 있습니다.

제거 가능한 매체 파일이 여러 볼륨을 스캔하는 경우 이를 볼륨 오버플로 파일이라고 합니다. 볼륨 오버플로 기능을 사용하면 시스템이 단일 파일을 여러 카트리지의 여러 볼륨으로 스캔할 수 있습니다. 볼륨 오버플로 파일은 제거 가능한 매체 파일의 한 유형입니다. 선택된 매체의 용량을 초과하는 아주 큰 파일이 있는 경우 볼륨 오버플로 기능이 유용합니다.

▼ 제거 가능한 매체 또는 볼륨 오버플로 파일 작성하기

1. `tplabel(1M)` 또는 `odlabel(1M)` 명령을 사용하여 테이프 또는 광자기 카트리지를 레이블할 수 있습니다.

이러한 명령에 대한 정보는 해당 각 매뉴얼 페이지를 참조하십시오.

2. `request(1M)` 명령을 사용합니다.

최소한의 경우 다음 옵션을 사용합니다.

```
request -m media_type -v vsn [vsn/vsn ...] [-l vsn_file] input_file
```

표 9-1 request(1M) 명령에 대한 인수

인수	의미
<i>media_type</i>	제거 가능한 매체 카트리지의 매체 유형. 올바른 <i>media_type</i> 에 대한 자세한 내용은 <code>mcf(4)</code> 매뉴얼 페이지를 참조하십시오.
<i>vsn</i>	제거 가능한 매체 카트리지의 볼륨 시리얼 이름. 두 개 이상의 <i>vsn</i> 을 지정한 경우, 볼륨 오버플로 파일을 작성합니다. 볼륨 오버플로 파일에 대해 최고 256 개의 <i>vsn</i> 을 지정할 수 있습니다. 정방향 슬래시 문자(/)를 사용하여 <i>vsn</i> 인수를 분리합니다. 지정된 <i>vsn</i> 은 자동화 아카이브를 위해 Sun StorEdge SAM-FS 환경에서 사용되는 볼륨 사이에 없어야 합니다. 아카이브는 아카이브할 다음 파일을 현재 테이더의 끝에 추가하고 EOF 레이블을 매번 데이터 외부로 이동합니다.
<i>vsn_file</i>	<i>vsn</i> 목록을 포함하는 입력 파일. <i>vsn</i> 이 많은 경우, 명령줄보다는 입력 파일에 <i>vsn</i> 목록을 지정하는 것이 더 쉬울 수도 있습니다.
<i>input_file</i>	제거 가능한 매체 카트리지에 기록되는 파일. 이 파일은 Sun StorEdge SAM-FS 파일 시스템에 상주해야 합니다.

예제 1. 다음 명령은 제거 가능한 매체 파일을 작성합니다.

```
# request -m lt -v aaa rem1
```

예제 2. 다음 명령을 세 개의 볼륨에 볼륨 오버플로 파일을 작성합니다.

```
# request -m lt -v TAPE01/TAPE02/TAPE03 large.file
```

제거 가능한 매체 파일을 순차적으로 읽고 써야 합니다. mcf 파일에 정의된 자동화 라이브러리에 볼륨이 상주하는 경우 Sun StorEdge SAM-FS 파일 시스템이 자동으로 요청된 볼륨을 마운트합니다.

볼륨에 제거 가능한 매체 파일이 있으면 해당 볼륨은 리사이클되지 않습니다. 리사이클러에서는 아카이브된 파일만이 아카이브 지정된 특정 볼륨에 상주해야 합니다. 이외에도 제거 가능한 매체 파일은 아카이브하지 않습니다.

제거 가능한 매체 파일은 NFS를 통해 지원되지 않습니다.

request(1) 명령을 사용하면 아카이버의 일반적인 기능들을 무시합니다.

제거 가능한 매체 파일 만들기 방법을 설명한 예제는 request(1) 매뉴얼 페이지를 참조하십시오.

세그먼트된 파일

Sun StorEdge SAM-FS 환경은 세그먼트된 파일을 지원합니다. 파일을 세그먼트하면 대용량 파일에 대한 테이프 저장 검색 속도, 액세스 및 관리 효율성을 향상시킬 수 있습니다. 세그먼트된 파일의 크기는 실제 디스크 캐시보다 클 수 있습니다. 세그먼트된 파일은 한 번에 파일의 일부만 디스크 캐시에 상주할 수 있습니다.

segment(1) 명령을 사용하면 세그먼트 크기를 지정할 수 있습니다. 세그먼트 크기는 현재 파일 크기보다 작게 설정할 수 없습니다.

세그먼트된 파일은 테이프 스트라이프를 지원합니다. 파일을 세그먼트하면 여러 테이프 장치에 대해 동시에 스트라이프를 할 수 있어 파일 세그먼트를 저장하는 데 필요한 시간을 크게 줄일 수 있습니다. 사용자가 전체 파일이 아닌 원하는 파일 세그먼트만 검색할 수 있도록 하면 데이터 액세스 속도가 빨라집니다.

세그먼트를 하면 변경된 파일의 부분만 재아카이브되기 때문에 아카이브 효율이 향상됩니다. 파일 세그먼트는 병렬로 아카이브할 수 있으며 세그먼트된 파일은 병렬로 스테이지할 수 있습니다. 이렇게 하면 아카이브와 검색 성능이 향상됩니다.

세그먼트는 파일, 디렉토리 또는 전체 파일 시스템에 대해 활성화할 수 있습니다. 세그먼트된 파일은 모든 다른 Sun StorEdge SAM-FS 기능을 지원합니다.

다음 섹션에서는 세그먼트된 파일과 세그먼트되지 않은 파일 간의 차이점에 대해 설명합니다. 세그먼트된 파일에 대한 자세한 내용은 `segment(1)` 또는 `sam_segment(3)` 매뉴얼 페이지를 참조하십시오.

아카이브

세그먼트된 파일의 경우 아카이브할 수 있는 단위는 파일이 아니라 세그먼트 자체입니다. 모든 아카이브 등록 정보와 우선 순위는 파일이 아니라 개별 세그먼트에 적용됩니다.

아카이브 단위는 세그먼트입니다. `archiver.cmd` 파일의 아카이브 세트에 대해 `-drives` 및 `-drivemin` 매개변수 모두를 지정하여 세그먼트를 스트라이프할 수 있습니다.

예를 들어 파일 시스템에 100MB 용량의 세그먼트된 파일이 있고, 이 파일의 세그먼트 크기가 10MB라고 가정합니다. `archiver.cmd` 파일이 `-drives 2` 명령을 사용해 설정한 아카이브를 정의한다면 이 파일은 2개의 드라이브로 병렬 아카이브됩니다. 세그먼트 1, 3, 5, 7 및 9는 첫 번째 드라이브를 사용해 아카이브되고 세그먼트 2, 4, 6, 8 및 10은 두 번째 드라이브를 사용해 아카이브됩니다.

전체 파일이 아카이브되는 것이 아니라 수정된 세그먼트만 아카이브됩니다. 각 세그먼트에 대해 아카이브 복사본을 최대 4개까지 만들 수 있습니다. Sun StorEdge SAM-FS는 세그먼트에 대한 볼륨 오버플로를 지원합니다.

참고 - 세그먼트된 파일의 인덱스에는 사용자 데이터가 포함되지 않습니다. 인덱스는 메타 데이터로 간주됩니다. 인덱스는 파일 시스템 아카이브 세트에 할당됩니다.

재해 복구

재해가 발생한 경우 세그먼트된 파일의 복구에 관한 내용은 *Sun QFS, Sun SAM-FS 및 Sun SAM-QFS 재난 복구 안내서*를 참조하십시오.

시스템 오류 기능 보고

시스템 오류 기능(SEF) 보고 시스템은 자동화 라이브러리의 테이프 장치에서 로그 감지 데이터를 캡처하여 이를 로그 파일에 작성하고 사람이 관독할 수 있는 형식으로 변환합니다. 시스템 오류 기능 보고 시스템은 다음 요소로 구성됩니다.

- 테이프 장치 로그 감지 페이지의 데이터가 포함된 로그 파일.
- stdout에 사람이 관독할 수 있는 형식으로 로그 파일을 작성하기 위한 `sefreport(1M)` 명령. 이 로그 파일은 사용자 제공 분석 스크립트에 대한 입력으로 사용할 수 있습니다.

`sefreport(1M)` 명령은 Sun StorEdge SAM-FS SEF 로그 파일의 내용을 읽습니다. 로그 파일에는 Sun StorEdge SAM-FS 환경에서 사용된 주변 테이프 장치의 로그 감지 페이지에서 모은 데이터가 있습니다. 로그 감지 페이지는 업체마다 다릅니다. 매개 변수 코드, 제어 비트 및 매개 변수 값의 의미에 대해서는 각각의 특정 장치에 대한 업체의 설명서를 참조하십시오.

SEF 보고는 독립형 테이프 드라이브에서 사용할 수 없습니다. SEF 보고는 `tapealert(1M)` 기능을 지원하지 않는 더 오래된 SCSI-2 장치에 대해 가장 유용합니다. 자세한 내용은 `tapealert(1M)` 매뉴얼 페이지를 참조하십시오.

▼ SEF 보고를 활성화하려면

1. 슈퍼유저가 됩니다.
2. `mkdir(1)` 명령을 사용해 SEF 디렉토리를 만듭니다.

예를 들어, 다음과 같습니다.

```
# mkdir /var/opt/SUNWsamfs/sef
```

3. `touch(1)` 명령을 사용해 SEF 보고를 활성화합니다.

`sefdata` 로그 파일을 만들어 설치를 끝낸 후에는 언제라도 SEF 보고를 활성화할 수 있습니다. 처음에는 SEF 로그 파일이 비어있어야 합니다.

```
# touch /var/opt/SUNWsamfs/sef/sefdata
```

위 예제의 명령에서는 `/var/opt/SUNWsamfs/sef/sefdata`에 만드는 SEF 로그 파일을 보여줍니다 이것이 기본 위치입니다.

4. `samd(1M) stop` 및 `samd(1M) start`를 사용하여 SEF 보고를 초기화합니다.
예를 들어, 다음과 같습니다.

```
# samd stop
# samd start
```

SEF 데이터는 생성되면서 로그 파일에 추가됩니다.

SEF 보고를 구성하여 대체 위치에서 로그 감지 데이터를 로그 및 읽을 수 있습니다. 대체 위치에서 로그 감지 데이터를 읽는 것에 관한 자세한 내용은 `sefreport(1M)` 매뉴얼 페이지를 참조하십시오.

SEF 보고 결과

`sefreport(1M)` 명령을 사용하려면 먼저 `/opt/SUNWsamfs/sbin`이 현재 명령 경로에 있는지 확인해야 합니다. SEF 보고 결과는 헤더 행과 로그 감지 데이터로 구성됩니다.

헤더 행 다음에 레코드의 각 페이지에 대한 로그 감지 데이터가 인쇄됩니다. 각 로그 감지 페이지에 대해 페이지 코드를 식별하는 행이 인쇄되고 나서 열 머리글 행이 인쇄됩니다. 그러면 다음 머리글과 함께(행당 세 개의 열) 데이터가 인쇄됩니다. `param code`, `control` 및 `param value`. 모든 데이터는 16진수로 생성됩니다.

▼ SEF 출력 생성하기

- **`sefreport(1M)` 명령을 사용하여 SEF 출력을 생성합니다.**

다음은 `sefreport(1M)` 명령에서 가장 많이 사용하는 2가지 옵션입니다.

- **`-d` 옵션.** `-d` 옵션은 추가 장치 정보를 생성합니다. 이 옵션은 각 레코드에 대한 장치의 장비 서수와 경로 이름이 포함된 추가 헤더 행을 씁니다. 이 옵션을 사용하면 특정 장치에 관련된 SEF 레코드를 쉽게 검색하고 찾을 수 있습니다.

- **`-v` 옵션 또는 `-t` 옵션.**

`-v` 옵션은 세부 정보 표시 모드로 정보를 생성합니다. 이 옵션은 레코드의 각 행에 장비 서수, 페이지 코드 및 VSN에 관한 정보를 추가합니다. 이 옵션을 사용하면 특정 장치나 특정 볼륨에 관련된 행만 선택할 수 있습니다.

`-t` 옵션은 텍스트 설명이 있는 로그 감지 출력을 생성합니다. 로그 감지 데이터 출력의 각 행의 경우, 해당 보고에는 장비 서수, 페이지 코드, VSN 및 매개변수 코드 설명이 포함된 추가 문자열이 포함됩니다.

동일한 명령줄에 `-t` 및 `-v`를 지정하지 마십시오. 이 명령들은 함께 사용할 수 없습니다.

예를 들어 다음 SEF 명령은 기본 위치에서 SEF 로그 파일을 읽고 각 장치의 장치 번호와 경로 이름을 쓴 다음, 결과를 생성합니다.

```
# sefreport -d /var/opt/SUNWsamfs/sef/sefdata > sef.output
```

코드 예 9-3에서는 sef.output 파일의 내용을 보여줍니다.

코드 예 9-3 sef.output 내용

```
Record no. 1
Mon Mar 26 11:17:48 2001 STK          9840          1.25 VSN 002981
  Eq no. 32   Dev name /dev/rmt/1cbn

PAGE CODE 2
param code  control  param value
  00h         74h     0x0
  01h         74h     0x0
  02h         74h     0x0
  03h         74h     0x0
  04h         74h     0x0
  05h         74h     0x40050
  06h         74h     0x0

PAGE CODE 3
param code  control  param value
  00h         74h     0x0
  01h         74h     0x0
  02h         74h     0x0
  03h         74h     0x0
  04h         74h     0x0
  05h         74h     0x140
  06h         74h     0x0

PAGE CODE 6
param code  control  param value
  00h         74h     0x0

Record no.
Mon Mar 26 11:30:06 2001 STK          9840          1.25 VSN 002999
  Eq no. 31   Dev name /dev/rmt/0cbn

PAGE CODE 2
param code  control  param value
  00h         74h     0x0
```

코드 예 9-3

sef.output 내용 (계속)

01h	74h	0x0
02h	74h	0x0
03h	74h	0x0
04h	74h	0x0
05h	74h	0x1400a0
06h	74h	0x0
PAGE CODE 3		
param code	control	param value
00h	74h	0x0
01h	74h	0x0
02h	74h	0x0
03h	74h	0x0
04h	74h	0x0
05h	74h	0x190
06h	74h	0x0
PAGE CODE 6		
param code	control	param value
00h	74h	0x0
Record no. 3		
Mon Mar 26 11:30:23 2001	STK 9840	1.25 VSN 002981
Eq no. 32	Dev name /dev/rmt/lcbn	
PAGE CODE 2		
param code	control	param value
00h	74h	0x0
01h	74h	0x0
02h	74h	0x0
03h	74h	0x0
04h	74h	0x0
05h	74h	0x18400f0
06h	74h	0x0
PAGE CODE 3		
param code	control	param value
00h	74h	0x0
01h	74h	0x0
02h	74h	0x0

코드 예 9-3

sef.output 내용 (계속)

03h	74h	0x0
04h	74h	0x0
05h	74h	0x1e0
06h	74h	0x0
PAGE CODE 6		
param code	control	param value
00h	74h	0x0
.		
.		
.		

참고 - 위 결과는 이 매뉴얼에 포함될 수 있도록 일부를 생략하였습니다.

파일의 내용과 형식 등 SEF 로그 파일에 대한 자세한 내용은 `sefdata(4)` 매뉴얼 페이지를 참조하십시오. 옵션 SEF 보고 형식에 관한 자세한 내용은 `sefreport(1M)` 매뉴얼 페이지를 참조하십시오.

SEF 로그 파일 관리

다른 모든 Sun StorEdge SAM-FS 로그 파일을 관리하는 것과 동일하게 SEF 로그 파일을 관리합니다. `cron(1)` 작업은 정기적으로 실행하여 현재 로그 파일을 다른 위치에 저장하거나 기존의 SEF 파일을 삭제하거나 새(비어있는) SEF 파일을 만들거나 다른 작업을 수행할 수 있습니다.

또한 `log_rotate.sh(1M)` 유틸리티를 사용해 SEF 로그 파일을 회전할 수 있습니다.

SEF 로그 파일 관리 도구에 대한 자세한 내용은 `cron(1)` 또는 `log_rotate.sh(1M)` 매뉴얼 페이지를 참조하십시오.

용어 해설

A

- 지정 가능한 저장소** Sun StorEdge QFS 또는 Sun StorEdge SAM-FS 파일 시스템을 통해 사용자가 참조하는 온라인, 근거리, 오프사이트 및 오프라인 저장소를 포함한 저장 공간.
- 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.
- 아카이버** 파일 복사를 제거 가능한 카트리지로 자동 제어하는 아카이브 프로그램.
- 감사(전체)** VSN을 확인하기 위해 카트리지를 로드하는 프로세스. 광자기 카트리지의 경우, 용량 및 공간 정보가 파악되고 자동화 라이브러리의 카탈로그에 입력됩니다.
- 자동화 라이브러리** 운영자의 간섭 없이 제거 가능한 매체 카트리지를 자동으로 로드 및 언로드하기 위해 설계된 로봇 제어 장치. 자동화 라이브러리에는 하나 이상의 드라이브와 카트리지를 저장소 슬롯 및 드라이브로 이동하는 전송 장치가 포함됩니다.

B

- 백업 저장소** 예기치 않은 손실을 대비하기 위한 파일 모음의 스냅샷. 백업에는 파일의 속성 및 관련 데이터가 모두 포함됩니다.
- 블록 할당 맵** 디스크에서 사용 가능한 저장소의 블록 및 해당 블록이 사용 중이거나 사용 가능한지의 여부를 나타내는 비트맵.
- 블록 크기** DAU를 참조하십시오.

C

- 카트리지** 데이터 기록을 위한 매체를 포함하고 있는 물리적인 엔티티. 테이프 또는 광 디스크입니다. *하나의 매체, 블록 또는 미디어*라고도 합니다.
- 카탈로그** 자동화 라이브러리에서 VSN 레코드. 각 자동화 라이브러리에는 하나의 카탈로그가 있고, 사이트에는 모든 자동화 라이브러리에 대한 하나의 기록자가 있습니다.
- 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.
- 연결** 안정적인 스트림 전달 서비스를 제공하는 두 개의 프로토콜 모듈 사이의 경로. TCP 연결은 한 컴퓨터의 TCP 모듈에서 다른 컴퓨터의 TCP 모듈로 확장됩니다.

D

- 데이터 장치** Sun StorEdge QFS 또는 Sun StorEdge SAM-FS 파일 시스템의 경우, 파일 데이터가 저장되는 장치 또는 장치 그룹.
- DAU** (디스크 할당 장치) 온라인 저장소의 기본 단위. 블록 크기라고도 합니다.
- 또한 Sun StorEdge QFS 파일 시스템은 16킬로바이트에서 65,528킬로바이트까지 완전히 조정 가능한 DAU를 지원합니다. 지정하는 DAU는 8킬로바이트의 배수여야 합니다.
- Sun StorEdge SAM-FS 파일 시스템인 작은 DAU와 큰 DAU 모두를 지원합니다. 작은 DAU는 4 킬로바이트(2¹⁴ 또는 4096 바이트)입니다. 큰 DAU는 16, 32 또는 64킬로바이트입니다. 사용 가능한 DAU 크기 쌍은 4 및 16, 4 및 32 및 4 및 64입니다.
- 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 Sun StorEdge SAM-FS 파일 시스템 내의 소프트웨어.
- 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 자동화 라이브러리. SCSI 연결 라이브러리는 자동화 라이브러리에 SCSI 표준을 사용하여 Sun StorEdge SAM-FS 소프트웨어에서 직접 제어합니다.

직접 I/O	대형 블록이 정렬된 순차적 I/O에 사용된 속성. <code>setfa(1)</code> 명령의 <code>-D</code> 옵션은 직접 I/O 옵션입니다. 이 옵션은 파일 또는 디렉토리에 대해 직접 I/O 속성을 설정합니다. 디렉토리에 적용되면 직접 I/O 속성이 상속됩니다.
디렉토리	파일 시스템 내에서 다른 파일 및 디렉토리를 가리키는 파일 데이터 구조.
디스크 할당 단위	DAU를 참조하십시오.
디스크 버퍼	Sun SAM-Remote 소프트웨어를 사용하는 경우, 디스크 버퍼는 클라이언트에서 서버로 데이터를 아카이브할 때 사용되는 서버 시스템의 버퍼입니다.
디스크 캐시	Sun StorEdge SAM-FS 파일 시스템 소프트웨어에서 디스크가 상주하는 부분. 온라인 디스크 캐시와 아카이브 매체 사이에서 데이터 파일을 만들고 관리하는 데 사용됩니다. 개별 디스크 파티션 또는 전체 디스크가 디스크 캐시로 사용될 수 있습니다.
디스크 공간 임계값	사용자가 사용할 수 있도록 관리자가 정의한 디스크 공간의 양. 적합한 디스크 캐시 사용 범위를 정의합니다. 상한 임계값은 디스크 캐시 사용량의 최대 수준을 나타냅니다. 하한 임계값은 디스크 캐시 사용량의 최소 수준을 나타냅니다. 릴리서는 이와 같이 미리 정의된 디스크 공간 임계값을 기준으로 디스크 캐시 사용량을 제어합니다.
디스크 스트라이프	여러 디스크에 걸쳐 파일을 기록하는 프로세스로, 액세스 성능이 높아지고 전체적인 저장 용량이 증가합니다. 스트라이프 항목도 참조하십시오.
드라이브	제거 가능한 미디어 볼륨 사이에 데이터를 전송하기 위한 메커니즘.

E

이더넷	근거리, 패킷 스위칭 네트워크 기술. 원래는 동축 케이블용으로 개발되었으며, 현재는 STP(shielded twisted-pair) 케이블을 통해 사용되고 있습니다. 이더넷은 초당 10 또는 100 메가바이트 LAN입니다.
위치 배열	파일에 할당된 각 데이터 블록이 디스크의 어디에 위치하는지 정의하는 파일 inode 내의 배열.

F

패밀리 장치 세트	패밀리 세트를 참조하십시오.
Family Set	디스크 모음이나 자동화 라이브러리 내의 드라이브와 같이 독립적인 물리적 장치의 그룹으로 표현되는 저장 장치. 또한 저장소 패밀리 세트를 참조하십시오.
FDDI	(광섬유 분산 데이터 인터페이스)100MB/초 광섬유 LAN.

광섬유 채널 장치간에 고속의 직렬 통신을 지정하는 ANSI 표준. 광섬유 채널은 SCSI-3에서 버스 아키텍처 중 하나로 사용됩니다.

광섬유 분산 데이터 인터페이스 FDDI를 참조하십시오.

파일 시스템 파일 및 디렉토리의 계층적 모음.

파일 시스템별 명령 전역 명령을 사용하는 아카이버 및 릴리서 명령으로, 특정 파일 시스템에 따라 다르고 fs =로 시작됩니다. 파일 시스템별 명령은 다음 fs = 명령행이 오거나 파일의 끝에 도달할 때까지 적용됩니다. 여러 명령이 파일 시스템에 영향을 미칠 경우, 파일 시스템별 명령은 전역 명령보다 우선합니다.

FTP (파일 전송 프로토콜) TCP/IP 네트워크를 통해 두 호스트 사이에 파일을 전송하기 위한 인터넷 프로토콜.

G

전역 명령 모든 파일 시스템에 적용되고 첫 번째 fs = 행 앞에 나타나는 아카이버 및 릴리서 명령.

유예 기간 디스크 할당량에서 사용자가 자신의 소프트 제한에 도달한 후 파일을 작성하고 저장소를 할당할 수 있는 시간.

H

하드 제한 디스크 할당량에서 사용자가 초과할 수 없는 파일 시스템 리소스, 블록 및 inode에 대한 최대 한도.

I

간접 블록 저장소 블록의 목록을 포함하는 디스크 블록. Sun StorEdge QFS 및 Sun StorEdge SAM-FS 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.

inode 인덱스 노드(index node). 파일을 기술하기 위해 파일 시스템에 의해 사용되는 데이터 구조. inode는 이름 이외의 파일과 관련된 모든 속성을 기술합니다. 속성에는 소유권, 액세스, 권한, 크기 및 디스크 시스템에서 파일 위치가 포함됩니다.

inode 파일 파일 시스템에 상주하는 모든 파일에 대한 inode 구조를 포함하는 특수한 파일 (.inodes). 모든 Sun StorEdge QFS 및 Sun StorEdge SAM-FS inode 길이는 512 바이트입니다. Inode 파일은 Sun StorEdge QFS 파일 시스템의 파일 데이터에서 분리된 메타데이터 파일입니다.

K

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.

L

LAN 근거리 통신망(Local Area Network).

임대 Sun StorEdge QFS 공유 파일 시스템에서, 클라이언트 호스트에게 임대가 유효한 동안 파일에 대한 작업을 수행할 수 있는 권한을 부여합니다. 메타 데이터 서버는 각 클라이언트 호스트에게 임대를 부여합니다. 파일 작업을 계속 수행할 수 있도록 필요에 따라 임대를 갱신할 수 있습니다

라이브러리 자동화 라이브러리를 참조하십시오.

라이브러리 카탈로그 카탈로그를 참조하십시오.

지역 파일 시스템 Sun Cluster의 한 노드에 설치되고 또 다른 노드에 크게 사용할 수 없는 파일 시스템. 또한 독립형 서버에 설치된 파일 시스템.

LUN 논리적 단위 번호(Logical Unit Number).

M

mcf 마스터 구성 파일. Sun StorEdge QFS 또는 Sun StorEdge SAM-FS 환경 내에서 장치 (토폴로지) 사이의 관계를 정의하는 초기화 시 읽게 되는 파일.

media 테이프 또는 광 디스크 카트리지.

매체 리사이클	적게 사용하는 아카이브 매체 리사이클 또는 재사용 프로세스. 적게 사용하는 아카이브 매체는 활성 파일이 거의 없는 아카이브 매체입니다.
메타 데이터	데이터에 대한 데이터. 메타 데이터는 디스크에서 파일의 정확한 데이터 위치를 찾는 데 필요한 인덱스 정보입니다. 파일, 디렉토리, 액세스 제어 목록, 심볼 링크, 제거 가능한 매체, 세그먼트된 파일 및 세그먼트된 파일의 인덱스에 대한 정보로 구성됩니다. 데이터가 손실된 경우, 손실된 데이터를 복원하려면 먼저 데이터를 찾는 메타 데이터가 복원되어야 하므로 메타 데이터는 보호되어야 합니다.
메타 데이터 장치	Sun StorEdge QFS 파일 시스템 메타 데이터가 저장되는 별도의 장치(예: 독립적으로 작동하는 디스크 또는 미러 장치). 파일 데이터와 메타 데이터를 분리하면 성능을 높일 수 있습니다. mcf 파일에서 메타 데이터 장치는 ma 파일 시스템 내에서 mm 장치로 선언됩니다.
미러 쓰기	단일 디스크 장애 발생 시 데이터 손실을 막기 위해 별도의 디스크 세트에 두 개의 파일 복사본을 유지하는 프로세스.
마운트 지점	파일 시스템이 마운트되는 디렉토리.
다중 판독기 파일 시스템	Sun StorEdge QFS 다중 판독기 파일 시스템은 여러 호스트에서 마운트될 수 있는 파일 시스템을 지정할 수 있는 단일 작성기, 다중 판독기 기능입니다. 여러 호스트가 파일 시스템을 읽을 수 있지만, 하나의 호스트만 파일 시스템에 쓸 수 있습니다. 다중 판독기 호스트는 mount(1M) 명령에서 -o reader 옵션으로 지정됩니다. 단일 작성기 호스트는 mount(1M) 명령에서 -o writer 옵션으로 지정됩니다. mount(1M) 명령에 대한 자세한 내용은 mount_samfs(1M) 매뉴얼 페이지를 참조하십시오.

N

이름 공간	파일, 해당 속성 및 해당 저장 위치를 식별하는 파일 모음의 메타 데이터 부분.
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.
네트워크로 연결된 자동화 라이브러리	StorageTek, ADIC/Grau, IBM, Sony 등 벤더에서 제공한 소프트웨어 패키지를 사용하여 제어되는 라이브러리. Sun StorEdge SAM-FS 파일 시스템은 자동화 라이브러리용으로 특별히 디자인된 Sun StorEdge SAM-FS 매체 교환기 데몬을 사용하는 업체 소프트웨어와 인터페이스합니다.
NFS	네트워크 파일 시스템(Network File System). 유형이 다른 환경에서 원격 파일 시스템에 대한 투명한 액세스를 제공하는 Sun의 분산 파일 시스템.
NIS	SunOS 4.0(최소) Network Information Service. 네트워크에서 시스템 및 사용자에게 대한 주요 정보를 포함하고 있는 분산 네트워크 데이터베이스. NIS 데이터베이스는 주 서버(master server) 및 모든 종 서버(slave server)에 저장됩니다.

O

- 오프라인 저장소** 로드를 위해 운영자의 간섭이 필요한 저장소.
- 외부 사이트 저장소** 서버와 떨어져 있고, 재난 복구를 위해 사용되는 저장소.
- 온라인 저장소** 즉시 사용이 가능한 저장소(예: 디스크 캐시 저장소).

P

- 파티션** 장치의 일부 또는 광자기 카트리지의 한 면.
- 사전 할당** 파일에 쓰기 위해 디스크 캐시에서 연속되는 공간을 확보하는 프로세스. 따라서 공간이 연속으로 유지됩니다. 사전 할당은 크기가 0인 파일에 대해서만 수행될 수 있습니다. 즉, `setfa -1` 명령은 크기가 0인 파일에 대해서만 지정할 수 있습니다. 자세한 내용은 `setfa(1)` 매뉴얼 페이지를 참조하십시오.
- 미리 보기 요청의 우선 순위 지정** 바로 처리할 수 없는 아카이브 및 스테이지 요청에 대한 우선 순위 지정.
- 가상 장치** 연결된 하드웨어가 없는 소프트웨어 하위 시스템 또는 드라이버.

Q

- 할당량** 사용자가 사용할 수 있는 시스템 리소스의 양.

R

- RAID** 독립된 디스크의 중복 배열(Redundant Array of 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.
- 리사이클러** 만료된 아카이브 복사본이 차지하는 카트리지의 공간을 사용하는 Sun StorEdge SAM-FS 유틸리티.

릴리스 우선 순위 여러 가중치와 해당 파일 등록 정보를 곱한 후 결과를 합산하여 파일 시스템 내에서 파일의 릴리스 우선 순위를 계산하는 방식.

릴리서 아카이브된 파일을 확인하고 해당 디스크 캐시 복사본을 릴리스해서 더 많은 디스크 캐시 공간을 사용하는 Sun StorEdge SAM-FS 구성요소 릴리서는 온라인 디스크 저장소의 양을 상한 및 하한 임계값으로 자동 조절합니다.

원격 프로시저 호출 RPC를 참조하십시오.

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.

로봇 저장소 슬롯과 드라이브 사이에 카트리지를 옮기는 자동화 라이브러리의 일부. 전송 장치라고도 합니다.

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 I/O의 크기를 결정합니다.

기본적으로 Sun StorEdge QFS 및 Sun StorEdge SAM-FS 파일 시스템은 스크라이프 그룹이 없는 경우 스트라이프 데이터 액세스를 구현합니다. 라운드 로빈 액세스가 지정된 경우에는 파일이 라운드 로빈됩니다. 파일 시스템에 일치하지 않는 스트라이프 그룹이 포함되어 있는 경우에는 스트라이프가 지원되지 않고 라운드 로빈이 강제 적용됩니다.

디스크 스트라이프 및 스트라이프 항목도 참조하십시오.

RPC 원격 프로시저 호출(Remote Procedure Call). 사용자 정의 네트워크 데이터 서버를 구현하기 위해 NFS에 의해 사용되는 기본 데이터 교환 메커니즘.

S

samfsdump 컨트롤 구조 덤프를 만들고 해당하는 파일 그룹에 대한 모든 컨트롤 구조 정보를 복사하는 프로그램. UNIX tar(1) 유틸리티와 유사하지만, 일반적으로 파일 데이터를 복사하지는 않습니다.

samfsrestore 컨트롤 구조 덤프로부터 inode 및 디렉토리 정보를 복원하는 프로그램.

SCSI 소형 컴퓨터 시스템 인터페이스(Small Computer System Interface). 디스크 및 테이프 드라이브, 자동화 라이브러리 등과 같은 주변 장치에 대해 일반적으로 사용되는 전기 통신 사양.

소형 컴퓨터 시스템 인터페이스 SCSI를 참조하십시오.

소프트 제한	디스크 할당량에서 일시적으로 초과할 수 있는 파일 시스템 리소스(블록 및 inode)에 대한 임계값 한도. 소프트 제한이 초과되면 타이머가 시작됩니다. 지정된 시간(기본값은 1주) 동안 소프트 제한을 초과한 경우에는 소프트 제한보다 낮게 파일 시스템 사용량을 줄일 때까지 더 이상 시스템 리소스를 할당할 수 없습니다.
스테이징	근거리 파일 또는 오프라인 파일을 아카이브 저장소에서 다시 온라인 저장소로 복사하는 프로세스.
저장소 패밀리 세트	단일 디스크 패밀리 장치로 집합적으로 표현되는 디스크 세트.
저장소 슬롯	카드리지가 드라이브에서 사용되지 않을 때 저장되는 자동화 라이브러리 내부의 위치. 라이브러리가 직접 연결되어 있는 경우, 저장소 슬롯의 내용이 자동화 라이브러리 카탈로그에 보관됩니다.
스트라이프 크기	다음 스트라이프 장치로 이동하기 전에 할당할 디스크 할당 단위(DAU)의 수. stripe=0인 경우, 파일 시스템은 스트라이프 액세스가 아닌 라운드 로빈 액세스를 사용합니다.
스트라이프 그룹	Sun StorEdge QFS 파일 시스템 내의 장치 모음 및 하나(대개 둘) 이상의 gXXX 장치로 mcf 파일에 정의된 장치 모음. 스트라이프 그룹은 하나의 논리적 장치로 취급되고 언제나 디스크 할당 단위(DAU)와 동일한 크기로 스트라이프됩니다. 파일 시스템 내에 최대 128개의 스트라이프 그룹을 지정할 수 있지만, 총 252개 이상의 장치를 지정할 수 없습니다.
스트라이프	파일이 인터레이스 방식으로 논리 디스크에 동시에 쓰여지는 데이터 액세스 방식. 모든 Sun StorEdge QFS 및 Sun StorEdge SAM-FS 파일 시스템을 사용하여 각 개별 파일 시스템에 대해 스트라이프 또는 라운드 로빈 액세스를 나타낼 수 있습니다. Sun StorEdge QFS 파일 시스템을 사용하여 각 파일 시스템 내에서 스트라이프 그룹을 나타낼 수 있습니다. 라운드 로빈 항목도 참조하십시오.
Sun StorEdge QFS	개별 장치에 저장하여 파일 데이터로부터 파일 시스템 메타데이터를 분리하는 고속 UNIX 파일 시스템. Sun StorEdge QFS 소프트웨어는 저장된 모든 파일 및 마스터 구성 파일(mcf)에서 구성된 모든 장치에 대한 액세스를 제어합니다.
Sun StorEdge SAM-FS	Sun 저장 및 아카이브 관리자 파일 시스템(Storage and Archive Manager File System). Sun StorEdge SAM-FS 소프트웨어는 저장된 모든 파일 및 마스터 구성 파일(mcf)에서 구성된 모든 장치에 대한 액세스를 제어합니다.
Sun SAM-QFS	Sun SAM-QFS 소프트웨어는 Sun StorEdge SAM-FS 소프트웨어를 Sun StorEdge QFS 파일 시스템과 결합시킵니다. Sun SAM-QFS는 저장 및 아카이브 관리 유틸리티와 함께 사용자 및 관리자에게 고속의 표준 UNIX 파일 시스템 인터페이스를 제공합니다. 표준 UNIX 파일 시스템 명령 뿐만 아니라 Sun StorEdge SAM-FS 명령에서 사용할 수 있는 많은 명령을 사용합니다.
Sun SAM-Remote 클라이언트	Sun SAM-Remote 클라이언트는 많은 의사 장치를 포함하는 Sun SAM-Remote 클라이언트 데몬을 설정하는 Sun StorEdge SAM-FS 시스템입니다. 자신의 라이브러리 장치를 포함하거나 포함하지 않을 수 있습니다. 클라이언트는 하나 이상의 아카이브 복사본을 위해 아카이브 매체의 Sun SAM-Remote 서버에 의존합니다.

Sun SAM-Remote 서버 Sun SAM-Remote 서버는 전체 용량 Sun StorEdge SAM-FS 저장 관리 서버 및 Sun SAM-Remote 클라이언트 사이에서 공유되는 라이브러리를 정의하는 Sun SAM-Remote 서버 데몬 모두입니다.

수퍼 블록 파일 시스템의 기본적인 매개 변수를 정의하는 파일 시스템의 데이터 구조. 저장소 패밀리 세트의 모든 파티션에 쓰여지고 이러한 세트에서 파티션의 구성원을 식별합니다.

T

tar 테이프 아카이브(tape archive). 아카이브 이미지에 대해 Sun StorEdge SAM-FS 소프트웨어가 사용하는 표준 파일 및 데이터 기록 형식.

TCP/IP 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.

임계값 온라인 저장소에서 사용 가능한 적정값을 정의하기 위한 메커니즘. 임계값은 릴리서에 대한 저장 목표값을 설정합니다. 디스크 공간 임계값도 참조하십시오.

타미머 사용자가 소프트웨어 제한에 도달하는 시간과 사용자에게 부여된 하드 제한 사이에 경과된 시간을 추적하는 할당량 소프트웨어.

V

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.

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.

W

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.

색인

A

-access 아카이버 명령, 78
ACSAPI 인터페이스, 4
ADIC/Grau 자동화 라이브러리
작업, 37
age_priority preview.cmd 명령, 156
allsets 아카이브 세트, 53, 87
API 루틴, 11
archive(1) 명령, 5
archive_audit(1M) command, 9
archiver(1M) 명령, 9, 63
archiver.cmd 파일, 52, 60, 63, 106, 124, 172
archiver.sh(1M) 명령, 9
archiver.sh(1M) 스크립트, 73
archmax 아카이버 명령, 68
auditslot(1M) 명령, 8, 21

B

bufsize 스테이지 명령, 150
-bufsize 아카이버 명령, 88
bufsize 아카이버 명령, 69
build_cat(1M) 명령, 8

C

chmed(1M) 명령, 8, 26, 170

cleandrive(1M) 명령, 8, 26
crontab 항목, 174

D

defaults.conf 파일, 187
dev_down.sh(1M) 명령, 10
Device
links, 9
Sun StorEdge SAM-FS 작성, 9
로그, 로그 파일 참조
상태, 17, 18
상태 설정, 9
devlog
file, 186
Directives
아카이버
파일 시스템 데이터 복사본 지정, 86
-disk_archive 아카이버 명령, 107
diskvols.conf 파일, 106
display_all_candidates 릴리서 명령, 139
DLT 테이프 드라이브, 182
dmpshm(1M) 명령, 9
-drivemin 아카이버 명령, 89
-drives 아카이버 세트 매개 변수 명령, 88
du(1) 명령, 5
dump_cat(1M) 명령, 8
DZC-8000S 인터페이스, 4

E

endparams 아카이버 명령, 87
endvsnpools 아카이버 명령, 103
endvsns 아카이버 명령, 101
exarchive(1M) 명령, 9
exported_media 명령, 30

F

find(1) 명령, 또한 sfind(1) 명령 참조, 6
fs 릴리서 명령, 138
fs 아카이버 명령, 76
Fujitsu LMF 자동화 라이브러리
작업, 39

G

-group 아카이버 명령, 79
GUI 도구, SAM-QFS Manager 참조

H

hlwm_priority 미리 보기 명령, 157
hwm_priority 미리 보기 명령, 157

I

IBM 3494 자동화 라이브러리
개요, 42
IBM 3584 자동화 라이브러리
가져오기, 41
개요, 41
클리닝, 41
파티션, 41
IBM 자동화 라이브러리, 4
ignore 리사이클러 명령, 175
import(1M), 38, 48
import(1M) 명령, 8, 24, 29, 31, 39, 44, 46
itemize(1M) 명령, 9

J

-join path 아카이버 명령, 93

L

lhwm_priority 미리 보기 명령, 157
libsam, 11
libsamrpc, 11
list_size releaser directive, 142
lmcpd 인터페이스, 4
load_notify.sh(1M) ??, 10
-lock 아카이버 명령, 91
log_rotate.sh(1M) 명령, 10
logfile
리사이클러 명령, 165
ls(1) 명령, 또한 sls(1) 명령 참조
lwm_priority 미리 보기 명령, 157

M

makedev(1M) 명령, samdev(1M) 명령 참조
maxactive 스테이지 명령, 153
-maxsize 아카이버 명령, 79
mcf 파일
라이브러리 기록자, 30
용도, 4
min_residence_age 릴리서 명령, 139
-minsize 아카이버 명령, 79
mount(1M) 명령, 7
move(1M) 명령, 8, 44, 45

N

-name 아카이버 명령, 80
no_archive 아카이브 세트, 53, 78
no_recycle 리사이클러 명령, 166
no_release 릴리서 명령, 139
-norelease 아카이버 명령, 85

O

odlabel(1M) 명령, 8, 21
ovflmin 아카이버 명령, 73

P

params 아카이버 명령, 87
-pool 아카이버 명령, 101
preview.cmd 파일, 미리 보기 요청 참조, 155, 158
-priority 아카이버 명령, 99

Q

qfsdump(1M) 명령, 7
qfsrestore(1M) 명령, 7

R

rearch(1M) 명령, 9
rearch_no_release 릴리서 명령, 141
recover.sh(1M) 명령, 10
-recycle_dataquantity 아카이버 명령, 172
-recycle_hwm 아카이버 명령, 172
-recycle_ignore 아카이버 명령, 172, 175
-recycle_mailaddr 아카이버 명령, 172
-recycle_mingain 아카이버 명령, 172
-recycle_vsncount 아카이버 명령, 172
recycler.cmd 파일, 169
recycler.sh 파일, 175
-release 아카이버 명령, 82, 84
release(1) 명령, 5, 129
releaser.cmd 파일, 135, 143
request(1) 명령, 5, 168, 188
-reserve 아카이버 명령, 95
reserve(1M) 명령, 9
restore.sh(1M) 명령, 10

S

sam_segment(3), 190
sam-archiverd 데몬, 60
sam-arcopy 데몬, 60
sam-arfind 데몬, 60
sambcheck(1M) 명령, 7
samchaid(1M) 명령, 7
samcmd(1M), 16
samcmd(1M) 명령, 6
samd(1M) 명령, 6, 16
samdev(1M) 명령, 8, 9
samexplorer(1M) 명령, 6
samexport(1M) 명령, 8, 29, 32, 38, 40, 45, 47, 48
samfsck(1M) 명령, 7
samfsconfig(1M) 명령, 7
sam-fsd 데몬, 60
samfsdump(1M) 명령, 7
samfsinfo(1M) 명령, 7
samfsrestore(1M) 명령, 7
sam-genericd 데몬, 4
samgrowfs(1M) 명령, 7
sam-ibm3494d 데몬, 4
sam-initd 데몬, 4
samload(1M) 명령, 8
sammkfs(1M) 명령, 7
samncheck(1M) 명령, 7
SAM-QFS Manager, 11
samquota(1M) 명령, 7
samquotastat(1M) 명령, 7
sam-recycler(1M) 명령, 9, 173
sam-releaser(1M) 명령, 9, 129
sam-robotsd 데몬, 4
sam-rpcd 데몬, 11
samset(1M) ??, 9
samset(1M) 명령, 6, 187
samsharefs(1M) 명령, 8
sam-sonyd 데몬, 4
sam-stkd 데몬, 4
samtrace(1M) 명령, 8

samu(1M) 호출 명령, 6
 samu(1M):호출 명령, 11
 samu(1M)를 사용하여 매체 언로드, 19
 samunhold(1M) 명령, 8
 sdu(1) 명령, 5
 SEF, 191
 sefdata 파일, 192
 sefreport(1M) ??, 191
 segment(1) 명령, 5, 190
 set_admin(1M) 명령, 9
 set_state(1M) ??, 9
 setfa(1) 명령, 6
 sfind(1) 명령, 6
 showqueue(1M) command, 9
 sls(1) 명령, 6
 Sony PetaSite 자동화 라이브러리
 개요, 43
 작업, 43
 Sony 네트워크로 연결된 자동화 라이브러리
 작업, 46
 Sony 자동화 라이브러리, 4
 -sort 아카이버 명령, 93
 squota(1) 명령, 6
 ssum(1) 명령, 6
 -stage 아카이버 명령, 82
 stage(1) 명령, 6
 stageback.sh(1M) 명령, 10
 star(1M) 명령, 10
 -startage 아카이버 명령, 100
 -startcount 아카이버 명령, 100
 -startsize 아카이버 명령, 100
 StorageTek ACSLS 연결 자동화 라이브러리
 작업, 47
 StorageTek 자동화 라이브러리, 4
 Sun, 11
 Sun SAM-QFS
 Sun StorEdge SAM-FS 참조
 Sun StorEdge SAM-FS 시작, 17
 Sun StorEdge SAM-FS 중지, 16
 Sun StorEdge SAM-FS, 중지, 16

T

-tapenonstop 아카이버 명령, 95
 tar(1) 명령, star(1M) 참조
 tarback.sh(1M) 명령, 10
 tplabel(1M) 명령, 8, 20
 trace_rotate(1M) 명령, 8

U

unarchive(1M) 명령, 10
 undamage(1M) 명령, 10
 unload(1M) 명령, 8
 unrearch(1M) 명령, 9
 unreserve(1M) 명령, 9
 -user 아카이버 명령, 79

V

VSN
 VSN 당 최소 증가, 170
 디스크 아카이브 명령, 105
 연결 명령, 101
 예약, 95
 일반 표현식 사용, 103
 풀 명령, 103
 vsn_priority preview.cmd 명령, 156
 vsnpools 아카이버 명령, 103
 vsns 아카이버 명령, 101

W

weight_age 릴리서 명령, 137
 weight_age_access 릴리서 명령, 137
 weight_age_modify 릴리서 명령, 137
 weight_age_residence 릴리서 명령, 137
 weight_size 릴리서 명령, 138
 wm_priority 요소, 156

ㄱ

- 간격 아카이버 명령, 71
- 감사
 - 블록, 21
 - 자동화 라이브러리, 22
- 광매체 카탈로그, 9
- 기본값
 - 시스템 기본값 설정, 9
 - 아카이버, 52

ㄴ

- 대기 아카이버 명령, 75
- 데몬
 - sam-archiverd, 60
 - sam-arcopy, 60
 - sam-arfind, 60
 - sam-fsd, 60
 - sam-genericd, 4
 - sam-ibm3494d, 4
 - sam-initd, 4
 - sam-robotsd, 4
 - sam-rpcd, 11
 - sam-sonyd, 4
 - sam-stkd, 4
 - 자동화 라이브러리 데몬, 4
- 데이터 검색
 - 또한 재해 복구 참조, 75
- 드라이브 스테이지 명령, 149
- 드라이브 아카이버 명령, 70
- 드라이브, 클리닝, 23, 25
- 디스크 아카이브, 105, 165
- 디스크 캐시
 - 릴리스 우선 순위, 2
 - 임계값, 2
 - 최고 워터마크, 130
 - 최저 워터마크, 130

ㄷ

- 라이브러리 기록자, 30
- 라이브러리 카탈로그
 - 개요, 29

보기, 35

- 라이브러리, 자동화 라이브러리 참조
- 라이센스
 - 일반 정보, xviii
 - 자동화 라이브러리에 슬롯 추가, 178
- 로그 파일
 - releaser 명령, 139
 - stager 명령, 151
 - 리사이클러, 173
 - 릴리서, 139
 - 스테이지, 151
 - 아카이버, 61, 72, 125
 - 아카이버 명령, 72
 - 장치 로그, 185
- 로봇, 자동화 라이브러리 참조
- 리사이클러
 - chmed(1M) 명령 강제, 170
 - crontab 항목, 174
 - ignore 옵션, 171
 - logfile 명령, 165
 - mail 옵션, 171
 - no_recycle 명령, 166
 - recycler.cmd 예제, 169
 - recycler.sh 파일, 175
 - 개요, 3, 163
 - 구성, 168
 - 디스크 아카이브 복사본, 165
 - 명령, 165
 - 문제 해결, 175
 - 작동 원리, 165
 - 정의, 2
- 릴리서
 - age, 131
 - candidate, 131
 - fs 명령, 138
 - priority, 132
 - 가중치, 132
 - 개요, 2, 129, 130
 - 구성, 143
 - 로그 파일, 139
 - 명령, 135
 - 명령 파일, 135
 - 문제 해결, 127, 144
 - 부분 릴리스, 132

수동 동작, 144
정의, 2

□

마스터 구성 파일.

mcf 파일 참조

만료된 데이터, 정의됨, 163

매체

교환기, 자동화 라이브러리 참조

라이브러리, 자동화 라이브러리 참조

로딩, 18

오류, 26

유형, 14

이동, 29

매체 가져오기

개요, 29

매체 내보내기

개요, 29

매체 로딩

samu(1M)으로, 18

수동으로 로드된 드라이브, 34

명령

archive(1), 5

archive_audit(1M), 9

archiver(1M), 9, 63

archiver.sh(1M), 9

auditslot(1M), 8, 21

build_cat(1M), 8

chmed(1M), 8, 26, 170

cleandrive(1M), 8, 26

dev_down.sh(1M), 10

dmpshm(1M), 9

du(1), 5

dump_cat(1M), 8

exarchive(1M), 9

find(1), 또한 sfind(1) 명령 참조, 6

GUI, SAM-QFS Manager 참조

import(1M), 8, 24, 29, 31, 38, 39, 44, 46, 48

itemize(1M), 9

load_notify.sh(1M), 10

log_rotate.sh(1M), 10

ls(1), 또한 sls(1) 명령 참조, 6

makedev(1M), samdev(1M) 참조

mount(1M), 7

move(1M), 8, 44, 45

odlabel(1M), 8, 21

qfsdump(1M), 7

qfsrestore(1M), 7

rearch(1M), 9

recover.sh(1M), 10

release(1), 5, 129

request(1), 5, 168, 188

reserve(1M), 9

restore.sh(1M), 10

sambcheck(1M), 7

samchaid(1M), 7

samcmd(1M), 6, 16

samd(1M), 6, 16

samdev(1M), 8, 9

samexplorer(1M), 6

samexport(1M), 8, 29, 32, 38, 40, 45, 47, 48

samfsck(1M), 7

samfsconfig(1M), 7

samfsdump(1M), 7

samfsinfo(1M), 7

samfsrestore(1M), 7

samgrowfs(1M), 7

samload(1M), 8

sammkfs(1M), 7

samncheck(1M), 7

samquota(1M), 7

samquotastat(1M), 7

sam-recycler(1M), 9, 173

sam-releaser(1M), 9, 129

samset(1M), 6, 9, 187

samsharefs(1M), 8

samtrace(1M), 8

samu(1M), 6, 11

samunhold(1M), 8

sdu(1), 5

sefreport(1M), 191

segment(1), 5, 190

set_admin(1M), 9

set_state(1M), 9

setfa(1), 6

sfind(1), 6

showqueue(1M), 9

sls(1), 6

squota(1), 6

ssum(1), 6

stage(1), 6

stageback.sh(1M), 10

star(1M), 10

- tar(1), star(1M) 명령 참조
- tarback.sh(1M), 10
- tplabel(1M), 8, 20
- trace_rotate(1M), 8
- unarchive(1M), 10
- undamage(1M), 10
- unload(1M), 8
- unresearch(1M), 9
- unreserve(1M), 9
- user, 5
- 리사이클러
 - ignore 명령, 175
 - logfile, 165
 - no_recycle, 166
- 릴리서
 - display_all_candidates, 139
 - fs, 138
 - list_size, 142
 - min_residence_age, 139
 - no_release, 139
 - research_no_release, 141
 - weight_age, 137
 - weight_age_access, 137
 - weight_age_modify, 137
 - weight_age_residence, 137
 - weight_size, 138
 - 로그 파일, 139
 - 릴리스 우선 순위 명령, 136
- 미리보기
 - age_priority, 156
 - hlwm_priority, 157
 - hwm_priority, 157
 - lhwm_priority, 157
 - lwm_priority, 157
 - vsnpriority, 156
- 스테이지
 - bufsize, 150
 - drives, 149
 - maxactive, 153
 - 로그 파일, 151
- 아카이버
 - access, 78
 - archmax, 68
 - bufsize, 88
 - bufsize, 69
 - disk archive, 105, 107
 - drivemin, 89
 - drives, 88
 - endparams, 87
 - endvsnpools, 103
 - endvsns, 101
 - fs, 76
 - group, 79
 - join path, 93
 - lock, 91
 - maxsize, 79
 - minsize, 79
 - name, 80
 - norelease, 85
 - ovflmin, 73
 - Params, 87
 - pool, 101
 - priority, 99
 - recycle_dataquantity 명령, 172
 - recycle_hwm ??, 172
 - recycle_ignore 명령, 172, 175
 - recycle_mailaddr ??, 172
 - recycle_mingain 명령, 172
 - recycle_vsncount 명령, 172
 - release, 82, 84
 - reserve, 95
 - sort, 93
 - stage, 82
 - startage, 100
 - startcount, 100
 - startsize, 100
 - tapenonstop, 95
 - user, 79
 - vsnpools, 103
 - vsns, 101
 - wait, 75
 - 간격, 71
 - 개요, 64, 67
 - 관찰, 70
 - 드라이브, 70
 - 로그 파일, 72
 - 명령 리사이클, 92
 - 아카이브 나이 설정, 85
 - 아카이브 세트 구성원, 76
 - 아카이브 세트 복사본 번호, 84
 - 자동 아카이브 해제 설정, 86
 - 통지, 73
 - 일반 시스템 관리자, 6
 - 자동화 라이브러리, 8
 - 파일 시스템, 7
- 문제 해결
 - 리사이클러, 175

릴리서, 144
아카이버, 125

미리보기 요청

age_priority 명령, 156
hlwm_priority 명령, 157
hwm_priority 명령, 157
lhwm_priority 명령, 157
lwm_priority 명령, 157
vsn_priority 명령, 156
계획, 158
우선 순위 계산, 158
우선 순위 지정, 154
초기화, 158

ㅂ

볼륨 시리얼 이름, VSN 참조

볼륨 오버플로 (ovflmin 아카이버 명령), 73

부분 릴리스 및 부분 스테이지 기능, 132

ㅅ

사용자 명령, 5

사용하지 않는 공간, 정의됨, 163

세그먼트된 파일, 190

소프트웨어 업그레이드, 177

소형 컴퓨터 시스템 인터페이스

손상된 파일 제거, 10

스테이지

bufsize 명령, 150
drives 명령, 149
maxactive 명령, 153
개요, 3, 147
로그 작업, 151
로그 파일 필드, 152
명령, 147
부분 스테이지, 132
스테이지 요청 처리 오류, 3
스테이지의 아카이브 역할, 154
정의, 2, 147

스테이지 요청 처리 오류, 3

슬롯 추가, 178

슬롯, 추가, 178

시스템 오류 기능, SEF 참조

ㅇ

아카이버

archive sets, 52

Archive sets:, 52

archiver.cmd 예제, 65

archiver.cmd에 파일 시스템 지정, 76

VSN pools, 103

VSN 연결 명령, 101

개요, 2

기본값, 52

데몬, 60

디스크 아카이브, 디스크 아카이브 참조

로그 파일, 61, 125

명령, 9

문제 해결, 125

복사 정의 명령, 84

복사본 작성, 2

스테이지에서의 역할, 154

아카이버, 또한 명령 참조, 64

아카이버, 또한 명령 참조, 67

아카이브 간격 정의, 52

아카이브 나이 결정, 94

아카이브 나이 설정, 85

아카이브 나이 정의, 52

아카이브 막기, 78

아카이브 매체 정의, 1, 51

아카이브 세트 구성원 명령, 76

아카이브 세트 구성원 충돌, 83

아카이브 세트 프로세싱 명령, 87

아카이브 해제, 94

연관 아카이브, 93

연속 아카이브, 70

예약된 VSN, 96

예제, 112

우선 순위 설정, 99

일반 표현식 사용, 80

자동 아카이브 해제 설정, 86

작동 원리, 51

정의, 1, 51

지침, 124

- 파일 시스템 데이터 복사본 지정, 86
- 파일 크기 제어, 68
- 항목 삭제, 10
- 아카이버 명령 관찰, 70
- 애플리케이션 프로그래머 인터페이스, API 참조
- 언아카이브, 86, 94
- 연관 아카이브, 93
- 연속 아카이브, 70
- 용량, 정의됨, 164
- 위터마크 명령, 156
- 일반 시스템 관리자 명령, 6
- 일반 표현, 80

ㄷ

- 자동화 라이브러리
 - IBM, 4
 - SCSI 연결, 자동화 라이브러리 참조, 직접 연결
 - Sony, 4
 - StorageTek, 4
 - 감사, 22
 - 기록자
 - 라이브러리 기록자 참조, 30
 - 끄기, 18
 - 네트워크 연결, 3, 29
 - 데몬, 4
 - 또한 개별 업체용 항목 참조, 4
 - 명령, 8
 - 작업, 29
 - 정의, 13
 - 직접 연결, 3, 29
 - 켜기, 17
- 재아카이브, 정의됨, 165
- 제거 가능한 매체 파일, 188
- 주크박스, 자동화 라이브러리 참조
- 직접 연결, 30
- 직접 연결 자동화 라이브러리, 자동화 라이브러리 참조

ㄸ

- 체크섬 속성, 6

- 최고 위터 마크
 - 리사이클러 사용, 170
 - 미리보기 명령, 156
- 최저 위터마크 명령, 156

ㅋ

- 카트리지
 - 가져오기, 31
 - 내보내기, 31
 - 로딩, 8, 18, 34
 - 언로딩, 19, 34
 - 업로딩, 8
 - 제거, 27
 - 클리닝, 23
- 클리닝 테이프 드라이브, 25

ㆁ

- 통지 아카이버 명령, 73

ㆁ

- 파일 시스템
 - data, 86
 - 개요, 1
 - 명령, 7
- 파일 요청, 제거 가능 매체 파일 참조
- 파일, 속성 설정, 82

ㆁ

- 하드웨어 및 소프트웨어 업그레이드, 177
- 하드웨어 업그레이드, 177
- 현재 데이터, 정의됨, 163

