

# Software Distributor 관리 설명서

HP-UX 11i v1, HP-UX 11i v2



i n v e n t

제품 제조 번호: B2355-90985

2006년 6월

© Copyright 1997, 2000-2003, 2006 Hewlett-Packard Development Company L.P.

---

## 알림

© Copyright 1997, 2000-2003, 2006 Hewlett-Packard Development Company, L.P.

기밀 컴퓨터 소프트웨어입니다. 소유, 사용 또는 복사를 위해서는 **HP**로부터 유효한 라이선스를 취득해야 합니다. **FAR 12.211** 및 **12.212**에 준거하여 상용 컴퓨터 소프트웨어, 컴퓨터 소프트웨어 문서 및 상용 항목의 기술 데이터에 대한 라이선스가 공급업체의 표준 상용 라이선스에 따라 미합중국 정부에 부여됩니다.

이 설명서의 내용은 예고 없이 변경될 수 있습니다. **HP** 제품과 서비스에 대한 보증은 오직 제품 및 서비스와 함께 제공되는 명시적 보증서만을 근거로 합니다. 이 설명서의 어떤 내용도 추가 보증 제정으로 해석할 수 없습니다. **HP**는 이 문서에 포함된 기술적 오류나 편집상의 오류에 대해 책임을 지지 않습니다.

**Intel® Itanium®** 로고, **Intel**, **Intel Inside** 및 **Itanium**은 미국과 다른 국가에서 **Intel Corporation** 또는 자회사의 상표 또는 등록 상표입니다.

**Microsoft®**와 **Windows®**는 미국에서 **Microsoft Corporation**의 등록 상표입니다.

**Oracle®**은 캘리포니아 레드우드 시에 있는 **Oracle Corporation**의 미국 등록 상표입니다.

**UNIX®**는 **Open Group**의 등록 상표입니다.

**1. Software Distributor 소개**

SD-UX 개요 ..... 24

    네트워크 요구 사항 ..... 24

    SD-UX 프로그램 및 명령 ..... 25

    SD-UX 온라인 설명서 ..... 28

SD-UX 개념 ..... 29

    중요한 용어 ..... 29

    소프트웨어 구조 ..... 30

    IPD(Installed Products Database) ..... 32

    제어 스크립트 ..... 32

    환경 변수 ..... 34

    소프트웨어 종속성 ..... 35

    보호되는 소프트웨어에 대한 작업 ..... 36

GUI 및 TUI 명령 사용 ..... 37

    터미널 사용자 인터페이스 ..... 37

    GUI/TUI 명령 시작 ..... 38

    Window 구성 요소 ..... 39

    객체 목록의 시작 항목 및 마감 항목 ..... 40

    객체 목록에 항목 표시 ..... 40

    호스트 파일 미리 선택 ..... 41

    Software Selection 창 ..... 42

    세션 및 파일 관리—File 메뉴 ..... 43

    소프트웨어 보기 변경—View 메뉴 ..... 44

    옵션 변경 및 객체 목록 새로 고침—Options 메뉴 ..... 48

    작업 수행—Actions 메뉴 ..... 50

    도움말 표시—Help 메뉴 ..... 54

    XToolkit 옵션 및 디스플레이 글꼴 변경 ..... 55

명령줄에서 작업 ..... 57

    소프트웨어 선택 사항 ..... 58

    대상 선택 사항 ..... 60

    명령 옵션 사용 ..... 61

    세션 파일 ..... 63

**2. 소프트웨어 설치**

swinstall을 사용한 설치 ..... 66

    기능 및 제한 사항 ..... 66

---

# 목차

GUI를 사용한 설치.....	67
명령줄에서 설치.....	75
설치 작업 및 예.....	78
설치 구성(swconfig).....	84
기능 및 제한 사항.....	84
구성 프로세스.....	85
swconfig 사용.....	88
구성 작업 및 예.....	90
설치 확인(swverify).....	91
기능 및 제한 사항.....	91
확인 프로세스.....	92
swverify 사용.....	94
확인 작업 및 예.....	96

## 3. 설치된 소프트웨어 관리

소프트웨어 나열(swlist).....	98
swlist 기능 및 제한 사항.....	98
swlist GUI 사용.....	98
명령줄 사용.....	100
소프트웨어 나열 작업 및 예.....	105
IPD 수정(swmodify).....	117
IPD 내용.....	117
swmodify 사용.....	119
swmodify 작업 및 예.....	122
설치된 소프트웨어 제거(swremove).....	124
swremove 기능 및 제한 사항.....	124
swremove GUI 사용.....	125
명령줄을 사용한 제거.....	130
제거 작업 및 예.....	133

## 4. 소프트웨어 저장소 관리

저장소 관리 명령 및 개념.....	136
저장소 개념.....	137
소프트웨어 저장소 복사.....	139
swcopy 기능 및 제한 사항.....	139
swcopy GUI 사용.....	140

swcopy 명령줄 사용 .....	149
복사 작업 및 예 .....	152
저장소 등록 및 등록 해제(swreg) .....	153
미디어 등록 또는 네트워크 저장소 작성? .....	153
등록 및 보안 .....	154
권한 부여 .....	154
swreg 사용 .....	154
swreg 예 .....	156
추가적인 저장소 관리 작업 및 예 .....	157
패치 저장소 병합 .....	157
배포용 테이프 저장소 만들기 .....	158
저장소 속성 설정 .....	158
네트워크 저장소 만들기 .....	159
HP-UX 여러 버전 관리 .....	160
등록된 저장소 나열 .....	160
저장소 내용 나열(swlist -d) .....	161
소스 저장소 감사 .....	162
저장소 확인(swverify -d) .....	163
저장소에서 소프트웨어 제거 .....	164
저장소 제거 .....	164

## 5. HP-UX 패치 및 패치 관리

## 6. 원격 작업 개요

소개 .....	168
원격 작업과 로컬 작업의 차이점 .....	168
원격 작업 GUI 사용 .....	171
Target Selection 창 .....	172
작업 수행 .....	172
여러 대상 선택 .....	173
개별 대상 선택 .....	175
대상 그룹 저장 .....	176
대상 그룹 추가 .....	176
원격 작업 설정 .....	177
원격 작업 자습서 .....	178
자습서 설치 .....	178

---

# 목차

단일 대상 설치 수행 방법 .....	179
원격 대화식 <b>swlist</b> .....	188
명령줄에서의 원격 작업 .....	189
대상 선택 사항 .....	189
예 .....	190

## 7. 작업 및 작업 브라우저 사용

소개 .....	194
작업 브라우저 시작하기 .....	194
작업 브라우저 사용하기 .....	195
작업 브라우저 아이콘 .....	196
<b>File</b> 메뉴 .....	199
<b>View</b> 메뉴 .....	199
<b>Options</b> 메뉴 .....	201
<b>Actions</b> 메뉴 .....	202
명령줄에서 작업 모니터링 .....	207
<b>swjob</b> 속성 .....	209
<b>swjob</b> 작업 및 예제 .....	210
명령 옵션을 사용하여 작업 관리 및 조정 .....	211
명령줄에서 작업 예약 .....	211
작업 제목 추가 .....	211
작업 정보 제거 .....	212

## 8. 안정성 및 성능

개요 .....	214
그룹 및 소스 옵션 .....	216
많은 수의 대상 .....	217
시간 제한 옵션 .....	218
재시도 <b>RPC</b> 및 재시도 간격 .....	219
재시도 명령 .....	220
데이터베이스 검사점 지정 .....	221
압축 .....	222
<b>INDEX</b> 및 <b>INFO</b> 압축 .....	222
단계 .....	224
복구(설치에 한함) .....	227
별도의 구성으로 설치 .....	229

여러 버전 .....	230
<b>9. SD-UX 보안</b>	
개요 .....	234
기본 보안 .....	234
저장소 및 저장소 등록 .....	235
대상 시스템 변경 .....	235
swacl 명령 .....	236
swacl 결과 .....	238
기본 보안 작업 .....	239
사용자 액세스 나열 .....	240
사용자에게 제품 관리 권한 부여 .....	243
사용자에게 관리 권한 부여(설치/제거) .....	243
저장소에 대한 액세스 제한 .....	244
대상 호스트 추가 .....	246
일시적인 액세스 제한 .....	246
SD-UX 네트워크 닫기 .....	246
ACL 편집 .....	247
ACL이 사용자에게 일치되는 방법 .....	250
ACL 항목 .....	251
ACL 키 .....	252
ACL 권한 .....	253
객체 보호 .....	254
ACL 템플릿 .....	260
SD-UX 시스템에서의 보안 .....	263
SD-UX 내부 권한 인증 .....	265
SD-UX 자격 증명 .....	265
호스트 간 보안: 공유 비밀 파일 .....	267
RPC 권한 .....	269
에이전트의 컨트롤러 요청 처리 방법 .....	269
로컬 슈퍼유저 권한 인증 .....	271
저장소 등록 및 데몬/에이전트 보안 .....	271
보안 사용 모델 .....	273
원격 배포 시 보안 .....	273
로컬 배포 시 보안 .....	274
소프트웨어 개발자를 위한 보안 .....	275

---

# 목차

명령별 권한 요구 사항 .....	276
패키징( <b>swpackage</b> ) .....	276
나열( <b>swlist</b> ) .....	276
작업 찾아보기( <b>sd, swjob</b> ) .....	276
복사( <b>swcopy</b> ) .....	276
설치( <b>swinstall</b> ) .....	277
제거( <b>swremove</b> ) .....	277
구성( <b>swconfig</b> ) .....	278
확인( <b>swverify</b> ) .....	278
저장소 등록( <b>swreg</b> ) .....	278
ACL 변경( <b>swacl</b> ) .....	278
Request 스크립트( <b>swask</b> ) .....	278
변경( <b>swmodify</b> ) .....	278

## 10. 소프트웨어 패키지 만들기

패키징 프로세스 개요 .....	280
전제 조건 .....	280
패키징할 제품 식별 .....	281
제품 내용 결정 .....	281
제품 구조 결정 .....	281
제어 스크립트 추가 .....	283
PSF(Product Specification File) 만들기 .....	285
PSF(Product Specification File)의 예 .....	286
PSF 구문 .....	289
소프트웨어 패키징( <b>swpackage</b> ) .....	324
<b>swpackage</b> 사용 .....	330
패키징 작업 및 예 .....	335
<b>swpackage</b> 로 만든 저장소 등록 .....	335
CD-ROM 저장소 작성 및 마스터 .....	336
성능 향상을 위한 파일 압축 .....	337
패키징 보안 .....	337
소프트웨어 패키지의 변경 또는 다시 패키징 .....	340
현재 위치에서 패키징 .....	341
소스의 심볼릭 링크 따라가기 .....	342
파일 버전 생성 .....	342
원격 파일 시스템의 저장소 .....	343
소프트웨어 패키지 확인 .....	344



패치 소프트웨어 패키징 ..... 344  
 여러 타입에 쓰기 ..... 345  
 기존 저장소에서 타입 만들기 ..... 346

**11. 제어 스크립트 사용**

제어 스크립트에 대한 소개 ..... 348  
 제어 스크립트의 종류 ..... 349  
 일반적인 스크립트 작성 지침 ..... 356  
 제어 스크립트 패키징하기 ..... 357  
 실행 시 파일 시스템에서의 제어 스크립트 위치 ..... 358  
 환경 변수 사용 ..... 359  
 모든 SD-UX 명령에 영향을 주는 변수 ..... 359  
 모든 SD-UX 스크립트에 영향을 주는 변수 ..... 360  
 swinstall과 swremove에 영향을 주는 변수 ..... 363  
 swverify에 영향을 주는 변수 ..... 364  
 제어 스크립트의 실행 ..... 365  
 모든 제어 스크립트에 적용되는 세부 사항 ..... 365  
 Checkinstall 스크립트 ..... 366  
 Preinstall 스크립트 ..... 367  
 Postinstall 스크립트 ..... 367  
 Configure 스크립트 ..... 368  
 Unconfigure 스크립트 ..... 369  
 Verify 스크립트 ..... 369  
 Fix 스크립트 ..... 370  
 Checkremove 스크립트 ..... 370  
 Preremove 스크립트 ..... 371  
 Postremove 스크립트 ..... 371  
 Request 스크립트 ..... 372  
 제어 스크립트에 의한 다른 명령 실행 ..... 374  
 제어 스크립트의 입력 및 출력 ..... 375  
 제어 스크립트에 의한 파일 관리 ..... 379  
 제어 스크립트 테스트 ..... 380  
 설치 스크립트 테스트 ..... 380  
 구성 스크립트 테스트 ..... 381  
 제거 스크립트 테스트 ..... 383  
 사용자 응답 요구(swask) ..... 385

---

# 목차

swask 사용 .....	385
request 스크립트 작업 및 예 .....	388
swask 예제 .....	388
swinstall 예제 .....	388
swconfig 예제 .....	389

## 12. 비권한 SD

개요 .....	392
혜택을 받는 대상은? .....	392
작동 방식? .....	392
제한 사항 .....	393
비권한 모드 설정 .....	394
비권한 모드에서 사용할 소프트웨어 패키징 .....	394
비권한 모드 켜기 .....	395
비권한 모드로 SD-UX 작동 방식을 바꾸는 방법 .....	395
기본 구성 .....	396
대체 구성 .....	397
관리 디렉토리 옵션 설정 .....	397

## A. 명령 옵션

명령 옵션 변경 .....	400
옵션(알파벳 순) .....	402

## B. 문제 해결

오류 기록 .....	440
오류 메시지 .....	441
경고 메시지 .....	441
참고 사항 .....	441
일반적인 문제점 .....	442
대상 호스트의 데몬이나 에이전트에 연결할 수 없는 경우 .....	443
GUI가 시작되지 않거나 지원 파일이 유실된 경우 .....	445
객체에 대한 액세스가 거부되는 경우 .....	446
네트워크 성능이 떨어지는 경우 .....	449
연결 시간 제한 및 기타 WAN 문제점 .....	450
디스크 공간 분석이 올바르지 않은 경우 .....	452
패키징 프로그램이 실패한 경우 .....	452

명령 로그 파일이 너무 커지는 경우 ..... 453  
데몬 로그 파일이 너무 긴 경우 ..... 453  
테이프 저장소를 읽을 수 없는 경우 ..... 454  
설치에 실패한 경우 ..... 454  
swinstall 또는 swremove가 실패하고 잠금 오류가 발생한 경우 ..... 455

**C. SD-UX 교체 또는 업데이트**

SD 재설치 ..... 458  
  전제 조건 ..... 458  
  install-sd 사용 ..... 459  
사용할 수 없는 SD-UX 버전 교체 ..... 460  
새로운 버전의 SD-UX 설치 ..... 461

**D. Software Distributor 파일 및**

**파일 시스템 구조**

에이전트 파일 시스템 구조 ..... 464  
Software Distributor 컨트롤러 파일 시스템 구조 ..... 467  
Installed Products Database ..... 468

용어집 ..... 469

색인 ..... 483

---

## 목차

표 1. HP-UX 11i 릴리즈 .....	18
표 1-1. 이 장의 항목 .....	23
표 1-2. SD 명령 .....	25
표 1-3. 연산자 종류 .....	46
표 2-1. 이 장의 항목 .....	65
표 2-2. GUI 설치 단계 .....	67
표 2-3. swinstall 명령 옵션 및 기본값 .....	77
표 2-4. 제품 호환성 .....	83
표 2-5. swconfig 명령 옵션 및 기본값 .....	89
표 2-6. swverify 명령 옵션 및 기본값 .....	95
표 3-1. 이 장의 항목 .....	97
표 3-2. -l 옵션 .....	102
표 3-3. swlist 명령 옵션 및 기본값 .....	104
표 3-4. -l 옵션 .....	109
표 3-5. 저장소 나열 .....	113
표 3-6. 샘플 속성 .....	114
표 3-7. swmodify 명령 옵션 및 기본값 .....	122
표 3-8. GUI 제거 단계 .....	125
표 3-9. swremove 명령 옵션 및 기본값 .....	133
표 4-1. 이 장의 항목 .....	135
표 4-2. 저장소 관리를 위한 명령 .....	136
표 4-3. 복사 프로세스 단계 .....	140
표 4-4. swcopy 명령 옵션 및 기본값 .....	151
표 4-5. swreg 명령 옵션 및 기본값 .....	156
표 6-1. 이 장의 항목 .....	167
표 6-2. 설치 단계 .....	179
표 6-3. 소프트웨어 선택 목록 .....	183
표 7-1. 이 장의 항목 .....	193
표 7-2. 작업 조치 옵션 .....	202
표 7-3. swjob 명령 옵션 및 기본값 .....	208
표 7-4. 일반적인 작업 속성 .....	209
표 8-1. 이 장의 항목 .....	213
표 9-1. 이 장의 항목 .....	233

---

## 표

표 9-2. swacl 명령 옵션 및 기본값	237
표 9-3. SD-UX ACL 항목 유형	251
표 9-4. SD-UX ACL 항목 키 값	252
표 9-5. ACL 권한	253
표 9-6. SD-UX ACL 권한 정의	254
표 9-7. 호스트 ACL 권한	256
표 9-8. root 권한	257
표 9-9. 저장소 권한	257
표 9-10. 제품 주체	258
표 9-11. 제품 권한	258
표 10-1. 이 장의 항목	279
표 10-2. PSF에 사용되는 키워드	290
표 10-3. 제어 파일 속성	293
표 10-4. swpackage 프로세스 단계	325
표 10-5. swpackage 명령 옵션 및 기본값	332
표 11-1. 이 장의 항목	347
표 11-2. 제어 스크립트 키워드	357
표 11-3. swask 명령 옵션 및 기본값	387
표 12-1. 이 장의 항목	391
표 A-1. 이 장의 항목	399
표 A-2. loglevel과 logdetail 조합	421
표 B-1. 이 장의 항목	439
표 B-2. 일반적인 문제점	442
표 C-1. 이 장의 항목	457
표 D-1. 이 장의 항목	463
표 D-2. 에이전트 구성 요소	464
표 D-3. 컨트롤러 파일 시스템 구조	467

그림 1-1. SD-UX 시스템	30
그림 1-2. HP-UX 소프트웨어 구조의 예	32
그림 1-3. 터미널 사용자 인터페이스(TUI)	37
그림 1-4. GUI 창 구성 요소	39
그림 1-5. Software Selection 창	42
그림 1-6. Column Editor	44
그림 1-7. Filter 대화 상자	45
그림 1-8. Sort 대화 상자	47
그림 1-9. Options 대화 상자	48
그림 1-10. Save Software Group 대화 상자	51
그림 1-11. Specify Source 대화 상자	52
그림 1-12. Select Target Path 대화 상자	53
그림 1-13. Shared Root Paths 대화 상자	53
그림 1-14. 일반적인 온라인 도움말 화면	54
그림 1-15. 명령 예제	57
그림 2-1. Specify Source 대화 상자	68
그림 2-2. swinstall Software Selection 창	69
그림 2-3. Analysis 대화 상자	71
그림 2-4. Disk Space Analysis 창	72
그림 2-5. 설치 창	74
그림 3-1. swlist 브라우저	99
그림 3-2. swremove Software Selection 창	126
그림 3-3. Remove Analysis 대화 상자	127
그림 3-4. Remove 창	129
그림 4-1. Select Target Depot Path 대화 상자	141
그림 4-2. Specify Source 대화 상자	142
그림 4-3. Software Selection 창	143
그림 4-4. Copy Analysis 대화 상자	145
그림 4-5. Disk Space Analysis 창	146
그림 4-6. Copy 창	148
그림 6-1. Target Selection 창	172
그림 6-2. Add Targets 대화 상자	173
그림 6-3. Target Selection 창	180

---

## 그림

그림 6-4. Add Target 대화 상자(여러 대상 및 기본값이 아닌 대상의 경우) .....	181
그림 6-5. Specify Source 대화 상자.....	182
그림 6-6. 소프트웨어 선택 창.....	183
그림 6-7. Install Preferences 대화 상자 .....	184
그림 6-8. Install Analysis 대화 상자.....	186
그림 6-9. Install Window 대화 상자 .....	186
그림 6-10. swlist 브라우저 .....	188
그림 7-1. SD Job Browser 창 .....	195
그림 7-2. 복사 아이콘 .....	196
그림 7-3. 활성화된 설치 작업 아이콘.....	196
그림 7-4. 예약된 설치 작업 아이콘 .....	197
그림 7-5. 경고가 발생한 설치 작업 아이콘.....	197
그림 7-6. 오류가 발생한 설치 작업 아이콘.....	197
그림 7-7. 설치된 소프트웨어의 예약된 제거 작업 아이콘 .....	198
그림 7-8. 저장소 소프트웨어의 예약된 제거 작업 아이콘 .....	198
그림 7-9. 확인 작업 아이콘 .....	198
그림 7-10. 등록 정보별로 표시된 작업.....	200
그림 7-11. Refresh Interval 대화 상자 .....	201
그림 7-12. Job Results 대화 상자 .....	203
그림 7-13. 작업 설명 표시 .....	204
그림 7-14. Remove a Job 대화 상자 .....	206
그림 9-1. 액세스 제어 목록 .....	239
그림 9-2. ACL 템플릿 .....	260
그림 9-3. SD-UX 보안 프로세스 .....	270
그림 10-1. 패키징 프로세스 개요.....	326



---

## 설명서 정보

이 설명서에서는 SD(Software Distributor)를 사용하여 HP-UX 운영 체제 및 계층형 소프트웨어 응용 프로그램을 제공, 관리 및 유지 보수하는 방법에 대해 설명합니다. 이 설명서는 HP-UX 서버와 워크스테이션을 설치 및 관리하는 경험이 풍부한 HP-UX 시스템 관리자를 대상으로 작성되었습니다.

## 대상 독자

이 설명서는 다음 유형의 사용자와 관리자를 대상으로 작성되었습니다.

- 일상적인 작업을 수행하기 위해 적절한 소프트웨어 관리 도구에 빠르고 쉽게 액세스하려는 독립형 HP-UX 사용자. 여기에는 소프트웨어 설치, 확인 또는 제거 및 기본적인 저장소 관리가 포함됩니다.
- 다른 사용자들의 설치 및 실행을 관리하는 HP-UX 시스템 관리자. 추가적인 소프트웨어 관리 작업에는 보다 복잡한 저장소 관리, 보안, 패치 관리, 원격 작업 및 성능 관리 등이 포함됩니다.
- 소프트웨어를 SD에서 사용할 수 있는 형식으로 패키지와 패키징된 소프트웨어 제공을 위한 스크립트 작성에 주로 관여하는 소프트웨어 패키지 작성자

## HP-UX 11i 릴리즈 이름 및 릴리즈 ID

각 HP-UX 11i 릴리즈에는 연관된 릴리즈 이름 및 릴리즈 ID가 있습니다. 다음 표에서는 HP-UX 11i에서 사용할 수 있는 릴리즈를 보여 줍니다.

표 1

HP-UX 11i 릴리즈

릴리즈 이름	릴리즈 ID	지원되는 프로세서 아키텍처
HP-UX 11i v1	B.11.11	PA-RISC
HP-UX 11i v1.5	B.11.20	Intel® Itanium®
HP-UX 11i v1.6	B.11.22	Intel® Itanium®
HP-UX 11i v2	B.11.23	Intel® Itanium® PA-RISC <sup>a</sup>

a. PA-RISC는 2004년 9월 릴리즈부터 HP-UX 11i v2에서 지원됩니다.

`uname (1)` 명령에 `-r` 옵션을 사용하면 릴리즈 ID가 반환됩니다.

다음을 입력하여 업데이트 릴리즈 날짜 및 운영 환경을 확인할 수 있습니다.

```
# swlist | grep HPUX11i
```

결과 출력에 현재 릴리즈 ID, 업데이트 릴리즈 날짜 및 운영 환경 목록이 나열됩니다.

예를 들면 다음과 같습니다.

```
HPUX11i-TCOE      B.11.230.0512 HP-UX Technical  
Computing Operating Environment Component
```

위의 버전 문자열은 다음을 나타냅니다.

B.11.23 = HP-UX 11i v2

0512 = 2005년 12월 업데이트 릴리즈

## 표기법

이 설명서에서는 다음과 같은 표기법을 사용합니다.

<i>audit</i> (5)	HP-UX 맨페이지입니다. <i>audit</i> 는 명령 이름이며 5는 <i>HP-UX Reference</i> 의 섹션입니다. 웹 및 Instant Information CD에서는 온라인 설명서 자체에 대한 링크일 수 있습니다. HP-UX 명령줄에서 “man audit” 또는 “man 5 audit”를 입력하면 맨페이지가 표시됩니다. <i>man</i> (1)을 참조하십시오.
<i>Book Title</i>	책 제목입니다. 웹 및 Instant Information CD에서는 책 자체에 대한 링크일 수 있습니다.
<i>Emphasis</i>	강조하는 텍스트입니다.
<b>Emphasis</b>	특별히 강조하는 텍스트입니다.
ComputerOut	컴퓨터 화면에 표시되는 텍스트입니다.
Command	명령 이름이거나 명령 구문입니다.
Computer	Computer font는 컴퓨터에 표시되는 내용을 그대로 나타냅니다. 예를 들면 file not found입니다.
Filename	파일 이름 및/또는 파일 경로를 나타내는 텍스트입니다.
<b>UserInput</b>	사용자가 입력하는 명령이나 텍스트입니다.
<i>Variable</i>	명령이나 함수에서 대체할 수 있는 변수 이름이거나 가능한 값이 여러 개인 정보를 표시할 때 사용합니다.
[ ]	형식 및 명령 설명에서 선택적인 내용입니다.
{ }	형식 및 내용 설명에서 필수적인 내용입니다. 내용이  로 구분된 목록인 경우 그 중 하나를 선택해야 합니다.
...	앞에 있는 요소를 원하는 만큼 반복할 수 있습니다.
	선택 목록에서 항목을 구분합니다.

## 발행 정보

발행 날짜와 제품 번호로 설명서의 버전을 확인할 수 있습니다. 새로운 버전의 설명서가 발행될 때 발행 날짜가 변경됩니다. 변경 사항이 많은 경우 설명서의 번호가 변경됩니다.

새 버전의 설명서를 받으려면 해당 제품 지원 서비스에 가입해야 합니다. 자세한 사항은 HP 영업 담당자에게 문의하십시오.

- **HP-UX 11i v1 및 v2용 Software Distributor 관리 설명서:**  
2006년 6월, B2355-90985
- **Software Distributor 관리 설명서:**  
2003년 9월, B2355-90790
- *Software Distributor Administration Guide for HP-UX 11i v1.6:*  
2002년 6월, B2355-90754
- *Managing HP-UX Software with SD-UX*  
HP-UX 11i 버전 1.5: 2001년 6월, B2355-90740
- *Managing HP-UX Software with SD-UX*  
HP-UX 11i: 2000년 12월, B2355-90699
- *Managing HP-UX Software with SD-UX*  
HP-UX 11.00: 1997년 11월, B2355-90154
- *Managing HP-UX Software with SD-UX*  
HP-UX 10.x: 1997년 4월, B2355-90107

새 발행본은 이전 발행본 이후에 업데이트된 모든 내용을 담고 있습니다. 최신 버전에 대한 정보는 다음 웹 사이트에서 HP-UX 11i Operating System Documentation 부분을 참조하십시오.

<http://docs.hp.com>(영문)

<http://docs.hp.com/ko>(한글)

및

<http://www.docs.hp.com/en/SD/>

## 관련 설명서

HP 기술 문서 웹 사이트(<http://docs.hp.com/>)에 있는 다음 문서들의 최신 버전은 특히 참조할 만합니다.

*Ignite-UX Reference*

*Successful System Cloning using Ignite-UX* 백서

*Configuring Bastille to Operate with Ignite-UX* 백서

*Ignite-UX Custom Configuration Files* 백서

*Successful System Recovery using Ignite-UX* 백서

*Troubleshooting Ignite-UX Installation Booting* 백서

**HP-UX를 설치하거나 업데이트하기 전에**

**HP-UX 설치 및 업데이트 설명서**

*HP-UX Reference*

시스템 및 작업 그룹 관리: **HP-UX 시스템 관리자를 위한 설명서**

**Software Distributor 관리 설명서**

*HP-UX Patch Management*

**HP 시스템 파티션 설명서**

*Getting Started with Software Package Builder*

*VERITAS File System 4.1 (HP OnlineJFS/JFS) and VERITAS Volume Manager 4.1 Installation Guide*

이들 문서 중 일부 또는 모두는 Instant Information 미디어 및 인쇄된 양식으로 볼 수 있습니다.

## 사용자 의견 접수

HP는 이 설명서에 대한 사용자 여러분의 의견을 기다리고 있습니다. HP는 사용자 요구를 충족하는 설명서를 만들기 위해 최선을 다할 것입니다.

여러분의 의견을 다음 주소로 보내주십시오.

**feedback@fc.hp.com**

의견을 보내실 때는 문서 제목, 제품 번호, 귀하의 의견, 발견한 오류 그리고 이 설명서를 개선하기 위해 도움이 될 만한 제안 사항 등을 함께 보내주십시오.

# 1 Software Distributor 소개

이 장에서는 Software Distributor(SD-UX)에 대한 개요 정보를 제공하고 SD 명령을 가장 효율적으로 사용하는 데 도움이 되는 중요한 개념에 대해 설명합니다.

표 1-1

## 이 장의 항목

설명하는 항목
24페이지의 “SD-UX 개요”
29페이지의 “SD-UX 개념”
37페이지의 “GUI 및 TUI 명령 사용”
57페이지의 “명령줄에서 작업”

## SD-UX 개요

HP-UX용 Software Distributor(SD-UX)는 중앙 집중식 HP-UX 소프트웨어 관리를 위한 강력한 도구를 제공합니다. SD-UX가 실행되는 각 컴퓨터를 LAN이나 WAN으로 연결하면, 서버로서 다른 시스템에서 해당 자원을 관리하거나 액세스하도록 할 수도 있고 클라이언트로서 다른 시스템의 자원을 관리하거나 사용할 수도 있습니다.

Software Distributor 명령은 HP-UX 운영 체제와 함께 포함되어 있으며, 기본적으로 로컬 호스트의 소프트웨어만 관리합니다. 또한 원격 작업을 통해 중앙 컨트롤러에 연결되어 있는 여러 대의 원격 호스트에서 동시에 소프트웨어를 설치 및 관리할 수 있습니다.

HP-UX 11.00 이상의 버전에서 실행되는 SD-UX는 디스크 없는 NFS 클러스터를 지원하지 않으므로 주의하십시오.

## 네트워크 요구 사항

- 네트워크로 연결된 시스템은 TCP/IP를 지원해야 합니다.
- Software Distributor는 분산, 클라이언트/서버 기술을 기반으로 하고 있으므로 제대로 실행되려면 호스트 시스템에서 특정 네트워크 기능을 사용해야 합니다. 이러한 네트워킹 서비스는 UNIX 실행 수준 2(다중 사용자 모드) 이상에서만 사용할 수 있습니다. Software Distributor는 단일 사용자 모드에서는 실행되지 않습니다.



## SD-UX 프로그램 및 명령

다음 목록에는 각 명령에 대한 간단한 설명과 보다 자세한 정보를 위한 참조가 나와 있습니다.

표 1-2

### SD 명령

명령 및 맨페이지	설명/기능	추가 정보
<i>swinstall</i> (1M)	<ul style="list-style-type: none"> <li>소프트웨어를 설치하거나 업데이트합니다.</li> <li>선택적 GUI</li> </ul>	<ul style="list-style-type: none"> <li>66페이지의 “swinstall을 사용한 설치”</li> </ul>
<i>swlist</i> (1M)	<ul style="list-style-type: none"> <li>설치되어 있는 소프트웨어나 또는 저장소나 미디어상의 소프트웨어를 나열합니다.</li> <li>선택적 GUI</li> </ul>	<ul style="list-style-type: none"> <li>98페이지의 “소프트웨어 나열 (swlist)”</li> <li>160페이지의 “등록된 저장소 나열”</li> </ul>
<i>swcopy</i> (1M)	<ul style="list-style-type: none"> <li>한 저장소에서 다른 저장소로 소프트웨어를 복사합니다.</li> <li>선택적 GUI</li> </ul>	<ul style="list-style-type: none"> <li>139페이지의 “소프트웨어 저장소 복사”</li> </ul>
<i>swremove</i> (1M)	<ul style="list-style-type: none"> <li>설치되어 있는 소프트웨어나 저장소의 소프트웨어를 제거합니다.</li> <li>선택적 GUI</li> </ul>	<ul style="list-style-type: none"> <li>124페이지의 “설치된 소프트웨어 제거 (swremove)”</li> <li>164페이지의 “저장소에서 소프트웨어 제거”</li> </ul>
<i>swpackage</i> (1M)	<ul style="list-style-type: none"> <li>다른 SD-UX 명령에 대한 소스로 사용할 수 있는 소프트웨어 패키지를 작성합니다.</li> </ul>	<ul style="list-style-type: none"> <li>10장, 279페이지의 “소프트웨어 패키지 만들기”</li> </ul>
<i>swconfig</i> (1M)	<ul style="list-style-type: none"> <li>설치된 소프트웨어에 구성 스크립트를 실행합니다.</li> <li>구성, 재구성 또는 구성 해제합니다.</li> </ul>	<ul style="list-style-type: none"> <li>84페이지의 “설치 구성 (swconfig)”</li> <li>11장, 347페이지의 “제어 스크립트 사용”</li> </ul>

표 1-2 SD 명령(계속)

명령 및 맨페이지	설명/기능	추가 정보
<i>swask</i> (1M)	<ul style="list-style-type: none"> <li>나중에 <i>swinstall</i> 또는 <i>swconfig</i>가 사용할 정보를 수집하는 대화식 요청 스크립트를 실행합니다.</li> </ul>	<ul style="list-style-type: none"> <li>11장, 347페이지의 “제어 스크립트 사용”</li> <li>385페이지의 “사용자 응답 요구 (<i>swask</i>)”</li> </ul>
<i>swacl</i> (1M)	<ul style="list-style-type: none"> <li>SD 보안을 위한 액세스 제어 목록(ACL)을 지정, 나열 및 변경합니다.</li> </ul>	<ul style="list-style-type: none"> <li>9장, 233페이지의 “SD-UX 보안”</li> </ul>
<i>swverify</i> (1M)	<ul style="list-style-type: none"> <li>IPD 정보와 실제 설치되어 있는 파일을 비교하여 설치되어 있는 소프트웨어나 저장소 소프트웨어의 무결성을 검증합니다.</li> <li>확인 및 수정 스크립트를 실행합니다.</li> </ul>	<ul style="list-style-type: none"> <li>91페이지의 “설치 확인(<i>swverify</i>)”</li> <li>163페이지의 “저장소 확인 (<i>swverify -d</i>)”</li> </ul>
<i>swmodify</i> (1M)	<ul style="list-style-type: none"> <li>IPD(Installed Products Database) 및 시스템상의 소프트웨어에 대한 정보를 갖고 있는 다양한 카탈로그 파일을 수정합니다.</li> </ul>	<ul style="list-style-type: none"> <li>117페이지의 “IPD 수정 (<i>swmodify</i>)”</li> </ul>
<i>swreg</i> (1M)	<ul style="list-style-type: none"> <li>새로 작성된 저장소를 등록하여 다른 시스템에서 볼 수 있도록 합니다.</li> </ul>	<ul style="list-style-type: none"> <li>153페이지의 “저장소 등록 및 등록 해제(<i>swreg</i>)”</li> </ul>
<i>sd</i> (5)	<ul style="list-style-type: none"> <li>작업을 작성, 모니터링, 예약 및 삭제하기 위해 작업 브라우저 GUI를 시작합니다.</li> <li>원격 작업이 가능한 상태여야 합니다.</li> </ul>	<ul style="list-style-type: none"> <li>6장, 167페이지의 “원격 작업 개요”</li> <li>7장, 193페이지의 “작업 및 작업 브라우저 사용”</li> </ul>

표 1-2 SD 명령(계속)

명령 및 맨페이지	설명/기능	추가 정보
<i>swjob</i> (1M)	<ul style="list-style-type: none"> <li>명령줄에서 작업을 모니터링합니다.</li> <li>원격 작업이 가능한 상태여야 합니다.</li> </ul>	<ul style="list-style-type: none"> <li>207페이지의 “명령줄에서 작업 모니터링”</li> <li>6장, 167페이지의 “원격 작업 개요”</li> <li>7장, 193페이지의 “작업 및 작업 브라우저 사용”</li> </ul>
<i>install-sd</i> (1M)	<ul style="list-style-type: none"> <li>미디어에서 SD-UX를 다시 설치합니다.</li> </ul>	<ul style="list-style-type: none"> <li>부록 C, 457페이지의 “SD-UX 교체 또는 업데이트”</li> </ul>
<i>swagentd</i> (1M)	<ul style="list-style-type: none"> <li>SD-UX 명령에 대한 데몬</li> <li>시스템을 SD-UX 명령에 대한 대상으로 사용하려면 먼저 구성해야 합니다.</li> </ul>	<ul style="list-style-type: none"> <li>맨페이지 참조</li> </ul>

*sd*, *swinstall*, *swcopy*, *swlist* 및 *swremove* 명령에는 창과 풀다운 메뉴로 구성된 선택적 그래픽 사용자 인터페이스(GUI)가 있습니다. 이 GUI 명령은 마우스로 스크린을 탐색하는 대신 키보드를 사용하는 터미널 사용자 인터페이스(TUI)를 제공함으로써 텍스트 기반 터미널에서도 작동합니다.

명령줄에서 모든 SD-UX 명령 및 프로그램을 호출할 수 있습니다. 구문, 옵션, 기본값 및 피연산자는 모든 명령이 비슷합니다. 자세한 내용은 57페이지의 “명령줄에서 작업”을 참조하십시오.

## SD-UX 온라인 설명서

각 명령의 맨 페이지를 보려면 다음과 같이 입력합니다.

**man** *command\_name*

자세한 기술 정보를 보려면 다음과 같이 입력합니다.

**man 5 sd**(SD-UX 개요)

**man 4 sd**(파일 레이아웃)

**man 4 swpackage**(파일 레이아웃 패키징)

## SD-UX 개념

SD-UX 개념, 용어, 소프트웨어 관리 모델 등을 이해하면 명령과 프로그램을 보다 효율적으로 사용할 수 있습니다. 그 밖의 다른 용어에 대해서는 용어집을 참조하십시오.

### 중요한 용어

**호스트**는 SD-UX 명령을 사용하여 소프트웨어를 설치하거나 관리할 모든 시스템을 말합니다. **로컬 호스트**는 SD-UX 명령을 호출할 시스템입니다.

원격 작업을 가능하게 하면 한 대 이상의 **원격 호스트**(SD-UX 명령을 호출한 시스템이 아닌 다른 호스트)에서 SD-UX를 사용할 수 있습니다. 원격 작업에 대한 자세한 내용은 6장, 167페이지의 “원격 작업 개요”를 참조하십시오.

**컨트롤러**는 시스템에서 호출한 SD-UX 프로그램 또는 명령(swininstall, swcopy 등)을 말합니다. 컨트롤러는 다른 시스템에서 데이터를 처리하거나 프로세서를 시작할 수도 있습니다.

**저장소**는 SD-UX로 관리할 수 있는 소프트웨어 제품 저장 장소를 말합니다. 저장소는 특수한 형식의 디렉토리거나 테이프, **CD-ROM**, **DVD**와 같은 물리적 미디어일 수 있습니다. (실제로는 **CD-ROM**과 **DVD** 저장소가 디렉토리 저장소의 특수한 경우입니다.) 디렉토리 저장소는 네트워크를 통해 액세스할 수 있으므로 유용합니다. 또한 다른 저장소에서 복사된 소프트웨어들을 저장하는 데에도 종종 사용됩니다.

일반적으로 **대상**은 호스트(구체적으로는 호스트의 파일 시스템) 또는 호스트에 있는 저장소를 의미합니다. **소스**는 소프트웨어가 설치되거나 복사되고 있는 저장소를 말하며 종종 **소스 저장소**라고도 합니다.

예를 들어, swinstall 명령을 사용하는 기본 설치 작업에는 소스 저장소에서 호스트 자체에 있는 대상 위치로 소프트웨어를 설치하는 것이 포함됩니다. 소스 저장소는 네트워크상의 특정 서버에 있는 대상이나 디렉토리 저장소에서 액세스 가능한 물리적인 미디어일 수 있습니다. 대상 호스트는 명령이 호출된 호스트와 동일한 호스트이거나 (즉, 로컬 호스트), 원격 작업이 가능한 경우 네트워크상의 다른 호스트일 수도 있습니다.

SD-UX 개념

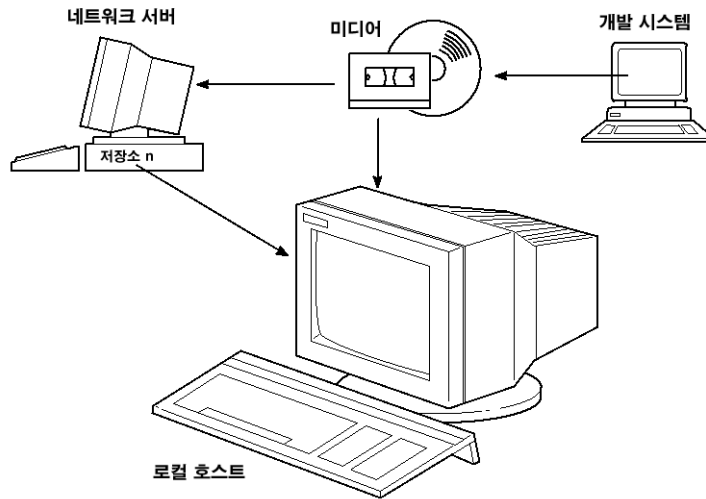
기본적인 복사 작업(`swcopy` 명령 사용)은 대상이 호스트 자체가 아닌 호스트상의 저장소라는 점을 제외하고는 매우 유사합니다.

대부분 작업의 경우 컨트롤러 프로그램에서 기본 소프트웨어 관리 작업을 실행하는 `swagent`라는 에이전트를 사용하여 호스트와 저장소에 액세스합니다. 에이전트에는 `swagentd`라는 데몬을 통해 액세스합니다. SD-UX가 로컬 호스트에서 작동할 때 컨트롤러와 에이전트는 모두 로컬 호스트에서 실행됩니다. 원격 작업의 경우 에이전트는 원격 호스트에서 실행됩니다.

그림 1-1, “SD-UX 시스템”에서는 소프트웨어를 개발한 다음 SD 형식의 미디어로 패키징하여 이를 직접 액세스하거나 서버에서 저장소 디렉토리로 복사한 다음 네트워크를 통해 액세스하는 방법을 보여줍니다.

그림 1-1

SD-UX 시스템



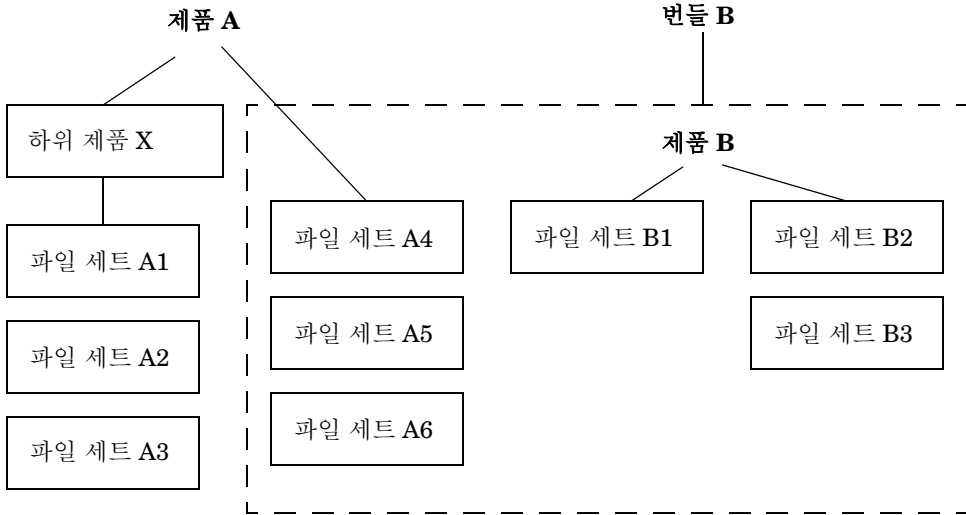
소프트웨어 구조

SD-UX 명령은 관리하려는 응용 프로그램이나 운영 체제 구성 요소를 구성하는 소프트웨어 객체 계층 구조에 대해 작동합니다.

## 소프트웨어 객체

번들	여러 제품에서 수집되어 특정 목적을 위해 캡슐화된 파일 세트 모음입니다. 번들은 소프트웨어 저장소에 위치할 수 있으며 <b>SD-UX</b> 명령은 번들을 하나의 엔터티로 취급합니다. 모든 <b>HP-UX OS</b> 소프트웨어는 번들로 패키징됩니다. 번들은 파일 세트 그룹이나 제품 그룹으로 구성됩니다. 고객이 번들을 만드는 것은 지원되지 않습니다.
제품	파일 세트 또는 (선택적으로) 하위 제품 및 제어 스크립트의 집합입니다. <b>SD-UX</b> 명령은 제품에 초점을 두고 있지만 사용자가 하위 제품과 파일 세트를 지정할 수 있도록 해줍니다.  제품 자체의 다른 릴리즈는 물론 다른 플랫폼과 운영 체제에 대해서도 서로 다른 버전의 제품을 정의할 수 있습니다. 하나의 배포 미디어나 저장소에 여러 개의 다른 버전이 포함될 수 있습니다.
하위 제품	제품에 파일 세트가 여러 개 포함되는 경우 논리적으로 연관된 파일 세트를 그룹화하기 위해 하위 제품을 사용할 수 있습니다.
파일 세트	파일 세트에는 제품을 구성하는 모든 파일과 제어 스크립트가 포함됩니다. 파일 세트는 한 제품의 일부만 될 수 있지만 여러 <b>HP-UX</b> 번들이나 하위 제품에 포함될 수 있습니다. 제품과 마찬가지로 다른 플랫폼과 운영 체제에 대해 다른 버전의 파일 세트를 정의할 수 있습니다.  파일 세트는 <b>SD-UX</b> 가 관리하는 객체 중 가장 낮은 수준의 객체입니다.

그림 1-2 HP-UX 소프트웨어 구조의 예



### IPD(Installed Products Database)

SD-UX는 IPD(Installed Products Database)를 사용하여 시스템에 어떠한 소프트웨어가 설치되었는지 추적합니다. IPD는 루트 디렉토리(/) 아래에 설치된 모든 제품에 대한 정보가 들어 있는 일련의 파일 및 하위 디렉토리입니다. (저장소의 경우 이 정보는 저장소 디렉토리 아래의 카탈로그 파일에서 관리됩니다.)

swinstall, swconfig, swcopy 및 swremove 명령은 IPD 및 카탈로그 파일 정보를 자동으로 추가, 변경 및 삭제합니다. swlist 및 swverify 명령은 IPD 및 카탈로그 정보를 사용하여 명령 동작에 영향을 줍니다.

IPD는 설치됨 또는 구성됨 등을 비롯한 소프트웨어 상태를 추적합니다.

### 제어 스크립트

제품 및 파일 세트에는 SD-UX 명령으로 수행할 수 없는 확인 및 기타 작업을 수행하는 제어 스크립트가 들어 있을 수 있습니다. SD-UX에서는 다음과 같은 종류의 스크립트를 지원합니다.



Checkinstall	각 대상을 분석하여 설치 및 구성이 수행되었는지 여부를 판별합니다( <code>swinstall</code> 명령으로 실행).
Checkremove	제거 및 구성 해제가 수행될 수 있는지 판별하기 위해 각 대상을 분석합니다( <code>swremove</code> 명령으로 실행).
Configure	설치되어 있는 파일 세트나 제품을 구성합니다( <code>swconfig</code> 및 <code>swinstall</code> 명령으로 실행).
Fix	설치된 소프트웨어의 문제점을 수정하고 보고합니다( <code>swverify</code> 명령으로 실행).
Postinstall	파일 세트나 제품이 설치된 직후 추가적인 설치 작업을 수행합니다( <code>swinstall</code> 명령으로 실행).
Postremove	파일 세트나 제품이 제거된 직후 추가적인 제거 작업을 수행합니다( <code>swremove</code> 명령으로 실행).
Preinstall	소프트웨어 파일을 설치하기 직전에 오래된 파일 제거와 같은 파일 작업을 수행합니다( <code>swinstall</code> 명령으로 실행).

Preremove	소프트웨어 파일을 제거하기 직전에 설치 전 스크립트에 의해 작성된 파일을 제거하는 등 추가적인 파일 작업을 수행합니다 (swremove 명령으로 실행).
Request	설치 또는 구성 프로세스의 일부로 사용자로부터의 대화식 응답을 요구합니다(swask, swconfig 및 swinstall 명령으로 실행).
Unconfigure	구성 스크립트로 수행한 구성을 취소합니다(swconfig 및 swremove 명령으로 실행).
Unpostinstall	설치 프로세스 중에 swinstall이 복구를 실행해야 하는 경우 설치 후 스크립트에 의해 수행된 작업을 취소합니다 (swinstall 명령으로 실행).
Unpreinstall	설치 프로세스 중에 SD가 복구를 실행해야 하는 경우 설치 전 스크립트에 의해 수행된 작업을 취소합니다(swinstall 명령으로 실행).
Verify	표준 swverify 확인에 추가하여 파일 세트나 제품의 구성을 확인합니다(swverify 명령으로 실행).

### 추가 정보

자세한 내용은 11장, 347페이지의 “제어 스크립트 사용”을 참조하십시오.

### 환경 변수

SD-UX 명령 및 프로그램은 외부 환경 변수(예: language 및 charset 변수)와 제어 스크립트가 사용하는 변수에 의해 영향을 받습니다. 외부 환경 변수에 대한 설명은 11장, 347페이지의 “제어 스크립트 사용”을 참조하십시오.

## 소프트웨어 종속성

올바르게 설치되거나 실행되기 위해서 다른 소프트웨어가 있어야 하는 소프트웨어는 종속성이 있는 것으로 간주됩니다. `swconfig`, `swcopy`, `swinstall`, `swremove`, `swverify` 등의 명령에 대해 소프트웨어를 지정할 때 종속성을 충족시키기 위해 이러한 명령이 자동으로 추가 소프트웨어를 선택할 수도 있습니다.

### 명령 및 옵션이 종속성과 상호 작용하는 방식

명령 옵션을 통해 소프트웨어 종속성을 처리하는 방법을 제어할 수 있습니다. 예를 들어 `swinstall` 및 `swcopy` 명령의 종속성 처리 방법은 `enforce_dependencies` 명령 옵션에 따라 달라집니다.

종속성을 규정하는 또 다른 옵션은 `autoselect_dependencies` 옵션입니다. 이 옵션은 종속성 충족 여부를 근거로 설치나 복사에 사용될 소프트웨어를 자동으로 표시할 것인지를 판별합니다. (옵션에 대한 자세한 내용은 61페이지의 “명령 옵션 사용”을 참조하십시오.)

### 종속성 해결 방식

소스 저장소의 다른 소프트웨어와 관련되어 종속성이 해결되려면 해당 소프트웨어는 다음과 같아야 합니다.

- 완전해야 합니다. (종속 소프트웨어가 제품 전체나 하위 제품인 경우 소스 저장소에 완전히 존재해야 합니다.)
- 소스에서 적절한 소프트웨어 상태여야 합니다.(즉, 사용 가능 상태).
- 오류가 없어야 합니다.(예: 비호환성 오류).

`swconfig`, `swcopy`, `swinstall` 또는 `swverify` 작업 중에 종속 소프트웨어를 소스에서 사용할 수 **없는** 경우, 해당 종속 소프트웨어는 다음과 같아야 합니다.

- 대상 호스트에 있어야 합니다.
- 완전해야 합니다. (종속 소프트웨어가 제품 전체나 하위 제품인 경우 완전히 설치되어야 합니다.)
- 적절한 소프트웨어 상태여야 합니다. (종속되는 소프트웨어가 설치 및 구성되어야 하는 경우 종속 소프트웨어가 구성 상태여야 하고, 종속되는 소프트웨어가 구성은 아니고 설치만 되어야 하는 경우 설치 상태여야 하며, 종속되는 소프트웨어가 복사되려면 사용 가능 상태여야 합니다.)

- 오류가 없어야 합니다(예: 비호환성 오류).

종속성이 있는 소프트웨어를 선택하고 하나 이상의 사용 가능 객체가 그 종속성을 해결하는 경우 SD-UX는 자동으로 최신 호환 버전을 선택합니다.

### 종속성의 종류

소프트웨어 패키징 프로그램에서 필수 조건, 전제 조건 및 제외 조건을 종속성으로 정의할 수 있습니다. 이러한 종속성은 제품 내 파일 세트 간에 지정할 수 있으며 해당 종속성을 충족시키는 파일 세트 버전도 지정할 수 있습니다. 종속성은 파일 세트와 다른 제품 사이에서도 지정할 수 있습니다. 버전 및 기타 제품 속성에 대한 표현식도 지원됩니다.

필수 조건	해당 객체가 정상 작동하려면 다른 객체를 필요로 하지만 로드 순서는 상관없습니다.
전제 조건	해당 객체가 정상적으로 설치되거나 구성되려면 먼저 다른 객체가 설치 및/또는 구성되어야 합니다. 전제 조건은 작업 순서를 <b>조정</b> 합니다.
제외 조건	해당 객체가 설치되거나 구성되기 위해서는 다른 객체가 없어야 합니다.

### 보호되는 소프트웨어에 대한 작업

일부 HP 소프트웨어 제품은 보호되는 소프트웨어입니다. 즉 코드워드와 고객 ID를 입력하지 않으면 해당 소프트웨어를 설치하거나 복사할 수 없습니다. 고객 ID는 코드워드의 소유자를 고유하게 식별하여 설치를 특정 소유자만 할 수 있도록 제한하는 역할을 합니다. 코드워드와 고객 ID를 찾으려면 소프트웨어와 함께 제공된 CD 인증서를 살펴 보십시오.

**사용자는 반드시 이러한 방식으로 코드워드 및 소프트웨어를 사용해야 합니다.**

하나의 코드워드도 미디어상의 제품 대부분 또는 모두에 대한 잠금을 해제합니다. 보호되는 제품을 추가로 구매하는 경우 HP는 코드워드를 추가로 제공합니다. 사용자가 코드워드를 입력하면 SD-UX는 이를 추적합니다. 즉, 해당 소프트웨어에 액세스할 때 매번 코드워드를 입력할 필요가 없습니다.

swinstall, swcopy 및 swlist 명령은 코드워드를 사용하여 소프트웨어를 관리합니다.

## GUI 및 TUI 명령 사용

swinstall, swcopy, swlist, swremove 명령은 그래픽 사용자 인터페이스(GUI)와 터미널 사용자 인터페이스(TUI)를 제공합니다. GUI와 TUI에는 다음과 같은 장점이 있습니다.

- 소프트웨어 관리 작업을 대화식으로 짧은 시간에 작성하여 시각적으로 모니터링할 수 있습니다.
- 쉽게 작업 효과를 분석하고 실패한 작업을 다시 시도해 볼 수 있습니다.
- 명령줄에서 복잡한 명령을 입력하는 데 필요한 광범위한 기본값, 옵션, 소프트웨어 선택 사항 및 기타 변수 등을 알지 못해도 상관없습니다.

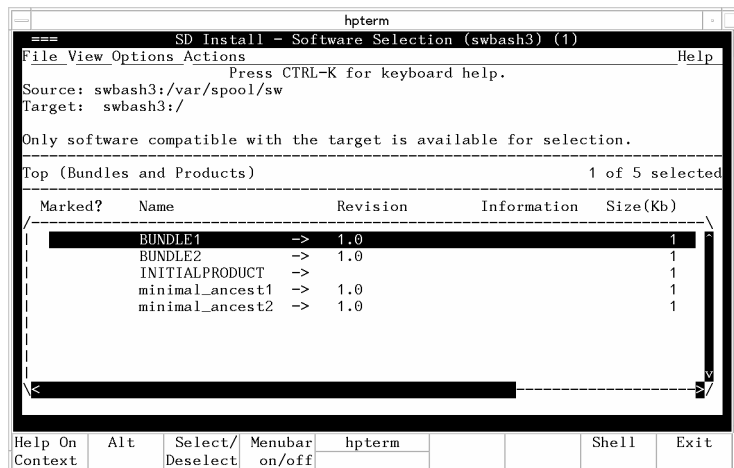
(원격 작업이 가능한 경우 추가로 GUI 인터페이스를 사용할 수 있습니다. 자세한 내용은 6장, 167페이지의 “원격 작업 개요”를 참조하십시오.)

## 터미널 사용자 인터페이스

터미널 사용자 인터페이스를 사용하여 텍스트 기반 터미널이 연결되어 있는 시스템에서 SD-UX GUI 기능을 사용할 수 있습니다. TUI의 경우 **화살표**, **Tab**, **스페이스** 및 **Return** 키를 사용하여 이동합니다.

그림 1-3

터미널 사용자 인터페이스(TUI)



---

**참고**

---

이 설명서의 GUI 명령의 모든 예는 TUI에도 적용됩니다.

## GUI/TUI 명령 시작

swinstall, swcopy 또는 swremove에서 GUI나 TUI를 시작하려면 다음을 입력합니다.

```
/usr/sbin/swinstall
```

—또는—

```
/usr/sbin/swcopy
```

—또는—

```
/usr/sbin/swremove
```

---

**팁**

---

**/usr/sbin** 접두어를 입력할 필요가 없도록 PATH에 **/usr/sbin**을 입력합니다.

DISPLAY 변수를 설정하지 않으면 기본적으로 TUI가 시작됩니다.

GUI를 호출하고 동시에 다른 명령줄 인수를 지정하려면 **-i** 옵션을 포함해야 합니다. 예를 들면 다음과 같습니다.

```
swinstall -i -s sw_server cc pascal
```

swlist GUI를 호출하려면 항상 **swlist -i** 옵션을 사용해야 합니다.

---

**참고**

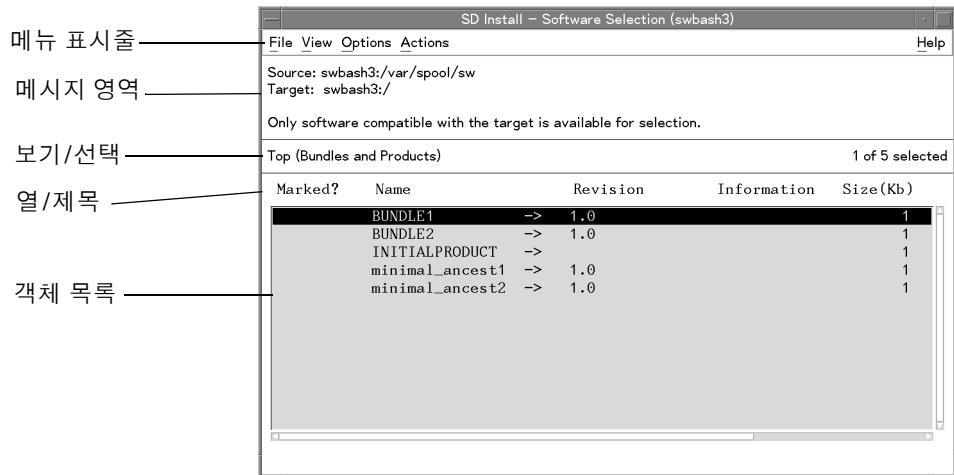
---

또한 HP의 Service Control Manager(SCM)나 Systems Administration Manager(SAM) 응용 프로그램에서도 SD-UX GUI를 실행할 수 있습니다.

## Window 구성 요소

주 GUI/TUI 창(그림 1-4, “GUI 창 구성 요소”)에는 다음 구성 요소가 들어 있습니다.

그림 1-4 GUI 창 구성 요소



메뉴 표시줄

**File, View, Options, Actions, Help** 등의 풀다운 메뉴를 제공합니다. 각 선택 항목에는 그에 따른 추가 작업을 위한 하위 메뉴가 들어 있습니다. 메뉴의 항목은 선택된 항목이 강조 표시되는지 여부에 따라 보일 수도 있고 보이지 않을 수도 있습니다. 일부 작업은 회색으로 열게 표시되어 특정 항목에 대해 사용할 수 없음을 나타낼 수도 있습니다.

메시지 영역

메시지 및 시스템 정보를 표시합니다.

보기/선택

현재의 소프트웨어 보기 객체 목록에서 선택된 항목의 수를 설명합니다.

열

객체 목록의 정보 열에 대한 제목입니다.

객체 목록

소프트웨어 선택 사항, 번들, 제품, 대상 또는 선택, 분석 및 세부 사항에 대한 기타 정보를 나열합니다.

## 객체 목록의 시작 항목 및 마감 항목

Software Selection 창 객체 목록은 계층 구조로 이루어집니다. 목록의 각 객체를 열고 그 내용을 표시할 수 있습니다. 열 수 있는 다른 객체가 들어 있는 객체는 객체 이름 옆에 화살표(→)가 표시됩니다.

- 하위 제품을 열려면 해당 객체를 두 번 클릭하거나 이름을 강조 표시한 다음 **Actions**→**Open Item**을 선택합니다. 예를 들어 SD-DATABASE 제품의 하위 제품을 보려면 SD-DATABASE를 두 번 클릭하여 엽니다. 그러면 객체 목록에 SD-DATABASE의 하위 제품이 표시됩니다.
- 객체를 닫고 이전 목록으로 되돌아가려면 목록의 첫 번째 항목(..(go up))을 두 번 클릭하거나 해당 항목을 강조 표시한 다음 **Actions**→**Close Level**을 선택합니다.

제품이 열리면 하위 제품과 파일 세트가 같은 목록에 표시될 수 있습니다. 제품 수준에서는 제품만 함께 나열됩니다.

파일 세트가 SD-UX에 의해 관리되는 객체 계층 구조에서 가장 낮은 단계의 객체입니다. 파일 내용은 볼 수 없지만 각 파일 세트의 파일 목록과 각 파일에 대한 정보는 볼 수 있습니다.

## 객체 목록에 항목 표시

객체 목록의 객체를 표시하는 데에는 두 가지 방법이 있습니다.

- 메뉴 표시줄 사용
  1. 해당 객체를 클릭하여 강조 표시합니다.
  2. **Actions**→**Mark for Install**(또는 **Mark for Copy**나 **Mark for Remove**)을 선택합니다.
- 팝업 메뉴 사용
  1. 해당 객체를 클릭하여 강조 표시합니다.
  2. 마우스 오른쪽 단추를 클릭하여 팝업 메뉴를 표시합니다.
  3. **Mark for Install**(또는 **Mark for Copy**나 **Mark for Remove**)을 선택합니다.

플래그(Yes, Partial 또는 공백)는 목록의 항목이 어떤 작업에 대해 표시되었는지 여부를 보여 줍니다(Marked? 열 참조).

(TUI의 경우 커서가 해당 항목에 있을 때 **스페이스** 키를 눌러서 표시한 다음 **m** 키를 누릅니다. **u** 키로 표시를 취소합니다.)



## 호스트 파일 미리 선택

defaults.hosts 파일에는 GUI/TUI 프로그램에서 사용되는 호스트의 목록이 들어 있습니다. 이 목록으로 소스 및 대상 시스템을 미리 선택할 수 있습니다. 이러한 목록 들은 \$HOME/.swdefaults.hosts 또는 /var/adm/defaults.hosts 파일에 저장됩니다.

각 대화식 명령에서 루트나 저장소를 포함하는 대상 호스트는 별도의 목록(hosts, hosts\_with\_depots)으로 이 파일에 지정됩니다. 호스트 목록은 {} 안에 표시되고 공백(공백, 탭, 줄바꿈 등)으로 구분됩니다. 예를 들면 다음과 같습니다.

```
swinstall.hosts={hostA hostB hostC hostD hostE hostF}  
swcopy.hosts_with_depots={hosts}
```

해당 프로그램을 사용할 때 목록에서 소스 시스템을 선택할 수 있는 대화 상자가 표시 되어 defaults.hosts에 지정되었거나 이전 세션에 저장된 모든 호스트를 표시합니다. 일단 소스 시스템에 정상적으로 액세스하면 해당 호스트는 자동으로 defaults.hosts 파일의 목록에 추가되어 대화 상자에 표시됩니다.

defaults.hosts에 지정된 호스트가 없는 경우 로컬 호스트와 기본 소스 호스트만 목록에 표시됩니다.

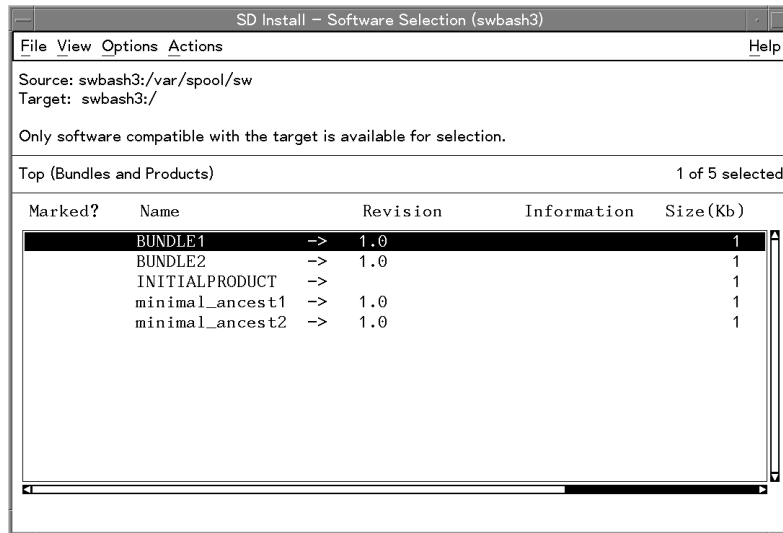
호스트 시스템이 목록에 없으면 GUI/TUI 프로그램에서 새로운 이름을 입력할 수 있습니다.

## Software Selection 창

Software Selection 창(그림 1-5, “Software Selection 창”)은 모든 SD-UX GUI 프로그램의 표준 창입니다. 이 창에는 표준 메뉴 표시줄, 메시지 영역 및 선택 가능한 소프트웨어의 객체 목록 등이 표시됩니다. 메뉴 항목은 다음에서 설명됩니다.

그림 1-5

### Software Selection 창



## 세션 및 파일 관리—File 메뉴

**File** 메뉴는 세션 파일 관리, 검색 및 인쇄를 위한 기본 도구입니다.

### GUI 세션 파일

GUI 명령 중 하나를 실행시킴으로써 세션이 정의됩니다. 명령 실행에 사용된 옵션, 소스 사양, 소프트웨어 선택 사항 및 대상 호스트 등 모든 세션 정보는 자동으로 저장됩니다. 따라서 세션이 정상적으로 완료되기 전에 종료되더라도 해당 명령을 다시 실행할 수 있습니다(63페이지의 “세션 파일” 참조).

**Save Session** 또는 **Save Session As**를 **File** 메뉴에서 선택하면 언제라도 세션 정보를 파일로 저장할 수 있습니다. **Recall Session**을 선택하면 이전에 저장된 세션 파일의 설정을 가져올 수 있습니다. **Clear Session**은 모든 옵션과 피연산자를 기본값으로 재설정합니다.

각 세션은 각 명령의 해당 파일에 저장됩니다. 예를 들면 다음과 같습니다.

```
$HOME/.sw/sessions/swinstall.last  
$HOME/.sw/sessions/swcopy.last  
$HOME/.sw/sessions/swremove.last
```

이 파일은 GUI를 시작할 때마다 덮어씁니다.

---

### 참고

세션 파일을 다시 실행할 때 세션 파일의 값이 시스템 기본 파일의 값에 우선합니다. 마찬가지로 GUI를 호출할 때 지정하는 명령줄 옵션이나 매개 변수는 모두 세션 파일의 값에 우선합니다.

### 텍스트 검색 수행

**Search...**를 선택하면 창에서 활성 목록의 텍스트 검색을 수행할 수 있습니다.

## 소프트웨어 보기 변경—View 메뉴

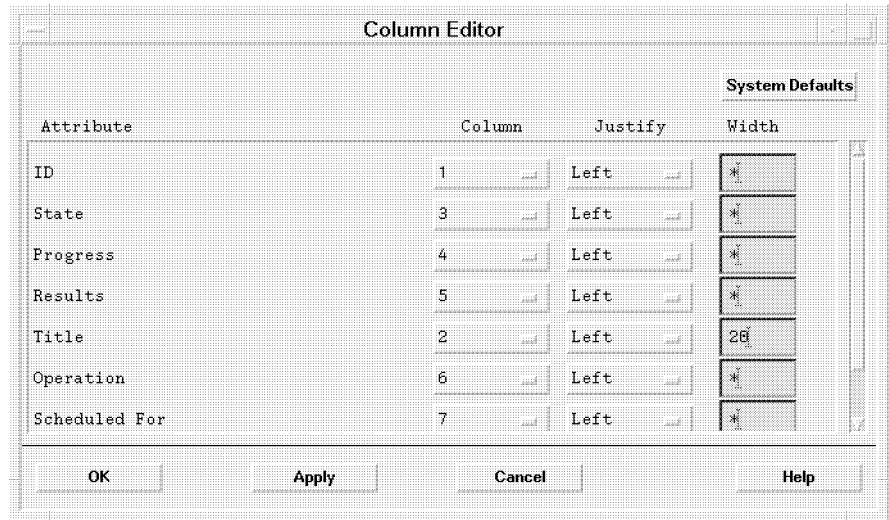
**View** 메뉴는 창 보기 기본 설정을 관리합니다.

### Columns...

**View**→**Columns...**를 선택하면 현재 객체 목록 열의 형식을 다시 지정할 수 있는 **Column Editor** 대화 상자(그림 1-6, “**Column Editor**”)가 표시됩니다. 보기 가능한 모든 객체 속성이 표시됩니다.

그림 1-6

### Column Editor



이 편집기에서는 1에서 속성 전체 수까지 표시하고 객체 목록에서 표시된 속성을 제거하는 **Ignore** 옵션도 표시합니다.

**Justify** 열의 **Left** 또는 **Right** 단추를 클릭하여 속성값을 조정하여 지정할 수 있습니다.

**Width** 열에서 커서를 적절한 텍스트 필드에 놓고 열 너비를 설정한 다음 해당 너비(문자 수)를 입력합니다. 별표(\*)를 사용하면 자동으로 열 크기가 설정됩니다.

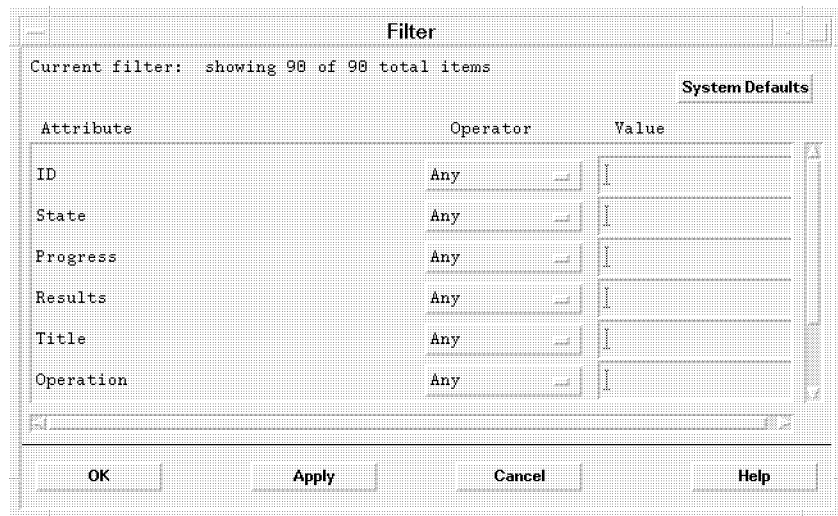
- 객체 목록에 변경 사항을 적용하려면 **Apply**를 선택합니다. 변경 사항을 반영하여 해당 목록이 업데이트되고 **Column Editor** 대화 상자는 그대로 열려 있습니다.
- 변경 사항을 적용하고 편집기를 닫으려면 **OK**를 선택합니다.

- 원래 기본값으로 돌아가려면 **System Defaults**를 선택합니다.
- 변경 사항을 취소하고 객체 목록 창으로 돌아가려면 **Cancel**을 선택합니다.
- 다음에 응용 프로그램을 실행할 때를 대비하여 변경 사항을 저장하려면 **View→Save View as Default**를 선택합니다.

### Filter...

그림 1-7

Filter 대화 상자



**View→Filter...**를 선택하면 각 속성에 대해 원하는 필터링 유형을 지정할 수 있는 Filter 대화 상자(그림 1-7, “Filter 대화 상자”)가 표시됩니다.

**Operator** 메뉴 단추를 사용하면 지정된 속성에 대해 연산자를 지정할 수 있습니다. 다음 표에서는 연산자 종류를 보여줍니다.

표 1-3

## 연산자 종류

Any	속성 값에 상관없이 객체를 표시합니다.
Matches	속성 값이 Value 열에 지정된 값과 정확하게 일치하면 객체를 표시합니다.
Not	속성 값이 Value 열에 지정된 값과 일치하지 않는 객체를 표시합니다.
Less Than	속성 값이 Value 열에 지정된 값보다 작은 객체를 표시합니다. <i>Less than</i> 은 정수 값에서 더 작은 수 또는 알파벳에서 앞에 나오는 문자를 의미합니다.
Greater Than	속성 값이 Value 열에 지정된 값보다 큰 객체를 표시합니다. <i>Greater than</i> 은 정수 값에서 더 큰 수 또는 알파벳에서 뒤에 나오는 문자를 의미합니다.

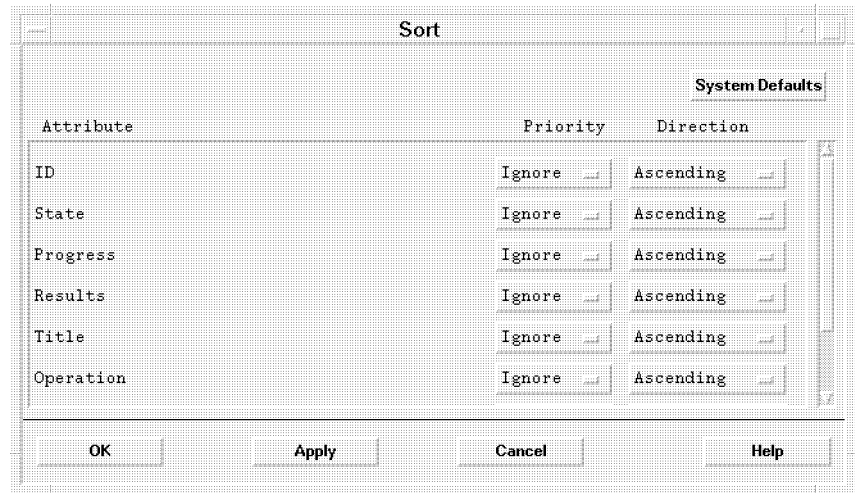
- Matches 및 Not의 경우 와일드카드로 별표(\*)를 사용하고 한 문자 일치를 나타내는 경우 물음표(?)를 사용합니다.
- 객체 목록에 변경 사항을 적용하고 **Filter** 대화 상자를 열린 상태로 두려면 **Apply**를 선택합니다.
- 변경 사항을 적용하고 **Filter** 대화 상자를 닫으려면 **OK**를 선택합니다.
- 원래 기본값으로 돌아가려면 **System Defaults**를 선택합니다.
- 변경 사항을 무시하고 **Filter** 대화 상자를 닫으려면 **Cancel**을 선택합니다.
- 다음에 응용 프로그램을 실행할 때를 대비하여 변경 사항을 저장하려면 **View**→**Save View as Default**를 선택합니다.

## Sort...

**View**→**Sort...**를 선택하면 객체 목록의 정렬 방법을 지정할 수 있는 **Sort** 대화 상자(그림 1-8, “**Sort** 대화 상자”)가 표시됩니다. 보기 가능한 모든 객체 속성이 표시됩니다. 각 속성에 대해 원하는 정렬 종류를 지정할 수 있습니다.

그림 1-8

## Sort 대화 상자



Priority 열에서 값 1에서 속성 전체 수까지 표시하고 정렬에서 속성을 제외시키는 Ignore 옵션을 표시합니다. 정렬 우선 순위 1은 해당 속성의 목록을 먼저 정렬합니다.

- 오름차순 또는 내림차순 정렬 여부를 지정하려면 Direction 메뉴 단추를 선택합니다.
- 객체 목록에서 변경 사항을 적용하고 **Sort** 대화 상자를 그대로 열어 두려면 **Apply**를 선택합니다.
- 정렬을 적용하고 **Sort** 대화 상자를 닫으려면 **OK**를 선택합니다.
- 변경 사항을 무시하고 **Sort** 대화 상자를 닫으려면 **Cancel**을 선택합니다.
- 원래 기본값으로 돌아가려면 **System Defaults**를 선택합니다.

### Save View as Default

차후 세션을 위해 변경 사항을 저장하려면 **View**→**Save View as Default**를 선택합니다. 보기 기본 설정에서 변경 사항은 모두 `/var/adm/sw/ui/preferences/username.prefs`에 저장됩니다. 여기에서 `username`은 사용자의 로그인 이름입니다.

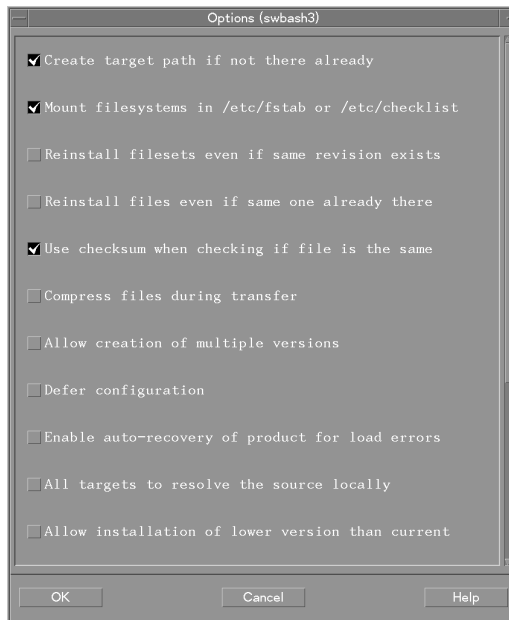
### 옵션 변경 및 객체 목록 새로 고침—Options 메뉴

**Options** 메뉴를 사용하여 객체 목록을 새로 고치고 명령 작동 방식과 정책을 제어하는 옵션 기본값을 변경할 수 있습니다. **Options**→**Refresh List**를 선택하면 객체 목록을 업데이트하여 변경 사항을 반영합니다.

**Options**→**Change Options**를 선택하면 명령에 대한 제한된 옵션 집합을 변경할 수 있는 **Options** 대화 상자(그림 1-9, “Options 대화 상자”)가 열립니다. 이 옵션들은 대화식 세션 동안에만 바뀌어 적용됩니다. 추후 세션에 대해 옵션을 변경하려면 세션 파일을 저장하거나(43페이지의 “세션 및 파일 관리—File 메뉴” 참조) 옵션 파일 중 하나를 편집해야 합니다(61페이지의 “명령 옵션 사용” 참조).

그림 1-9

### Options 대화 상자





---

**참고**

옵션 값을 변경할 때는 주의해야 합니다. 옵션 값을 융통성있게 사용할 수 있지만 부적절한 값으로 변경하면 부정적인 결과가 발생할 수 있습니다. 온라인 도움말과 부록 A, 399페이지의 “명령 옵션”을 참조하여 각 옵션을 충분히 이해한 다음 변경하십시오.

---

## 작업 수행—Actions 메뉴

GUI/TUI 프로그램의 각 **Action** 메뉴에는 해당 명령에 대한 일련의 작업이 있습니다. 이들 작업은 어떤 명령을 실행하느냐에 따라 달라집니다. (회색으로 열게 표시된 작업을 사용할 수 있도록 하려면 객체 목록에서 항목을 클릭해야 할 수도 있습니다.) 다음 작업은 `swinstall`, `swcopy`, `swlist` 및 `swremove`에 공통적입니다.

### Open Item/Close Level

**Open Item** 또는 **Close Level** 메뉴를 선택하면 선택한 객체의 내용을 보거나 닫을 수 있습니다.

각 객체 목록은 계층 구조로 이루어집니다. 이름 다음에 화살표(→)가 있는 객체는 열어서 다른 항목을 확인할 수 있습니다. 예를 들어 특정 제품의 하위 제품을 보려면 객체를 두 번 클릭하거나 **Actions ->Open Item**을 선택하여 해당 객체를 열 수 있습니다. 그러면 객체 목록에 해당 제품에 대한 하위 제품 목록이 표시됩니다. 하위 제품을 열려는 경우 해당 하위 제품을 두 번 클릭하면 그에 따른 파일 세트가 표시됩니다. TUI에서 열 항목으로 커서를 이동한 다음 **Return**을 클릭합니다.

제품을 열면 해당 하위 제품 및 하위 제품에 속하지 않은 파일 세트가 모두 목록에 표시됩니다. 제품 수준에서는 제품만 표시됩니다. 소프트웨어 보기가 **Bundle**이고 해당 번들을 열면 그 번들에 전체적으로 또는 부분적으로 포함되어 있는 모든 **HP-UX OS** 제품이 표시됩니다. 제품 중 하나를 열면 열린 제품과 열린 번들의 하위 제품과 파일 세트만 표시됩니다.

객체를 닫고 이전 목록으로 돌아가려면 목록의 첫 번째 항목(..(go up))을 두 번 클릭하거나 **Actions->Close Level**을 선택합니다. TUI에서 **Actions** 메뉴의 **Close Level**을 사용하거나 (..(go up)) 항목을 강조 표시한 채로 **Return**을 눌러야 합니다.

### Add/Save Software Group

이 선택 항목을 통해 표시된 소프트웨어 그룹을 저장하고 다시 사용할 수 있습니다.

**Save Software Group** 메뉴를 선택하면 표시된 소프트웨어의 현재 목록을 그룹으로 저장하는 **Save Software Group** 대화 상자(그림 1-10, “**Save Software Group** 대화 상자”)가 열립니다. SD는 `$HOME/.sw/software/` 또는 사용자가 지정한 디렉토리에 그룹 정의를 저장합니다.

**Add Software Group** 메뉴를 사용하면 이전에 저장된 소프트웨어 선택 사항 그룹을 다시 불러 사용할 수 있습니다.

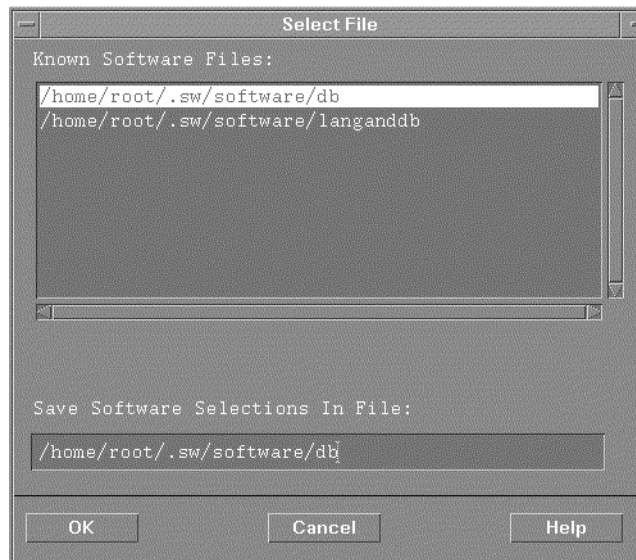
---

### 참고

중속성으로 인해 자동으로 표시된 소프트웨어는 소프트웨어 그룹에 포함되지 않습니다. 중속성은 **Add Software Group**을 선택할 때마다 매번 다시 계산됩니다. 중속성에 대한 자세한 내용은 35페이지의 “소프트웨어 중속성”을 참조하십시오.

---

그림 1-10 **Save Software Group** 대화 상자

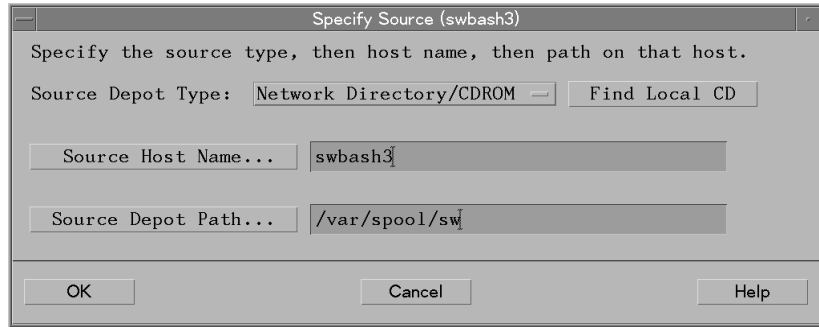


## 소스 변경

**Change Source...** 메뉴를 선택하면 사용할 소프트웨어 소스를 변경할 수 있는 **Specify Source** 대화 상자(그림 1-11, “Specify Source 대화 상자”)가 열립니다.

그림 1-11

### Specify Source 대화 상자



- (선택) 다른 호스트 시스템을 지정하려면 소스 호스트 이름을 입력하거나 다음을 수행합니다.
  - Source Host Name** 단추를 클릭합니다. 대화 상자가 표시되어 `defaults.hosts` 파일(`$HOME/.sw/defaults.hosts` 또는 `/var/adm/sw/defaults.hosts`)에 들어 있는 모든 호스트 시스템 이름이 나열됩니다.
  - 목록에서 호스트 이름을 선택합니다.
  - OK**를 클릭합니다. 호스트 이름이 **Specify Source** 대화 상자의 해당 상자에 표시됩니다.
- (선택) 저장소에 대한 경로를 지정하려면 새로운 경로를 입력하거나 다음을 수행합니다.
  - Source Depot Path** 단추를 클릭하여 소스 호스트에 등록되어 있는 저장소 목록을 표시합니다.
  - 저장소 중 하나를 강조 표시합니다.
  - OK**를 클릭하면 **Specify Source** 대화 상자에 표시됩니다.
- OK**를 클릭합니다. **Specify Source** 대화 상자가 닫히고 지정된 저장소에 들어 있는 소프트웨어가 **Software Selection** 창에 표시됩니다.

## Change Target

**Change Target...** 메뉴를 선택하면 소프트웨어 작업의 대상을 변경할 수 있는 **Select Target Path** 대화 상자(그림 1-12, “**Select Target Path** 대화 상자”)가 열립니다. **Root Path** 단추를 누르면 선택할 대상 경로 목록이 열립니다(그림 1-13, “**Shared Root Paths** 대화 상자”).

SD-UX 로컬 작업의 경우 대상은 항상 로컬 호스트상의 디렉토리입니다. 원격 대상 지정에 대한 자세한 내용은 6장, 167페이지의 “원격 작업 개요”를 참조하십시오.

그림 1-12

### Select Target Path 대화 상자

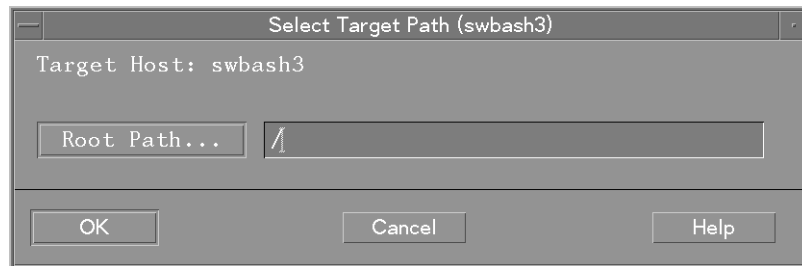
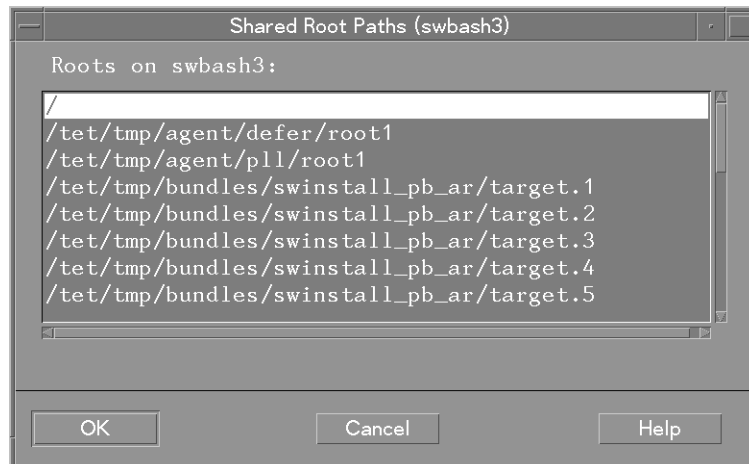


그림 1-13

### Shared Root Paths 대화 상자



## 도움말 표시—Help 메뉴

모든 GUI 및 TUI 프로그램에는 온라인 도움말 시스템이 있습니다. 각 화면, 대화 상자 또는 메뉴 선택 항목에는 해당 작업을 설명하는 연관된 도움말 설명이 있습니다.

그림 1-14

### 일반적인 온라인 도움말 화면



다양한 창과 메뉴에서 선택한 개별 메뉴, 필드, 옵션 또는 단추에 대한 상황에 맞는 도움말을 얻으려면 항목에 커서를 놓은 다음 키보드에서 **F1** 키(TUI에서 **Ctrl-F**)를 누릅니다. 이것은 해당 항목에 대한 특정 도움말을 표시합니다.

각 주요 화면에 대한 개요 정보를 표시하거나 키보드 사용법에 대한 도움말을 표시하거나 다른 제품 정보를 보려면 메뉴 표시줄에서 **Help** 메뉴를 선택합니다.

### Overview...

이 메뉴 항목은 현재 활성화되어 있는 SD-UX 화면에 대한 정보를 제공합니다. 여기에는 해당 화면에서 할 수 있는 작업 목록과 화면의 다른 영역에 대한 짧은 설명 및 관련 주제에 대한 링크가 포함됩니다.

### Keyboard...

이 메뉴 항목은 선택, 메뉴 표시줄 활성화 및 이동, 대화 상자 이동 등과 같은 주제를 포함하여 응용 프로그램을 제어하기 위해 키보드를 사용하는 방법에 대한 도움말을 불러 옵니다.

### Using Help...

이 메뉴 항목에서는 도움말 시스템 사용 방법에 대한 정보를 표시합니다.

### Product Information...

이 메뉴 항목은 SD-UX의 저작권 및 개정본 정보를 표시합니다.

## XToolkit 옵션 및 디스플레이 글꼴 변경

GUI 명령은 다음과 같은 HP-UX Xtoolkit 명령줄 옵션의 일부를 지원합니다.

- -bg 또는 -background
- -fg 또는 -foreground
- -display
- -name
- -xrm

SD-UX 명령은 디스플레이 글꼴을 변경하는 데 사용하는 XToolkit -fn 또는 -font 옵션을 지원하지 않습니다.

그러나 SD-UX 명령은 X11/Motif 환경에서 실행될 때 대부분의 Motif™ 표준 리소스 및 다음과 같은 추가 리소스를 인식합니다.

\*systemFont      가변 너비 글꼴이 적용되는 GUI 메뉴 표시줄과 다른 영역에 사용되는 가변 너비 글꼴을 지정합니다. 기본 크기는 8x13입니다.

`*userFont`        다른 모든 GUI 디스플레이에 사용되는 고정 너비 글꼴을 지정합니다. 이 글꼴은 `*systemFont`와 동일한 기본 크기여야 하며 너비는 고정된 스타일이어야 합니다. 기본 크기는 **8x13**입니다.

다음은 고정 너비 글꼴 크기를 **8x13**에서 **6x13**으로 변경하는 방법의 예입니다.

```
swinstall -xrm 'Swinstall*userFont: user6x13'
```

다음은 가변 너비 글꼴 스타일을 **12** 포인트 **HP Roman 8**로 변경하는 방법을 보여줍니다.

```
swinstall -xrm 'Swinstall*systemFont: \  
-adobe-courier-medium-r-normal12-120-75-75-m-70-hp-\  
roman8'
```

그래픽 사용자 인터페이스를 사용하여 각 명령에 대한 기본값 파일 (`/usr/lib/X11/app-defaults`에 있음)을 수정하여 특정 명령을 실행할 때마다 리소스가 설정되도록 할 수 있습니다. 다음은 `swremove`에 대한 응용 프로그램 기본값 파일의 예입니다.

#### **swremove app-defaults**

```
Swremove*foreground: red  
Swremove*background: white  
Swremove*userFont:    hp8.8x16b  
Swremove*systemFont: -adobe-courier-medium-r-normal12-120-75-75-m  
-70-hp-roman8
```





## 소프트웨어 선택 사항

소프트웨어 선택 사항을 통해 소프트웨어를 매우 세부적으로 지정할 수 있습니다. 입력 파일을 사용하여 소프트웨어를 지정할 수도 있습니다.

### 구문

*software\_selections* 구문은 이 구문이 필요한 모든 **SD-UX** 명령에 대해 동일합니다.

```
bundle[.product[.subproduct][.fileset]][,version]  
product[.subproduct][.fileset][,version]
```

- =(같음) 관계 연산자를 통해 다음과 같은 셸 와일드카드와 패턴 일치 표기를 사용하여 선택 사항을 지정할 수 있습니다.

[ ]	대괄호 — 표현식을 그룹으로 묶음
*	별표 — 여러 문자에 대한 와일드카드
?	물음표 — 문자 하나에 대한 와일드카드

예를 들어 다음 표현식은 `man`으로 끝나는 태그를 가진 모든 번들 및 제품을 설치합니다.

```
swinstall -s sw_server \*man
```

- 번들과 제품은 재귀적입니다. 번들 안에 다른 번들이 있을 수 있습니다. 예를 들면 다음과 같습니다.

```
swinstall bun1.bun2.prod.sub1.fset,r=1.0
```

또는(표현식을 사용하여):

```
swinstall bun[12].bun?.prod.sub*,a=HP-UX
```

- `\*` 소프트웨어 사양은 모든 제품을 선택합니다.

---

### 주의

데이터 손실을 방지하기 위해 `\*` 사양은 신중하게 사용해야 합니다(예를 들면 루트 디렉토리 `/`에서 소프트웨어를 제거하는 경우).

버전 구성 요소는 다음과 같은 형식을 가집니다.

```
[, r <op> revision][, a <op> arch][, v <op> vendor]  
[, c <op> category][, q=qualifier][, l=location]  
[, fr <op> revision][, fa <op> arch]
```

여기서 각 항목에 대한 설명은 다음과 같습니다.

- 정규화된 소프트웨어 사양에는 빈 문자열이 들어 있더라도 r=, a= 및 v= 버전 구성 요소가 포함됩니다. 설치된 소프트웨어의 경우 l=도 필요합니다.
- 모든 버전 구성 요소는 단일 사양 내에서 반복이 가능합니다(예: **r>=A.12, r<A.20**). 구성 요소가 여러 개 사용되는 경우 선택 사항은 모든 구성 요소와 일치해야 합니다.
- <op>(관계 연산자) 구성 요소는 점으로 구분되는 필드에서 개별적인 비교 연산을 수행하고 다음과 같은 형식을 갖습니다.

=, ==, >=, <=, <, > 또는 !=

예를 들어 **r>=B.11.11**은 B.11.11과 같거나 이보다 높은 버전을 모두 선택합니다. 시스템은 각각의 점으로 구분되는 필드를 비교하여 일치하는 것을 찾습니다.

- =(같음) 관계 연산자를 통해 [ ], \*, ? 및 !와 같은 셸 와일드카드와 패턴 일치 표기를 사용하여 선택 사항을 지정할 수 있습니다.

예를 들어 **r=1[01].\*** 표현식은 버전 10과 버전 11의 새 버전을 반환합니다.

- 소프트웨어 선택에서 공백이나 탭 문자는 허용되지 않습니다.
- *qualifier*는 소프트웨어 사양을 필터링하는 데 도움이 되도록 제품이나 번들에 첨부될 수 있는 문자열입니다.
- *location*은 설치된 소프트웨어에만 적용되며 기본 제품 디렉토리가 아닌 위치에 설치된 소프트웨어를 참조합니다.
- *fr* 및 *fa*는 파일 세트에만 적용됩니다.
- 소프트웨어 *instance\_id*는 버전 구성 요소를 대신할 수 있습니다. 형식은 다음과 같습니다.

```
[instance_id]
```

내보내진 카탈로그 컨텍스트 내에서 **instance\_id**는 동일한 태그를 갖는 제품과 번들의 버전을 구분하는 정수입니다.

## 소프트웨어 파일

소프트웨어 선택 입력 파일을 사용하여 긴 소프트웨어 제품 목록을 지정함으로써 명령줄을 더 짧게 만들 수 있습니다. 소프트웨어 선택 파일을 사용하는 경우 파일 이름을 하나만 지정해야 합니다.

-f 명령줄 옵션을 사용하면 소프트웨어 선택 파일을 지정할 수 있습니다. 예를 들면 다음과 같습니다.

```
swinstall -f mysoft -s /mnt/cd @ targetB
```

이 예에서 `mysoft` 파일(소프트웨어 파일의 현재 작업 디렉토리에 있음)에 `/mnt/cd` 저장소에 대한 소프트웨어 선택 사항 목록이 들어 있습니다.

소프트웨어 파일에서 공백 줄과 주석(#으로 시작되는 줄)은 무시됩니다. 각 소프트웨어 선택 사항은 각각 다른 줄에 지정해야 합니다.

## 대상 선택 사항

대상 선택 사항은 소프트웨어 및 소스 저장소 선택 사항을 따릅니다. 대상 선택 사항을 지정하지 않으면 작업이 수행될 대상이 로컬 호스트 상의 루트(/) 디렉토리로 간주됩니다. 그러므로 로컬 호스트 또는 기본 저장소 디렉토리에서 작업 중인 경우에는 @ 기호와 `[host][:]/[directory]` 지정(아래에 설명됨)을 사용할 필요가 없습니다.

## 구문

`target_selections` 구문은 이 구문이 필요한 모든 SD-UX 명령에 대해 동일합니다.

```
@ [host][:]/[directory]
```

- @ 문자는 로컬 호스트와 기본 디렉토리를 사용하는 경우 선택입니다. @ 문자를 사용하는 경우 피연산자와 대상 간의 구분 문자 역할을 합니다.
- @ 문자는 단 하나만 필요합니다.
- 호스트 이름, 도메인 이름 또는 인터넷 주소 등으로 호스트를 지정할 수 있습니다. 디렉토리는 절대 경로로 지정해야 합니다.
- :(콜론)은 호스트와 디렉토리를 모두 지정하는 경우에 필요합니다.
- 일부 시스템에서 @ 문자는 강제 종료(kill) 기능으로 사용됩니다. @ 문자가 시스템상의 다른 기능에 매핑되어 있는지 알아보려면 시스템에서 `stty`를 입력합니다. 다른 기능에 매핑되어 있는 경우 해당 매핑을 제거하거나, 변경하거나 또는 \@를 사용합니다.

## 대상 파일

대상 선택 입력 파일을 사용하여 긴 대상 목록을 지정함으로써 명령줄을 더 짧게 만들 수 있습니다. 대상 선택 파일을 사용하는 경우 파일 이름을 하나만 지정해야 합니다.

-t 명령줄 옵션을 사용하면 대상 파일을 지정할 수 있습니다. 예를 들면 다음과 같습니다.

```
swinstall -f mysoft -s /mnt/cd -t mytargs
```

이 예에서 **mytargs** 파일(현재 작업 디렉토리에 있음)에 **swinstall** 명령에 대한 대상 선택 사항 목록이 들어 있습니다.

대상 파일에서 공백 줄과 주석(#으로 시작되는 줄)은 무시됩니다. 각 선택 사항은 각기 다른 줄에 지정해야 하고 호스트 이름이나 네트워크 주소로 구성되어야 하며 그 다음에 선택적으로 콜론과 전체 경로가 나올 수 있습니다(*host[:/directory]*).

## 명령 옵션 사용

적절한 명령 옵션을 설정하여 많은 **SD-UX** 명령 정책과 동작을 제어할 수 있습니다. 명령줄에 직접 지정한 사전 정의된 파일이나 값을 사용하여 옵션 기본값을 변경할 수 있습니다. 파일을 사용하여 기본값을 변경하면 명령을 호출할 때마다 명령 작동 방식을 지정하고 싶지 않을 때 도움이 됩니다.

이 규칙은 기본값이 작동하는 방식을 제어합니다.

1. /var/adm/sw/defaults에 있는 옵션은 해당 시스템의 모든 **SD-UX** 명령에 영향을 줍니다. 이 파일은 옵션이 적용되는 모든 명령 또는 특정 명령에 한해 기본 작동 방식을 변경할 수 있습니다.
2. 개인 \$HOME/.swdefaults 파일에 있는 옵션은 사용자에게만 영향을 주며 전체 시스템에는 영향을 주지 않습니다.
3. 세션 파일에서 읽은 옵션은 해당 세션에만 영향을 미칩니다.
4. -X *option\_file* 또는 -x *option=value* 인수를 사용하여 명령줄에서 변경된 옵션은 시스템 전체 및 사용자 개인의 옵션 파일에 우선하지만 해당 명령 호출 시에만 유효합니다.

시스템 전체에 적용되는 정책을 설정하려면 /var/adm/sw/defaults 파일을 사용합니다. 그러나 사용자의 \$HOME/.swdefaults 파일, 세션 파일 또는 명령줄 변경을 통해 이러한 옵션을 대체할 수 있음을 명심하시기 바랍니다.

/usr/lib/sw/sys.defaults 템플릿 파일은 시스템 전체 또는 개인 옵션 파일을 쉽게 변경할 수 있는 방법을 제공합니다.

템플릿 파일에는 다음과 같은 내용이 들어 있습니다(주석으로).

- 모든 명령 옵션
- 각 옵션이 적용되는 명령
- 각 옵션에 가능한 값
- 각 값이 적용되었을 때 그 결과 나타나는 시스템 작동 방식

이 파일의 값을 시스템 기본 파일(/var/adm/sw/defaults), 개인 기본 파일(\$HOME/.swdefaults) 또는 입력 파일(-X *input\_file* 옵션 사용)에 복사하고 SD-UX 동작에 영향을 주도록 편집할 수 있습니다.

옵션 파일은 다음과 같은 구문을 사용합니다.

`[command.]option=value`

- 선택적 *command*는 SD-UX 명령의 이름입니다. 명령 이름을 지정하면 해당 명령의 기본 작동만 변경됩니다. 명령 이름 다음에 점(.)이 있어야 합니다.
- *option*은 기본 옵션의 이름입니다. 옵션 이름 다음에 = 기호가 있어야 합니다.
- *value*는 해당 옵션에 사용할 수 있는 값 중 하나입니다.

---

## 참고

swagentd 옵션을 변경한 뒤에 반드시 SD-UX 데몬을 다시 시작해야 하며, 그렇지 않으면 데몬이 변경 사항을 인식하지 못합니다. 데몬을 다시 시작하려면 다음과 같이 입력합니다.

```
/usr/sbin/swagentd -r
```

---

## 예

use\_alternate\_source의 기본값을 이 옵션이 적용되는 모든 명령, 모든 차후 세션 및 모든 사용자에게 대해 true로 변경하려면 다음 줄을 /var/adm/sw/defaults 파일에 넣습니다.

```
use_alternate_source=true
```

사용자가 명령을 호출할 때 use\_alternate\_source의 기본값을 false로 변경하려면 다음 줄을 사용자의 \$HOME/.swdefaults 파일에 넣습니다.

```
swinstall.use_alternate_source=false
```

`my_install_defaults`에 저장되어 있는 옵션으로 대화식 `swinstall` 세션을 시작하여 시스템 차원의 값이나 개인적인 기본 파일 값을 대체하려면 다음과 같이 입력합니다.

```
swinstall -i -X my_install_defaults=true
```

대화식 설치 세션을 시작하고 이 세션에 대해서만 `use_alternate_source` 기본값을 다시 설정하려면 다음과 같이 입력합니다.

```
swinstall -i -x use_alternate_source
```

기본값과 해당 값 및 설명에 대한 자세한 내용은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.

---

## 주의

명령 옵션에 대한 기본값을 변경하는 경우 부적절한 값을 지정하면 부정적인 결과가 발생할 수도 있습니다.

---

## 세션 파일

SD-UX 작업이 시작되기 전에 시스템은 현재 명령 옵션, 소스 정보, 소프트웨어 선택 사항, 대상 선택 사항 등을 **세션 파일**에 자동으로 저장합니다. 그런 다음 이 명령이 실패하더라도 나중에 이 세션 정보를 다시 사용할 수 있습니다.

세션 정보는 `$HOME/.sw/sessions/` 디렉토리에 `command.last`의 형태로 저장되며, 여기서 `command`는 명령 이름입니다. 세션 파일을 저장할 때마다 이전에 저장했던 내용을 덮어씁니다. (명령을 호출한 다음 각 세션 파일 이름을 다시 지정하여 여러 개의 세션 파일을 저장할 수 있습니다.)

자동으로 저장된 세션 파일을 다시 사용하려면 `-S swcommand.last` 인수를 지정하여 명령을 호출합니다. 예를 들면 다음과 같습니다.

```
swinstall -S swinstall.last
```

세션 파일을 기본 세션 디렉토리가 아닌 다른 위치에 저장하려면 `-C session_file` 인수를 사용하고 저장하려는 파일의 절대 경로를 입력합니다. 디렉토리를 지정하지 않으면 세션 파일의 기본 위치는 `$HOME/.sw/sessions/`입니다.

명령줄에서 세션을 다시 실행하려면 `-S session_file` 옵션에 세션 파일을 인수로 지정합니다.

세션 파일을 다시 실행할 때 세션 파일의 값이 시스템 기본 파일이나 개인적인 기본값 파일에 우선합니다. 마찬가지로 명령을 호출할 때 지정한 명령줄 옵션이나 매개 변수가 세션 파일의 값에 우선합니다.

다음은 세션 파일의 예입니다. 여기에서는 기본값 파일과 동일한 구문을 사용합니다.

```
# swinstall session file
#
# Filename      /users/fred/.sw/sessions/swinstall.last
# Date saved    05/26/01 15:59:41 MDT
swinstall.allow_downdate = true
swinstall.allow_incompatible = false
swinstall.allow_multiple_versions = false
swinstall.autoreboot = false
swinstall.autorecover_product = false
swinstall.compress_files = false
swinstall.create_target_path = true
...
```

(일반적으로 swinstall 세션 파일은 약 70줄 정도 됩니다.)



## 2 소프트웨어 설치

이 장에서는 `swinstall`, `swconfig` 및 `swverify` 명령을 사용하여 소프트웨어를 설치, 구성 및 검증하는 방법에 대해 설명합니다.

- `swinstall`은 저장소로부터 소프트웨어를 설치하고 소프트웨어의 자동 구성을 수행합니다.
- `swconfig`는 이전에 설치한 소프트웨어를 구성, 구성 해제 또는 재구성하는 데 사용합니다.
- `swverify`는 소프트웨어가 올바르게 설치되었는지 확인하고 추가적인 확인 작업을 수행하거나 특정 문제를 해결하는 데 사용합니다.

표 2-1

### 이 장의 항목

설명하는 항목
66페이지의 “ <code>swinstall</code> 을 사용한 설치”
84페이지의 “설치 구성( <code>swconfig</code> )”
91페이지의 “설치 확인( <code>swverify</code> )”

## swinstall을 사용한 설치

swinstall 명령은 소프트웨어 소스(저장소나 물리적 미디어)로부터 로컬 호스트로 소프트웨어를 설치합니다.

### 기능 및 제한 사항

- 선택적 GUI
- 설치된 시스템에서 소프트웨어가 실행되도록 하는 호환성 필터링
- 커널 재구축 또는 재부팅을 수행하는 기능
- 종속성 자동 사용을 통해 작동될 소프트웨어를 자동으로 선택(직접 지정한 소프트웨어에 추가하여)
- 설치 중 다음과 같은 제어 스크립트를 실행하는 기능

<b>Checkinstall</b>	설치 및 구성이 실행될 수 있는지를 판별하기 위해 각 대상을 분석합니다
<b>Preinstall</b>	소프트웨어 파일을 설치하기 전에 오래된 파일 제거와 같은 파일 작업을 수행합니다.
<b>Request</b>	설치 또는 구성 프로세스의 일부로 사용자로부터의 대화식 응답을 요구합니다(swask, swconfig 및 swinstall 명령으로 실행합니다).
<b>Configure</b>	설치되어 있는 파일 세트나 제품을 구성합니다(84페이지의 “설치 구성(swconfig)”을 참조하십시오).
<b>Postinstall</b>	파일 세트나 제품이 설치된 직후 기본 파일 재설정과 같은 추가적인 설치 작업을 수행합니다.
<b>Unpostinstall</b>	설치 프로세스 중에 swinstall이 복구 작업을 시작해야 하는 경우 설치 후 스크립트를 취소합니다.
<b>Unpreinstall</b>	설치 프로세스 중에 SD가 복구를 시작해야 하는 경우 설치 전 스크립트를 취소합니다

(11장, 347페이지의 “제어 스크립트 사용” 참조).

- 루트 디렉토리를 대체하기 위해 소프트웨어 설치 가능

## GUI를 사용한 설치

### 개요

이 절에서는 swinstall GUI에 대한 개요를 설명합니다.

- 일반적으로 75페이지의 “명령줄에서 설치”의 모든 정보는 swinstall GUI에도 적용됩니다.
- 이 절에서는 37페이지의 “GUI 및 TUI 명령 사용”에 설명된 표준 GUI 요소에 대한 추가 정보도 참조합니다.
- 이 절의 모든 정보는 달리 명시되지 않은 한 TUI 프로그램에도 적용됩니다. 자세한 내용은 37페이지의 “터미널 사용자 인터페이스”를 참조하십시오.

GUI 설치 프로세스에는 다섯 단계가 있습니다.

표 2-2

### GUI 설치 단계

I. 시작	swinstall GUI를 시작합니다.
II. 소스 선택	설치될 소프트웨어의 소프트웨어 저장소 위치를 제공합니다.
III. 소프트웨어 선택	설치할 소프트웨어를 선택합니다.
IV. 분석(미리 보기)	선택한 소프트웨어가 정상적으로 설치될 수 있는지 판별하기 위해 설치를 분석(미리 보기)합니다.
V. 설치	실제 소프트웨어 설치를 수행합니다.

### 단계 I: 시작

설치 세션을 위해 GUI나 TUI를 시작하려면 다음과 같이 입력합니다.

```
/usr/sbin/swinstall
```

명령줄에서도 소프트웨어를 지정하지 **않은 한 GUI**가 자동으로 호출됩니다. GUI를 호출하고 소프트웨어를 지정하려면 **-i** 옵션을 지정합니다. 예를 들어 GUI를 사용하여 BUNDLE1에 대한 미리 보기(분석만 수행) 세션을 열려면 다음과 같이 입력합니다.

```
swinstall -i -p /MyDepot/BUNDLE1
```

Software Selection 창 위에 Specify Source 대화 상자가 겹쳐져 표시됩니다.

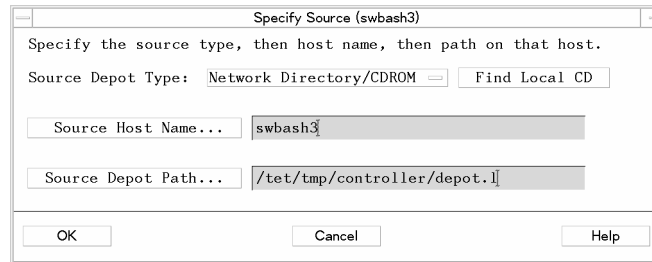
## 단계 II: 소스 선택

이 단계에서는 설치하려는 소프트웨어가 들어 있는 소스 저장소를 지정해야 합니다. Specify Source 대화 상자(그림 2-1, “Specify Source 대화 상자”)에 자동으로 로컬 호스트 및 기본 저장소 경로가 나열됩니다.

(GUI를 호출할 때 `-s source` 옵션을 지정한 경우 이 단계는 생략됩니다.)

그림 2-1

### Specify Source 대화 상자



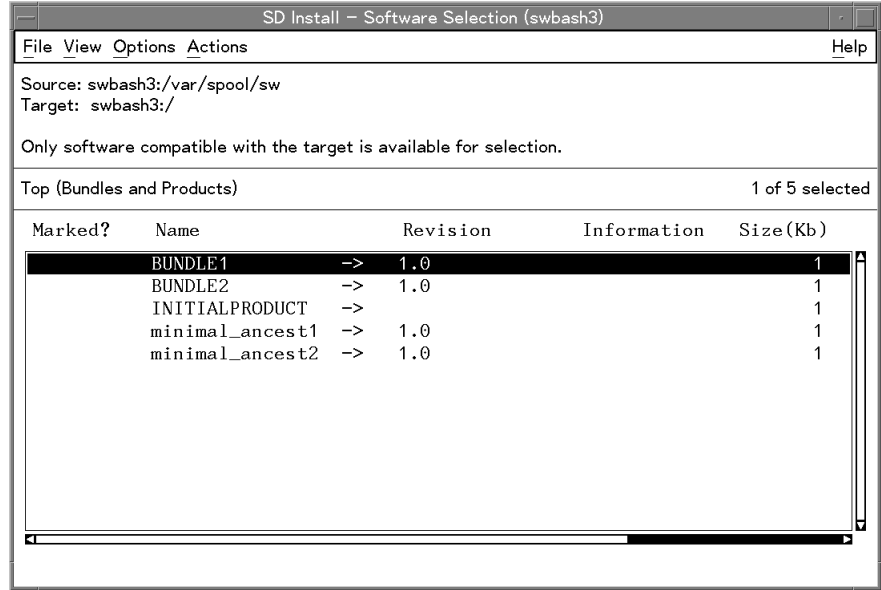
- (선택) 다른 호스트 시스템을 지정하려면 소스 호스트 이름을 입력하거나 다음을 수행합니다.
  - Source Host Name** 단추를 클릭합니다. 대화 상자가 표시되어 `defaults.hosts` 파일(`$HOME/.sw/defaults.hosts` 또는 `/var/adm/sw/defaults.hosts`)에 들어 있는 모든 호스트 시스템 이름이 나열됩니다.
  - 목록에서 호스트 이름을 선택합니다.
  - OK**를 클릭합니다. 호스트 이름이 Specify Source 대화 상자의 해당 상자에 표시됩니다.
- (선택) 저장소에 대한 경로를 지정하려면 새로운 경로를 입력하거나 다음을 수행합니다.
  - Source Depot Path** 단추를 클릭하여 소스 호스트에 등록되어 있는 저장소 목록을 표시합니다.
  - 저장소 중 하나를 강조 표시합니다.
  - OK**를 클릭하면 Specify Source 대화 상자에 표시됩니다.
- OK**를 클릭합니다. Specify Source 대화 상자가 닫히고 지정한 저장소에 들어 있는 소프트웨어가 Software Selection 창에 표시됩니다.

**단계 III: 소프트웨어  
선택**

이 단계에서는 Software Selection 창에서 설치하려는 소프트웨어를 선택합니다.

그림 2-2

**swinstall Software Selection 창**



1. 객체 목록에서 소프트웨어를 선택합니다.

- a. 항목을 강조 표시합니다.
- b. **Actions** → **Mark For Install**을 선택합니다.

— 또는 —

마우스 오른쪽 단추를 클릭하여 팝업 메뉴를 표시한 다음 **Mark For Install**을 선택합니다.

객체 목록의 Marked? 플래그가 선택한 내용과 일치하게 Yes로 변경됩니다. (소프트웨어 객체의 구성 요소만을 선택하거나 종속성으로 인해 그러한 구성 요소가 자동으로 선택된 경우 Partial 플래그가 표시될 수 있습니다.)

**참고**

같은 저장소에 한 제품의 여러 버전이 들어 있는 경우 SD-UX에서는 각 설치 세션 중에 단 하나의 버전만을 선택해야 합니다.

2. (선택) **Actions** 메뉴의 선택 사항을 선택합니다.

- **Match What Target Has**는 현재의 IPD(Installed Product Database)를 조사하여 기존 파일 세트를 설치하려는 (동일한 이름의) 새 파일 세트와 일치시킵니다. 이 기능은 같은 소프트웨어의 새로운 버전으로 업데이트할 때 가장 유용합니다. 이 옵션은 **Options Editor**에서 설정할 수 있습니다.
  - **Add Software Group**에서는 이전에 저장된 소프트웨어 그룹 파일 목록을 표시하고 디렉토리를 지정할 수 있습니다. 파일을 선택하면 **Software Selection** 창에서 이미 선택한 내용에 파일에서의 소프트웨어 선택을 추가하게 됩니다.
  - **Save Software Group**을 통해 현재 표시된 소프트웨어 목록을 그룹으로 저장할 수 있습니다.
  - **Manage Patch Selections**를 통해 패치 목록에서 설치할 패치를 선택하고 패치용 필터를 선택하고 다른 패치 옵션을 설정할 수 있습니다. (자세한 내용은 79페이지의 “패치 설치”를 참조하십시오.)
  - **Change Source...**를 클릭하면 소프트웨어 선택 사항을 취소하고 **Specify Source** 대화 상자로 돌아갑니다.
  - **Add New Codeword**를 통해 보호된 소프트웨어의 잠금을 해제하기 위한 새로운 코드워드를 추가할 수 있습니다. (이 옵션은 **SD-UX**가 소스에 보호된 소프트웨어가 포함되어 있다는 것을 검색한 경우에만 사용할 수 있습니다.)
  - **Change Target...**은 **Select Target Path** 대화 상자를 표시합니다. 변동 가능한 제품에 대해 대체 루트를 지정할 수 있습니다.
  - **Show Description of Software**(객체 목록에서 강조 표시된 항목 하나에 대해서만 사용 가능)는 선택한 소프트웨어에 대해 자세한 정보를 표시합니다.
3. 분석(미리 보기) 단계를 시작하려면 **Actions**→**Install**을 선택합니다. **Analysis** 대화 상자가 표시됩니다.

## 단계 IV: 분석 (미리 보기)

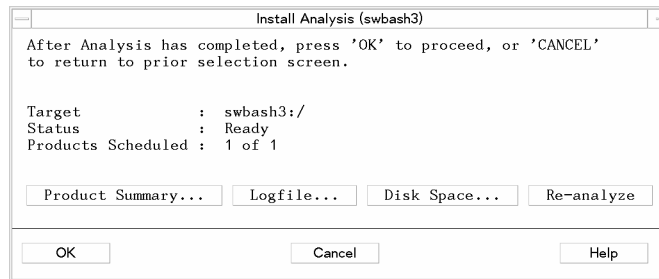
이 단계에서는 선택한 소프트웨어를 SD-UX가 분석합니다.

**Analysis** 창에 분석 프로세스에 대한 상태 정보가 표시됩니다. 분석이 완료되고 호스트 상태가 Ready가 되면 **OK**를 클릭하여 실제 설치를 시작합니다(74페이지의 “단계 V: 설치” 참조). **Analysis** 대화 상자가 **Install** 대화 상자로 바뀝니다.

미리 보기 세션을 시작했으면 분석 후 설치가 중단됩니다. **OK**를 클릭하면 **Software Selection** 창으로 돌아갑니다.

그림 2-3

### Analysis 대화 상자



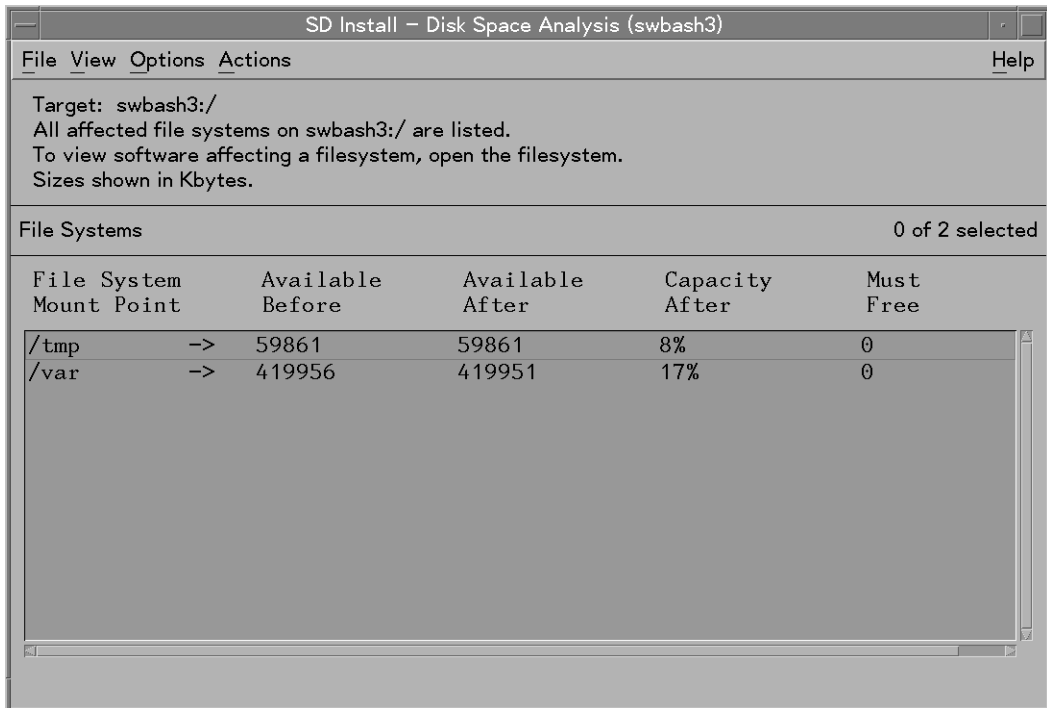
다음 작업을 수행할 수 있습니다.

- **Product Summary**에서는 제품이나 번들에 대한 추가 정보를 제공하며 종속성, 저작권, 공급업체 등에 대한 추가 정보를 표시하는 **Product Description** 단추가 있습니다.
- **Logfile**에서는 로그 파일에 기록된 자세한 설치 정보를 여러 화면에 걸쳐 보여줍니다.
- **Disk Space**는 **Disk Space Analysis** 창(그림 2-4, “Disk Space Analysis 창”)을 표시하며 이 창에는 다음과 같은 내용이 포함됩니다.
  - 파일 시스템 마운트 지점
  - 설치 전에 사용할 수 있었던 디스크 공간의 양
  - 설치 후 사용 가능한 디스크 공간의 양
  - 사용될 디스크 용량의 비율
  - 작업을 완료하기 위해 확보되어야 할 디스크 공간의 양

이 창의 메뉴를 통해 다음과 같은 작업을 수행할 수 있습니다.

- 객체 목록 검색
- 특정 파일 세트에 대해 예상되는 용량 요구 사항을 보기 위한 항목 열기
- **Re-analyze**는 분석 프로세스를 반복합니다.

그림 2-4 **Disk Space Analysis 창**





분석이 완료되면 모든 호스트의 상태는 Ready 또는 Excluded from task 중 하나로 표시됩니다. 선택한 소프트웨어 중 **하나라도** 호스트에 설치될 수 있으면 상태는 Ready로 표시됩니다. 선택한 소프트웨어 중 **아무 것도** 호스트에 설치될 수 없으면 상태는 Excluded from task로 표시됩니다.

다음은 상태 결과에 대한 요약입니다. **Logfile** 단추를 클릭하면 대부분의 문제에 대한 자세한 내용을 찾을 수 있습니다.

Ready                    분석이 수행되는 동안 어떠한 오류나 경고도 발생하지 않았습  
다. 아무런 문제 없이 설치가 진행됩니다.

Ready with Warnings                    분석이 수행되는 동안 경고가 생성되었습니다. 오류와 경고가 로  
그 파일에 기록됩니다.

Ready with Errors.                    선택된 하나 이상의 제품이 설치되거나 복사됩니다. 그러나 선택  
된 제품 중 하나 이상이 분석 오류로 인해 작업에서 제외됩니다.  
오류와 경고가 로그 파일에 기록됩니다.

Communication failure                    대상 또는 소스와의 접속 또는 통신이 유실되었습니다.

Excluded due to errors                    일종의 글로벌 오류가 발생했습니다. 예를 들면 시스템이 파일 시  
스템을 마운트할 수 없는 경우일 수 있습니다.

Disk Space Failure                    설치 시 디스크 저장소에서 사용할 수 있는 공간을 초과했습니다.  
자세한 내용은 **Disk Space** 단추를 클릭하십시오.

Products Scheduled 행에서는 선택한 모든 제품 중 설치할 준비가 된 제품의 수를 보여줍니다. 여기에는 다음과 같은 항목이 포함됩니다.

- 단지 종속 관계로 인해 선택된 제품
- 부분적으로 선택된 제품
- 기타 선택된 제품 및 번들

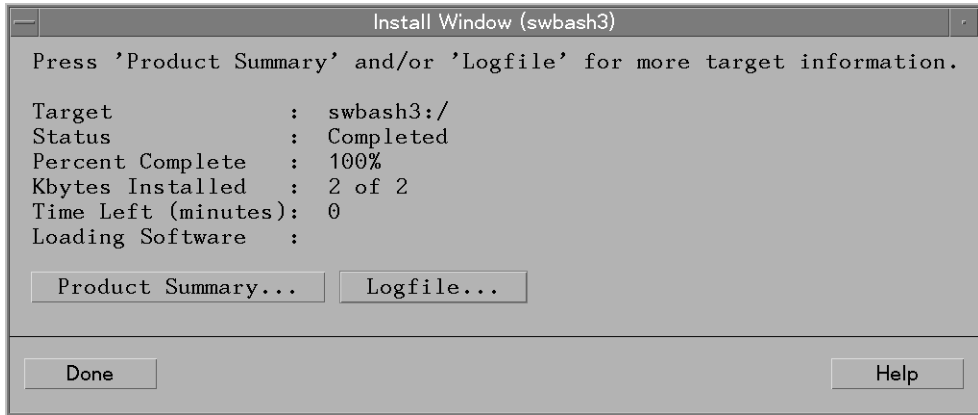
### 단계 V: 설치

이 단계에서는 SD-UX가 실제 설치를 수행합니다.

Analysis 창에서 **OK**를 클릭하면 SD-UX는 설치를 시작하고 상태 정보를 보여 주는 Install 창을 표시합니다.

그림 2-5

설치 창



다음과 같은 작업 단추를 사용할 수 있습니다.

- **Done**을 클릭하면 **Software Selection** 창으로 돌아갑니다. 그러면 다른 설치를 시작하거나 GUI를 종료(**File**→ **Exit**)할 수 있습니다.
- **Product Summary**는 설치 및 제품 정보(이름, 버전, 설치 결과, 설치 요약)를 표시합니다.
- **Logfile**을 클릭하면 로그 파일을 표시합니다.
- (커널 설치에서만 표시됨) **Resume**은 일시 중단되었던 설치를 다시 시작합니다. 이를 통해 문제를 해결한 후 작업을 진행할 수 있습니다.
- (커널 설치에서만 표시됨) **Abort**는 일시 중단되었던 설치를 취소합니다.

다음과 같은 경우에 설치가 일시적으로 중단됩니다.

- 파일 로드 오류
- 스크립트에 오류 발생
- 커널 관련 파일 세트의 사용자 정의 오류
- 커널 구축 오류
- 테이프 변경이 필요한 경우(다중 테이프 미디어에서 설치할 때)

## 명령줄에서 설치

### Swinstall 구문

swinstall 구문은 다음과 같습니다.

```
swinstall [XToolkit Options] [-i] [-p] [-r] [-v] [-c catalog]
[-C session_file] [-f software_file] [-Q date] [-s source]
[-S session_file] [-t target_file] [-x option=value]
[-X option_file] [software_selections] [@ target_selections]
```

### 옵션 및 피연산자

XToolkit 옵션	
-i	GUI용 X window 옵션입니다. 자세한 내용은 55페이지의 “XToolkit 옵션 및 디스플레이 글꼴 변경”을 참조하십시오.
-p	GUI나 TUI를 호출하여 대화식 모드에서 명령을 실행합니다. 자세한 내용은 67페이지의 “GUI를 사용한 설치”를 참조하십시오.
-r	설치 작업을 미리 봅니다(분석만 수행).
-v	대체 루트 디렉토리에서 작동합니다. 자세한 내용은 82페이지의 “대체 루트에 설치”를 참조하십시오.
-c catalog	stdout으로의 상세 출력을 활성화하고 모든 활동을 화면에 표시합니다.
-C session_file	카탈로그의 응답 파일 또는 요청 스크립트에 의해 작성된 파일의 사본을 저장합니다. 자세한 내용은 385페이지의 “사용자 응답 요구 (swask)”를 참조하십시오.
-f software_file	다른 세션에서 다시 사용하기 위해 현재 옵션 및 피연산자 값을 session_file에 저장합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.
-Q date	명령줄에서 지정한 소프트웨어 대신(또는 그에 추가하여) software_file에서 소프트웨어 선택 사항을 읽습니다. 자세한 내용은 60페이지의 “소프트웨어 파일”을 참조하십시오.
-s source	원격 작업을 수행하는 경우 해당 날짜에 작업이 실행되도록 일정을 예약합니다. 211페이지의 “명령줄에서 작업 예약” 및 6장, 167페이지의 “원격 작업 개요”를 참조하십시오.
	기본값인 /var/spool/sw 대신 source를 통해 지정한 소프트웨어 소스를 사용합니다. 구문은 다음과 같습니다.
	[host:][/directory]

*host*는 호스트 이름, 도메인 이름 또는 인터넷 주소(예: 15.1.48.23)가 될 수 있습니다. *directory*는 절대 경로입니다.

*-S session\_file*

이전 설치 세션에서 저장되어 *session\_file*에 보관되어 있는 옵션 및 피연산자를 사용합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.

*-t target\_file*

명령줄에서 지정한 대상 대신(또는 그에 추가하여) *target\_file*에서 대상 선택 사항을 읽습니다. 자세한 내용은 61페이지의 “대상 파일”을 참조하십시오.

*-x command\_option=value*

*command\_option*을 *value*로 설정하며 기본값이나 옵션 파일의 값을 대체합니다. 자세한 내용은 76페이지의 “명령 옵션 변경”을 참조하십시오.

*-X option\_file*

*option\_file*에서 세션 옵션 및 기능을 읽습니다. 자세한 내용은 76페이지의 “명령 옵션 변경”을 참조하십시오.

*software\_selections*

설치될 하나 이상의 소프트웨어 객체입니다. 자세한 내용은 58페이지의 “소프트웨어 선택 사항”을 참조하십시오.

*target\_selections*

소프트웨어 선택 사항을 설치할 대상입니다. 자세한 내용은 60페이지의 “대상 선택 사항”을 참조하십시오.

## 명령 옵션 변경

명령을 호출할 때 명령줄 옵션을 추가로 지정(-x 옵션 사용)하거나 파일에서 미리 정의된 값을 읽어서 이 명령의 작동 방식을 변경할 수 있습니다. 다음 표에서는 swinstall에 적용되는 기본값과 옵션을 보여줍니다.

표 2-3

swinstall 명령 옵션 및 기본값

- admin\_directory=/var/adm/sw
- agent\_auto\_exit=true
- agent\_timeout\_minutes=10000
- allow\_downdate=false
- allow\_incompatible=false
- allow\_multiple\_versions=false
- allow\_split\_patches=false
- ask=false
- autoreboot=false
- autorecover\_product=false
- autoremove\_job=false
- autoselect\_dependencies=true
- autoselect\_patches=true
- autoselect\_reference\_bundles=true
- codeword=
- compress\_index=false
- controller\_source=
- create\_target\_path=true
- customer\_id=
- defer\_configure=false
- distribution\_source\_directory=  
/var/spool/sw
- enforce\_dependencies=true
- enforce\_dsa=true
- enforce\_kernbld\_failure=true
- enforce\_scripts=true
- installed\_software\_catalog=products
- job\_title=
- layout\_version=1.0
- log\_msgid=0
- logdetail=false
- logfile=/var/adm/sw/swinstall.log
- loglevel=1
- match\_target=false
- max\_targets=25
- mount\_all\_filesystems=true
- os\_name
- os\_release
- patch\_filter=  
software\_specification
- patch\_match\_target=false
- patch\_save\_files=true
- polling\_interval=2
- preview=false
- register\_new\_root=true
- reinstall=false
- reinstall\_files=false
- reinstall\_files\_use\_cksum=true
- retry\_rpc=1
- retry\_rpc\_interval=
- reuse\_short\_job\_numbers=true
- rpc\_binding\_info=  
ncacn\_ip\_tcp:[2121]  
ncadg\_ip\_udp:[2121]
- rpc\_timeout=5
- run\_as\_superuser=true
- select\_local=true
- software=
- software\_view=all\_bundles
- source=
- source\_cdrom=/SD\_CDROM
- source\_tape=/dev/rmt/0m
- source\_type=directory
- targets=
- use\_alternate\_source=false
- verbose=1
- write\_remote\_files=true

## 추가 정보

옵션 설정 및 각 옵션에 대한 전체 목록 및 설명은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.

## 설치 작업 및 예

이 절에서는 소프트웨어 제품 설치 명령에 대해 예를 들어 설명합니다. \\*는 “모든 제품 및 파일 세트 또는 사용 가능한 모든 소프트웨어”를 나타내는 간략한 선택적 와일드카드입니다.

명령줄에서 설치 세션을 시작하려면 필요에 따라 명령 문자열에 몇 가지 옵션, 호스트 및 소스 이름, 소프트웨어 선택 사항 등을 조합하여야 합니다. 예를 들면 다음과 같습니다.

```
swinstall -p -s softsource -f softlist \  
@ myhost:/mydirectory
```

@ myhost:/mydirectory는 로컬 호스트 및 기본 디렉토리(루트)에 설치하는 경우 선택입니다.

---

## 참고

소스를 지정하지 않으면 swinstall은 로컬 호스트의 기본 저장소 디렉토리인 /var/spool/sw를 사용합니다.

- CD-ROM(/mnt/cd에 로컬로 마운트됨)의 mysoft 파일에 있는 사전 결정된 소프트웨어 제품들을 로컬 호스트의 기본 디렉토리(/)에 설치하려면 다음과 같이 입력합니다.

```
swinstall -f mysoft -s /mnt/cd
```

- server라는 이름의 호스트에 위치한 기본 저장소(/var/spool/sw)의 소프트웨어를 myhost 호스트의 기본 디렉토리로 모두 선택한 다음 실제 설치하지 않고 프로세스만을 미리 보려면(-p) 다음과 같이 입력합니다.

```
swinstall -p -s server \* @ myhost
```

소프트웨어가 server의 기본 /var/spool/sw에 위치해 있고 myhost의 /에 설치될 것으로 가정하므로 저장소 위치(/depot)는 지정하지 않습니다. -p 분석 옵션은 89페이지의 “명령 옵션 변경”에서 설명됩니다.

- sw\_server 호스트의 기본 저장소에서 C 및 Pascal이라는 제품을 모두 선택하고 대화식 GUI 세션을 시작하려면(-i) 다음과 같이 입력합니다.

```
swinstall -i -s sw_server C Pascal
```

- HP Omniback 소프트웨어(이미 로컬 호스트의 기본 디렉토리에 설치되어 있음)를 /mnt/cd에 마운트된 CD-ROM에서 새 버전으로 업데이트하려면 다음과 같이 입력합니다.

```
swinstall -s /mnt/cd Omniback
```

### HP-UX 11i로 업데이트

이전 HP-UX 릴리즈를 HP-UX 11i로 업데이트하는 자세한 지침을 보려면 **HP-UX 11i 설치 및 업데이트 설명서**의 제 2장에 설명된 대로 새로운 `update-ux` 명령을 사용합니다.

이 설명서와 OS 관련 전체 문서는 CD-ROM의 HP-UX Instant Information CD-ROM 및 다음 주소에서 구할 수 있습니다.

<http://docs.hp.com/hpux/os/11i>(영문)

<http://docs.hp.com/ko/hpux11i>(한글)

### 패치 설치

swinstall 명령에는 GUI로 된 패치 관리 대화 상자와 같은 다양한 패치 관리 기능이 있습니다. 패치 및 swinstall GUI 패치 기능 사용에 대한 자세한 내용은 5장, 165 페이지의 “HP-UX 패치 및 패치 관리”를 참조하십시오.

### 업데이트된 파일 복구

autorecover\_product 옵션을 통해 설치를 시작했거나 프로세스에 장애가 발생했을 경우 원래의 제품 파일로 자동 복구 또는 자동 롤백할 수 있습니다.

파일을 업데이트할 때 swinstall은 업데이트하면서 원래의 파일을 제거합니다.

swinstall이 새 파일 세트를 로드하는 중에 오류가 발생하면 로드되던 제품은 손상된 것으로 표시되고, 원래의 파일은 유실되어 다시 설치해야 합니다.

autorecover\_product 옵션을 `true`로 설정하면 업데이트되는 파일 세트가 모두 백업 사본으로 먼저 저장됩니다. 이 파일 세트는 제품 내 모든 파일 세트의 로드가 완료될 때까지 제거되지 않습니다. 오류가 발생하여 swinstall이 중단된 경우 오류를 수정한 다음 swinstall을 다시 실행할 수 있습니다. swinstall은 인터럽트되었던 지점에서 로드 프로세스를 자동으로 다시 시작합니다.

---

**주의**

HP-UX 제품에는 대개 사전 설치 및 사후 설치 스크립트는 있으나 취소 스크립트는 없습니다. 이로 인해 autorecover\_products 옵션을 사용할 때의 장점이 사라집니다. autorecover\_products는 연관된 취소 스크립트가 있는 소프트웨어에서만 사용하십시오.

---

**시스템을 재부팅해야 하는 소프트웨어 설치**

is\_reboot 속성이 true로 설정되어 패키징된 소프트웨어는 소프트웨어를 설치한 후 호스트를 다시 부팅해야 합니다. 그러나 대체 루트 파일 시스템에 설치하는 경우에는 호스트를 다시 부팅하지 않습니다.

로컬 설치 시 다시 부팅해야 하는 경우 시스템은 대상 및 컨트롤러를 다시 부팅하므로 성공 또는 오류를 보고하는 프로세스가 없습니다. (SD-UX는 다시 부팅된 후 대상에 자동으로 다시 연결되지 않습니다.)

해당 소프트웨어에서 로컬 호스트를 다시 부팅해야 하는지 알아 보려면 **Show Description of Software** 메뉴 항목을 사용하여 **Software Selection** 창에서 소프트웨어 설명을 참조하거나 **Product Summary** 및 **Product Description** 단추를 사용하여 **Analysis** 대화 상자에서 소프트웨어 설명을 참조하십시오.

**소프트웨어 코드워드 및 고객 ID 사용**

소프트웨어 무단 사용을 방지하기 위해 HP(및 기타 업체)는 특별한 코드워드와 고객 식별 번호를 통해 소프트웨어를 특정 소유자만 사용할 수 있도록 “잠급니다”. 이러한 코드워드와 고객 ID는 소프트웨어를 구입하거나 업데이트를 받을 때 제공됩니다. HP에서는 소프트웨어에 동봉된 소프트웨어 인증서에 이를 표시합니다.

CD-ROM에 대한 customer\_id와 코드워드를 적절하게 저장하려면 CD-ROM이 들어 있는 호스트에서 swinstall(또는 swcopy 또는 swlist)을 실행할 수 있습니다. 코드워드가 저장되면 해당 호스트에서 CD ROM을 소스로 소프트웨어를 설치하려는 클라이언트는 코드워드와 customer\_id를 다시 입력할 필요가 없습니다.

SD는 보호된 소프트웨어를 다른 호스트로 제공하려는 서버에서 .codewords 파일을 검색합니다. 유효한 customer\_id와 코드워드 쌍을 찾습니다. 이를 통해 SD는 해당 소프트웨어를 “사용하려는” 모든 호스트에서 코드워드와 customer\_id를 입력할 필요를 없애줍니다.

따라서 여러 대의 시스템을 업데이트하는 경우 시간을 절약할 수 있습니다.



SD-UX는 보호되는 소프트웨어의 설치에 앞서 이러한 코드워드 또는 번호를 묻는 메시지를 표시합니다. GUI의 **Actions** 메뉴에서 **Add New Codeword**를 선택하거나 명령 줄에서 적절한 기본 옵션(-x codeword=xxxx 및 -x customer\_id=xxx)을 사용하여 번호를 입력하거나 변경할 수 있습니다.

예를 들어 코드워드 123456789101bcdf를 저장하려 하고(/CD-ROM 마운트 지점에서) customer\_id는 xyzCorp였다면 다음과 같이 명령줄에 입력합니다.

```
swinstall -p -x customer_id=xyzCorp \  
-x codeword=123456789101bcdf \  
-s /CD_ROM
```

(이 명령의 목적은 단지 코드워드와 고객 ID를 저장하는 것이므로 -p 옵션은 명령을 미리 보기 모드에서 실행시켜 실제 소프트웨어 설치는 일어나지 않습니다.)

codeword 및 customer\_id 옵션에 대한 자세한 내용은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.

## SD(Software Distributor) 다시 설치

SW-DIST 소프트웨어 제품은 모든 SD(Software Distributor) 기능, 명령 및 도구를 제공합니다. SW-DIST에 포함된 파일이 삭제되거나 손상되는 경우 제품을 다시 설치해야 합니다. 이 프로세스는 새로운 install-sd 명령을 사용하며, 이 명령은 부록 C, 457페이지의 “SD-UX 교체 또는 업데이트”에서 설명됩니다.

## 여러 버전 설치

네트워크의 다양한 호스트에 여러 버전의 소프트웨어 제품이 설치되어 있는 경우가 많습니다. 여러 버전을 설치함으로써 다음이 가능합니다.

- 결함이 있는 버전을 가려냅니다(필요에 따라 새 버전을 제거하고 이전 버전을 재구성).
- 사용자들은 자신의 속도에 맞게 새로운 소프트웨어 버전으로 마이그레이션할 수 있습니다.

allow\_multiple\_versions 명령 옵션을 통해 여러 버전 허용 여부를 결정할 수 있습니다. false로 설정하면 여러 버전을 설치하거나 구성하는 작업(즉, 동일 제품의 다른 개정본을 다른 위치에 설치)이 허용되지 않습니다. 여러 소프트웨어를 설치하는 것은 허용되지만 여러 버전을 구성하는 것은 권장되지 않습니다.

한 위치에 소프트웨어의 여러 버전이 설치되면 SD-UX 명령의 소프트웨어 사양에 제품 속성을 지정하여 이를 관리할 수 있습니다. (이는 수정 및 아키텍처 등과 같은 다른 버전 속성을 지정하는 것과는 반대가 됩니다). 이를 통해 소프트웨어의 이전 버전과 새로운 버전을 동시에 설치할 수 있으며 두 가지 버전을 모두 구성할 수 있습니다(소프트웨어 패키징에서 지원되는 경우).

IPD에 대한 액세스를 제어하고 swinstall 도구 사용을 제한함으로써 소프트웨어를 개인적으로 무단 설치하는 것을 방지할 수 있습니다.

---

## 참고

여러 버전의 소프트웨어 제품을 관리할 때는 각 버전에 따라 있을 수 있는 제품 상호간의 종속성에 매우 주의해야 합니다. 여러 버전을 설치할 때는 제품 상호간의 다양한 종속 버전도 반드시 설치해야 합니다. 종속성을 재배치할 수 없고 설치하려는 각 버전이 동일한 제품의 서로 다른 버전에 따라 달라지는 경우, 원래 제품의 여러 버전을 설치할 수 없습니다.

---

## 대체 루트에 설치

소프트웨어는 일반적으로 루트 디렉토리에 대해 상대적인 위치에 설치(/)되지만 대체 루트 디렉토리에 도 설치할 수 있습니다.

대체 루트에 설치할 때에는 swinstall 명령에 포함되어 있는 자동 구성 및 호환성 필터링 기능은 수행되지 않습니다. 그러나 swconfig 명령을 사용하여 설치와는 별도로 구성을 수행할 수 있습니다. 자세한 내용은 84페이지의 “설치 구성(swconfig)”을 참조하십시오.

## 호환성 필터링 및 검사

SD-UX는 보통 선택된 다른 대상과 호환되지 않는 소프트웨어 제품을 가려냅니다. 호환성이 있다는 것은 하드웨어의 아키텍처가 소프트웨어에서 필요로 하는 것과 일치한다는 뜻입니다(uname 속성으로 알 수 있음). 또한 OS 버전도 소프트웨어에 적절한 버전이어야 함을 의미합니다. 호환되지 않는 소프트웨어를 실제로 검사하는 것은 선택 단계에서 수행됩니다. 호환성 필터링 및 검사는 allow\_incompatible 옵션으로 결정하고 호스트의 uname 속성에 따라 달라집니다.

**참고**

HP 고객 지원 센터 담당자가 제시한 경우를 제외하고는 호환되지 않는 소프트웨어는 설치하지 않는 것이 바람직합니다.

**표 2-4**

**제품 호환성**

제품 속성	제품 값 (일치시킬 패턴)	대상 루트 속성	
machine_type	ia64*	IA	uname -m
machine_type	9000/*	IA 또는 PA	uname -m
os_name	HP-UX	HP-UX	uname -s
os_release	?11.*	B0.110.11	uname -r
os_version	*	C	uname -v

allow\_incompatible=false(기본값)의 경우 swinstall은 호환되지 않는 소프트웨어의 설치를 제한하고 자동으로 제품을 필터링합니다. **Software Selection** 창에는 모든 대상 시스템의 하드웨어 및 OS와 호환되는 제품만 표시됩니다.

allow\_incompatible=true의 경우 swinstall은 모든 소프트웨어의 설치를 허용합니다. GUI는 모든 제품을 선택할 수 있도록 표시합니다.

-x os\_name 및 -x os\_release 옵션을 사용하여 호환성을 검사할 수 있습니다. 예를 들어 OS 업데이트 중에 시스템이 11.0/32비트로 설치되었고 64비트 버전의 HP-UX로 업데이트하려는 경우, -x os\_name=HP-UX:64 및

-x os\_release=B.11.00 옵션을 지정하여 저장소 병합에 대한 호환성 검사를 목적으로 시스템을 64비트 시스템처럼 보이도록 할 수 있습니다. (파일 세트 레벨에서 이러한 옵션을 지정할 수도 있습니다.)

**참고**

호환성 필터링은 대체 루트 파일 시스템에는 적용되지 않습니다. 대체 루트와 호환되는 소프트웨어를 직접 선택해야 합니다.

## 설치 구성(swconfig)

swconfig 명령은 구성 스크립트를 실행합니다. swinstall 및 swremove가 구성 또는 구성 해제 스크립트를 자동으로 실행하기는 하지만 swconfig를 통해 이들 명령과는 별도로 구성 작업을 수행할 수 있습니다. 이 명령을 통해 다음이 가능합니다.

- 구성에 실패하거나 구성이 지연되거나 구성을 변경해야 하는 경우 스크립트를 실행하여 문제를 해결합니다.
- 연관된 구성 스크립트를 가지고 있는 모든 설치된 소프트웨어를 명시적으로 구성, 구성 해제 또는 재구성합니다.
- 다른 호스트에 있는 소프트웨어를 공유하는 호스트를 구성 또는 구성 해제합니다.

## 기능 및 제한 사항

- swconfig는 다음과 같은 종류의 스크립트를 실행할 수 있습니다.

<b>Configure</b>	설치되어 있는 파일 세트나 제품을 구성합니다(swconfig 및 swinstall 명령으로 실행).
<b>Request</b>	구성 프로세스의 일부로 사용자로부터의 대화식 응답을 요구합니다.
<b>Unconfigure</b>	구성 스크립트로 수행한 구성을 취소합니다. 예를 들면 호스트의 /etc/profile 또는 /sbin/rc 파일에서 구성을 제거합니다. 소프트웨어를 구성 상태에서 설치 상태로 되돌립니다.
- swconfig 명령은 명령줄 인터페이스를 통해서만 사용할 수 있습니다.
- swconfig는 소프트웨어가 실행될 호스트를 구성합니다.
- 파일 세트나 제품에는 구성(구성 해제) 스크립트가 포함될 수 있습니다.
- 이 명령에 대체 루트 디렉토리를 지정하면 swinstall 및 swremove는 구성 스크립트를 자동으로 실행하지 않습니다. 대체 루트를 구성하거나 구성 해제하려면 swconfig를 실행해야 합니다.

- `swinstall` 또는 `swremove`에 `defer_configure` 명령 옵션을 사용하여 루트 디렉토리 /에 설치된 소프트웨어에서 자동 구성을 연기할 수 있습니다(예를 들어 여러 버전이 설치된 경우).
- 기본적으로 `swconfig`는 호환되는 소프트웨어의 구성만을 지원합니다. `allow_incompatible` 옵션을 사용하여 이 기능을 설정하거나 설정하지 않을 수 있습니다.
- 적절하게 작동되기 위해 파일 세트가 다른 소프트웨어 제품을 필요로 하는 경우, 소프트웨어 제품은 구성 상태여야 하고 `enforce_dependencies` 옵션으로 제어합니다.
- `swconfig`는 한 번에 한 가지 버전의 파일 세트만을 구성하며 `allow_multiple_versions` 옵션으로 제어할 수 있습니다.
- `swconfig`를 사용하여 설치 상태와 구성 상태 간에 소프트웨어를 전환할 수 있습니다.
- `swconfig`는 종속성을 통해 작동할 소프트웨어를 자동으로 선택합니다(직접 지정한 소프트웨어에 추가하여). 자세한 내용은 35페이지의 “소프트웨어 종속성”을 참조하십시오.

---

## 참고

`swinstall` 세션에 재부팅 파일 세트가 포함되어 있는 경우(예를 들면 핵심 HP-UX 운영 체제를 새로운 릴리즈로 업데이트하려는 경우) 시스템 재부팅 후 시스템 시작 프로세스의 일부로 구성 스크립트가 자동으로 실행됩니다. 구성을 완료하기 위해 `swconfig`를 실행할 필요는 없습니다.

---

## 구성 프로세스

구성 프로세스는 선택, 분석 및 구성의 세 단계로 구성됩니다.

### 단계 I: 선택

이 단계에서 `swconfig`는 소프트웨어를 선택합니다.

### 단계 II: 분석

이 단계에서 `swconfig`는 소프트웨어를 정상적으로 구성할 수 있는지 여부를 판별합니다(소프트웨어 존재 여부 및 전제 조건 검사 포함). `-p`(미리 보기) 옵션을 지정하여 `swconfig`를 실행하는 경우 분석 완료 후 명령이 중단되고 호스트에서 아무 것도 바꾸지 않습니다.

분석은 로컬 호스트에서 수행됩니다. 분석 단계에서 오류가 발생하면 구성 단계는 수행되지 않습니다. 분석 단계에서 오류가 발생한 경우 오류가 발생한 제품만 제외됩니다. 경고만 발생하면 작업은 계속됩니다.

호스트에서 분석 작업은 다음과 같은 순서로 이루어집니다.

1. 분석을 시작합니다.
2. 소프트웨어 선택을 처리합니다.

IPD(Installed Product Database)에서 정보를 가져와서 호환성을 검사합니다.

시스템은 모든 소프트웨어가 호스트의 `uname` 속성과 호환되는지 검사합니다. 이 검사는 `allow_incompatible` 명령 옵션으로 제어합니다. 이 옵션을 `false`로 설정하면 시스템에서 오류가 발생하고, `true`로 설정하면 경고가 발생합니다.

3. 현재 설치되어 있는 버전 상태를 검사합니다.

- 제품이 없거나 손상되어 있으면 해당 제품을 구성할 수 없으며 이 제품을 설치 및 구성하려면 `swinstall`을 사용하라는 내용의 오류가 표시됩니다.
- 현재 설치되어 있는 버전이 구성되어 있지 않고 `-u`(구성 해제) 옵션이 설정되어 있다면, 시스템은 선택된 파일이나 파일 세트가 이미 구성이 해제되어 있다는 내용의 메시지를 보냅니다.
- 현재 설치되어 있는 버전 상태가 구성 상태이면 검사는 `reconfigure` 옵션으로 제어합니다. 파일 세트가 이미 구성되어 있으며 다시 구성되거나 (`reconfigure`가 `true`인 경우) 다시 구성되지 않을(`reconfigure`가 `false`인 경우) 것이라는 메시지가 발생합니다.

4. 두 번째 버전을 구성하기 위해 검사합니다.

`allow_multiple_versions` 옵션이 `false`로 설정된 경우 이 제품의 다른 버전이 이미 구성되어 있으며 파일 세트는 구성되지 않는다는 내용의 오류가 생성됩니다. 이 옵션을 `true`로 설정하면 두 번째 버전도 구성됩니다.

5. 종속성 상태에 대한 검사가 필요합니다.

- 종속성을 충족시키지 못하면 오류나 경고가 발생합니다. 이는 `enforce_dependencies` 옵션으로 제어됩니다. `enforce_dependencies`가 `true`로 설정되면 파일 세트는 구성되지 않습니다. `enforce_dependencies`가 `false`이면 파일 세트가 구성됩니다.

- 종속성이 전제 조건이라면 구성은 실패합니다.
- 종속성이 필수 조건인 경우 이 파일 세트의 구성은 성공할 것이지만, 해당 제품은 필수 조건의 종속성이 설치 및 구성될 때까지 사용할 수 없습니다.

### 단계 III: 구성

이 단계에서 실제 소프트웨어 구성이 이루어집니다. 구성 또는 구성 해제 스크립트가 실행되고 소프트웨어 상태는 설치 상태에서 구성(또는 구성 해제) 상태로 변경됩니다.

구성 목적은 소프트웨어에 대해 호스트를 구성하고 호스트 특정 정보에 대해 제품을 구성하는 것입니다. 예를 들어 소프트웨어는 호스트의 `.rc` 설정이나 `/etc/profile`의 기본 환경 설정을 변경해야 할 필요가 있을 수 있습니다. 또는 해당 호스트에 대해 적절한 코드워드가 마련되어 있는지 확인해 보고 그렇지 않으면 컴파일 작업을 해야 할 수도 있습니다. 구성 해제는 이 단계를 거꾸로 수행합니다.

구성 작업의 순서가 아래에 나와 있습니다. 제품 순서는 전제 조건 종속성이 있는 경우 이것에 의해 정해집니다. 파일 세트 작동 순서 역시 전제 조건이 있는 경우 전제 조건에 의해 결정됩니다.

1. 각 제품을 구성 (해제)합니다.
2. 연관된 파일 세트에 대한 스크립트를 실행하고 반환 값을 검사합니다.  
오류가 발생하면 파일 세트는 설치 상태로 남게 됩니다. 경고가 발생해도 파일 세트는 그대로 구성됩니다.
3. IPD를 업데이트하여 적절한 설치 또는 구성 상태를 표시합니다.

구성 스크립트는 특정 지침을 준수해야 합니다. 예를 들어 이 스크립트는 소프트웨어가 실행될 호스트에서만 실행되므로, 사용자 정의 스크립트만큼 제한적이지는 않습니다. 스크립트에 대한 자세한 내용은 11장, 347페이지의 “제어 스크립트 사용”을 참조하십시오.

## swconfig 사용

### 구문

```
swconfig [-p] [-u] [-v] [-c catalog] [-C session_file]  
[-f software_file] [-Q date] [-S session_file] [-t target_file]  
[-x option=value] [-X option_file]  
[software_selections] [@ target_selections]
```

### 옵션 및 피연산자

- p 분석 단계를 통해 구성 작업을 실행한 다음 종료함으로써 구성 작업을 미리 봅니다.
- u 소프트웨어를 구성하는 대신 구성을 해제합니다.
- v stdout으로의 상세 출력을 활성화하고 모든 활동을 화면에 표시합니다.
- c *catalog* 응답 파일 또는 요청 스크립트에 의해 작성된 파일의 사본을 저장합니다. 자세한 내용은 11장, 347페이지의 “제어 스크립트 사용”을 참조하십시오.
- C *session\_file* 명령을 실행하고 현재 옵션과 피연산자 값을 다른 세션에 다시 사용할 수 있도록 **session\_file**에 저장합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.
- f *software\_file* 명령줄 대신(또는 그에 추가하여) 별도의 파일에서 대상 선택 목록을 읽습니다. 자세한 내용은 60페이지의 “소프트웨어 파일”을 참조하십시오.
- Q *date* 원격 작업을 수행하는 경우 해당 날짜에 작업이 실행되도록 일정을 예약합니다. 211페이지의 “명령줄에서 작업 예약” 및 6장, 167페이지의 “원격 작업 개요”를 참조하십시오.
- S *session\_file* *session\_file*에 정의된 대로 이전 설치 세션에서 저장된 값을 바탕으로 명령을 실행합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.
- t *target\_file* 명령줄 대신(또는 그에 추가하여) 별도의 파일에서 대상 선택 목록을 읽습니다. 자세한 내용은 61페이지의 “대상 파일”을 참조하십시오.



`-x option=value`

명령 `option`을 `value`로 설정하고 기본값 또는 옵션 파일의 값을 대체합니다. 자세한 내용은 89페이지의 “명령 옵션 변경”을 참조하십시오.

`-X option_file`

`option_file`에서 세션 옵션 및 기능을 읽습니다. 자세한 내용은 89페이지의 “명령 옵션 변경”을 참조하십시오.

`software_selections`

구성될 소프트웨어 객체입니다. 자세한 내용은 58페이지의 “소프트웨어 선택 사항”을 참조하십시오.

`target_selections`

명령의 대상입니다. 자세한 내용은 60페이지의 “대상 선택 사항”을 참조하십시오.

## 명령 옵션 변경

명령을 호출할 때 명령줄 옵션을 추가로 지정(-x 옵션 사용)하거나 파일에서 미리 정의된 값을 읽어서 이 명령의 작동 방식을 변경할 수 있습니다. 다음 표에서는 swconfig에 적용되는 기본값과 옵션을 보여줍니다.

표 2-5

swconfig 명령 옵션 및 기본값

<ul style="list-style-type: none"> <li>• admin_directory=/var/adm/sw</li> <li>• agent_auto_exit=true</li> <li>• agent_timeout_minutes=10000</li> <li>• allow_incompatible=false</li> <li>• allow_multiple_versions=false</li> <li>• ask=false</li> <li>• autoremove_job=false</li> <li>• autoselect_dependencies=true</li> <li>• autoselect_dependents=true</li> <li>• compress_index=false</li> <li>• controller_source=</li> <li>• enforce_dependencies=true</li> <li>• enforce_scripts=true</li> <li>• installed_software_catalog=products</li> <li>• job_title=</li> <li>• log_msgid=0</li> <li>• logdetail=false</li> </ul>	<ul style="list-style-type: none"> <li>• logfile=/var/adm/sw/swconfig.log</li> <li>• loglevel=1</li> <li>• mount_all_filesystems=true</li> <li>• preview=false</li> <li>• reconfigure=false</li> <li>• reuse_short_job_numbers=true</li> <li>• rpc_binding_info=</li> <li>• ncaen_ip_tcp:[2121]</li> <li>• ncaen_ip_udp:[2121]</li> <li>• rpc_timeout=5</li> <li>• run_as_superuser=true</li> <li>• select_local=true</li> <li>• software=</li> <li>• targets=</li> <li>• verbose=1</li> <li>• write_remote_files=false</li> </ul>
--	--

## 추가 정보

옵션 설정 및 각 옵션에 대한 전체 목록 및 설명은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.

## 구성 작업 및 예

로컬 호스트의 루트에 위치한 productA를 구성하려면 다음과 같이 입력합니다.

```
swconfig productA
```

로컬 호스트의 기본 디렉토리에 설치되어 있는 mysoft 파일에서 소프트웨어 선택 사항을 구성 해제하려면 다음과 같이 입력합니다.

```
swconfig -u -f mysoft
```

기본 옵션 값을 사용하여 Omniback 제품을 다시 구성하려면 다음과 같이 입력합니다.

```
swconfig -x reconfigure=true Omniback
```

Omniback의 특정 버전을 구성하려면 다음과 같이 입력합니다.

```
swconfig Omniback,r=2.0
```

로컬 호스트에 C 및 Pascal 제품을 구성하려면 다음과 같이 입력합니다.

```
swconfig cc pascal
```

Product1을 구성하려면 요청 스크립트에 의해 생성된 연관된 응답 파일을 사용하고 /tmp/resp1 아래에 응답 파일을 저장합니다.

```
swconfig -x ask=true -c /tmp/resp1 Product1
```

HP Omniback 제품을 다시 구성하려면 다음과 같이 입력합니다.

```
swconfig -x reconfigure=true Omniback
```

/opt/Omniback\_v2.0에 설치된 HP Omniback 버전을 구성하려면 다음과 같이 입력합니다.

```
swconfig Omniback,l=/opt/Omniback_v2.0
```

/tmp/install.hosts 파일에 나열된 호스트에서/tmp/install.products 파일에 나열된 소프트웨어 선택 사항을 구성 해제하려면 다음과 같이 입력합니다.

```
swconfig -u -f /tmp/install.products \  
-t /tmp/install.hosts
```

## 설치 확인(swverify)

swverify 명령은 지정된 호스트에서 저장소와 설치 및 구성된 소프트웨어 제품을 확인합니다.

### 기능 및 제한 사항

- 설치되었거나 구성된 소프트웨어가 설치된 호스트와 호환되는지 여부를 판별합니다.
- 모든 종속성(전제 조건, 필수 조건)이 충족되는지(설치된 소프트웨어에 대해) 또는 충족될 수 있는지(복사된 소프트웨어에 대해) 확인합니다.
- 제품 구성이 올바른지 확인할 수 있는 확인 스크립트(소프트웨어의 설치 상태가 구성되었는지 확인하는 스크립트)를 실행합니다.
- 설치된 소프트웨어의 문제를 수정하거나 보고하기 위해 수정 스크립트를 실행합니다.

**Fix**                      설치된 소프트웨어의 문제점을 수정하고 보고합니다. 일반적으로는 유실된 디렉토리를 작성하고, 파일 변경 사항(모드, 소유자, 그룹, 1차, 2차)을 수정하며 잃어버린 심볼릭 링크를 다시 작성하는 데 사용됩니다.

**Verify**                  일반적인 swverify 확인에 추가하여 파일 세트 또는 제품의 구성을 확인합니다

(자세한 내용은 11장, 347페이지의 “제어 스크립트 사용”을 참조하십시오.)

- 유실된 파일을 보고하고 권한, 파일 유형, 크기, 체크섬, mtime, 링크 소스 및 1차/2차 속성과 같은 모든 파일 속성을 검사합니다.
- 종속성을 통해 작동할 소프트웨어를 자동으로 선택합니다(직접 지정한 소프트웨어에 추가하여). 자세한 내용은 35페이지의 “소프트웨어 종속성”을 참조하십시오.

## 확인 프로세스

소프트웨어 확인 프로세스는 선택 및 분석의 두 가지 단계로만 구성됩니다.

### 단계 I: 선택

이 단계는 필요한 모든 호스트, 소프트웨어, 종속성 및 제품 정보 등 명령줄에서 모든 정보를 해석하는 **swverify** 기능으로 구성됩니다.

### 단계 II: 분석

**swverify**의 분석 기능이 호스트에서 수행됩니다. 호스트 환경은 변경되지 않습니다. 각 분석 작업은 다음과 같은 순서로 이루어집니다.

1. 분석을 시작합니다.
2. 소프트웨어 선택을 처리합니다. 시스템은 IPD(Installed Products Database)나 저장소 카탈로그에 액세스하여 선택된 소프트웨어에 대한 제품 정보를 구합니다. 설치된 소프트웨어의 경우 시스템은 모든 제품이 해당 `uname` 속성과 호환되는지 확인합니다. 이 검사는 `allow_incompatible` 기본 옵션으로 제어합니다.
  - `allow_incompatible`을 **false**로 설정하면 시스템은 제품이 호스트와 호환되지 않는다는 내용의 오류를 생성합니다.
  - `allow_incompatible`을 **true**로 설정하면 제품이 호환되지 않는다는 경고가 발생합니다.
3. 파일 세트의 올바른 상태(설치, 구성 또는 사용 가능)를 확인합니다. 설치된 소프트웨어의 경우 **swverify**는 여러 버전을 확인하고 이는 `allow_multiple_versions` 옵션으로 제어합니다.
  - `allow_multiple_versions`가 **false**이면 해당 제품의 여러 버전이 존재하고 이 옵션은 사용 불가라는 오류가 발생합니다.
  - `allow_multiple_versions`가 **true**이면 여러 버전이 존재한다는 내용의 경고가 발생합니다.
4. 종속성을 확인합니다. 종속성을 충족시키지 못하면 오류나 경고가 발생합니다. 종속성은 `enforce_dependencies` 옵션으로 제어합니다.
  - `enforce_dependencies`가 **true**이면 종속성의 유형 및 제품이 현재 어떤 상태인지를 나타내는 오류가 발생합니다.

- `enforce_dependencies`가 **false**이면 같은 내용의 경고가 발생합니다.
  - 종속성이 필수 조건인 경우 소프트웨어가 작동되기 전에 존재해야 합니다.
  - 종속성이 전제 조건인 경우 소프트웨어가 설치되거나 구성되기 전에 존재해야 합니다.
5. 전제 조건 순서에 따라 설치된 소프트웨어에 확인 또는 수정 스크립트를 실행합니다.

확인 스크립트는 소프트웨어의 구성이 올바른지 확인하는 데 사용됩니다. 확인 스크립트에서 업체에 따라 다를 수 있는 작업은 다음과 같습니다.

- 제품의 활성화 또는 비활성 상태인지 여부를 판별합니다.
- 제품 구성 파일이 손상되었는지 확인합니다.
- OS 플랫폼, 서비스 또는 구성 파일에 제품이 올바르게(올바르지 않게) 구성되어 있는지 확인합니다.
- 사용권 요소를 확인합니다.

업체에서 공급하는 스크립트가 실행되고 반환 값은 오류 (1인 경우) 또는 경고 (2인 경우)를 생성합니다.

스크립트는 전제 조건 순서에 따라 실행됩니다.

6. 다음에 대해 파일 수준에 대한 확인을 수행합니다.

- `control_files`의 내용(**mtime**, 크기 및 체크섬)
- 파일의 내용(**mtime**, 크기 및 체크섬)
- 누락된 `control_files`, 파일 및 디렉토리
- 설치된 파일의 권한(소유자, 그룹, 모드)
- 적절한 심볼릭 링크 값

## swverify 사용

### 구문

```
swverify [-d|-r] [-F][-v] [-C session_file] [-f software_file]  
[-Q date] [-S session_file] [-t target_file] [-x option=value]  
[-X option_file] [software_selections][@ target_selections]
```

### 옵션 및 피연산자

- d 설치된 소프트웨어가 아닌 저장소에 대해 작동합니다. 163페이지의 “저장소 확인(**swverify -d**)”을 참조하십시오.
- r /가 아닌 대체 루트에서 작동합니다. 확인 스크립트는 실행되지 않습니다.
- v stdout으로의 상세 출력을 활성화하고 모든 활동을 화면에 표시합니다. 명령이 실행될 때 결과를 볼 수 있습니다.
- C *session\_file* 명령을 실행하고 현재 옵션과 피연산자 값을 다른 세션에 다시 사용할 수 있도록 *session\_file*에 저장합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.
- F 수정 스크립트를 실행합니다. 자세한 내용은 370페이지의 “Fix 스크립트”를 참조하십시오.
- f *software\_file* 명령줄 대신(또는 그에 추가하여) 별도의 파일에서 대상 선택 목록을 읽습니다. 자세한 내용은 60페이지의 “소프트웨어 파일”을 참조하십시오.
- Q *date* 원격 작업을 수행하는 경우 해당 날짜에 작업이 실행되도록 일정을 예약합니다. 211페이지의 “명령줄에서 작업 예약” 및 6장, 167페이지의 “원격 작업 개요”를 참조하십시오.
- S *session\_file* *session\_file*에 정의된 대로 이전 확인 세션에서 저장된 값을 바탕으로 명령을 실행합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.
- t *target\_file* 명령줄 대신(또는 그에 추가하여) 별도의 파일에서 대상 선택 목록을 읽습니다. 자세한 내용은 61페이지의 “대상 파일”을 참조하십시오.
- x *option=value* 명령 *option*을 *value*로 설정하고 기본값 또는 옵션 파일의 값을 대체합니다. 자세한 내용은 95페이지의 “명령 옵션 변경”을 참조하십시오.

*-X option\_file*

*option\_file*에서 세션 옵션 및 기능을 읽습니다. 자세한 내용은 95페이지의 “명령 옵션 변경”을 참조하십시오.

*software\_selections*

확인될 소프트웨어 객체입니다. 자세한 내용은 58페이지의 “소프트웨어 선택 사항”을 참조하십시오.

*target\_selections*

명령의 대상입니다. 자세한 내용은 60페이지의 “대상 선택 사항”을 참조하십시오.

**명령 옵션 변경**

명령을 호출할 때 명령줄 옵션을 추가로 지정(-x 옵션 사용)하거나 파일에서 미리 정의된 값을 읽어서 이 명령의 작동 방식을 변경할 수 있습니다. 다음 표에서는 swverify에 적용되는 기본값과 옵션을 보여줍니다.

**표 2-6**

**swverify 명령 옵션 및 기본값**

- |   |                                       |
|---|---------------------------------------|
| • admin_directory=/var/spool/sw               | • enforce_dependencies=true           |
| • agent_auto_exit=true                        | • installed_software_catalog=products |
| • agent_timeout_minutes=10000                 | • job_title=                          |
| • allow_incompatible=false                    | • log_msgid=0                         |
| • allow_multiple_versions=false               | • logdetail=false                     |
| • autoremove_job=false                        | • logfile=/var/adm/sw/swverify.log    |
| • autoselect_dependencies=true                | • loglevel=1                          |
| • check_contents=true                         | • mount_all_filesystems=true          |
| • check_contents_uncompressed=false           | • reuse_short_job_numbers=true        |
| • check_contents_use_cksum=true               | • rpc_binding_info=                   |
| • check_permissions=true                      | • ncacn_ip_tcp:[2121]                 |
| • check_requisites=true                       | • ncadg_ip_udp:[2121]                 |
| • check_scripts=true                          | • rpc_timeout=5                       |
| • check_volatile=false                        | • run_as_superuser=true               |
| • controller_source=                          | • select_local=true                   |
| • distribution_target_directory=/var/spool/sw | • software=                           |
|   | • verbose=1                           |

**추가 정보**

옵션 설정 및 각 옵션에 대한 전체 목록 및 설명은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.

## 확인 작업 및 예

*myhosts*에 위치한 설치된 파일 세트인 *mysoft.myfileset*를 확인하려면 다음과 같이 입력합니다.

```
swverify -d mysoft.myfileset @ myhosts
```

(확인할 소프트웨어가 로컬 호스트의 기본 저장소에 있으므로 @기호와 *myhost* 대상 지정은 선택적입니다.)

로컬 호스트에 설치된 C 및 Pascal 제품을 확인하려면 다음과 같이 입력합니다.

```
swverify C Pascal
```

로컬 호스트에 설치된 HP Omniback 제품을 확인하고 프로세스로부터의 자세한 메시지(-v)를 stdout으로 표시하려면 다음과 같이 입력합니다.

```
swverify -v Omniback
```

/opt/Omniback의 로컬 호스트에 설치된 Omniback 2.0 버전을 확인하려면 다음과 같이 입력합니다.

```
swverify Omniback,r=2.0 @ /opt/Omniback
```

HP Omniback 특정 버전을 확인하려면 다음과 같이 입력합니다.

```
swverify Omniback,l=/opt/Omniback_v2.0
```

로컬 저장소의 내용 전체를 확인하려면 다음과 같이 입력합니다.

```
swverify -d \*@/var/spool/sw
```



## 3 설치된 소프트웨어 관리

이 장에서는 저장소가 아닌 소프트웨어를 설치 후에 관리하는 방법에 대해 개략적으로 설명합니다. `swlist`, `swmodify` 및 `swremove` 명령은 이러한 소프트웨어 관리 작업을 수행하는 데 도움을 줍니다.

표 3-1

이 장의 항목

설명하는 항목
98페이지의 “소프트웨어 나열( <code>swlist</code> )”
117페이지의 “IPD 수정( <code>swmodify</code> )”
124페이지의 “설치된 소프트웨어 제거( <code>swremove</code> )”

## 소프트웨어 나열(**swlist**)

**swlist** 명령은 로컬 호스트에 설치되었거나 차후 배포를 위해 저장소에 저장해 놓은 소프트웨어 제품에 대한 사용자 정의 목록을 작성합니다.

### **swlist** 기능 및 제한 사항

**swlist**를 사용하여 다음을 수행할 수 있습니다.

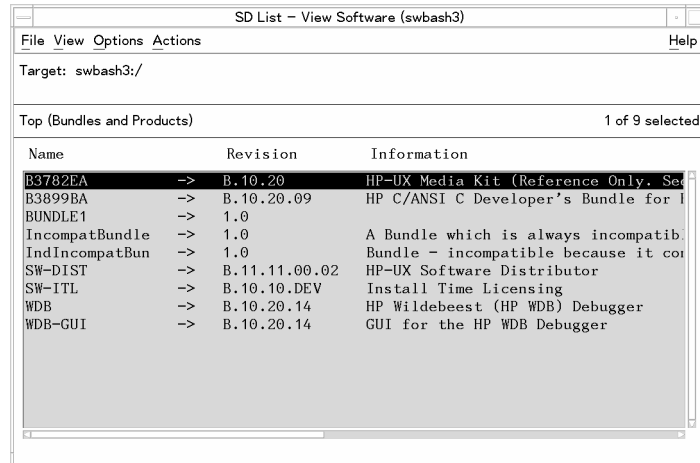
- 선택적으로 **GUI**를 사용합니다.
- 목록의 표시 수준(번들, 제품, 하위 제품, 파일 세트 또는 파일)을 지정합니다.
- 각 수준에 대해 표시할 소프트웨어 속성 세트를 지정합니다. 소프트웨어 속성이란 **IPD(Installed Products Database)**나 카탈로그 파일에 들어 있는 제품에 대한 정보 항목을 말합니다. 이 항목에는 제품 이름이나 태그, 크기(**KB** 단위), 개정 번호 등이 포함됩니다.
- **swinstall** 또는 **swremove** 명령에 대한 입력으로 사용하기 위해 제품, 하위 제품 또는 파일 세트 목록을 작성합니다.
- 소프트웨어 소스에 대한 목차를 표시합니다.
- 각 수준에 대해 선택된 소프트웨어 속성을 표시합니다.
- 소프트웨어 선택 사항의 제품 구조를 표시합니다.
- 대체 루트 디렉토리에 저장되어 있는 소프트웨어를 나열합니다.
- 지정된 호스트에서 저장소를 표시합니다.
- 사용 가능한 또는 적용된 패치 범주를 나열합니다.
- 파일 세트의 적용된 패치 값을 나열합니다.

### **swlist** GUI 사용

**swlist -i** 명령은 대화식으로 소프트웨어를 나열하고 소프트웨어 정보를 표시하는 **swlist** GUI 프로그램을 시작합니다. **swlist -i -d** 명령을 통해 저장소 또는 실제 미디어에서 사용할 수 있는 소프트웨어에 대한 정보를 표시할 수 있습니다.

그림 3-1

swlist 브라우저



- 번들 및 제품은 기본적인 최상위 표시 항목입니다.
- 목록에서 특정 항목을 열려면 해당 항목을 두 번 클릭합니다.
- 파일을 두 번 클릭하면 파일 속성이 표시됩니다.

목록 탐색 및 이동

다음과 같은 기능은 목록을 탐색하고 이동하는 데 도움이 됩니다.

- 현재 목록을 검색하려면 **File**→**Search...**를 선택합니다.
- 항목에 대한 보기 옵션 팝업 메뉴를 표시하려면 해당 항목을 마우스 오른쪽 단추로 클릭합니다. 팝업 옵션은 다음과 같습니다.
  - **Open Item**은 항목의 내용을 표시합니다.
  - **Close Level**은 현재 항목을 닫고 객체의 다음 상위 수준을 표시합니다.
  - **Show Description of Software...**는 현재 항목의 속성 정보를 표시합니다.

보기 변경

**View** 메뉴를 사용하여 표시된 열을 변경하고 필터를 선택하고 정보를 정렬합니다.

- **Columns**는 **Columns Editor**를 표시합니다. 표시할 소프트웨어 정보 열(즉 소프트웨어 이름, 개정 번호, 정보, 크기(KB), 아키텍처, 범주 등)과 열의 순서를 선택할 수 있습니다.
- **Filter...**는 각 필드에 대해 논리 연산자와 관계 연산자로 표시 목록을 필터링할 수 있는 대화 상자를 표시합니다.
- **Sort...**를 통해 표시된 정보에 대해 정렬 필드, 순서 및 기준을 선택할 수 있습니다.
- **Change Software View**를 통해 최상위 보기와 제품 보기 간에 전환을 할 수 있습니다.
- **Change Software Filter...**를 통해 미리 정의된 필터 목록에서 선택할 수 있습니다. (최상위 소프트웨어 객체에만 적용됩니다.)

### 작업 수행

**Actions** 메뉴를 사용하여 표시된 항목을 열거나 닫고 로그 파일 정보 및 소프트웨어 설명을 표시합니다.

- **Open Item**은 항목을 엽니다. (해당 항목을 두 번 클릭하는 것과 동일합니다.)
- **Close Level**은 현재 수준을 닫습니다. (**..(go up)**을 두 번 클릭하는 것과 동일합니다.)
- **Change Target**은 대체 루트(**swlist -i**) 또는 대체 저장소(**swlist -i -d**)를 선택하기 위한 경로를 입력할 수 있는 대화 상자를 엽니다.
- **Show Logfile**은 시스템 로그 파일을 표시합니다.
- **Show Audit Log**는 감사 로그(**swlist -i -d**에 한함)에 저장된 소프트웨어 저장소 감사 정보를 표시합니다. 자세한 내용은 162페이지의 “소스 저장소 감사”를 참조하십시오.
- **Show Description of Software**는 현재 선택된 항목에 대한 속성 정보를 표시합니다.

### 명령줄 사용

#### 구문

```
swlist [-d|-r]] [-i] [-R] [-v] [-a attribute] [-c catalog]
[-C session_file] [-f software_file] [-l level] [-s source]
[-S session_file] [-t target_file] [-x option=value]
[-X option_file] [software_selections] [@ target_selections]
```

**옵션 및 피연산자**

- d** 저장소에서 사용할 수 있는 제품을 나열합니다. 자세한 내용은 161페이지의 “저장소 내용 나열(*swlist -d*)”을 참조하십시오.
- i** GUI를 시작합니다(98페이지의 “*swlist* GUI 사용” 참조).
- r** 대체 루트(/ 대신)의 제품을 나열합니다.
- R** 다음에 대한 축약 옵션입니다.  
`-l bundle -l product -l subproduct -l fileset`
- a *attribute*** 특정 속성을 표시합니다. 속성을 여러 개 표시하려면 **-a** 옵션을 여러 개 지정합니다. 소프트웨어 객체에 대한 속성을 모두 나열하려면 **-v** 옵션을 사용합니다. 태그 속성은 제품, 하위 제품 및 파일 세트에 대해 항상 표시되고, 경로(파일 이름) 속성은 파일 객체에 대해 항상 표시됩니다. **-c** 옵션을 사용하는 경우에는 이 옵션을 적용할 수 없습니다.
- v** **-a** 옵션이 지정되지 않으면 객체에 대한 속성을 모두 나열합니다. (업체에서 정의한 속성은 포함되지 않습니다.) 다음과 같은 형식으로 한 줄에 속성 하나씩 표시됩니다.  
`attribute_name attribute_value`
- c *catalog*** *catalog*로 지정된 디렉토리에 자세한 카탈로그 구조 정보를 기록합니다. 이 정보를 배포 및 설치된 소프트웨어 카탈로그 정보 나열에 사용할 수 있습니다. 파일 수준 및 제어 스크립트에 이르는 모든 속성이 기록됩니다. 이 옵션을 사용하는 경우 **-a** 및 **-l** 옵션은 적용되지 않습니다. 자세한 내용은 385페이지의 “사용자 응답 요구(*swask*)”를 참조하십시오.
- C *session\_file*** 명령을 실행하고 현재 옵션과 피연산자 값을 다른 세션에 다시 사용할 수 있도록 *session\_file*에 저장합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.
- f *software\_file*** 명령줄 대신(또는 그에 추가하여) 별도의 파일에서 대상 선택 목록을 읽습니다. 자세한 내용은 60페이지의 “소프트웨어 파일”을 참조하십시오.

-l level depot, bundle, product, subproduct, fileset 또는 file 과 같이 지정된 수준에 **이르는** 모든 소프트웨어 객체를 나열합니다. (수준에 대한 자세한 내용은 109페이지의 “수준별 소프트웨어 나열”을 참조하십시오.) 명령에 한 가지 수준만을 지정할 수 있습니다. 수준을 지정할 때 소프트웨어 이름이나 하위 제품 이름 등은 사용할 수 없습니다. -c 옵션을 사용하는 경우에는 이 옵션을 적용할 수 없습니다.

표 3-2

-l 옵션

옵션	작업
swlist -l root	루트(지정된 대상 호스트의 루트) 수준을 표시합니다.
swlist -l shroot	공유 루트를 표시합니다.
swlist -l prroot	개별 루트를 표시합니다.
swlist -l bundle	번들만 표시합니다.
swlist -l product	제품만 표시합니다.
swlist -l subproduct	제품 및 하위 제품을 표시합니다.
swlist -l fileset	제품, 하위 제품 및 파일 세트를 표시합니다.
swlist -l file	제품, 하위 제품, 파일 세트, 파일 및 번호(소프트웨어 라이선스에 사용되는 번호)를 표시합니다.
swlist -l category	정의에 범주 객체를 포함하는 패치의 경우 사용 가능한 패치의 모든 범주를 표시합니다.
swlist -l patch	적용된 패치를 모두 표시합니다.

- s source*      나열할 소프트웨어 소스를 지정합니다. 기본 소스 유형은 로컬 호스트의 디렉토리나 저장소(일반적으로 `/var/spool/sw`)입니다. 구문은 다음과 같습니다.
- `[host][:[/directory]]`
- 호스트는 호스트 이름, 도메인 이름 또는 인터넷 주소 등으로 지정할 수 있습니다. 디렉토리는 절대 경로로 지정해야 합니다.
- S session\_file*
- session\_file*에 정의된 대로 이전 설치 세션에서 저장된 값을 바탕으로 명령을 실행합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.
- t target\_file*
- 명령줄 대신(또는 그에 추가하여) 별도의 파일에서 대상 선택 목록을 읽습니다.
- x option=value*
- 명령 *option*을 *value*로 설정하고 기본값 또는 옵션 파일의 값을 대체합니다. 자세한 내용은 104페이지의 “명령 옵션 변경”을 참조하십시오.
- X option\_file*
- option\_file*에서 세션 옵션 및 기능을 읽습니다. 자세한 내용은 104페이지의 “명령 옵션 변경”을 참조하십시오.
- software\_selections*
- 나열될 소프트웨어 객체입니다. 자세한 내용은 58페이지의 “소프트웨어 선택 사항”을 참조하십시오.
- target\_selections*
- 명령의 대상입니다. (*swlist*의 경우 *target\_selections*는 소프트웨어 선택 사항을 나열하는 또 다른 방법입니다.)

### 명령 옵션 변경

명령을 호출할 때 명령줄 옵션을 추가로 지정(-x 옵션 사용)하거나 파일에서 미리 정의된 값을 읽어서 이 명령의 작동 방식을 변경할 수 있습니다. 다음 표에서는 `swlist`에 적용되는 기본값과 옵션을 보여줍니다.

표 3-3

#### `swlist` 명령 옵션 및 기본값

- `admin_directory=/var/adm/sw`
- `agent_timeout_minutes=10000`
- `codeword=`
- `customer_id=`
- `distribution_target_directory=/var/spool/sw`
- `installed_software_catalog=products`
- `layout_version=1.0`
- `level=`
- `log_msgid=1`
- `one_liner=revision title`
- `patch_one_liner=title patch_state`
- `rpc_binding_info=ncacn_ip_tcp:[2121]ncadg_ip_udp:[2121]`
- `rpc_timeout=5`
- `run_as_superuser=true`
- `select_local=true`
- `show_superseded_patches=false`
- `software=`
- `software_view=all_bundles`
- `targets=`
- `verbose=1`

### 추가 정보

각 기본값에 대한 자세한 설명은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.



## 소프트웨어 나열 작업 및 예

**swlist** 대화식 인터페이스를 실행하려면 다음과 같이 입력합니다.

```
swlist -i @ host1
```

대화식 **swlist**를 사용하여 저장소를 보려면 다음과 같이 입력합니다.

```
swlist -i -d @ /tmp/depot
```

로컬 호스트에서 루트(/)에 설치된 소프트웨어 목록(이름별)을 표시하려면 다음과 같이 입력하면 됩니다.

```
swlist
```

그러면 다음과 같은 내용이 표시됩니다.

```
# Initializing...
# Contacting target "xyyyzz"...
#
# Target:  xyyzz:/

# Bundle(s):

B3782CA B.11.00 HP-UX Media Kit (Reference Only. See Descr.)
B3898AA B.11.00 HP C/ANSI C Developer's Bundle for HP-UX 11.00
HPUXEngRT B.11.00 English HP-UX Run-time Environment

# Product(s) not contained in a Bundle:

HMS                1.01
OBAM5_0            B.11.00          ObAM 5.0
```

옵션 세트를 지정하지 않고 소프트웨어를 선택하지 않은 상태에서 **swlist**를 사용하면 모든 소프트웨어 번들과 번들에 포함되지 않은 모든 제품이 나열됩니다. **-d** 옵션을 지정하면 로컬 호스트의 기본 저장소에 있는 소프트웨어에 대한 동일한 목록이 생성됩니다.

다음 예에서 **swlist** 요청은 표준 출력으로 보내집니다. 모든 예에서 **one\_liner=** 기본값은 “revision size title”이고 **level=** 기본값은 “product”입니다.

- 로컬 테이프 저장소 **/dev/rmt/0m**의 내용을 나열하려면 다음과 같이 입력합니다.

```
swlist -d @ /dev/rmt/0m
```

— 또는 —

```
swlist -s /dev/rmt/0m
```

그러면 다음과 같이 출력이 생성됩니다.

```
AUDIT      3.5   9834  Trusted Systems Auditing Utils
COMMANDS  1.7   4509  Core Command Set
C-LANG    2.5   5678  C Programming Language
NETWORKING 2.1   9072  Network Software
KERNEL    1.4   56908 Kernel Libraries and Headers
VUE       1.3   5489  Vue (Instant Ignition Release)
WINDOWS   2.06 10423 Windowing Products
```

- 로컬 테이프 저장소 `/dev/rmt/0m`의 미디어 속성을 모두 나열합니다.

```
swlist -v -l depot @ /dev/rmt/0m
```

— 또는 —

```
swlist -vl depot -s dev/rmt/0m
```

```
type          distribution
tag           CORE OS
description   HP-UX Core Operating System Software Disk
number        B2358-13601
mod_date      June 1998
```

- 로컬 호스트에 설치된 `OS_CORE` 제품에 대한 `README` 파일을 표시합니다.

```
swlist -a readme OS-Core | more
```

```
readme
*****
* Introduction *
*****
```

```
The Release Notes for HP-UX Release X.0 contain an
overview of the new/changed product features that
are included in the release. For detailed
information about these features, refer to the
appropriate product manuals. This document does not
contain information about software changes made as a
result of a Service Request; that information may be
found in the Software Release Bulletin (SRB) for Release
X.0.
```

```
*****
* Hardware Support *
*****
The HP 9000 Model XXX is no longer supported.
...
```

- /swmedia에 위치한 host1의 소프트웨어 저장소에 저장된 제품을 나열합니다. 이 예에서 swlist one\_liner는 “title size architecture”로 가정합니다.

```
swlist -d @ host1:/swmedia
```

```
FRAME      Frame Doc. Pkg 2319  HP-UX_9000_Series_AorB
FRAME      Frame Doc. Pkg 2458  OSF1_9000_Series_1.0
ME30       3-D Mech. Eng  5698  HP-UX_9000_Series300_AorB
SOFTBENCH  Dev Env        4578  HP-UX_9000_Series300
TEAMWORK   Design/Analysis 3478  HP-UX_9000_Series 300/400
```

(미디어에 두 가지 개정본의 **FRAME** 제품이 들어 있습니다.)

### 목록 형태 변경 옵션 사용

옵션 파일에서 목록 기본값을 변경함으로써 목록의 형태와 내용을 제어할 수 있습니다. swlist를 호출할 때마다 매번 소프트웨어 수준과 속성을 반복하는 대신 다음과 같이 사용할 수 있습니다.

**level** 이 옵션은 제품, 하위 제품, 파일 세트 또는 파일과 같은 목록 표시 수준을 미리 결정합니다. 예를 들어 이 기본값을 level=fileset로 설정하면 차후 swlist 명령은 선택한 각 호스트, 저장소 또는 제품에 대해 파일 세트를 포함하여 파일 세트에 이르는 모든 항목을 항상 나열할 것입니다.

**one\_liner="attribute attribute attribute"**

이 옵션은 기본 목록에 표시될 속성(개정본, 크기, 제목 등)을 지정합니다. 이러한 속성은 <탭>이나 <공백>으로 구분하며 인용 부호(" ")로 묶어 표시합니다. 속성을 여러 개 선택할 수 있지만 모든 속성이 적용 가능한 모든 소프트웨어 수준(제품, 하위 제품 또는 파일 세트 등)에 존재하는 것은 아닙니다. 예를 들어 title이라는 소프트웨어 속성은 번들, 제품, 하위 제품 및 파일 세트에 대해 사용할 수 있지만 architecture 속성은 제품에 대해서만 사용할 수 있습니다.

명령에 -v나 -a 옵션을 지정하지 않으면 swlist는 파일이 아닌 각 소프트웨어 객체 수준(번들, 제품, 하위 제품 및 파일 세트)에 대해 one\_liner 기본값에 설명된 대로 정보를 표시합니다.

### 속성 나열

`-a` 옵션 하나에 하나의 속성만을 지정할 수 있습니다. 그러나 `tag` 속성은 항상 기본적으로 포함되므로 `-a revision`은 모든 제품 이름 및 이 제품들의 버전 번호를 나열합니다.

예를 들어 CD-ROM(/SD\_CDROM 디렉토리에 마운트됨)의 소프트웨어 번들에 코드워드가 필요한지 여부를 표시하려면 다음과 같은 명령을 사용합니다.

```
swlist -d -a is_protected @ /SD_CDROM
```

정보량이 많은 속성(예를 들면 `README`)은 물리적으로 별도의 파일에 저장되었다가 `-a README`가 요청될 때 직접 표시됩니다.

SD-UX 속성의 전체 목록은 `sd(4)` 맨페이지를 참조하십시오.

### 사용자 정의 목록 작성

`swlist` 옵션 및 기본값을 통해 사용자의 특정 요구 사항에 맞는 목록을 작성할 수 있습니다. 이러한 목록은 로컬 호스트에 설치된 소프트웨어 제품을 나열하는 것 만큼 간단할 수도 있고, 설치된 파일, 파일 세트, 하위 제품, 제품 및 번들에 대한 여러 열로 구성된 목록처럼 복잡할 수도 있습니다.

예를 들어 명령줄에서 `one-liner` 옵션을 변경하려면 다음과 같은 명령을 사용합니다.

```
swlist -x one_liner="name revision size title"
```

그러면 로컬 호스트에 설치된 제품을 모두 나열한 아래와 같은 목록이 만들어집니다.

RX	1.98	9845	RX X Terminal - all software
ALLBASE	8.00.1	6745	Database Products
C-LANG	2.5	5678	Programming Language
DIAGNOSTICS	2.00	56870	Hardware Diagnostic Programs
DTP68	2.00	26775	Desktop Publishing
LISP-LANG	8.00.1	90786	LISP Programming Language
WINDOWS	2.06	10423	Windowing Products

이 목록에는 왼쪽에서 오른쪽으로 차례로 제품의 태그, 버전 번호, 크기(KB) 및 제목 또는 전체 이름이 표시됩니다.

---

### 참고

소프트웨어 수준 및 속성에 대해 명령줄에 지정한 값이 기본 옵션 파일의 값을 재정의합니다.

---

one\_liner 기본값을 기본 파일의 {revision size title}로 바꿀 수도 있습니다. 그러면 host2의 C-LANG 제품의 목록은 다음과 같을 것입니다.

**swlist C-LANG @ host2**

```
C-LANG.C-COMPILE 8.0    1346      C Compiler Components
C-LANG.C-LIBS    8.0    2356      Runtime Libraries
C-LANG.C-MAN     8.0    1976      Programming Reference
```

**패치 나열**

swlist를 사용하여 소프트웨어 패치와 이들의 상태를 나열할 수 있습니다.

**소프트웨어 코드워드 및 고객 ID 사용**

코드워드로 보호된 소프트웨어를 보려고 하면 swlist 명령이 코드워드를 입력하라는 메시지를 표시합니다. 또한 명령줄이나 GUI에서 새로운 코드워드를 입력할 수도 있습니다. 이 프로세스는 swinstall에서 사용된 것과 동일합니다. 자세한 내용은 80페이지의 “소프트웨어 코드워드 및 고객 ID 사용”을 참조하십시오.

**수준별 소프트웨어 나열**

-l level 옵션을 통해 depot, bundle, product, subproduct, fileset 또는 file과 같이 지정된 수준에 **이르는** 모든 소프트웨어 객체를 나열할 수 있습니다. 시작점으로 수준을 선택하면 그 수준까지의 항목을 나열합니다.

표 3-4

**-l 옵션**

옵션	작업
swlist -l root	루트(지정된 대상 호스트의 루트) 수준을 표시합니다.
swlist -l shroot	공유 루트를 표시합니다.
swlist -l prroot	개별 루트를 표시합니다.
swlist -l bundle	번들만 표시합니다.
swlist -l product	제품만 표시합니다.
swlist -l subproduct	제품 및 하위 제품을 표시합니다.

표 3-4 -l 옵션(계속)

옵션	작업
swlist -l fileset	제품, 하위 제품 및 파일 세트를 표시합니다.
swlist -l file	제품, 하위 제품, 파일 세트, 파일 및 번호(소프트웨어 라이선스에 사용되는 번호)를 표시합니다.
swlist -l category	정의에 범주 객체를 포함하는 패치의 경우 사용 가능한 패치의 모든 범주를 표시합니다.
swlist -l patch	적용된 패치를 모두 표시합니다.

소프트웨어 목록의 시작점은 항상 -l 및 -a 옵션(또는 level 또는 one\_liner 옵션)의 피연산자에서 가져옵니다. 어떤 수준에서 목록을 표시할 것인지 및 제품 이름과 함께 어떤 소프트웨어 속성을 표시할 것인지를 결정해야 합니다.

---

## 참고

---

다음에 나오는 예에는 one\_liner 옵션에 대한 값은 포함되지 않습니다.

**제품 수준 지정** 소프트웨어 선택 사항에 수준을 지정하면 swlist는 해당 수준의 모든 객체와 그보다 **상위** 수준에 있는 모든 객체를 나열합니다. 상위 수준은 # 기호로 주석 처리됩니다. 따라서 지정된 수준(제품, 하위 제품, 파일 세트 또는 파일)만 주석 처리가 되지 않습니다. 이렇게 하면 swlist의 출력이 다른 명령의 입력으로 사용될 수 있습니다. 그러나 다음과 같은 예외가 있습니다.

- 1) 파일만 들어 있는 목록. 파일 수준의 출력은 다른 명령의 입력으로 받아들여지지 않습니다.
- 2) 소프트웨어 속성이 들어 있는 목록(-a 및 -v).

예를 들어 로컬 호스트에 설치된 **제품** 을 모두 보려면 다음과 같은 명령을 사용합니다.

```
swlist -l product
```

목록은 다음과 같을 것입니다.

```
NETWORKING  
SAM  
OPENVIEW  
PRODUCT A  
SOFTWARE Z  
PRODUCT B  
.  
.  
.
```

제품 이름이 주석 처리되지 않았는데, 이는 이것이 사용자가 표시를 요청한 수준이고 그보다 상위 수준이 없기 때문입니다.

**하위 제품 수준 지정** 이 예의 로컬 호스트에서 NETWORKING 제품에는 ARPA와 NFS라는 하위 제품이 들어 있으며 사용자는 각 객체의 크기(KB)를 보려고 합니다.

```
swlist -l subproduct -a size NETWORKING
```

```
# NETWORKING          9072  
  NETWORKING.ARPA     4412  
  NETWORKING.NFS      4660
```

이 목록에는 파일이나 파일 세트가 표시되지 않았는데, 그 이유는 명령줄에서 사용자가 해당 수준을 지정하지 않았기 때문입니다.

로컬 호스트에서 NETWORKING 제품의 이름과 개정본 번호를 보려면 다음과 같은 명령을 사용합니다.

```
swlist -l subproduct -a revision NETWORKING
```

제품 이름은 항상 포함되므로 -a 옵션에 지정할 필요가 없습니다.

**파일 세트 수준 지정** -l 옵션을 사용하여 로컬 호스트에서 NETWORKING 제품에 대한 모든 파일 세트와 각각에 대한 설명적 제목을 포함하는 목록을 생성하는 예는 다음과 같습니다.

```
swlist -l fileset -a title NETWORKING
```

```
# NETWORKING                Network Software
  NETWORKING.ARPA-INC        ARPA include files
  NETWORKING.ARPA-RUN        ARPA run-time commands
  NETWORKING.ARPA-MAN        ARPA manual pages
  NETWORKING.LANLINK         CORE ARPA software
  NETWORKING.NFS-INC         NFS include files
  NETWORKING.NFS-RUN         NFS run-time commands
  NETWORKING.NFS-MAN         NFS manual pages
```

주석 처리된 줄(#)은 하위 제품(NETWORKING.ARPA 및 NETWORKING.NFS)과 제품(NETWORKING) 수준의 항목이라는 것을 다시 한 번 유념해 주십시오. 다른 줄은 파일 세트입니다.

**파일 수준 지정** -l 옵션을 사용하는 다음 예에서는 NETWORKING.ARPA 하위 제품에 대한 모든 파일을 포함하는 광범위한 목록을 생성합니다.

```
swlist -l file NETWORKING.ARPA
```

```
# NETWORKING.ARPA
# NETWORKING.ARPA_INC
  NETWORKING.ARPA_INC:/usr/include/arpa/ftp.h
  NETWORKING.ARPA_INC:/usr/include/arpa/telnet.h
  NETWORKING.ARPA_INC:/usr/include/arpa/tftp.h
  NETWORKING.ARPA_INC:/usr/include/protocols/rwhod.h
.
.
.
# NETWORKING.ARPA_RUN
  NETWORKING.ARPA_RUN:/etc/freeze
  NETWORKING.ARPA_RUN:/etc/ftpd
  NETWORKING.ARPA_RUN:/etc/gated
  NETWORKING.ARPA_RUN:/etc/named
.
.
.
# NETWORKING.ARPA_MAN
  NETWORKING.ARPA_MAN:/usr/man/man8/ftpd
  NETWORKING.ARPA_MAN:/usr/man/man8/gated
```



주석 처리된 줄은 요청된 수준(**NETWORKING.ARPA**)과 지정된 파일 수준에서 한 단계 상위 수준(파일 세트)입니다. (**NETWORKING.ARPA\_INC**, **NETWORKING.ARPA\_RUN** 및 **NETWORKING.ARPA\_RUN**은 모두 파일 세트입니다.) 주석 처리되지 않은 줄은 파일입니다.

**저장소 목록** **swlist**가 표시할 수 있는 또 다른 객체 클래스는 저장소 목록입니다. 이를 통해 호스트에 있는 등록 저장소를 모두 나열할 수 있습니다. 이를 위해 **-l depot** 옵션을 조합하여 사용할 수 있습니다.

표 3-5

저장소 나열

swlist 구문	결과
<code>swlist -l depot</code>	로컬 호스트에 있는 저장소를 모두 나열합니다.
<code>swlist -l depot @ hostA</code>	<b>hostA</b> 에 있는 저장소를 모두 나열합니다.
<code>swlist -l depot -v @ hostB</code>	<b>hostB</b> 에 있는 모든 저장소를 상세 모드로 나열합니다.

**상세 목록** **-v** 옵션을 사용하여 상세 목록을 생성할 수 있습니다. 상세 목록은 제품, 하위 제품, 파일 세트 또는 파일에 대한 속성을 모두 표시할 때 사용됩니다.

상세 출력에서는 이름(키워드)과 함께 각 속성이 나열됩니다. 속성은 한 줄에 하나씩 나열됩니다. 이 목록의 길이가 지정되면 **grep**이나 **sed**를 통해 결과를 사후 처리(필터링)하여 특정 필드를 볼 수 있습니다.

**swlist** 명령에 지정한 소프트웨어 제품 이름에 기초하여 특정 소프트웨어 수준의 속성이 표시됩니다. 예를 들어 **swlist -v NETWORKING**은 다음과 같은 결과를 표시합니다.

```
tag                NETWORKING
instance_id       7869
control_directory
size              9072
revision          2.1
title             Network Software
mod_time
directory
```

```

vendor.information  Hewlett-Packard Company
is_locatable        true
architecture        HP-UX_9000
machine_type        9000
os_name              HP-UX
target.os_release    B.11.00*
    
```

-v 옵션을 -l 옵션과 함께 사용하는 경우 다음과 같은 작업을 수행할 수 있습니다.

- 번들에 대한 속성을 모두 표시하려면 `swlist -v -l bundle`을 사용합니다.
- 제품에 대한 속성을 모두 표시하려면 `swlist -v -l product`를 사용합니다.
- 제품 및 하위 제품에 대한 속성을 모두 표시하려면 `swlist -v -l subproduct`를 사용합니다.
- 제품, 하위 제품 및 파일 세트에 대한 속성을 모두 표시하려면 `swlist -v -l fileset`를 사용합니다.
- 제품, 하위 제품, 파일 세트 및 파일에 대한 속성을 모두 표시하려면 `swlist -v -l file`을 사용합니다.

아래의 표에서는 `swlist`가 표시할 수 있는 종류의 샘플 속성 목록을 보여줍니다. 각 소프트웨어 수준이나 객체에 대해 이들 속성이 모두 존재하는 것은 아닙니다. 이 목록은 업체에서 제공하는 정보에 따라 변경될 수도 있습니다. 이 목록을 모든 속성에 대한 공식적인 목록으로 사용하지 마십시오. 특정 수준이나 객체에 대한 전체 속성 목록을 구하려면 다음 형식을 사용합니다.

```
swlist -v -l level
```

(위의 예 참조) 또는 다음을 사용합니다.

```
swlist -v software_selections
```

(아래 예 참조)

표 3-6

샘플 속성

속성	설명
architecture	제품에서 지원되는 대상 시스템을 설명
category	소프트웨어 유형
copyright	객체에 대한 저작권 정보
mod_time	배포 미디어를 위한 실제 수행 시간

표 3-6 샘플 속성(계속)

속성	설명
description	객체에 대한 자세한 설명 정보
instance_id	이 소프트웨어 제품을 고유하게 식별
title	객체에 대한 긴 공식적 이름
mode	파일에 대한 권한 모드
mtime	파일의 최종 수정 시간
owner	파일의 소유자(문자열)
path	파일에 대한 전체 경로 이름
corequisite	현재의 파일 세트(configured)가 작동하기 위해 필요로 하는 파일 세트
prerequisite	현재의 파일 세트가 올바르게 설치되거나 구성되기 위해 필요한 파일 세트
readme	일반적인 readme와 유사한 정보, 릴리즈 정보 등.
revision	객체의 수정 번호
size	포함된 모든 파일 세트의 크기를 나타내는 크기(바이트)
state	파일 세트의 현재 상태

다음은 상세 목록의 예입니다.

로컬 호스트에서 다음 명령을 입력합니다.

```
swlist -v -l file NETWORKING.ARPA-RUN
```

그러면 다음과 같은 목록이 생성됩니다.

```
#NETWORKING.ARPA
tag:          ARPA-RUN
instance_id   1
revision      1.2
title         ARPA run_time commands
size          556
state         configured
```

```
corequisite NETWORKING.LANLINK
is_kernel    false
file         etc/freeze
path         /etc/freeze
type         f
mode         0755
owner        bin
group        bin
uid          2
gid          2
mtime        721589735
size         24
file         etc/ftpd
path         /etc/ftpd
type         file
mode         0555
owner        bin
group        bin
uid          2
gid          2
mtime        721589793
size         9
...
```

다음 명령을 입력합니다.

```
swlist -v NETWORKING.ARPA-RUN
```

그러면 다음과 같은 목록이 생성됩니다.

```
# NETWORKING.ARPA
fileset
tag          ARPA-RUN
instance_id  1
revision     1.2
title        ARPA run_time commands
size         556
state        configured
corequisite NETWORKING.LANLINK
is_kernel    true
mod_time     733507112
```

## IPD 수정(swmodify)

SD-UX는 설치된 소프트웨어에 대해서는 IPD(Installed Products Database)를 사용하고 저장소에 있는 소프트웨어에 대해서는 카탈로그 파일을 사용하여 시스템에 있는 소프트웨어 설치, 제품 및 파일 세트를 추적합니다.

IPD와 카탈로그 파일은 다른 SD-UX 기능(swinstall, swcopy 및 swremove)에 의해 작성되고 지속적으로 수정되지만 그 안에 들어 있는 정보를 변경하려는 경우 직접 액세스할 수는 없습니다. IPD나 저장소의 카탈로그 파일 중 하나에 들어 있는 정보를 편집해야 하는 경우 swmodify 명령을 사용해야 합니다.

swmodify 명령은 소프트웨어 저장소, 기본 루트 또는 대체 루트에 정의된 소프트웨어 객체나 속성을 추가, 수정 또는 삭제합니다. 이것은 저장소의 카탈로그 파일이나 루트의 IPD(Installed Products Database)에 대한 직접적인 인터페이스입니다. 이 명령은 객체를 구성하는 파일은 변경하지 않고 그 객체를 **설명하는** 정보만 조작합니다.

swmodify를 사용하여 다음을 수행할 수 있습니다.

- 기존 객체에 새로운 번들, 제품, 하위 제품, 파일 세트, 제어 스크립트 또는 파일 정의를 추가합니다.
- 저장소 카탈로그 파일 또는 루트 IPD에서 소프트웨어 객체에 대한 설명을 제거합니다.
- 기존 객체의 속성 값을 변경합니다.
- 추가하는 새로운 객체에 대한 속성을 정의합니다.

저장소의 경우 IPD 파일에 해당되는 것이 카탈로그 파일입니다. swcopy를 사용하여 저장소를 작성하거나 변경하면 저장소와 해당 내용을 설명하는 카탈로그 파일이 만들어집니다(기본 위치: /var/spool/sw/catalog).

## IPD 내용

IPD는 /var/adm/sw/products 디렉토리에 위치하며 루트 디렉토리(/) 아래에 설치된 모든 제품에 대한 정보를 포함하는 파일 및 하위 디렉토리입니다. 이 정보에는 설치된 각 제품에 대한 “태그” 또는 제품 이름, 한 줄짜리 제목 필드, 단락 길이의 긴 설명 텍스트, 긴 README 파일, 저작권 정보, 업체 정보 및 부품 번호 등이 포함됩니다. 또

한 IPD에는 수정 정보 및 네 가지 **uname** 속성(운영 체제 이름, 릴리즈, 버전 및 하드웨어 시스템 유형)을 포함한 사용자를 대상으로 하는 아키텍처 필드가 들어 있습니다. 다음은 “Accounting”이라는 제품에 대한 IPD INFO 파일의 형식입니다.

```
fileset
tag ACCOUNTNG
data_model_revision 2.4
instance_id 1
control_directory ACCOUNTNG
size 292271
revision B.11.00
description Vendor Name: Hewlett-Packard Company
Product Name: Accounting
Fileset Name: ACCOUNTING

Text: "HP-UX System Accounting feature set. Use these
features to gather billing data for such items as disk space
usage, connect time or CPU resource usage.
"

timestamp 797724879
install_date 199504121614.39
install_source hpfcfc.fc.hp.com:/release/11.00_gsL/goodsystem
state configured
ancestor HPUX10.20.ACCOUNTNG
corequisite OS-Core.CMDS-MIN,r>=B.11.00,a=HP-UX_B.11.00_32/64,fa=HP-UX_B.11.00_32/64,v=HP
```

카탈로그 파일은 IPD 파일에 해당되지만 저장소에 저장된 소프트웨어에 대한 내용을 담고 있습니다. swcopy를 사용하여 저장소를 작성하고 수정할 때 카탈로그 파일이 지정된 저장소(또는 기본적인 /var/spool/sw 저장소)에서 작성되고 보관됩니다. 카탈로그 파일은 저장소와 그 내용을 설명합니다.

swinstall, swconfig, swcopy 및 swremove 작업은 명령이 실행될 때 자동으로 IPD 및 카탈로그 파일 정보를 추가, 변경 및 삭제합니다. swlist 및 swverify 작업은 IPD 정보를 읽고 이 정보를 사용하여 명령 작동 방식에 영향을 미칩니다.

IPD에는 소프트웨어 객체에 대한 읽기/쓰기 동시 액세스를 관리하는 swlock 파일도 포함됩니다.

## swmodify 사용

### 구문

```
swmodify [-d] [-p] [-r] [-u] [-v [-V] [-a attribute=[value]]
[-c catalog][-C session file] [-f software_file]
[-P pathname_file] [-s product_specification_file]
[-S session_file] [-x option=value][-X option_file]
[software_selections] [@ target_selection]
```

### 옵션 및 피연산자

- d                    기본 루트나 대체 루트가 아닌 저장소에서 변경 작업을 수행합니다. *target\_selection*이 저장소여야 합니다.
- p                    *target\_selection* 내에서 아무 것도 변경하지 않고 수정 세션을 미리 봅니다.
- r                    기본 루트가 아닌 대체 루트에서 변경 작업을 수행합니다. *target\_selection*이 대체 루트여야 합니다.
- u                    -a *attribute* 옵션이 지정되지 않으면 *target\_selection* 내에서 지정된 *software\_selections*를 삭제합니다. 이 작업을 수행하면 저장소 카탈로그나 IPD(Installed Products Database)로부터 소프트웨어 객체 정의가 삭제됩니다.  
  
-a *attribute* 옵션이 지정되면 해당 *target\_selection* 내에서 소프트웨어 객체 정의를 삭제합니다.
- v                    stdout으로의 상세 출력을 활성화합니다. (swmodify 로그 파일은 이 옵션의 영향을 받지 않습니다.)
- V                    이 명령이 지원하는 모든 SD *layout\_versions*를 나열합니다.
- a *attribute=value*  
**속성 값**을 추가, 변경 또는 삭제합니다. 또는 각 *software\_selection*에 대해 속성을 지정된 *value*로 설정하여 속성을 추가하거나 변경합니다.  
  
-a 옵션을 여러 개 지정할 수 있습니다. 각 속성 변경은 모든 *software\_selection*에 적용됩니다.  
  
-s와 -a 옵션은 상호 배타적입니다. 즉 -a 옵션을 지정하면 -s 옵션은 지정할 수 없습니다.

-a 옵션을 사용하는 경우 *tag*, *revision*, *instance\_id*, *vendor\_tag*, *corequisite* 또는 *prerequisite* 등의 속성은 변경할 수 없습니다.

-c *catalog* *catalog*로 지정된 디렉토리에 자세한 카탈로그 구조 정보를 기록합니다. 파일 수준 및 제어 스크립트에 이르는 모든 속성이 기록됩니다. 자세한 내용은 385페이지의 “사용자 응답 요구(*swask*)”를 참조하십시오.

-C *session\_file* 명령을 실행하고 현재 옵션과 피연산자 값을 다른 세션에 다시 사용할 수 있도록 *session\_file*에 저장합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.

-f *software\_file* 명령줄 대신(또는 그에 추가하여) 별도의 파일에서 대상 선택 목록을 읽습니다. 자세한 내용은 60페이지의 “소프트웨어 파일”을 참조하십시오.

-P *pathname\_file* IPD에 추가되거나 IPD에서 삭제되는 파일의 경로 이름이 들어 있는 파일을 지정합니다.

-s *product\_specification\_file*

소스 PSF(Product Specification File)는 *swmodify*에 의해 추가되거나 수정될 제품, 하위 제품, 파일 세트 또는 파일의 정의를 설명합니다.

*product\_specification\_file*을 지정한 경우 *swmodify*는 PSF에 정의된 전체 세트에서 개별 *software\_selections*를 선택합니다. *software\_selections*를 지정하지 않으면 *swmodify*는 PSF에 정의된 소프트웨어를 모두 선택합니다. PSF에서 선택된 소프트웨어는 *target\_selection*에 적용되고, 선택된 소프트웨어 객체는 그에 추가되거나, 변경되거나 삭제됩니다.

PSF를 지정하지 않으면 *software\_selections*를 지정해야 합니다. *swmodify*는 지정된(또는 기본) *target\_selection*에 정의된 소프트웨어에서 *software\_selections*를 선택합니다.



swmodify용 PSF는 285페이지의 “PSF(Product Specification File) 만들기”에 정의된 것과 동일한 swpackage PSF 형식을 사용합니다.

*-S session\_file*

*session\_file*에 정의된 대로 이전 설치 세션에서 저장된 값을 바탕으로 명령을 실행합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.

*-x option=value*

명령 *option*을 *value*로 설정하고 기본값 또는 옵션 파일의 값을 대체합니다. 자세한 내용은 122페이지의 “명령 옵션 변경”을 참조하십시오.

*-X option\_file*

*option\_file*에서 세션 옵션 및 기능을 읽습니다. 자세한 내용은 122페이지의 “명령 옵션 변경”을 참조하십시오.

*software\_selections*

정보가 수정될 소프트웨어 객체입니다. 자세한 내용은 58페이지의 “소프트웨어 선택 사항”을 참조하십시오.

*target\_selection*

단일 로컬 *target\_selection*입니다(60페이지의 “대상 선택 사항” 참조). 기본 루트에서 작동 중이면 대상으로 /를 가정하므로 *target\_selection*을 지정할 필요가 없습니다.

소프트웨어 저장소에서 작동 중이라면 *target\_selection*에 해당 저장소로의 경로를 지정해야 합니다. *-a* 옵션이 지정되고 *target\_selection*이 지정되지 않으면 기본값인 *depot\_directory*로 가정합니다.

---

## 참고

일반적으로 *-u* 옵션을 *-a* 옵션과 함께 사용할 때는 주의해야 합니다. *-u*를 사용할 때 *-a*도 함께 지정하면 *-a* 옵션은 해당 *software\_selections*에서 **속성**을 삭제합니다(또는 **속성**에 대해 현재 정의된 값 세트에서 **값**을 삭제합니다).

---

### 명령 옵션 변경

명령을 호출할 때 명령줄 옵션을 추가로 지정(-x 옵션 사용)하거나 파일에서 미리 정의된 값을 읽어서 이 명령의 작동 방식을 변경할 수 있습니다. 다음 표에서는 **swmodify**에 적용되는 기본값과 옵션을 보여줍니다.

표 3-7 **swmodify** 명령 옵션 및 기본값

- `admin_directory=/var/adm/sw`
- `compress_index=false`
- `control_files=`
- `distribution_target_directory=/var/spool/sw`
- `files=`
- `installed_software_catalog=products`
- `layout_version=1.0`
- `log_msgid=0`
- `logdetail=false`
- `logfile=/var/adm/sw/swmodify.log`
- `loglevel=1`
- `patch_commit=false`
- `run_as_superuser=true`
- `software=`
- `source_file=`
- `targets=`
- `verbose=1`

### 추가 정보

각 기본값에 대한 자세한 설명은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.

### swmodify 작업 및 예

다음은 **swmodify**를 사용하여 카탈로그 파일이나 IPD를 변경하는 예입니다.

#### IPD에 정보 추가

`/tmp/a`, `/tmp/b` 및 `/tmp/c` 파일의 설명을 기존 파일 세트에 추가하려면 다음과 같이 입력합니다.

```
swmodify -x files=/tmp/a /tmp/b /tmp/c PRODUCT.FILESET
```

제어 스크립트가 설치된 파일 시스템에 새로운 파일을 추가할 때 제어 스크립트는 **swmodify**를 사용하여 새 파일을 기록할 수 있습니다.

#### 기존 IPD 정보 변경

기존 저장소의 제품에 대해 새로운 번들 정의를 작성하려면 다음과 같이 입력합니다.

```
swmodify -d -s new_bundle_definitions \  
  \* @ /mfg/master_depot
```

제품의 구성 프로세스가 더 복잡한 경우 구성이 정상적으로 완료될 때까지 스크립트는 파일 세트의 상태를 `configured`로 설정할 수 있습니다.

파일 세트의 속성 값을 변경하려면 다음과 같이 입력합니다.

```
swmodify -a state=installed PRODUCT.FILESET
```

저장소의 속성을 변경하려면 다음과 같이 입력합니다.

```
swmodify -a title=Master Depot \  
          -a description=/tmp/mfg.description \  
          @ /mfg/master_depot
```

### 새로운 객체 정의

제품을 설명하는 간단한 PSF(Product Specification File)를 만들고 기존 응용 프로그램(SD-UX가 설치하지 않음)을 가져온 다음 swmodify를 호출하여 해당 정의를 IPD로 로드할 수 있습니다.

새로운 파일 세트 정의를 작성하려면(PSF에 파일 정의가 들어 있으면 해당 파일을 새 파일 세트에 추가함) 다음과 같이 입력합니다.

```
swmodify -s new_fileset_definition
```

---

## 설치된 소프트웨어 제거(swremove)

swremove 명령은 호스트에 설치된 소프트웨어를 제거합니다. 제거하기 전에 먼저 소프트웨어의 구성을 해제합니다. swremove는 소프트웨어 저장소에 복사되어 있는 소프트웨어 제품도 제거합니다.

### swremove 기능 및 제한 사항

- 지정된 위치에서 파일을 제거합니다. 심볼릭 링크는 제거되지만 심볼릭 링크의 대상은 제거되지 않습니다. 제거되지 않은, 사용 중인 파일도 나열합니다.
- 종속성 자동 사용을 통해 작동될 소프트웨어를 자동으로 선택합니다(직접 지정한 소프트웨어에 추가하여).
- 제거 작업의 일부로 다음과 같은 제어 스크립트를 실행합니다.

**Unconfigure** 구성 스크립트에 의해 수행된 호스트 구성을 취소합니다.

**Checkremove** 제거 및 구성 해제가 수행될 수 있는지 판별하기 위해 각 대상을 분석합니다. 이 검사에 실패하면 해당 객체는 제거할 수 없습니다.

**Preremove** 사전 설치 스크립트에 의해 작성된 파일을 제거하는 등 추가적인 파일 작업을 수행합니다.

**Postremove** 파일 세트나 제품이 제거된 직후에 추가적인 제거 작업("롤백"된 파일을 복원하는 등)을 수행합니다.

자세한 내용은 11장, 347페이지의 "제어 스크립트 사용"을 참조하십시오.

- swremove는 대체 루트에서 소프트웨어를 제거할 경우에는 자동 구성 해제를 수행하지 않습니다.

## swremove GUI 사용

이 절에서는 swremove GUI에 대한 개요를 설명합니다.

- 일반적으로 124페이지의 “설치된 소프트웨어 제거(swremove)”의 모든 정보는 swinstall GUI에도 적용됩니다.
- 이 절에서는 37페이지의 “GUI 및 TUI 명령 사용”에 설명된 표준 GUI 요소에 대한 추가 정보를 참조합니다.
- 이 절의 모든 정보는 달리 명시되지 않은 한 TUI 프로그램에도 적용됩니다. 자세한 내용은 37페이지의 “터미널 사용자 인터페이스”를 참조하십시오.

swremove 명령은 기본 루트 파일 시스템, 대체 루트 파일 시스템 및 저장소에 따라 제거 시 약간 다르게 작동합니다. 저장소 제거 작업에 대한 인터페이스 변경 사항은 164페이지의 “저장소에서 소프트웨어 제거”에 요약되어 있습니다.

제거 프로세스는 네 단계로 구성됩니다.

표 3-8

### GUI 제거 단계

I. 시작	swremove GUI를 시작합니다.
II. 소프트웨어 선택	제거할 소프트웨어를 선택합니다.
III. 분석(미리 보기)	선택된 소프트웨어가 정상적으로 제거될 수 있는지 여부를 판별하기 위해 제거를 분석(미리 보기)합니다.
IV. 제거	실제 제거를 수행합니다.

### 단계 I: 시작

제거 세션을 위해 GUI나 TUI를 시작하려면 다음과 같이 입력합니다.

```
/usr/sbin/swremove
```

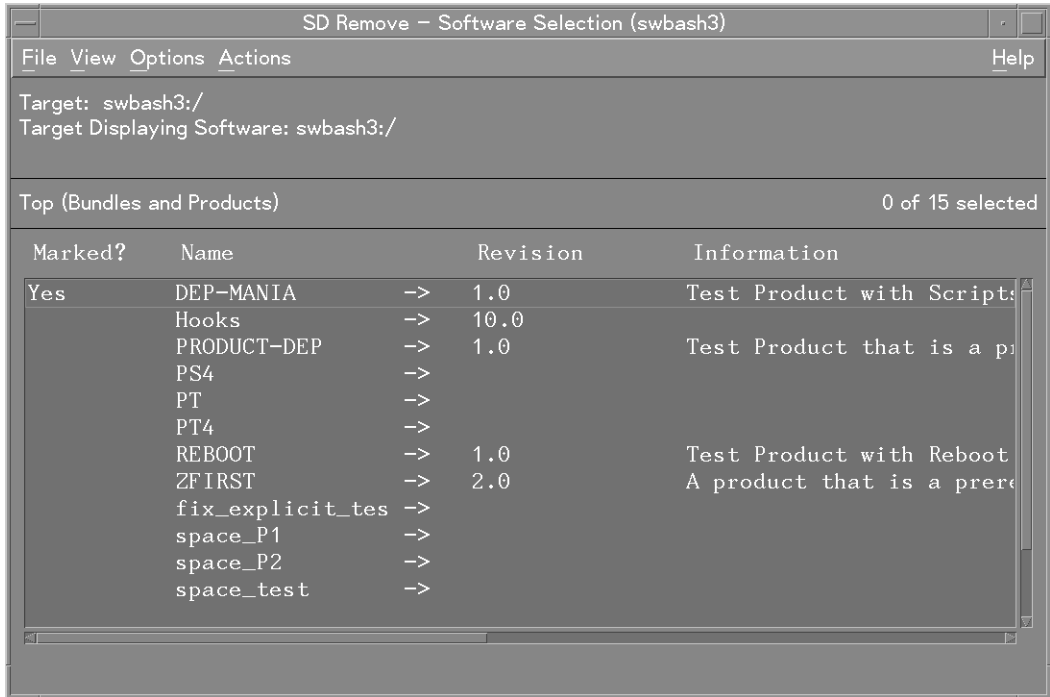
명령줄에서도 소프트웨어를 **지정하지 않은 한 GUI**가 자동으로 호출됩니다. GUI를 호출하고 소프트웨어를 지정하려면 **-i** 옵션을 지정합니다. 예를 들어 GUI를 사용하여 BUNDLE1에 대한 미리 보기(분석만 수행) 세션을 열려면 다음과 같이 입력합니다.

```
swremove -i -p /MyDepot/BUNDLE1
```

Software Selection 창이 표시됩니다.

**단계 II: 소프트웨어 선택** 이 단계에서는 Software Selection 창에서 제거하려는 소프트웨어를 선택합니다.

그림 3-2 swremove Software Selection 창



1. 객체 목록에서 소프트웨어를 선택합니다.

- a. 항목을 강조 표시합니다.
- b. **Actions** → **Mark For Remove**를 선택합니다.

— 또는 —

마우스 오른쪽 단추를 클릭하여 팝업 메뉴를 표시한 다음 **Mark For Remove**를 선택합니다.

객체 목록의 Marked? 플래그가 선택한 내용과 일치하게 Yes로 변경됩니다. (소프트웨어 객체의 일부 구성 요소만 선택한 경우 Partial 플래그가 표시될 수도 있습니다.)

- (선택) **Actions** 메뉴의 선택 사항을 사용하여 추가로 소프트웨어를 선택합니다.
  - Change Target**을 통해 소프트웨어를 제거할 대체 루트를 선택할 수 있습니다.
  - Add Software Group**을 통해 이전에 저장된 소프트웨어 선택 사항 그룹을 다시 불러 사용할 수 있습니다.
  - Save Software Group**은 표시된 소프트웨어 목록을 그룹으로 저장합니다. SD는 \$HOME/.sw/software/ 또는 사용자가 지정한 디렉토리에 그룹 정의를 저장합니다.
  - Show Description of Software**(객체 목록에서 강조 표시된 항목 하나에 대해서만 사용 가능)는 선택한 소프트웨어에 대해 자세한 정보를 표시합니다.
- 분석(미리 보기) 단계를 시작하려면 **Actions**→**Install**을 선택합니다. **Analysis** 대화 상자가 표시됩니다.

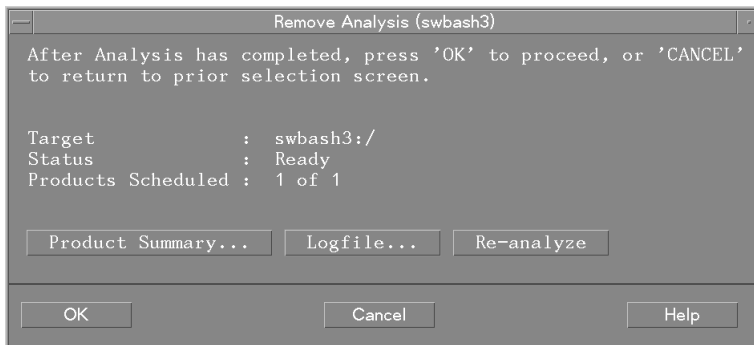
### 단계 III: 분석 (미리 보기)

이 단계에서는 선택한 소프트웨어를 SD-UX가 분석합니다.

**Remove Analysis** 대화 상자에는 분석 프로세스에 대한 상태 정보가 표시됩니다. 분석이 완료되고 호스트 상태가 Ready가 되면 **OK**를 클릭하여 실제 설치를 시작합니다 (129페이지의 “단계 III: 제거” 참조). **Analysis** 대화 상자가 **Remove** 창으로 바뀝니다. (미리 보기 세션을 시작했으면 분석 후 제거가 중단됩니다. **OK**를 클릭하면 **Software Selection** 창으로 돌아갑니다.)

그림 3-3

### Remove Analysis 대화 상자



분석 후 선택된 소프트웨어 중 하나라도 제거될 수 있으면 상태는 Ready 또는 Ready with Warnings를 나타냅니다. 선택된 소프트웨어 중 제거될 수 있는 것이 하나도 없는 경우 상태는 Excluded from task를 나타냅니다.

Products Scheduled 열에서는 선택된 제품 전체에서 제거될 준비가 된 제품의 수를 보여줍니다. 전체 제품에는 다음과 같은 제품들이 포함됩니다.

- 종속성으로 인해 표시된 제품
- 번들 내에서 표시된 제품
- 부분적/전체적으로 표시된 제품

해당 제품에 오류가 발생하면 자동으로 제거에서 제외됩니다. 호스트 대상이 제거에서 제외되는 경우 제거를 수행할 수 없습니다. 호스트가 분석에 실패하면 경고 대화 상자가 표시됩니다.

다음과 같은 작업을 수행할 수 있습니다.

- **Product Summary**에서는 제품이나 번들에 대한 추가 정보를 제공하며 종속성, 저작권, 공급업체 등에 대한 추가 정보를 표시하는 **Product Description** 단추가 있습니다.

Projected Action 열에서는 어떤 유형의 제거가 수행될 것인지 설명합니다. 가능한 유형은 다음과 같습니다.

**Remove**                    제품이 존재하고 제거됩니다.

**Filesets Not Found**

시스템이 지정된 파일 세트를 찾지 못했습니다.

**Skipped**                    제품이 제거되지 않습니다.

**Excluded**                    분석 단계에서 발생한 오류로 인해 제품이 제거되지 않습니다. 오류에 대한 자세한 내용은 로그 파일을 참조하십시오.

(**Product Summary List**는 객체 목록이 아닙니다. 제품을 열거나 작업을 수행하거나 열 보기를 변경할 수 없습니다.)

- **Logfile**에서는 로그 파일에 기록된 자세한 제거 정보를 여러 화면에 걸쳐 보여줍니다.
- **Re-analyze**는 분석 프로세스를 반복합니다.



### 단계 III: 제거

이 단계에서는 SD-UX가 실제 제거를 수행합니다.

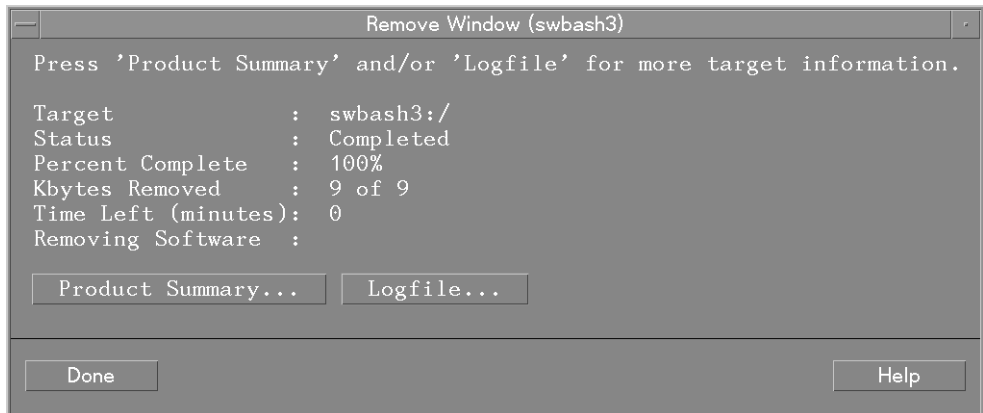
Analysis 창에서 **OK**를 클릭하면 SD-UX는 제거를 시작하고 상태 정보를 보여 주는 Remove 창(그림 3-4, “Remove 창”)을 표시합니다.

다음과 같은 단추를 사용할 수 있습니다.

- **Done**을 클릭하면 **Software Selection** 창으로 돌아갑니다. 그러면 다른 제거를 시작하거나 GUI를 종료(**File**→**Exit**)할 수 있습니다.
- **Product Summary**는 제거 및 제품 정보(이름, 버전, 제거 결과, 제거 요약)를 표시합니다.
- **Logfile**을 클릭하면 로그 파일을 표시합니다.

그림 3-4

### Remove 창



## 명령줄을 사용한 제거

### 구문

```
swremove [XToolkit Option] [-d|-r] [-i] [-p] [-v]
[-C session_file] [-f software_file] [-Q date] [-s source]
[-S session_file] [-t target_file] [-x option=value]
[-X option_file] [software_selections] [@ target_selections]
```

### 옵션 및 피연산자

<b>XToolkit 옵션</b>	GUI용 X window 옵션입니다. 자세한 내용은 55페이지의 “XToolkit 옵션 및 디스플레이 글꼴 변경”을 참조하십시오.
-d	설치된 소프트웨어가 아닌 저장소에 대해 작동합니다. 자세한 내용은 164페이지의 “저장소에서 소프트웨어 제거”를 참조하십시오.
-i	GUI 또는 TUI 대화식 세션을 실행합니다. GUI/TUI에서 사용하기 위해 소프트웨어 선택 사항을 “미리 지정”하는 데 사용됩니다.
-p	분석 단계를 통해 제거 작업을 실행한 다음 종료함으로써 제거 작업을 미리 봅니다.
-r	대체 루트 디렉토리에서 작동합니다.
-v	stdout으로의 상세 출력을 활성화하고 모든 활동을 화면에 표시합니다.
-C session_file	명령을 실행하고 현재 옵션과 피연산자 값을 다른 세션에 다시 사용할 수 있도록 session_file에 저장합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.
-f software_file	명령줄 대신(또는 그에 추가하여) 별도의 파일에서 대상 선택 목록을 읽습니다. 자세한 내용은 60페이지의 “소프트웨어 파일”을 참조하십시오.
-Q date	원격 작동을 수행하는 경우 해당 날짜에 작업이 실행되도록 일정을 예약합니다. 211페이지의 “명령줄에서 작업 예약” 및 6장, 167페이지의 “원격 작업 개요”를 참조하십시오.
-S session_file	session_file에 정의된 대로 이전 제거 세션에서 저장된 값을 바탕으로 명령을 실행합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.

*-t target\_file*

명령줄 대신(또는 그에 추가하여) 별도의 파일에서 대상 선택 목록을 읽습니다. 자세한 내용은 61페이지의 “대상 파일”을 참조하십시오.

*-x option=value*

명령 *option*을 *value*로 설정하고 기본값 또는 옵션 파일의 값을 대체합니다. 자세한 내용은 133페이지의 “명령 옵션 변경”을 참조하십시오.

*-X option\_file*

*option\_file*에서 세션 옵션 및 기능을 읽습니다. 자세한 내용은 133페이지의 “명령 옵션 변경”을 참조하십시오.

*software\_selections*

제거될 소프트웨어 객체입니다. 자세한 내용은 58페이지의 “소프트웨어 선택 사항”을 참조하십시오.

*target\_selections*

명령의 대상입니다. 자세한 내용은 60페이지의 “대상 선택 사항”을 참조하십시오.

### 명령 옵션 변경

명령을 호출할 때 명령줄 옵션을 추가로 지정(-x 옵션 사용)하거나 파일에서 미리 정의된 값을 읽어서 이 명령의 작동 방식을 변경할 수 있습니다. 다음 표에서는 **swremove**에 적용되는 기본값과 옵션을 보여줍니다.

표 3-9 **swremove** 명령 옵션 및 기본값

- `admin_directory=/var/adm/sw`
- `agent_auto_exit=true`
- `agent_timeout_minutes=10000`
- `auto_kernel_build=true`
- `autoreboot=false`
- `autoremove_job=false`
- `autoselect_dependents=false`
- `autoselect_reference_bundles=true`
- `compress_index=false`
- `controller_source=`
- `distribution_target_directory=/var/spool/sw`
- `enforce_dependencies=true`
- `enforce_scripts=true`
- `force_single_target=false`
- `installed_software_catalog=products`
- `job_title=`
- `log_msgid=0`
- `logdetail=false`
- `logfile=/var/adm/sw/swremove.log`
- `loglevel=1`
- `mount_all_filesystems=true`
- `polling_interval=2`
- `preview=false`
- `remove_empty_depot=true`
- `reuse_short_job_numbers=true`
- `rpc_binding_info=ncacn_ip_tcp:[2121]ncadg_ip_udp:[2121]`
- `rpc_timeout=5`
- `run_as_superuser=true`
- `select_local=true`
- `software=`
- `software_view=products`
- `targets=`
- `verbose=1`
- `write_remote_files=false`

### 추가 정보

각 기본값에 대한 자세한 설명은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.

### 제거 작업 및 예

로컬 호스트의 기본 저장소에서 **MYSOFT**라는 소프트웨어 제품을 제거하려면 다음과 같이 입력합니다.

```
swremove -d MYSOFT
```

로컬 호스트에 설치되어 있는 C와 Pascal 제품의 제거를 미리 보려면 다음과 같이 입력합니다.

```
swremove -p cc pascal
```

HP Omniback 특정 버전을 제거하려면 다음과 같이 입력합니다.

```
swremove Omniback,l=/opt/Omniback_v2.0
```

로컬 저장소의 내용 전체를 제거하려면 다음과 같이 입력합니다.

```
swremove -d * @ /var/spool/sw
```

### 번들 제거

번들을 제거해도 번들 내의 파일 세트가 모두 제거되는 것은 아닙니다. SD-UX의 종속성 관리 기능으로 인해 다른 번들에서 필요로 하는 파일 세트는 제거되지 않기 때문입니다. 예를 들어 Pascal과 FORTRAN이 모두 Debugger.Run 파일 세트를 사용하는데 FORTRAN을 제거하려 한 경우, Debugger.Run 파일 세트는 Pascal 번들에서도 사용되기 때문에 제거되지 않습니다. 이는 하나의 번들을 제거할 때 실수로 다른 번들에서 필요로 하는 파일 세트가 제거되지 않도록 합니다.

### 패치 제거

다음과 같은 경우를 제외하고는 패치 소프트웨어를 제거할 수 없습니다.

- 다시 설치할 때 패치에 대응하는 롤백 파일을 사용할 수 있는 경우

— 또는 —

- 패치에 의해 수정된 기본 소프트웨어가 동시에 제거되는 경우 (기본 소프트웨어를 제거하면 그와 연관된 패치도 제거됩니다.)

패치 제거에 대한 자세한 내용은 5장, 165페이지의 “HP-UX 패치 및 패치 관리”를 참조하십시오.

### 여러 버전 제거

swremove GUI를 사용하면 제품의 여러 버전을 제거하는 작업이 보다 간단해집니다.

객체 목록에 각각의 제품 버전과 해당 위치 디렉토리 쌍이 나열됩니다. 여러 버전을 선택하는 것은 product:/location directory 쌍을 의미하는 것입니다. 기본적으로 위치는 Software Selection 창에 표시되지 않습니다. 위치는 GUI Columns Editor View→Columns... 메뉴 항목을 선택하고 Product Directory 메뉴 항목을 사용 가능으로 설정하여 볼 수 있습니다.

선택 단계에서 하나 이상의 제품 버전을 선택할 수 있습니다. 분석 단계에서 해당 제품 버전이 대상의 다른 위치에 존재하는 경우 경고가 발생합니다. 제품이 대상에 존재하면 제품은 제거됩니다. 대상에 존재하지 않으면 그 제품은 생략됩니다. **Remove Analysis** 창의 **Product Summary...** 단추를 클릭하면 제거 단계가 시작될 때 어떤 항목이 제거될 것인지에 대한 제품별 요약 설명이 표시됩니다.

(제품의 여러 버전은 저장소에서도 가능합니다. 저장소에서 제거할 때 특별한 처리나 검사가 필요하지 않습니다.)

### 대체 루트에서 소프트웨어 제거

소프트웨어는 기본 루트(/) 디렉토리에 상대적으로 또는 대체 루트 디렉토리에 상대적으로 제거될 수 있습니다. 대체 루트는 마운트되지 않고 자체적으로 운영되는 독립형 시스템의 루트로 기능할 수 있는, 루트가 아닌 위치를 말합니다. 소프트웨어 제거에 사용되는 모든 정보 파일은 루트 볼륨의 IPD가 아닌 대체 루트 아래의 **Installed Product Database**에서 검색됩니다(32페이지의 “IPD(Installed Products Database)” 참조).

## 4 소프트웨어 저장소 관리

SD-UX는 등록된 저장소에 패키징되어 저장된 소프트웨어를 사용합니다. 이 장에서는 저장소 소프트웨어를 복사, 나열, 등록, 제거 및 확인하는 작업에 대해 설명합니다.

표 4-1

### 이 장의 항목

설명하는 항목
136페이지의 “저장소 관리 명령 및 개념”
139페이지의 “소프트웨어 저장소 복사”
153페이지의 “저장소 등록 및 등록 해제(swreg)”
157페이지의 “추가적인 저장소 관리 작업 및 예”

## 저장소 관리 명령 및 개념

다음 명령은 저장소 관리 작업을 수행하는 데 도움이 됩니다.

표 4-2 저장소 관리를 위한 명령

저장소 작업	명령	자세한 정보/예
복사	swcopy	<ul style="list-style-type: none"> <li>139페이지의 “소프트웨어 저장소 복사”</li> <li>157페이지의 “패치 저장소 병합”</li> </ul>
작성	swcopy, swpackage	<ul style="list-style-type: none"> <li>10장, 279페이지의 “소프트웨어 패키지 만들기”</li> <li>158페이지의 “배포용 테이프 저장소 만들기”</li> </ul>
나열	swlist -d	<ul style="list-style-type: none"> <li>160페이지의 “등록된 저장소 나열”</li> <li>161페이지의 “저장소 내용 나열(swlist -d)”</li> </ul>
등록	swreg	<ul style="list-style-type: none"> <li>153페이지의 “저장소 등록 및 등록 해제 (swreg)”</li> </ul>
제거	swremove -d	<ul style="list-style-type: none"> <li>164페이지의 “저장소에서 소프트웨어 제거”</li> <li>164페이지의 “저장소 제거”</li> </ul>
등록 해제	swreg -u	<ul style="list-style-type: none"> <li>153페이지의 “저장소 등록 및 등록 해제 (swreg)”</li> </ul>
확인	swverify -d	<ul style="list-style-type: none"> <li>163페이지의 “저장소 확인(swverify -d)”</li> </ul>
추가 작업		<ul style="list-style-type: none"> <li>157페이지의 “추가적인 저장소 관리 작업 및 예”</li> </ul>



## 저장소 개념

저장소는 **SD-UX** 명령이 사용할 수 있는 형식으로 만들어진 특수한 유형의 디렉토리로서 소프트웨어 제품을 보관하는 데 사용됩니다. 소프트웨어를 물리적 미디어에서 직접 복사하는 **swcopy**나 저장소를 포함하는 소프트웨어 패키지를 작성하는 **swpackage**를 사용하여 저장소를 만들 수 있습니다.

네트워크에 연결된 시스템에 저장소가 있는 경우 해당 시스템은 소프트웨어에 대한 소스 역할을 하며, 네트워크상의 다른 시스템은 매번 미디어에서 소프트웨어 제품을 설치하는 대신 해당 서버에서 소프트웨어 제품을 설치할 수 있습니다.

네트워크 저장소는 미디어로부터 직접 설치하는 것에 비해 다음과 같은 장점을 제공합니다.

- 사용자가 각각 미디어를 가져오지 않고도 여러 사용자가 (네트워크를 통해) 해당 시스템에서 소프트웨어를 다운받을 수 있습니다.
- 네트워크 서버로부터 설치하는 것이 미디어로부터 설치하는 것보다 빠릅니다.
- 여러 미디어나 네트워크 서버로부터의 서로 다른 소프트웨어 제품을 하나의 저장소로 병합할 수 있습니다.

## 저장소의 종류

저장소는 대개 디렉토리상에 존재합니다. 이 소프트웨어는 특정 미디어 형식에 따라 구성된 하위 디렉토리와 파일 세트의 계층 구조에 존재합니다. 호스트에는 여러 개의 저장소가 있을 수 있습니다. 예를 들어 네트워크상의 소프트웨어 배포 서버에는 응용 프로그램 소프트웨어 저장소, 패치 소프트웨어 저장소 및 **OS** 소프트웨어 저장소 등이 있을 수 있습니다.

저장소의 종류에는 디렉토리와 테이프가 있습니다.

## 디렉토리 저장소

- 디렉토리 저장소는 파일 시스템상에서 특수한 **SD-UX** 관리 디렉토리인 `/var/spool/sw`에 저장되는 소프트웨어로 구성됩니다.
- 디렉토리 저장소는 쓰기 가능 또는 읽기 전용입니다.
- **SD-UX** 명령을 사용하여 디렉토리 저장소를 참조할 때 저장소의 최상위 디렉토리만을 참조해야 합니다. **CD-ROM** 저장소에서 이 디렉토리는 `/cdrom/mydepot`와 같은 **CD-ROM** 마운트 지점이 됩니다.

## 테이프 저장소

- 테이프(순차) 저장소는 웹과 같은 느리고 불안정한 네트워크 연결을 통해 소프트웨어를 복사하거나 설치해야 하는 경우에 유용합니다. (먼저 저장소를 로컬 호스트에 복사한 다음 로컬 저장소에서 설치합니다.)
- 테이프 저장소의 소프트웨어는 **tar** 아카이브 형식으로 지정됩니다.
- 실제 카트리지, **DAT** 및 **9트랙** 테이프에 대한 저장소는 테이프 드라이브의 장치 파일에 대한 경로에 의해 참조됩니다(예: `/dev/rmt/0m`).
- 테이프 저장소를 변경하거나 확인할 수 없습니다.
- `swpackage` 명령을 통해서만 테이프 저장소를 만들 수 있습니다. `swcopy`를 사용하여 소프트웨어를 직접 테이프에 복사할 수 없습니다. `swpackage`에 대한 자세한 내용은 10장, 279페이지의 “소프트웨어 패키지 만들기”를 참조하십시오.
- 네트워크상의 다른 호스트에서 액세스하기 전에 테이프 저장소의 소프트웨어를 디렉토리 저장소로 전송해야 합니다.
- 테이프 저장소에는 한 번에 한 명령만 액세스할 수 있습니다.

## 저장소 등록

네트워크를 통해 **SD-UX** 명령으로 저장소의 소프트웨어를 사용하려면 저장소를 등록해야 합니다. 사용하지 않으려면 저장소 등록을 해제할 수도 있습니다. 자세한 내용은 153페이지의 “저장소 등록 및 등록 해제(`swreg`)”를 참조하십시오.

---

## 소프트웨어 저장소 복사

swcopy 명령은 저장소 간에 소프트웨어를 복사합니다. **저장소에 복사된 소프트웨어는 직접 사용될 수 없습니다.** 이 소프트웨어는 설치 및 기타 SD-UX 작업을 위한 소스로서만 사용될 수 있습니다.

### swcopy 기능 및 제한 사항

- swcopy는 호환성 검사를 수행하지 않습니다.
- swcopy는 제어 스크립트를 실행하지 않습니다.
- swcopy는 디스크 공간 분석 및 조건 선택과 같은 다른 사전 설치 및 사후 설치 검사는 수행하지만, 커널 구축이나 재부팅 등은 수행하지 않습니다.
- swcopy를 사용하여 저장소를 만들거나 수정할 때 SD-UX는 저장소를 설명하는 카탈로그 파일을 자동으로 작성합니다. 카탈로그 파일은 IPD에 저장됩니다. 자세한 내용은 117페이지의 “IPD 수정(swmodify)”을 참조하십시오.
- 소프트웨어 종속성은 swcopy GUI를 통해 선택된 사항에 적용됩니다.

## swcopy GUI 사용

### 개요

이 절에서는 swcopy GUI에 대한 개요를 설명합니다.

- 일반적으로 149페이지의 “swcopy 명령줄 사용”의 모든 정보는 swcopy GUI에도 적용됩니다.
- 이 절에서는 37페이지의 “GUI 및 TUI 명령 사용”에서 설명된 표준 GUI 요소에 대한 정보도 참조합니다.
- 이 절의 모든 정보는 달리 명시되지 않은 한 TUI 프로그램에도 적용됩니다. 자세한 내용은 37페이지의 “터미널 사용자 인터페이스”를 참조하십시오.

복사 프로세스는 다음과 같은 여섯 단계로 구성됩니다.

표 4-3

### 복사 프로세스 단계

I. 시작	swcopy GUI를 시작합니다.
II. 대상 지정	소프트웨어를 복사하려는 위치를 지정합니다.
III. 소스 지정	복사될 소프트웨어의 소프트웨어 저장소 위치를 제공합니다.
IV. 선택 소프트웨어	복사하려는 소프트웨어를 선택합니다.
V. 분석 (미리 보기)	swcopy는 복사 작업이 성공적으로 이루어질 수 있을지 여부를 판별합니다.
VI. 복사	실제 소프트웨어를 복사합니다.

### 단계 I: 시작

복사 세션을 위해 GUI나 TUI를 시작하려면 다음과 같이 입력합니다.

```
/usr/sbin/swcopy
```

명령줄에서도 소프트웨어를 지정하지 않은 한 GUI가 자동으로 호출됩니다. GUI를 호출하고 소프트웨어를 지정하려면 `-i` 옵션을 지정합니다. 예를 들어 GUI를 사용하여 MyDepot에 대한 미리 보기(분석만 수행) 세션을 열려면 다음과 같이 입력합니다.

```
swcopy -i -p /MyDepot
```

Software Selection 창 위에 Specify Source 대화 상자와 Select Target Depot Path 대화 상자가 겹쳐 표시됩니다.

## 단계 III: 대상 지정

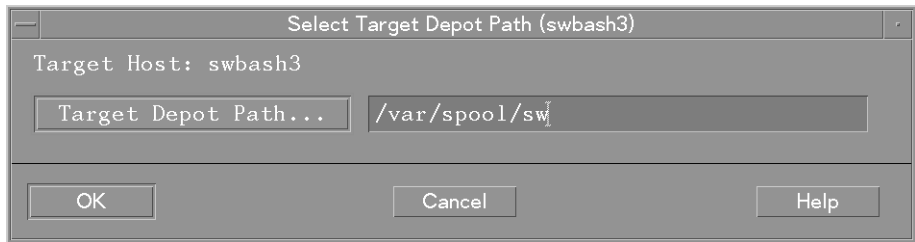
이 단계에서는 SD-UX가 소프트웨어를 복사할 대상을 지정합니다.

(GUI를 호출할 때 `-t target` 옵션을 지정한 경우 이 단계는 생략됩니다.) 149페이지의 “swcopy 명령줄 사용”을 참조하십시오.

Select Target Depot Path 대화 상자에는 기본 대상 저장소가 표시됩니다. 이 저장소는 기본 소스 저장소 경로와 일치하므로 새로운 대상을 선택해야 합니다.

그림 4-1

### Select Target Depot Path 대화 상자



1. 대상 경로를 입력합니다.

- 텍스트 상자에 새로운 대상 경로를 입력합니다.

— 또는 —

- **Target Depot Path...** 단추를 클릭합니다. Depot Paths 대화 상자가 표시되고 호스트에 등록된 저장소가 나열됩니다.
  - a. 목록에서 저장소를 클릭합니다.
  - b. **OK**를 클릭합니다. Target Depot Path 대화 상자가 없어집니다. 이제 선택한 저장소가 Select Target Depot Path 대화 상자에 표시됩니다.

2. **OK**를 클릭합니다.

Select Target Depot Path 대화 상자가 없어지고 Specify Source 대화 상자가 강조 표시됩니다.

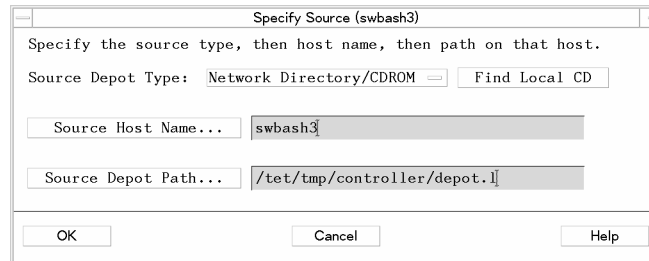
### 단계 III: 소스 지정

이 단계에서는 복사하려는 소프트웨어가 들어 있는 소스 저장소를 지정해야 합니다. **Specify Source** 대화 상자(그림 4-2, “Specify Source 대화 상자”)에 자동으로 로컬 호스트 및 기본 저장소 경로가 나열됩니다.

(GUI를 호출할 때 `-s source` 옵션을 지정한 경우 이 단계는 생략됩니다. 149페이지의 “`swcopy` 명령줄 사용”을 참조하십시오.)

그림 4-2

### Specify Source 대화 상자



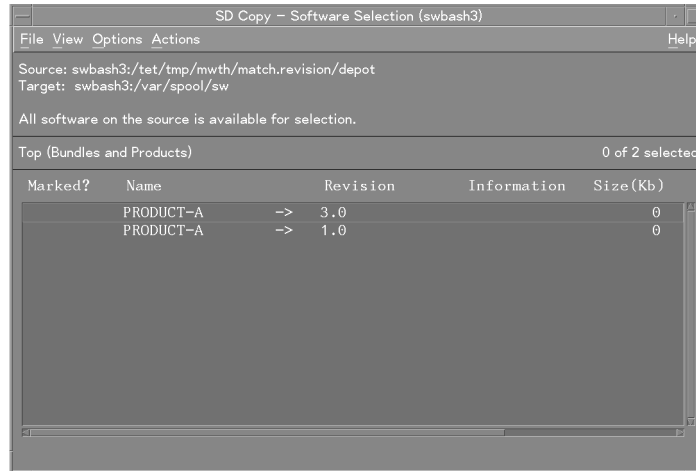
- (선택) 다른 호스트 시스템을 지정하려면 소스 호스트 이름을 입력하거나 또는,
  - Source Host Name** 단추를 클릭합니다. `defaults.hosts` 파일 (`$HOME/.sw/defaults.hosts` 또는 `/var/adm/sw/defaults.hosts`)에 들어 있는 모든 호스트 시스템 이름이 나열된 대화 상자가 표시됩니다.
  - 목록에서 호스트 이름을 선택합니다.
  - OK**를 클릭합니다. 호스트 이름이 **Specify Source** 대화 상자의 해당 상자에 표시됩니다.
- (선택) 저장소에 대한 경로를 지정하려면 새로운 경로를 입력하거나 다음을 수행합니다.
  - Source Depot Path** 단추를 클릭하여 소스 호스트에 등록되어 있는 저장소 목록을 표시합니다.
  - 저장소 중 하나를 강조 표시합니다.
  - OK**를 클릭하면 **Specify Source** 대화 상자에 표시됩니다.
- OK**를 클릭합니다. **Specify Source** 대화 상자가 닫히고 지정한 저장소에 들어 있는 소프트웨어가 **Software Selection** 창에 표시됩니다.

#### 단계 IV: 소프트웨어 선택

이 단계에서는 Software Selection 창(그림 4-3, “Software Selection 창”)에서 복사하려는 소프트웨어를 선택합니다.

그림 4-3

#### Software Selection 창



1. 객체 목록에서 소프트웨어를 선택합니다.

- a. 항목을 강조 표시합니다.
- b. **Actions**→**Mark For Copy**를 선택합니다.

— 또는 —

마우스 오른쪽 단추를 클릭하여 팝업 메뉴를 표시한 다음 **Mark For Copy**를 선택합니다.

객체 목록의 Marked? 플래그가 선택한 내용과 일치하게 Yes로 변경됩니다. (소프트웨어 객체의 일부 구성 요소만 선택한 경우 Partial 플래그가 표시될 수도 있습니다.)

2. (선택) **Actions** 메뉴의 선택 사항을 추가 선택합니다.

- **Add Software Group**에서는 이전에 저장된 소프트웨어 그룹 파일 목록을 표시하고 디렉토리를 지정할 수 있습니다. 파일을 선택하면 **Software Selection** 창에서 이미 선택한 내용에 파일에서의 소프트웨어 선택을 추가하게 됩니다.
- **Save Software Group**을 통해 현재 표시된 소프트웨어 목록을 그룹으로 저장할 수 있습니다.
- **Manage Patch Selections**를 통해 패치 목록에서 복사할 패치를 선택하고 패치용 필터를 선택하고 다른 패치 옵션을 설정할 수 있습니다.
- **Change Source...**를 클릭하면 소프트웨어 선택 사항을 취소하고 **Specify Source** 대화 상자로 돌아갑니다.
- **Add New Codeword**를 통해 보호된 소프트웨어의 잠금을 해제하기 위한 새로운 코드워드를 추가할 수 있습니다. (이 옵션은 **SD-UX**가 소스에 보호된 소프트웨어가 포함되어 있다는 것을 검색한 경우에만 사용할 수 있습니다.)
- **Show Description of Software**(객체 목록에서 강조 표시된 항목 하나에 대해서만 사용 가능)는 선택한 소프트웨어에 대해 자세한 정보를 표시합니다.
- **Change Target...**을 클릭하면 **Select Target Depot Path** 대화 상자로 돌아갑니다(141페이지의 “단계 II: 대상 지정”).

3. 분석(미리 보기) 단계를 시작하려면 **Actions**→**Copy**를 선택합니다. **Analysis** 대화 상자가 표시됩니다.



## 단계 V: 분석 (미리 보기)

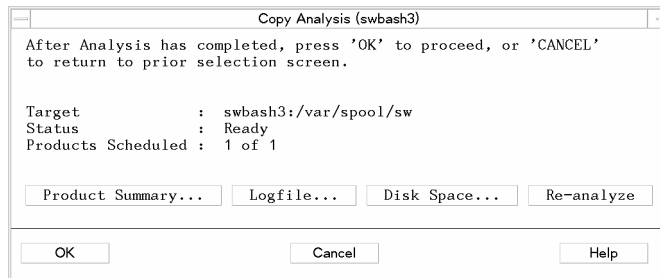
이 단계에서는 선택한 소프트웨어를 SD-UX가 분석합니다.

**Analysis** 창에 분석 프로세스에 대한 상태 정보가 표시됩니다. 분석이 완료되고 호스트 상태가 Ready가 되면 **OK**를 클릭하여 실제 복사를 시작합니다(148페이지의 “단계 VI: 복사” 참조). 그러면 **Analysis** 대화 상자가 **Copy** 대화 상자로 바뀝니다.

미리 보기 세션을 시작했으면 분석 후 복사가 중단됩니다. **Cancel**을 클릭하면 **Software Selection** 창으로 돌아갑니다.

그림 4-4

### Copy Analysis 대화 상자



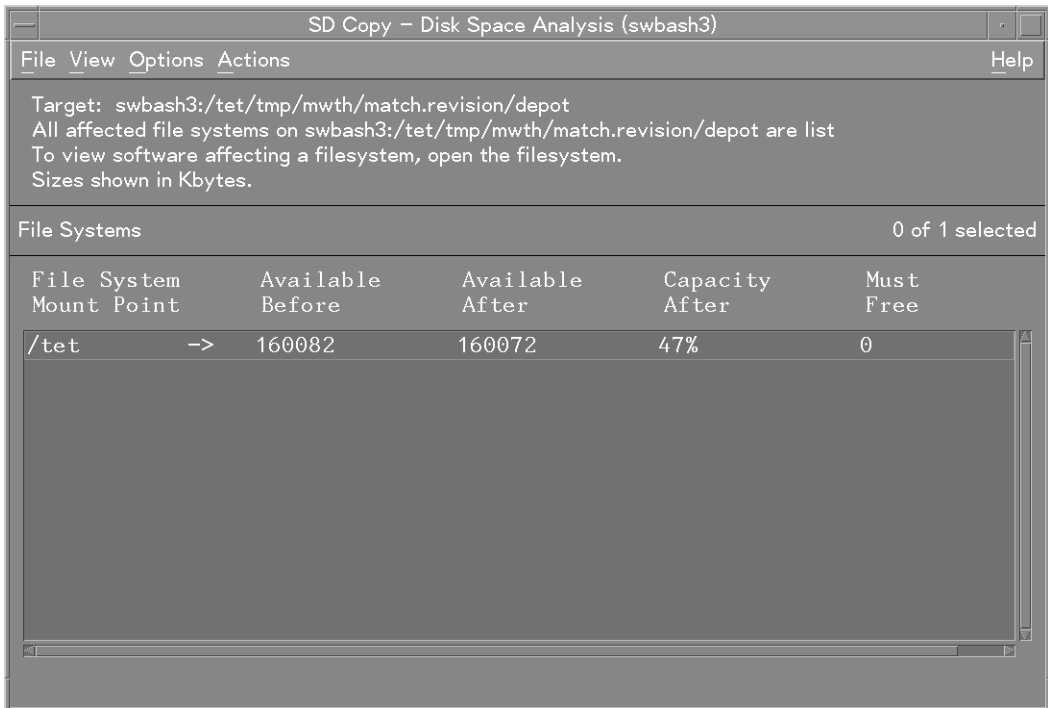
다음 작업을 수행할 수 있습니다.

- **Product Summary**에서는 제품이나 번들에 대한 추가 정보를 제공하며 중속성, 저작권, 공급업체 등에 대한 추가 정보를 표시하는 **Product Description** 단추가 있습니다.
- **Logfile**에서는 로그 파일에 기록된 자세한 복사 정보를 여러 화면에 걸쳐 보여줍니다.
- **Disk Space**는 **Disk Space Analysis** 창(그림 4-5, “Disk Space Analysis 창”)을 표시하며 이 창에는 다음과 같은 내용이 포함됩니다.
  - 파일 시스템 마운트 지점
  - 복사 전에 사용할 수 있었던 디스크 공간의 양
  - 복사 후 사용 가능한 디스크 공간의 양
  - 사용될 디스크 용량의 비율
  - 작업을 완료하기 위해 확보되어야 할 디스크 공간의 양

이 창의 메뉴를 통해 다음과 같은 작업을 수행할 수 있습니다.

- 객체 목록 검색
- 특정 파일 세트에 대해 예상되는 용량 요구 사항을 보기 위한 항목 열기
- **Re-analyze**는 분석 프로세스를 반복합니다.

그림 4-5 **Disk Space Analysis 창**



분석이 완료되면 모든 호스트의 상태는 Ready 또는 Excluded from task 중 하나로 표시됩니다. 선택한 소프트웨어 중 **하나라도** 호스트에 복사될 수 있으면 상태는 Ready로 표시됩니다. 선택한 소프트웨어 중 **아무 것도** 호스트에 복사될 수 없으면 상태는 Excluded from task로 표시됩니다.

다음은 상태 결과에 대한 요약입니다. **Logfile** 단추를 클릭하면 대부분의 문제에 대한 자세한 내용을 찾을 수 있습니다.

**Ready**                      분석이 수행되는 동안 어떠한 오류나 경고도 발생하지 않았습  
다. 아무런 문제 없이 복사가 진행됩니다.

**Ready with Warnings**                      분석이 수행되는 동안 경고가 생성되었습니다. 오류와 경고가 로  
그 파일에 기록됩니다.

**Ready with Errors**                      선택된 하나 이상의 제품이 복사됩니다. 그러나 선택된 제품 중  
하나 이상이 분석 오류로 인해 작업에서 제외됩니다. 오류와 경고  
가 로그 파일에 기록됩니다.

**Communication failure**                      대상 또는 소스와의 접속 또는 통신이 유실되었습니다.

**Excluded due to errors**                      일종의 글로벌 오류가 발생했습니다. 예를 들면 시스템이 파일 시  
스템을 마운트할 수 없는 경우일 수 있습니다.

**Disk Space Failure**                      복사 시 디스크 저장소에서 사용할 수 있는 공간을 초과했습니다.  
자세한 내용은 **Disk Space** 단추를 클릭하십시오.

Products Scheduled 열에서는 선택한 모든 제품 중 복사할 준비가 된 제품의 수를 보여줍니다. 여기에는 다음과 같은 작업이 포함됩니다.

- 단지 종속 관계로 인해 선택된 제품
- 부분적으로 선택된 제품
- 기타 선택된 제품 및 번들

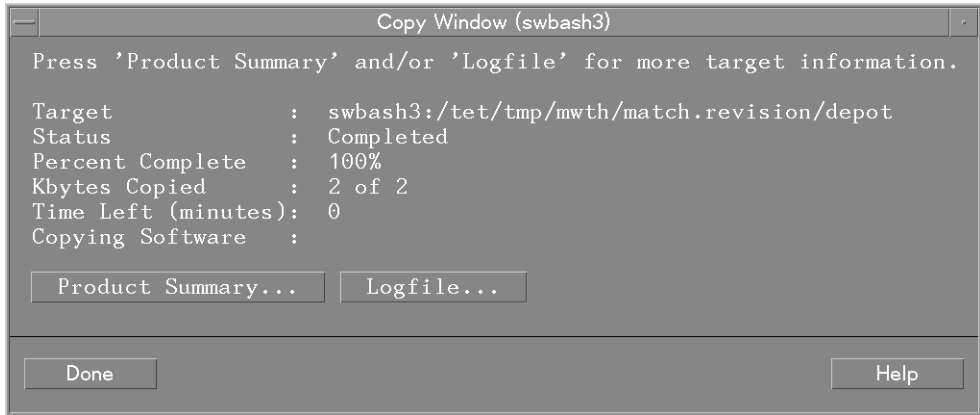
**단계 VI: 복사**

이 단계에서는 SD-UX가 실제 복사를 수행합니다.

Analysis 창에서 **OK**를 클릭하면 SD-UX는 복사를 시작하고 상태 정보를 보여주는 Copy 창(그림 4-6, “Copy 창”)을 표시합니다.

그림 4-6

**Copy 창**



다음과 같은 작업 단추를 사용할 수 있습니다.

- **Done**을 클릭하면 **Software Selection** 창으로 돌아갑니다. 그러면 다른 복사를 시작하거나 GUI를 종료(**File→Exit**)할 수 있습니다.
- **Product Summary**는 복사 및 제품 정보(이름, 버전, 복사 결과, 제품 설명)를 표시합니다.
- **Logfile**을 클릭하면 로그 파일을 표시합니다.

## swcopy 명령줄 사용

### swcopy 구문

```
swcopy [XToolkit Options] [-i] [-p] [-v] [-C session_file]
[-f software_file] [-Q date] [-s source] [-S session_file]
[-t target_file][-x option=value] [-X option_file]
[software_selections] [@ target_selections]
```

### 옵션 및 피연산자

<b>XToolkit 옵션</b>	GUI용 X window 옵션입니다. 자세한 내용은 55페이지의 “XToolkit 옵션 및 디스플레이 글꼴 변경”을 참조하십시오.
<b>-i</b>	GUI 프로그램을 실행합니다. 자세한 내용은 140페이지의 “swcopy GUI 사용”을 참조하십시오.
<b>-p</b>	분석 단계를 통해 복사 작업을 실행한 다음 종료하여 복사 작업을 미리 봅니다.
<b>-v</b>	stdout으로의 상세 출력을 활성화하고 모든 활동을 화면에 표시합니다.
<b>-C session_file</b>	명령을 실행하고 현재 옵션과 피연산자 값을 다른 세션에 다시 사용할 수 있도록 <b>session_file</b> 에 저장합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.
<b>-f software_file</b>	명령줄 대신(또는 그에 추가하여) 별도의 파일에서 대상 선택 목록을 읽습니다. 자세한 내용은 60페이지의 “소프트웨어 파일”을 참조하십시오.
<b>-Q date</b>	원격 작업을 수행하는 경우 해당 날짜에 작업이 실행되도록 일정을 예약합니다. 211페이지의 “명령줄에서 작업 예약” 및 6장, 167페이지의 “원격 작업 개요”를 참조하십시오.
<b>-s source</b>	기본값인 <code>/var/spool/sw</code> 대신 <b>source</b> 를 통해 지정한 소프트웨어 소스를 사용합니다. 구문은 다음과 같습니다.  [ <i>host</i> ][:/ <i>directory</i> ]  <i>host</i> 는 호스트 이름, 도메인 이름 또는 인터넷 주소(예: 15.1.48.23)가 될 수 있습니다. <i>directory</i> 는 절대 경로입니다.

`-S session_file`

`session_file`에 정의된 대로 이전 세션에서 저장된 값을 바탕으로 명령을 실행합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.

`-t target_file`

명령줄 대신(또는 그에 추가하여) 별도의 파일에서 대상 선택 목록을 읽습니다. 자세한 내용은 61페이지의 “대상 파일”을 참조하십시오.

`-x option=value`

명령 `option`을 `value`로 설정하고 기본값 또는 옵션 파일의 값을 대체합니다. 자세한 내용은 150페이지의 “명령 옵션 변경”을 참조하십시오.

`-X option_file`

`option_file`에서 세션 옵션 및 기능을 읽습니다. 자세한 내용은 150페이지의 “명령 옵션 변경”을 참조하십시오.

`software_selections`

복사될 소프트웨어 객체입니다. 자세한 내용은 58페이지의 “소프트웨어 선택 사항”을 참조하십시오.

`target_selections`

명령의 대상입니다. 자세한 내용은 60페이지의 “대상 선택 사항”을 참조하십시오.

## 명령 옵션 변경

명령을 호출할 때 명령줄 옵션을 추가로 지정(-x 옵션 사용)하거나 파일에서 미리 정의된 값을 읽어서 이 명령의 작동 방식을 변경할 수 있습니다. 다음 표에서는 `swcopy`에 적용되는 기본값과 옵션을 보여줍니다.

표 4-4

**swcopy 명령 옵션 및 기본값**

- admin\_directory=/var/adm/sw
- agent\_auto\_exit=true
- agent\_timeout\_minutes=10000
- allow\_split\_patches=false
- autoremove\_job=false
- autoselect\_dependencies=true
- autoselect\_patches=true
- autoselect\_reference\_bundles=true
- codeword=
- compress\_files=false
- compress\_index=false
- controller\_source=
- create\_target\_path=true
- customer\_id=
- distribution\_source\_directory=/var/spool/sw
- distribution\_target\_directory=/var/spool/sw
- enforce\_dependencies=true
- enforce\_dsa=true
- job\_title=
- layout\_version=1.0
- log\_msgid=0
- logdetail=false
- logfile=/var/adm/sw/swcopy.log
- loglevel=1
- max\_targets=
- mount\_all\_filesystems=true
- patch\_filter=software\_specification
- patch\_match\_target=false
- polling\_interval=2
- preview=false
- register\_new\_depot=true
- reinstall=false
- reinstall\_files=true
- reinstall\_files\_use\_cksum=true
- remove\_obsolete\_filesets=false
- retry\_rpc=1
- retry\_rpc\_interval={0}
- reuse\_short\_job\_numbers=true
- rpc\_binding\_info=ncacn\_ip\_tcp:[2121]  
ncadg\_ip\_udp:[2121]
- rpc\_timeout=5
- run\_as\_superuser=false
- select\_local=true
- software=
- software\_view=all\_bundles
- source=
- source\_cdrom=/SD\_CDROM
- source\_tape=/dev/rmt/0m
- source\_type=directory
- targets=
- uncompress\_files=false
- use\_alternate\_source=false
- verbose=1
- write\_remote\_files=true

**추가 정보**

각 기본값에 대한 자세한 설명은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.

## 복사 작업 및 예

이 절에서는 소프트웨어 제품을 복사하기 위한 명령을 예를 들어 설명합니다(157페이지의 “추가적인 저장소 관리 작업 및 예” 참조).

### 간단한 swcopy 예

/dev/rmt/0m의 DAT 테이프에서 로컬 호스트의 기본 저장소(/var/spool/sw)로 모든 제품을 복사하려면 다음과 같이 입력합니다.

```
swcopy -s /dev/rmt/0m \*
```

mysoft 파일에서 명명된 로컬 CD-ROM의 소프트웨어 선택 목록을 *hostA* 호스트의 /depots/mydep/ 경로에 있는 저장소로 복사하고, 실제 소프트웨어를 복사하기 전에 미리 보려면 다음과 같이 입력합니다.

```
swcopy -p -f mysoft -s /mnt/cd @ hostA:/depots/mydep/
```

### 소프트웨어 코드워드 및 고객 ID 사용

코드워드로 보호된 소프트웨어에 액세스하려고 하면 **swcopy** 명령이 코드워드를 입력하라는 메시지를 표시합니다. 또한 명령줄이나 GUI에서 새로운 코드워드를 입력할 수도 있습니다. 이 프로세스는 **swinstall**에서 사용된 것과 동일합니다. 자세한 내용은 80페이지의 “소프트웨어 코드워드 및 고객 ID 사용”을 참조하십시오.

### 저장소의 여러 소프트웨어 제품

소프트웨어는 제품으로 패키징됩니다. 저장소에는 제품의 여러 버전을 저장할 수 있습니다.

제품 버전이 이미 저장소에 있는 경우 **reinstall** 옵션을 **true**로 설정한 경우가 아니면 **swcopy**는 이를 대체하지 **않습니다**. 이 옵션이 **true**인 경우 제품은 다시 복사됩니다.

저장소에 제품의 다른 버전이 이미 존재하는 경우 **swcopy**는 새 버전을 복사하고 다른 버전은 바뀌지 않습니다.

제품의 여러 버전이 존재할 때 **swcopy**는 자동으로 알려주지 않습니다. **swcopy**는 정확하게 일치하는 버전이 존재하고 이를 생략(또는 다시 복사)할 때에만 알려줍니다.



## 저장소 등록 및 등록 해제(swreg)

네트워크를 통해 다른 SD-UX 명령으로 저장소의 소프트웨어를 사용하려면 저장소를 등록해야 합니다. 사용하지 않으려면 저장소 등록을 해제할 수도 있습니다.

저장소는 다음과 같은 방법으로 등록하거나 등록 해제할 수 있습니다.

- `swcopy` 명령은 새로 만들어진 저장소를 자동으로 등록합니다.  
(`register_new_depot` 옵션으로 이 기능을 켜거나 끌 수 있습니다.)
- `swremove` 명령은 저장소에 들어 있는 소프트웨어를 모두 제거한 다음 저장소를 자동으로 등록 해제합니다.
- `swreg` 명령은 명시적으로 저장소를 등록 또는 등록 해제합니다.

`swreg` 명령을 통해 `swcopy` 또는 `swremove`의 자동 등록 기능이 충분하지 않을 때 명시적으로 저장소를 등록하거나 등록 해제할 수 있습니다. 예를 들면 `swreg`를 다음과 같이 사용할 수 있습니다.

- CD-ROM이나 기타 미디어를 등록된 저장소로 사용하려고 할 때
- `swpackage`를 통해 만든 저장소를 등록할 때
- 호스트에서 실제로 저장소를 제거하지 않고 네트워크 액세스를 제한하기 위해 저장소를 등록 해제하려 할 때

## 미디어 등록 또는 네트워크 저장소 작성?

소프트웨어 미디어를 등록된 저장소로 사용하는 것과 미디어를 사용하여 네트워크 저장소를 만드는 것 중 어떤 것이 더 적합할까요? 일반적으로 미디어를 저장소로 사용하는 것은 한 대나 두 대의 시스템만이 미디어에 액세스하는 작은 규모의 사용에 적합합니다. 보다 많은 수의 시스템이 미디어에 액세스해야 한다면 개별 미디어에서 네트워크 저장소를 만드는 것이 성능 향상에 도움이 됩니다. 157페이지의 “추가적인 저장소 관리 작업 및 예”를 참조하십시오.

## 등록 및 보안

SD-UX는 객체를 파일 시스템에 저장하므로 누군가 소프트웨어 저장소에 대한 “트로이 목마” 파일 시스템 이미지를 만들 수 있습니다. 이로 인해 잘못된 저장소에서 제품을 설치한 시스템의 경우 보안이 위협받을 수 있습니다. 이러한 상황으로부터 시스템을 보호하려면 저장소로부터 소프트웨어를 설치하거나 복사하기 전에 저장소를 등록해야 합니다. 이러한 검사는 로컬 슈퍼유저가 `swinstall`을 실행하는 경우를 제외하고는 액세스 권한을 부여하기 전에 항상 수행됩니다.

---

### 참고

액세스 제한 없이 저장소 등록을 수행할 수 있습니다. 액세스 제어는 SD 보안 관련 영역입니다(9장, 233페이지의 “SD-UX 보안” 참조). `swreg`를 사용하여 등록하는 경우 호스트 ACL에 대한 삽입 권한이 필요합니다.

---

## 권한 부여

새로운 저장소를 등록하거나 기존 저장소를 등록 해제하기 위해 `swreg`는 해당 저장소에서의 읽기 권한과 호스트에서의 삽입 권한이 필요합니다. 등록된 저장소를 등록 해제하기 위해 `swreg` 명령은 호스트에서의 쓰기 권한이 필요합니다. 자세한 내용은 9장, 233페이지의 “SD-UX 보안”을 참조하십시오.

## swreg 사용

### swreg 구문

```
swreg -l level [-u] [-v] [-C session_file] [-f object_file]
[-S session_file] [-t target_file] [-x option=value]
[-X option_file] [objects_to_register] [@ target_selections]
```

### 옵션 및 피연산자

<code>-l level</code>	등록하거나 등록을 해제할 객체의 수준을 지정합니다. 여기서 수준은 <code>depot</code> 또는 <code>root</code> 가 될 수 있습니다.
<code>-u</code>	<code>swreg</code> 가 지정된 객체를 등록하는 대신 등록 해제하도록 합니다.
<code>-v</code>	<code>stdout</code> 으로의 상세 출력을 활성화하고 모든 활동을 화면에 표시합니다.

`-C session_file`

명령을 실행하고 현재 옵션과 피연산자 값을 다른 세션에 다시 사용할 수 있도록 `session_file`에 저장합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.

`-f object_file`

명령줄 대신(또는 그에 추가하여) `object_file`에서 등록 또는 등록 해제할 저장소나 루트 객체 목록을 읽습니다.

`-S session_file`

`session_file`에 정의된 대로 이전 세션에서 저장된 값을 바탕으로 명령을 실행합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.

`-t target_file`

명령줄 대신(또는 그에 추가하여) 별도의 파일에서 대상 선택 목록을 읽습니다. 자세한 내용은 61페이지의 “대상 파일”을 참조하십시오.

`-x option=value`

명령 `option`을 `value`로 설정하고 기본값 또는 옵션 파일의 값을 대체합니다. 자세한 내용은 61페이지의 “명령 옵션 사용”을 참조하십시오.

`-X option_file`

`option_file`에서 세션 옵션 및 기능을 읽습니다. 자세한 내용은 61페이지의 “명령 옵션 사용”을 참조하십시오.

`objects_to_register`

등록 또는 등록 해제될 소프트웨어 객체입니다.

`@target_selections`

객체가 등록 또는 등록 해제될 대상입니다. 자세한 내용은 60페이지의 “대상 선택 사항”을 참조하십시오.

### 명령 옵션 변경

명령을 호출할 때 명령줄 옵션을 추가로 지정(-x 옵션 사용)하거나 파일에서 미리 정의된 값을 읽어서 이 명령의 작동 방식을 변경할 수 있습니다. 다음 표에서는 **swreg**에 적용되는 기본값과 옵션을 보여줍니다.

표 4-5

#### **swreg** 명령 옵션 및 기본값

- `admin_directory=/var/adm/sw`
- `distribution_target_directory=/var/spool/sw`
- `level=`
- `log_msgid=0`
- `logfile=/var/adm/sw/swreg.log`
- `logdetail=false`
- `loglevel=1`
- `objects_to_register=`
- `rpc_binding_info=ncacn_ip_tcp:[2121]ncadg_ip_udp:[2121]`
- `rpc_timeout=5`
- `run_as_superuser=true`
- `select_local=true`
- `select_local=true`
- `targets=`
- `verbose=1`

### 추가 정보

각 기본값에 대한 자세한 설명은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.

### **swreg** 예

`/mnt/cd`에 마운트된 CD-ROM 저장소를 등록 해제하려면 다음과 같이 입력합니다.

```
swreg -l depot -u /mnt/cd
```

동일한 저장소(로컬 호스트에서 `/mnt/cd`에 마운트됨)를 네트워크에서 사용 가능한 저장소로 등록하려면 다음과 같이 입력합니다.

```
swreg -l depot /mnt/cd
```

다음 예에서는 Support Plus CD가 `/cdrom`에 마운트되어 있다고 가정하고, 한 대 또는 두 대의 다른 시스템에서 Support Plus CD의 `HWEnable11i` 저장소로 직접 액세스할 수 있게 합니다.

```
swreg -l depot /cdrom/HWEnable11i
```

---

## 추가적인 저장소 관리 작업 및 예

이 절에서는 일반적인 저장소 관리 작업을 보여주며 SD-UX를 사용하여 환경을 관리하는 방법에 대한 포괄적인 예를 제공합니다.

### 패치 저장소 병합

이 예에서는 HP에서 다운로드한 다섯 개의 패치(테이프 저장소임)를 하나의 저장소로 병합하는 방법을 보여줍니다. 이 예에서는 또한 `patch_match_target` 옵션을 사용하여 저장소를 등록하고 저장소 내용을 나열하며 새로운 저장소로부터 패치를 설치하는 방법에 대해 보여줍니다. 이 예에서는 HP ITRC(<http://itrc.hp.com/>)로부터 이미 PHKL\_20349, PHKL\_22161, PHSS\_21906, PHSS\_21950 및 PHCO\_22923 패치를 다운로드했다고 가정합니다.

```
swcopy -s /tmp/PHKL_20349.depot \* @ /depots/mypatches
swcopy -s /tmp/PHKL_22161.depot \* @ /depots/mypatches
swcopy -s /tmp/PHSS_21906.depot \* @ /depots/mypatches
swcopy -s /tmp/PHSS_21950.depot \* @ /depots/mypatches
swcopy -s /tmp/PHCO_22923.depot \* @ /depots/mypatches
swreg -l depot @ /depots/mypatches
swlist -d -s /depots/mypatches
swinstall -x patch_match_target=true -s /depots/mypatches
```

## 배포용 테이프 저장소 만들기

이 예에서는 ftp나 웹을 통해 배포할 수 있는 하나의 파일로 테이프 저장소를 만드는 방법에 대해 보여줍니다. 이 예에서는 앞의 예에서 나온 다섯 개의 패치(테이프 저장소로 포맷됨)를 사용하며 /depots/mypatches에 있는 기존 저장소를 사용합니다. swlist 명령은 저장소 내용을 보여줍니다(161페이지의 “저장소 내용 나열(swlist -d)” 참조).

```
swpackage -x media_type=tape -s /depots/mypatches \
          @ /tmp/mypatches.depot
```

```
swlist -d -s /tmp/mypatches.depot
```

유효한 PSF(Product Specification File)인 myproduct.psf에서 테이프 저장소를 만들려면 다음과 같이 입력합니다.

```
swpackage -x media_type=tape -s myproduct.psf \
          @ /tmp/myproduct.depot
```

```
swlist -d -s /tmp/myproduct.depot
```

swpackage에 대한 자세한 내용은 10장, 279페이지의 “소프트웨어 패키지 만들기”를 참조하십시오.

## 저장소 속성 설정

저장소를 만들 때 저장소 및 그 안에 들어 있는 내용을 식별하는 데 도움이 될 제목과 설명 속성을 설정하고자 할 수 있습니다.

저장소에 대한 PSF(Product Specification File)의 맨 처음에 다음과 같은 줄을 입력합니다.

```
distribution
title Optional Development Tools
description "Text processing and programming tools
            \generally useful for source code development"
```

그런 다음 저장소를 패키징하고 등록합니다.

```
swpackage -s mydepot.psf @ /depots/mydepot
```

```
swreg -l depot @ /depots/mydepot
```

시스템의 모든 저장소에 대한 제목과 설명을 보려면 다음과 같이 입력합니다.

```
swlist -v -a title -a description -l depot
```

## 네트워크 저장소 만들기

설치할 소프트웨어를 가져올 네트워크 저장소를 만들면 많은 수의 시스템에 소프트웨어를 설치해야 할 때 성능 및 사용의 편의성을 향상시킬 수 있습니다. 예를 들어 HP-UX 11i는 두 장의 CD에 제공되므로 업데이트 중에 CD를 교체해야 합니다. CD를 교체하지 않고 업데이트하려면 기존 11i 시스템에 필요한 소프트웨어를 모두 포함한 원격 저장소를 만들고 해당 소스에서 업데이트해야 합니다. (업데이트 프로세스에 대한 자세한 내용은 **HP-UX 11i 설치 및 업데이트 설명서**를 참조하십시오.)

루트로서 이 절차를 수행하여 HP-UX 11i CD에서 저장소 서버로 또는 네트워크상에서 HP-UX 11i를 실행하는 다른 시스템으로 네트워크 저장소를 만듭니다.

1. 네트워크상의 다른 시스템에 네트워크 저장소를 만들 수 있도록 최소 1,230MB의 사용 가능한 공간이 있는지 확인합니다. 이 정도의 공간이 없으면 SAM을 사용하여 새로운 볼륨 그룹을 만들거나 기존의 볼륨 그룹을 확장합니다. 자세한 내용은 SAM 도움말 또는 **시스템 및 작업 그룹 관리 설명서**를 참조하십시오.
2. 루트로 로그인하고 `/update`라는 새로운 디렉토리에 논리 볼륨을 마운트합니다. 이 디렉토리에 네트워크 저장소가 만들어집니다.
3. HP-UX 11i CD1을 넣고 CD 드라이브의 사용 표시등이 깜박임을 멈출 때까지 기다립니다.
4. CD-ROM 장치 파일 이름을 찾습니다.

```
ioscan -fn | more
```

대개 CD-ROM 장치 이름은 `/dev/dsk/c1t2d0`입니다.

5. 루트(/) 아래에 `/cdrom`이라는 디렉토리를 만듭니다.

```
mkdir /cdrom
```

6. CD를 `/cdrom` 디렉토리에 파일 시스템으로 마운트합니다. 예를 들면 다음과 같습니다.

```
mount /dev/dsk/c1t2d0 /cdrom
```

7. 마운트된 CD에 있는 모든 제품을 대상 저장소(예: `/update/update-depot`)로 병합합니다.

```
swcopy -s /cdrom \* @ /update/update-depot
```

8. CD를 `/cdrom` 디렉토리에서 마운트 해제합니다.

```
umount /cdrom
```

9. HP-UX 11i CD2를 넣고 드라이브의 사용 표시등이 깜박임을 멈출 때까지 기다립니다.

10. CD2 및 Support Plus CD를 사용하여 6단계부터 8단계까지 반복합니다.

이제 네트워크 저장소를 사용하여 HP-UX 10.20 또는 11.0 시스템을 HP-UX 11i로 업데이트할 수 있습니다.

## HP-UX 여러 버전 관리

HP-UX 11i 시스템을 사용하여 HP-UX 11.00 및 10.20에 대한 저장소를 관리할 수 있으며 다음과 같은 지침에 유의합니다.

- HP에서는 같은 저장소 안에 여러 OS 버전을 함께 보관하지 않도록 권장합니다. 즉 10.20 소프트웨어는 10.20 저장소에, 11.00 소프트웨어는 11.00 저장소에 그리고 11i 소프트웨어는 11i 저장소에 보관합니다.
- 11.00 소프트웨어와 11i 소프트웨어는 별도의 저장소에서 따로 관리해야 하지만, 11i와 분리하여 11.00 저장소를 관리할 때 특별한 고려 사항은 없습니다. 형식은 유사하지만 11i 소프트웨어에는 11.00 시스템이 인식할 수 없는 업체에서 정의한 속성이 들어 있을 수 있습니다. 이런 경우 11.00 시스템이 11i 소프트웨어에 액세스할 때 경고가 발생합니다.
- 10.20 저장소의 경우:

- 11i 시스템에서 10.20 저장소를 만들려면 올바른 `layout_version`을 사용해야 합니다. 예를 들면 다음과 같습니다.

```
swcopy -x layout_version=0.8 ...
```

```
swpackage -x layout_version=0.8 ...
```

그러면 이제부터 11i 시스템은 10.20 저장소를 관리할 수 있습니다.

- `layout_version=1.0` 소프트웨어(11.00 또는 11i 형식)를 `layout_version=0.8`(10.20) 저장소에 넣으려고 하면 SD-UX에서 경고가 발생합니다.

## 등록된 저장소 나열

`swlist`는 호스트에 있는 등록된 저장소 목록을 표시합니다. 이렇게 하려면 `-l depot` 옵션을 조합하여 사용합니다.

로컬 호스트에 있는 저장소를 모두 나열하려면 다음과 같이 입력합니다.



```
swlist -l depot
```

원격 시스템(hostA)에 있는 저장소를 모두 나열하려면 다음과 같이 입력합니다.

```
swlist -l depot @ hostA
```

시스템에 있는 저장소를 최신에서 오래된 순서로(가장 최근에 변경된 순서에 따라) 모두 나열하려면 다음과 같이 입력합니다.

```
swlist -l depot -a mod_date -a mod_time | sort -rn -k 7,7
```

---

## 팁

mod\_time을 편리한 정렬 필드(단일 정수)로 사용하고 mod\_date에 관독 가능한 출력을 포함시킵니다. (mod\_time은 화면 맨 끝의 눈에 덜 띄는 위치에 배치합니다.)

---

## 저장소 내용 나열(swlist -d)

swlist를 사용하면 패키징되고 저장되어 설치될 수 있는 소프트웨어를 모두 나열할 수 있습니다.

swlist -d 옵션을 통해 로컬 호스트의 기본 저장소에 있는 소프트웨어를 나열할 수 있습니다. GUI로 저장소를 찾아 보려면 swlist -i -d를 사용합니다. 연관된 세션 및 감사 로그 파일도 볼 수 있습니다.

---

## 참고

기본적으로 swlist의 출력에 표시되는 속성은 POSIX 형식으로 되어 있습니다. 이 출력을 분석하는 사용자들은 이를 고려해야 할 것입니다.

---

다음 예에서 swlist 출력 요청은 표준 출력으로 보내집니다. 모든 예에서 one\_liner 옵션은 revision size title이고 level 옵션은 product이거나 정의되지 않은 것으로 가정합니다.

로컬 테이프 저장소인 /dev/rmt/0의 내용을 나열합니다.

```
swlist -d @ /dev/rmt/0
```

AUDIT	3.5	9834	Trusted Systems Auditing Utils
COMMAND	1.7	4509	Core Command Set
C-LANG	2.5	5678	C Programming Language
DISKLES	1.8	6745	HP Cluster Commands
KERNEL	1.4	56908	Kernel Libraries and Headers

```
VUE      1.3   5489   Vue (Instant Ignition Release)
WINDOWS  2.06  10423  Windowing Products
```

로컬 테이프 저장소인 /dev/rmt/0의 미디어 속성을 나열합니다.

```
swlist -d -v -l depot @ /dev/rmt/0
```

```
type      distribution
tag       CORE OS
description HP-UX Core Operating System Software Disc
number    B2358-13601
date      June 1991
```

/swmedia에 위치한 host1의 소프트웨어 저장소에 저장된 제품을 나열합니다. 이 예에서 one\_liner는 title size architecture로 가정합니다.

```
swlist -d @ host1:/swmedia
```

```
FRAME Frame Document Pkg 2319 HP-UX_9000_Series700/800_AorB
FRAME Frame Document Pkg 2458 OSF1_9000_Series700_1.0
ME30   3-D Mechanical Eng 5698 HP-UX_9000_Series300/800_AorB
SOFTBENCH Softbench Development Env 4578 HP-UX_9000_Series300
TEAMWORK Tmwk. Design/Analysis 3478 HP-UX_9000_Series300/400
```

미디어에는 FRAME 제품의 두 가지 버전이 들어 있습니다.

## 소스 저장소 감사

소스 및 대상 시스템이 모두 10.30 이상 버전의 HP-UX인 경우 swlist를 사용하여 저장소를 감사할 수 있습니다. 소스 저장소 시스템의 시스템 관리자는 감사 기능을 켜거나 끌 수 있습니다. 이 기능은 사용자 및 이들의 소프트웨어 선택 내용을 추적합니다. 또한 저장소가 사용되고 있는 시점을 판별할 수 있습니다.

관리자는 swagent의 /var/adm/sw/defaults 파일에 있는 swagent.source\_depot\_audit 값을 true로 설정해야 합니다. 그러면 swaudit.log 파일이 소스 저장소(쓰기 가능한 디렉토리 저장소의 경우)에 만들어지거나 /var/tmp(tar 이미지, CD-ROM 또는 기타 쓰기 불가능한 저장소의 경우)에 만들어집니다. 이는 소스 저장소에 대해 swagent.log처럼 작동합니다.

**swlist -i -d**를 입력하여 감사 파일을 볼 수 있습니다. 시스템에 해당 SD 메시지 카탈로그 파일이 있는 한 원하는 언어를 설정하여 원격/로컬 저장소에서 감사 정보를 볼 수 있습니다.

## 저장소 확인(swverify -d)

swverify 명령을 사용하여 저장소의 소프트웨어를 확인할 수 있습니다. swverify는 다음과 같은 작업을 수행합니다.

- 모든 종속성(전제 조건과 필수 조건)이 충족될 수 있는지 확인합니다.
- 유실된 파일을 보고합니다.
- 권한, 파일 유형, 크기, 체크섬, mtime 및 1차/2차 속성과 같은 파일 속성을 검사합니다.

예를 들어 로컬 저장소의 내용 전체를 확인하려면 다음과 같이 입력합니다.

```
swverify -d \* @ /var/spool/sw
```

---

### 참고

swverify 명령은 업체에서 제공하는 확인 스크립트를 저장소 내에서 실행하지 않습니다.

---

## 저장소에서 소프트웨어 제거

`swremove`를 호출할 때 `-d` 옵션을 지정하면 루트 파일 시스템이 아닌 저장소에서 소프트웨어를 제거합니다. 즉 소프트웨어를 제거할 저장소를 나타내는 경로를 지정해야 한다는 의미입니다. 예를 들면 다음과 같습니다.

```
swremove -d Old-Software @ /var/spool/sw
```

`swremove -d` GUI의 경우 GUI를 호출한 다음 나타나는 대화 상자에서 저장소를 지정하라는 메시지가 표시됩니다. 이 대화 상자는 `swcopy` 작업 시 저장소 대상을 지정할 때 사용한 것과 동일합니다. 이 대화 상자 사용 방법에 대한 자세한 내용은 141페이지의 “단계 II: 대상 지정”을 참조하십시오.

## 저장소 제거

저장소를 제거하고 자동으로 등록을 해제하려면 다음과 같이 입력합니다.

```
swremove -d \* @ /tmp/MyDepot
```

## 5 HP-UX 패치 및 패치 관리

HP는 시스템에 대한 증분 업데이트를 제공하는 패치를 릴리즈합니다. 패치는 결함 수정본을 제공하는 것으로 잘 알려져 있지만 새로운 기능을 제공하고 새 하드웨어를 사용할 수 있게 하며 펌웨어를 업데이트하는 역할도 합니다. HP-UX 패치를 사용하면 시스템 응용 프로그램을 완전히 다시 설치하지 않아도 HP-UX 소프트웨어를 업데이트할 수 있습니다.

*Patch Management User Guide for HP-UX 11.x Systems*는 Hardware Enablement(HWE) 및 Quality Pack(QPK) 번들을 포함하여 표준 HP-UX 패치 번들을 얻어서 설치하는 방법에 대한 정보와 절차를 제공합니다. 다음 HP 기술 설명서 웹사이트를 참조하십시오.

**<http://docs.hp.com/en/oshpux11iv2.html#Patch%20Management>**

이 설명서의 내용은 HP-UX 11.0(B.11.00), HP-UX 11i v1(B.11.11), HP-UX 11i v1.6(B.11.22), 및 HP-UX 11i 2.0(B.11.23) 릴리즈에 적용됩니다.

또한 다음 IT 리소스 센터도 패치, 도구 및 정보를 제공하는 중요한 리소스입니다.

**<http://itrc.hp.com>**

---

### 참고

HP 패치 리소스는 HP 관련 패치에만 사용됩니다. 다른 공급업체의 패치와 응용 프로그램을 설치하거나 업그레이드하려면 해당 공급업체의 설명서를 참조하십시오.

---



## 6 원격 작업 개요

이 장에서는 설치, 기능 및 여러 시스템에서 소프트웨어를 효율적으로 관리하는 데 도움이 될 수 있는 중요한 개념 등, 원격 작업에 대한 개요 정보를 제공합니다. 원격 작업에 대한 자세한 내용은 7장, 193페이지의 “작업 및 작업 브라우저 사용”에도 설명되어 있습니다.

표 6-1

이 장의 항목

설명하는 항목
168페이지의 “소개” 178페이지의 “소개”
177페이지의 “원격 작업 설정”
189페이지의 “명령줄에서의 원격 작업”
171페이지의 “원격 작업 GUI 사용”
178페이지의 “원격 작업 자습서”
188페이지의 “원격 대화식 swlist”

---

## 소개

중앙 저장소에서 소프트웨어를 “가져오는(pull)” 기능과 함께 **Software Distributor**는 로컬 호스트에서 원격 시스템으로 소프트웨어를 “보내는(push)” 원격 작업을 위한 강력한 기능도 제공합니다. 이러한 기능을 대화식으로 사용하고 모든 **SD-UX** 명령의 결과를 작업 브라우저를 사용하거나 명령줄에 **swjob** 명령을 지정하여 모니터링할 수 있습니다.

---

## 참고

원격 작업에서는 터미널 사용자 인터페이스(TUI)를 사용할 수 없습니다.

## 원격 작업과 로컬 작업의 차이점

일반적으로 로컬 작업에 적용되는 모든 **Software Distributor** 기능은 원격 작업에도 적용됩니다. 원격 작업에 추가되는 기능이 이 절에 요약되어 있습니다.

### 원격 대상

로컬 작업의 경우 대상은 로컬 호스트나 로컬 호스트상의 저장소로 구성됩니다. 원격 작업의 경우 대상은 하나 이상의 원격 시스템이 될 수 있습니다. 대상에는 저장소가 포함될 수 있으며 다른 대상에 대한 소스로 작동할 수도 있습니다.

### 컨트롤러, 데몬 및 에이전트 프로그램

컨트롤러 프로그램은 **SD-UX** 작업 및 프로그램에 대한 사용자 인터페이스를 제공합니다. 컨트롤러는 작업 시작에 필요한 데이터를 수집 및 검증하고 작업 상태에 대한 정보를 표시하는 역할을 수행합니다. 또한 컨트롤러는 원격지 대상 시스템으로 소프트웨어를 배포합니다.

각 대상에서 **SD-UX** 데몬이 백그라운드로 실행되면서 컨트롤러로부터 수신되는 요청을 청취합니다. 요청이 수신되면 데몬은 작업을 수행할 수 있도록 **SD-UX** 에이전트를 예약합니다. 데몬은 또한 호스트의 저장소 중 하나를 소스로 사용하려고 하는 다른 에이전트 프로그램으로부터의 요구에 응답하도록 에이전트를 예약합니다.



---

## 참고

데몬 옵션을 변경하면 **SD-UX** 데몬을 다시 시작해야 합니다. 그렇지 않으면 시스템에서 변경 사항을 인식하지 못합니다. 자세한 내용은 61페이지의 “명령 옵션 사용”을 참조하십시오.

---

### 작업 관리

**SD-UX** 원격 작업을 사용하여 즉시 실행할 작업을 작성하거나 차후에 실행되도록 작업을 예약할 수 있습니다. 또한 명령줄 인터페이스(`swjob` 명령 사용)나 대화식 인터페이스(`sd` 명령 사용) 중 하나를 사용하여 예약된 작업, 활성화된 작업 및 완료된 작업들을 찾아볼 수 있습니다.

### 호환되는 소프트웨어

`swconfig`, `swinstall` 및 `swverify` 명령으로 호환 가능한 소프트웨어 사용을 감지하고 강제 사용할 수 있습니다(즉, 소프트웨어 제품이 시스템 종류 및 운영 체제와 호환 가능한지 확인). 원격 작업을 위한 대상을 여러 개 선택한 경우 **SD-UX**를 사용하여 **모든** 대상과 호환되는 소프트웨어만을 선택할 수 있습니다.

### 소프트웨어 간의 종속성

로컬 작업에서와 마찬가지로 `swask`, `swconfig`, `swcopy`, `swinstall`, `swremove`, 및 `swverify` 명령이 파일 세트와 제품 간의 종속성을 지원합니다. 다른 파일 세트나 제품에 대한 종속성을 지정하는 소프트웨어 선택 사항이 있는 경우, 명령은 해당 소프트웨어를 자동으로 선택합니다. (이 단계는 로컬 호스트에서 실행됩니다.

`autoselect_dependencies` 기본 옵션을 사용하여 이 정책을 재정의할 수 있습니다.)

원격 작업을 사용하여 각 대상 시스템에서 종속성이 분석되고 해당 대상 시스템에서 종속성이 충족되지 않으면 파일 세트는 설치되지 않습니다. (이 정책은 `enforce_dependencies` 기본 옵션을 사용하여 재정의할 수 있습니다.)

### 세션 파일

대부분의 명령에 세션 파일 명령 옵션을 지정하여 세션을 빌드, 저장 및 재사용할 수 있습니다. 원격 작업을 사용하는 경우 대상 선택 사항은 옵션, 소스 정보 및 소프트웨어 선택 사항과 함께 저장됩니다.

### 추가 GUI 구성 요소

SD-UX는 원격 작업이 사용 가능하면 GUI 프로그램에 추가 구성 요소를 추가합니다. 그렇지 않으면 프로그램은 로컬 작업에 사용된 것과 거의 동일합니다(171페이지의 “원격 작업 GUI 사용” 참조).

### 소프트웨어 및 대상 목록

대개의 SD-UX 명령을 통해 별도의 입력 파일로부터 소프트웨어 선택 사항을 읽을 수 있습니다. 원격 작업을 사용하여 별도의 파일에서 대상 목록도 읽을 수 있습니다. 로컬 컨트롤러 GUI를 통해 소프트웨어 및 대상 목록을 사용할 수 있습니다.

### 원격 패치 작업

패치 관리를 위해 SD-UX 원격 작업 기능을 사용할 수 있습니다. 한 번에 여러 대의 원격 시스템을 위한 패치를 명시적으로 선택할 수 있습니다. 그러나 patch\_match\_target 옵션은 한 번에 한 대의 원격 시스템에 대해서만 작동합니다.

### 제한 사항

- 원격 작업을 통해 직접 HP-UX OS 업데이트를 원격 시스템으로 “보낼(push)” 수는 없습니다.
- 원격 작업은 다음과 같은 SD-UX 명령에는 적용되지 않습니다.
  - install-sd
  - swpackage
  - swmodify

---

## 원격 작업 GUI 사용

SD-UX는 원격 작업이 사용 가능하면 GUI 프로그램에 추가 구성 요소를 추가합니다. 원격 작업을 위한 추가 구성 요소에는 대상 선택 창과 대상 목록 관리를 위한 기능, 작업 기본 설정 및 작업 모니터링 창 등이 포함됩니다. 그렇지 않으면 GUI 프로그램은 로컬 작업에 사용된 것과 동일합니다.

원격 작업을 설치하고 중앙 컨트롤러에서 원격 작업 GUI를 사용할 수 있도록 설정하고 나면 보통 때와 같이 `swinstall`, `swcopy` 또는 `swremove` GUI를 시작할 수 있습니다. 예를 들면 다음과 같습니다.

```
/usr/sbin/swinstall
```

또는

```
/usr/sbin/swinstall -i
```

---

### 참고

원격 작업에서는 터미널 사용자 인터페이스(TUI)를 사용할 수 없습니다.

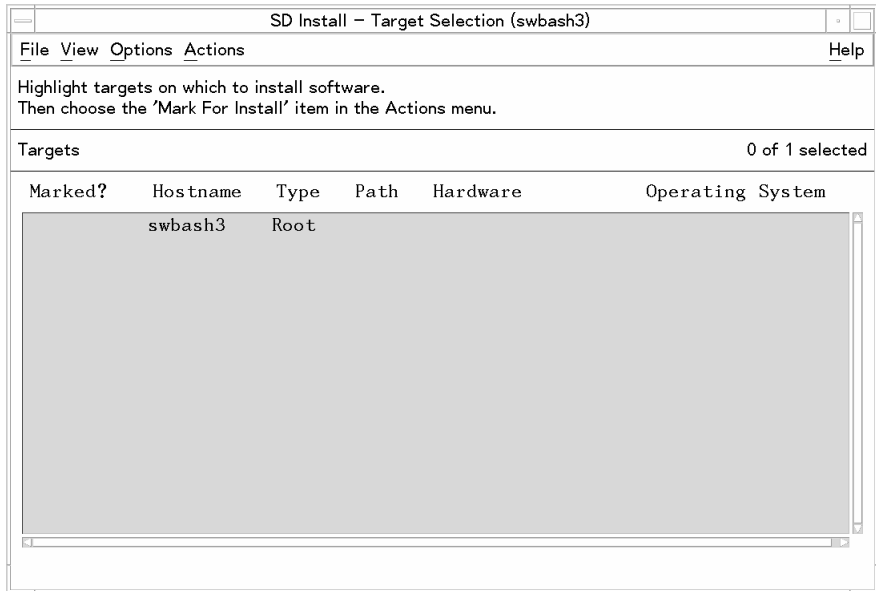
---

## Target Selection 창

Target Selection 창은 원격 GUI 프로그램에서 항상 맨 먼저 표시됩니다. Software Selection 창과 마찬가지로 표준 메뉴 표시줄, 메시지 영역 및 선택 가능한 대상의 객체 목록 등이 표시됩니다. 소프트웨어를 선택하는 대신 원격 작업이 수행될 원격 대상을 선택합니다. 메뉴 항목과 대상 선택 사항은 다음에서 설명됩니다.

그림 6-1

### Target Selection 창



## 작업 수행

원격 작업 GUI를 사용하기 위한 일반적인 절차는 다음과 같습니다.

1. Target Selection 창의 객체 목록에서 대상을 강조 표시하여 한 번에 하나의 대상을 선택합니다.
2. **Actions**→**Mark for Install....**을 선택합니다. (또는 **Actions**→**Mark for Copy....**를 선택하거나 **Actions**→**Mark for Remove....**를 선택합니다.)
3. 다른 대상에 대해 1과 2를 반복합니다.
4. 작업을 위한 대상을 모두 선택했다면 **Actions**→**Show Software for Selection....**을 선택하여 Software Selection 창을 표시합니다.

## 여러 대상 선택

이 절에서는 여러 대상에 설치하고, 대상 그룹을 작성하며, 차후 소프트웨어 설치를 위해 이들 그룹을 저장하는 방법에 대해 설명합니다. (로컬(기본) 대상에 대한 단일 대상 설치하는 178페이지의 “원격 작업 자습서”의 절차를 참조하십시오.)

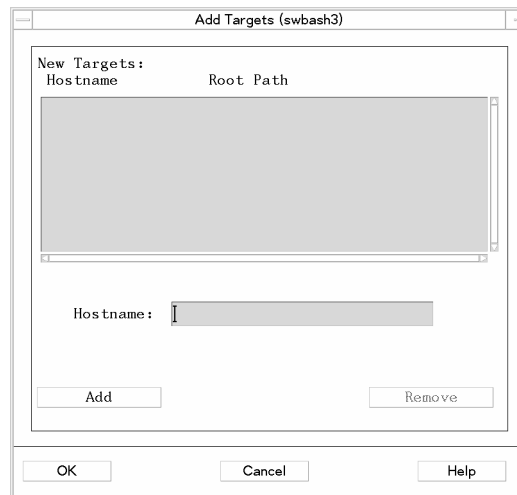
Target Selection 창에 대상 목록이 표시됩니다.

- 세션 파일을 다시 호출(**File**→**Recall Session**)한 경우 해당 세션에 정의된 호스트가 모두 표시됩니다.
- 그렇지 않으면 기본 호스트 파일(/var/adm/sw/defaults.hosts 또는 \$HOME/.sw/defaults.hosts)에 지정된 호스트가 모두 표시됩니다(41페이지의 “호스트 파일 미리 선택” 참조).
- ServiceControl Manager에서 SD-UX를 시작한 경우 대상은 사전에 선택되므로 변경할 수 없습니다.

설치에 필요한 대상이 목록에 없는 경우 다음과 같이 하십시오.

1. **Actions**→**Add Targets....**를 선택합니다. Add Targets 대화 상자(그림 6-2, “Add Targets 대화 상자”)가 표시됩니다.

그림 6-2 Add Targets 대화 상자



2. Hostname: 영역에 기본 루트 이름을 입력하고 **Add**를 선택합니다.
3. **Select Target Path** 대화 상자가 표시됩니다. 기본 경로는 루트(/)입니다. 기본 루트(/)를 그대로 사용하려면 **OK**를 클릭합니다.
4. 루트 경로를 선택하면 Hostname과 Root Path가 **Add Target** 대화 상자(그림 6-2, “Add Targets 대화 상자”)에 자동으로 업데이트됩니다. 대상을 추가하려면 2를 반복합니다.

5. **Add Targets** 대화 상자에서 **OK**를 선택합니다. 이렇게 하면 **Target Selection** 창에 선택 사항이 추가됩니다. **Target Selection** 창에 추가될 때마다 각 대상에 연결됩니다. 대상 시스템에서 **SD-UX** 데몬이 실행 중이 아닌 경우 네트워크 연결로 인해 지연이 발생하고 이는 데몬 시간이 만료될 때까지 계속됩니다.

**Target Selection** 창에서 **Add Targets...**를 사용하여 추가된 대상은 모두 자동으로 **Yes**로 표시됩니다.

6. **Target Selection List**에 표시되지 않은 다른 원하는 대상이 있어서 여기에 설치하고자 하는 경우 그 대상을 클릭하여 강조 표시합니다.

**Actions**→**Mark for Install**을 선택합니다. 해당 대상에 대해 **Marked?** 열이 **Yes**로 설정됩니다.

— 또는 —

마우스 **오른쪽** 단추를 클릭하여 표시되는 메뉴에서 **Mark for Install**을 선택합니다.

7. **Target Selection** 창(즉, 객체 목록)에서 대상 표시를 취소하려면 다음을 수행합니다.

- a. 대상을 강조 표시합니다.
- b. **Actions**→**Unmark for Install**을 선택합니다. 해당 대상에 대해 **Marked?** 열이 지워집니다.

— 또는 —

마우스 **오른쪽** 단추를 클릭하여 표시되는 메뉴에서 **Unmark for install**을 선택합니다.

그러면 모든 원하는 대상이 나열되고 **Marked?** 열에 **Yes**가 표시됩니다. 대상을 하나도 표시하지 않으면 소프트웨어 선택 단계로 넘어갈 수 없습니다.

## 개별 대상 선택

대상을 개별적으로 추가하거나 삭제할 수 있습니다.

새로운 대상을 추가하려면 다음을 수행합니다.

1. **Actions**→**Add Targets...**를 선택합니다. **Add Targets** 대화 상자가 표시됩니다.
2. 원하는 대상의 이름을 입력하고 **Add**를 클릭합니다. **Select Target Path** 대화 상자가 표시됩니다.
3. **OK**를 클릭하여 기본값(/)을 그대로 사용하거나 **Root Path...** 단추를 클릭하여 더 자세한 선택 옵션이 들어 있는 **Shared Root Paths** 대화 상자를 표시합니다.
4. 원하는 루트를 선택하고 **OK**를 클릭하여 **Select Target Path** 대화 상자로 돌아갑니다.
5. **OK**를 클릭하여 **Add Targets** 대화 상자로 돌아갑니다.
6. **OK**를 클릭합니다. 이제 대상을 추가로 표시했습니다.

대상을 삭제하려면 **Target Selection** 창에서 하나 이상의 대상을 선택한 다음 **Actions**→**Unmark for Install**을 선택합니다.

## 대상 그룹 저장

차후 세션을 위해 대상 목록을 다시 사용하고자 할 수 있습니다. 그렇게 하려면 다음을 수행합니다.

1. **Actions**→**Save Target Group...**을 선택합니다.

**Select File** 대화 상자가 나타납니다. 대상 그룹이 이미 존재하면 대화 상자 맨 아래의 텍스트 상자에 첫 번째 파일 경로가 표시됩니다. 새로운 그룹 이름을 입력하거나 기존 그룹을 재사용합니다. (현재 목록을 기존 대상 그룹에 저장하면 그 그룹이 대체됩니다.) 그룹이 다음 디렉토리에 저장됩니다.

```
$HOME/.sw/targets
```

2. 그룹을 저장하려면 **OK**를 클릭합니다.

그러면 방금 표시한 대상 선택 사항(Marked? 열이 Yes인 모든 대상)이 모두 저장됩니다. 이 그룹은 모든 후속 대상 그룹 선택 사항에 대한 **Select File** 대화 상자에 자동으로 표시됩니다.

## 대상 그룹 추가

이전에 저장한 대상 그룹을 다시 사용하려면 다음을 수행합니다.

1. **Actions**→**Add Target Group...**을 선택합니다. **Select File** 대화 상자가 나타납니다. 기존의 대상 그룹이 모두 목록에 표시됩니다.
2. 원하는 대상 그룹을 선택하고 **OK**를 클릭합니다.  
이미 표시한 다른 대상과 함께 해당 그룹의 대상이 표시되었습니다.



## 원격 작업 설정

SD-UX는 액세스 제어 목록을 사용하여 저장소의 소프트웨어 제품을 작성, 수정 또는 읽으려 하거나 루트 파일 시스템에 소프트웨어를 설치하려는 사용자에게 권한을 부여합니다. (ACL은 9장, 233페이지의 “SD-UX 보안”에 자세히 설명되어 있습니다.) 원격 작업을 가능하게 하려면 관리할 각 원격 시스템에서 특별한 HP ServiceControl Manager 파일 세트를 설치해야 합니다. 그런 다음 원격 작업 GUI를 사용 가능하게 설정할 수 있습니다.

1. 루트 자격으로 컨트롤러 시스템에서 다음 명령을 입력합니다.

```
swreg -l depot /var/opt/mx/depot11
```

- 원격 호스트에 HP-UX 10.20이 실행 중인 경우 동일한 명령을 사용하되 **depot10**을 **depot11**로 대체합니다.
- 이는 원격 시스템을 사용할 수 있도록 하는 데 필요한 저장소 공유를 설정하는 것입니다.

2. 루트 자격으로 관리할 각 원격 시스템에서 다음 명령을 입력합니다. 이는 컨트롤러 시스템에서의 루트 액세스를 허용하는 방식으로 루트, 호스트 및 템플릿 ACL을 설정합니다.

```
swinstall -s central_node:/var/opt/mx/depot11 \
AgentConfig.SD-CONFIG
```

- 이 예에서 **central\_node**는 컨트롤러의 이름입니다.
- 원격 호스트에 HP-UX 10.20이 실행 중인 경우 동일한 명령을 사용하되 **depot10**을 **depot11**로 대체합니다.
- OpenView Software Distributor를 사용하여 이전에 설치된 원격 시스템은 이 단계가 필요 없습니다.
- Software Distributor는 다른 ServiceControl Manager 파일 세트를 필요로 하지 않습니다.

3. (선택) 중앙 컨트롤러 시스템에 한해 다음 명령을 입력하여 원격 작업 GUI 인터페이스를 사용할 수 있도록 합니다.

```
touch /var/adm/sw/.sdkey
```

(이 단계는 HP ServiceControl Manager 내에서 SD-UX를 사용할 때는 필요하지 않습니다.)

---

## 원격 작업 자습서

이 자습서에서는 원격 작업 사용자 인터페이스와 소프트웨어를 다른 시스템으로 배포하는 일반적인 흐름에 대해 소개합니다. 또한 배포 작업을 미리 보고, 예약하며 모니터링하는 방법에 대해서도 배우게 됩니다. 여기에서는 예시된 GUI로 `swinstall`을 사용하고 있지만 `swcopy`와 `swremove` GUI 프로그램도 거의 동일합니다. 이 자습서에서 얻은 지식을 위의 작업에 적용할 수 있습니다.

기본적인 작업을 다양하게 변화시켜 시험해보기 위해 이 자습서를 두 번 이상 살펴 보고자 할 수도 있습니다.

### 자습서 설치

- 컨트롤러 시스템과 원격 테스트 시스템에 원격 작업을 설치합니다(177페이지의 “원격 작업 설정” 참조).

- 컨트롤러 시스템에서 루트 자격으로 원격 작업 GUI를 사용 가능하게 설정합니다.

```
touch /var/adm/sw/.sdkey
```

- PATH 변수에 `/usr/sbin`이 들어 있는지 확인합니다. 확인하려면 다음과 같이 입력합니다.

```
echo $PATH
```

- 다음과 같이 입력하여 `DISPLAY` 변수가 적절하게 설정되어 있는지 확인합니다.

```
echo $DISPLAY
```

- 예제가 설치되었는지 확인합니다. 다음을 입력합니다.

```
swlist SW-DIST.SD-EXAMPLES
```

- 예제 패키지(즉, `SD-DATABASE`)가 들어 있는 저장소를 만듭니다.

```
cd /usr/lib/sw/examples/swpackage/depot_src
```

```
swpackage -s psf @ /var/adm/sw/examples/depot
```

```
swreg -l depot @ /var/adm/sw/examples/depot
```

- 소프트웨어가 저장소에 있는지 및 대상으로 배포할 수 있는지 확인하려면 다음을 입력합니다.

```
swlist -s /var/adm/sw/examples/depot
```

결과 목록에 `SD-DATABASE`가 있어야 합니다.

## 단일 대상 설치 수행 방법

**개요** 자습서는 다음과 같은 단계로 구성됩니다.  
**표 6-2** 설치 단계

### 설치 단계 개요

<b>I. 시작</b>	작업 브라우저를 시작합니다.
<b>II. 대상 선택</b>	소프트웨어가 설치될 대상을 지정합니다. 기본 로컬 대상을 사용하거나 다른 대상을 지정합니다.
<b>III. 소스 선택</b>	Specify Source 대화 상자에서 소프트웨어를 가져와 설치할 소프트웨어 저장소의 위치를 입력합니다.
<b>IV. 소프트웨어 선택</b>	Software Selection 창을 사용하여 설치할 소프트웨어를 선택합니다.
<b>V. 설치 기본 설정 지정</b>	Install Preference 대화 상자를 사용하여 미리 보기 또는 예약 옵션을 설정합니다.
<b>VI. 분석 및 설치</b>	실제 소프트웨어 설치 또는 미리 보기를 수행합니다.
<b>VII. 결과 모니터링</b>	작업 브라우저 GUI를 사용하여 작업 진행 상황 및 결과를 모니터링합니다.
<b>VIII. 작업 제거</b>	작업 브라우저를 사용하여 완료된 작업을 삭제합니다.

**단계 I: 시작** 설치 세션을 시작하려면 다음을 수행합니다.

1. 다음을 입력하여 작업 브라우저를 시작합니다.

**sd**

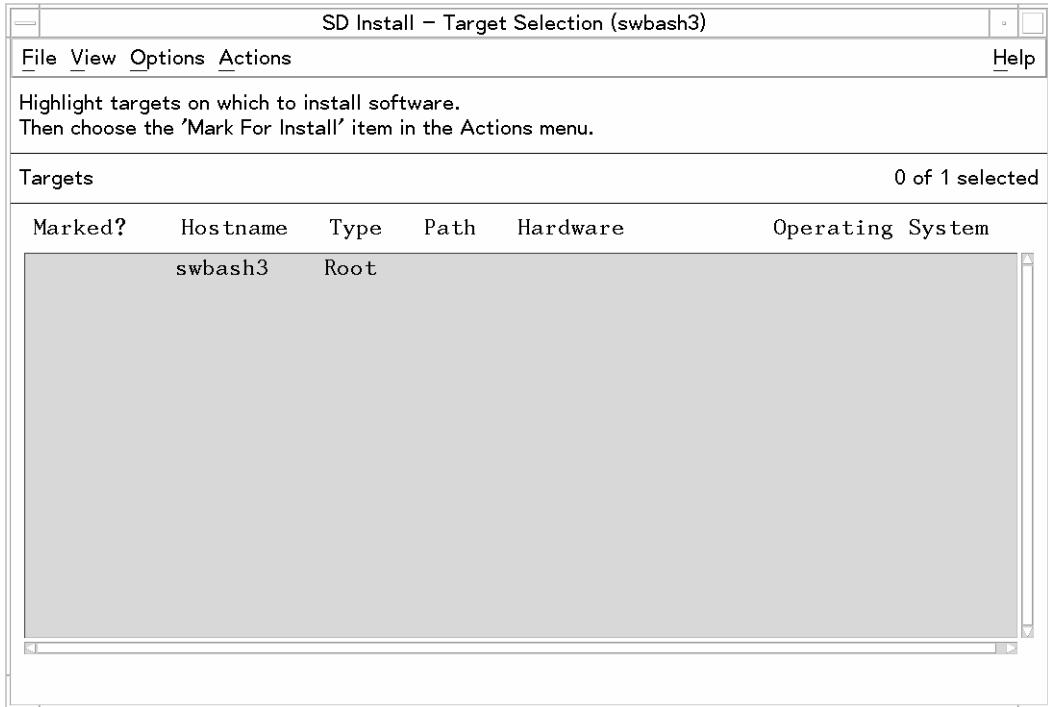
2. Job Browser 창에서 다음을 선택합니다.

**Actions→Create Job→Install Software...**

창의 맨 아래에 Invoking a swinstall process라는 메시지가 표시된 다음 Target Selection 창이 표시됩니다.

**단계 II:** Target Selection 창에 로컬 기본 대상이 표시됩니다. 대상은 설치를 수행하려는 시스템입니다. 아래 예에서 대상은 swbash3 시스템입니다. 기본적으로 현재 시스템이 나열됩니다(그림 6-3, “Target Selection 창”).

그림 6-3 Target Selection 창



설치에 사용할 대상을 지정합니다.

**1. 로컬 기본값의 경우:**

- a. 마우스 단추를 클릭하여 로컬 대상 시스템을 강조 표시합니다. 그런 다음 **Actions→Mark for Install**을 선택합니다. (또는 마우스 오른쪽 단추를 클릭하여 팝업 메뉴를 표시한 다음 **Mark for Install**을 선택합니다.)
- b. **Actions→Show Software for Selection...**을 선택합니다.

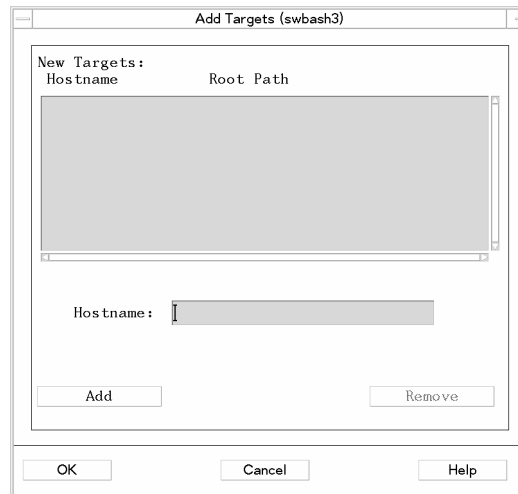
그러면 Specify Source 대화 상자가 표시됩니다. 이 자습서를 처음으로 학습하는 것이라면 바로 182페이지의 “단계 III: 소스 선택”으로 건너뛰십시오. 이 자습서를 학습하고 나면 원격 대상을 사용하여 단계 I을 다시 수행해 보십시오.

— 또는 —

**원격 대상의 경우:** 다른 대상에 설치하려면 **Actions**→**Add Targets**를 선택합니다. 그러면 **Add Targets** 대화 상자(그림 6-4, “**Add Target** 대화 상자(여러 대상 및 기본값이 아닌 대상의 경우)”)가 표시됩니다.

2. **Hostname:** 영역(예: `system_two`)에 대상 이름을 입력하고 **Add**를 선택합니다. 그러면 **Select Target Path** 대화 상자가 표시됩니다.

그림 6-4 **Add Target** 대화 상자(여러 대상 및 기본값이 아닌 대상의 경우)



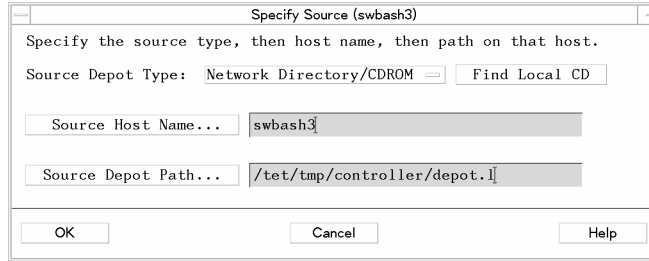
3. **OK**를 선택하여 현재의 루트 경로(/)를 사용합니다. 그러면 **Add Targets** 대화 상자가 표시됩니다.
4. **Add Targets** 대화 상자에서 **OK**를 선택합니다. 그러면 사용자가 선택한 대상으로 **Target Selection** 창이 업데이트됩니다. **Yes**가 **Marked** 열에 표시되어 대상이 설치 표시가 되었음을 나타냅니다.
5. **Actions**→**Show Software for Selection**을 선택합니다. **Specify Source** 대화 상자가 표시됩니다.

### 단계 III: 소스 선택

이 단계에서 Specify Source 대화 상자를 사용하여 Source Host Name(저장소가 있는 소스 시스템)과 Source Depot Path(소프트웨어가 들어 있는 저장소 경로)를 선택합니다.

그림 6-5

#### Specify Source 대화 상자



1. Specify Source 대화 상자에서 **Source Host Name...** 필드에 컨트롤러 이름이나 원격 테스트 시스템 이름이 표시되고 **Source Depot Path...** 필드에 작성한 예제 저장소(/var/adm/sw/examples/depot)가 표시됩니다.

이 대화 상자에서 다음과 같은 작업을 수행할 수도 있습니다.

- **Source Host Name...** 단추를 클릭하여 선택 가능한 호스트 목록을 표시합니다.
- **Source Depot Path...**를 클릭하여 선택 가능한 등록된 저장소 목록을 표시합니다.

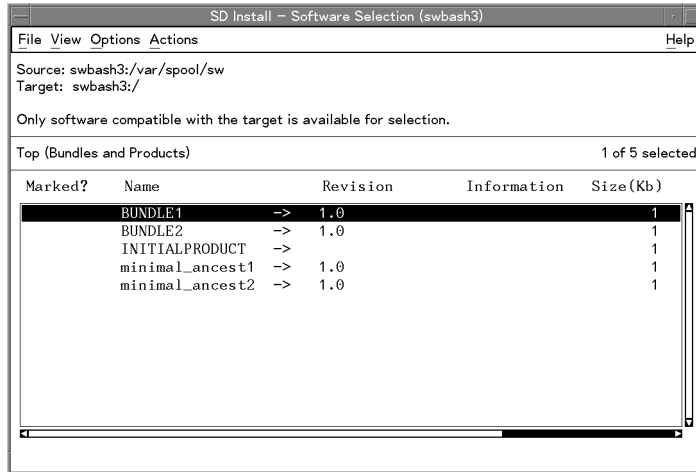
**OK**를 클릭합니다. **Software Selection** 창이 표시됩니다(그림 6-6, “소프트웨어 선택 창”). 이 창에는 선택한 저장소에서 사용 가능한 모든 소프트웨어가 표시됩니다.

**단계 IV: 소프트웨어  
선택**

Software Selection 창을 사용하여 설치할 소프트웨어를 선택합니다.

그림 6-6

소프트웨어 선택 창



1. 마우스 왼쪽 단추로 SD-DATABASE(즉, 예제 소프트웨어)를 클릭하여 강조 표시합니다.
2. **Actions** → **Mark for Install**을 선택합니다. (또는 마우스 오른쪽 단추를 클릭하여 팝업 메뉴를 표시한 다음 **Mark for Install**을 선택합니다.)

SD-DATABASE에 대해 Marked? 열이 Yes로 설정됩니다.

표 6-3

소프트웨어 선택 목록

Software Selection 창 객체 목록
Software Selection 창 객체 목록은 계층 구조로 이루어집니다. 목록의 각 객체를 열고 그 안의 객체를 표시할 수 있습니다. 열 수 있는 다른 객체가 들어 있는 객체는 객체 이름 옆에 화살표(→)가 표시됩니다.

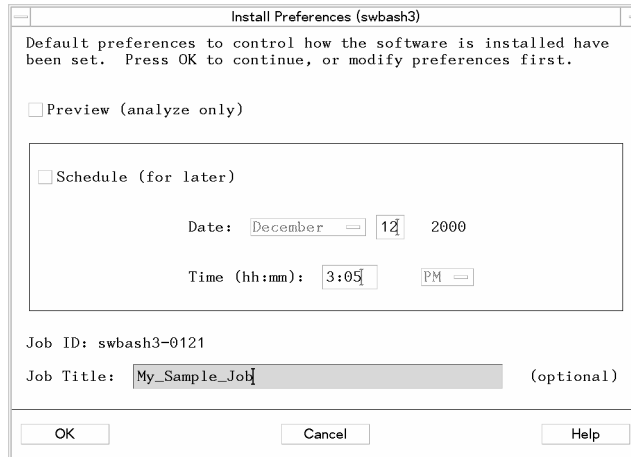
표 6-3 소프트웨어 선택 목록(계속)

Software Selection 창 객체 목록
<p>예를 들면 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>SD-DATABASE 제품의 하위 제품을 보려면 해당 제품을 두 번 클릭합니다. 객체 목록에 하위 제품이 표시됩니다. 하위 제품을 열려면 해당 이름을 두 번 클릭합니다. (또는 이름을 강조 표시한 다음 <b>Actions</b>→<b>Open Item</b>을 선택합니다.)</li> <li>객체를 닫고 이전 목록으로 돌아가려면 목록의 첫 번째 항목(.. (go up))을 두 번 클릭하거나 해당 항목을 강조 표시한 다음 <b>Actions</b>→<b>Close Level</b>을 선택합니다.</li> </ul> <p>제품은 함께 나열되지만 하위 제품과 파일 세트는 제품을 열면 동일한 목록에 표시될 수 있습니다.</p>

3. **Actions**→**Install**을 선택합니다. 그러면 **Install Preferences** 대화 상자(그림 6-7, “**Install Preferences** 대화 상자”)가 표시됩니다.

**단계 V: 설치 기본 설정 지정** 대화 상자에서는 **Preview**, **Schedule** 및 **OK**와 같은 선택 사항을 제공합니다. **Job Title**도 입력할 수 있습니다.

그림 6-7 **Install Preferences** 대화 상자



1. **Job Title** 다음의 텍스트 영역을 선택하고 다음을 입력합니다.



### SDTESTJOB

이것은 설치 작업의 이름입니다.

2. **OK**를 선택하여 소프트웨어를 지금 설치합니다.

자습서와 같이 단일 대상 설치의 경우 **Install Analysis** 대화 상자가 표시됩니다 (186페이지의 그림 6-8).

3. 자습서를 처음 학습하는 경우 단계 **V**로 이동하십시오.

4. (선택) 작업 미리 보기

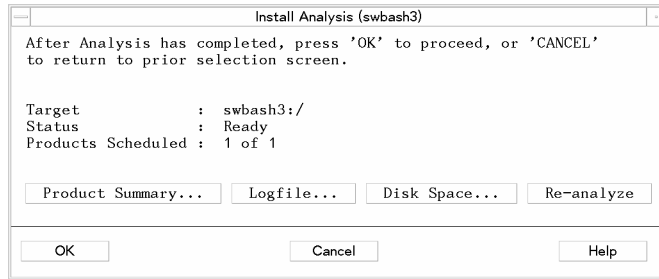
- a. **Preview** 단추를 선택합니다. 이는 **SD**에게 설치하지 않고 소프트웨어를 분석하도록 알려 주는 것입니다.
- b. **OK**를 클릭합니다. **Install Analysis** 대화 상자가 표시됩니다. 이 대화 상자를 통해 단일 대상 작업의 분석을 모니터링할 수 있습니다. 또한 로그 파일과 제품 요약 정보를 찾아볼 수도 있습니다.
- c. 대상 상태가 **Ready**(분석 성공)면 **OK**를 선택합니다. 그러면 **Software Selection** 창으로 돌아갑니다.
- d. **Actions**→**Install**을 선택합니다. **Install Preferences** 대화 상자가 표시됩니다.
- e. **Install Preference** 대화 상자에서 **OK**를 선택하여 설치를 계속 진행합니다.  
이 자습서를 반복 학습하고 여러 대상을 사용하는 작업을 미리 보는 경우 **Install Analysis** 대화 상자가 표시되지 않음을 알 수 있습니다. 작업 브라우저에서는 작업 진행 상황만 미리 볼 수 있습니다. 자세한 내용은 187페이지의 “단계 VII: 결과 모니터링”을 참조하십시오.

5. (선택) 작업 예약

- a. **Schedule** 단추를 선택합니다. 그러면 작업 실행 시간 및 날짜를 지정할 수 있는 필드가 활성화됩니다. (예를 들어 로그인 사용자가 거의 없는 자정에 작업을 예약할 수도 있습니다.)
- b. 예약 정보를 지정한 다음 **OK**를 클릭합니다. 시스템은 작업이 예약되었음을 나타내는 메시지를 내보냅니다.
- c. 대화 상자에서 **OK**를 클릭합니다. **Target Selection** 창이 다시 표시됩니다. **File**→**Exit**를 선택하여 예약된 작업을 모니터링할 수 있는 작업 브라우저로 돌아갑니다.

**단계 VI: 분석 및 설치** SD-UX는 실제 설치, 복사 또는 제거 작업을 수행하기 전에 대상을 분석합니다.(단계 V에서 작업 미리 보기를 설정한 경우 설치하는 분석이 완료된 후 중단됩니다.)

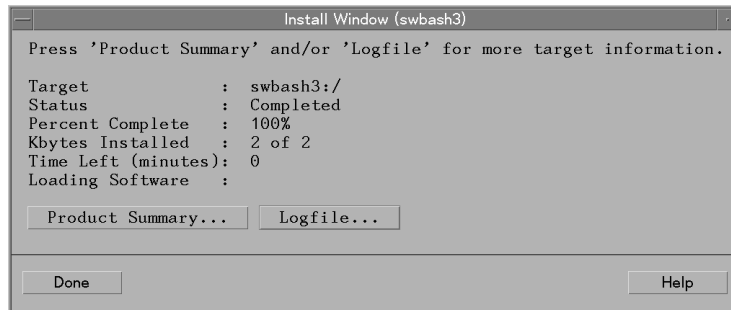
그림 6-8 **Install Analysis 대화 상자**



1. 분석이 완료되면 사용자가 선택한 대상의 상태에 Ready가 표시되어 분석 중 오류나 경고가 발생하지 않았음을 나타냅니다. **OK**를 선택하여 설치를 계속 진행합니다.

**Install Window** 대화 상자(그림 6-9, “Install Window 대화 상자”)가 나타나고 설치가 자동으로 시작됩니다. 대화 상자의 상태가 Completed가 되면 설치가 정상적으로 완료된 것입니다.

그림 6-9 **Install Window 대화 상자**



2. **Done**을 선택하여 Install Window 대화 상자를 종료합니다. 그러면 Target Selection 창으로 돌아갑니다.
3. **File→Exit**를 선택하여 작업 브라우저로 돌아갑니다.
4. (선택) 설치할 다른 대상을 선택합니다(즉, **Actions→Mark for Install**).

**단계 VII: 결과 모니터링** Target Selection 창을 종료하면 작업 브라우저로 돌아갑니다. 작업 목록의 아이콘이 작업 상태를 나타내기 위해 변경됩니다. 아이콘이 다르면 작업 상태도 다릅니다. (샘플 아이콘은 196페이지의 “작업 브라우저 아이콘”을 참조하십시오.)

SDTESTJOB이라는 레이블이 붙은 작업은 확인 표시나 자 아이콘 중 하나와 함께 표시되어야 합니다. 작업 목록에서 SDTESTJOB에 대한 상태 정보를 확인하려면 다음을 수행합니다.

- SDTESTJOB을 두 번 클릭하여 **Job Results** 대화 상자를 호출합니다.
- 대상을 두 번 클릭하여 자세한 대상 로그를 표시합니다.
- 대화 상자를 본 다음 **OK**를 클릭하여 각 대화 상자를 닫습니다.

— 또는 —

1. SDTESTJOB 아이콘을 선택합니다.
2. **Actions**→**Show Job Description...**을 선택합니다. **Job Description** 대화 상자가 표시됩니다. 여기에는 작업 속성, 관련된 소프트웨어 및 대상이 모두 표시됩니다.
3. (선택) **Show Options...**를 선택하여 작업 옵션 설정을 볼 수 있습니다.
4. (선택) **Show Results...**를 선택하여 최신 작업 상태를 볼 수 있습니다.

**단계 VIII: 작업 제거** 자습서를 실행한 다음 작업 브라우저를 사용하여 작업 예제를 제거합니다.

1. SDTESTJOB 아이콘을 클릭합니다.
2. **Actions**→**Remove Job...**을 선택합니다. **Remove a Job** 대화 상자가 표시됩니다.
3. **OK**를 선택합니다.

— 또는 —

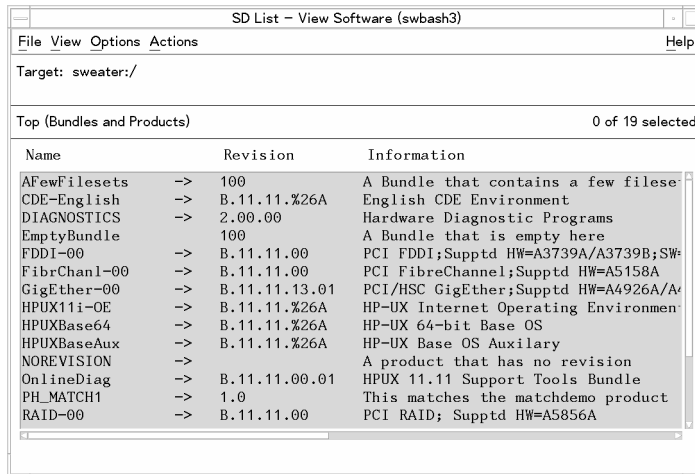
1. 작업 아이콘을 선택하고 마우스 오른쪽 단추를 클릭합니다.
2. 팝업 메뉴에서 **Actions**→**Remove Job...**을 선택합니다. **Remove a Job** 대화 상자가 나타나고 SDTESTJOB이 표시됩니다.
3. **OK**를 선택합니다. 작업 브라우저에서 SDTESTJOB 아이콘이 없어지고 해당 작업이 SD-UX 데이터베이스에서 제거됩니다.

## 원격 대화식 swlist

원격 작업의 경우 `swlist -i` 명령은 원격 호스트에 설치되어 있는 소프트웨어를 대화식으로 나열할 수 있는 목록 브라우저를 시작합니다. 원격 작업과 로컬 작업의 유일한 차이점은 **Software Browsing** 창의 메시지 영역에 표시되는 대상의 이름입니다.

그림 6-10

swlist 브라우저



swlist에 대한 자세한 내용은 98페이지의 “소프트웨어 나열(swlist)”을 참조하십시오.

## 명령줄에서의 원격 작업

명령줄에서 원격 작업을 실행하는 것은 로컬 작업과 거의 동일합니다. 주요 차이점은 다음과 같습니다.

- 대상 선택 사항을 지정해야 합니다.
- 207페이지의 “명령줄에서 작업 모니터링”에서 설명된 대로 `swjob` 명령을 사용하여 작업을 모니터링할 수 있습니다.
- 추가 명령 옵션을 사용하여 작업을 예약하고 관리할 수 있습니다. 211페이지의 “명령 옵션을 사용하여 작업 관리 및 조정”을 참조하십시오.

### 대상 선택 사항

원칙적으로 원격 작업에는 원격 대상을 지정해야 합니다. 대상이 로컬 호스트 시스템의 디렉토리일 수도 있는 로컬 작업에서와는 달리 원격 작업의 대상으로 원격 시스템을 지정해야 합니다.

```
swinstall -s sw_server cc pascal @ hostA hostB hostC
```

이 명령은 세 대의 원격 호스트에 C와 Pascal 제품을 설치합니다.

### 구문

소프트웨어 및 소스 저장소를 선택한 다음 대상을 선택합니다. 피연산자는 “@” 문자로 구분됩니다. 이 구문은 해당 명령이 선택된 대상에서 작동함을 의미합니다.

`target_selections` 구문은 이를 필요로 하는 모든 **Software Distributor** 명령에 대해 동일합니다.

```
@ [host][:]/[directory]
```

- @ 문자는 단 하나만 필요합니다.
- 호스트 이름, 도메인 이름 또는 인터넷 주소 등으로 호스트를 지정할 수 있습니다. 디렉토리는 절대 경로로 지정해야 합니다.
- : (콜론)은 호스트와 디렉토리를 모두 지정하는 경우에 필요합니다.

- 일부 시스템에서 @ 문자는 강제 종료(kill) 기능으로 사용됩니다. @ 문자가 시스템상의 다른 기능에 매핑되어 있는지 알아보려면 시스템에서 stty를 입력합니다. 다른 기능에 매핑되어 있는 경우 해당 매핑을 제거하거나 변경하거나 또는 \@ 을 사용합니다.

### 대상 파일

입력 파일을 사용하여 대상을 지정할 수도 있습니다. 대상 선택 입력 파일을 사용하여 긴 대상 목록을 지정함으로써 명령줄을 더 짧게 만들 수 있습니다. 대상 선택 파일을 사용하는 경우 파일 이름을 하나만 지정해야 합니다.

-t 명령줄 옵션으로 대상 파일을 지정할 수 있습니다. 예를 들면 다음과 같습니다.

```
swinstall -f mysoft -s /mnt/cd -t mytargs
```

이 예에서 mytargs 파일(기본 디렉토리에 있음)에 /mnt/cd 저장소에 대한 대상 선택 사항 목록이 들어 있습니다.

대상 파일에서 공백 줄과 주석(#으로 시작되는 줄)은 무시됩니다. 각 선택 사항은 각기 다른 줄에 지정해야 하고 호스트 이름이나 네트워크 주소로 구성되어야 하며 그 다음에 선택적으로 콜론과 전체 경로가 나올 수 있습니다.

```
host[:/directory]
```

## 예

### swacl

원격 호스트 gemini에서 글로벌 제품 템플릿 ACL을 나열하려면 다음과 같이 입력합니다.

```
swacl -l global_product_template @ gemini
```

### swask

원격 시스템 swposix의 저장소 /var/spool/sw에서 모든 요청 스크립트를 실행시키고 다시 동일한 저장소로 응답 파일을 작성하여 보내려면 다음과 같이 입력합니다.

```
swask -s swposix:/var/spool/sw \*
```

### swconfig

세 대의 원격 호스트에 C와 Pascal 제품을 구성하려면 다음과 같이 입력합니다.

```
swconfig cc pascal @ hostA hostB hostC
```

### swcopy

C와 Pascal 제품을 로컬 저장소 한 곳과 원격 저장소 두 곳으로 복사하려면 다음과 같이 입력합니다.

```
swcopy -s sw_server cc pascal @ /var/spool/sw \  
hostA:/tmp/sw hostB
```

### swinstall

세 대의 원격 호스트로 C와 Pascal 제품을 설치하려면 다음과 같이 입력합니다.

```
swinstall -s sw_server cc pascal @ hostA hostB hostC
```

### swjob

swjob 명령을 통해 명령줄에서 작업을 모니터링할 수 있습니다. 작업에 대한 자세한 내용은 7장, 193페이지의 “작업 및 작업 브라우저 사용” 및 207페이지의 “명령줄에서 작업 모니터링”을 참조하십시오.

hostA-0001 작업에 대해 원격 시스템 TargetA의 에이전트 로그를 나열하려면 다음과 같이 입력합니다.

```
swjob -a log hostA-0001 @ targetA/
```

### swlist

세 대의 원격 호스트에 있는 C 제품을 나열하려면 다음과 같이 입력합니다.

```
swlist cc @ hostA hostB hostC
```

### swreg

세 대의 원격 호스트에 있는 기본 저장소 등록을 해제하려면 다음과 같이 입력합니다.

```
swreg -u -l depot /var/spool/sw @ hostA hostB hostC
```

### swremove

세 대의 원격 호스트에서 C와 Pascal 제품을 제거하려면 다음과 같이 입력합니다.

```
swremove cc pascal @ hostA hostB hostC
```

### swverify

세 대의 원격 호스트에서 C와 Pascal 제품을 확인하려면 다음과 같이 입력합니다.

```
swverify cc pascal @ hostA hostB hostC
```

원격 작업 개요  
명령줄에서의 원격 작업



## 7

## 작업 및 작업 브라우저 사용

이 장에서는 원격 작업용 SD-UX 작업 및 작업 브라우저 인터페이스에 대해 설명합니다. 원격 작업에 대한 추가 정보는 6장, 167페이지의 “원격 작업 개요”를 참조하십시오.

표 7-1

## 이 장의 항목

설명하는 항목
194페이지의 “소개”
195페이지의 “작업 브라우저 사용하기”
207페이지의 “명령줄에서 작업 모니터링”
211페이지의 “명령 옵션을 사용하여 작업 관리 및 조정”

## 소개

작업 브라우저 GUI는 원격 작업 관리를 위한 대화식 인터페이스로, 작업 브라우저를 통해 다음과 같은 작업을 수행할 수 있습니다.

- 작업 작성, 설치 또는 제거
- 작업 상태 및 로그 파일 모니터링
- 작업 정보 나열
- 작업 예약

`swjob` 명령을 통해 명령줄에서 작업을 모니터링할 수 있습니다. 다양한 명령 옵션은 작업 및 원격 작업의 성능을 관리 및 조정하는 데 도움이 됩니다.

---

## 참고

터미널 사용자 인터페이스(TUI)를 작업 브라우저와 함께 사용하는 경우는 약간의 제한이 뒤따릅니다.

- 오류 처리 메시지가 화면에 표시될 수 있습니다. 이 경우 **Ctrl-L** 키를 입력하여 화면을 새로 고칩니다.
- **Actions**→**Job Description**→**Show Options**를 표시하는 경우 전체 화면을 보려면 스크롤해야 합니다.

---

## 작업 브라우저 시작하기

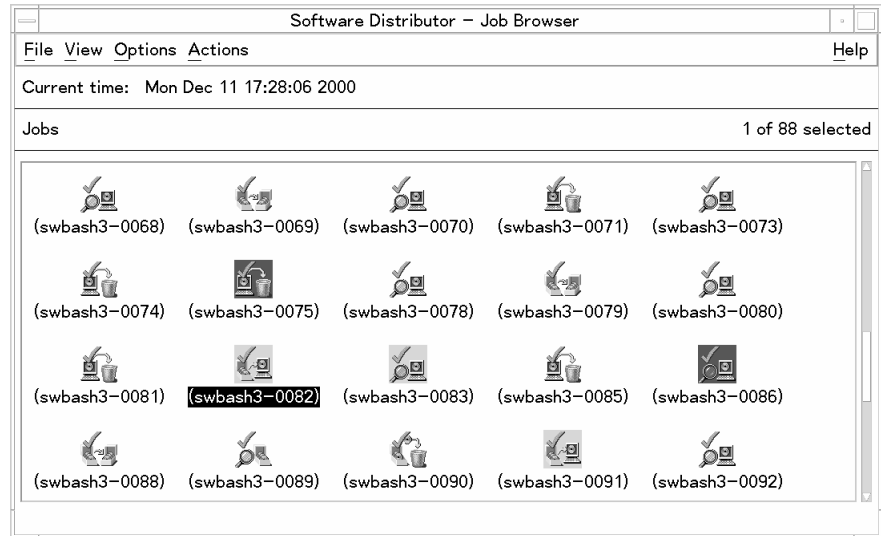
작업 브라우저를 시작하려면 다음과 같이 입력합니다.

**sd**

Job Browser 창이 화면에 표시됩니다.

## 작업 브라우저 사용하기

그림 7-1 SD Job Browser 창



창은 다음 세 부분으로 나뉘어져 있습니다.

- 메뉴 표시줄. 여기에는 37페이지의 “GUI 및 TUI 명령 사용”에서 설명된 대부분의 표준 SD-UX 메뉴가 있습니다. 작업 브라우저에만 있는 고유한 메뉴 항목은 이 장의 뒷 부분에서 설명합니다.
- 현재 시간을 표시하는 메시지 영역
- 각 작업을 나타내는 작업 아이콘(기본값)을 표시하는 작업 목록. 아이콘 아래에 작업 제목이 있습니다. 작업을 선택하려면 해당 아이콘을 클릭합니다. 자세한 내용은 196페이지의 “작업 브라우저 아이콘”을 참조하십시오.

### 참고

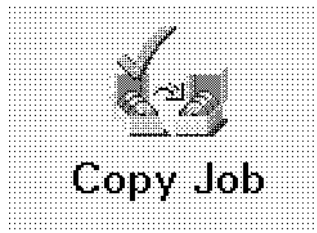
작업이 만들어질 때까지 작업 목록은 비어 있습니다.

## 작업 브라우저 아이콘

- 시계는 작업이 예약되었으나 아직 실행되지 않았음을 나타냅니다.
- 체크 표시는 작업이 완료되었음을 나타냅니다.
- 자는 작업이 활성화 상태임을 나타냅니다.
- 빨간색 배경은 작업에 오류가 있음을 나타냅니다.
- 노란색 배경은 작업에 경고가 있음을 나타냅니다.

그림 7-2

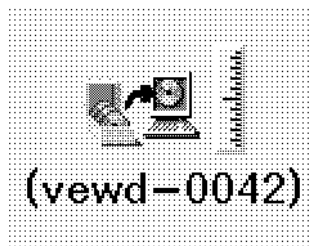
### 복사 아이콘



이 아이콘은 복사 작업(저장소에서 저장소로)을 나타냅니다. 체크 표시는 작업이 완료되었음을 나타냅니다.

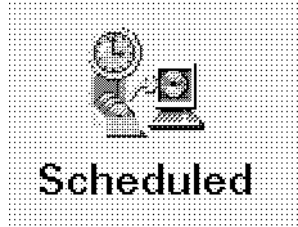
그림 7-3

### 활성화된 설치 작업 아이콘



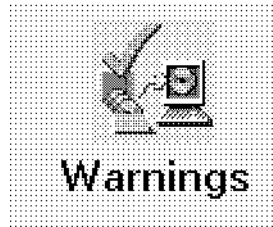
이 아이콘은 설치 작업을 나타냅니다. 옆에 있는 자는 작업이 활성화되어 있음을 나타냅니다.

그림 7-4 예약된 설치 작업 아이콘



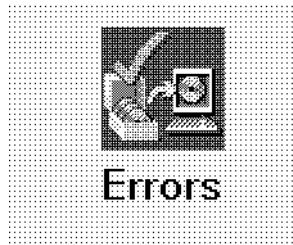
이 아이콘은 나중에 실행하도록 예약된 설치 작업을 나타냅니다. 시계는 예약된 작업임을 나타냅니다.

그림 7-5 경고가 발생한 설치 작업 아이콘



이 아이콘은 설치가 완료되었지만 경고가 발생한 설치 작업을 나타냅니다. 아이콘 주위의 배경은 노란색입니다.

그림 7-6 오류가 발생한 설치 작업 아이콘



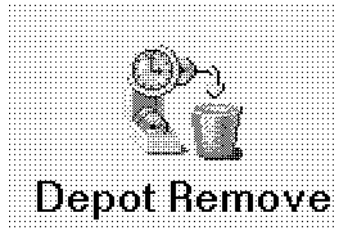
이 아이콘은 설치가 완료되었지만 오류가 발생한 설치 작업을 나타냅니다. 아이콘 주위의 배경은 빨간색입니다.

그림 7-7 설치된 소프트웨어의 예약된 제거 작업 아이콘



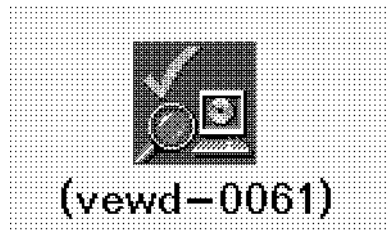
이 아이콘은 설치된 소프트웨어에서 예약된 제거 작업을 나타냅니다.

그림 7-8 저장소 소프트웨어의 예약된 제거 작업 아이콘



이 아이콘은 저장소의 소프트웨어에서 예약된 제거 작업을 나타냅니다.

그림 7-9 확인 작업 아이콘



이 아이콘은 완료되었지만 오류가 발생한 확인 작업(확대경으로 표시)을 나타냅니다. 아이콘 주위의 배경은 빨간색입니다.

## File 메뉴

**File** 메뉴에는 다음과 같은 옵션이 있습니다.

<b>Search</b>	작업 ID나 제목에 대한 텍스트 검색을 수행합니다.
<b>Print</b>	작업 목록을 인쇄합니다.
<b>Exit</b>	작업 브라우저를 종료합니다.

### 작업 목록 인쇄

이 옵션은 작업 목록을 지정된 프린터로 인쇄하거나 파일에 저장합니다. (작업 목록은 등록 정보별로 나열된 경우에만 인쇄할 수 있습니다. 자세한 내용은 199페이지의 “View 메뉴”를 참조하십시오.) 이름 및 아이콘별(기본값) 작업 목록이 표시되는 경우 이 메뉴 항목은 회색으로 표시되어 선택할 수 없습니다. 작업 목록을 인쇄하려면 다음을 수행합니다.

1. **View**→**By Properties**를 선택합니다.
2. **File**→**Print...**를 선택합니다. **Print Objects** 대화 상자가 표시됩니다.
3. **Print Objects** 대화 상자에 필수 정보를 입력한 다음 **OK**를 선택합니다.

## View 메뉴

**View** 메뉴를 사용하여 작업 브라우저 정보가 표시되는 방식을 변경할 수 있습니다. 이 메뉴의 표준 선택 항목(**Columns...**, **Filter...**, **Sort...** 및 **Save View as Default**)은 44페이지의 “소프트웨어 보기 변경—View 메뉴”에 설명된 것과 일치합니다. 하지만 **Columns...** 선택 항목은 **View**→**By Properties** (아래 설명됨)에 대해서만 유효합니다.

그림 7-10

등록 정보별로 표시된 작업

ID	State	Progress	Results	Title
swbash3-0080	Complete			(swbash3-0080)
swbash3-0081	Complete			(swbash3-0081)
swbash3-0082	Complete		1 With Warnings	(swbash3-0082)
swbash3-0083	Complete		1 With Warnings	(swbash3-0083)
swbash3-0085	Complete			(swbash3-0085)
swbash3-0086	Complete		1 With Errors	(swbash3-0086)
swbash3-0088	Complete			(swbash3-0088)
swbash3-0089	Complete			(swbash3-0089)
swbash3-0090	Complete			(swbash3-0090)
swbash3-0091	Complete		1 With Warnings	(swbash3-0091)
swbash3-0092	Complete			(swbash3-0092)
swbash3-0093	Complete		1 With Warnings	(swbash3-0093)
swbash3-0094	Complete			(swbash3-0094)
swbash3-0096	Complete			(swbash3-0096)
swbash3-0097	Complete		1 With Warnings	(swbash3-0097)

### 이름 및 아이콘별 보기

- **View→By Name and Icon**을 선택하면 작업 목록이 이름과 아이콘 형식으로 표시됩니다.
- 이 메뉴 항목과 **View→By Properties** 메뉴 선택 항목은 라디오 단추로 작동합니다. 하나를 선택하면 다른 하나가 선택 취소됩니다.

### 등록 정보별 보기

- **View→By Properties**를 선택하면 작업 목록이 등록 정보(작업 제목, ID, 작업 유형, 예약 날짜, 상태, 진행 상황, 결과 및 마지막 업데이트 날짜)별로 표시됩니다.
- **View→Columns...**, **View→Filter...** 또는 **View→Sort...** 메뉴 선택 항목을 변경하면 등록 정보 목록을 표시하는 방법이 달라집니다.
- 이 메뉴 항목과 **By Name and Icon** 메뉴 선택 항목은 라디오 단추로 작동합니다. 하나를 선택하면 다른 하나가 선택 취소됩니다.
- **File→Print....**를 선택하여 이 보기에서 작업 목록을 인쇄할 수 있습니다.



## Options 메뉴

**Options** 메뉴로 작업 브라우저의 선택적 동작을 제어할 수 있습니다.

### 새로 고침 간격 변경

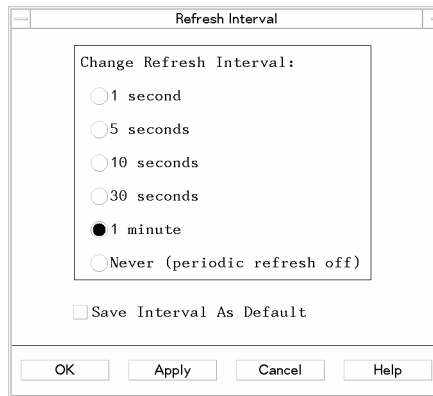
기본적으로 작업 목록은 1분마다 새로 표시됩니다. 많은 수의 작업을 모니터링하는 경우 더 자주 목록을 업데이트하기를 원할 수 있습니다. 또는 성능을 높이기 위해 자동 새로 고침 기능을 끌 수도 있습니다.

목록이 업데이트되는 시간 간격을 변경하려면 다음을 수행합니다.

1. **Options**→**Change Refresh Interval....**을 선택합니다. **Refresh Interval** 대화 상자가 표시됩니다.
2. 목록에서 새로운 새로 고침 간격을 선택합니다.

그림 7-11

### Refresh Interval 대화 상자



- **Apply**는 선택한 간격을 바로 적용합니다.
- **Save Interval as Default**는 선택한 새로 고침 간격을 이후 세션에 대한 기본값으로 설정합니다.

SD-UX 데몬에 대한 새로 고침 간격을 변경하려면 211페이지의 “명령 옵션을 사용하여 작업 관리 및 조정”을 참조하십시오.

### 작업 목록 새로 고침

작업 목록을 즉시 업데이트하려면 **Options**→**Refresh List**를 선택합니다.

### Actions 메뉴

**Actions** 메뉴의 항목을 사용하여 작업 작성과 관리 작업을 수행할 수 있습니다. 작업을 선택했다면 사용 가능한 조치가 해당 작업에 적용됩니다. 작업을 선택하지 않은 경우에는 작업 작성 조치만 사용할 수 있습니다.

### 바로 가기

작업 특정 조치의 팝업 메뉴를 표시하려면 작업 아이콘을 마우스 오른쪽 단추로 클릭한 다음 마우스 왼쪽 단추를 클릭합니다. 이렇게 하면 팝업 **Actions** 메뉴 항목이 나타납니다. 두 개의 마우스 단추 중 하나를 클릭하여 조치를 선택합니다.

작업을 두 번 클릭하면 Job Results 대화 상자(**Actions**→**Show Job Results....**와 같음)가 표시됩니다.

### 작업 작성

작업을 작성하려면 **Actions**→**Create Job**을 선택합니다. 그러면 다른 세션을 시작하는 다음 선택 항목이 포함된 하위 메뉴가 표시됩니다.

표 7-2

### 작업 조치 옵션

작업 조치 및 선택	
<b>Install Software...</b>	swinstall 세션.
<b>Remove Installed Software...</b>	swremove 세션.
<b>Copy Software to a Depot...</b>	swcopy 세션.
<b>Remove Software from a Depot...</b>	swremove -d 세션.

### 작업 결과 표시

**Actions**→**Show Job Results...**를 선택하면 선택한 작업의 결과를 나열하는 Job Results 대화 상자를 표시합니다. (작업 브라우저 아이콘을 두 번 클릭해도 이 대화 상자가 표시됩니다.)

- 객체 목록은 해당 작업, 작업 유형 및 작업 상태에 대한 대상 목록을 표시합니다.
- **Show only warnings or errors** 토글 단추를 사용하여 모든 대상을 표시하거나 작업에 경고 또는 오류가 있는 대상 목록만을 표시하도록 정보 표시를 변경합니다.

- **Show Log** 단추를 선택하거나 객체 목록의 대상을 두 번 클릭하면 해당 대상의 로그가 열립니다.
- **OK**를 선택하여 작업 브라우저로 돌아갑니다. (대화 상자를 닫아도 활성화되어 있는 작업은 계속 표시됩니다.)

그림 7-12

### Job Results 대화 상자



---

### 참고

성능 상의 이유로 한 번에 최대 250개의 대상만 나열됩니다. 작업에 대상이 251개 이상인 경우에는 **Next**와 **Previous** 단추가 표시되어 대상을 250개씩 볼 수 있습니다. Showing only warnings or errors 옵션을 사용하면 표시되는 대상의 수를 줄일 수 있습니다.

---

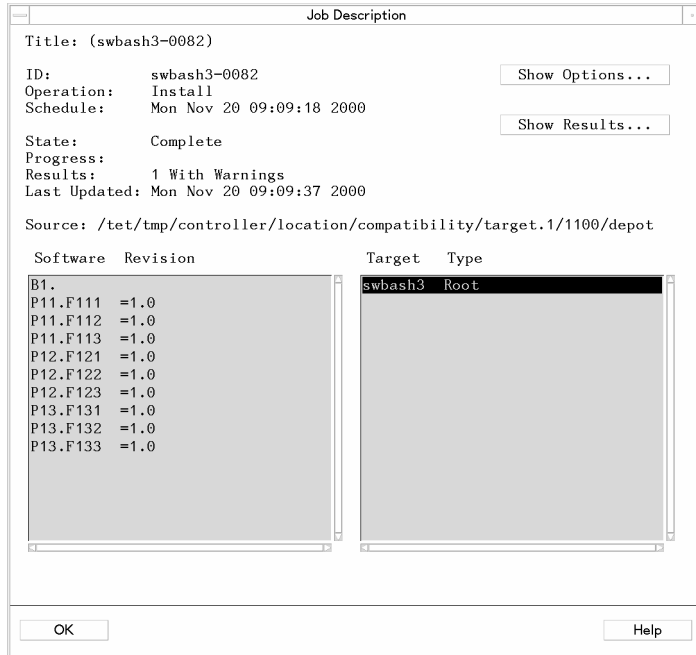
### 작업 설명 표시

**Actions**→**Show Job Description...**을 선택하면 작업을 생성할 때 지정한 모든 정보가 포함된 Job Description 대화 상자를 엽니다.

- 모든 작업 속성(ID, 작업 유형, 예약, 현재 상태, 결과, 마지막으로 업데이트된 날짜, 소스)
- 소프트웨어의 이름과 개정 버전
- 관련된 대상 및 대상 유형:
  - 기본 루트
  - 대체 루트
  - 저장소

그림 7-13

작업 설명 표시



- **Show Options...**를 선택하면 **Job Options** 대화 상자가 표시됩니다. 여기에서 이 작업을 작성할 때 사용된 옵션을 볼 수 있습니다.
- **Show Results...**를 선택하면 작업에 대한 가장 마지막 상태 정보를 보여주는 **Job Results** 대화 상자가 표시됩니다.
- **Job Description** 대화 상자에서 **OK**를 선택하여 작업 브라우저로 돌아갑니다. 해당 작업을 두 번 클릭해도 동일한 정보가 표시됩니다.

## 작업 로그 표시

**Actions**→**Show Job Log...**를 선택하면 선택한 작업의 컨트롤러(요약) 로그를 표시하는 **Job Log** 대화 상자가 표시됩니다. 로그 파일을 새로 고치거나 인쇄할 수 있는 단추가 제공됩니다. 작업 브라우저로 돌아가려면 **OK**를 선택합니다. (선택한 작업이 활성화되어 있지 않거나 완료되지 않은 경우에는 이 메뉴 항목이 회색으로 표시됩니다.)

## 작업 복사

작업 복사는 대상 및 소프트웨어 선택 사항을 **swinstall**, **swcopy** 또는 **swremove**의 새로운 세션에 사용할 수 있게 하는 작업입니다. 선택한 작업에서 동일한 호스트, 소스, 소프트웨어 및 대상 선택 사항을 사용하여 자동으로 새로운 세션이 호출됩니다. 동일한 설정을 그대로 재사용하거나 필요에 따라 변경할 수 있습니다.

이 기능은 **swinstall**, **swremove** 또는 **swcopy** 세션에서 세션 파일을 사용할 때와 같은 장점이 있습니다. 이는 다음과 같은 작업에 도움이 됩니다.

- 동일한 소프트웨어를 새로운 대상 세트에 배포하는 경우
- 새로운 소프트웨어를 이전에 정의된 대상 세트에 배포하는 경우
- 미리 보기에서 완전 실행으로 작업을 변경하는 경우

작업을 복사하려면 다음을 수행합니다.

1. 작업 목록에서 원하는 작업 아이콘이나 항목을 선택합니다.
2. **Actions**→**Create Job from Selected Job....**을 선택합니다.

원래 작업과 일치하는 **SD-UX** 프로그램이 자동으로 호출됩니다. (예를 들어, 설치 작업은 **swinstall**을 호출합니다.)

- **Target Selection** 창은 이전에 지정된 대상을 표시합니다.
- **Software Selection** 창은 이전에 지정된 소프트웨어 목록을 표시합니다.

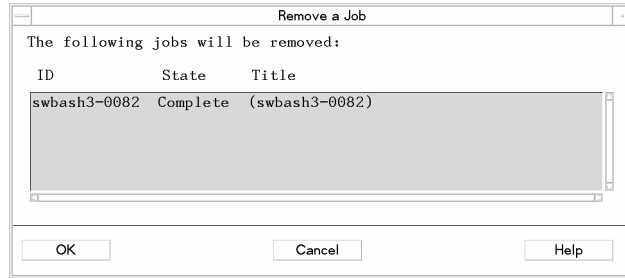
3. 복사된 설정으로 프로그램을 실행하거나 실행 전에 설정을 변경할 수 있습니다.

## 작업 제거

**Actions**→**Remove a Job...** 메뉴 선택 항목을 사용하여 현재 선택한 작업을 작업 목록에서 제거할 수 있습니다. (작업을 두 개 이상 선택하려면 **Ctrl** 키를 누른 상태로 작업 브라우저 창에서 작업을 선택합니다.) **Remove a Job** 대화 상자가 표시되고 선택한 작업에 대한 정보가 나열됩니다.

그림 7-14

**Remove a Job 대화 상자**



- 예약되어 있지만 아직 실행되지 않은 작업을 제거하면 그 작업은 실행되지 않습니다.
- 진행 중인 작업은 제거할 수 없습니다.

---

## 명령줄에서 작업 모니터링

swjob 명령으로 명령줄을 사용하여 작업 정보를 표시하고 모니터링할 수 있습니다. 이 명령은 특정 작업을 확인하려고 할 때 작업 브라우저 대신 빠른 저대역폭을 제공합니다.

### 구문

```
swjob [XToolkit Options][-i][-R][-u][-v][-a attribute]
[-C session_file][-f jobid_file][-s source]
[-S session_file][-t target_file][-x option=value]
[-X option_file][software_selections][@ target_selections]
```

### 옵션 및 피연산자

<b>XToolkit 옵션</b>	swjob -i에 사용할 X window 옵션입니다. 자세한 내용은 55페이지의 “XToolkit 옵션 및 디스플레이 글꼴 변경”을 참조하십시오.
-i	SD 대화식 작업 브라우저를 시작합니다. 자세한 내용은 195페이지의 “작업 브라우저 사용하기”를 참조하십시오.
-u	swjob이 지정된 작업을 제거하도록 합니다.
-v	사용 가능한 작업 속성을 한 줄에 하나씩 나열합니다.
-a attribute	작업에 대한 특정 속성을 표시합니다. 자세한 내용은 210페이지의 “swjob 작업 및 예제”를 참조하십시오.
-C session_file	명령을 실행하고 현재 옵션과 피연산자 값을 다른 세션에 다시 사용할 수 있도록 session_file에 저장합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.
-f jobid_file	명령줄에 지정하는 작업 대신(또는 추가적으로) 별도의 파일에서 작업 ID 목록을 읽습니다. (이 파일을 소프트웨어 파일과 똑같이 사용합니다. 자세한 내용은 60페이지의 “소프트웨어 파일”을 참조하십시오.)
-S session_file	session_file에 정의된 대로 이전 설치 세션에서 저장된 값을 기초로 명령을 실행합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.

`-t target_file`

명령줄에 지정하는 작업 대신(또는 추가적으로) 별도의 파일에서 대상 선택 사항 목록을 읽습니다. 자세한 내용은 61페이지의 “대상 파일”을 참조하십시오.

`-x option=value`

`option` 명령을 `value`로 설정하고 기본값 또는 옵션 파일의 값을 대체합니다. 자세한 내용은 61페이지의 “명령 옵션 사용”을 참조하십시오.

`-X option_file`

`option_file`에서 세션 옵션 및 동작을 읽습니다. 자세한 내용은 61페이지의 “명령 옵션 사용”을 참조하십시오.

`jobid`

**SD-UX** 작업에 대한 하나 이상의 식별 번호입니다. 대화식으로 작업을 설정하거나 모니터링할 때 작업 브라우저에서 작업 ID 번호를 읽을 수 있습니다.

`target_selections`

명령의 대상입니다. 자세한 내용은 60페이지의 “대상 선택 사항”을 참조하십시오.

## 명령 옵션 변경

명령을 호출할 때 명령줄 옵션을 추가로 지정(-x 옵션 사용)하거나 파일에서 미리 정의된 값을 읽어서 이 명령의 동작을 변경할 수 있습니다. 다음 표에서는 `swjob`에 적용되는 기본값과 옵션을 보여줍니다.

표 7-3

### `swjob` 명령 옵션 및 기본값

- |   |  |
|---|--|
| <ul style="list-style-type: none"><li>• <code>admin_directory=/var/adm/sw</code></li><li>• <code>agent_timeout_minutes=10000</code></li><li>• <code>log_msgid=0</code></li><li>• <code>one_liner={jobid operation state progress results title}</code></li><li>• <code>rpc_binding_info=ncacn_ip_tcp:[2121]ncadg_ip_udp:[2121]</code></li></ul> | <ul style="list-style-type: none"><li>• <code>loglevel=1</code></li><li>• <code>match_target=false</code></li><li>• <code>rpc_timeout=5</code></li><li>• <code>run_as_superuser=true</code></li><li>• <code>targets=</code></li><li>• <code>verbose=1</code></li></ul> |
|---|--|

## 추가 정보

옵션 설정 및 각 옵션의 전체 목록과 설명에 대한 자세한 내용은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.



## swjob 속성

각 작업에는 각각의 속성 세트가 있습니다. 이 속성에는 작업 제목, 예약된 실행 날짜 및 결과가 포함됩니다. -a 옵션은 표시할 특정 속성을 선택합니다. -a 옵션을 여러 번 지정하여 여러 속성을 표시할 수 있습니다.

표 7-4

### 일반적인 작업 속성

jobid	SD-UX에 의해 지정된 작업 식별 번호
operation	작업 유형(설치, 복사, 제거, 확인 등)
state	현재 작업 상태(완료됨, 예약됨 또는 활성)
progress	완료된 대상 수
results	성공, 경고 또는 오류를 나타내는 완료 상태
title	swconfig, swcopy, swinstall, swremove 또는 swverify 명령에서 job_title 옵션을 사용하여 작업에 대해 지정한 제목(211페이지의 “명령 옵션을 사용하여 작업 관리 및 조정” 참조)
schedule	작업 실행이 예약된 날짜
lastupdate	이 작업에 대한 정보가 마지막으로 업데이트된 날짜

## swjob 작업 및 예제

사용 가능한 작업을 간단하게 나열하려면 다음과 같이 입력합니다.

```
swjob
```

로컬 시스템에서 작업 속성을 표시하려면 다음과 같이 입력합니다.

```
swjob -v
```

원격 시스템 `swbash3`에서 작업 속성을 표시하려면 다음과 같이 입력합니다.

```
swjob -v @ swbash3:/var/spool/sw
```

`-a log` 옵션을 사용하여 작업에 대한 로그 파일을 표시할 수 있습니다. 작업 로그 파일은 작업 세부 사항과 대상 조치를 요약합니다. 예를 들어 원격 시스템 `swbash3`의 `swbash3-0008` 작업에 대한 저장소 로그 파일을 표시하려면 다음과 같이 입력합니다.

```
swjob -a log swbash3-0008 @ swbash3:/var/spool/sw
```

---

### 참고

---

`-a log` 옵션을 사용하는 경우에는 동일한 명령에 `-a` 옵션을 지정할 수 없습니다.

모든 이전 작업의 작업 정보를 제거하려면 다음과 같이 입력합니다.

```
swjob -u \*
```

---

## 명령 옵션을 사용하여 작업 관리 및 조정

SD-UX 명령 옵션을 사용하면 특히 대량의 작업을 실행할 때 사용자 환경에 가장 적절한 작업 동작을 관리하고 조정할 수 있습니다.

명령 옵션 설정에 대한 자세한 내용은 61페이지의 “명령 옵션 사용”을 참조하십시오.

### 명령줄에서 작업 예약

-Q *date* 옵션을 사용하여 작업 브라우저를 시작하지 않고 작업을 예약할 수 있습니다. 이 옵션은 `swinstall`, `swcopy`, `swconfig`, `swremove` 및 `swverify`에 사용할 수 있습니다.

*date*의 형식은 다음과 같습니다.

MM/DD[/YYYY] [, HH:MM] [AM|PM]

예를 들어 C와 Pascal 제품을 2001년 6월 23일 오전 10시 14분에 설치하려면 다음과 같이 입력합니다.

```
swinstall -Q 06/23/2001,10:15AM -s sw_server cc pascal
```

### 작업 제목 추가

**목적:** 작업을 식별하는 데 도움이 됩니다.

대량의 작업을 실행할 때 특정 작업을 식별하는 데 도움이 되는 정보를 추가하기를 원할 수 있습니다. 작업 브라우저에서 또는 `swconfig`, `swcopy`, `swinstall`, `swremove` 및 `swverify`를 `job_title` 명령 옵션과 함께 호출하여 이를 실행할 수 있습니다. 이를 통해 `swjob` 또는 작업 브라우저를 호출할 때 ID 및 다른 작업 속성과 함께 표시될 ASCII 문자열을 추가할 수 있습니다.

예를 들어 `sw_server`에서 02-HLLs라는 작업 제목으로 세 개의 호스트에 C와 Pascal 제품을 설치하려면 다음과 같이 입력합니다.

```
swinstall -s sw_server -x job_title=02-HLLs cc pascal \  
@ hostA hostB hostC
```

## 작업 정보 제거

**목적:** 대량의 작업을 실행할 때 성능을 높이고 사용 가능한 디스크 공간을 늘립니다.

SD-UX는 각 작업에 대해 작은 양의 정보(작업 상태, 컨트롤러 또는 에이전트 로그 파일 등)만을 저장합니다. 이 정보를 작업 브라우저나 **swjob**을 사용하여 표시할 수 있습니다. 많은 수의 작업 정보 파일을 유지하면 성능이 저하되고 작업 브라우저의 유용성이 떨어집니다.

`swconfig`, `swcopy`, `swinstall`, `swremove` 또는 `swverify`에서 `autoremove_job` 명령 옵션을 `true`로 설정하여 실행하면 **SD-UX**가 작업 정보를 저장하지 않습니다. (그러나 이 방법의 단점은 작업 정보를 보려면 로그 파일을 열어 봐야 한다는 것입니다.)

```
swinstall -s sw_server -x autoremove_job=true \  
MySoftware @ hostA hostB hostC
```

또한 `-u` 옵션을 사용하여 오래된 작업 정보를 제거할 수 있습니다.

```
swjob -u \*
```

작업 브라우저에서 개별 작업을 제거할 수도 있습니다(205페이지의 “작업 제거” 참조).

## 8                    안정성 및 성능

이 장에서는 **Software Distributor**의 안정성 기능과 성능 간의 상호 관계를 설명합니다. 이 기능들이 서로 어떻게 작용하는지 이해하면 소프트웨어 배포 시스템의 전반적인 안정성을 향상시키는 데 도움이 될 것입니다.

표 8-1

### 이 장의 항목

설명하는 항목
214페이지의 “개요”
216페이지의 “그룹 및 소스 옵션”
217페이지의 “많은 수의 대상”
218페이지의 “시간 제한 옵션”
219페이지의 “재시도 RPC 및 재시도 간격”
220페이지의 “재시도 명령”
221페이지의 “데이터베이스 검사점 지정”
222페이지의 “압축”
224페이지의 “단계”
227페이지의 “복구(설치에 한함)”
229페이지의 “별도의 구성으로 설치”
230페이지의 “여러 버전”

## 개요

SD-UX의 설치와 복사 처리량은 다음 요소에 따라 달라집니다.

- 네트워크 속도
- 전송되는 제품의 크기(즉, 바이트 수)
- 전송되는 파일의 수
- 소스 시스템 한 대가 동시에 처리하는 대상 시스템의 수

SD-UX는 분산 설치 및 복사의 속도와 성공 비율을 증가시키기 위해 사용할 수 있는 여러 기능을 제공합니다. 시스템 기본값 파일, 별도의 옵션 파일, 명령줄 또는 GUI에서 설정된 옵션으로 다양한 기능을 제어할 수 있습니다. 이를 통해 사용자의 고유한 네트워크 및 소프트웨어 요구에 가장 잘 맞도록 이러한 기능과 옵션을 설정할 수 있습니다.

이 옵션과 기능은 다음과 같은 범주로 나눌 수 있습니다.

- **그룹 및 소스 옵션:** SD-UX는 일반적으로 사용되는 대상 및 소프트웨어 그룹을 중복하여 지정할 필요가 없도록 합니다. 또한 소스 옵션을 사용하여 기본 저장소를 지정함으로써 대화 상자의 수를 줄일 수 있습니다.
- **많은 수의 대상:** 동시에 운영되는 대상의 수를 제한하는 옵션입니다.
- **시간 제한 옵션:** 파일 전송 시 포기하기 전까지 낮은 수준의 통신을 계속 시도하는 시간을 제어하는 옵션입니다.
- **재시도 RPC 및 재시도 간격:** 컨트롤러나 대상이 끊어진 연결을 다시 설정하려고 시도할 때 재시도 간격을 제어하는 옵션입니다.
- **재시도 명령:** 실패한 작업의 재시도를 용이하게 하는 옵션입니다. 검사점 기능과 연계하여 사용할 수도 있고 작업을 처음부터 시작할 수도 있습니다.
- **데이터베이스 검사점 지정:** SD-UX 명령은 SD-UX 저장소 카탈로그(IPD)에 데이터베이스 트랜잭션을 기록함으로써 파일 세트 수준에서 자동 검사점 지정을 수행합니다. 또한 파일 수준에서의 검사점 지정은 파일과 함께 저장된 속성을 통해 지원됩니다.

- **압축:** SD-UX는 압축 기능을 지원하여 전송되는 데이터 양을 줄입니다.
- **단계(Staging):** SD-UX는 단계 소프트웨어를 지원하여 설치 대상이나, 사전에 대체 소스를 사용하도록 구성된 대상에 보다 쉽게 액세스할 수 있는 소스에서 저장소를 중계합니다.
- **복구:** SD-UX는 설치가 실패했을 때 시스템을 이전과 같은 상태로 되돌리고 복구하는 자동 프로시저를 지원합니다. 또한 여러 버전을 사용하는 수동 방법도 있습니다.
- **여러 버전:** SD-UX는 해당 소프트웨어에서 지원되는 경우에 한해 시스템에 동시에 동일한 소프트웨어의 여러 버전을 설치할 수 있습니다. 그런 다음 이전 버전은 구성을 해제하고 새로운 버전을 활성 버전으로 구성합니다. 문제가 발생하는 경우 새로운 버전의 구성을 해제하고 이전 버전을 다시 구성함으로써 이전 버전을 활성 버전으로 복원할 수 있습니다.

이러한 내용은 다음에서 설명됩니다.

## 그룹 및 소스 옵션

그룹 및 소스 옵션을 통해 일부 명령의 성능이 개선될 수 있습니다.

대상 및 소프트웨어 선택 사항이 그룹 파일로 저장되어 재사용됩니다. 이를 통해 자주 사용되는 선택 사항을 다시 지정해야 할 필요가 줄어들고 `swinstall`, `swcopy` 또는 `swremove` 명령을 수행하는 데 필요한 시간이 감소합니다. 자세한 내용은 51페이지의 “Add/Save Software Group” 및 170페이지의 “소프트웨어 및 대상 목록”을 참조하십시오.

먼저 소스 저장소를 지정하면 `swcopy` 또는 `swinstall` 명령을 예약하는 데 필요한 화면 수를 줄일 수 있습니다. `source=` 기본 옵션을 기본 소스 저장소에 설정하면 SD-UX는 즉시 사용 가능한 제품을 나열할 수 있습니다. 따라서 기본값을 변경하는 경우를 제외하고는 **Source Selection** 대화 상자가 표시될 필요가 없습니다.



---

## 많은 수의 대상

`max_targets` 옵션은 `swinstall`과 `swcopy` 작업에 적용됩니다. 이 옵션으로 동시에 운영되는 대상 수를 정의된 값으로 제한함으로써 단일 작업으로 수백 개의 대상을 관리할 수 있습니다. 각 대상이 설치나 복사 작업을 완료하면 다른 대상이 선택되어 시작되고 이런 방식으로 모든 대상에 대한 작업이 완료됩니다. 이를 통해 활성화 상태의 작업의 수가 유지되거나 사용자가 정의한 한계 아래로 떨어집니다.

이 옵션을 지정한 결과, 한 번의 작업으로 수백 개의 대상을 안정적으로 관리할 수 있습니다. 또한 이 옵션으로 작업을 예약할 때 소요되는 시간을 대폭 줄일 수도 있습니다.

사용자의 서버와 네트워크 성능이 이 옵션에 대한 최적 설정값을 결정합니다. 기본 설정은 25입니다.

---

## 시간 제한 옵션

시간 제한 옵션은 파일 전송 시 포기하기 전까지 낮은 수준의 통신을 계속 시도하는 시간을 제어합니다.

한 가지 제어 방법은 포기하기 전까지 단일 **RPC** 호출이 기다리는 시간의 양을 조정하는 것입니다. 이 시간 제한은 `rpc_timeout` 옵션으로 설정됩니다. 허용되는 값은 **0**에서 **9**까지입니다. 기본값은 **5**이고 이는 **UDP** 프로토콜의 경우 약 **30초**에 해당됩니다. 각 값은 바로 앞의 값에 두 배에 해당되는 시간입니다(즉, **4** 값은 **15초**).

또 다른 방법은 `rpc_timeout`이 파일 세트 설치 중에 연결 소스가 끊어진 것을 감지한 다음 대상 에이전트가 소스 에이전트로 재연결을 시도하는 횟수를 제어하는 것입니다. 재시도 횟수는 `retry_rpc` 옵션으로 제어합니다. 값의 범위는 **0**에서 **9**까지입니다. **0** 값은 재시도를 하지 않는 것으로 연결이 끊어지면 명령은 실패합니다. 기본값은 **1**입니다. 안정성이 떨어지는 네트워크의 경우 **5** 값이 권장됩니다.

---

### 참고

`retry_rpc` 값을 **0**보다 큰 값으로 설정하면 `reinstall_files` 옵션을 `false`(기본 값)로 설정해야 파일 세트를 재시도할 때 동일한 파일을 다시 복사하지 않게 됩니다.

시간 제한을 대기하는 데 소요되는 최대 시간은 `rpc_timeout`, `retry_rpc` 및 `retry_rpc_interval`의 조합에 영향을 받습니다.

시간 제한 옵션에 대한 문제 해결 정보는 [450페이지](#)의 “연결 시간 제한 및 기타 WAN 문제점”을 참조하십시오.

---

## 재시도 RPC 및 재시도 간격

swinstall 또는 swcopy 작업 시 `retry_rpc_interval`은 최초 실패 후 대상이나 소스 에이전트로의 연결을 반복 시도하기 위한 간격 일정을 제어합니다. 이 옵션은 대상이나 소스로 다시 연결되는 횟수를 제어하는 `retry_rpc`와 함께 사용됩니다.

`retry_rpc`의 기본값은 1이고 `retry_rpc_interval`의 기본값은 {0}입니다.

`retry_rpc_interval` 알고리즘에 권장되는 값은 {1 2 4 8 15}이고 `retry_rpc`는 5로 설정됩니다. 어떤 이유로든 에이전트 세션을 시작하는 데 실패하면 컨트롤러 및/또는 대상은 다음과 같은 방법으로 재연결을 시도합니다(즉 두 옵션이 위와 같이 설정된 경우).

- 1분 후에 첫 번째 시도
- 2분 후에 두 번째 시도
- 4분 후에 세 번째 시도
- 8분 후에 네 번째 시도
- 마지막으로 15분 후에 최종 시도

`retry_rpc_interval`의 값 수가 `retry_rpc`보다 작으면 실제 재시도 횟수가 `retry_rpc`와 같아질 때까지 마지막으로 제공된 값을 반복합니다.

---

### 참고

`retry_rpc` 값이 0(영)이면 재시도를 실행하지 않습니다. WAN 환경에서는 최초 값으로 5가 권장됩니다.

---

## 재시도 명령

SD-UX는 재시도 작업이 실패한 경우 이를 지원하는 옵션을 제공합니다. 검사점 기능과 연계하여 사용할 수도 있고 작업을 처음부터 시작할 수도 있습니다.

각 명령이 실행될 때 모든 대상, 소프트웨어 및 옵션 선택 사항이 `command.last` 세션 파일에 자동으로 저장됩니다. GUI를 사용하여 다른 파일에 세션 정보를 저장할 수도 있습니다. 세션 파일은 `$HOME/.sw/sessions` 디렉토리에 저장됩니다. 전체 명령은 GUI에서 세션을 다시 호출하거나 해당 작업을 다시 실행시킴으로써 재시도될 수 있습니다.

```
command -S command.last
```

작업을 재시도할 때 최신 파일 세트(동일한 제품 및 파일 세트 버전으로 사용 가능하거나 설치되어 있으며 *transient* 또는 *corrupt* 상태가 아님)에 대해서는 재설치를 시도하지 않고(기본 동작) 모든 다른 파일 세트에 대해서는 재설치를 시도합니다. 이 동작은 옵션 설정 또는 CLI에서 `reinstall=true`로 설정하여 모든 파일의 처음부터 재시도를 시작하도록 재정의할 수 있습니다.

`reinstall_files` 옵션을 `true`로 설정하면 해당 파일 세트의 모든 파일이 다시 전송됩니다. 그러나 설치 중인 파일 세트에 대해 `preinstall` 및 `postinstall` 스크립트가 정상적으로 실행되더라도 `reinstall_files` 옵션을 `false`(기본값)로 두면 최신 파일의 파일 전송은 수행되지 않습니다.

`reinstall_files` 옵션이 `false`이면 사용자는 파일 세트가 이미 설치되어 있는지 또는 사용 가능한지를 판별하기 위해 어떤 속성을 점검할 것인지를 제어할 수도 있습니다. `reinstall_files_use_cksum` 옵션을 `true`로 설정하면 `size`, `mtime` 및 `cksum` 속성을 점검합니다.

`reinstall_files_use_cksum` 옵션이 `false`이면 `size`와 `mtime`만 점검합니다. `cksum` 속성을 점검하는 것은 시간이 많이 소요되지만 보다 안정적입니다. `size`와 `mtime` 점검은 매우 빠르게 수행됩니다.

`loglevel` 옵션을 2로 설정하면 실제 어떤 파일이 설치 또는 복사되고 어떤 파일이 이미 최신 파일이기 때문에 생략되는지를 알 수 있습니다.

---

## 데이터베이스 검사점 지정

이 도구는 자동 검사점 지정 기능을 제공하고 트랜잭션을 파일 세트 수준에서 **SD-UX** 저장소 카탈로그나 **IPD(Installed Products Database)**에 기록합니다. 또한 파일 수준에서 검사점을 지정할 때 파일과 함께 저장된 속성을 통해 지원됩니다.

`swinstall` 또는 `swcopy` 작업 시 로드 중인 현재 제품의 모든 파일 세트는 저장소 카탈로그 또는 **IPD**에 **transient** 상태로 기록됩니다. 제품의 파일 세트가 모두 복사나 설치를 완료하면 상태는 **available** 또는 **installed**로 바뀌고 다음 제품에 대한 작업이 시작됩니다. 이 때 작업을 재시도하면 이미 설치되어 있는 파일 세트는 다시 복사하거나 다시 설치하지 않습니다(220페이지의 “재시도 명령” 참조).

---

### 참고

이를 위해서는 제품이나 파일 세트 중 어느 하나에 버전 정보가 정의되어 있어야 합니다.

파일 세트의 현재 상태 및 버전은 다음 명령으로 표시할 수 있습니다.

```
swlist [-d] -l fileset -a revision -a state
```

제품의 파일 세트를 설치하는 동안 오류가 발생하여 설치가 실패하면(즉 소스와의 연결이 끊어짐) 제품의 파일 세트 상태가 모두 **transient**에서 **corrupt**로 변경됩니다. (제품 수준의 `postinstall` 스크립트가 아직 실행되지 않았으므로 파일 세트는 모두 손상된 것으로 간주됩니다. 실제로 파일 세트가 올바르게 설치되었을 수도 있습니다.)

설치되는 파일 세트(정상적으로 설치된 것이든 *corrupt* 상태이든)와는 별도로 파일의 크기, `mtime` 및 체크섬 속성을 통해 특정 파일이 정상적으로 설치되었는지 여부를 매우 확실하게 판별할 수 있습니다. 이러한 파일 속성을 통해 파일 수준의 검사점 지정을 대략적으로 수행할 수 있습니다(앞에서 설명).

## 압축

`swinstall`과 `swcopy` 명령은 저장소에서 대상 시스템으로 네트워크를 통해 대량의 데이터를 전송할 수 있습니다. `SD-UX compress_files` 옵션을 통해 먼저 전송할 파일을 압축함으로써 성능을 향상시킬 수 있습니다. 이는 네트워크 사용량을 대략 50% 정도 감소시킬 수 있으며 정확한 압축률은 파일의 형식에 따라 다릅니다. 바이너리 파일의 압축률은 50% 이하이고, 텍스트 파일은 일반적으로 그보다 더 많이 압축됩니다.

네트워크 대역폭이 전체적인 처리량에 미치지 못한다고 확신하는 경우에만 이 옵션을 `true`로 설정합니다. 이 옵션이 도움이 될 수 있을지 확실하지 않다면 옵션 값을 바꾸기 전에 `swinstall` 또는 `swcopy` 작업의 처리량을 먼저 비교해 보십시오(압축했을 때와 압축하지 않았을 때를 비교).

`swcopy`를 사용하여 파일을 압축하고 압축된 파일 그대로 대상 저장소에 보관하거나 네트워크 전송을 하기 전에 압축한 다음 전송 후에 압축을 풀 수도 있습니다.

저장소를 미리 압축하면 여러 대상으로 설치하거나 복사할 때 유용합니다. 소스 저장소가 아직 압축되어 있지 않으면 각 파일이 각 대상에 대해 다시 압축됩니다.

`uncompress_files`를 `true`로 설정하면 `swcopy`로 복사한 다음 저장소를 압축하지 않은 상태로 둘 수 있습니다. `swinstall`의 경우 `compress_files` 옵션은 네트워크 전송 전에 압축되지 않은 모든 파일을 압축합니다. 파일은 항상 대상 파일 시스템에 설치되기 전에 압축이 해제됩니다.

## INDEX 및 INFO 압축

네트워크 처리량을 감소시키는 또 다른 방법은 `INDEX` 및 `INFO` 파일을 소스 저장소에서 대상 시스템으로 압축시켜 보내는 것입니다. 기본 파일 (`/var/adm/sw/defaults`)에서 `compress_index` 옵션을 `true`로 설정하면 `INDEX` 또는 `INFO` 압축을 활성화할 수 있습니다.

`SD-UX` 컨트롤러와 대상 에이전트는 소스 에이전트로부터 압축된 `INDEX` 파일을 요청합니다. 소스 에이전트가 읽기 전용이거나 이전 버전의 `SD-UX`인 경우 에이전트는 요청을 따를 수 없으므로 클라이언트는 일반 `INDEX`를 요청합니다. 그렇지 않으면 소스 에이전트는 사전에 압축된 `INDEX`와 `INFO`를 보내거나 전송 중에 압축합니다.

그러면 대상 에이전트는 대상, 저장소 또는 루트에 영구적인 압축 INDEX를 작성합니다. 이렇게 하면 압축 INDEX 또는 INFO에 대한 다음 번 요청이 있을 때 전송 중에 압축할 필요가 없습니다.

## 단계

소프트웨어를 여러 대상에 설치하는 일반적인 방법은 하나의 소스 저장소와 해당 소프트웨어를 수신할 각 대상을 지정하는 것입니다. 그러나 일부 소프트웨어 분산 환경에서는 지리적으로 멀리 떨어져있는 많은 수의 대상 시스템상에서 소프트웨어를 관리하는 경우가 있습니다. 이런 경우 하나 이상의 중간 소스 저장소나 중간 영역을 사용할 필요가 있습니다. 표준 모델의 이러한 변형을 **단계적 설치**라고 합니다.

단계적 설치를 사용하는 데에는 두 가지 이유가 있습니다.

1. 네트워크상에서 속도가 느리고 비용이 많이 드는 세그먼트를 통과하여 전송되는 데이터의 양을 최소화합니다.
2. 네트워크에서 불안정한 세그먼트의 위험을 감소시킴으로써 보다 손쉽게 정상적으로 설치를 완료할 수 있습니다.

사용자 환경에서 대상 시스템이 별도의 LAN으로 구성되어 있고 이것이 다시 처리량이 낮고 불안정한 WAN에 연결되어 있는 경우, 각 대상 그룹에 로컬로 연결되어 있는 중간 저장소에 대한 단계(staging) 소프트웨어와 이러한 중간 저장소를 사용하여 설치 작업을 수행하면 WAN 세그먼트를 지나가는 데이터 양을 감소시킬 수 있습니다.

이렇게 함으로써 WAN에서 발생하는 문제로 인해 설치 작업이 중단될 가능성도 줄어듭니다.

단계적 설치를 수행하기 전에 먼저 중간 저장소의 위치를 결정해야 합니다. 이에 대한 두 가지 접근 방법이 있습니다.

1. 대상이 그룹화되어 있는 경우 각 그룹에 있는 한 시스템에 중간 저장소를 두고 다른 대상이 이 시스템을 대체 소스로 사용하도록 구성할 수 있습니다. 이 방법을 사용하려면 그룹의 각 대상이 지정된 중간 저장소를 사용하도록 구성해야 합니다.
2. 반드시 설치를 성공시켜야 하는 경우에는 대상 하나에 중간 저장소를 하나씩 위치시킬 수 있습니다. 이 방법의 장점은 각 대상에 반드시 대체 소스를 구성할 필요가 없다는 점입니다. 하지만 이 방법을 사용하려면 각 대상 시스템에 중간 저장소를 수용할 수 있는 충분한 디스크 공간이 있어야 합니다.



단계적 설치를 수행하려면 다음과 같이 합니다.

1. 먼저 중간 저장소의 위치를 결정한 다음 `swcopy` 명령을 사용하여 마스터 저장소에서 중간 저장소로 소프트웨어를 복사합니다. 이 단계는 일반적인 여러 대상 복사 작업과 다를 바가 없습니다.

**swcopy -s master -t depot\_list NewApp**

이 예에서 `NewApp` 제품을 포함하는 마스터 소스 저장소의 기본 저장소 위치는 `/var/spool/sw`이고 `depot_list`라는 파일에 중간 저장소 목록이 포함되어 있습니다.

`depot_list`를 통해 각 대상 그룹에 대해 구성된 지정된 중간 저장소를 확인하거나 각 대상에 위치한 중간 저장소를 확인할 수 있습니다.

2. 그런 다음 `swinstall` 명령을 `use_alternate_source=true` 옵션과 함께 사용하여 실제 설치를 수행합니다. `use_alternate_source` 옵션은 CLI에서 지정할 수도 있고(즉, `-x use_alternate_source=true`) GUI의 Options Editor 창을 통해 지정할 수도 있습니다. 기본값은 `false`입니다.

**swinstall -s master -x use\_alternate\_source=true \**  
**-t targ\_list NewApp**

`use_alternate_source=true` 옵션은 설치 시 각 대상이 자체 구성된 소스를 사용하도록 지시합니다. `swinstall` CLI에 지정된 소스는 소프트웨어 선택 사항을 검증하기 위한 목적으로 컨트롤러에 의해서만 사용됩니다. `targ_list` 파일에는 대상 목록이 포함되어 있습니다.

`use_alternate_source`가 `true`이면 각 대상 에이전트가 자체 기본 파일에서 해당 `swagent.alternate_source` 옵션을 찾습니다. 에이전트가 `swagent.alternate_source`에 의해 지정된 저장소에 연결하려고 할 때 `swagent.rpc_binding_info` 옵션으로 제공되는 프로토콜 순서와 엔드포인트가 사용됩니다. 대체 소스는 `host:/path, /path` 또는 `host` 구문을 사용하여 지정합니다.

- 대상의 `swagent.alternate_source` 옵션에 `host:/depot_path`가 지정되어 있으면 에이전트가 이 소스에서 소프트웨어를 가져옵니다. `host`만 지정되어 있으면 대상 에이전트는 컨트롤러가 사용한 것과 동일한 저장소 경로를 사용합니다.

- 대상에 대체 소스가 없는 경우 에이전트는 컨트롤러가 사용하는 저장소 경로를 사용하지만 자체 파일 시스템에 이 경로를 적용합니다. 이렇게 하면 중간 저장소를 마스터 저장소와 동일한 파일 시스템 위치에 있는 대상 시스템 각각에 배치함으로써(위 방법 2) 대상을 전혀 구성하지 않고도 단계적 설치를 수행할 수 있습니다.

단계적 설치에서 `swcopy`와 `swinstall` 단계는 별개이므로 **SD-UX**는 마스터 저장소와 중간 저장소의 일관성을 강제하지는 않습니다. 사용자는 중간 저장소에 있는 소프트웨어가 마스터 저장소에 있는 소프트웨어와 일치하도록 해야 합니다.

`swinstall` 단계를 수행할 때 마스터와 중간 저장소가 일치하지 않아 마스터 저장소에 있는 소프트웨어를 하나 이상의 중간 저장소에서 찾을 수 없는 경우 오류가 발생할 것입니다.

---

## 복구(설치에 한함)

---

### 참고

다음 내용은 `unpreinstall` 및 `unpostinstall` 해제 스크립트를 사용하는 고객 작성 소프트웨어에만 적용됩니다. HP에서 제공하는 소프트웨어에 이들 스크립트는 포함되지 않습니다.

`autorecover_product` 옵션을 `true`로 설정한 경우 **SD-UX**는 설치 실패 시 복구할 수 있는 자동 절차를 제공하여 시스템을 이전과 동일한 상태로 되돌리려고 시도합니다. 또한 수동 방법도 사용할 수 있습니다(230페이지의 “여러 버전” 참조).

롤백은 설치 작업에 지정된 모든 대상 시스템이 아닌 제품 설치에 실패한 시스템에 한해 적용됩니다.

자동 복구 기능은 해당 시점까지 설치된 모든 파일을 제거하므로 앞서 설명한 검사점 및 재시도 기능과는 반대 개념입니다. 복구는 대체하는 각 파일의 사본을 저장하고 그 다음 제품 설치가 정상적으로 완료되면 해당 파일을 제거합니다. 설치에 실패하면 저장된 파일이 복원됩니다.

일단 제품이 정상적으로 업데이트되면 다시 설치하는 방법을 제외하고는 복원할 수 없습니다. 또한 나중 제품을 설치하는 데 실패하면 이전 제품을 복구할 수 없습니다. 여러 제품을 복구 가능하게 하려면 여러 버전을 설치해야 합니다.

`preinstall`과 `postinstall` 스크립트가 실행한 단계를 취소하려면 `unpreinstall`과 `unpostinstall` 스크립트가 필요합니다. 각 제품에 대한 일반적인 작동 순서는 다음과 같습니다.

1. 제품 `preinstall` 스크립트를 실행합니다.
2. 각 파일 세트에 대해
  - a. `preinstall` 스크립트를 실행합니다.
  - b. 파일을 설치합니다.
  - c. `postinstall` 스크립트를 실행합니다.
3. 제품 `postinstall` 스크립트를 실행합니다.

안정성 및 성능  
복구(설치에 한함)

이들 단계 중 어느 하나가 실패하면(예: 소스 유실 또는 스크립트 오류) 취소 스크립트가 실행되고 실패 시점부터 역순으로 파일이 복원됩니다.

---

**참고**

설명된 기능과 특징을 사용하여 작성된 패치로 저장된 파일을 관리할 수 있습니다. 이 경우 패치는 제거(롤백)할 수 있고 커밋(저장된 파일을 제거함으로써)할 수도 있습니다. 패치에 대한 자세한 내용은 5장, 165페이지의 “HP-UX 패치 및 패치 관리”를 참조하십시오.

---

**참고**

HP-UX OS를 업데이트하는 동안에는 `autorecover_product=true` 사용이 지원되지 않습니다.

---

## 별도의 구성으로 설치

---

### 참고

OS 소프트웨어와 패치의 구성을 뒤로 미루면 시스템이 사용할 수 없는 상태가 되므로 HP에서 제공하는 소프트웨어에 대해서는 이 방법을 사용하지 마십시오.

설치 후 자동으로 수행되는 스크립트를 사용자의 소프트웨어에 포함시켜 작성하는 경우 구성을 별도로 수행하면 전체적인 설치 프로세스의 안정성이 향상될 수 있습니다.

구성하지 않고 설치하려면 `swinstall`에 대해 `defer_configure` 옵션을 `true`로 설정합니다. 그런 다음 모든 설치가 성공적으로 완료된 후에 `swconfig` 명령을 사용하여 모든 대상에 대해 한 번에 구성 스크립트를 실행할 수 있습니다.

---

## 여러 버전

SD-UX는 해당 소프트웨어에서 지원되는 경우에 한해 시스템에 동시에 동일한 소프트웨어의 여러 버전을 설치할 수 있습니다. 여러 버전을 설치함으로써 시스템이나 작업(모든 시스템) 수준의 복구를 수행할 수 있습니다.

두 번째 버전을 설치하려면 시스템상의 여러 버전을 확인하는 방법을 이해하는 것은 물론 신중한 계획이 필요합니다.

각 제품에는 제품 디렉토리 속성이 있습니다. 대상 시스템에서 설치된 위치는 기본적으로 제품 디렉토리와 동일합니다. 예를 들어 Foo 제품에는 /opt/foo 제품 디렉토리가 있을 수 있습니다. 설치된 소프트웨어의 위치를 표시하려면 다음과 같이 입력합니다.

```
swlist -l product -a location
```

또는

```
swlist -l product -a software_spec
```

일반적으로 많이 사용하는 방법은 제품의 두 번째 버전을 설치하는 것입니다. 이 소프트웨어를 설치할 때 위치를 새로 선택해야 합니다. Foo 제품의 경우 새 위치는 /opt/foo.v2가 될 수 있습니다. 새로운 위치를 지정한 후에(즉, GUI나 CLI에서 제품 태그 뒤에 l=/opt/foo.v2 추가) swinstall은 모든 파일의 제품 디렉토리 부분을 새로운 제품 위치로 바꾸어 줍니다.

---

### 참고

소프트웨어의 여러 버전을 설치하려면 swinstall의 allow\_multiple\_versions 옵션을 true로 설정해야 합니다. 다른 버전이 구성되어 있으면 swinstall이 새로운 버전을 구성하지 않습니다.

두 번째 버전을 설치한 후에 각 버전은 위치(Foo, l=/opt/foo 및 Foo, l=/opt/foo.v2), 버전(Foo, r=1.0 및 Foo, r=2.0) 또는 이 두 가지 모두로 확인할 수 있습니다. 모든 버전의 위치와 버전을 표시하려면 다음과 같이 입력합니다.

```
swlist -l fileset -a location -a revision
```

또한 다음을 입력하여 위치와 버전뿐 아니라 다른 **버전별** 속성(공급업체 및 아키텍처)을 포함하는 **정규화된 소프트웨어 사양**을 나열할 수 있습니다.

```
swlist -l fileset -a software_spec
```

새로운 버전이 모든 제품에 대해 또는 모든 호스트상에서 설치되었으면 이전 버전에 **swconfig -u** 명령을 사용하고 새 버전에 swconfig 명령을 사용하여 이전 버전의 구성을 해제하고 새 버전을 활성 버전으로 구성할 수 있습니다.

---

## 참고

---

기본적으로 한 번에 한 가지 소프트웨어 버전만 구성할 수 있습니다.

첫 번째 버전의 구성이 해제되어야 두 번째 버전을 구성할 수 있습니다. 229페이지의 “별도의 구성으로 설치”에서 설명한 대로 두 번째 버전을 설치하더라도 자동으로 새 버전을 구성하지는 않습니다.

수동으로 이전 버전을 구성 해제한 다음 새로운 버전을 구성해야 합니다. 소프트웨어에서 여러 구성 버전을 지원하는 경우(여러 버전 설치와 함께)

swconfig.allow\_multiple\_versions 옵션을 true로 설정할 수 있습니다.

문제가 발생하는 경우 새로운 버전을 구성 해제하고 이전 버전을 다시 구성하여(즉, 새로운 버전에 **swconfig -u** 명령을 사용하고 이전 버전에 swconfig를 사용) 이전 버전을 활성 버전으로 복원할 수 있습니다.

여러 버전을 지원하려면 소프트웨어 구조에 있어서 제품 디렉토리 아래에 모든 파일이 들어 있어야 하고 여러 버전 지원을 염두에 두고 스크립트를 작성해야 합니다. 간단한 예로 **configure** 스크립트는 /usr/bin/foo에서 \$SW\_LOCATION/bin/foo로의 심볼릭 링크를 추가하고 **unconfigure** 스크립트는 해당 링크를 제거할 수 있습니다. 이 예에서 이 소프트웨어의 각 버전에 대한 구성 및 구성 해제가 매우 쉽게 이루어집니다.

---

## 참고

---

HP-UX를 새 버전으로 업데이트할 때는 allow\_multiple\_versions=true 명령 옵션과 l=<alternate location> 소프트웨어 사양은 지원되지 않습니다.

안정성 및 성능  
여러 버전



## 9 SD-UX 보안

SD-UX를 설치하는 동안 기본적인 보안 설정이 만들어집니다. 이 장에서는 기본적인 SD-UX 보안에 대해 설명하고, `swacl` 명령을 소개하며, 일반적인 작업의 예를 보여주고, SD-UX의 보안 관리 방법에 대해 자세히 설명합니다.

표 9-1

### 이 장의 항목

항목 및 참조 페이지
234페이지의 “개요”
236페이지의 “swacl 명령”
239페이지의 “기본 보안 작업”
250페이지의 “ACL이 사용자에게 일치되는 방법”
251페이지의 “ACL 항목”
263페이지의 “SD-UX 시스템에서의 보안”
265페이지의 “SD-UX 내부 권한 인증”
269페이지의 “RPC 권한”
273페이지의 “보안 사용 모델”
276페이지의 “명령별 권한 요구 사항”

## 개요

기존의 HP-UX 파일 액세스 보호 기능과 함께 SD-UX는 액세스 제어 목록(ACL)을 사용하여 소프트웨어 관리 기반이 되는 기본 객체를 보호합니다.

- 호스트
- 루트(호스트에 설치된 소프트웨어)
- 저장소
- 저장소 내 제품

ACL은 객체가 만들어질 때 그 객체와 연관되는 항목의 집합으로 구성됩니다.

## 기본 보안

기본적으로 다음과 같은 보안 체계를 가지고 있습니다.

- 로컬 슈퍼유저는 항상 모든 로컬 객체에 액세스할 수 있습니다.
- 읽기 액세스는 any\_other ACL을 통해 동일한 SD-UX 공유 비밀을 사용하는 네트워크상의 모든 사용자에게 제공됩니다.
- 루트, 저장소 또는 제품 객체를 만드는 사람은 누구든지 해당 객체에 대해 object\_owner로서 모든 액세스 권한을 가집니다.
- 원격 작업을 위해 시스템을 설정하는 경우(177페이지의 “원격 작업 설정”에서 설명한 절차 사용) root@central\_controller는 user:root@central\_controller ACL을 통해 모든 대상 객체에 대한 모든 액세스 권한을 가집니다.

root@central\_controller 자격으로 실행하는 경우에는 제시된 보안 설정이 모든 작업을 수행하는 데 적합해야 합니다.

기본 ACL을 작성하는 데에는 두 가지 템플릿이 사용됩니다.

- global\_soc\_template(호스트에 추가된 모든 새 저장소와 루트에 적용)
- global\_product\_template(저장소의 새 제품에 적용)

## 저장소 및 저장소 등록

Software Distributor는 일반적으로 중앙 저장소를 사용하여 소프트웨어를 배포합니다. 소프트웨어를 설치할 사용자들이 이 저장소에 액세스하는 것을 제어할 수 있습니다.

보안상 중요한 점은 로컬이 아닌 사용자가 액세스할 수 있도록 저장소를 **반드시** 등록해야 한다는 것입니다. 로컬 슈퍼유저나 해당 호스트에 대한 삽입 권한을 갖고 있는 사용자만 등록되지 않은 저장소로부터 설치할 수 있습니다.

자세한 내용은 153페이지의 “저장소 등록 및 등록 해제(swreg)” 및 136페이지의 “저장소 관리 명령 및 개념”을 참조하십시오.

## 대상 시스템 변경

SD-UX 컨트롤러에 관리 액세스 권한을 부여하고 다른 시스템과 사용자에게 대한 액세스는 제한하도록 각 시스템을 설정하려는 경우가 있습니다.

다음과 같은 경우에 대상 시스템에서 ACL을 수정해야 합니다.

- SD-UX 관리자의 로그인 이름을 변경하는 경우(기본값은 루트)
- SD-UX 관리자 또는 관리자 그룹에 대한 권한을 변경하는 경우

## swacl 명령

swacl 명령을 사용하여 ACL 항목과 권한을 보거나 변경할 수 있습니다.

### swacl 구문

```
swacl ] -l level [-D acl_entry|-F acl_file|-M acl_entry]
        [-f software_file][-t target_file]
        [-x option=value] [-X option_file]
        [software_selections] [@ target_selection]
```

### 옵션 및 피연산자

**-l level** 편집 수준. 수준은 host, depot, root, product, product\_template, global\_soc\_template 또는 global\_product\_template과 같이 리터럴로 지정합니다 (자세한 내용은 260페이지의 “ACL 템플릿” 참조).

### 참고

-D, -F 또는 -M 명령 옵션을 사용하여 ACL을 변경할 수 있습니다. 이 옵션들은 동시에 지정할 수 없으므로 명령 하나에 한 옵션만 지정할 수 있습니다. -D, -F 또는 -M 옵션을 지정하지 않으면 swacl이 지정된 ACL을 인쇄합니다.

**-D acl\_entry** 지정된 객체와 연관된 ACL에서 기존 항목을 삭제합니다. -D 옵션은 여러 개 입력할 수 있습니다.

**-F acl\_file** acl\_file에 포함된 ACL 정보를 객체에 할당합니다. 기존 항목이 모두 제거되고 파일의 항목으로 대체됩니다. -F 옵션은 하나만 입력할 수 있습니다.

**-M acl\_entry** 새로운 ACL 항목을 추가하거나 기존 항목의 권한을 변경합니다. -M 옵션은 여러 개 입력할 수 있습니다.

**-f software file**

CLI 대신 별도의 파일에서 소프트웨어 선택 항목 목록을 읽습니다(60페이지의 “소프트웨어 파일” 참조).

`-t target file`

CLI 대신 별도의 파일에서 대상 호스트 선택 항목 목록을 읽습니다 (61페이지의 “대상 파일” 참조).

`-x option=value` 명령줄 인터페이스(CLI)에서 기본값 또는 대체 옵션 파일(`-x option file`)의 값을 재정의하는 옵션을 변경할 수 있게 합니다. 자세한 내용은 237페이지의 “명령 옵션 변경”을 참조하십시오.

`-X option file` 지정된 `option file`의 옵션 값을 사용합니다. 자세한 내용은 61페이지의 “명령 옵션 사용”을 참조하십시오.

`software_selections`

swacl 작업에 대한 소프트웨어 객체입니다. 자세한 내용은 58페이지의 “소프트웨어 선택 사항”을 참조하십시오.

`target_selections`

명령의 대상입니다. 자세한 내용은 60페이지의 “대상 선택 사항”을 참조하십시오.

## 명령 옵션 변경

명령을 호출할 때 명령줄 옵션을 추가로 지정(`-x` 옵션 사용)하거나 파일에서 미리 정의된 값을 읽어서 이 명령의 작동 방식을 변경할 수 있습니다. 다음 표에서는 swacl에 적용되는 기본값과 옵션을 보여줍니다.

표 9-2

### swacl 명령 옵션 및 기본값

- `admin_directory=/var/adm/sw`
- `distribution_target_directory=/var/spool/sw`
- `installed_software_catalog=products`
- `level=`
- `log_msgid=0`
- `rpc_binding_info=ncacn_ip_tcp:[2121]  
ncadg_ip_udp:[2121]`
- `rpc_timeout=5`
- `run_as_superuser=true`
- `select_local=true`
- `targets=`
- `verbose=1`

## 추가 정보

각 기본값에 대한 자세한 설명은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.

## swacl 결과

swacl 명령의 일반적인 목록 출력은 다음과 같습니다.

```
# swacl      Installed Software Access Control List
#
# For host:   prewd:/
#
# Date:      Mon Nov 06 16:39:58 2001
#
# Object Ownership: User=root
#                  Group=sys
#                  Realm=prewd.fc.hp.com
# default_realm=prewd.fc.hp.com
object_owner:crwit
user:rml:crwit
user:root@newdist.fc.hp.com:crwit
group:swadm:crwit
any_other:-r--t
```

머리말 정보(#으로 표시된 줄)는 객체 이름과 소유자, 사용자의 **영역** 이름 또는 사용자의 시스템 이름을 제공합니다. 이 경우 다음 사항이 적용됩니다.

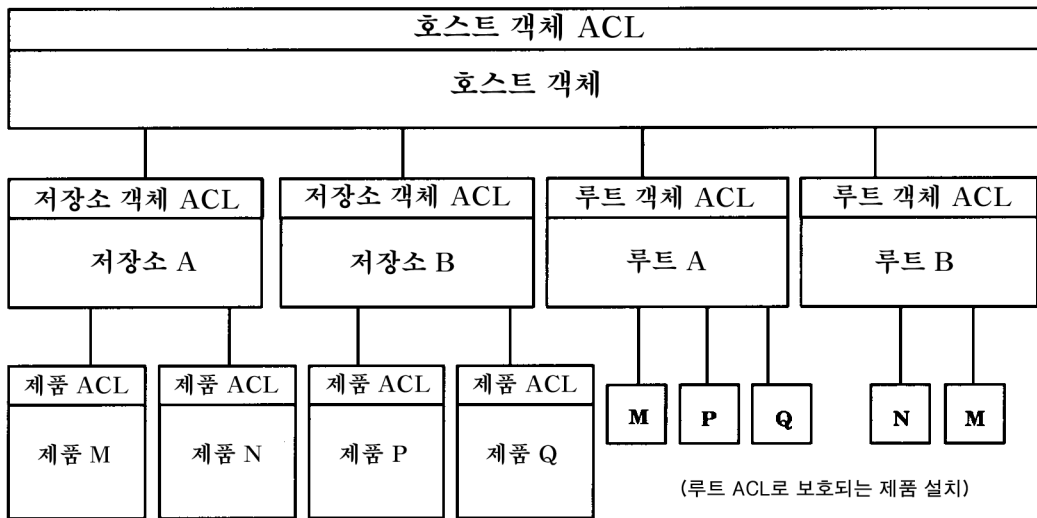
- 객체는 루트입니다(설치된 소프트웨어).
- 객체의 작성자(object\_owner)는 root@prewd입니다.
- object\_owner는 모든 권한을 가집니다.
- 로컬 사용자 rml과 원격 사용자 root@newdist.fc.hp.com은 모든 권한을 가집니다.
- 로컬 그룹 구성원 swadm은 모든 권한을 가집니다.
- 위 항목에 하나도 해당되지 않는 사람(any\_other)은 읽기 및 테스트 권한을 가집니다.

ACL 파일의 형식에 대한 자세한 내용은 251페이지의 “ACL 항목”을 참조하십시오.

## 기본 보안 작업

기존의 HP-UX 파일 액세스 보호 기능과 함께 모든 SD-UX 객체(호스트, 저장소, 루트 및 제품)는 ACL에 의해서도 보호를 받습니다.

그림 9-1 액세스 제어 목록



ACL은 권한 비트에 비해서 한층 강화된 선별 기능을 제공합니다. ACL은 개인이나 그룹 중 하나에만 액세스 권한을 지정하는 대신 여러 개인과 그룹에 서로 다른 액세스 권한을 지정함으로써 HP-UX 파일 시스템 권한 비트의 개념을 확장시킵니다.

예를 들어 원격 작업을 설정하는 경우, 원격 시스템상의 보안 ACL의 기본적인 부분을 변경하는 HP ServiceControl Manager 파일 세트를 설치해야 합니다. 변경되는 사항 중 하나가 소스 시스템에서 대상 시스템으로 세 개의 ACL을 복사하는 것입니다(177 페이지의 “원격 작업 설정” 참조).

복사되는 ACL은 소스 호스트를 보호하는 호스트 ACL, 제품에 대한 ACL을 작성하는 후속 작업에 사용되는 호스트의 템플릿 ACL(global\_product\_template) 및 저장소/루트 컨테이너(global\_soc\_template)입니다. 이 ACL들이 복사되면 소스 호스트의 사용자들에게 로컬로 소스 호스트상에서 가지고 있었던 것과 동일한 권한을 대

상 호스트에서 부여합니다. 또한 소스 호스트에서 슈퍼유저에 대한 항목이 추가되었습니다. 이렇게 하면 컨트롤러 시스템의 슈퍼유저가 ACL을 다시 구성하지 않고도 원격 시스템에 대해 소프트웨어 배포 작업을 수행할 수 있습니다.

보안을 변경해야 하는 경우 기본 설정을 이해하고 변경하려면 다음과 같은 작업을 수행합니다.

- 사용자 액세스 나열
- 사용자가 저장소의 제품을 관리
- 사용자가 루트를 관리
- 저장소에 대한 읽기 액세스 제한
- 대상 호스트 추가
- 저장소에 대한 액세스를 일시적으로 제한
- SD-UX 네트워크 달기
- ACL 편집

## 사용자 액세스 나열

다음 예에서는 사용자와 함께 저장소, 대상 호스트, 대상 루트 및 모든 제품에 대한 액세스 권한을 나열하는 방법을 보여줍니다.

- 새로 설치된 HP-UX 11i 시스템에서 기본 루트 ACL을 표시합니다.

```
swacl -l root

#
# swacl  Installed Software Access Control List
#
# For host:  swelter:/
#
# Date:   Wed Feb 28 14:58:02 2001
#
# Object Ownership:  User= root
#                   Group=sys
#                   Realm=swelter.fc.hp.com
#
# default_realm=swelter.fc.hp.com
object_owner:crwit
any_other:-r---
```



이 ACL은 파일 시스템 소유자가 루트 사용자이고 따라서 이 소유자가 모든 ACL 권한 (crwit)을 가지고 있음을 보여줍니다. 또한 다른 모든 사용자들은 swlist 명령을 사용하여 이 루트 파일 시스템에 대한 SD 정보를 읽을 수 있습니다.

- 기본 저장소(선택적으로 원격 호스트에 위치)에 대한 액세스 권한을 가진 사용자를 나열하려면 다음과 같이 입력합니다.

```
swacl -l depot @ newdist

#
# swacl      Depot Access Control List
#
# For depot:  newdist:/var/spool/sw
#
# Date:      Fri Nov 03 11:23:51 2001
#
# Object Ownership:  User= root
#                    Group=other
#                    Realm=newdist.fc.hp.com
#
# default_realm=newdist.fc.hp.com
object_owner:crwit
user:rmr:crwit
user:root:crwit
user:fred@hpfred.fc.hp.com:crwit
user:root@hpfcpm.fc.hp.com:crwit
user:root@wookie.fc.hp.com:crwit
any_other:-r---
```

- 설치된 소프트웨어에 대한 액세스를 표시하려면 다음과 같이 입력합니다.

```
swacl -l root @ newdist

# swacl      Installed Software Access Control List
#
# For host:   newdist:
#
# Date:      Fri Nov 03 10:33:04 2001
#
# Object Ownership:  User= root
#                    Group=other
#                    Realm=newdist.fc.hp.com
```

```
#
# default_realm=newdist.fc.hp.com
object_owner:crwit
user:root:crwit
user:root:crwit
any_other:-r---
```

- 대상 호스트에서 저장소와 루트를 만들 수 있는 권한을 표시하려면 다음과 같이 입력합니다.

```
swacl -l host @ newdist
```

```
#
# swacl      Host Access Control List
#
# For host:  newdist
#
# Date:     Fri Nov 03 10:34:06 2001
#
# Object Ownership:  User= root
#                   Group=sys
#                   Realm=newdist.fc.hp.com
#
# default_realm=newdist.fc.hp.com
user:fred:crwit
user:root:crwit
user:smp:crwit
user:root@udltools.fc.hp.com:crwit
user:fred@hpfred.fc.hp.com:crwit
user:chrisr@prewd.fc.hp.com:crwit
any_other:-r---
```

- 저장소의 모든 제품(“\\*”)에 대해 액세스 권한을 가진 사용자를 나열하려면 다음과 같이 입력합니다.

```
swacl -l product \* @ newdist:/var/spool/sw
```

```
#
# swacl      Product Access Control Lists
#
# For depot: newdist:/var/spool/sw
#
# Date:     Fri Nov 03 10:34:06 2001
#
```

```
# For product:  product1,r=1.0
#
# Object Ownership:  User= root
#                   Group=other
#                   Realm=newdist.fc.hp.com
#
# default_realm=newdist.fc.hp.com
object_owner:crwit
user:root:crwit
user:root@prewd.fc.hp.com:crwit
any_other:-r---
```

## 사용자에게 제품 관리 권한 부여

제품을 패키징하는 사용자들은 제품을 저장하기 위해 SD-UX 저장소에 액세스할 필요가 있습니다.

ACL에서 **a**는 모든 권한을 나타내는 축약형 표기입니다(crwit).

사용자 mary에게 저장소에 새 제품을 추가할 수 있는 권한을 부여하려면 다음과 같이 입력합니다.

```
swacl -l depot -M user:mary:a [@ host:depot]
```

사용자 mary에게 저장소의 모든 기존 제품을 수정할 수 있는 권한을 부여하려면 다음과 같이 입력합니다.

```
swacl -l product -M user:mary:a \* [@ host]
```

템플릿을 수정하여 저장소에서 다른 사람이 작성한 새로운 제품을 사용자 mary가 수정할 수 있도록 하려면 다음과 같이 입력합니다.

```
swacl -l global_product_template -M user:mary:a [@ host]
```

(위 예에서 user를 group으로 변경하고 그룹 이름을 사용하면 저장소 구조에 그룹 액세스를 추가할 수 있습니다.)

## 사용자에게 관리 권한 부여(설치/제거)

사용자(mary)에게 호스트 mysys에 소프트웨어를 설치하거나 제거하는 데 필요한 권한을 부여하려면 다음과 같이 입력합니다.

```
swacl -l root -M user:mary:a @ mysys
```

사용자 mary에게 기본 루트에 소프트웨어를 설치할 수 있는 권한을 부여하려면 다음과 같이 입력합니다.

```
swacl -l root -M user:mary:ri
```

사용자 mary에게 읽기 위해 루트를 열 수 있는 권한을 부여하려면 다음과 같이 입력합니다.

```
swacl -l root -M user:mary:r
```

사용자 mary에게 새로운 소프트웨어를 루트 객체에 설치할 수 있는 권한을 부여하려면 다음과 같이 입력합니다.

```
swacl -l root -M user:mary:i
```

원격 사용자 allen@swelter에게 swcrunch의 루트 파일 시스템을 전체적으로 관리할 수 있는 권한을 부여하려면 다음과 같이 입력합니다.

```
swacl -l root -M user:allen@swelter:a
```

(위 예에서 user를 group으로 변경하고 그룹 이름을 사용하면 저장소 구조에 그룹 액세스를 추가할 수 있습니다.)

---

## 참고

일반적으로 소프트웨어 설치 시 구성 중에 시스템 파일을 수정하게 되므로 슈퍼유저 자격으로 소프트웨어 설치 및 구성 스크립트를 실행하게 됩니다. 따라서 루트에서 쓰기 권한을 부여하는 것은 사실상 소프트웨어를 관리할 수 있는 슈퍼유저 액세스를 부여하는 것과 같습니다.

---

## 저장소에 대한 액세스 제한

저장소에 대한 읽기 액세스를 제한하려면 먼저 저장소와 저장소에 포함되어 있는 제품, 그리고 저장소의 제품을 제어하는 템플릿에서 any\_other 액세스를 제거해야 합니다.

다음과 같이 입력하여 drgw 호스트상의 저장소 alpine에 대한 액세스를 제한할 수 있습니다.

```
swacl -l depot -D any_other @ drgw:/alpine
swacl -l product -D any_other \* @ drgw:/alpine
swacl -l global_product_template -D any_other \* \
    @ drgw:/alpine
```

그런 다음 저장소 보안에서 any\_other를 제거한 후 읽기 액세스 권한을 갖는 특정 사용자(그런 다음 호스트)를 추가해야 합니다. 다음 명령은 hostA의 모든 사용자에게 저장소, 저장소에 포함되어 있는 제품 및 향후 제품에 대해 각각 read 액세스 권한을 추가합니다.

```
swacl -l depot -M other:@hostA:r @ drgw:/alpine
swacl -l product -M other:@hostA:r \* @ drgw:/alpine
swacl -l global_product_template -M other:@hostA:r \
    @ drgw:/alpine
```

다음 예에서는 로컬 슈퍼유저가 모든 원격 사용자를 swelter의 /simple\_1.depot에 액세스할 수 없게 하지만 로컬 사용자는 저장소에 액세스할 수 있도록 합니다.

```
swacl -l depot -D any_other @ /simple_1.depot
swacl -l depot -M other:r @ /simple_1.depot
swacl -l depot @ /simple_1.depot
```

```
#
# swacl      Depot Access Control List
#
# For depot:  swelter:/simple_1.depot
#
# Date:      Thu Mar  1 16:19:57 2001
#
# Object Ownership:  User= allen
#                   Group=users
#                   Realm=swelter.fc.hp.com
#
# default_realm=swelter.fc.hp.com
object_owner:crwit
other:-r---
```

이제 로컬 사용자는 other ACL의 결과로 이 저장소에 액세스할 수 있지만 원격 사용자는 액세스가 거부됩니다.

swcrunch 호스트의 사용자 shelly에게만 swelter에 있는 저장소의 소프트웨어에 액세스할 수 있는 권한을 부여하려면 다음과 같이 shelly에 대한 사용자 ACL을 추가하기만 하면 충분한 것처럼 보입니다.

```
swacl -l depot -M user:shelly@swcrunch:r @ /simple_1.depot
```

그러나 이것만으로는 충분하지 않습니다. shelly가 이 저장소에 액세스를 시도하면 보안 위반으로 실패하게 됩니다. 왜냐하면 SD-UX에서는 저장소 서버에 연결하는 SD 에이전트(swagent 프로세스)도 host ACL entry\_type을 통해 권한을 부여 받아야 하기 때문입니다.

```
swacl -l depot -M host:swcrunch:r @ /simple_1.depot
```

(사용자 shelly도 swcrunch에 소프트웨어를 설치하려면 적절한 ACL 권한이 필요 합니다.)

---

## 참고

**r**(읽기) 권한을 통해 사용자는 저장소와 제품에 액세스할 수 있고 **t**(테스트) 권한을 통해 ACL을 나열할 수 있습니다.

## 대상 호스트 추가

swinstall과 swcopy의 경우 권한이 없는 원격 호스트의 사용자가 권한이 있는 사용자로 전환하지 않도록 하기 위해 사용자와 대상 호스트를 모두 검증합니다. 다음 명령은 target이라는 호스트에게 로컬 호스트의 기본 저장소, 저장소의 현재 제품 및 저장소에 추가되는 향후 제품에 대한 읽기 권한을 global\_product\_template을 사용하여 추가합니다.

```
swacl -l depot -M host:target:r
swacl -l product -M host:target:r \*
swacl -l global_product_template -M host:target:r
```

사용자는 항상 검증되므로 많은 수의 호스트를 더 쉽게 관리할 수 있는 또 다른 방법은 모든 호스트에 읽기 권한을 허용하는 것입니다.

```
swacl -l depot -M host*:r
swacl -l product -M host*:r \*
swacl -l global_product_template -M host*:r
```

---

### 참고

“\*”는 host ACL 유형에 대해 유일하게 지원되는 값입니다.

---

## 일시적인 액세스 제한

ACL을 수정하지 않고 로컬 슈퍼유저를 제외한 누구에게도 액세스를 제한하는 간단한 방법은 저장소 등록을 해제하는 것입니다.

```
swreg -u -l depot [@ depot]
```

그런 다음 나중에 다시 등록할 수 있습니다.

```
swreg -l depot [@ depot]
```

## SD-UX 네트워크 달기

SD-UX 비밀은 호출자의 자격 증명에 대한 신뢰를 입증하는 데 사용됩니다. 이는 SD-UX가 호출자 호스트의 자격을 검증하기 위해 사용하는 암호입니다. 기본 비밀 필드는 HP-UX 컨트롤러의 기본 설정과 일치하도록 제조 시 설정됩니다. 모든 비밀(즉, 컨트롤러, 대상 및 저장소)은 반드시 동일해야 합니다.

---

## 참고

---

HP-UX SD-UX 컨트롤러의 기본 비밀번호도 함께 변경하지 않는 한 기본 비밀번호를 변경하지 마십시오. 이들 두 비밀번호는 서로 일치해야 합니다.

SD-UX가 관리할 수 있는 호스트 집합은 네트워크상의 모든 SD-UX 컨트롤러와 대상 호스트에서 기본 비밀번호를 변경함으로써 제한할 수 있습니다. 기본 비밀번호는 `/var/adm/sw/security/secrets`에 있습니다.

이 파일의 기본 비밀번호는 변경할 수 있습니다.

기본값 `new secret`

자세한 내용은 267페이지의 “호스트 간 보안: 공유 비밀번호 파일”을 참조하십시오.

## ACL 편집

`swacl` 명령이 `-M`, `-D` 또는 `-F` 옵션 없이 호출되면 지정된 ACL을 읽고 일반 텍스트로 변환한 다음 `stdout`로 출력합니다. 명령 출력은 파일로 재지정하여 인쇄하거나 편집할 수 있습니다. 편집한 다음 위에서 설명한 `-F file` 옵션을 사용하여 이전 ACL 전체를 교체할 수 있습니다. 이 절차에서는 모든 ACL 편집 기능을 제공합니다.

편집 파일(ACL 나열)을 만들고 `-F`, `-D` 또는 `-M` 옵션으로 해당 파일을 수정하는 제어 권한을 만들려면 ACL 내에 테스트 권한이 있어야 합니다. 모든 ACL 항목에 테스트 권한이 포함되어 있어야 합니다.

대체 ACL에 오류가 발견되지 않고 ACL에 대해 적절한 권한을 가지고 있는 경우 정상적으로 대체됩니다. 변경할 수 있는 권한이 충분하지 않아 대체에 실패하는 경우 오류가 발생하고 해당 객체는 생략됩니다.

기존 항목을 변경 또는 삭제하거나 항목을 ACL에 추가할 수 있습니다.

---

**참고**

ACL을 편집하여 ACL에 액세스하지 못하게 할 수도 있습니다. ACL에서 사용자 자신의 제어(c) 권한을 부주의하게 제거하는 일이 없도록 조심해야 합니다. 안전 장치의 하나로, 로컬 슈퍼유저는 항상 swacl을 사용하여 SD-UX ACL을 편집할 수 있습니다.

---

로컬 호스트 lehi.fc.hp.com의 사용자 rob가 제품(FORTRAN)을 보호하는 다음과 같은 ACL을 기반으로 몇 가지 예를 설명합니다.

```
# swacl      Product Access Control Lists
#
# For host:  lehi:/
#
# Date:     Mon Nov 06 16:39:58 2001
#
# For product:  FORTRAN,r=9.0,v=HP
# Object Ownership:  User=root
#                   Group=sys
#                   Realm=lehi.fc.hp.com
# default_realm=lehi.fc.hp.com
object_owner:crwit
user:barb:-rt
user:ramon:-rt
group:swadm:crwit
host:alma.fc.hp.com:-rt
any_other:-rt
```

/var/spool/sw 저장소(기본 저장소)의 FORTRAN 제품에 대한 ACL을 나열하고 편집을 준비할 수 있습니다.

```
swacl -l product FORTRAN >acl_tmp
```

이렇게 하면 위의 ACL을 acl\_tmp 파일로 가져와 편집할 준비를 하게 됩니다. 적당한 텍스트 편집기로 acl\_tmp 파일을 편집합니다.

FORTRAN에 대해 ACL의 모든 항목을 교체하려면 다음과 같이 입력합니다.

```
swacl -l product -F acl_tmp FORTRAN
```

/var/spool/sw\_dev 저장소의 기본 제품 템플릿을 편집하려면 다음과 같이 입력합니다.

```
swacl -l product_template @ /var/spool/sw_dev >tmp_file
```



그런 다음 tmp\_file을 편집하여 ACL을 교체합니다.

```
swacl -l product_template -F tmp_file \  
@ /var/spool/sw_dev
```

사용자 barb와 그룹 swadm에 대한 항목을 삭제하려면 다음과 같이 입력합니다.

```
swacl -D user:barb -D group:swadm -l product FORTRAN
```

사용자 ramon에게 FORTRAN 제품을 수정할 권한을 부여하려면 다음과 같이 입력합니다.

```
swacl -M user:ramon:trw -l product FORTRAN
```

사용자 pam의 항목에 전체 관리 권한("a"는 crwit에 대한 축약형임)을 추가하려면 다음과 같이 입력합니다.

```
swacl -M user:pam:a
```

원격 호스트 dewd와 stewd의 그룹 swadm에 있는 모든 사용자에게 기본 로컬 저장소의 FORTRAN 제품에 대한 전체 관리 권한을 부여하는 항목을 추가하려면 다음과 같이 입력합니다.

```
swacl -M group:swadm@dewd:a -M group:swadm@stewd:a \  
-l product FORTRAN
```

dewd 호스트의 기본 저장소를 보호하는 ACL을 나열하려면 다음과 같이 입력합니다.

```
swacl -l depot @ dewd
```

---

## ACL이 사용자에게 일치되는 방법

ACL 권한은 일치하는 전체 항목이 아닌 **단일 ACL** 항목의 일치로 결정됩니다. 점검은 가장 제한적인 항목 유형에서 가장 광범위한 항목 유형으로 수행됩니다.

사용자 항목 유형에서 일치가 발견되면 더 이상의 확인은 수행되지 않고 일치 항목의 권한 필드에 의해 해당 사용자에 대한 권한이 완전하게 정의됩니다. 일치된 사용자는 더 폭넓은 권한을 가진 그룹의 구성원일 수 있지만 이것은 중요하지 않습니다.

---

### 참고

로컬 슈퍼유저는 ACL에 관계없이 모든 로컬 SD-UX 객체에 액세스할 수 있습니다.

ACL 일치 알고리즘은 다음과 같습니다.

1. 사용자가 로컬 슈퍼유저이면 모든 권한이 부여됩니다.
2. 사용자가 객체의 소유자이면 `object_owner` 권한이 부여됩니다.
3. 사용자가 `user` 항목에 일치하면 `user` 권한이 부여됩니다.
4. 일치하는 `group` 항목이 있으면 사용자의 1차 그룹 및 2차 그룹에 일치하는 모든 그룹 항목에 대해 부여되는 권한이 축적됩니다.
5. 적절한 다른 항목이 일치하면 다른 권한이 부여됩니다.
6. `any_other` 항목이 있으면 `any_other` 권한이 부여됩니다.
7. 권한을 부여하지 않습니다.

## ACL 항목

ACL은 객체가 만들어질 때 그 객체와 연관되는 **항목**의 집합으로 구성됩니다. 이 항목은 어떤 사용자, 그룹 또는 호스트가 객체에 액세스할 수 있는 권한을 가지고 있는지 정의합니다. ACL 항목에는 다른 시스템에 대한 호출을 시작하는(RPC를 호출하는 에이전트에 대한) 사용자, 그룹 또는 호스트 시스템인 **주체** 개념이 포함됩니다.

ACL 항목은 다음과 같은 세 가지 필드로 구성됩니다.

```
entry_type[:key]:permissions
```

예를 들어 SD-UX 객체에 대한 ACL 항목은 다음과 같습니다.

**user:fred:r-ctw**

즉 fred라는 이름의 사용자는 객체를 제어하고(c) , 읽고(r) 쓰고(w) 테스트(t) 할 수 있지만 대시(-)를 통해 새 객체를 삽입 및 작성(i)할 수 없다는 것을 나타냅니다.

### 참고

crwit 권한은 순서에 관계없이 지정할 수 있습니다.

ACL entry\_type은 다음 값 중 하나여야 합니다.

표 9-3

### SD-UX ACL 항목 유형

유형	적용되는 권한
user	key 필드에 이름이 지정되는 사용자 주체
group	key 필드에 이름이 지정되는 그룹 주체
host	호스트 시스템(설치나 복사 시 사용자를 대신하여 작업을 수행하는 대상 에이전트)
other	일치하는 사용자 및 그룹 항목이 없는 주체

표 9-3 SD-UX ACL 항목 유형(계속)

유형	적용되는 권한
any_other	어떤 항목과도 일치하지 않는 주체
object_owner	객체의 소유자
object_group	객체가 속하는 그룹의 구성원

호스트 객체(저장소, 루트 및 소프트웨어를 포함하고 있는 컴퓨터 시스템)를 **host** 항목 유형(대상 시스템에 대한 액세스 권한을 정의)과 혼동하지 마십시오.

객체 소유자의 사용자 및 그룹은 해당 객체가 작성될 때 결정되어 객체를 만든 사람의 ID를 기반으로 자동으로 기록됩니다. 이 정보는 `user`, `group` 및 `realm`으로 기록됩니다. ACL의 `object_owner` 또는 `object_group` 항목 유형을 통해 SD-UX ACL 관리자는 객체에 대한 소유자 및 그룹 정보를 검색할 수 있으며 요청자와 일치하는 항목이 있으면 지정된 대로 권한을 부여합니다.

ACL 하나에 `user`, `group` 및 `host` 유형의 항목은 여러 개 있을 수 있지만 `object_owner`, `object_group` 및 `any_other`는 각각 하나씩만 있을 수 있습니다. 로컬(키 없음) **other** 항목은 최대 하나가 가능하고 원격(키 있음) **other** 항목은 무제한으로 가능합니다.

## ACL 키

ACL 항목의 두 번째 부분은 키입니다. 아래 표에서는 특정 항목 유형에 대해 가능한 키 값을 보여줍니다.

표 9-4 SD-UX ACL 항목 키 값

항목 유형	키 내용
user	사용자 이름 [선택적으로, @ <i>remote-host</i> ]
group	그룹 이름 [선택적으로, @ <i>remote-host</i> ]
host	호스트 이름

표 9-4 SD-UX ACL 항목 키 값(계속)

항목 유형	키 내용
other	[선택적으로, @ <i>remote-host</i> ]
any_other	키가 허용되지 않음

ACL을 나열할 때 로컬 시스템이 호스트 검색 메커니즘(DNS, NIS 또는 /etc/hosts)을 통해 주소를 해석할 수 없는 경우 원격 호스트가 인터넷 주소 형식(예: 15.12.89.10)으로 인쇄됩니다. 원격 호스트를 -M 및 -D 옵션에서 사용하면 인식(해석)할 수 있어야 합니다. 인식되지 않는 원격 호스트 값은 -F 옵션으로 제공된 파일에 보관됩니다.

### ACL 권한

ACL이 부여할 수 있는 권한은 5가지(crwit)입니다.

표 9-5

#### ACL 권한

제어(c)	ACL을 편집하거나 변경할 수 있는 권한
테스트(t)	객체에 대한 액세스를 테스트하는 권한(즉, ACL 읽기)
삽입(i)	새로운 제품, 저장소 또는 루트를 설치할 수 있는 권한
쓰기(w)	호스트, 저장소, 루트 또는 제품을 변경할 수 있는 권한
읽기(r)	저장소, 루트, 제품 및 속성을 표시할 수 있는 권한

ACL 항목에서 이러한 권한은 c, t, i, w 및 r로 축약됩니다. 모든 권한을 부여하려면 crwit를 사용하는 대신 축약형 문자인 a를 사용하여 모든 권한을 나타낼 수 있습니다.

권한의 의미는 객체의 유형에 따라 모두 다르고, 권한이 특정 순서에 따라 표시될 필요도 없습니다. 루트는 제품 수준의 보호를 제공하지 않으므로 루트에 설치된 제품에 대한 모든 권한은 루트 자체를 보호하는 ACL에 의해 제어됩니다.

제품 수준 보호는 저장소 ACL을 통해 저장소 자체를 보호하고 제품 ACL을 통해 저장소 내의 제품을 보호하는 방식으로 저장소에 제공됩니다.

아래 표에서 SD-UX 객체 권한 및 이 객체 권한이 적용될 수 있는 ACL을 간략하게 보여줍니다.

표 9-6

SD-UX ACL 권한 정의

권한	가능한 작업:			
	호스트 시스템	루트	저장소	저장소상의 제품
c(제어)	모든 ACL을 편집합니다.			
t(테스트)	객체에 대한 액세스를 테스트하고, ACL 자체를 읽습니다(나열).			
i(삽입)	새로운 저장소나 루트를 삽입합니다.	새로운 제품을 삽입합니다.	새로운 제품을 삽입합니다.	N/A
w(쓰기) <sup>a</sup>	호스트를 변경합니다.	루트나 제품을 변경합니다.	저장소를 변경합니다.	제품을 변경합니다.
r(읽기) <sup>b</sup>	저장소 및 루트를 나열합니다.	루트 및 제품 속성을 나열합니다.	저장소 및 제품 속성을 나열합니다.	제품 파일을 읽습니다.

a. 쓰기 권한은 객체를 변경하거나 삭제할 수 있는 권한을 의미하는데 단, 호스트 소스 객체는 삭제할 수 없습니다.

b. 컨테이너(호스트, 루트 및 저장소)에 대한 읽기 권한을 통해 사용자는 컨테이너 내용을 나열할 수 있고, 저장소 내 제품에 대한 읽기 권한을 통해 사용자는 제품을 복사 또는 설치할 수 있습니다.

## 객체 보호

제품 삽입 및 삭제 권한의 제어는 루트와 저장소 간에 차이가 있습니다.

루트에서 누구든지 제품을 삽입하거나 삭제할 수 있는 권한은 루트의 ACL에 들어 있습니다. 루트에서 쓰기 권한을 가지고 있으면 해당 루트에서는 어떤 제품이든 변경하거나 삭제할 수 있지만, 루트에서는 제품 수준의 제어가 이루어지지 않습니다.

저장소 ACL은 새로운 제품의 삽입(작성)을 제어하고 삽입된 객체는 수정 및 삭제를 제어하는 자체 ACL을 갖고 있습니다. 이를 통해 저장소에서 제품의 작성자(소유자)는 같은 저장소에서 다른 사용자의 제품에 영향을 미칠 수 있는 더 넓은 범위의 쓰기 권한이 없어도 해당 제품을 변경하거나 삭제할 수 있습니다.

이는 특정 제품에 대한 관리 제어를 지정된 관리자에게 위임할 수 있으므로 제품 제어에 유용합니다. 또한 저장소에서 제품이 작성될 때 작성자의 사용자와 그룹 ID가 제품 정보에 기록됩니다.

제품 ACL에 소유자에게 쓰기 권한을 부여하는 `object_owner` 항목이 들어 있는 경우, 제품 작성자는 자동으로 제품을 변경하거나 삭제할 수 있는 권한을 갖게 됩니다. 그러므로 삽입 권한이 있는 사용자는 새로운 제품을 복사하거나 자신의 제품을 삭제만 할 수 있기 때문에 저장소에서는 삽입 작업을 보다 폭넓게 허용할 수 있습니다. 즉 사용자는 자신이 제어할 수 없는 중요한 제품을 실수로 삭제하는 경우에 대해 걱정할 필요가 없습니다.

이 보호 체계의 근거는 BSD 파일 시스템에 도입된 메커니즘으로부터 빌려 온 것입니다. BSD 디렉토리에 대해 `write` 권한이 있으면 디렉토리에 파일을 작성할 수 있습니다. 만일 해당 디렉토리에 스티키 모드 비트가 설정되어 있으면 파일 소유자, 디렉토리 소유자 또는 슈퍼유저만이 이 파일을 제거하거나 이름을 변경할 수 있습니다.

예를 들어 루트가 소유하고 있고 “넓은 범위의” 쓰기 권한을 가지고 있으며 스티키 비트가 수동으로 설정되어 있는(즉, 모드 `1777`) `/tmp`에서는 누구나 다른 사람이 제거할 수 없는(작성자 자신과 슈퍼유저는 제외) 파일을 만들 수 있습니다. 이러한 특성으로 인해 `/tmp`에 있는 사용자 파일을 다른 사람이 삭제할 수 없기 때문에 이 디렉토리를 임시 작업을 저장하는 보다 안전한 공간으로 사용할 수 있습니다.

등록되지 않은 저장소에서 설치하거나 복사하려면 사용자와 대상 에이전트의 호스트에 저장소 호스트에 대한 삽입 권한이 있어야 합니다. 이 권한이 대상 호스트에 대해 거부되면 저장소의 데몬 로그에 다음과 같은 메시지가 생성됩니다.

```
ERROR: Access denied to SD agent at host lucille on
behalf of rob@lucille to start agent on unregistered
depot "/users/rob/depot." No (i)nsert permission on
host.
```

```
07/23/01 15:51:06 MDT
```

이 메시지를 통해 저장소 호스트에 대한 삽입 권한이 없는 것은 사용자 `rob@lucille`이 아니라 `lucille`의 에이전트라는 것을 알 수 있습니다.

원격 호스트 ACL에는 삽입 권한을 부여하는 항목이 사용자와 대상 호스트에 대해 각각 하나씩, 즉 두 개 있어야 합니다.

예를 들어 소스 호스트 `desi`의 등록되지 않은 저장소로부터 사용자 `rob`이 대상 호스트 `lucille`에 제품을 설치할 수 있도록 권한이 부여된 경우

```
swacl -l host @ desi
```

위 명령은 최소 ACL 항목을 표시해야 합니다.

```
user:rob@lucille:-i-
host:lucille:-i-
```

또 다른 방법으로 `rob`은 `swinstall`이나 `swcopy`를 실행하기 전에 `swreg` 명령을 사용하여 위의 첫 번째 항목만 갖는 저장소를 등록할 수 있습니다.

### 호스트 시스템 ACL

호스트 시스템은 SD-UX에서 보호되는 가장 높은 수준의 객체입니다. 호스트 ACL은 각 호스트 시스템을 보호하여 저장소와 루트를 작성하는 권한을 제어합니다. 호스트 ACL은 다음과 같은 권한을 부여할 수 있습니다.

표 9-7

#### 호스트 ACL 권한

r(읽기)	호스트상의 저장소 및 루트 목록을 포함하여 호스트 속성을 표시할 수 있는 권한
w(쓰기)	호스트 객체를 변경할 수 있는 권한
i(삽입)	호스트에서 새로운 저장소나 루트를 작성하고 등록할 수 있는 권한
c(제어)	ACL을 편집하거나 변경할 수 있는 권한
t(테스트)	객체에 대한 액세스 권한을 테스트하고 ACL을 나열할 수 있는 권한

저장소와 루트 소스를 작성하고, 소스를 나열하고 ACL을 관리할 수 있는 권한을 `rob`이라는 사용자에게 부여하고 호스트의 저장소와 루트를 나열할 수 있는 권한을 부여하는 호스트 시스템 ACL은 다음과 같을 것입니다.

```
user:rob:r-ic-
any_other:r
```

`any_other`에 `t`(테스트) 권한이 없기 때문에 `c`(제어 권한)가 있는 `rob`만이 이 ACL을 나열할 수 있습니다.



**루트 ACL**

루트를 보호하는 ACL에 명시된 주체(사용자)에게는 설치된 제품을 관리할 수 있는 권한이 부여됩니다. 루트와 연관된 권한은 다음과 같습니다.

표 9-8

**root 권한**

i(삽입)	새로운 제품을 설치할 수 있는 권한
r(읽기)	루트의 내용을 나열할 수 있는 권한
w(쓰기)	루트 자체 또는 루트 내 제품을 삭제할 수 있는 권한
c(제어)	ACL을 편집하거나 변경할 수 있는 권한
t(테스트)	객체에 대한 액세스 권한을 테스트하고 ACL을 나열할 수 있는 권한

lois라는 사용자에게 소프트웨어를 읽고 쓰며 삽입하는 권한을 부여하고 swadm이라는 그룹의 구성원에게 모든 가능한 권한을 부여하는 루트 ACL의 예는 다음과 같습니다.

```
user:lois:rwi-
group:swadm:crwit
```

루트가 작성되면 루트는 호스트로부터 파생된 기본 ACL에 의해 자동으로 보호됩니다. swacl을 사용하여 이 ACL의 초기 값을 변경합니다. 자세한 내용은 260페이지의 “ACL 템플릿”을 참조하십시오.

**저장소 ACL**

저장소를 보호하는 ACL에 명시된 주체는 저장소를 관리하고 새로운 제품을 작성할 수 있는 권한이 부여된 사용자입니다. 저장소와 연관된 권한은 다음과 같습니다.

표 9-9

**저장소 권한**

i(삽입)	새로운 제품을 저장소에 복사할 수 있는 권한
r(읽기)	저장소 소스의 내용(제품)을 나열할 수 있는 권한
w(쓰기)	비어 있는 저장소를 삭제하고 저장소 자체(저장소 내 제품이 아닌)의 등록을 해제할 수 있는 권한
c(제어)	ACL을 편집하거나 변경할 수 있는 권한

표 9-9

## 저장소 권한(계속)

t(테스트)	객체에 대한 액세스 권한을 테스트하고 ACL을 나열할 수 있는 권한
--------	---------------------------------------

저장소 작성자에게는 모든 권한을 부여하고, 사용자 george에게는 소프트웨어 제품을 나열 및 삽입할 수 있는 권한을 부여하며 swadm 그룹의 구성원에게는 제품을 나열 및 삽입하고 ACL을 변경하며 저장소 자체를 삭제할 수 있는 권한을 부여하며, 기타 다른 사람들에게는 저장소의 내용을 표시할 수 있는 권한을 부여하는 저장소 ACL에는 다음과 같습니다.

```
object_owner:crwit
user:george:-r-i-
group:swadm:crwi-
any_other:-r-
```

저장소 소스 객체가 작성되면 루트는 호스트로부터 파생된 기본 ACL에 의해 자동으로 보호됩니다. 이 저장소에 삽입된 제품은 이 저장소에서 파생된 ACL에 의해 자동으로 보호됩니다. 이 개념은 260페이지의 “ACL 템플릿”에서 설명합니다.

## 제품 ACL

제품 ACL은 저장소의 제품에만 적용됩니다. 루트의 제품은 루트 ACL에 의해 보호됩니다. 제품에 대한 액세스 권한이 부여되는 두 가지 주체 클래스가 있습니다.

표 9-10

## 제품 주체

users	다양한 관리 권한이 부여됩니다. 이 클래스에는 로컬 및 원격의 그룹(group) 및 기타 사용자(others)가 포함됩니다.
hosts	제품을 설치할 수 있는 읽기 권한이 부여되는 대상 시스템(에이전트/데몬)입니다.

제품의 권한은 다음과 같습니다.

표 9-11

## 제품 권한

w(쓰기)	제품 및/또는 제품 정보를 변경 및 삭제할 수 있도록 users에 부여된 권한
-------	---

표 9-11

## 제품 권한(계속)

r (읽기)	소스 저장소 제품을 읽을 수 있도록 <code>target_hosts</code> 에 부여된 권한 (즉, 보호된 제품을 설치할 수 있도록 원격 시스템에 권한을 부여합니다.)
c (제어)	ACL을 편집하거나 변경할 수 있는 권한
t (테스트)	객체에 대한 액세스를 테스트할 수 있는 권한

사용자 `swadm`과 제품 작성자에게 모든 권한을 부여하고 개방형 읽기 권한(모든 시스템에 자유로운 배포를 가능하게 함)을 허용하는 제품 ACL의 예는 다음과 같습니다.

```
user:swadm:crw
object_owner:crw
any_other:-r-
```

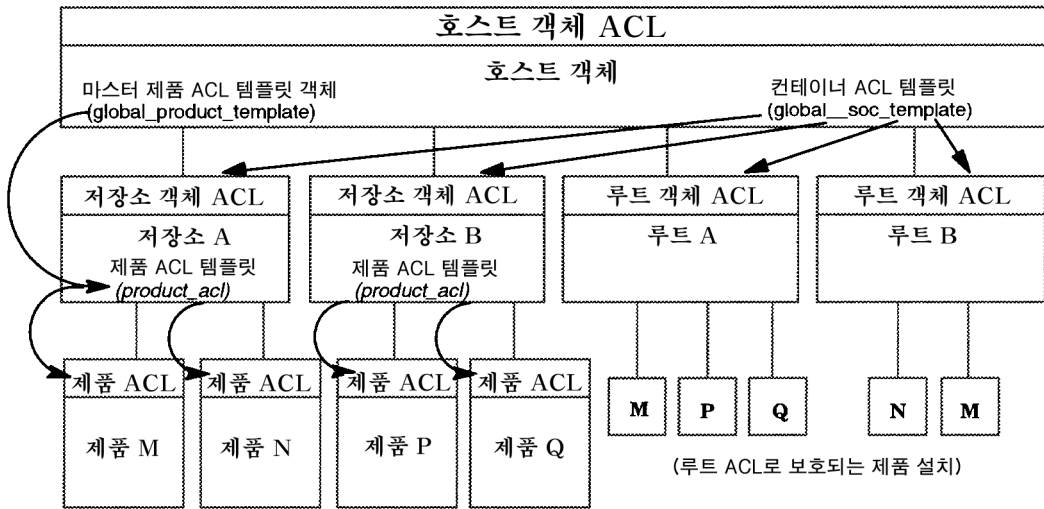
## 참고

제품 객체가 작성되면 저장소/루트 소스의 기본 ACL에 의해 자동으로 보호되며, 저장소/루트 소스가 없는 경우 호스트의 ACL에 의해 자동으로 보호됩니다.

### ACL 템플릿

새로 작성되는 객체를 보호하기 위해 초기 ACL을 작성하는 데 사용되는 ACL은 제품 ACL 템플릿(global\_product\_template 또는 product\_template)과 컨테이너 ACL 템플릿(global\_soc\_template) 두 가지가 있습니다.

그림 9-2 ACL 템플릿



swcopy 또는 swpackage를 사용하여 제품이 저장소에 들어갈 때 SD-UX는 제품 ACL 템플릿(제품이 들어 있는 저장소에서 제공)을 사용하여 새로운 제품 ACL의 초기 권한을 정의합니다.

SD-UX는 호스트 시스템의 제품 ACL 템플릿(global\_product\_template)을 사용하여 새로운 저장소의 제품 ACL 템플릿을 초기화하고 호스트 시스템의 컨테이너 ACL 템플릿(global\_soc\_template)을 사용하여 저장소와 루트 ACL을 초기화합니다.

따라서 호스트에는 세 개의 ACL이 존재합니다.

- 호스트 ACL
  - 호스트 객체 자체에 첨부되어 그에 대한 액세스를 제어합니다.
- 컨테이너 ACL 템플릿(global\_soc\_template)

새로운 저장소와 호스트에서 작성된 루트를 보호하는 ACL을 초기화하는 데 사용됩니다.

- 제품 ACL 템플릿(global\_product\_template)

호스트에서 작성된 저장소상의 제품 ACL 템플릿을 초기화하는 데 사용됩니다.

제품 저장소에는 두 가지 ACL이 있습니다.

- 제품에 대한 권한을 결정하는 데 사용되는 저장소의 ACL
- 저장소의 새 제품을 보호하는 ACL을 초기화하기 위해 사용하는 저장소의 제품 ACL 템플릿(product\_template)

설치 시 하나의 ACL이 존재합니다(루트).

- 루트 및 루트에 설치된 제품을 보호하는 루트 ACL

마지막으로 제품에 대한 ACL이 하나 존재합니다.

- 제품에 대한 권한을 결정하는 데 사용되는 제품의 ACL

모든 호스트에는 호스트를 보호하는 ACL이 있어야 하고 암시적인 저장소와 제품 ACL에 대한 초기화 데이터를 제공하는 두 개의 템플릿 ACL(제품 및 컨테이너)이 있어야 합니다. 이 세 가지 ACL은 모두 호스트에 SD-UX가 설치될 때 만들어집니다.

### 기본 ACL 템플릿 항목

호스트 시스템의 컨테이너 ACL 템플릿은 해당 호스트의 모든 저장소와 루트에서 초기 권한을 지정합니다. 호스트에는 새로운 각각의 저장소로 복사되는 제품 ACL 템플릿의 마스터 사본이 들어 있습니다.

호스트 ACL 기본 세트는 SD-UX가 설치될 때 제공되며 이는 SD-UX 관리자가 변경할 수 있습니다. SD-UX 설치 직후 호스트 시스템 ACL의 내용은 다음과 같습니다.

#### 호스트 ACL

- 아래의 호스트 ACL은 호스트의 저장소와 루트를 나열할 수 있는 글로벌 (any\_other) 권한을 허용합니다.

```
object_owner:swadm:crwit
```

```
any_other:-r---
```

---

**참고**

---

로컬 슈퍼유저는 ACL 항목이 없어도 항상 모든 권한을 가지고 있음을 기억하십시오.

**컨테이너 ACL 템플릿**

- 아래의 컨테이너 ACL 템플릿은 새로운 저장소나 루트의 소유자나 작성자 (object\_owner)에게 새로운 저장소나 루트를 관리하고 해당 ACL을 변경할 수 있는 권한을 부여합니다. 또한 새로운 저장소나 루트의 제품을 나열하는 글로벌 권한(any\_other)도 부여합니다.

```
object_owner:crwit
any_other:-r---
```

**제품 ACL 템플릿**

- 아래의 제품 ACL 템플릿은 각 제품의 object\_owner 항목을 통해 이 호스트의 저장소에 설치된 제품에 대해 모든 작업을 수행할 수 있는 권한을 각각의 작성자 (즉, 소유자)에게 부여합니다. 또한 제품을 읽고(즉, 설치하고) 테스트할 수 있는 권한을 모든 호스트에 부여합니다(any\_other 항목).

```
object_owner:crwit
any_other:-r---
```

- 모든 호스트를 포괄함과 동시에 any\_other 항목은 이 경우 제품 소유자를 제외한 모든 다른 사용자에게도 적용됩니다. 그러나 SD-UX에서 제품 읽기 권한은 호스트 주체에 대해서만 의미를 지니며 다른 가능한 제품 권한은 호스트에 전혀 적용되지 않기 때문에 any\_other 항목은 모호해질 가능성 없이 필요에 따라 사용자 및 호스트 권한이 지나치게 많이 부여될 수 있습니다. SD-UX를 사용하여 솔루션을 실행할 때 이러한 권한 과다 부여를 염두에 두어야 합니다.

이러한 호스트 ACL 기본값은 SD-UX의 관리 기능을 제어할 수 있는 훌륭한 시작점이 되는 동시에 루트 대상에 설치할 소프트웨어를 읽을 수 있는 열기 액세스를 제공합니다.

## SD-UX 시스템에서의 보안

데이터에 대한 액세스를 제어하는 것이 컴퓨터 보안의 가장 큰 현안입니다. SD-UX에서 파일 소유자와 수퍼유저는 파일의 권한 비트를 설정하거나 조작하여 소유자, 그룹 및 기타 사용자의 액세스 권한을 부여하거나 제한함으로써 필요에 따라 파일에 대한 액세스를 허용하거나 거부합니다. 예를 들어 다음과 같은 파일 목록은

```
-rwxr-xr 1 doug admin 738 Mar 26 12:25 datafile
```

아래의 내용을 보여줍니다.

- 파일 소유자는 사용자 doug입니다.
- 파일의 그룹은 admin입니다.
- 파일의 이름은 datafile입니다.
- 소유자 권한은 read, write 및 execute(rwx)입니다.
- 그룹 권한은 read 및 execute(r-x)입니다.
- 기타 권한은 read only(r-)입니다.

SD-UX 명령은 기본적으로 객체를 저장하는 SD-UX 파일 시스템을 사용하는 객체 관리 명령입니다. 파일 시스템을 통해 어떤 객체에 대해서든 액세스를 확보할 필요가 없으므로, 파일 시스템 보호 체계는 이러한 객체가 저장된 파일 디렉토리에 대한 액세스를 차단하는 것을 기본으로 합니다.

SD-UX 객체와 더불어 SD-UX에 의해 사용되거나 관리되는 몇몇 관리 파일(로그, 구성 및 세션 파일)이 있습니다. 이들 파일은 실제로는 SD-UX 객체가 아니며 편집기와 인쇄 유틸리티와 같은 일반적인 명령을 통해 액세스할 수 있습니다. 이러한 파일은 일반 파일 시스템 보호 모드로 보호됩니다.

SD-UX 에이전트가 수행하는 대다수의 기능에는 권한이 필요합니다. 시스템 디렉토리(예: /etc 및 /dev 디렉토리)에 파일을 설치하고, 제어 스크립트를 통해 시스템 파일을 사용자 정의하는 것과 같은 일부 작업에는 수퍼유저의 권한이 필요합니다. 이런 이유로 SD-UX 에이전트는 항상 수퍼유저로 실행해야 합니다.

모든 시스템 사용자는 SD-UX 컨트롤러를 실행할 수 있으며 이는 수퍼유저만 사용하도록 제한된 것이 아닙니다. 일반적으로 컨트롤러는 대상 호스트에 원격 프로시저 호출(RPC)을 실행함으로써 작업을 수행하지만, 때로 중요한 로그, 구성 및 세션 보안 파

일에 액세스하기 위해서는 특별한 권한이 필요하기도 합니다. 컨트롤러는 권한이 필요한 중요한 작업에 한해 유효한 슈퍼유저 권한을 가지고 실행되는 `set-uid root` 프로그램으로서 작업을 처리한 후에는 실제 사용자의 `uid`로 전환됩니다.

다음은 SD-UX 파일 시스템 보호 체계에 대한 간략한 설명입니다.

- SD-UX 파일은 그룹 및 기타 중요한 디렉토리 권한 모드를 0으로 설정하여 슈퍼유저가 아닌 다른 사람이 액세스하지 못하게 합니다.
- 로컬 호스트에서 실행되는 에이전트와 데몬만 SD-UX 파일에 직접 액세스할 수 있습니다. 다른 모든 기능(컨트롤러, 유틸리티 등)은 파일에 간접적으로 액세스하기 위해 RPC를 사용하여 에이전트를 거칩니다. 에이전트 또는 데몬은 이러한 모든 작업에 대해 자격 인증 및 권한 인증 검사를 수행합니다.
- SD-UX 디렉토리의 디렉토리 보호 계층 구조를 피해 가는 하드 링크가 존재하지 않거나 유효한 제한을 나열하는 객체가 들어 있는 디렉토리 내용의 기밀성을 침해하는 심볼릭 링크가 없을 수도 있습니다. SD-UX 객체에 대해 단일(정식) 경로를 사용하면 이렇게 링크 지정으로 인해 생기는 문제를 방지할 수 있습니다.

따라서 SD-UX 파일은 슈퍼유저가 아닌 사용자가 액세스할 수 없도록 완전히 보호되고 숨겨집니다.



## SD-UX 내부 권한 인증

이 절에서는 다음과 같은 내용에 대해 설명합니다.

- SD-UX 자격 증명
  - 사용자의 자격 증명 및 권한을 가지고 실행되는 컨트롤러
  - 시스템 ID를 가지고 실행되는 에이전트
- 호스트 간 보안: 공유 비밀 파일

SD-UX 보안이 DCE 보안을 대신하지는 못합니다. SD-UX는 사용자들이 손해를 입히기 위해 적대적인 공동의 노력을 하지는 않는다는 전제 하에 유용한 보호 체계를 제공합니다.

SD-UX가 사용하는 DCE 보안 기능의 대부분은 SD-UX에 포함된 DCE 런타임 라이브러리의 기능입니다. 이 라이브러리는 DCE RPC 기능 및 ACL을 지원하는 데 필요한 DCE 보안 서비스의 일부를 제공합니다.

완전한 DCE 보안 서비스가 없으면 RPC 호출의 소스 및 대상 시스템이 로컬이라 하더라도 SD-UX RPC 호출을 하는 사용자의 ID를 신뢰성 있게 파악하는 것이 불가능합니다. RPC는 호출하는 클라이언트의 네트워크 주소만을 식별합니다.

## SD-UX 자격 증명

특정 작업에 어떤 사용자가 참여할 수 있는지 판별하는 것이 SD-UX 보안의 핵심입니다. SD-UX 내부 인증에서는 HP-UX uid, gid 및 호스트 이름이 사용자의 ID를 설정하는 데 사용됩니다. SD-UX 컨트롤러가 유효한 루트의 uid를 가지고 실행된다는(컨트롤러가 setuid-root 프로그램이므로) 사실이 사용자의 실제 uid로부터 확보되는 사용자의 ID에 영향을 미치지 않습니다.

RPC를 시작할 때(SD-UX 컨트롤러로), 사용자의 ID를 설명하는 구조가 에이전트로 보내지는 각 호출에 첨부되어, 컨트롤러는 컨트롤러가 실행되는 시스템의 호스트 이름(DCE에서는 **realm**이라고 함)과 함께 RPC를 호출하는 사람의 사용자 및 그룹 이름을 보냅니다.

이 구조를 **자격 증명**이라고 합니다. 자격 증명은 다음과 같이 구성됩니다.

- 사용자(주체) 이름

RPC 호출을 시작하는 사용자(RPC를 다른 에이전트에 호출하는 에이전트의 경우 호스트 시스템)

- **그룹 이름**  
사용자의 기본 그룹
- **영역 또는 로컬 호스트**  
사용자의 호스트 이름

사용자의 자격 증명이 RPC 매개 변수로 전달됩니다. RPC를 수신하는 에이전트는 이 정보를 사용하여 권한 자격 증명을 비교합니다.

### 사용자의 자격 증명 및 권한을 가지고 실행되는 컨트롤러

swinstall 또는 swremove와 같은 SD-UX 컨트롤러 프로그램은 해당 프로그램을 호출한 사용자의 권한을 가지고 작동합니다. 에이전트는 사용자가 해당 객체의 ACL을 검토하여 객체에 필요한 권한이 있는지 확인합니다. 권한이 부여되지 않으면 작동되지 않습니다.

컨트롤러는 시스템상에 있는 누구든지 실행시킬 수 있지만 작업은 (다양한 객체 ACL에서 부여된 권한에 따라) 제한됩니다. SD-UX 에이전트는 항상 사용자가 요구한 작업을 수행하기 전에 해당 권한이 부여되었는지 확인합니다.

### 시스템 ID를 가지고 실행되는 에이전트

SD-UX 에이전트와 데몬은 슈퍼유저의 권한을 가지고 실행되지만 이들이 실행되는 호스트 시스템의 특수 ID도 가지고 있습니다. 대상 에이전트가 소스 에이전트로 RPC 호출을 하면 다음과 같은 두 세트의 자격 증명이 호출과 함께 전달됩니다.

- 에이전트 시스템의 자격 증명
- 대상 에이전트가 대신 실행되는 컨트롤러에서 실행하는 사용자의 자격 증명

로컬 슈퍼유저 권한은 파일의 작성과 삭제, ACL 관리 등의 필요한 로컬 파일 시스템 작업을 수행하는 데 필요하지만, 다른 SD-UX 프로세스와 연관된 DCE RPC 작업에는 이 수준의 권한은 필요하지도 않고 바람직하지도 않습니다.

SD-UX 에이전트가 RPC를 수행할 때 SD-UX 에이전트는 특정 사용자의 ID가 아닌 실행되는 시스템의 ID를 가정합니다.

## 호스트 간 보안: 공유 비밀 파일

호출자의 자격 증명과 더불어 다른 신뢰 증명도 **RPC**에서 보내집니다. **SD-UX** 에이전트는 호출자의 자격 증명을 수락하기 전에 이 증명을 검사합니다. 이 자격 증명은 비밀 암호의 암호화를 전달하는 것으로 구성됩니다. 암호는 공유 비밀 파일로부터 읽어 옵니다. 이 파일은 시스템의 `/var/adm/sw/security/secrets`에 있습니다.

---

### 참고

---

**SD-UX Secret**은 대상 시스템과 컨트롤러에서 모두 동일해야 합니다.

에이전트는 이렇게 암호화된 비밀을 컨트롤러의 호스트와 공유하는 암호화된 로컬 비밀과 비교합니다. 비밀이 일치하지 않으면 호출이 인증되지 않고 실패합니다.

비밀은 비밀 파일의 호스트 이름에 의해 저장되고 두 시스템 간의 신뢰를 수립하는 데 사용됩니다. 컨트롤러는 컨트롤러가 실행되는 시스템의 호스트 이름에 해당하는 파일에서 비밀을 선택합니다. 에이전트는 컨트롤러에서 **RPC**를 수신하고 컨트롤러의 호스트와 연관된 비밀을 찾습니다.

예를 들어 **컨트롤러**가 `alma.fc.hp.com`에서 실행 중이고 `lehi.fc.hp.com`에서 실행 중인 **에이전트**에게 요청을 보내면 두 프로세스는 각각 자신의 비밀 파일에서 `alma.fc.hp.com`(컨트롤러의 호스트)과 연관된 비밀을 찾습니다.

다음은 공유 비밀 파일 형식의 예입니다.

```
default          quicksilver
lehi.fc.hp.com   s28ckjd9
alma.fc.hp.com   32hwt
newdist.fc.hp.com zztopy
noway.fc.hp.com  daisey
```

첫 번째 열은 컨트롤러의 호스트 이름을 나타내고 두 번째 열은 컨트롤러의 비밀을 나타냅니다.

또한 비밀 파일에 일치하는 시스템 이름이 없는 경우 사용될 비밀 기본값(위 예에서 `quicksilver`)이 명시됩니다. 이 항목은 기본 의사 호스트 이름으로 확인할 수 있습니다. 이 항목으로 같은 기본 항목을 공유하는 호스트 간에 개방적인 **SD-UX** 상호 연결이 가능해집니다. **SD-UX**에는 사용자 사이트에 맞게 변경해야 하는 비밀 `-sdu-`가 함께 제공됩니다.

호스트의 비밀을 변경할 때 반드시 사용자가 작업하는 모든 호스트의 비밀 파일을 변경해야 합니다. 비밀 파일은 한 사이트에서 만들어서 그 사본을 관련된 모든 호스트로 배포할 수 있습니다.

---

**참고**

여기에서 논의된 비밀은 SD-UX 객체에 아무런 액세스 권한도 부여하지 않지만 호스트가 SD-UX 작업에 참여할 수 있도록 허용합니다.

---

---

## RPC 권한

이 절에서는 에이전트가 컨트롤러 요청, 로컬 슈퍼유저 권한, 저장소 등록 및 데몬/에이전트 보안을 처리하는 방법에 대해 설명합니다.

SD-UX에서 객체는 ACL에 의해 보호됩니다. ACL은 객체에 첨부된 구조로서 여러 사용자와 그룹에 대한 액세스 권한을 정의합니다. ACL은 개별 사용자와 그룹이 아닌 다수의 사용자와 그룹에 대해 액세스 권한을 지정하고, 개별 파일이 아닌 전체 SD-UX 객체를 보호하는 두 가지 방법을 통해 HP-UX 파일 시스템의 모드 비트에 의해 정의된 개념을 확장합니다.

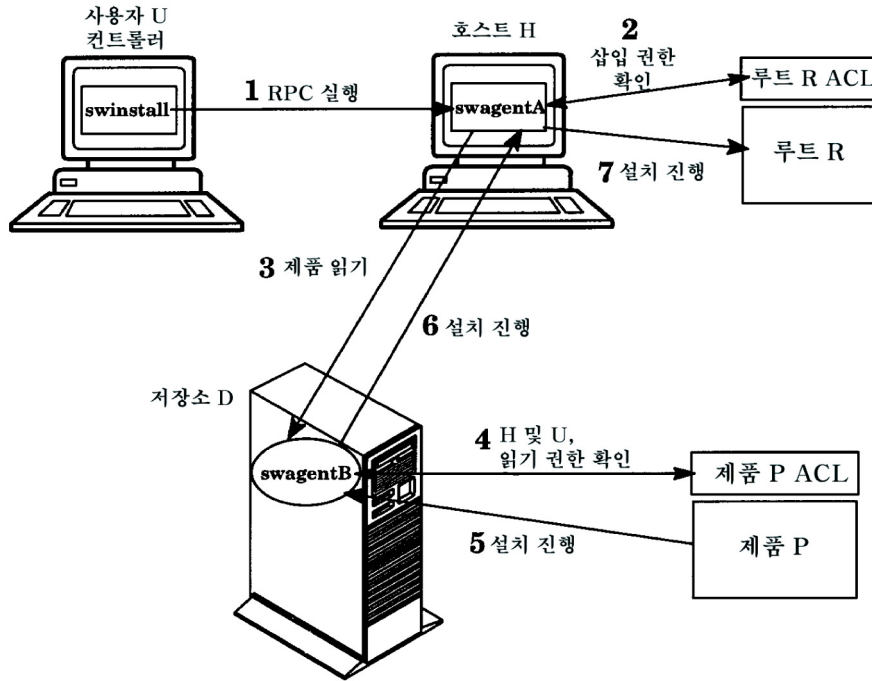
일반적으로 컨트롤러는 객체에 대한 특정 작업을 수행하도록 에이전트에 요청합니다. SD-UX는 ACL을 통해 각 호스트, 저장소, 저장소 제품 및 설치 객체(루트)를 보호합니다. 호출이 인증되면 ACL 관리자는 작업을 허락하기 전에 보호된 객체에 대한 호출자의 액세스 권한을 검토합니다.

SD-UX 권한은 ACL을 사용하여 RPC 호출자가 특정 방식(즉, 읽기, 쓰기)으로 특정 SD-UX 객체에 액세스할 수 있는 권한을 결정합니다. 객체의 ACL을 검색하여 호출자와 일치하는 항목이 있는지 확인합니다. 일치하는 항목을 찾으면 그 항목에 부여된 권한과 작업에 필요한 권한을 비교합니다. 작업에 필요한 권한이 해당 항목에 의해 모두 부여되면 액세스가 허용되고 SD-UX는 요청된 작업을 수행합니다.

## 에이전트의 컨트롤러 요청 처리 방법

컨트롤러가 에이전트에게 다른 에이전트의 참여가 필요한 작업을 수행하도록 요청하면 두 에이전트 모두 각자 자신의 제어 하에 있는 객체에 대한 액세스 권한을 부여해야 작업을 완료할 수 있습니다.

그림 9-3 SD-UX 보안 프로세스



예를 들어 저장소 D에서 루트 R로 제품 P를 설치하는 경우 다음과 같습니다.

1. 사용자 U는 대상 호스트 H에 있는 swagentA로 RPC를 보냅니다. 사용자 U는 대상 호스트에 있는 루트 R에 제품을 설치하고자 합니다.
2. SwagentA는 루트 R을 보호하는 ACL을 점검하여 사용자 U에게 제품을 삽입할 수 있는 권한이 있는지 확인합니다.
3. swagentA(주체 H로 실행)는 swagentB(저장소 D가 있는 위치에서 실행)로 제품을 읽기 위한 요청을 합니다.
4. swagentB는 제품을 보호하는 ACL을 점검하여 요청을 받아들이기 전에 대상 시스템(주체 H)과 사용자 U가 모두 읽기 권한을 가지고 있는지 확인한 다음 설치를 진행합니다.

swagentB의 ACL은 사용자 U를 알지도 못하고 사용자 U에 의존하지도 않습니다. 루트 R에 있는 ACL이 화면 U에 대해 수행된 다음에만 제품 ACL이 화면 H에 대해 수행됩니다.

특별한 경우로, 슈퍼유저는 항상 로컬 시스템에 대해 모든 권한을 갖습니다.

## 로컬 슈퍼유저 권한 인증

특별한 경우로, SD-UX는 항상 로컬 슈퍼유저에게 ACL 보호와 상관없이 모든 로컬 객체에 대한 모든 액세스 권한을 허용합니다. 이를 통해 로컬 슈퍼유저는 손상된 ACL을 복구하거나 다른 작업을 수행할 수 있습니다.

## 위임

SD-UX는 저장소의 제품에 대한 액세스를 제어하기 위해 위임 형식을 제공합니다. 대상 에이전트가 실행 중인 호스트와 호출을 시작하는 사용자 모두에게 읽기 액세스 권한이 있어야 합니다.

이러한 위임 형식은 호출자의 자격 증명 정보를 RPC 옵션의 저장소 에이전트로 전달합니다. 위임 형식은 에이전트가 DCE 또는 SD-UX 내부 권한 인증 중 어느 것을 사용하도록 구성되어 있는지에 상관없이 동일하게 작동합니다.

이러한 위임 기법을 통해 저장소에 있는 제품에 대한 사용자 수준의 액세스가 가능해진다는 점이 중요합니다.

## 저장소 등록 및 데몬/에이전트 보안

SD-UX는 객체를 파일 시스템에 저장하므로 누군가 소프트웨어 저장소에 대한 "트로이 목마" 파일 시스템 이미지를 만들 수 있습니다. 이로 인해 잘못된 저장소에서 제품을 설치한 시스템의 경우 보안이 위협받을 수 있습니다. 이러한 상황으로부터 시스템을 보호하기 위해 저장소로부터 소프트웨어를 설치하거나 복사하기 전에 SD-UX를 사용하여(swcopy 또는 swreg를 사용하여) 저장소를 등록해야 합니다. 액세스 권한을 부여하기 전에 이러한 점검은 항상 수행됩니다. swreg를 사용하여 등록하는 경우 호스트 ACL에 대한 삽입 권한이 필요합니다.

특별한 경우로 시작 프로그램이 로컬 슈퍼유저이거나 저장소 등록 권한(호스트에서 삽입 권한)을 가지고 있는 경우 로컬 설치(즉, 저장소와 대상 루트가 같은 시스템에 있음)에 등록되지 않은 저장소가 사용될 수 있습니다.

호스트 시스템 관리자는 새로운 저장소를 등록하기 전에 저장소의 무결성을 확인하고 신뢰할 수 있는 사용자에게만 호스트에서 삽입 권한을 부여할 수 있도록 해야 합니다.

---

**참고**

사용자를 등록하는 경우와 마찬가지로 등록되지 않은 저장소로부터 설치하거나 복사하는 경우에도 신중해야 합니다.

---



---

## 보안 사용 모델

아래의 사용 모델은 SD-UX 설치 시 설치된 기본 호스트 ACL에 제공된 swadm 그룹을 사용합니다. 이 그룹은 기본 HP-UX 구성에 포함되지는 않지만 쉽게 추가할 수 있습니다. 먼저 HP-UX System Administration Manager 제품을 사용하여 swadm 그룹과 적절한 그룹 구성원을 추가합니다. 그런 다음 /etc/logingroup 링크를 /etc/group에 추가하여 HP-UX 보완 그룹을 활성화합니다.

---

### 참고

/etc/logingroup은 SVR2/3과 BSD 그룹의 구문을 선택적으로 지원하는 HP-UX 유틸리티입니다. /etc/logingroup이 /etc/group에 링크되면 HP-UX가 BSD(및 SVR4) 구문을 제공합니다.

/etc/logingroup 파일이 SD-UX 컨트롤러로 대상 지정이 된 시스템에 없는 경우 각 해당 시스템에서 (수퍼유저로) 다음 명령을 실행합니다.

```
ln -s /etc/group /etc/logingroup
```

## 원격 배포 시 보안

SD-UX 원격 작업은 일반적으로 관리자가 로컬 저장소에서 여러 원격 대상으로 소프트웨어를 **넣을(push)** 때 사용합니다.

다음과 같은 종류의 구성을 설정할 수 있습니다.

1. 위에서 설명한 대로 컨트롤러 호스트에 swadm 그룹을 설정합니다.
2. 각 대상 시스템에서 세 가지 호스트 ACL을 편집합니다. 177페이지의 “원격 작업 설정”에 설명된 설정을 사용하여 대상 시스템에 에이전트를 설치한 경우, 설정을 수행한 시스템에서 수퍼유저 권한으로 대상에 대한 세 가지 호스트 ACL을 편집할 수 있습니다.

```
swacl -l host \
  -M group:swadm@`hostname`:a @ remsys1. . .remsysN
swacl -l global_soc_template \
  -M group:swadm@`hostname`:a @ remsys1. . .remsysN
swacl -l global_product_template \
  -M group:swadm@`hostname`:a @ remsys1. . .remsysN
```

기본 저장소에서 특정 제품을 관리할 수 있는 권한을 특정 사용자에게 부여하려는 경우가 있습니다. 예를 들어 사용자 ramon은 저장소에서 ALLBASE 제품을 관리할 책임, 즉 사용할 수 있을 때 새로운 버전과 패치를 설치하는 책임을 맡을 수 있습니다. ramon을 로컬 저장소의 ALLBASE에 대한 ACL에 추가하고 이 제품에 대한 모든 권한을 부여하려면 다음 명령을 실행합니다.

```
swacl -l product -M user:ramon:a ALLBASE
```

동시에 같은 제품에 대해 swadm 그룹을 ACL 항목에서 제거하려면 다음 명령을 수행합니다.

```
swacl -l product -D group:swadm ALLBASE
```

## 로컬 배포 시 보안

호스트 관리자는 로컬 호스트에서 신뢰할 수 있는 개별 사용자나 그룹에 소프트웨어를 로컬로 관리할 수 있는 권한을 부여할 수 있습니다. 신뢰할 수 있는 로컬 사용자는 삽입 및 쓰기 권한을 부여하는 루트 ACL 항목을 가지고 있습니다. 소스 저장소에서 모든 소프트웨어 제품에 대한 액세스는 호스트, 저장소 및 제품에 대한 무제한의 읽기 액세스에 의해 허용됩니다. 이것이 소프트웨어 배포의 가져오기(pull) 모델의 기본입니다.

### 특정 사용자에 의한 특정 대상 시스템으로 설치 제한

소프트웨어 소스 저장소의 관리자는 위에서 설명된 대로 개방적으로 소프트웨어를 설치할 수 있도록 할 수도 있고 특정 시스템으로 배포를 제한할 수도 있습니다. 소스 저장소 제품을 보호하는 ACL에는 제품 읽기 액세스 권한을 지정된 시스템에만 제한하고 이들 시스템에만 설치를 허용하는 항목이 있을 수 있습니다. 이 제한은 넣기(push)와 가져오기(pull) 모델에 모두 적용됩니다.

다음은 systemA와 systemB로 읽기 권한을 제한하고 사용자 swadm에 모든 권한을 부여하는 제품 ACL 예입니다.

```
user:swadm:rwict
host:systemA.loc.company.com:r
host:systemB.fc.hp.com:r
```

## 소프트웨어 개발자를 위한 보안

소프트웨어 개발자는 제품을 배포하기 전에 반복적으로 제품을 패키징하고 테스트합니다. 여기에는 저장소로 제품을 패키징하고 테스트를 위해 루트에 설치하는 작업이 포함됩니다. 사용자 조정이 올바르게 수행될 때까지 여러 차례 반복 작업이 필요하므로 이러한 테스트를 위해 소프트웨어 개발자들의 저장소와 루트에 대한 자유로운 액세스를 막는 것은 적절하지 않습니다.

또한 광범위하게 사용되는 저장소와 루트에 들어오고 나가고 테스트되는 제품을 두지 말아야 합니다. 준비되기 전에 실수로 설치하거나 사용할 수 있습니다.

권장되는 개발 방법은 테스트용으로 개발용 저장소와 루트를 하나 이상 확보하고 이를 사용하는 개발 그룹의 요구를 충족시킬 수 있도록 각각에 맞는 보호 조치를 취하는 것입니다. 이러한 목적으로는 제품이 들어오고 나가는 속도가 짧기 때문에 앞서 설명된 기본 **ACL** 템플릿 메커니즘을 사용하는 것이 편리합니다.

호스트 관리자(호스트에서 삽입 권한을 가지고 있는 사용자)가 개발자를 위한 테스트 저장소를 작성한 다음 저장소 관리자를 지정하고 저장소 **ACL**을 편집하여 저장소 관리자에게 해당 저장소에서의 제어(**ACL** 편집) 권한을 부여해야 합니다. 그런 다음 저장소의 제품 **ACL** 템플릿을 설정하여 제품을 삽입하는 사용자가 쓸(변경 및 삭제할) 수도 있고 이미 알려진 테스트 시스템에 의해서만 읽힐 수 있도록 해야 합니다.

마찬가지로 테스트 루트도 작성할 수 있는데, 테스트 루트는 개발자들이 테스트 제품을 설치할 수 있는 다른 테스트 호스트에 작성하는 경우가 많습니다. 테스트 루트에 설치할 수 있는 액세스 권한은 개발 그룹으로 제한해야 합니다.

테스트 및 제품 발표 준비가 완료되면, 제품을 일반 배포용 저장소로 복사하여 보다 널리 읽힐 수 있도록 하고 테스트 저장소에 있는 테스트가 완료되지 않은 제품은 노출되지 않도록 합니다.

제품 개발 시 원하는 보안 정책을 구현하기 위해 이러한 기본적인 개념이 사용될 수 있는 방법은 여러 가지가 있습니다.

---

## 명령별 권한 요구 사항

### 패키징(`swpackage`)

- 저장소가 없는 경우 `swpackage`는 사용자가 대상 호스트에 대해 삽입 권한을 가지고 있는지 확인합니다.
- `swpackage`는 사용자가 대상 저장소에 대해 삽입 권한을 가지고 있는지 확인합니다.
- `swpackage`는 제품이 이미 존재하는 경우 사용자가 대상 제품에 대해 쓰기 권한을 가지고 있는지 확인합니다.

### 나열(`swlist`)

- 가능성 있는 저장소를 나열하기 위해 소스 에이전트는 컨트롤러 사용자가 호스트에 대해 읽기 권한을 가지고 있는지 확인합니다.
- 가능성 있는 제품을 나열하기 위해 소스 에이전트는 컨트롤러 사용자가 저장소나 루트에 대해 읽기 권한을 가지고 있는지 확인합니다.

### 작업 찾아보기(`sd`, `swjob`)

- CLI(`swjob`) 또는 GUI(`sd`)를 사용하여 로컬 호스트에서 시작된 작업에 대한 정보를 보기 위해 컨트롤러는 사용자가 호스트에 대해 읽기 권한을 가지고 있는지 확인합니다.
- 명령줄 또는 GUI를 사용하여 대상 로그 파일을 검색하려면 대상 에이전트는 컨트롤러 사용자가 루트나 저장소 대상에 대해 읽기 액세스 권한을 가지고 있는지 확인합니다.

### 복사(`swcopy`)

- 이 기능을 지원하는 데 필요한 나열 작업은 위의 `swlist` 절에서 설명한 대로 점검되어야 합니다.

- 저장소가 없는 경우 `swcopy`는 사용자가 대상 호스트에 대해 삽입 권한을 가지고 있는지 확인합니다.
- 대상 에이전트는 컨트롤러 사용자가 대상 저장소에 대해 삽입 권한을 가지고 있는지 확인합니다.
- 제품이 이미 존재하는 경우 대상 에이전트는 컨트롤러 사용자가 대상 제품에 대해 쓰기 권한을 가지고 있는지 확인합니다.
- 소스 에이전트는 대상 에이전트 시스템이 소스 제품에 대해 읽기 권한을 가지고 있는지 확인합니다.
- 소스(저장소) 에이전트는 저장소가 등록되어 있는지 확인합니다. 등록되어 있지 않으면 에이전트는 컨트롤러 사용자와 대상 에이전트 시스템이 각각 소스 호스트에 대해 삽입 권한을 가지고 있는지 확인합니다.

### 설치(`swinstall`)

- 이 기능을 지원하는 데 필요한 나열 작업은 위의 `swlist` 절에서 설명한 대로 점검되어야 합니다.
- 대상 에이전트는 컨트롤러 사용자가 대상 루트에 대해 삽입 권한을 가지고 있는지 확인합니다.
- 제품이 이미 존재하는 경우 대상 에이전트는 컨트롤러 사용자가 대상 루트에 대해 쓰기 권한을 가지고 있는지 확인합니다.
- 소스(저장소) 에이전트는 대상 에이전트 시스템이 소스 제품에 대해 읽기 권한을 가지고 있는지 확인합니다.

### 제거(`swremove`)

- 객체가 저장소의 제품인 경우 대상 에이전트는 컨트롤러 사용자가 대상 제품에 대해 쓰기 권한을 가지고 있는지 확인합니다.
- 객체가 루트의 제품인 경우 대상 에이전트는 컨트롤러 사용자가 대상 루트에 대해 쓰기 권한을 가지고 있는지 확인합니다.
- 객체가 저장소나 루트 또는 이들 중 어느 하나에 들어 있는 마지막 제품인 경우 해당 컨테이너를 제거하기 전에 대상 에이전트는 컨트롤러 사용자가 대상 루트나 저장소에 대해 삭제 권한을 가지고 있는지 확인해야 합니다.

## 구성(swconfig)

- 위의 **swremove** 작업에 대해 동일한 권한 점검이 이루어지며 예외적으로 이 명령은 저장소에는 적용되지 않습니다.

## 확인(swverify)

- 객체가 저장소에 있는 제품인 경우 대상 에이전트는 컨트롤러 사용자가 대상 제품에 대해 읽기 권한을 가지고 있는지 확인합니다.
- 객체가 루트에 있는 제품인 경우 대상 에이전트는 컨트롤러 사용자가 대상 루트에 대해 쓰기 권한을 가지고 있는지 확인합니다(스크립트가 실행된 이후).

## 저장소 등록(swreg)

- 새로운 저장소를 등록하려면 대상 데몬은 등록할 저장소에서 읽기 권한을 확인하고 호스트에서 삽입 권한을 확인합니다.

## ACL 변경(swacl)

- ACL을 변경하려면 쓰기 권한이 필요합니다.
- ACL을 나열하려면 나열 권한이 필요합니다.

## Request 스크립트(swask)

- 사용자를 조회하고 설치 정보를 얻기 위해 대화식 제어 스크립트를 사용합니다.

## 변경(swmodify)

- IPD(Installed Products Database)나 저장소 카탈로그 파일의 정보를 변경하거나 추가하려면 쓰기 권한이 필요합니다.

# 10 소프트웨어 패키지 만들기

이 장에서는 배포용 소프트웨어 패키지를 만드는 것과 관련된 작업에 대해 설명합니다.

표 10-1

이 장의 항목

설명하는 항목
280페이지의 “패키징 프로세스 개요”
281페이지의 “패키징할 제품 식별”
283페이지의 “제어 스크립트 추가”
285페이지의 “PSF(Product Specification File) 만들기”
324페이지의 “소프트웨어 패키징(swpackage)”
335페이지의 “패키징 작업 및 예”

## 패키징 프로세스 개요

저장소에서 소프트웨어를 배포할 때 **Software Distributor**를 사용하여 소프트웨어를 **SD-UX** 형식으로 쉽게 패키징할 수 있습니다. 패키징 프로세스를 통해 저장소를 직접 작성하거나 나중에 저장소에 추가할 수 있는 패키지를 작성할 수 있습니다. 패키징 사양은 매우 유연하므로 다양한 소프트웨어 빌드 및 제조 과정에서의 요구 사항에 맞출 수 있습니다.

패키징 프로세스는 다음과 같은 작업으로 이루어집니다.

1. 패키지를 식별합니다.

소프트웨어에 어떤 파일과 디렉토리를 포함시킬 것인지를 결정하고 제품 구조를 결정합니다. 소프트웨어 패키지에는 파일, 파일 세트, 하위 제품, 제품 및 번들이 포함될 수 있습니다.

2. 제어 스크립트를 작성합니다(선택).

제어 스크립트를 작성하여 패키지에 포함시킬 수 있습니다. 이 스크립트를 통해 **SD-UX**에서 지원되는 것 이외의 추가 검사 및 작업을 수행할 수 있습니다.

3. **Product Specification File(PSF)**을 만들어서 제품 패키지를 정의합니다.

4. `swpackage` 명령을 실행하여 소프트웨어 패키지를 만듭니다.

`swpackage` 명령은 **PSF** 파일을 읽고 제품 정의를 분석하며 소스 파일과 정보를 패키징하여 제품 객체로 만듭니다. 그런 다음 제품을 만들어 배포 저장소에 보관합니다.

## 전제 조건

소프트웨어 패키징을 시작하기 전에 다음을 확인합니다.

- 소프트웨어 패키지를 만들려고 하는 시스템에 **SD-UX**가 설치 및 구성되어 있는지 확인합니다.
- 패키징할 소프트웨어가 패키징 시스템에 설치되어 있는지 또는 필요한 파일을 원격으로 사용할 수 있는지 확인합니다.



## 패키징할 제품 식별

### 제품 내용 결정

소프트웨어 패키징의 첫 번째 단계는 소프트웨어 제품에 어떤 파일과 디렉토리를 포함시킬 것인지 결정하는 것입니다. 이러한 파일과 디렉토리가 사용자가 원하는 구성을 지원하도록 하려면 특정 지침을 따라야 합니다.

이 구조의 핵심은 다음과 같습니다.

- 공유 가능 파일(예: 실행 파일)과 공유 불가 파일(예: 구성 파일)이 어디에 설치되어 있는가?
- 공유 불가 파일을 올바른 위치에 두기 위해 구성이 어떻게 사용되는가?

### 제품 구조 결정

소프트웨어가 따라야 하는 제품 구조를 결정합니다. **SD-UX**에서는 다음과 같은 4가지 수준의 소프트웨어 객체를 제공합니다.

수준	객체
파일 세트	(필수) 파일 세트에는 실제 제품 파일, 그러한 파일을 설명하는 정보(속성) 및 이 파일 세트가 설치, 복사 또는 제거되기 전과 후 또는 그 동안에 실행되는 별도의 제어 스크립트가 포함됩니다. 파일 세트는 관리 가능한 가장 작은(선택 가능한) 소프트웨어 객체입니다. 파일은 하나 이상의 파일 세트로 그룹화되어야 합니다. 또 파일 세트는 하나 이상의 제품으로 그룹화되어야 합니다. (파일 세트는 단 하나의 제품만을 구성할 수도 있습니다.)
하위 제품	(선택) 하위 제품은 제품에 여러 개의 파일 세트가 포함되어 있는 경우 제품 내에서 관련된 파일 세트를 그룹화하는 데 사용됩니다. 하위 제품 정의는 선택입니다.
제품	(필수) 파일 세트(및/또는 하위 제품)는 하나 이상의 제품으로 그룹화되어야 합니다. 대개 관련된 소프트웨어 세트를 구성하거나 고객이 구매하는 제품과 일치하는 컬렉션으로 그룹화됩니다. <b>SD-UX</b> 명령으로 제품의 핵심적인 부분을 관리하는 한편 하위 제품이나 파일 세트를 통해 제품의 하위 집합을 관리할 수 있는 유연성도 제공합니다.

**변들**

(선택) 변들은 **HP** 제조시에만 제공되는 것입니다. 고객의 변들 패키징은 지원되지 않습니다.

---

**참고**

제품 자체의 다른 버전(릴리즈)뿐만 아니라 다른 플랫폼과 운영 체제에 대해서도 다른 버전의 제품을 정의할 수 있습니다. 동일한 배포 미디어에 다른 제품 버전을 포함시킬 수 있습니다.

---

## 제어 스크립트 추가

SD-UX는 다른 HP-UX 명령 및 기능을 통한 추가 검사 및 작업을 가능하게 하는 제품 및 파일 세트 제어 스크립트의 실행을 지원합니다. `swask`, `swinstall`, `swconfig`, `swverify` 및 `swremove` 명령은 각기 기본 루트에서 제어 스크립트를 하나 이상 실행시킬 수 있습니다. 스크립트를 작성하여 이를 소프트웨어 패키지에 포함시킬 수 있습니다. 스크립트는 모두 선택적이지만 소프트웨어 패키지가 수행할 작업을 올바르게 완료할 수 있도록 스크립트를 작성해야 하는 경우가 많습니다. 제어 스크립트에 대한 자세한 설명은 11장, 347페이지의 “제어 스크립트 사용”을 참조하십시오.

SD-UX는 다음과 같은 종류의 스크립트를 지원하며, 이러한 스크립트는 제품과 파일 세트에 대해 정의할 수 있습니다.

Checkinstall	설치 및 구성이 실행될 수 있는지를 판별하기 위해 각 대상을 분석합니다( <code>swinstall</code> 명령으로 실행).
Checkremove	제거 및 구성 해제가 수행될 수 있는지 판별하기 위해 각 대상을 분석합니다( <code>swremove</code> 명령으로 실행).
Configure	설치되어 있는 파일 세트나 제품을 구성합니다( <code>swconfig</code> 및 <code>swinstall</code> 명령으로 실행).
Fix	설치된 소프트웨어의 문제점을 수정하고 보고합니다( <code>swverify</code> 명령으로 실행).
Postinstall	파일 세트나 제품이 설치된 직후 기본 파일 재설정과 같은 추가적인 설치 작업을 수행합니다( <code>swinstall</code> 명령으로 실행).
Postremove	파일 세트나 제품이 제거된 직후에 추가적인 제거 작업(“롤백”된 파일을 복원하는 등)을 수행합니다( <code>swremove</code> 명령으로 실행).
Preinstall	소프트웨어 파일을 설치하기 직전에 오래된 파일 제거와 같은 파일 작업을 수행합니다( <code>swinstall</code> 명령으로 실행).
Preremove	소프트웨어 파일을 제거하기 직전에 설치 전 스크립트에 의해 작성된 파일을 제거하는 등 추가적인 파일 작업을 수행합니다( <code>swremove</code> 명령으로 실행).

Request	설치 또는 구성 프로세스의 일부로 사용자로부터의 대화식 응답을 요구합니다( <code>swask</code> , <code>swconfig</code> 및 <code>swinstall</code> 명령으로 실행).
Unconfigure	구성 스크립트로 수행한 구성을 취소합니다( <code>swconfig</code> 및 <code>swremove</code> 명령으로 실행).
Unpostinstall	설치 프로세스 중에 <code>swinstall</code> 이 복구 작업을 시작해야 하는 경우 설치 후 스크립트를 취소합니다( <code>swinstall</code> 명령으로 실행).
Unpreinstall	설치 프로세스 중에 <b>SD</b> 가 복구를 시작해야 하는 경우 설치 전 스크립트를 취소합니다( <code>swinstall</code> 명령으로 실행).
Verify	일반적인 <code>swverify</code> 확인에 추가하여 파일 세트 또는 제품의 구성을 확인합니다( <code>swverify</code> 명령으로 실행).

---

## PSF(Product Specification File) 만들기

SD-UX는 PSF(Product Specification File)를 사용하여 실제 제품 패키지를 정의합니다. PSF는 제품의 속성, 내용, 호환성, 종속성 및 설명에 따라 제품을 식별하는 “로드맵”을 제공합니다. PSF는 swpackage 세션을 수행합니다. 제품 구조를 설명하고 여기에 적용되는 속성을 정의합니다.

SD-UX는 파일을 패키징하고 배포하며 설치합니다. SD-UX 패키징 프로그램에서는 이 파일들이 특정 디렉토리 위치에 빌드되고 설치된 후에 이 파일들을 사용합니다. 이러한 디렉토리 위치는 별도의 연결되지 않은 디렉토리 트리에 있을 수도 있고, 시스템 상에서 소프트웨어가 실행되도록 하기 위해 필요한 특정 파일 위치에 있을 수도 있습니다. 루트 디렉토리(그 아래에 모든 파일을 보관)나 명시적인 개별 파일 경로를 통해 파일을 지정할 수 있습니다. 파일 속성은 파일 자체에서 가져오거나 각 파일에 대해 별도로 또는 파일 세트에 대해 지정할 수 있습니다.

PSF는 다음과 같은 기능을 제공합니다.

- 제품 그룹(모든 제품 포함) 및 개별 제품에 대한 공급업체 정보를 정의합니다(선택).
- 하나 이상의 제품을 지정합니다(필수).
- 각 제품에서 하나 이상의 하위 제품(선택), 파일 세트(필수) 및 파일(필수)에 대한 속성을 정의합니다.
- 배포 저장소/미디어에 대한 속성을 정의합니다(선택).
- 제품이 지원하는 컴퓨터와 운영 체제를 지정합니다.
- 소프트웨어 객체를 설명하는 속성을 정의합니다.

## PSF(Product Specification File)의 예

### 최소 PSF

다음은 필수 키워드만 포함된 최소 PSF의 예입니다. 이 PSF는 `commands` 파일 세트를 가지는 SD라는 제품을 만들고 한 개의 파일 `/usr/sbin/swcopy`를 포함합니다.

```
product
  tag    SD
fileset
  tag    commands
file    swcopy /usr/sbin/swcopy
```

---

### 참고

이 최소 형식에서 두 번째 `file` 용어에 대해서는 절대 경로를 사용해야 합니다.

---

### 일반적인 PSF

다음은 SD-UX 제품을 설명하는 샘플 PSF입니다.

```
# PSF defining SD as a sample product.
depot
  layout_version 1.0
# Vendor definition:
vendor
  tag            HP
  title          Hewlett-Packard Company
  description    < data/description.hp
category
  tag            system_mgt
  title          Systems Management Applications
  description    These are the system management applications
  revision       1.0
end
# Product definition:
product
  tag            SD
  revision       A.01.00
  architecture   HP-UX_B.11_32/64
```

```
vendor_tag      HP
is_patch        false
title           HP-UX Distributor
number          B2000A
category_tag    system_mgt
description     < data/descr.sd
copyright       < data/copyr.sd
readme          < data/README.sd
machine_type    *
os_name         HP-UX
os_release      ?.11.*
os_version      ?
directory       /
is_locatable    false
# Specify a checkremove script that executes during the
# swremove analysis phase. (This script prevents the
# removal of the SD product and returns an ERROR.
    checkremove    scripts/checkremove.sd

# Subproduct definitions:
subproduct
    tag            Manager
    title          Management Utilities
    contents       commands agent data man
end
subproduct
    tag            Agent
    title          Agent component
    contents       agent data man
end
# Fileset definitions:
fileset
    tag            commands
    title          Commands (management utilities)
    revision       2.42
    description    < data/descr.commands
# Dependencies
```

```

        corequisites SD.data
        corequisites SD.agent
# Control files:
        configure      scripts/configure.commands
# Files:

        directory     ./commands=/usr/sbin
        file           swinstall
        file           swcopy
# (...Other file definitions can go here...)
        directory     ./nls=/usr/lib/nls/C
        file           swinstall.cat
        file           swpackage.cat

        directory     ./ui=/var/adm/sw/ui
        file           *
# (...Other file definitions can go here...)
end
# Commands
# (...Other fileset definitions can go here...)
# Manpage fileset definitions:
fileset
    tag              man
    title            Manual pages for the SD-UX
    revision         2.05
    directory        ./man/man1m=/usr/man/man1m.Z
    file             *
    directory        ./man/man4=/usr/man/man4.Z
    file             *
    directory        ./man/man5=/usr/man/man5.Z
    file             *
end
#man
end
#SD

```



## PSF 구문

SD-UX 객체(제품, 하위 제품, 파일 세트 및 파일)는 각각의 **속성** 세트를 가지고 있고 각 속성에는 해당 속성을 정의하는 **키워드**가 있습니다. 대부분의 속성은 선택이며 PSF에 속성을 모두 지정할 필요는 없습니다. 속성은 각각의 특정 요구 사항을 갖지만 다음과 같은 규칙이 적용됩니다.

- 키워드 구문은 다음과 같습니다.

`keyword value`

- 모든 키워드는 별도로 명시된 경우를 제외하고는 값을 하나 이상 필요로 합니다. 키워드는 있지만 값이 없는 경우 경고 메시지가 생성되고 그 키워드는 무시됩니다.
- 별도의 줄에 주석을 달 수도 있고 키워드-값 구문 다음에 주석을 붙일 수도 있습니다. 주석 줄 앞에는 #이 표시됩니다.
- 여러 행에 이어지는 값을 정의할 때에는(예를 들면 설명) 인용 부호를 사용합니다. 공백이 들어 있는 한 줄로 값을 정의할 때에는 인용 부호가 필요 없습니다.
- PSF를 읽을 때 오류가 발생하면 `swpackage` 명령이 중단됩니다. 오류는 `stderr`와 로그 파일에 모두 기록됩니다.

## PSF 객체 구문

다음 표와 절에서는 PSF 키워드와 각 키워드에 사용할 수 있는 값, PSF에 정의할 수 있는 객체에 대한 구문 등이 나와 있습니다.

- +가 표시된 키워드는 제품에만 적용됩니다.
- -가 표시된 키워드는 번들에만 적용됩니다.
- \*가 표시된 키워드는 표에 나와 있는 유형인 동시에 `version_component` 유형이기도 합니다.

표 10-2 PSF에 사용되는 키워드

키워드	값	최대 크기 (바이트)	예제
배포 클래스			
distribution layout_version tag copyright description number title	revision_string tag_string multi_line_string multi_line_string one_line_string one_line_string	64 64 8K 8K 256 256	1.0 EXAMPLE_DEPOT < data/copyr.depot data/descr.depot B2358-13601 Example packages
공급업체 클래스			
vendor tag description title	tag_string multi_line_string one_line_string	64 8K 256	HP <data/desc.hp HP Company
범주 클래스			
category tag description revision title	tag_string multi_line_string revision_string one_line_string	64 8K 64 256	patch_normal Normal problems 0.0 Category of Patches

표 10-2 PSF에 사용되는 키워드(계속)

키워드	값	최대 크기 (바이트)	예제
제품 클래스			
product or bundle			
* tag	tag_string	64	SD-UX
* architecture	one_line_string	256	HP-UX_B.11.11_32/64
category_tag	one_line_string	256	Systems Management
- contents	repeatable list of software specs	없음	pr.fs,r=1.0,a=,v=
copyright	multi_line_string	8K	<data/copyr.sd
description	multi_line_string	8K	<data/descr.sd
directory	path_string	255/102	/
is_locatable	boolean	9	false
is_patch	boolean	9	false
machine_type	uname_string	64	9000/800
number	one_line_string	256	B2000A
os_name	uname_string	64	HP-UX
os_release	uname_string	64	? .11.*
os_version	uname_string	64	A
+ postkernel	path_string	255/1024	/usr/bin/kern_bld
+ readme	multi_line_string	8K	<data/README.sd
+ revision	revision_string	64	A.01.00
+ share_link	one_line_string	256	
title	one_line_string	64	Software Distributor
* vendor_tag	tag_string		HP
하위 제품 클래스			
subproduct			
tag	tag_string	64	Manager
contents	one-line list of tag string values	없음	commands agent data man
description	multi_line_string	8K	<data/desc.mgr
title	one_line_string	256	Management Utilities

표 10-2 PSF에 사용되는 키워드(계속)

키워드	값	최대 크기 (바이트)	예제
파일 세트 클래스			
fileset			
* tag	tag_string	64	commands
ancestor	repeatable list of product.fileset	없음	prod.oldfileset
architecture	revision_string	64	HP-UX_B.11.11_32/64
category_tag	tag_string	64	patch_normal
corequisite	software_spec	없음	SD-UX.man.r>=2.0
description	multi_line_string	8K	<data/descr.cmd
exrequisite	software_spec	없음	SD-UX.data,R>=2.1
is_kernel	boolean	9	false
is_patch	boolean	9	false
dynamic_module	one_line_string	256	ipf pfil
is_reboot	boolean	9	false
is_sparse	boolean	64	false-
machine_type	uname_string	64	9000/8*
os_name	uname_string	64	HP-UX
os_release	uname_string	64	? .11.*
os_version	uname_string	64	A
prerequisite	software_spec	없음	SD-UX.agent,r>=2.0
* revision	revision_string	64	2.42
supersedes	software_spec	없음	product.fileset
title	one_line_string	256	SD-UX Commands
클래스			
control_files			
directory	path_mapping_string	없음	./commands=/usr/sbin
file_permissions	permission_string	없음	-u 0222 -o root -g sys
file	file_specification	없음	-m 04555 bin/swinstall (or) *

표 10-3 제어 파일 속성

키워드	유형	크기 (바이트)	예제
checkinstall	path_string	1K	./scripts/checkinstall
checkremove	path_string	1K	./scripts/checkremove
configure	path_string	1K	./scripts/configure
control_file	path_string	1K	./scripts/subscripts
fix	path_string	1K	./scripts/fix
postinstall	path_string	1K	./scripts/postinstall
postremove	path_string	1K	./scripts/postremove
preinstall	path_string	1K	./scripts/preinstall
preremove	path_string	1K	./scripts/preremove
request	path_string	1K	./scripts/request
unconfigure	path_string	1K	./scripts/unconfigure
unpreinstall	path_string	1K	./scripts/unpreinstall
unpostinstall	path_string	1K	./scripts/unpostinstall
verify	path_string	1K	./scripts/verify

제어 파일 SD-UX는 제품 및 파일 세트 수준에서 제어 파일(제어 스크립트라고도 함)의 실행을 지원합니다. 제어 스크립트를 통해 추가적인 검사 및 작업을 수행할 수 있습니다. swinstall, swconfig, swverify 및 swremove 명령은 각각 공급업체에서 제공하는 하나 이상의 스크립트를 실행합니다. 스크립트는 모두 선택적이지만 소프트웨어 패키지가 수행할 작업을 올바르게 완료할 수 있도록 스크립트를 작성해야 하는 경우가 많습니다. 제어 스크립트에 대한 자세한 설명은 11장, 347페이지의 “제어 스크립트 사용”을 참조하십시오.

### PSF 레이아웃 버전 선택

PSF의 저장소 정의에서 레이아웃 버전을 선택하거나(299페이지의 “PSF(Product Specification File) 의미 분석” 참조) swpackage, swmodify, swcopy 또는 swlist에 layout\_version 옵션을 지정합니다.

PSF 구문은 IEEE 표준 1387.2: Software Administration(POSIX)의 layout\_version=1.0을 준수합니다. SD 이전 버전에서는 POSIX layout\_version=0.8 구문을 지원했는데, 이 구문은 계속 지원됩니다.

소프트웨어 저장소의 레이아웃 버전은 함께 사용할 수 없고 두 가지 중 어느 한 가지 버전이어야 합니다.

두 버전 간의 차이점은 다음과 같습니다.

- 공급업체 사양이 다르게 처리됩니다.

현재 표준(layout\_version=1.0)의 경우, 각 공급업체 클래스 정의는 공급업체 클래스 정의 내 tag 속성과 일치하는 vendor\_tag 속성을 포함하는 후속 제품이나 번들에만 연관됩니다.

이전 표준(layout\_version=0.8)의 경우나 layout\_version을 지정하지 않은 경우, 제품이나 번들은 배포 수준에서 사용자가 마지막으로 정의한 공급업체 클래스에 자동 연관되거나 제품이나 번들 내에서 정의한 공급업체의 공급업체 클래스에 자동 연관됩니다. 값의 유무에 상관없이 명시적으로 정의된 vendor\_tag 속성이 우선합니다.

- layout\_version=0.8에서 필수 조건과 전제 조건은 별도의 제목을 갖습니다 (즉, 필수 조건과 전제 조건). 자세한 내용은 314페이지의 “중속성 사양”을 참조하십시오.
- 범주 객체와 키워드가 다르게 처리됩니다.

layout\_version=1.0(현재 표준):

- category\_tag는 category와 category\_title 속성을 대체하는 유효한 제품 속성입니다.
- category 클래스 객체를 정의할 수 있습니다.

layout\_version=0.8(이전 표준):

- category와 category\_title은 category\_tag 속성을 대체하는 유효한 제품 속성입니다.
- category 클래스 객체는 인식되지 않습니다.

layout\_version=0.8에서 PSF 요구 사항에 대한 자세한 설명은 HP-UX 이전 버전의 *swpackage.4* 설명서 페이지를 참조하십시오.

## PSF 값의 유형

공급업체 정의 속성(299페이지의 “공급업체 정의 속성” 참조)을 제외하고 PSF의 각 속성 키워드 값은 아래 설명되는 유형 중 하나와 일치해야 합니다.

---

### 참고

PSF 구문은 POSIX 1387.2 Software Administration 표준의 `layout_version=1.0`을 준수합니다. SD-UX 이전 버전에서는 POSIX `layout_version=0.8` 구문을 지원했는데, 이 구문은 계속 지원됩니다. 자세한 내용은 293페이지의 “PSF 레이아웃 버전 선택”을 참조하십시오.

---

`boolean`

- 최대 길이: 9바이트
- `true`나 `false` 중 한 가지 값
- 예: `true`, `false`

`file_specification`

- 최대 길이: 없음
- 다음 형식을 사용하여 패키징할 파일이나 디렉토리를 명시적으로 지정합니다.  
`[-m mode] [-o [owner[,]] [uid]] [-g [group[,]] [gid]] [-v][source] [destination]`
- 소스와 대상은 `path_mapping_string`에 지정한 소스 및 대상 디렉토리에 상대적인 경로가 될 수 있습니다.
- `*`를 사용하여 디렉토리 키워드로 지정한 소스 디렉토리 아래에 있는 파일을 모두 포함시킬 수도 있습니다.
- 예: `-m 04555 sbin/swinstall` 또는 `*`(모든 파일과 디렉토리를 의미함)

`multi_line_string`

- 최대 길이: 8Kbyte(`readme` 파일은 1Mbyte)

- 여러 줄로 된 각 문자열은 모든 `isascii` 문자를 지원합니다. (`ctype(3)` 맨 페이지를 참조하십시오.) 하나 이상의 텍스트 단락을 나타냅니다. 인용 부호로 묶어 인라인으로 지정하거나 파일에서 읽도록 할 수 있습니다.

파일 항목은 다음 구문을 사용해야 합니다.

< filename

- 예: </mfg/sd/description

one\_line\_string • 최대 길이: 256바이트

- 한 줄 문자열은 `isascii` 문자의 일부만을 지원합니다. (`ctype(3)` 맨 페이지를 참조하십시오.)
- 공백과 탭을 제외한 어떠한 `isspace` 문자도 허용되지 않습니다.
- 예: Hewlett-Packard Company

path\_mapping\_string

- 최대 길이: 없음
- 값의 형식: `source[=destination]`. 여기서 `source`는 나중에 정의된 파일이 위치하게 될 디렉토리를 정의합니다. `destination`은 선택으로 파일이 실제 설치되는 대상 디렉토리에 소스를 매핑합니다.
- 예: /mfg/sd/files/usr = /usr

path\_string

- 최대 길이: 테이프는 255바이트, 저장소는 1024바이트
- 파일에 대한 절대 또는 상대 경로. 이 유형의 속성은 길이가 대개 255바이트로 제한됩니다. 이러한 제한은 파일의 `basename(1)` <= 100바이트이고 파일의 `dirname(1)` <= 155바이트여야 하는 `tar(1)` 명령으로 인한 것입니다. (`tar` 구현에 따라 <=가 아닌 <를 사용하기도 합니다.)
- 예: /usr /mfg/sd/scripts/configure



permission\_  
string

- 최대 길이: 없음
- 값의 형식:

```
[-m mode] [-u umask] [-o [owner[,]][uid]]
[-g [group[,]][gid]]
```

여기서 각 구성 요소는 파일 세트에 정의된 각 파일과 디렉토리에 대해 기본 권한 값을 정의합니다. 기본값은 각 파일의 정의에서 변경할 수 있습니다. 소유자 및 그룹 필드는 tag\_string 유형입니다. uid 및 gid 필드는 부호 없는 정수 (unsigned integer) 유형입니다. mode 및 umask는 부호 없는 정수지만 8진수 값 0-7만을 지원합니다.

- SD-UX는 파일이 이미 대상에 존재하는 경우 이 속성에 기초한 기존 권한을 변경하지 않습니다.
- 예: -u 0222 -o root -g sys

revision\_string

- 최대 길이: 64바이트
- 버전 문자열에는 점으로 구분된 one\_line\_string(위)이 없거나 하나 이상 있을 수 있습니다.
- 예: 2.0, B.11.00

software\_specification

- 최대 길이: 없음
- 소프트웨어 사양은 명령줄 선택 사항은 물론 종속성, 상위 및 기타 속성에 대해 소프트웨어를 지정하는 데 사용됩니다. 이 속성은 SD-UX software\_selections에 대해 표준 구문을 사용합니다. 자세한 내용은 58페이지의 “소프트웨어 선택 사항”을 참조하십시오.
- 예: SD.agent 또는 SD,r=2.0,a=HP-UX\_B.11.00\_32

tag\_string

- 최대 길이: 64바이트
- 태그 문자열은 isascii() 문자의 일부만을 지원합니다.

- 첫 번째 문자를 포함하여 “A-Z”, “a-z”, “0-9” 중 하나 이상의 문자를 필요로 합니다.
- `isspace()` 문자는 허용되지 않습니다.
- 다음과 같은 SDU 메타 문자는 허용되지 않습니다.  
., :=
- 다음과 같은 셸 메타 문자는 허용되지 않습니다.  
# ; & () {} | < >
- 다음과 같은 셸 인용 문자는 허용되지 않습니다.  
" ' ` \
- 디렉토리 경로 문자(/)는 허용되지 않습니다.

- 예: HP, SD-UX

`uname_string`

- 최대 길이: 64바이트
- Uname 문자열은 *isascii()* 문자의 일부만을 포함합니다.
- `isspace()` 문자는 허용되지 않습니다.
- 다음과 같은 셸 패턴 일치 표기가 허용됩니다. [] \* ? !
- 패턴은 구분 기호 | 를 사용하여 “Ored” 연산을 수행할 수 있습니다.
- 예: 9000/7\*:\*|9000/8\*:\* , HP-UX, ?.11.\*

## PSF(Product Specification File) 의미 분석

다음 절에서는 PSF를 지정하고 키워드를 정의하는 방법에 대해 설명합니다.

**공급업체 정의 속성** 소프트웨어를 패키징할 때 자신만의 고유한 소프트웨어 속성을 만들 수 있습니다.

패키징하거나 `swmodify`를 사용하여 변경할 때 공급업체 정의 속성을 참고합니다. `swlist`로 이러한 속성을 나열할 수 있습니다.

PSF에서 표준 키워드가 아닌 키워드가 발견되면 이러한 키워드와 연관된 값은 `swpackage`가 작성한 **INDEX** 또는 **INFO** 파일로 전송되어 보관됩니다.

표준 키워드가 아닌 키워드는 파일 이름 문자열로 정의됩니다. 키워드와 연관된 값은 `attribute_value`로서 처리됩니다. 여러 줄에 걸쳐 입력할 수도 있고 키워드 값이 들어 있는 파일을 참조할 수도 있습니다.

---

### 주의

표준 키워드를 잘못 입력하면 **SD-UX**는 이 키워드를 공급업체 정의 속성으로 잘못 인식하고 패키징 오류가 발생할 수 있습니다.

**배포(저장소) 사양** 배포 속성을 통해 저장소가 들어 있는 미디어(테이프나 CD/디렉토리)에 대한 정보를 표시할 수 있습니다. 저장소에 대한 자세한 내용은 136페이지의 “저장소 관리 명령 및 개념”을 참조하십시오. 다음은 배포용 PSF입니다.

```
distribution
  layout_version      1.0
  tag                 APPLICATIONS_CD
  copyright           < data/copyright.cd
  description         <data/description.cd
  number              B1234-56789
  title               HP-UX Applications Software Disk
# Optional vendor specification can be included.
# AT LEAST ONE PRODUCT SPECIFICATION MUST BE INCLUDED.
# Other product specifications are optional.
end
```

`distribution` 키워드는 항상 필요합니다. 나머지 다른 속성은 모두 선택입니다.

distribution 또는 depot

배포 사양을 시작하는 키워드. 각 키워드는 배포 저장소나 테이프 자체 속성을 정의합니다. 배포 사양이 PSF에 포함되어 있더라도 키워드는 모두 선택입니다.

layout\_version PSF 구문은 POSIX 1387.2 Software Administration 표준의 layout\_version=1.0을 준수합니다. SD-UX 이전 버전에서는 POSIX layout\_version=0.8 구문을 지원했는데, 이 구문은 계속 지원됩니다. (swpackage, swmodify, swcopy 또는 swlist에 layout\_version 옵션을 지정하여 레이아웃 버전을 선택할 수도 있습니다.) 자세한 내용은 293페이지의 “PSF 레이아웃 버전 선택”을 참조하십시오.

tag swpackage에 의해 작성 또는 변경되는 대상 저장소(테이프)에 대한 축약 이름

copyright 저장소 내용에 대한 저작권 정보 텍스트(또는 파일을 포인터)

description 대상 저장소에 대한 설명으로, 텍스트 자체일 수도 있고 텍스트가 들어 있는 파일을 가리키는 포인터일 수도 있습니다.

number 배포 미디어(CD 또는 테이프 저장소)의 부품 번호나 제조 번호

title swpackage에 의해 작성 또는 변경되는 대상 저장소(테이프)의 전체 이름

end 배포 사양을 종료하며 값은 필요하지 않습니다. 이 키워드는 선택입니다. 이 키워드를 잘못된 위치에 사용하는 경우 이 사양에 오류가 발생할 것입니다.

**공급업체 사양** 공급업체 속성을 통해 PSF에 설명을 추가할 수 있습니다.

PSF 파일에 정의된 `layout_version`은 공급업체 사양이 제품 및 번들과 연관되는 방식을 결정합니다. `layout_version`이 정의되지 않았거나 1.0으로 정의된 경우 공급업체 사양은 일치하는 `vendor_tag` 속성을 정의하는 모든 후속 제품 및 번들과 연관됩니다.

`layout_version`을 0.8로 지정하면 `vendor_tag`가 명시적으로 정의된 경우를 제외하고는 모든 후속 제품 및 번들은 배포 수준에서 마지막으로 정의된 공급업체 객체나 제품 또는 번들 내에서 정의된 공급업체 객체의 `vendor_tag`로 자동 지정됩니다.

다음은 공급업체 사양의 예입니다.

```
vendor
  tag          HP
  description  < data/description.hp
  title       Hewlett-Packard Company
end
```

각 키워드는 공급업체 객체의 속성을 정의합니다. 공급업체 사양이 PSF에 포함되면 `swpackage`는 공급업체 및 태그 키워드를 필요로 합니다.

---

## 참고

---

공급업체 사양은 공급업체 정의 속성과 같지 않습니다. 자세한 내용은 299페이지의 “공급업체 정의 속성”을 참조하십시오.

<code>vendor</code>	공급업체 사양을 시작하는 키워드
<code>tag</code>	공급업체에 대한 식별자(축약 이름)를 정의합니다.
<code>title</code>	전체 이름(한 줄로 된 설명)을 정의합니다.
<code>description</code>	공급업체에 대한 여러 단락으로 된 설명을 정의하는 것으로, 텍스트 자체(인용 부호 사용)일 수도 있고 텍스트가 포함된 파일을 가리키는 포인터일 수도 있습니다.
<code>end</code>	공급업체 사양을 종료합니다. 이 키워드는 선택입니다.

**범주 사양** (레이아웃 버전 0.8에는 적용되지 않습니다.) 소프트웨어 집합에는 선택 메커니즘으로 사용되는 범주 객체 목록이 들어 있을 수 있습니다. 범주 객체는 “category” 키워드로 식별되며 범주에 대한 추가 정보가 들어 있습니다.

category\_tag 속성은 특정 범주 객체를 가리키며 제품, 번들, 하위 제품 또는 파일 세트 내 어디에든 나타날 수 있습니다.

is\_patch 속성이 true로 설정된 모든 소프트웨어 객체는 자동으로 “patch” 범주가 할당됩니다.

---

## 참고

distribution class의 layout\_version 키워드는 범주가 제품 및 번들과 연관되는 방식에 영향을 미칩니다. 자세한 내용은 293페이지의 “PSF 레이아웃 버전 선택” 및 299페이지의 “PSF(Product Specification File) 의미 분석”을 참조하십시오.

---

범주 사양은 다음과 같습니다.

```
category
  tag          patch_normal
  title        Category of patches
  description  For normal problems
  revision     0.0
end
```

각 키워드는 범주 객체의 속성을 정의합니다. 범주 사양이 PSF에 포함되면 swpackage는 category 및 tag 키워드만 필요로 합니다.

category	범주 사양을 시작하는 키워드
tag	범주 축약 이름 식별자. 이 객체를 제품 또는 번들과 연관시킵니다. 이 tag 속성은 제품이나 번들의 category_tag 속성과 일치해야 합니다.
title	범주의 전체 이름을 정의하는 한 줄로 된 문자열
description	범주에 대한 여러 줄 설명. 설명 값은 텍스트 또는 텍스트 파일의 파일 이름으로 구성될 수 있습니다.

revision	버전 정보(릴리즈 번호, 버전). 설치되거나 복사되는 정의가 이미 저장소에 있는 동일한 범주 태그를 가지고 있는 정의와 일치하지 않을 때 저장소에서 어떤 범주 객체 정의를 관리할 것인지를 결정합니다.
end	선택적 키워드로서 사양을 종료합니다. 값은 필요하지 않습니다. <b>PSF</b> 에 이 키워드의 위치를 잘못 지정하면 사양에 오류가 발생합니다.

**제품 또는 번들 사양** 제품 사양은 PSF에서 필수 클래스입니다. 제품 사양을 통해 패키징하려는 제품을 식별할 수 있습니다.

---

## 참고

*distribution class*의 *layout\_version* 키워드는 범주 및 공급업체 객체가 제품 및 번들과 연관되는 방식에 영향을 미칩니다. 자세한 내용은 293페이지의 “PSF 레이아웃 버전 선택” 및 299페이지의 “PSF(Product Specification File) 의미 분석”을 참조하십시오.

---

제품 사양은 다음과 같습니다.

```
product
tag                SD
architecture      HP-UX_B.11.00_32/64
category_tag      systems_management
contents          prod.fsl,r=1.0,a=,v=
copyright         </mfg/sd/data/copyright
description       </mfg/sd/data/description
directory         /usr
is_locatable      false
is_patch          false
machine_type      *
number            J2326AA
os_name           HP-UX
os_release        ?.11.00.*
os_version        B.11.**
postkernel        /usr/lbin/kernel_build
+ readme          </mfg/sd/data/README
revision          2.0
title             Software Distributor
vendor_tag        HP

# Optional vendor specification
# Optional subproduct specification
# REQUIRED FILESET SPECIFICATION
```

```
end
```

지정된 각 제품 객체에 대해 `swpackage`는 제품 및 태그 키워드와 하나 이상의 파일 세트 정의만을 필요로 합니다. 지정된 각 번들의 경우 `swpackage`는 `bundle`, `tag` 및 `contents` 키워드를 필요로 합니다.



product	제품 사양을 시작하는 필수 키워드입니다.
tag	제품의 식별자(축약 이름)
architecture	제품이나 번들이 실행될 대상 시스템. 제품이 지원하는 정확한 대상 시스템을 정의하는 네 가지 uname 속성(machine_type, os_name, os_release 및 os_version)에 대한 판독 가능한 요약 형태로 제공합니다.
bundle	번들 사양을 시작하는 필수 키워드입니다.
category_tag	반복 가능한 태그 기반 속성으로 소프트웨어 객체가 속해 있는 범주 집합을 식별합니다. 선택 메커니즘으로 사용되며 패치와는 별도로 사용할 수 있습니다. is_patch 속성이 true로 설정된 경우 기본값은 공백 목록이거나 patch입니다. vendor_tag와 마찬가지로 이 속성은 범주에 대한 추가 정보가 들어 있는 범주 객체를 가리키는 포인터로 사용될 수 있습니다(예: 한 줄로 된 제목 정의 및 범주 설명).

---

## 참고

---

범주 태그 patch는 예약되어 있습니다. is\_patch 제품 속성이 true로 설정되면 값이 patch인 내장 category\_tag 속성은 자동으로 제품 정의에 포함됩니다.

contents	번들에 대해 <b>정규화된</b> (모든 버전 식별 속성이 포함된) 소프트웨어 사양 목록
copyright	제품 저작권에 대한 여러 줄로 된 설명으로, 텍스트 자체(인용 부호 사용)일 수도 있고 텍스트가 포함된 파일을 가리키는 포인터일 수도 있습니다.
description	제품에 대한 여러 단락으로 된 설명으로, 텍스트 자체(인용 부호 사용)일 수도 있고 텍스트가 포함된 파일을 가리키는 포인터일 수도 있습니다.
directory	제품 파일이 설치될 디렉토리(제품의 루트 디렉토리)에 대한 기본 절대 경로 이름입니다. 값을 지정하지 않으면 swpackage가 / 값을 지정합니다.

is_locatable	제품이나 번들을 모든 제품 디렉토리에 설치할 수 있는지 또는 특정 디렉토리에 설치해야 하는지 여부를 정의합니다. 이 속성은 <b>true</b> 또는 <b>false</b> 로 설정할 수 있습니다. 값을 설정하지 않으면 swpackage가 기본 속성을 “ <b>false</b> ”로 지정합니다.
is_patch	소프트웨어 객체를 패치로 식별하는 부울 값 플래그. 기본값은 <b>false</b> 입니다. <b>true</b> 로 설정하면 값이 patch인 내장 category_tag 속성이 자동으로 제품 정의에 포함됩니다.
machine_type	제품이 실행될 시스템의 종류. 이 키워드를 지정하지 않으면 키워드에는 모든 시스템에서 실행됨을 의미하는 와일드카드 값인 *가 지정됩니다. 여러 플랫폼의 경우 각 시스템을  (수직 막대)로 구분하여 지정해야 합니다. 예를 들어 키워드 값 <b>9000/7* 9000/8*</b> 는 제품이 모든 <b>HP Series 9000 Model 7XX</b> 또는 모든 <b>HP 9000 Series 8XX</b> 시스템에서 실행됨을 의미합니다. 또는 <b>9000/[78]*</b> 값도 마찬가지로 작동합니다.

기타 예:

\* (시스템 종류에 상관없는 경우)

**9000/7??:32\***

(Series 700, 32비트 가능 하드웨어 필요)

**\*:\*64** (64비트 가능 하드웨어 필요)

**\*:32:** (32비트 가능 하드웨어 필요)

**9000/7??:\*64** (Series 700, 64비트 가능 하드웨어 필요)

**9000/[78]??:32\*** (Series 800, 32비트 가능 하드웨어 필요)

**9000/[78]??:\*64** (Series 800, 64비트 가능 하드웨어 필요)

이 값은 대상의 **uname -m** 또는

**getconf \_CS\_HW\_CPU\_SUPP\_BITS** 결과와 일치합니다.

number

제품의 부품 번호 또는 주문 번호

os\_name

제품이 실행될 운영 체제 이름. 이 속성을 지정하지 않은 경우 모든 운영 체제에서 실행됨을 의미하는 \* 값이 지정됩니다. 여러 운영 체제의 경우 와일드카드나 | 기호를 사용하여 구분하십시오. 이 값은 대상의 **uname -s** 또는 **getconf \_CS\_KERNEL\_BITS** 결과와 일치합니다.

<code>os_release</code>	제품 운영 체제의 릴리즈 번호. 이 속성을 지정하지 않은 경우 모든 운영 체제에서 실행됨을 의미하는 * 값이 지정됩니다. 여러 운영 체제의 경우 와일드카드나   기호를 사용하여 구분하십시오. 이 값은 대상의 <code>uname -r</code> 결과와 일치합니다.
<code>os_version</code>	제품이 실행될 운영 체제의 버전 번호. 이 속성을 지정하지 않은 경우 모든 버전에서 실행됨을 의미하는 * 값이 지정됩니다. 여러 운영 체제의 경우 와일드카드나   기호를 사용하여 구분하십시오. 이 값은 대상의 <code>uname -v</code> 결과와 일치합니다.
<code>postkernel</code>	커널 파일 세트가 로드될 때 실행될 커널 빌드 스크립트를 정의합니다. 커널 파일 세트의 <code>is_kernel</code> 속성은 <code>true</code> 로 설정됩니다. 기본 커널 스크립트는 <code>/usr/sbin/mk_kernel</code> 입니다. (자세한 내용은 <code>mk_kernel (1M)</code> 의 설명서 참조 페이지를 참조하십시오.) 기본 스크립트는 <code>postkernel</code> 속성이 지정되지 않은 경우 실행됩니다. 제품 하나 당 단 하나의 커널 빌드 스크립트가 허용되며, 여러 파일 세트에 대해 정의되었다고 하더라도 스크립트는 단 한 번만 실행됩니다.
<code>readme</code>	제품에 대한 <b>README</b> 정보의 텍스트 파일. 이 값은 텍스트가 들어 있는 파일을 가리키는 포인터여야 합니다.
<code>revision</code>	제품 또는 번들에 대한 버전 정보(릴리즈 번호, 버전).
<code>title</code>	제품이나 번들을 더 자세히 식별하는 한 줄로 된 문자열
<code>vendor_tag</code>	공급업체 객체에 일치하는 <code>tag</code> 속성이 있는 경우 이 제품이나 번들을 <b>PSF</b> 에 별도로 정의된 해당 객체와 연관시킵니다.
<code>end</code>	제품 또는 번들 사양을 종료합니다. 값은 필요하지 않습니다. 이 키워드는 선택입니다. 이 키워드를 잘못된 위치에 사용하는 경우 이 사양에 오류가 발생할 것입니다.

**제어 스크립트 사양** SD-UX는 다른 HP-UX 명령 및 기능을 통한 추가 검사 및 작업을 가능하게 하는 제품 및 파일 세트 제어 스크립트의 실행을 지원합니다. `swask`, `swinstall`, `swconfig`, `swverify` 및 `swremove` 명령은 각각 기본 루트에서 제어 스크립트를 하나 이상 실행시킬 수 있습니다. 스크립트는 모두 선택적이지만 소프트웨어 패키지가 수행할 작업을 올바르게 완료할 수 있도록 스크립트를 작성해야 하는 경우가 많습니다. 제어 스크립트에 대한 자세한 설명은 11장, 347페이지의 “제어 스크립트 사용”을 참조하십시오.

**하위 제품 사양** 하위 제품 사양을 통해 더 큰 범위의 제품 사양 내에서 파일 세트를 그룹화할 수 있습니다. 하위 제품은 선택입니다. 하위 제품 사양은 다음과 같습니다.

```
subproduct
  tag          Manager
  contents     manager agent packager man doc
  description  </mfg/sd/data/manager/description
  title        SD Management Interfaces Subset
end
```

각 키워드는 하위 객체의 속성을 정의합니다. 하위 제품 객체가 지정되는 경우 `swpackage`에 `subproduct`, `tag` 및 `contents` 키워드가 필요합니다.

<code>subproduct</code>	하위 제품 사양을 시작하는 키워드
<code>tag</code>	하위 제품의 식별자(축약 이름)
<code>contents</code>	공백으로 구분한 하위 제품 파일 세트의 tag 값 목록(즉, <code>contents fileset1 fileset2 fileset3 ...filesetN</code> ). <b>PSF</b> 에서 파일 세트 정의는 하위 제품 정의에 포함되지 않습니다. <code>contents</code> 키워드는 하위 제품에 파일 세트를 할당하는 데 사용됩니다. 이 링크를 통해 하나의 파일 세트가 여러 하위 제품에 포함될 수 있습니다.
<code>description</code>	하위 제품에 대한 여러 줄로 된 설명으로, 텍스트 자체(인용 부호 사용)일 수도 있고 텍스트가 포함된 파일을 가리키는 포인터일 수도 있습니다.
<code>title</code>	하위 제품을 더 자세히 식별하는 한 줄로 된 문자열
<code>end</code>	하위 제품 사양을 종료합니다. 값은 필요하지 않습니다. 이 키워드는 선택입니다. 이 키워드를 잘못된 위치에 사용하는 경우 이 사양에 오류가 발생할 것입니다.

**파일 세트 사양** 파일 세트 사양은 PSF에서 필수적입니다. 파일 세트를 사용하여 파일들을 서로 그룹화합니다.

파일 세트 사양은 다음과 같습니다.

```
fileset
    tag            manB
    ancestor       OLDS.D.MAN
    architecture   HP-UX_B.11.00_32/64
    category_tag   manpg
    description    </mfg/sd/data/man/description
    is_kernel      false
    is_locatable   false
    is_patch       false
    is_reboot      false
    is_sparse      false
    machine_type   *
    os_name        HP-UX
    os_release     ?.11.00.*
    os_version     ?
    revision       2.40
    supersedes     product.fileset,fr=revision
    title          Commands (management utilities)
# Optional control script specification
# Optional dependency specification
# REQUIRED FILE SPECIFICATION
# Additional file specifications optional.
end
```

각 키워드는 속성을 파일 세트 객체로 정의합니다. 지정된 각 파일 세트 객체에 대해 swpackage는 fileset 및 tag 키워드와 하나 이상의 파일 사양을 필요로 합니다.

```
tag            파일 세트 식별자(축약 이름)

architecture   제품 하나에 복수 아키텍처에 대한 파일 세트가 포함되어 있는 경우 해당 파일 세트가 실행될 대상 시스템을 설명합니다. 제품이 지원하는 정확한 대상 시스템을 정의하는 네 가지 uname(1) 속성을 판독 가능한 요약 형태로 제공합니다. 많은 파일 세트에 아키텍처가 포함되지 않으며 제품 아키텍처만 정의해야 합니다.
```

ancestor	<p>match_target 설치 옵션을 지정한 경우 대상 시스템에 설치되었을 때 현재 파일 세트와 일치하는 파일 세트 목록.</p> <p><b>patch_match_target</b>이 정의된 경우 검사할 상위 파일 세트도 지정합니다.</p>
category_tag	<p>반복 가능한 태그 기반 속성으로 소프트웨어 객체가 속해 있는 범주 집합을 식별합니다. 선택 메커니즘으로 사용되며 패치와는 별도로 사용할 수 있습니다. <b>is_patch</b> 속성이 <b>true</b>인 경우 기본값은 공백 목록이거나 패치입니다.</p> <p><b>vendor_tag</b>와 마찬가지로 이 속성은 범주에 대한 추가 정보가 들어 있는 범주 객체를 가리키는 포인터로 사용될 수 있습니다(예: 한 줄로 된 제목 정의 및 범주 설명).</p>

---

**참고**

---

범주 태그 **patch**는 예약되어 있습니다. **is\_patch** 파일 속성이 **true**로 설정되면 값이 **patch**인 내장 **category\_tag** 속성은 자동으로 파일 정의에 포함됩니다.

description	<p>파일에 대한 여러 단락으로 된 설명을 정의하며, 텍스트 자체(인용 부호 사용)일 수도 있고 텍스트가 포함된 파일을 가리키는 포인터일 수도 있습니다.</p>
is_kernel	<p><b>true</b> 값을 지정하면 파일 세트는 운영 체제 시스템에 작동하는 것으로 정의되며, 파일 세트가 설치된 후 대상 시스템 커널 빌드 프로세스가 호출됩니다. 이 속성을 지정하지 않으면 <b>swpackage</b>는 기본값으로 <b>false</b>를 가정합니다.</p>
is_locatable	<p>파일 세트를 모든 제품 디렉토리에 설치할 수 있는지 또는 특정 디렉토리에 설치해야 하는지 여부를 정의합니다. 이 속성은 <b>true</b> 또는 <b>false</b>로 설정할 수 있습니다. 값을 정의하지 않으면 <b>swpackage</b>가 기본 속성을 <b>false</b>로 설정합니다.</p>
is_patch	<p>소프트웨어 객체를 패치로 식별합니다. 기본값은 <b>false</b>입니다. <b>true</b>로 설정하면 값이 <b>patch</b>인 내장 <b>category_tag</b> 속성이 자동으로 포함됩니다.</p>



```
*:32:          32비트 가능 하드웨어 필요
9000/7??:*64   Series 700, 64비트 가능 하드웨어 필요
9000/[78]??:32* Series 800, 32비트 가능 하드웨어 필요
9000/[78]??:*64 Series 800, 64비트 가능 하드웨어 필요
이 값은 대상의 uname -m 또는
getconf _CS_HW_CPU_SUPP_BITS 결과와 일치합니다.
```



<code>os_name</code>	파일 세트 아키텍처가 정의되어 있는 경우 파일이 실행될 운영 체제를 정의합니다. (이를 지정하지 않으면 <code>swpackage</code> 는 모든 운영 체제에서 파일이 실행됨을 의미하는 * 값을 지정합니다.) 여러 운영 체제의 경우 와일드카드나 ' ' 문자를 사용하여 구분하십시오. 이 속성은 지원되는 대상 시스템에서 <code>uname -s</code> 또는 <code>getconf KERNEL_BITS</code> 와 패턴이 일치해야 합니다.
<code>os_release</code>	파일이 실행될 운영 체제 릴리즈를 정의합니다. (이를 지정하지 않으면 <code>swpackage</code> 는 모든 릴리즈에서 파일이 실행됨을 의미하는 * 값을 지정합니다.) 여러 운영 체제 릴리즈가 있는 경우 와일드카드나 ' ' 문자를 사용하여 구분하십시오. 이 속성은 지원되는 대상 시스템에서 <code>uname -r</code> 값과 패턴이 일치해야 합니다.
<code>os_version</code>	제품이 실행될 운영 체제의 버전 번호. 이 속성을 지정하지 않은 경우 모든 버전에서 실행됨을 의미하는 * 값이 지정됩니다. 여러 운영 체제가 있는 경우 와일드카드나   기호를 사용하여 구분하십시오. 이 값은 대상의 <code>uname -v</code> 결과와 일치합니다.
<code>revision</code>	파일 세트의 버전(릴리즈 번호, 버전 번호)을 정의합니다.
<code>supersedes</code>	패치가 나중의 패치로 대체(또는 병합)될 때 사용됩니다. 이 속성은 설치되거나 복사되는 패치에 의해 어떤 이전 패치가 대체되는지를 나타냅니다. 이 속성 값은 이 패치가 대체하는 다른 패치의 소프트웨어 사양 목록입니다.
<code>title</code>	파일 세트의 전체 이름(한 줄로 된 설명)을 정의합니다.
<code>end</code>	파일 세트 사양을 종료하는 선택 키워드입니다. 값은 필요하지 않습니다. 이 키워드의 위치를 잘못 지정하면 파일 사양에 오류가 발생합니다.

**종속성 사양** `swinstall`, `swcopy`, `swverify` 및 `swremove` 명령은 소프트웨어 종속성을 인식합니다. 예를 들어 `swinstall`은 기본적으로 모든 종속성이 충족되지 않으면 설치가 이루어지지 않도록 합니다.

PSF는 파일 세트 간의 종속성을 지정합니다. 종속성은 파일 세트 클래스 정의 내에서 정의됩니다(309페이지의 “파일 세트 사양” 참조).

또한 다음 항목 간의 종속성을 정의할 수 있습니다.

- 파일 세트 및 다른 제품(즉, 해당 제품의 하위 제품)
- 해당 제품 내 특정 파일 세트
- 제품 전체

SD-UX는 다음과 같은 종류의 종속성을 지원합니다.

**Corequisite**      파일 세트가 올바르게 작동하기 위해 반드시 있어야 하는 소프트웨어. 예를 들어 설치 파일 세트에 대한 필수 조건을 지정하면 파일 세트 자체가 설치될 때 필수 조건 소프트웨어가 이미 설치되어 있거나 함께 설치되어야 함을 의미합니다.

(필수 조건 종속성이 “”런타임 종속성”(로드 순서)을 의미하는 것은 아님에 유의하십시오.)

**Exrequisite**      SD-UX에 의해 파일 세트가 작동될 때 없어야 하는 소프트웨어. 예를 들어 파일 세트에 대한 제외 조건을 지정하면 지정된 제외 조건 소프트웨어가 설치되어 있거나 동시에 설치되는 경우 해당 파일 세트가 설치되지 않습니다.

**Prerequisite**    SD-UX에 의해 파일 세트가 올바르게 작동하려면 미리 설치 및 구성되어야 하는 소프트웨어. 전제 조건은 `swinstall` 명령의 설치 순서를 제어합니다(설치 시 종속성).

종속성은 PSF에서 `software_specification` 값의 유형으로 지정됩니다. (자세한 내용은 295페이지의 “PSF 값의 유형”을 참조하십시오.) 예를 들면 다음과 같습니다.

```

corequisites    SD.data
prerequisites   productA,r>=2.1
exerequisites   productB,r>=2.1

```

---

## 참고

---

종속성은 조건 소프트웨어에 대한 제품 태그로 시작하는 소프트웨어 사양을 사용하여 지정해야 합니다.

종속성 간의 **AND** 관계를 정의하여 여러 종속성을 지정할 수 있습니다(**AND**는 모든 종속성이 충족되어야 함을 의미합니다).

또한 **or()** 문자를 사용하여 **OR** 관계를 정의할 수도 있습니다. 이때 다음과 같은 규칙이 적용됩니다.

- **OR** 문자 앞 뒤로 공백이 허용됩니다.
- **OR** 종속성은 왼쪽에서 오른쪽으로 해석됩니다.

다음은 그 예입니다.

```
corequisite P.F
prerequisite ProdA | ProdB | ProdC.F | ProdC.FS
corequisite ProdX | ProdY | ProdZ | ProdW.FS
```

**제어 스크립트 사양 SD-UX**는 다른 **HP-UX** 명령 및 기능을 통한 추가 검사 및 작업을 가능하게 하는 제품 및 파일 세트 제어 스크립트의 실행을 지원합니다. **swask**, **swinstall**, **swconfig**, **swverify** 및 **swremove** 명령은 각각 기본 루트에서 제어 스크립트를 하나 이상 실행시킬 수 있습니다. 스크립트를 작성하여 이를 소프트웨어 패키지에 포함시킬 수 있습니다. 스크립트는 모두 선택적이지만 소프트웨어 패키지가 수행할 작업을 올바르게 완료할 수 있도록 스크립트를 작성해야 하는 경우가 많습니다. 제어 스크립트에 대한 자세한 설명은 11장, 347페이지의 “제어 스크립트 사용”을 참조하십시오.

**파일 사양** 파일 세트 사양 내에서 `swpackage`에 의해 다음과 같은 파일 종류가 파일 세트에 패키징되도록 지정할 수 있습니다.

- 제어 스크립트
- 디렉토리
- 하드 링크
- 일반 파일
- 심볼릭 링크
- 아카이브

PSF에 인식할 수 없거나 패키징할 수 없는 파일 종류가 들어 있는 경우 `swpackage`에 오류가 발생합니다.

`swpackage` 명령은 파일 세트에 포함되어 있는 파일을 지정하기 위한 특별한 메커니즘을 지원합니다.

**기본 권한 사양** 파일 세트의 파일 전체 또는 일부에 대해 기본 권한 세트를 정의할 수 있습니다.

**디렉토리 매핑** 파일 세트의 파일이 위치한 소스 디렉토리에서 `swpackage`를 연결시킬 수 있습니다. 또한 이 소스 디렉토리를 제품 파일의 일부가 위치할 적절한(대상) 디렉토리로 매핑할 수 있습니다.

**명시적 파일 사양** 파일 세트의 파일 전체 또는 일부에 대해 각각의 소스 파일과 대상 위치를 지정할 수 있습니다.

**재귀적(암시적) 파일 사양** 디렉토리 매핑이 활성화되어 있는 경우 디렉토리의 모든 파일을 파일 세트에 재귀적으로 포함시키도록 `swpackage`에 지정하기만 하면 됩니다.

**PSF 확장** `include` 및 `exclude` 파일을 사용하여 파일 정의를 확장할 수 있습니다.

이러한 메커니즘은 서로 조합하여 사용할 수 있습니다.

**기본 권한 사양** 기본적으로 대상 파일은 소스 파일의 모드, 소유자 및 그룹을 상속받습니다. `file_permissions` 키워드를 사용하여 파일 세트에 패키징되는 모든 파일에 대한 기본 권한 마스크, 소유자 및 그룹을 설정할 수 있습니다.

```
file_permissions [-m mode/ -u umask] [-o [owner[,]][uid]]\  
[-g [group[,]][gid]][-t type]
```

`file_permissions`

이 키워드는 이 키워드가 정의된 파일 세트에만 적용됩니다. `file_permissions`을 여러 개 지정할 수 있으며, 나중 정의가 이전 정의를 대체합니다.

`-m mode`

이 옵션은 모든 파일에 대한 기본(8진수) 모드를 정의합니다.

`-u umask`

기본값으로 8진수 모드를 지정하는 대신 기존 소스 파일의 모드에서 그 값을 빼서 대상 파일의 모드를 산출할 수 있는 8진수 `umask` (1) 값을 지정할 수 있습니다.

`umask`를 지정하여 실행 파일, 비실행 파일 및 디렉토리에 대한 기본 모드를 설정할 수 있습니다. (`-m`을 사용하여 모든 파일에 특정 모드를 설정할 수 있습니다.)

`-o [owner[,]][uid]`

이 옵션은 대상 파일의 소유자 이름 및/또는 `uid`를 정의합니다. 자세한 내용은 319페이지의 “명시적 파일 사양”의 `-o` 옵션에 대한 설명을 참조하십시오.

`-g [group[,]][gid]`

이 옵션은 대상 파일의 그룹 이름 및/또는 `gid`를 정의합니다. 자세한 내용은 319페이지의 “명시적 파일 사양”의 `-g` 옵션에 대한 설명을 참조하십시오.

`-t type`

패키징 전에 존재할 필요가 없는 파일을 정의합니다.

다음 예에서는 `file_permission` 키워드의 사용을 보여줍니다.

- 모든 파일 객체에 대해 읽기 전용 444 모드를 설정합니다(모든 실행 파일과 디렉토리에 대한 설정을 재정의합니다).

```
file_permissions -m 444
```

- 비실행 파일에 대해 읽기 모드를 설정하고 실행 파일과 디렉토리에 대해 읽기/실행 모드를 설정합니다.

```
file_permissions -u 222
```

- 동일 모드의 기본값과 소유자 및 그룹을 설정합니다.

```
file_permissions -u 222 -o bin -g bin
```

## PSF(Product Specification File) 만들기

- 동일 모드의 기본값과 **uid** 및 **gid**를 설정합니다.

```
file_permissions -u 222 -o 2 -g 2
```

- 위의 권한과 함께 소유자 쓰기 권한을 설정합니다.

```
file_permissions -u 022 -o 2 -g 2
```

- `file_permissions`를 정의하지 않으면 `swpackage`는 기존 소스 파일을 근거로 대상 파일 객체에 대해 기본값 `file_permissions -u 000`을 사용합니다. (대상 파일에 대해 특정 값으로 대체하지 않는 한 `mode`, `owner/uid`, `group/gid`는 소스 파일을 근거로 함을 의미합니다.)

**디렉토리 매핑** (선택) `directory source [= destination]` 사양은 그 뒤에 나열되는 파일이 위치하는 소스 디렉토리를 정의합니다. 또한 `source` 디렉토리를 패키징된 파일이 설치될 `destination` 디렉토리로 매핑할 수 있습니다.

예를 들어 다음 정의를 지정하면

```
directory /build/hpux/mfg/usr = /usr
```

`/build/hpux/mfg/` 디렉토리의 파일이 설치될 때 `/usr/sbin`이라는 접두어가 붙게 됩니다. 대상 디렉토리를 제품 사양에 정의하면 대상 디렉토리는 제품의 `directory` 속성의 상위 집합이 되어야 합니다. 제품의 `directory`를 정의하면 대상이 상위 집합이 아니므로 `swpackage`에서 오류가 발생합니다.

대상 디렉토리는 절대 경로 이름이어야 합니다. 그렇지 않으면 `swpackage`에서 오류가 발생합니다.

소스 디렉토리는 절대 경로 이름일 수도 있고 상대 경로 이름일 수도 있습니다. 상대 경로인 경우 `swpackage`는 이를 명령을 호출한 현재 작업 디렉토리에 대한 상대 경로로 해석합니다.

소스 디렉토리가 존재하지 않으면 `swpackage`에서 오류가 발생합니다.

**명시적 파일 사양** 파일이 파일 세트에 패키징되도록 명시적으로 지정할 수 있습니다. 재귀적으로 모든 파일과 디렉토리를 포함시키려는 경우 재귀적 파일 사양(`file *`)을 사용합니다.

`directory` 키워드를 사용하여 명시적으로 지정된 파일에 대해 소스와 대상을 정의할 수 있습니다. `directory` 키워드가 활성화되어 있지 않은 경우 각 파일에 대해 전체 소스 경로와 절대 대상 경로를 지정해야 합니다. 명시적 파일 사양은 파일 단위로 `directory` 및/또는 `file_permissions` 키워드가 설정한 사양을 대체하거나 이 사양에 추가합니다.

명시적 파일 사양은 다음과 같은 형식을 사용합니다.

```
file [-v] [-m mode] [-o [owner,][uid]] [-g [group,][gid]]
[-t type] [source] [destination]
```

**file** 이 키워드는 파일 세트에 포함시킬 기존 파일(대개 현재 활성화된 소스 디렉토리 내에 있음)을 지정합니다.

**source** 이 값은 패키지에 포함시키려는 파일에 대한 경로를 정의합니다.  
이것이 상대 경로인 경우 `swpackage`는 `directory` 키워드에 의해 설정된 소스 디렉토리에 대한 상대 경로를 검색합니다. 소스 디렉토리가 활성화되어 있지 않은 경우 `swpackage`는 명령이 호출된 현재 작업 디렉토리에 대한 상대 경로를 검색합니다.

`file_permission` 키워드가 활성화되어 있거나 `-m`, `-o` 또는 `-g` 옵션이 파일 사양에도 포함되어 있는 경우를 제외하고는 대상 파일 객체에 대한 모든 속성은 소스 파일에서 가져옵니다.

**destination** 이 값은 파일이 설치될 대상 경로를 정의합니다. `destination`이 상대 경로인 경우 `directory` 키워드로 설정된 활성화된 대상 디렉토리가 상대 경로 이름 앞에 붙습니다. 상대 경로이고 대상 디렉토리가 활성화되어 있지 않으면 `swpackage`에 오류가 발생합니다. 대상을 지정하지 않으면 소스 경로가 대상으로 사용되고 활성화된 대상 디렉토리에 적절한 매핑이 이루어집니다(있는 경우).

`-m mode` 이 옵션은 대상에서 파일이나 디렉토리에 대한 (8진) 모드를 정의합니다.

`-o [owner[,]][uid]` 이 옵션은 대상에서 파일의 소유자 이름 및/또는 `uid`를 정의합니다. `owner`를 지정하면 `owner`와 `uid` 속성은 호스트의 `/etc/owner` 패키징을 근거로 대상 파일에 대해 설정됩니다. `uid`만을 지정한 경우 대상의 `uid` 속성으로 설정되며 소유자 이름은 지정되지 않습니다. 둘 다 지정하면 각각 파일 객체에 대한 해당 속성을 설정합니다.

소유자 이름이 대상 시스템의 `/etc/passwd` 파일에 지정되거나 정의되어 있지 않으면 설치 중에 `owner` 속성을 사용하여 소유자 이름과 `uid`를 설정합니다. 이 경우 `uid` 속성은 `uid`를 설정하는 데 사용됩니다.

`-g [group[,]][gid]` 이 옵션은 대상에서 파일의 그룹 이름 및/또는 `gid`를 정의합니다. `group` 을 지정하면 `group`과 `gid` 속성은 호스트의 `/etc/group` 패키징을 근거로 대상 파일에 대해 설정됩니다. `gid`만을 지정한 경우 대상의 `gid`로 설정되며 그룹 이름은 지정되지 않습니다. 둘 다 지정하면 각각 파일 객체에 대한 해당 속성을 설정합니다.

그룹 이름이 지정되지 않았거나 대상 시스템의 `/etc/group`에 정의되어 있지 않으면 설치 중에 그룹 속성을 사용하여 그룹 이름과 `gid`를 설정합니다. 이 경우 `gid` 속성은 `gid`를 설정하는 데 사용됩니다.

`-t type` 패키징 전에 존재하면 안 되는 파일에 대해 파일 유형 `d`(디렉토리), `s`(심볼릭), `h`(하드 링크) 또는 `a`(아카이브)를 정의합니다.

`-v` 이 옵션은 파일을 휘발성(`volatile`)으로 표시하는데 이는 설치된 후에 파일 세트에 영향을 미치지 않고 수정(즉, 삭제)될 수 있음을 의미합니다.

설치 후 정상적인 사용을 통해 속성(크기, 최종 수정 시간 등)이 변경될 수 있는 파일은 **PSF** 파일에 휘발성으로 지정되어야 합니다(파일 정의 줄에 `-v` 지정). `swverify`는 기본적으로 `is_volatile` 속성이 `true`로 설정된 파일에 대해 파일 속성을 검사하지 않습니다(`swverify`에 대해서는 `check_volatile` 옵션 참조).

### 오류 메시지

소스 디렉토리의 기존 파일을 처리할 때 `swpackage`는 다음과 같은 네 종류의 오류를 식별합니다.

- 디렉토리를 검색할 수 없습니다(권한이 거부됨).
- 파일을 읽을 수 없습니다(권한이 거부됨).
- 지원되지 않는 파일 유형입니다(소스 파일은 제어 스크립트, 일반 파일, 디렉토리, 하드 링크 또는 심볼릭 링크여야 합니다).
- 파일이 없습니다.



## 디렉토리 및 파일 키워드 사용

다음 예에서는 `directory` 및 `file` 키워드의 사용을 보여줍니다.

- `/build/hpux/mfg` 아래의 모든 파일을 `/usr` 아래에 포함시킵니다.

```
directory /build/hpux/mfg=/usr
file      *
```

- `/build/hpux/mfg/` 아래에 있는 특정 파일만을 `/usr` 및 `/var/adm/sw` 아래에 포함시킵니다.

```
directory /build/hpux/mfg=/usr
file      sbin/swinstall
file      sbin/swcopy
. . .
directory /build/hpux/mfg=/var/adm/sw
file      nls/swinstall.cat  nls/en_US.88591/swinstall.cat
file      defaults newconfig/defaults
file      defaults defaults
```

- 명시적으로 파일을 나열하며 디렉토리 매핑은 지정되지 않습니다.

```
file /build/hpux/mfg/usr/bin/swinstall /usr/sbin/swinstall
file /build/hpux/mfg/usr/bin/swcopy /usr/sbin/swcopy
file /build/hpux/mfg/data/nls/swinstall.cat
    /var/adm/sw/nls/en_US.88591/swinstall.cat
file /build/hpux/mfg/data/defaults
    /var/adm/sw/newconfig/defaults
file /build/hpux/mfg/data/defaults /var/adm/sw/defaults
```

- 모든 사양 유형을 사용하여 파일을 포함시킵니다.

```
directory /build/hpux/mfg/usr=/usr
file      *
directory /build/hpux/mfg/data=/var/adm/sw
file      defaults newconfig/defaults
file      /build/hpux/mfg/data/defaults=/var/adm/sw/defaults
```

**재귀적 파일 사양** `file *` 키워드는 `swpackage`에 파일 세트 내 현재 소스 디렉토리에 있는 모든 파일과 디렉토리를 포함시키도록 지시합니다. `swpackage`는 파일 세트 내 소스 디렉토리의 내용을 재귀적으로 모두 포함시키려고 시도합니다. (예를 들어 “**dm**”으로 시작하는 모든 파일을 나타내는 `file dm*`과 같은 부분적인 와일드 카드는 지원되지 않습니다.)

`file_permission` 키워드(아래에서 설명)가 활성화되어 있는 경우를 제외하고는 대상 파일 객체에 대한 모든 속성을 소스 파일에서 가져옵니다.

사용자는

```
directory source[=destination]
```

```
file *
```

쌍을 여러 개 지정하여 서로 다른 소스 디렉토리의 모든 파일을 하나의 파일 세트로 모을 수 있습니다.

모든 파일과 디렉토리를 재귀적으로 포함시키지 않으려면 명시적 파일 사양을 사용합니다.

`file *` 사양을 지정하기 전에 `directory` 키워드를 지정해야 합니다. 그렇지 않으면 `swpackage`에서 오류가 발생합니다.

### 오류 메시지

재귀적으로 디렉토리를 처리할 때 `swpackage`에서 다음과 같은 오류가 발생할 수 있습니다.

- 디렉토리를 검색할 수 없습니다(권한이 거부됨).
- 파일을 읽을 수 없습니다(권한이 거부됨).
- 지원되지 않는 파일 유형입니다.

**PSF 확장** PSF에는 확장된 파일 정의가 포함될 수 있습니다. SD에서는 현재 `exclude` 및 `include` 파일을 지원합니다.

**Exclude** 파일을 통해 명시적으로 파일을 제외시킬 수 있습니다(그렇지 않으면 PSF에 포함됩니다). 구문은 다음과 같습니다.

```
exclude filename
```

`exclude` 파일은 파일 정의 뒤에만 지정할 수 있습니다. `exclude` 키워드 뒤에 나열된 파일은 현재 컨텍스트에서 제외됩니다(예를 들어 재귀적 파일 정의나 와일드카드).

파일 이름으로 디렉토리를 지정한 경우 그 디렉토리 아래에 있는 파일이 모두 제외됩니다.

**Include** 파일을 통해 별도의 파일로부터 파일 정의를 포함시킬 수 있습니다. 구문은 다음과 같습니다.

```
file < filename
```

`include` 파일은 부등호 기호(<)를 사용하여 파일 키워드와 구분해야 합니다.

### 파일 재지정

일반적인 속성에 대해 파일을 그룹으로 지정(`file *` 사용)할 수 있을 뿐만 아니라 PSF를 통해 해당 정의 내에서 파일을 “재지정”하여 속성을 개별적으로 변경할 수 있습니다.

예를 들어 100개의 파일이 들어 있는 파일 세트의 모든 파일을 지정한다고 가정합니다. 이 파일들은 모두 재귀적으로 “찾아서” 파일 세트에 패키징되어야 합니다. 파일들의 소유자, 그룹 및 모드 및 기타 파일 속성 등은 대부분 동일합니다.

100개의 파일 중에 변경될 가능성이 있는 것은 5개입니다(즉, 이 파일들이 삭제되든 변경되든 상관하지 않습니다). 따라서 100개의 파일을 개별적으로 모두 나열하고 다섯 개의 파일에 대해 `-v` 옵션을 사용하는 대신 `file *`를 사용하여 100개 파일을 모두 지정한 다음 다섯 개 파일을 개별적으로 변경합니다. 예를 들어 파일 1, 2, 3, 4 및 5의 경우 다음과 같습니다.

```
directory source = /product file *

file -v 1
file -v 2
file -v 3
file -v 4
file -v 5
```

이는 권한에 대해서도 제대로 작동합니다. 예를 들어 앞의 예에서 100개 파일 거의 모두가 동일한 권한 속성을 갖는데, 파일 1, 2 및 3에서만 다른 소유자 및 모드를 필요로 한다고 가정합니다.

```
directory source = /product

file_permissions -o bin -g bin -m 555
file *

file_permissions -o root -g other -m -04555
file 1
file 2
file 3
```

이 기능은 재귀적 파일 사양 기능과 명시적 파일 사양 기능을 결합한 것입니다(319페이지의 “명시적 파일 사양” 참조).

## 소프트웨어 패키징(`swpackage`)

`swpackage` 명령은 PSF에 정의된 소프트웨어 제품을 저장소로 패키징합니다. 그러면 다른 SD-UX 명령을 통해 저장소의 소프트웨어를 사용할 수 있습니다.

### 개요

기능 및 제한 사항은 다음과 같습니다.

- PSF를 사용하여 파일을 제품, 하위 제품 및 파일 세트로 구성합니다.
- 제어 스크립트와 PSF를 포함시켜 소프트웨어를 대상 시스템에 설치할 때 소프트웨어를 처리하는 방식을 보다 자세하게 지정할 수 있습니다.
- 패키징되는 파일의 권한을 설정합니다.
- 단일 파일 세트의 간단한 제품을 패키징할 수도 있고 여러 파일 세트와 하위 제품으로 구성되는 복잡한 제품을 패키징할 수도 있습니다.
- 기존 제품을 다시 패키징(변경)할 수 있는 방법을 제공합니다.
- `swpackage` 명령은 명령줄 사용자 인터페이스만을 제공합니다. 패키징 작업에는 그래픽 사용자 인터페이스를 사용할 수 없습니다.
- 디렉토리 저장소(CD 포함) 또는 테이프 저장소(인터넷을 통해 소프트웨어 배포 시 유용)를 만들 수 있습니다.
- 새로 만들어진 저장소를 자동으로 등록하지 않습니다. `swreg` 명령을 사용해야 합니다(153페이지의 “저장소 등록 및 등록 해제(`swreg`)” 참조).

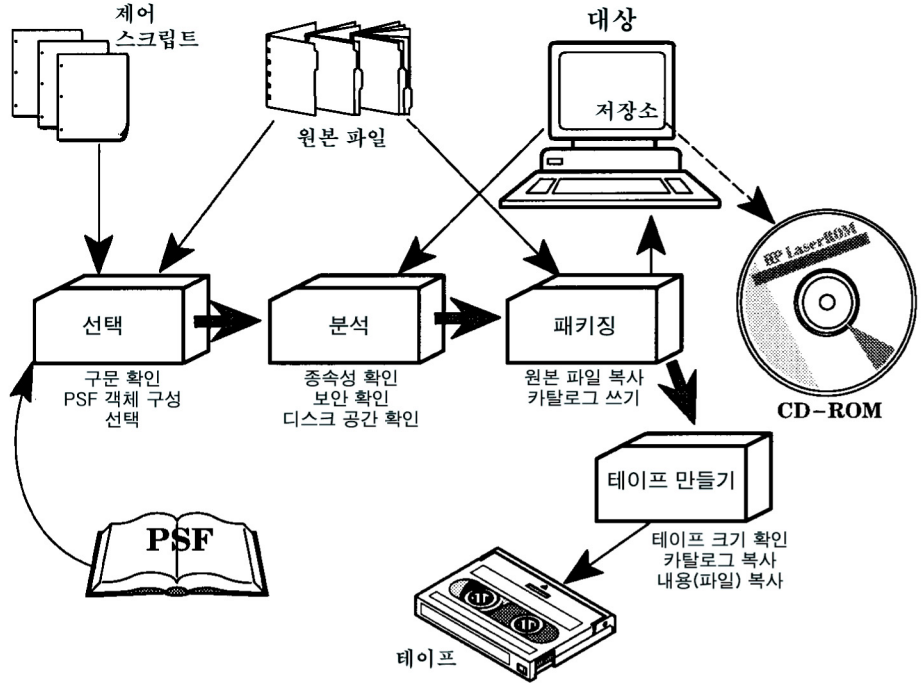
**swpackage 프로세스** `swpackage` 프로세스는 다음과 같은 4 단계로 구성됩니다.

**표 10-4** `swpackage` 프로세스 단계

<b>I. 선택</b>	<code>swpackage</code> 가 PSF를 읽습니다.
<b>II. 분석</b>	<code>swpackage</code> 가 대상 저장소나 테이프로 실제 소프트웨어를 패키징하기 전에 먼저 패키징 작업 및 요구 사항을 분석합니다. <code>swpackage</code> 는 패키징될 소프트웨어를 대상 저장소와 비교하여 패키징 작업이 정상적으로 이루어질 수 있도록 합니다.
<b>III. 빌드</b>	<code>swpackage</code> 는 소스 파일과 정보를 제품 객체에 패키징하고 해당 제품을 배포 저장소에 넣습니다. <code>swpackage</code> 는 저장소를 만들지만 등록하지는 않습니다. 로컬 호스트에 이러한 새로운 저장소를 만들기 위해서는 적절한 <b>SD-UX</b> 권한이 있어야 합니다. 대상(목적지)이 테이프 미디어인 경우 임시 저장소가 만들어집니다.
<b>IV. 테이프 만들기</b>	(선택) 이 단계는 배포 테이프로 패키징하는 경우에만 해당됩니다. <code>swpackage</code> 는 소스 파일과 임시 저장소 카탈로그를 테이프에 복사합니다. ( <code>swpackage</code> 는 테이프에 파일을 기록할 때 파일을 압축하지 못하므로 유의하십시오.)

그림 10-1, “패키징 프로세스 개요”에서는 `swpackage` 세션에 대한 개요를 보여줍니다.

그림 10-1 패키징 프로세스 개요



**단계 I: 선택**

swpackage를 실행할 때 PSF 및 포함시키려는 다른 옵션을 지정해야 합니다. swpackage 명령은 다음과 같은 작업에 사용된 소스, 소프트웨어 선택 사항 및 옵션을 사용자에게 알려주며 세션을 시작합니다.

- 구조에 필요한 제품, 하위 제품 및 파일 세트를 결정합니다.
- 각 파일 세트에 포함될 파일을 결정합니다.
- 각 객체와 연관되는 속성을 결정합니다.
- PSF 구문을 검사하고 오류가 있는 경우 세션을 종료합니다.

## 단계 II: 분석

swpackage는 이 단계에서 네 가지 검사를 수행합니다.

### 1. 분석되지 않은 종속성에 대한 검사

선택된 각 제품의 모든 파일 세트에 대해 swpackage는 해당 파일 세트의 필수 항목이 선택되지 **않았는지** 또는 대상 저장소에 아직 존재하지 **않는지**를 검사합니다. 제품 내 분석되지 않은 종속성이 있는 경우 오류가 발생합니다. 제품 전체에서 분석되지 않은 종속성이 있는 경우 주의를 필요로 합니다.

### 2. 제품을 패키징(또는 다시 패키징)하기 위한 권한을 검사

새로운 각각의 제품(대상 저장소에 존재하지 **않는** 제품)에 대해 swpackage는 대상 저장소를 검사하여 사용자가 대상 저장소에 새로운 제품을 만들 수 있는 권한(삽입 권한)을 가지고 있는지 확인합니다. 권한이 없으면 제품이 선택되지 않습니다.

각각의 기존 제품(다시 패키징하고 있는)에 대해 swpackage는 사용자가 이를 변경할 수 있는 권한(쓰기 권한)을 가지고 있는지 확인합니다. 권한이 없으면 제품은 선택되지 않습니다.

권한이 거부되어 모든 제품이 선택되지 않은 경우 세션에 오류가 발생하고 세션이 종료됩니다.

저장소가 새로 만든 저장소거나 테이프로 패키징하는 경우 이 권한 검사는 생략됩니다. 새로운 저장소를 만들 수 있는 권한을 가지고 있는 경우 그 안에서 제품을 만들 수 있는 권한도 갖게 됩니다. 테이프 세션은 먼저 임시 저장소에 기록한 다음 테이프로 복사하므로, 새로운(임시) 저장소를 만들 수 있는 권한을 가지고 있는 경우 테이프로 패키징할 수 있습니다.

### 3. 다시 패키징되는 소프트웨어 검사

선택된 각 제품에 대해 swpackage는 제품이 이미 대상 저장소에 있는지 여부를 확인합니다.

- 존재하는 경우 swpackage는 어떤 파일 세트가 추가되거나(새로운 파일 세트) 변경되는지를 검사합니다.
- 이미 대상 저장소에 있고 모든 파일 세트가 선택된 경우 swpackage는 기존 파일 세트가 새로운 제품에 의해 대체되었는지 여부를 검사합니다.

#### 4. 디스크 공간 분석(DSA) 수행

`swpackage`는 선택된 제품을 패키징하기 위한 충분한 공간이 대상 저장소에 있는지 확인합니다.

- 패키징 작업을 수행하기에 적절한 디스크 공간을 사용할 수 있으면 `swpackage`는 디스크 공간에 미치는 영향을 로그 파일에 기록합니다.
- 패키지가 디스크의 최소 여유 공간을 침해하는 경우 오류가 발생합니다.
- 사용할 수 있는 디스크 공간보다 더 많은 용량이 패키지 단계에서 필요하게 되면 오류가 발생합니다.
- `enforce_dsa` 명령 옵션을 **false**로 설정하면 `swpackage`는 디스크 공간 오류를 경고로 수정하고 계속 작업을 수행합니다. 이를 통해 최소 여유 공간을 통과하여 패키징 작업을 완료할 수 있습니다.

### 단계 III: 빌드

제품을 패키징할 때 대상 저장소가 없으면 `swpackage`가 저장소를 만듭니다. 대상 저장소가 있으면 `swpackage`는 새로운 제품이 이 저장소에 병합합니다. 각기 다른 버전의 제품인 경우 동일한 태그를 갖는 모든 제품 버전에 대해 정의된 제품 태그 속성 및 고유한 인스턴스 번호(인스턴스 ID)를 사용하여 디렉토리가 만들어집니다.

새로운 저장소 디렉토리가 만들어지기 전에 `swpackage`는 이 제품 버전이 기존 제품 버전과 동일한 식별 속성을 가지고 있는지 검사합니다.

식별 속성이 모두 일치하면 기존 버전을 다시 패키징(변경)하게 됩니다. 그렇지 않으면 `swpackage`는 대상 배포 시 새로운 버전을 만듭니다.

패키징 프로세스는 명시적으로 순서를 지정하여 치명적인 오류가 발생하더라도 대상 배포가 손상되는 것을 막을 수 있도록 합니다. 각 제품은 전체로 패키징되며 지정된 모든 제품이 정상적으로 패키징되면 배포의 글로벌 INDEX 파일이 빌드/재빌드됩니다. 각 제품 구조 내에서 다음과 같은 순서가 적용됩니다.

1. 제품이 새로 만들어진 것인지 기존에 있던 것인지 확인합니다. 새로 만들어진 제품이면 제품의 저장소 디렉토리를 만듭니다.
2. 제품의 각 파일 세트의 경우 파일 세트의 파일을 저장소 위치(제품의 저장소 디렉토리 내)에 복사하고 파일 세트의 카탈로그(데이터베이스 정보) 파일을 작성합니다.
3. 개별 파일 세트를 만든 다음에는 제품의 정보 파일(메타 파일)을 만듭니다.



대상 저장소는 CD-ROM 작성에 있어 맨 처음 단계일 뿐입니다. ISO 9660 표준 형식이 필요한 경우 이러한 변환을 수행하기 위한 유틸리티가 필요합니다. `swpackage`는 이러한 변환을 지원하지 않습니다.

배포 테이프는 `tar` 형식으로 만들어집니다. (`SD-UX` 명령은 `cpio` 형식의 테이프에서 저장소를 읽을 수도 있습니다.) 테이프를 만들기 위해 `swpackage`는 먼저 제품을 임시 배포 저장소에 빌드합니다. (`swpackage`가 완료되면 저장소가 제거됩니다.) 공간을 절약하기 위해 모든 파일은 실제 소스 파일에 대한 참조 형태로 존재합니다. 배포 저장소가 만들어진 다음 `swpackage`는 이를 실제 파일과 함께 테이프 저장소에 아카이브합니다.

커널 파일 세트가 들어 있는 제품을 테이프 미디어에 아카이브할 때 `swpackage`는 이러한 파일 세트를 먼저 아카이브에 넣어 `swinstall`이 효율적으로 액세스할 수 있도록 합니다. 또한 `swpackage`는 전제 조건의 종속성 관계에 따라 파일 세트의 순서를 정합니다.

**단계 IV: 테이프 만들기** 이 단계는 선택적으로 배포 테이프에 패키징할 때에만 해당됩니다.

- 이 단계에서 `swpackage`는 소스 파일과 임시 저장소 카탈로그를 테이프에 복사합니다.
- `swpackage`는 테이프 공간을 계산하여 테이프에 소프트웨어 패키지가 올바르게 들어갈 수 있도록 합니다. 테이프 하나에 모두 들어가지 않는 경우 `swpackage`는 여러 테이프에 소프트웨어를 나누어 넣습니다.
- `swpackage`는 테이프에 기록할 때 파일을 압축하지 않습니다.

## swpackage 사용

### swpackage 구문

```
swpackage [-p] [-v] [-V] [-C session_file]
[-d directory|device] [-f software_file]
[-s product_specification_file|directory]
[-S session_file] [-x option=value] [-X option_file]
[software_selections] [@ target_selection]
```

### 옵션 및 피연산자

- p                    실제로 저장소나 테이프를 작성 또는 변경하지 않고 지정된 패키지 세션을 미리 봅니다.
- v                    상세 출력을 stdout으로 내보내고 패키징되는 각 제품, 하위 제품 및 파일 세트에 대한 메시지를 나열합니다.  
(`/var/adm/sw/swpackage.log`의 `swpackage` 로그 파일은 이 옵션의 영향을 받지 않습니다.)
- V                    `swpackage`가 읽을 수 있는 데이터 모델 버전을 표시합니다.  
`swpackage`는 항상 최신 데이터 모델 버전을 사용하여 패키징합니다.
- C *session\_file*    명령을 실행하고 현재 옵션과 피연산자 값을 다른 세션에 다시 사용할 수 있도록 **session\_file**에 저장합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.
- d *directory|device*    배포 디렉토리를 만드는 경우 이 옵션은 *directory*의 경로를 이름을 정의합니다.  
  
배포 테이프를 만드는 경우 이 옵션은 배포를 작성할 *device* 파일을 정의합니다. 배포 테이프를 만들 때 테이프 장치(파일)가 존재해야 하며 `target_type=tape` 옵션을 지정해야 합니다.
- f *software\_file*    명령줄 대신(또는 그에 추가하여) 별도의 파일에서 대상 선택 목록을 읽습니다. 자세한 내용은 60페이지의 “소프트웨어 파일”을 참조하십시오.
- s *product\_specification\_file|directory*    사용할 **PSF** 또는 패키징 세션의 소스로 사용할 기존 디렉토리를 지정합니다.

*-S session\_file*

*session\_file*에 정의된 대로 이전 설치 세션에서 저장된 값을 바탕으로 명령을 실행합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.

*-x option=value*

명령 *option*을 *value*로 설정하고 기본값 또는 옵션 파일의 값을 대체합니다. 자세한 내용은 332페이지의 “명령 옵션 변경”을 참조하십시오.

*-X option\_file*

*option\_file*에서 세션 옵션 및 기능을 읽습니다. 자세한 내용은 332페이지의 “명령 옵션 변경”을 참조하십시오.

*software\_selections*

설치될 소프트웨어 객체. 자세한 내용은 58페이지의 “소프트웨어 선택 사항”을 참조하십시오.

이 사양을 포함시키지 **않으면** *swpackage*는 PSF에 나열되어 있는 제품을 모두 패키징합니다.

*@ target\_selections*

명령의 대상. 자세한 내용은 60페이지의 “대상 선택 사항”을 참조하십시오.

배포 저장소(디렉토리)를 만드는 경우 이 피연산자는 *directory*의 위치를 정의합니다. 이 피연산자를 지정하지 않으면 */var/spool/sw*가 기본 저장소 디렉토리로 사용됩니다.

배포 테이프를 만드는 경우 이 피연산자는 tar 아카이브를 기록하는 *device* 파일의 이름을 지정합니다. *swpackage*는 미디어가 DDS 테이프인지 디스크 파일인지를 판별할 수 있습니다. 이 피연산자를 지정하지 않으면 *swpackage*는 장치 파일 */dev/swtape*를 사용합니다.

### 명령 옵션 변경

명령을 호출할 때 명령줄 옵션을 추가로 지정(-x 옵션 사용)하거나 파일에서 미리 정의된 값을 읽어서 이 명령의 작동 방식을 변경할 수 있습니다. 다음 표에서는 `swconfig`에 적용되는 기본값과 옵션을 보여줍니다.

표 10-5 `swpackage` 명령 옵션 및 기본값

- `admin_directory=/var/spool/sw`
- `allow_partial_bundles=true`
- `compress_command=/usr/contrib/bin/gzip`
- `compress_files=false`
- `compress_index=false`
- `compression_type=gzip`
- `create_target_acls=true`
- `distribution_source_directory=/var/spool/sw`
- `distribution_target_directory=/var/spool/sw`
- `distribution_target_serial=/dev/rmt/0m`
- `enforce_dsa=true`
- `follow_symlinks=false`
- `include_file_revisions=false`
- `layout_version=1.0`
- `log_msgid=0`
- `logdetail=false`
- `logfile=/var/adm/sw/swpackage.log`
- `loglevel=1`
- `media_capacity=1330`
- `media_type=directory`
- `package_in_place=false`
- `reinstall_files=true`
- `reinstall_files_use_cksum=true`
- `run_as_superuser=true`
- `software=`
- `source_files=psf`
- `source_type=directory`
- `targets=`
- `uncompress_cmd=`
- `verbose=`
- `write_remote_files=false`

### 추가 정보

각 기본값에 대한 자세한 설명은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.

## 로그 파일 메시지 출력

/var/adm/sw/swpackage.log 로그 파일은 swpackage 세션의 출력을 캡처합니다.

- 메시지 로깅은 기본적으로 상세 메시지를 stdout으로 보냅니다.  
(verbose 옵션을 0으로 설정하면 stdout으로 출력되는 정보의 양을 감소시킬 수 있습니다.)
- 메시지 로깅은 기본적으로 오류와 경고를 stderr로 보냅니다.
- 로그 파일 메시지는 미리 보기(-p) 모드에서는 기록되지 않습니다.
- 로그 파일은 stdout과 stderr를 결합한 것과 같습니다.

다음은 샘플 로그입니다.

```
===== 01/27/01 18:58:45 MST BEGIN swpackage SESSION
* Session started for user "root@sdtest.myco.com".

* Source:          vewd:test.psf
* Target:          vewd:/var/spool/sw
* Software selections:
*
* Options:
  preview          true
  verbose          1
  loglevel         1
  logfile          /var/adm/sw/swpackage.log

  source_type      file
  target_type      directory

  package_in_place false
  follow_symlinks  false
  include_file_revisions false
  enforce_dsa      true
  reinstall_files  true
  reinstall_files_use_cksum false
  write_remote_files false
  create_target_acls true
```

```
* Beginning Selection Phase.  
* Reading the Product Specification File (PSF) "test.psf".  
* Reading the product "SD" at line 1.  
* Reading the fileset "commands" at line 4.  
===== 01/27/01 18:58:45 MST  END swpackage SESSION
```

---

## 패키징 작업 및 예

PSF `product.psf`에 정의된 소프트웨어 제품을 배포 저장소 `/var/spool/sw`에 패키징하고 작업을 실제 수행하기 전에 상세 수준에서 작업을 미리 보려면 다음과 같이 입력합니다.

```
swpackage -p -v -s product.psf @ /var/spool/sw
```

### swpackage로 만든 저장소 등록

swpackage로 새 저장소를 만들 때 로컬 호스트의 `swagentd` 데몬에 자동으로 등록되지 않습니다.

저장소가 등록되었는지 확인하려면 다음과 같이 입력합니다.

```
swlist -l depot @ MyDepot
```

저장소를 등록하려면 `swreg` 명령을 실행해야 합니다.

```
swreg -l depot depot_to_register
```

저장소를 등록하면 `swinstall`과 `swcopy` 작업에서 이 저장소를 소스로 사용할 수 있습니다.

등록을 통해 패키징된 저장소를 공식적으로 알릴 수 있게 됩니다.

- `swinstall/swcopy` GUI에서 저장소를 볼 수 있으며 `swlist` 저장소 수준 목록에서도 볼 수 있습니다.
- 저장소에서 제품을 입을 수 있습니다(예를 들면 설치하기 위해).

저장소 등록에 대한 자세한 내용은 153페이지의 “저장소 등록 및 등록 해제(`swreg`)”를 참조하십시오.

---

## 참고

패키징한 사용자가 `swpackage`로 만든 저장소를 로컬에서만 액세스하는 경우에는 저장소를 등록할 필요가 없습니다.

## CD-ROM 저장소 작성 및 마스터

swpackage가 새로운 저장소를 만들거나 새로운 제품을 패키징할 때 해당 저장소/제품에 대해 항상 ACL을 만듭니다. 저장소를 작성한 다음 CD-ROM에 마스터해야 하는 경우 그 CD-ROM에는 모든 ACL이 포함되는데, 이는 다음과 같은 문제를 일으킬 수 있습니다.

- CD-ROM 저장소에 부여된 권한이 너무 제한적입니다.
- CD-ROM에 사용자 고유 ACL이 너무 많이 포함될 수 있습니다.

이러한 문제를 해결하기 위해 swpackage에서 create\_target\_acls 옵션을 false로 설정함으로써 저장소에서 ACL을 작성하지 **않도록** 지시할 수 있습니다.

이 기능은 로컬 슈퍼유저만이 ACL을 변경 및 삭제하고 ACL이 없는 저장소에 ACL을 추가할 수 있기 때문에 슈퍼유저에게만 제공됩니다. 로컬 슈퍼유저는 항상 모든 권한을 갖습니다.

create\_target\_acls를 false로 설정하면 swpackage는 패키징되는 각각의 새로운 제품에 대해 ACL 작성을 생략합니다(새로 작성하는 경우 저장소에 대해서도 생략함). 이 옵션은 이미 저장소에 들어 있는 ACL에는 아무런 영향도 미치지 않습니다.

저장소가 다른 SD-UX 작업의 소스로 사용되는 경우 ACL(또는 ACL 부재)은 이 작업의 대상에 대해 작성된 ACL과는 아무런 관계가 없습니다. 소스 ACL은 대상 ACL과 관련이 없습니다.

swpackage 명령은 소프트웨어를 테이프에 패키징할 때는 ACL을 작성하지 않습니다.



## 성능 향상을 위한 파일 압축

패키징 프로세스를 처리할 때 많은 양의 데이터를 네트워크를 통해 전달하는 경우가 있는데, 이는 네트워크 성능을 저하시킬 수 있습니다. `compress_files` 옵션을 통해 먼저 전송할 파일을 압축함으로써 성능을 향상시킬 수 있습니다. 성능이 어느 정도 향상되는지는 전송되는 파일의 종류에 따라 다릅니다. 바이너리 파일은 50% 이하로 압축되고 텍스트 파일은 일반적으로 좀 더 압축됩니다. 속도가 느린 네트워크(초당 약 50KB 이하)에서 전송될 때 성능이 가장 큰 폭으로 개선됩니다.

`true`로 설정하면 `compress_files`는 소스에서 전송하기 전에 파일을 압축합니다(이전에 `SD-UX`가 이미 압축하지 않은 경우). `compression_type` 옵션으로 압축 유형을 지정하거나 압축 명령에 `compression_command` 옵션을 지정할 수도 있습니다.

이 옵션은 네트워크 대역폭이 전체 처리량을 명확하게 제한하는 경우에만 `true`로 설정해야 합니다. 이 옵션이 도움이 될 수 있는지 확실하지 않다면 이 옵션을 지속적으로 사용하기 전에 압축한 패키징 옵션과 압축하지 않은 패키징 옵션을 비교하여 보십시오. 명령 옵션 사용에 대한 자세한 내용은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.

---

### 참고

---

`swpackage`는 테이프에 기록할 때 파일을 압축하지 않습니다.

## 패키징 보안

`SD-UX`는 액세스 제어 목록(ACL)을 제공하여 저장소에서 특정 작업을 수행하기 위한 권한을 부여합니다. `swpackage` 명령은 로컬 저장소만 작성하고 수정하므로 `SD-UX`에서 제공하는 원격 작업에 관련된 보안 기능은 `swpackage`에는 적용되지 않습니다. ACL에 대한 자세한 내용은 9장, 233페이지의 “SD-UX 보안”을 참조하십시오.

`swpackage` 명령은 `setuid root`로서 작동하는데, 즉 패키지 선택 단계에서는 호출 사용자 권한으로 수행하고 분석 및 패키징 단계에서는 수퍼유저 권한으로 수행합니다. 수퍼유저는 모든 저장소를 소유하고 관리하며 저장소에서의 모든 작업에 대한 모든 권한을 갖습니다. 저장소가 NFS 볼륨에 있는 경우 ACL로 인한 액세스 문제는 발생하지 않지만, 로컬 수퍼유저가 NFS에 마운트된 파일 시스템에 대한 NFS 루트 액세스를 가지고 있지 않은 경우 문제가 발생할 것입니다.

사용자가 로컬 수퍼유저가 아닌 경우 로컬 수퍼유저가 권한을 부여한 경우를 제외하고는 저장소를 작성하거나 변경할 수 있는 권한을 갖지 못합니다.

swpackage는 다음과 같은 권한을 점검하고 실행합니다.

**1. 새로운 저장소를 만들 수 있습니까?**

수퍼유저           예

기타                 로컬 호스트의 ACL이 사용자에게 “삽입” 권한, 즉 호스트에 새로운 저장소를 삽입할 수 있는 권한을 부여한 경우에는 그렇습니다.

적절한 권한이 부여되지 않고 저장소가 새로 만들어진 것이라면 swpackage에 오류가 발생하고 종료됩니다.

**2. 새로운 제품을 만들 수 있습니까?**

수퍼유저           예

기타                 저장소가 새로 만들어졌고 위의 #1 항목을 통과했거나 기존 저장소에 대한 ACL이 삽입 권한, 즉 저장소의 내용을(새로운 제품을 추가함으로써) 변경할 수 있는 권한을 부여한 경우에는 그렇습니다.

새로운 제품을 만들 수 있는 권한이 거부된 경우 swpackage는 오류 메시지를 표시하고 해당 세션에서 그 제품을 제외시킵니다.

**3. 기존 제품을 변경할 수 있습니까?**

수퍼유저           예

기타                 기존 제품에 대한 ACL이 쓰기 권한 즉, 제품의 내용을 덮어쓰거나 변경할 수 있는 권한을 부여한 경우에는 그렇습니다. 기존 제품을 변경할 수 있는 권한이 거부된 경우

swpackage는 오류 메시지를 표시하고 해당 세션에서 그 제품을 제외시킵니다.

선택된 모든 제품에 대해 삽입 및 쓰기 권한이 거부된 경우 swpackage에 오류가 발생하고 종료됩니다.

**4. 저장소 수준의 속성을 변경할 수 있습니까?**

수퍼유저           예

기타

저장소가 새로 만들어졌고 위의 #1 항목을 통과했거나 기존 저장소에 대한 ACL이 쓰기 권한, 즉 저장소의 내용을 쓰거나 변경할 수 있는 권한을 부여한 경우에는 그렇습니다.

기존 저장소를 변경할 수 있는 권한이 거부되고 PSF에 저장소 수준의 속성을 지정한 경우 swpackage는 경고 메시지를 내보내고 저장소 속성을 변경하지 않습니다.

### ACL 작성

swpackage가 새로운 저장소나 새로운 제품을 만들 때 그에 대한 ACL도 작성합니다.

#### 새로운 저장소

swpackage는 저장소에 대해 ACL을 만들고 해당 저장소에 패키징될 모든 제품에 대한 템플릿 ACL을 만듭니다.

저장소 ACL은 호스트의 global\_soc\_template ACL(즉, 새로운 저장소와 새로운 루트 파일 시스템에 대해 설정된 템플릿 ACL)로부터 만들어집니다.

저장소의 product\_template ACL은 호스트의 global\_product\_template ACL(즉, 새로운 제품에 대한 호스트의 템플릿 ACL)로부터 만들어집니다.

swpackage를 실행하는 사용자는 새로운 저장소의 소유자로 설정되고 저장소 ACL(global\_soc\_template에서 생성)에 정의된 권한이 부여됩니다.

#### 새로운 제품

swpackage는 제품에 대한 ACL을 만듭니다. 이 ACL은 저장소의 product\_template ACL에서 생성됩니다.

create\_target\_acls 명령 옵션을 false로 설정함으로써 ACL 작성을 비활성화할 수 있습니다.

저장소에 대한 ACL이 없는 경우 슈퍼유저만이 새로운 제품을 만들고 저장소 속성을 추가/변경할 수 있습니다. 제품에 대한 ACL이 없는 경우 슈퍼유저만이 이를 변경할 수 있습니다.

## 소프트웨어 패키지의 변경 또는 다시 패키징

다시 패키징에는 두 종류가 있습니다.

### 1. 기존 제품에 파일 세트를 추가 또는 변경

- 새로운 파일 세트 정의를 추가하거나 기존 파일 세트 정의를 변경하여 PSF를 수정합니다.
- 수정된 PSF에서 `swpackage`를 실행하고 명령줄에서 새로운 또는 변경된 파일 세트를 지정합니다.

```
swpackage -s psf <other options> \  
product.fileset @ depot
```

이 명령은 하위 제품이 제품에 정의되어 있는지 여부와 상관없이 작동합니다.

- `tag` 속성을 변경하여 파일 세트를 변경하는 경우 `swpackage`는 기존의 오래된 파일 세트를 새로운 파일 세트와 연관시킬 수 없습니다. 둘 다 변경된 제품의 일부가 됩니다. 오래된(이름이 바뀐) 파일 세트를 제거하려면 `swremove`를 사용합니다.

```
swremove -d product.old_fileset @ depot
```

### 2. 기존 제품 전체를 변경

- 새로운 파일 세트 정의를 추가하거나, 기존 파일 세트 정의를 변경하거나, 기존 파일 세트 정의를 삭제하거나 제품의 정의(제품 수준 속성)를 변경함으로써 PSF를 수정합니다.
- PSF에서 `swpackage`를 실행하고 명령줄에서 제품을 지정합니다.

```
swpackage -s psf <other options> product @ depot
```

- PSF에서 일부 파일 세트 정의를 삭제했거나 파일 세트의 태그 속성을 변경하여 파일 세트를 수정한 경우 `swpackage`는 PSF의 수정된 제품 정의에 속하지 않는 기존 파일 세트에 대한 경고 메시지를 내보냅니다. PSF의 제품 정의에서 기존 파일 세트와 새로운 파일 세트는 모두 변경된 제품에 포함됩니다.

분석 단계에서 경고가 표시되고(제품 일부가 아닌) 제품 전체가 다시 패키징될 때에만 표시됩니다.

- 삭제된(이름이 바뀐) 파일 세트를 제거하려면 `swremove`를 사용합니다.

```
swremove -d product.old_fileset @ depot
```

- 변경 사항을 다시 패키징하기 전에 제품 전체를 `swremove`하고자 할 수 있습니다.

```
swremove -d product @ depot
```

```
swpackage -s psf <other options> product @ depot
```

## 현재 위치에서 패키징

`package_in_place` 옵션을 `true`로 설정하면 `swpackage`는 지정된 제품 각각을 패키징하여 소스 파일이 대상 저장소로 복사되지 않도록 합니다. 대신 `swpackage`는 각 파일 세트의 내용을 구성하는 소스 파일에 대한 참조를 삽입합니다. 제어 스크립트는 항상 복사됩니다.

이 기능을 통해 소스 파일을 대상 저장소로 복사하느라 디스크 공간을 모두 사용하지 않고도 개발 또는 테스트 환경에서 제품을 패키징할 수 있습니다.

`package_in_place` 옵션을 `true`로 설정하면 디스크 공간 분석이 생략됩니다.

소스 파일은 계속 남아 있어야 합니다. 일부 삭제되면 저장소를 소스로 사용하는 작업(예: `swinstall`을 사용한 제품 설치)에서 파일에 액세스하려 할 때 오류가 발생하게 됩니다.

소스 파일이 변경되고 제품이 다시 패키징되지 않는 경우 소스 파일을 설명하는 정보가 올바르지 않을 것입니다(예: `checksum` 파일). 이러한 잘못된 정보로 인해 대상 저장소를 소스로 사용(예: `swinstall`을 사용한 설치)할 수 없게 됩니다. 그러나 잘못된 정보는 저장소에서 제품이 복사되거나 설치될 때마다 퍼지게 됩니다. 그 결과 파일 내용 검사를 설정해 놓은 경우 설치된 제품에 대한 `swverify` 작업에서 항상 오류가 발생하고 불일치 플래그가 표시됩니다.

## 소스의 심볼릭 링크 따라가기

`follow_symlinks` 옵션을 `true`로 설정한 경우 `swpackage`는 심볼릭 링크이고 심볼릭 링크가 가리키는 파일을 패키징된 파일 세트에 포함하고 있는 모든 소스 파일을 따라갑니다.

`swpackage`는 또한 심볼릭 링크이며 `file *` 키워드(재귀적 파일 사양)의 작동에 영향을 미치는 각각의 소스 디렉토리를 따라갑니다. 패키징된 파일 세트에 심볼릭 링크만을 포함시키는 대신 심볼릭 링크가 가리키는 디렉토리 및 그 아래의 모든 파일이 패키징된 파일 세트에 포함됩니다.

이 옵션의 기본값은 `false`이며 이를 설정하면 소스에 나오는 심볼릭 링크가 심볼릭 링크로 패키징됩니다. 심볼릭 링크는 파일 세트에 속하는 파일과 파일 세트에 속하지 않는 파일을 모두 가리킬 수 있습니다.

## 파일 버전 생성

`include_file_revisions` 옵션을 `true`로 설정하면 `swpackage`는 `what` 및 `ident` 명령으로 각 소스 파일을 검사하여 `SCCS`나 `RCS` 버전 값을 추출한 다음 이를 파일의 `revision` 속성으로 지정합니다.

파일에는 버전 문자열이 들어 있을 수 있으므로 `swpackage`는 맨 처음 반환되는 값을 사용합니다. 버전 문자열 전체에서 버전 값을 추출하여 이를 저장합니다.

특히 `what` 검색에 실패한 다음 `ident` 명령이 실행되는 경우 이 옵션은 시간이 오래 걸립니다.

이 옵션의 기본값은 `false`이고 이 값을 지정하면 `swpackage`는 검사를 생략합니다. 패키징되는 파일에 `revision` 속성 값이 지정되지 않습니다.

## 원격 파일 시스템의 저장소

swpackage 분석 및 빌드 단계는 슈퍼유저 권한으로 작동하므로 swpackage가 NFS 마운트 파일 시스템에 있는 저장소에서 제품을 작성, 추가 또는 수정하는 방식에 제한이 있습니다.

슈퍼유저가 원격 파일 시스템에 대한 쓰기 권한을 가지고 있지 않다면 swpackage는 새로운 저장소를 만들지 못하고 분석 단계가 시작되기 전에 종료됩니다.

슈퍼유저가 원격 파일 시스템에 대한 쓰기 권한을 가지고 있지 **않고** write\_remote\_files 옵션이 **false**인 경우 swpackage는 새로운 저장소를 만들지 못하고 분석 단계가 시작되기 전에 종료됩니다.

슈퍼유저가 원격 파일 시스템에 대한 쓰기 권한을 **가지고 있고** write\_remote\_files 옵션을 **true**로 설정한 경우 swpackage는 새로운 저장소를 만들고 이 저장소에 제품을 패키징합니다.

기존 NFS에 마운트된 저장소에 적용되는 제한 사항은 새로운 저장소를 만들 때에도 마찬가지로 적용됩니다.

따라서 다음을 준수해야 합니다.

1. write\_remote\_files 옵션을 **true**로 설정합니다.
2. NFS에 마운트된 파일 시스템에서 저장소를 패키징하려면 슈퍼유저가 NFS 파일 시스템에 대한 쓰기 권한을 가지고 있어야 합니다.

이러한 제한 사항이 충족되면 ACL 보호 메커니즘은 로컬 저장소에서 작동될 때 제어하는 방식과 마찬가지로 NFS에 마운트된 저장소에서의 작동을 제어합니다.

## 소프트웨어 패키지 확인

swpackage가 기존의 등록된 저장소에 패키지를 저장하지 않고 저장소를 만들었다면 swreg 명령을 사용하여 이 저장소를 등록해야 합니다(335페이지의 “swpackage로 만든 저장소 등록” 참조).

저장소가 등록된 후에는 swverify 명령을 사용하여 이를 확인할 수 있습니다. 예를 들어 로컬 기본 저장소에서 Pascal 제품의 무결성을 확인하려면 다음과 같이 입력합니다.

```
swverify -d Pascal
```

저장소 확인에 대한 자세한 내용은 163페이지의 “저장소 확인(swverify -d)”을 참조하십시오.

또한 패키지를 시스템에 설치하여 테스트할 수도 있습니다. 예를 들어 svrhost 호스트의 기본 저장소인 /var/spool/sw에 있는 Pascal이라는 이름의 패키지를 myhost 호스트의 기본 루트에 설치하려면 다음과 같이 입력합니다.

```
swinstall -s svrhost Pascal @ myhost
```

(이 예에서 해당 소프트웨어는 svrhost의 기본 /var/spool/sw에 있다고 가정하므로 저장소 위치를 지정하지 않습니다.)

설치된 소프트웨어를 확인하는 것에 대한 자세한 내용은 91페이지의 “설치 확인(swverify)”을 참조하십시오.

## 패치 소프트웨어 패키징

패치 소프트웨어 패키징을 허용하는 모든 소프트웨어 수준(번들, 제품, 하위 제품 및 파일 세트)에서 몇 가지 소프트웨어 속성을 사용할 수 있습니다. 패치 속성 및 PSF 예제에 대한 자세한 내용은 5장, 165페이지의 “HP-UX 패치 및 패치 관리”를 참조하십시오.



## 여러 테이프에 쓰기

제품을 배포 테이프에 패키징할 때 `media_capacity` 옵션은 테이프 미디어의 크기 (100만 바이트 단위)를 정의합니다. 이 옵션의 기본값은 `media_capacity=1330`이고 이는 HP DDS 테이프의 크기입니다. 대상 테이프가 DDS 테이프가 아닌 경우 `media_capacity` 값을 지정해야 합니다.

---

### 참고

DDS 테이프의 용량은 메가바이트(1,048,576바이트) 단위가 아니고 100만 (1,000,000)바이트 단위입니다. 대부분의 테이프 드라이브 제조업체에서는 백만 단위로 용량을 지정합니다.

패키징되는 제품이 지정된 미디어 용량보다 더 많은 용량을 필요로 하는 경우 `swpackage`는 제품을 여러 테이프에 나누어 패키징할 수 있습니다.

테이프가 여러 개 필요한지 판별하기 위해 `swpackage`는 저장소 카탈로그와 각각의 제품 내용을 저장하는 데 필요한 테이프 블록을 계산합니다.

테이프가 여러 개 필요한 경우 `swpackage`는 카탈로그 전체를 첫 번째 테이프에 넣고 나머지 용량에 맞추어 제품 내용을 기록합니다. 다음 테이프를 준비할 때마다 매번 `swpackage`는 계속 수행하기 전에 사용자가 “**tape is ready**” 응답을 하도록 프롬프트를 표시합니다.

다음 테이프에 계속 내용을 쓰려면 다음 응답 중 하나를 입력합니다.

**Return**                    같은 장치를 사용합니다.

**경로 이름**                새로운 장치/파일 `pathname`을 사용합니다.

**quit**                        테이프에 쓰기 작업을 종료합니다.

파티션은 파일 세트 수준에서 이루어지므로 제품은 여러 테이프에 나뉘어 존재할 수 있습니다. 한 파일 세트의 내용이 여러 테이프에 나뉘어 있을 수는 없습니다. 한 파일 세트의 크기가 미디어 용량을 초과하는 경우 `swpackage`에 오류가 발생하고 종료됩니다. 카탈로그가 첫 번째 테이프에 들어가지 않는 경우에도 오류가 발생합니다.

## 기존 저장소에서 테이프 만들기

swpackage를 사용하여 기존 저장소에서 제품을 한 개 이상 복사할 수 있습니다. PSF를 패키징 세션의 소스로 지정하는 대신 기존 저장소를 지정하면 됩니다. 예를 들면 다음과 같습니다.

```
swpackage -s /var/spool/sw ...
```

저장소에 있는 제품을 모두 테이프로 복사하려면 다음과 같이 입력합니다.

```
swpackage -s depot -d tape -x target_type=tape
```

저장소에 있는 일부 제품만을 테이프로 복사하려면 소프트웨어 선택 시와 마찬가지로 제품을 지정합니다.

```
swpackage -s depot -d tape -x target_type=tape \  
  product1 product2 ...
```

-f *file* 옵션을 사용하여 명령줄에 나열하는 대신 소프트웨어 선택 사항을 여러 개 지정할 수도 있습니다.

저장소에서 테이프로 제품이 복사될 때 저장소 내의 ACL은 복사되지 **않습니다**.

(swpackage 명령은 소프트웨어를 테이프로 패키징할 때는 ACL을 작성하지 않습니다.)

swpackage는 테이프로 기록할 때 파일을 압축하지 않습니다.

# 11 제어 스크립트 사용

이 장에서는 제어 스크립트 사용 방법에 대해 설명합니다.

표 11-1

이 장의 항목

설명하는 항목
349페이지의 “제어 스크립트의 종류”
359페이지의 “환경 변수 사용”
365페이지의 “제어 스크립트의 실행”
374페이지의 “제어 스크립트에 의한 다른 명령 실행”
375페이지의 “제어 스크립트의 입력 및 출력”
379페이지의 “제어 스크립트에 의한 파일 관리”
380페이지의 “제어 스크립트 테스트”
385페이지의 “사용자 응답 요구(swask)”

## 제어 스크립트에 대한 소개

SD-UX에서는 제품 및 파일 세트 **제어 스크립트** 실행을 모두 지원합니다. 이러한 셸 스크립트를 통해 사용자 정의된 추가적인 점검 및 작업을 일상적인 소프트웨어 관리 작업에 포함시켜 수행할 수 있습니다. `swinstall`, `swconfig`, `swverify`, `swask` 및 `swremove` 명령은 제어 스크립트를 하나 이상 실행할 수 있습니다. 제어 스크립트는 보통 소프트웨어 공급업체로부터 제공받지만 사용자가 직접 작성할 수도 있습니다. 모든 제어 스크립트는 선택 가능합니다.

**제품 수준의 제어 스크립트**는 설치, 구성, 확인 또는 제거를 위해 해당 제품 내의 파일을 선택할 때 실행됩니다. 따라서 해당 제품 제어 스크립트의 작업에는 특정 파일 세트가 아니라 해당 제품의 모든 파일 세트와 관련되어 있어야 합니다. 제품 내 모든 파일 세트에 적용할 조치는 적절한 제품 수준 제어 스크립트에 지정되어야 합니다.

**파일 세트 스크립트**는 상위 제품 또는 다른 파일 세트가 아니라 해당 파일 세트의 설치, 구성 또는 제거와 관련되어 있어야 합니다.

제어 스크립트를 통해 다음과 같은(그러나 이에 제한되지 않는) 광범위한 사용자 정의 및 구성 작업을 수행할 수 있습니다.

- 다른 사용자가 제품을 사용 중인지 확인하고 그런 경우 재설치, 업데이트 또는 제거하지 않도록 합니다.
- 로컬 호스트 시스템이 소프트웨어와 호환되는지 확인합니다. (스크립트는 제품의 `uname` 속성을 사용하여 호환성을 확인할 수 있습니다.)
- 제품의 이전 파일이나 이전에 설치된 버전을 제거합니다.
- 파일이 설치된 후 이 파일에 대한 링크나 추가 복사본을 작성합니다.
- 처음 설치할 때 구성 가능 파일을 적절한 위치에 복사합니다.
- 이후 업데이트할 때 구성 가능 파일을 적절한 위치에 조건부로 복사합니다.
- 새로운 기능을 반영하도록 기존 구성 파일을 수정합니다.
- 구성 파일의 사용자 정의 버전을 다시 빌드합니다.
- 장치 파일이나 사용자 정의 프로그램을 작성합니다.
- 데몬을 종료하거나 시작합니다.

## 제어 스크립트의 종류

다음은 SD-UX가 지원하는 제어 스크립트입니다.

- **Checkinstall 스크립트**

이 스크립트는 분석 단계에서 `swinstall`에 의해 실행되며 설치 및 구성을 시도할 수 있도록 보장합니다. 예를 들면 OS 실행 상태, 실행 중인 프로세스 또는 종속성과 관련 없는 기타 전제 조건 등을 검사합니다. 시스템 상태는 변경되지 않습니다.

`checkinstall` 스크립트의 최대 장점은 선택한 소프트웨어 설치를 진행하도록 허용된 경우에 재난, 즉 시스템 부팅 불능 또는 파일 시스템 손상을 일으킬 수 있는 하드웨어 구성이 시스템에 포함되어 있는지 감지할 수 있다는 점입니다. 또한 다른 소프트웨어 선택이나 이미 설치된 소프트웨어와의 충돌을 미리 예상할 수 있는 테스트로서의 역할도 수행합니다.

- **Preinstall 스크립트**

이 스크립트는 소프트웨어 파일을 로드하기 전에 `swinstall`에 의해 실행됩니다. 예를 들어 이 스크립트는 업데이트 시 오래된 파일을 제거하거나 기존 파일을 이동시킵니다.

`preinstall` 스크립트는 `swinstall` 명령의 실행 단계에서 호출됩니다. 각 파일에 대한 `preinstall` 스크립트는 파일 세트의 파일이 대상 시스템에 설치되기 직전에 실행됩니다. 제품 수준의 `preinstall` 스크립트는 제품의 파일 세트보다 먼저 호출됩니다.

모든 커널 파일 세트 및 해당 전제 조건에 대한 `preinstall` 스크립트는 모두 커널 빌드가 발생하기 전에 실행됩니다. 커널 빌드에 실패하고 `swinstall`이 종료되면 `preinstall` 스크립트는 시스템에서 제거됩니다. 제품 수준의 `preinstall` 스크립트는 커널 파일 세트를 포함하는 모든 제품에 대해 두 번 호출됩니다. 한 번은 커널 파일 세트와 전제 조건이 설치될 때이고 두 번째는 나머지 파일 세트가 설치될 때입니다.

- **Postinstall 스크립트**

이 스크립트는 소프트웨어 파일을 로드한 후 `swinstall`에 의해 실행됩니다. 예를 들어 이 스크립트를 통해 기본 파일을 제 위치로 이동시킬 수 있습니다.

`postinstall` 스크립트는 `swinstall` 명령의 로드 단계 부분입니다. 파일이 로드 되면 파일 세트의 `postinstall` 스크립트가 실행됩니다. 그러면 제품의 `postinstall` 스크립트(있는 경우)가 실행됩니다.

- **Unpreinstall 스크립트**

`unpreinstall` 스크립트는 복구가 시작된 경우 `swinstall`의 로드 단계에서 실행됩니다.

모든 취소 스크립트는 일반 스크립트의 실행 순서와 반대로 실행됩니다. 복구할 각 파일 세트에 대해 `unpostinstall` 스크립트가 실행되고 파일 세트 파일이 복원된 다음 `unpreinstall` 스크립트가 실행됩니다. 해당 스크립트가 실행되었다면 그에 대한 취소 스크립트가 실행됩니다.

`unpreinstall` 스크립트는 `preinstall` 스크립트가 실행한 모든 작업을 취소합니다. 예를 들어 `preinstall` 스크립트가 파일을 이동했으면 `unpreinstall` 스크립트는 이를 다시 원래 위치로 이동시킵니다. `preinstall` 스크립트가 파일을 복사했으면 `unpreinstall` 스크립트는 이를 제거합니다.

복구 가능한 제품의 경우 `preinstall` 또는 `postinstall` 스크립트로 파일이 제거되지 않습니다. 오래된 파일을 제거할 때는 `configure` 스크립트를 사용하는 것이 좋습니다.

제품에 대한 `unpreinstall` 스크립트는 파일 세트에 대한 `unpreinstall` 스크립트 다음에 실행합니다.

- **Unpostinstall 스크립트**

`unpostinstall` 스크립트는 복구가 시작된 경우 `swinstall`의 로드 단계에서 실행됩니다.

모든 취소 스크립트는 일반 스크립트의 실행 순서와 반대로 실행됩니다. 해당 스크립트가 실행되었다면 그에 대한 취소 스크립트가 실행됩니다.

`unpostinstall` 스크립트는 `postinstall` 스크립트가 실행한 모든 작업을 취소합니다. 예를 들어 `postinstall` 스크립트가 파일을 이동했으면 `unpostinstall` 스크립트는 이를 다시 원래 위치로 이동시킵니다. `postinstall` 스크립트가 파일을 복사했으면 `unpostinstall` 스크립트는 이를 제거합니다.

복구 가능한 제품의 경우 `preinstall` 또는 `postinstall` 스크립트로 파일이 제거되지 않습니다. 오래된 파일을 제거할 때는 `configure` 스크립트를 사용하는 것이 좋습니다.

---

## 참고

---

제품 수준의 `unpostinstall` 스크립트는 지원되지 않습니다.

- **Configure 스크립트**

이 스크립트는 `swinstall`이나 `swconfig`에 의해 실행되어 소프트웨어에 호스트를 구성하거나 소프트웨어에 호스트 특정 정보를 구성합니다. 예를 들어 이 스크립트는 `/etc/services`와 같은 호스트 특정 구성 파일을 변경하거나 호스트 이름 또는 사용 가능한 프린터와 같은 다른 호스트 리소스를 구성 파일에 추가하거나 컴파일을 수행할 수 있습니다.

`configure` 스크립트는 제품의 로드 단계가 완료된 **후에** 모든 제품에 대해(전제 조건 순으로) `swinstall` 명령으로 실행됩니다. 그러나 이 스크립트는 소프트웨어를 실제로 사용할 시스템에 설치될 때에만 실행됩니다. 이 스크립트가 대체 루트에 설치될 때(예를 들어 디스크 없는 파일 시스템이나 테스트 파일 시스템 구축 시)에는 지연되며, 대체 루트가 해당 소프트웨어를 사용하는 시스템의 루트일 경우에는 대신 `swconfig` 명령에 의해 실행됩니다.

정상적인 설치 시 실패한 `configure` 스크립트를 다시 실행시킬 때에도 `swconfig` 명령을 사용할 수 있습니다. 구성 단계 실행이 정상적으로 완료되면(스크립트가 있든 없든) 소프트웨어는 설치 상태에서 구성 상태 또는 사용 대기 상태로 바뀝니다. `configure` 스크립트 및 다른 모든 스크립트는 여러 번 실행할 수 있어야 합니다(즉, 재실행이 가능해야 합니다).

오래된 파일을 제거할 때는 `configure` 스크립트를 사용하는 것이 좋습니다.

대체 루트에 설치하는 경우에는 `configure` 스크립트가 실행되지 않습니다.

- **Verify 스크립트**

`verify` 스크립트는 소프트웨어가 설치되고 구성된 후 언제든지 `swverify` 명령으로 실행할 수 있습니다. 다른 스크립트와 마찬가지로 `verify` 스크립트는 SD-UX 소프트웨어 관리 도구가 기본적으로 확인하지 않는 사항을 확인하기 위한 것입니다. 예를 들어 이 스크립트로 소프트웨어가 올바르게 구성되었는지 이 소프트웨어를 사용할 수 있는 적절한 사용권을 가지고 있는지 등을 확인할 수 있습니다.

- **Fix 스크립트**

swverify로 실행되는 **fix** 스크립트를 정의하여 설치된 소프트웨어에서 문제를 해결하고 보고할 수 있습니다. **fix** 스크립트는 유실된 디렉토리를 작성하고, 파일 변경 사항(모드, 소유자, 그룹, 1차 및 2차)을 수정하며 심볼릭 링크를 다시 작성합니다.

- **Unconfigure 스크립트**

이 스크립트는 swconfig나 swremove에 의해 실행되어 **configure** 스크립트로 수행한 원래 호스트 구성 또는 소프트웨어 구성을 취소합니다. 예를 들어 **unconfigure** 스크립트는 /etc/services 파일에서 구성을 제거할 수 있습니다. (구성 취소 작업을 통해 소프트웨어는 구성 상태에서 다시 설치 상태로 되돌아갑니다.)

swconfig를 사용하여 **unconfigure** 스크립트를 실행할 수 있지만 swremove 명령에 의해서만 소프트웨어를 실제로 제거할 수 있습니다. 대체 루트에서 제거하는 경우에는 **unconfigure** 스크립트가 실행되지 않습니다.

- **Checkremove 스크립트**

**checkremove** 스크립트는 제거 분석 단계에서 swremove에 의해 실행되어 소프트웨어가 영구적으로 제거되기 전에 확인 작업을 수행합니다. 예를 들어 이 스크립트를 통해 소프트웨어를 제거하기 전에 현재 이를 사용 중인 사용자가 있는지 확인할 수 있습니다.

- **Preremove 스크립트**

이 스크립트는 파일 제거 직전에 실행합니다. 그런 다음 파일이 제거되므로 응용 프로그램에 영향을 미칠 수 있습니다. 이 스크립트는 **postinstall** 스크립트가 작성한 파일을 제거할 수 있습니다. 예를 들어 **preremove** 스크립트로 제품의 나머지 파일 세트를 제거하기 전에 특정 파일 세트를 다른 위치에 저장할 수 있습니다.

이 스크립트와 **postremove** 스크립트는 **swremove**의 제거 단계의 일부입니다. 각 제품 내에서 **preremove** 스크립트가 실행되고(전제 조건에 의해 지정된 순서와 반대로) 파일이 제거된 다음 모든 **postremove** 스크립트가 실행됩니다.

- **Postremove 스크립트**

이 스크립트는 파일을 제거한 직후에 실행됩니다. 이 스크립트는 **postinstall** 스크립트와 짝이 되는 스크립트입니다. 예를 들어 패치 파일 세트가 있고 **preinstall** 스크립트가 원래의 파일을 이동시켰다면 이 **postremove** 스크립트를 통해 패치를 제거한 다음 파일을 원래의 위치로 다시 되돌릴 수 있습니다.



- **Request 스크립트**

이 대화식 스크립트는 소프트웨어 설치나 구성 단계의 일부로서 사용자의 응답이 필요합니다. **request** 스크립트는 **configure** 스크립트나 다른 스크립트가 차후에 사용할 수 있도록 응답 파일에 정보를 기록합니다. 선택 이후 및 분석 단계 전에 **swask** 명령을 사용하거나 **swinstall** 또는 **swconfig** 명령을 **ask** 옵션과 함께 사용하여 **request** 스크립트를 실행할 수 있습니다.

- **기타 스크립트**

그 밖에 위에서 설명한 스크립트에서 파생되는 하위 스크립트 같은 기타 제어 스크립트가 있습니다. 제어 스크립트의 위치는 **SW\_CONTROL\_DIRECTORY** 환경 변수를 통해 모든 스크립트에 전달되며 **PSF** 내에서 **control\_file** 키워드로 나타냅니다.

## 공간 파일

공간 제어 파일은 스크립트가 아닙니다. 공간 제어 파일을 통해 파일 세트에 필요한 추가 디스크 공간 요구 사항을 정의하고, 제어 스크립트 작업으로 인해 디렉토리 또는 파일에 미치는 디스크 공간의 영향에 주목합니다.

각 파일 세트나 제품에 공간 파일이 포함될 수 있습니다. 공간 파일에는 각 경로에 대한 경로와 바이트 크기가 나열되어 있습니다. 예를 들면 다음과 같습니다.

```
/tmp/space_dummy1      2000
/opt/space_dummy2      2000
/tmp/space_dummy3      3000
/mydir/                 4000
```

공간 파일에 나열되어 있는 각 디렉토리 또는 파일 경로에 대해 **swinstall**은 크기를 바이트 단위로 디스크 요구 사항에 추가합니다. 이 크기는 설치에 필요한 임시 또는 영구 최대 디스크 공간을 나타냅니다.

### 스크립트 인터프리터

기본적으로 SD는 스크립트를 POSIX 셸(sh)로 해석합니다. 다음 두 가지 방법으로 다른 스크립트 인터프리터를 지정할 수 있습니다.

먼저 모든 제어 스크립트는 첫째 줄에 인터프리터를 정의할 수 있습니다.

두 번째로 `interpreter` 키워드를 사용하여 특정 스크립트에 대해 다른 인터프리터를 정의할 수 있습니다. 구문은 다음과 같습니다.

```
interpreter interpreter_name
```

예를 들면 다음과 같습니다.

```
control_file
source      scripts
tag         checkinstall
interpreter  ksh
```

SD는 인터프리터가 사용 가능한지 확인합니다. 인터프리터를 사용할 수 없으면 스크립트에 오류가 발생합니다. (이러한 문제를 방지하기 위해 `checkinstall` 스크립트를 사용하여 사용자가 지정한 스크립트 인터프리터가 있는지 확인할 수 있습니다.) SD가 인터프리터를 찾으면 사용자가 지정한 인터프리터를 사용하여 정상적으로 스크립트를 처리합니다.

## 제어 스크립트 형식

제어 스크립트는 셸 스크립트(바이너리가 아님)여야 하며 **Posix.2** 셸 `/sbin/sh`로 해석되도록 작성해야 합니다. **Korn** 셸(이전의 `/bin/ksh`) 구문을 **Posix.2** 셸에서 사용할 수 있습니다. `csh`로 작성된 스크립트는 지원되지 않습니다.

스크립트에는 아래 예에서와 같이 간단한 헤더가 있어야 합니다. 또한 해당 스크립트가 속한 제품과 파일 세트, 스크립트 이름, *what(1)* 명령에서 필요로 하는 버전 문자열과 간단한 저작권 표시를 명시하는 주석 줄도 헤더에 포함되어야 합니다.

```
#!/sbin/sh
#####
# Product: <PRODUCT>
# Fileset: <FILESET>
# configure
# @(#) $Revision: 10.30 $
#####
#
# (c) Copyright MyCompany, 2001
#
#####
```

---

## 일반적인 스크립트 작성 지침

다음은 제어 스크립트를 작성할 때 참고할 수 있는 지침입니다.

- **configure** 스크립트 내에서 대부분의 제어 스크립트 작업을 수행하도록 합니다.
- 모든 스크립트는 순서대로 하나씩 실행되며 설치, 구성 또는 제거 작업을 완료하는 데 소요되는 전체 시간에 직접 영향을 미칩니다. 제어 스크립트가 성능에 미칠 수 있는 영향을 고려합니다.
- 에이전트가 제어 스크립트를 실행하는 현재 작업 디렉토리가 정의되어 있지 않습니다. 모든 경로 이름 참조에는 에이전트가 제공하는 환경 변수를 사용하십시오.
- 디스크 공간 분석은 제어 스크립트에 의해 작성, 복사 또는 제거된 파일은 고려하지 않습니다.
- 사용자가 작성하는 제어 스크립트는 여러 번 실행될 수도 있으므로(예: **configure**, **unconfigure**, **configure**의 순서로) 복수 실행을 지원할 수 있어야 합니다.
- 제어 스크립트에 오류나 경고가 발생하는 경우 제어 스크립트를 다시 실행하거나 디버그해야 하므로 신중하게 작성하고 주석을 표기해야 합니다.
- 제어 스크립트 **stdout** 및 **stderr**은 모두 기록되므로 사용자에게 필요한 정보만 출력하도록 제한해야 합니다.
- 사용자 시스템에 적절한 셸에 대한 경로를 반드시 지정해야 합니다. 스크립트를 실행할 때 다음과 같은 메시지가 표시됩니다.

```
Cannot execute /var/adm/sw/products/PRODUCT/FILESET/  
configure. Bad file number (9).
```

이는 사용자 스크립트의 셸 경로가 사용자 시스템에 맞지 않음을 의미합니다. (HP-UX 9.X 스크립트는 `#!/bin/sh`이며 HP-UX 10.X 및 11.X 스크립트는 `#!/sbin/sh`입니다.)

## 제어 스크립트 패키징하기

다음 표에서는 PSF에서 사용되는 제어 스크립트 키워드를 설명합니다.

표 11-2 제어 스크립트 키워드

키워드	유형	크기 (바이트)	예제
checkinstall	path_string	1024	/mfg/sd/scripts/checkinstall
preinstall	path_string	1024	/mfg/sd/scripts/preinstall
postinstall	path_string	1024	/mfg/sd/scripts/postinstall
unpreinstall	path_string	1024	/mfg/sd/scripts/unpreinstall
unpostinstall	path_string	1024	/mfg/sd/scripts/unpostinstall
configure	path_string	1024	/mfg/sd/scripts/configure
unconfigure	path_string	1024	/mfg/sd/scripts/unconfigure
verify	path_string	1024	/mfg/sd/scripts/verify
checkremove	path_string	1024	/mfg/sd/scripts/checkremove
preremove	path_string	1024	/mfg/sd/scripts/preremove
postremove	path_string	1024	/mfg/sd/scripts/postremove
request	path_string	1024	/mfg/sd/scripts/request
control_file	path_string	1024	/mfg/sd/scripts/subscripts
fix	path_string	1024	/mfg/sd/scripts/fix
space	path_string	1024	/mfg/sd/scripts/space

각 키워드의 값은 특정 제어 스크립트에 대한 소스 파일 이름입니다. swpackage는 키워드를 저장된 스크립트(예: “configure”)의 태그로 사용하여 지정된 제어 스크립트의 파일 이름을 연관된 제품이나 파일 세트에 대한 저장소의 저장 디렉토리로 복사합니다.

제품이나 파일 세트와 함께 제어 스크립트 사양이나 데이터 파일을 포함할 수 있습니다. 이는 표준 **SD-UX** 제어 스크립트와 함께 저장됩니다. 예를 들어 지원되는 제어 스크립트가 호출하는 하위 스크립트나 이러한 스크립트가 읽는 데이터 파일을 지정할 수 있습니다. 이러한 추가 스크립트는 다음과 같은 구문을 사용하여 지정됩니다.

```
PATH[=tag]
```

tag 구성 요소를 지정하지 않으면 `swpackage`가 소스 경로 이름의 *basename(1)* 값을 태그로 사용합니다.

## 실행 시 파일 시스템에서의 제어 스크립트 위치

파일 세트에 대한 `checkinstall`, `preinstall`, `postinstall` 및 보조 스크립트는 이들이 호출되는 임시 디렉토리로 다운로드됩니다.

```
<FILESET>/control_script/var/tmp/<CATALOG_DIR>/ \
  catalog/<PRODUCT>/
```

<CATALOG\_DIR>의 형식은 `aaaa<pid>`로, 여기서 <pid>는 `swinstall` 프로세스의 ID 번호입니다.

스크립트는 분석 단계가 시작될 때 그리고 제품 선택이 완료된 직후 시스템 확인이 시작되기 전 저장소에서 해당 위치로 전달됩니다. `swinstall`이 종료되면 임시 디렉토리는 자동으로 제거됩니다.

파일 세트가 정상적으로 설치되면 다른 모든 제어 스크립트도 **IPD**에 위치하게 됩니다. 이들은 파일 세트의 다른 파일 설치에 포함되어 저장소에서 해당 위치로 전달됩니다.

```
/var/adm/sw/products/<PRODUCT>/<FILESET>/control_script
```

**IPD**의 위치는 소프트웨어 설치가 완료된 루트 디렉토리를 기준으로 합니다. 예를 들어 `/mnt/disk2` 대체 루트에 설치하는 경우 해당 소프트웨어에 대한 **IPD**는 다음 위치가 됩니다.

```
/mnt/disk2/var/adm/sw/products/<PRODUCT>/<FILESET>
```

---

## 참고

**SD-UX** 프로세스에 의해 필요한 모든 디렉토리가 `/var/adm/sw` 아래에 작성됩니다. 이들 디렉토리 아래에 있는 파일은 모두 **SD-UX**에서 시작한 프로세스로 채워집니다. `/var`은 개인 디렉토리이기 때문에 이 디렉토리로 파일을 직접 전달해서는 안됩니다.

---

---

## 환경 변수 사용

모든 제어 스크립트는 슈퍼유저로 호출되고 에이전트 프로세스가 실행합니다. HP-UX에서는 SD-UX 명령과 스크립트를 제어하는 환경 변수를 제공합니다. 이 변수는 다음과 같이 분류됩니다.

- 모든 SD-UX 명령에 영향을 주는 변수
- 모든 SD-UX 스크립트에 영향을 주는 변수
- `swinstall`과 `swremove`에 영향을 주는 변수

### 모든 SD-UX 명령에 영향을 주는 변수

#### LANG

- 이 외부 변수는 `install-sd`를 제외한 모든 SD 명령에 적용됩니다.
- 메시지가 표시되는 언어를 결정합니다. LANG이 지정되지 않았거나 빈 문자열로 지정되면 “C”의 기본값이 사용됩니다.
- SD 에이전트와 데몬 로그 메시지가 표시되는 언어는 시스템 구성 변수 스크립트인 `/etc/rc.config.d/LANG`에 의해 설정됩니다. 예를 들어 에이전트와 데몬 로그 메시지를 일본어로 표시하려면 `/etc/rc.config.d/LANG`을 “`LANG=ja_JP.SJIS`”로 설정하거나 “`LANG=ja_JP.eucJP`”로 설정해야 합니다.  
`export LANG=` 명령을 사용할 수도 있습니다.
- 자세한 내용은 *lang(5)* 맨 페이지를 참조하십시오.

#### LC\_ALL

- LANG의 설정으로 지정된 로케일 범주나 LC\_로 시작하는 환경 변수에 대한 값을 재정의하는 데 사용될 로케일을 결정합니다.

#### LC\_CTYPE

- 텍스트 데이터의 바이트 시퀀스를 문자로 해석하는 것을 결정합니다(예를 들어 공급업체가 정의한 속성 값에서 싱글 바이트 문자인지 멀티 바이트 문자인지 여부).

### LC\_MESSAGES

- 메시지가 기록되는 언어를 결정합니다.

### LC\_TIME

- `swlist` 명령으로 표시되는 날짜 형식(`create_date` 및 `mod_date`)을 결정합니다. `stdout`, `stderr` 및 로깅에서 날짜와 시간을 표시할 때 모든 유틸리티가 사용합니다.

### TZ

- 날짜와 시간을 표시할 때 사용될 시간대를 결정합니다.

## 모든 SD-UX 스크립트에 영향을 주는 변수

### SW\_CATALOG

- `SW_ROOT_DIRECTORY` 환경 변수의 경로를 기준으로 하여 IPD(Installed Products Database)에 대한 경로를 유지합니다.  
(`installed_software_catalog` 기본 옵션을 사용하여 IPD에 대한 경로를 지정할 수 있습니다.)

### SW\_CONTROL\_DIRECTORY

- 스크립트를 포함하는 디렉토리의 전체 경로 이름을 정의합니다. 이는 소프트웨어에 대한 다른 제어 스크립트(예: 하위 스크립트)의 위치를 다른 스크립트에 알려주는 역할을 합니다.

또한 `request` 스크립트에 의해 생성된 응답 파일이 포함됩니다. 응답 파일을 참조하는 다른 스크립트는 이 변수를 참조하여 응답 파일에 액세스합니다.

디렉토리는 임시 카탈로그 디렉토리나 IPD(Installed Products Database) 내의 디렉토리 중 하나입니다.

다음은 소싱의 예입니다.

```
. ${SW_CONTROL_DIRECTORY}subscript  
grep something ${SW_CONTROL_DIRECTORY}datafile
```



## SW\_CONTROL\_TAG

- 실행할 `control_file`의 태그 이름을 유지합니다. 소프트웨어를 패키징할 때 저장소의 제어 파일에 대한 실제 이름과 경로를 정의할 수 있습니다. `control_file`을 태그 이름이 아닌 다른 이름으로 정의할 수 있고 같은 파일을 가리키는 여러 `control_file` 정의를 사용할 수 있습니다. `control_file`은 `SW_CONTROL_TAG` 변수를 조회하여 실행할 태그를 결정합니다.

## SW\_LOCATION

- 기본 **제품 디렉토리**(제품이 검색 가능한 경우)에서 변경되었을 수도 있는 제품의 위치를 정의합니다.  
기본 루트 디렉토리("/")에 설치할 때 이 변수는 제품 디렉토리에 대한 절대 경로가 됩니다. 대체 루트 디렉토리에 대한 작업의 경우 제품 파일을 올바르게 참조하도록 `SW_ROOT_DIRECTORY` 접두어를 변수 앞에 붙입니다.  
제품을 검색할 수 없는 경우 `SW_LOCATION` 값은 제품이 패키징될 때 정의된 기본 제품 디렉토리가 됩니다.

## SW\_PATH

- 명령에 대한 검색 경로. `PATH` 변수는 제어 스크립트에서 사용할 수 있는 명령의 최소 세트를 정의합니다(예: `/sbin:/usr/bin:/usr/ccs/sbin`).  
제어 스크립트는 항상 자체 `PATH` 변수를 정의해야 하고 `PATH` 변수는 `$SW.PATH`로 시작해야 합니다. `PATH`는 다음과 같이 설정됩니다.

```
PATH=$SW_PATH
```

```
export PATH
```

`/usr/local/bin`과 같은 추가 디렉토리를 `PATH`에 추가할 수도 있지만 그러한 디렉토리에 명령이 존재하는지 확인해야 합니다.

## SW\_ROOT\_DIRECTORY

- 세션이 작동되는 루트 디렉토리인 “/” 또는 대체 루트 디렉토리를 정의합니다. 이 변수는 제품이 설치되는 루트 디렉토리를 제어 스크립트에 알려줍니다. 스크립트가 제품의 설치된 파일 위치를 찾기 위해서는 이 디렉토리를 SW\_LOCATION에 접두어로 사용해야 합니다.

모든 제어 스크립트(`configure` 및 `unconfigure` 스크립트는 제외)는 대체 루트에서 설치 또는 제거 작업이 수행되는 동안 실행될 수 있습니다. 스크립트가 제품 파일을 참조하는 경우에는 각 참조의 파일 경로 이름에 {SW\_ROOT\_DIRECTORY}를 포함해야 합니다.

스크립트가 기본 루트 디렉토리(“/”)에 설치하거나 기본 루트 디렉토리에서 제거할 때 조치만 수행해야 할 수도 있습니다. 그런 경우 작업이 대체 루트 디렉토리에서 수행될 때 SW\_ROOT\_DIRECTORY는 단순 `exit 0`이 발생하도록 하는 데 사용될 수 있습니다.

```
if test "${SW_ROOT_DIRECTORY}" != "/"
then
    exit 0
fi
```

## SW\_SESSION\_OPTIONS

- 소프트웨어 및 대상 선택을 포함하여 특정 명령에 대한 모든 옵션 값이 들어 있는 파일의 경로 이름이 포함됩니다. 이를 통해 스크립트는 다른 환경 변수가 명시적으로 제공하는 것 이외의 명령 옵션 및 값을 검색할 수 있습니다.

## SW\_SOFTWARE\_SPEC

- 현재의 제품이나 파일 세트에 대한 정규화된 소프트웨어 사양이 포함됩니다. 소프트웨어 사양을 통해 제품이나 파일 세트를 고유하게 식별할 수 있습니다. (정규화된 소프트웨어 사양에는 사양에 공백 문자열이 포함되어 있는 경우에도 `r=`, `a=` 및 `v=` 버전 구성 요소가 포함됩니다. 설치된 소프트웨어에는 `l=`도 포함되어야 합니다.)

## swinstall과 swremove에 영향을 주는 변수

### SW\_DEFERRED\_KERNBLD

- 일반적으로 이 변수는 설정하지 않습니다. 이 변수를 설정하면 /stand/system 시스템 파일을 준비하는 데 필요한 조치를 postinstall 스크립트 내에서 수행할 수 없으며 대신 configure 스크립트를 통해 수행해야 합니다. 이것은 root(/)가 아닌 다른 디렉토리에 소프트웨어를 설치할 때마다 발생합니다.
- 이 변수는 커널 파일 세트의 configure 및 postinstall 스크립트에 의해서만 읽혀 집니다.

### SW\_INITIAL\_INSTALL

- 일반적으로 이 변수는 설정하지 않습니다. 이 변수를 설정하면 swinstall 세션이 초기 시스템 소프트웨어 설치의 백엔드로 실행됩니다(즉, “cold” 설치).

### SW\_KERNEL\_PATH

- 커널에 대한 경로. 기본값은 /stand/vmunix입니다.

### SW\_SESSION\_IS\_KERNEL

- 커널 빌드가 현재 설치/제거 세션에 대해 예약되어 있는지 여부를 나타냅니다.
- “true” 값은 선택된 커널 파일 세트가 커널 빌드에 대해 예약되어 있고 /stand/system을 변경할 필요가 있음을 나타냅니다.
- 널 값은 커널 빌드가 예약되어 있지 않고 /stand/system을 변경할 필요가 없음을 나타냅니다.
- 이 변수 값은 항상 SW\_SESSION\_IS\_REBOOT 값과 동일합니다.

### SW\_SESSION\_IS\_REBOOT

- 제거를 위해 선택된 파일 세트에 대해 재부팅이 예약되어 있는지 여부를 나타냅니다. HP-UX 커널 파일 세트는 모두 재부팅 파일 세트이기도 하므로 이 변수의 값은 항상 SW\_SESSION\_IS\_KERNEL 값과 동일합니다.

### **SW\_SYSTEM\_FILE\_PATH**

- 커널의 시스템 파일에 대한 경로. 기본값은 `/stand/system`입니다.

### **swverify에 영향을 주는 변수**

#### **SW\_IS\_COMPATIBLE**

- 설치된 소프트웨어가 호환성이 없는지 및 시스템에서 제거해야 하는지 여부를 결정하는 데 도움이 되도록 설계되었습니다.
- `swverify` 명령에 의해 호출되는 `verify` 스크립트 실행 중에 사용됩니다.
- 해당 소프트웨어가 설치되어 있는 시스템과 호환성이 있다고 판단되는 경우 이 변수는 `true`로 설정됩니다.
- 해당 소프트웨어가 설치되어 있는 시스템과 호환성이 없다고 판단되는 경우 `false`로 설정됩니다.

## 제어 스크립트의 실행

이 절에서는 각 제어 스크립트가 실행되는 방법에 대해 자세히 설명합니다.

### 모든 제어 스크립트에 적용되는 세부 사항

- 에이전트는 수퍼유저로 실행되므로 제어 스크립트는 항상 수퍼유저로 실행됩니다. 적절한 주의 사항을 지키시기 바랍니다.
- 제어 스크립트는 기본 루트("/") 또는 대체 루트 디렉토리에서 설치, 제거 또는 확인되는 소프트웨어에 대해서만 실행됩니다. 저장소에 있는 소프트웨어에 대해서는 실행되지 않습니다.
- 각 스크립트는 `SW_PATH`를 사용하여 고유한 `PATH` 변수를 설정해야 합니다.
- `swinstall`과 `swremove` 모두 시스템을 종료할 필요가 없습니다. 제어 스크립트는 단일 사용자 시스템이나 여러 사용자 시스템에서 모두 제대로 작동되어야 합니다. 또한 제거할 수 없는 실행 프로그램을 적절하게 처리할 수 있어야 합니다. 스크립트 자체가 정상적으로 실행되도록 하기 위해 프로세스를 종료하거나 시작해야 할 수도 있습니다.
- 제어 스크립트는 반복해서 실행할 수 있습니다. 스크립트가 두 번 이상 실행되는 경우 매번 결과는 동일해야 합니다. 두 번째 실행될 때 오류 메시지가 발생하거나 실행되기 전과는 다른 시스템 상태가 되어서는 안됩니다.

해당 파일 세트가 연관된 새로운 파일 세트를 손상시키지 않고 로드된 후 스크립트가 실행될 수 있어야 합니다.

예를 들어 `/usr/newconfig` 아래에 있는 파일을 다른 위치로 복사해야 하는 경우 `cp` 명령보다는 `cpio -p` 명령을 사용하여 이동하는 것이 더 좋습니다. 또는 파일을 이동하기 전에 `/usr/newconfig` 버전이 있는지 확인합니다. (`cpio(1)` 명령은 모드, 소유자 및 그룹 권한을 복사하기 때문에 `cp(1)` 명령보다 선호됩니다.)

- 제어 스크립트는 다음 반환 값과 함께 종료할 수 있습니다.
  - `SUCCESS (0)` - 오류나 경고 없이 종료합니다.
  - `ERROR (1)` - 심각한 오류와 함께 종료합니다.
  - `WARNINGS (2)` - 경고와 함께 종료합니다.

- REBOOT (12) - 파일 세트가 설치될 때 시스템 재부팅이 필요하다고 표시하고 종료합니다. 이 반환 값은 `dynamic_module` 속성을 포함하는 파일 세트에 `checkinstall`과 `checkremove` 스크립트를 실행하는 경우에만 유효합니다.

제어 스크립트에서 생성되는 메시지는 에이전트 로그 파일에 모두 기록됩니다.

- 작업의 특정 단계에서 실행되는 제어 스크립트 세트는 종속 파일 세트의 스크립트가 실행되기 전에 전제 조건이 되는 각 제품/파일 세트의 스크립트가 실행되는 전제 조건 순서에 따라 실행됩니다.
- 모든 제어 스크립트는 다른 제어 스크립트가 읽을 수 있습니다.

## Checkinstall 스크립트

- `checkinstall` 스크립트는 `swinstall` 세션의 분석 단계에서 실행됩니다. 실행되는 스크립트의 경로 이름은 다음과 같습니다.

```
$(SW_CONTROL_DIRECTORY)checkinstall
```

- `checkinstall` 스크립트는 시스템을 변경하지 않습니다.
- `checkinstall` 스크립트는 `swinstall`이 수행하는 확인 작업 외의 사항을 확인하여 제품/파일 세트가 설치될 수 있는지 결정합니다. 예를 들어 제품/파일 세트가 사용 중인지 또는 시스템 실행 수준이 적합한지 등을 확인합니다.
- 설치의 일부로 `request` 스크립트를 사용하는 경우 `checkinstall` 스크립트는 다음을 수행해야 합니다.

- 응답 파일이 존재하는지 확인합니다.
- 다음과 같은 경우 `swinstall`이 “중지”되지 않도록 합니다.
  - 스크립트가 존재하지 않는 응답 파일을 읽으려 하는 경우 또는
  - 설치 또는 구성이 유실된 응답 파일의 정보에 의존하고 있는 경우

- `checkinstall` 스크립트에 오류가 발생하면 해당 파일 세트는 설치되지 않습니다. `swinstall`의 대화식 인터페이스를 통해 `checkinstall` 스크립트가 실패했다는 메시지를 사용자에게 표시합니다. 문제를 진단하고 수정한 다음 분석 단계를 다시 실행하거나 제품/파일 세트의 선택을 취소할 수 있습니다. 비대화식 인터페이스의 경우 `checkinstall`에 오류가 발생했고 파일 세트가 설치되지 않았음을 개별적으로 각각 알려줍니다.

- **checkinstall** 스크립트는 기본 루트("/")나 대체 루트에 설치하는 경우 실행됩니다. 이 스크립트에서 하는 대부분의 조치는 실행 중인 시스템(즉, 기본 루트)의 현재 상태를 확인하는 일이므로 제품/파일 세트가 대체 루트에 설치될 때에는 조치를 수행할 필요가 없습니다.

## Preinstall 스크립트

- **preinstall** 스크립트는 **swinstall** 세션의 로드 단계에서 실행됩니다. 실행되는 스크립트의 경로 이름은 다음과 같습니다.

```
$(SW_CONTROL_DIRECTORY)preinstall
```

- 제품의 **preinstall** 스크립트는 파일 세트의 파일이 설치되기 직전에 실행됩니다.
- **preinstall** 스크립트는 설치되는 파일에 대한 특정 준비 작업을 합니다. **swinstall** 세션은 **preinstall** 스크립트의 반환 값과는 상관없이 파일을 계속 설치합니다. 예를 들면 오래된 파일을 제거(업데이트 시나리오에서)합니다.
- **preinstall** 스크립트는 기본 루트("/")나 대체 루트에 설치하는 경우 실행됩니다. **preinstall** 스크립트 조치의 범위는 제품 자체(즉, 제품 디렉토리의 파일)로 국한됩니다.

## Postinstall 스크립트

- **postinstall** 스크립트는 **swinstall** 세션의 로드 단계에서 실행됩니다. 실행되는 스크립트의 경로 이름은 다음과 같습니다.

```
$(SW_CONTROL_DIRECTORY)postinstall
```

- 제품의 **postinstall** 스크립트는 파일 세트의 파일이 설치된 직후에 실행됩니다.
- **postinstall** 스크립트는 방금 설치된 파일에 관련된 특정 작업을 수행합니다. **swinstall** 세션은 **postinstall** 스크립트의 반환 값과는 상관없이 세션의 나머지 부분(예: 구성)을 수행합니다. 예를 들면 커널 드라이버를 시스템 파일에 추가하거나 **/usr/newconfig** 아래의 파일을 파일 시스템 내 올바른 위치로 이동시키는 것 등이 있습니다.

- `postinstall` 스크립트는 기본 루트("/")나 대체 루트에 설치하는 경우 실행됩니다. `postinstall` 스크립트 조치의 범위는 제품 자체(즉, 제품 디렉토리의 파일)로 국한됩니다.
- 제품/파일 세트를 범용으로 사용할 수 있도록 하기 위해 수행해야 하는 사용자 정의 작업이나 구성 작업은 `postinstall` 스크립트가 아닌 `configure` 스크립트(아래에 설명)에서 수행됩니다.

## Configure 스크립트

- `configure` 스크립트는 `swinstall` 세션의 구성 단계에서 실행됩니다. `swinstall` 세션이 시스템 재부팅을 트리거하는 경우 `SD`는 시스템 시작 시에 구성 스크립트를 예상합니다. `swconfig` 명령으로도 `configure` 스크립트를 실행할 수 있습니다. 실행되는 스크립트의 경로 이름은 다음과 같습니다.  

```
$(SW_CONTROL_DIRECTORY)configure
```
- `configure` 스크립트는 기본 루트("/")에 설치되는 경우에만 실행됩니다. `swinstall` 세션에서 구성 작업을 연기하면 `configure` 스크립트는 설치가 완료된 후 특정 시점에 `swconfig` 세션에 의해 실행됩니다.
- `configure` 스크립트는 일반적으로 제품/파일 세트의 상태가 설치 상태인 경우에만 실행됩니다.
- `configure` 스크립트는 제품/파일 세트의 상태를 설치 상태에서 구성 상태로 바꿀 수 있는 1차적인 방법입니다. 이 스크립트는 제품/파일 세트를 사용할 수 있도록 하는 데 필요한 모든(또는 대부분의) 조치를 수행합니다.
- `configure` 스크립트는 사용자가 제공하고 `request` 스크립트에 의해 수집된 구성 정보를 사용할 수 있습니다.
- 제품의 기존 버전을 새 버전으로 업데이트하는 경우 새 버전에 대한 `configure` 스크립트는 새 버전을 적절하게 구성하기 위해 필요한 이전 버전의 구성 해제-구성 작업을 수행해야 합니다. 이전 버전에 대한 `unconfigure` 스크립트는 실행되지 않습니다.
- `configure` 스크립트는 항상 설치 대상의 아키텍처 위에서 실행되므로 아키텍처에 따라 달라질 수 있는 조치입니다.
- `configure` 스크립트를 실행할 때는 시스템이 전환 상태(예: 업데이트)가 아니므로 `configure` 스크립트는 파일을 제거하고 `IPD`를 업데이트하기 위한 가장 적절한 위치입니다.



- **configure** 스크립트는 새로운 설치는 물론 소프트웨어 업데이트에도 도움이 됩니다. **configure** 스크립트는 재설치를 처리할 수 있어야 하며 데이터가 손상될 가능성이 있는 경우 적절한 오류 제어 기능도 들어 있어야 합니다.

## Unconfigure 스크립트

- **unconfigure** 스크립트는 **swremove** 세션의 구성 해제 - 구성 단계에서 실행됩니다. **unconfigure** 스크립트는 **swconfig** 명령에 의해서도 실행될 수 있습니다. 실행되는 스크립트의 경로 이름은 다음과 같습니다.

```
`${SW_CONTROL_DIRECTORY}unconfigure
```

- **unconfigure** 스크립트는 기본 루트("/")에 설치되는 소프트웨어에 대해서만 실행됩니다.
- **unconfigure** 스크립트는 제품/파일 세트가 구성 상태여도 다시 실행됩니다.
- **unconfigure** 스크립트는 제품/파일 세트의 상태를 구성 상태에서 설치 상태로 다시 되돌릴 수 있는 1차적인 방법입니다. 이 스크립트는 제품/파일 세트를 사용할 수 없도록 하는 데 필요한 모든(또는 대부분의) 조치를 수행합니다.
- **unconfigure** 스크립트는 해당 구성 스크립트에 의해 수행된 모든 구성 작업을 취소해야 합니다. 사용자는 설치된 제품/파일 세트에 대해 구성, 구성 해제, 구성 등을 반복할 수 있으며 항상 구성 결과가 동일해야 합니다.

## Verify 스크립트

- **verify** 스크립트는 **swverify** 명령에 의해 실행됩니다. 실행되는 스크립트의 경로 이름은 다음과 같습니다.

```
`${SW_CONTROL_DIRECTORY}verify
```

- **verify** 스크립트는 시스템을 변경해서는 안됩니다.
- **verify** 스크립트는 **configure** 스크립트에 의해 수행된 구성 작업이 올바르게 완전한지를 확인하기 위한 1차적인 방법입니다.

- **verify** 스크립트는 기본 루트("/")나 대체 루트에 설치하는 경우 실행됩니다. 이 스크립트에서 하는 대부분의 조치는 구성된(기본 루트에) 제품/파일 세트 상태를 확인하는 일이므로 대체 루트 디렉토리에 설치된 제품/파일 세트에는 조치를 수행할 필요가 없습니다.
- 환경 변수인 **SW\_IS\_COMPATIBLE**은 **verify** 스크립트가 자신이 설치된 시스템과 설치된 소프트웨어가 호환되는지 확인하는 데 도움을 줍니다. 자세한 내용은 364페이지의 "SW\_IS\_COMPATIBLE"을 참조하십시오.

## Fix 스크립트

- **fix** 스크립트는 **swverify** 명령에 의해 실행됩니다. 실행되는 스크립트의 경로 이름은 다음과 같습니다.  

```
${SW_CONTROL_DIRECTORY}fix
```
- **fix** 스크립트는 **verify** 스크립트를 사용하여 검색된 속성 문제를 해결하는 데 사용할 수 있습니다.
- **fix** 스크립트는 유실된 디렉토리를 작성하고, 파일 변경 사항(모드, 소유자, 그룹, 1차 및 2차)을 수정하며 심볼릭 링크를 다시 작성할 수 있습니다.

## Checkremove 스크립트

- **checkremove** 스크립트는 **swremove** 세션의 분석 단계에서 실행됩니다. 실행되는 스크립트의 경로 이름은 다음과 같습니다.  

```
${SW_CONTROL_DIRECTORY}checkremove
```
- **checkremove** 스크립트는 시스템을 변경해서는 안됩니다.
- **checkremove** 스크립트는 **swremove**가 확인 작업 외의 사항을 확인하여 제품/파일 세트가 제거될 수 있는지 결정합니다. 예를 들면 제품/파일 세트가 사용 중인지 여부를 확인하는 것 등이 있습니다.
- **checkremove** 스크립트에 오류가 발생하면 해당 제품의 파일 세트는 제거되지 않습니다. **swremove**의 GUI/TUI 인터페이스는 **checkremove** 스크립트에 오류가 발생했음을 사용자에게 알려 줍니다. 문제를 진단하고 수정한 다음 분석 단계를 다시 실행하거나 문제가 되는 대상 시스템 선택을 취소하거나 제품/파일 세트의 선택을 취소할 수 있습니다. 명령줄 인터페이스에서는 **checkremove**에 오류가 발생했으며 해당 제품의 파일 세트가 제거되지 않음을 각각 개별적으로 알려줍니다.

- **checkremove** 스크립트는 기본 루트("/")나 대체 루트에 설치하는 경우 실행됩니다. 이 스크립트에서 하는 대부분의 조치는 실행 중인 시스템(즉, 기본 루트)의 현재 상태를 확인하는 일이므로 제품/파일 세트가 대체 루트에서 제거될 때에는 조치를 수행할 필요가 없습니다.

## Preremove 스크립트

- **preremove** 스크립트는 **swremove** 세션의 제거 단계에서 실행됩니다. 실행되는 스크립트의 경로 이름은 다음과 같습니다.  

```
$(SW_CONTROL_DIRECTORY)preremove
```
- 제품에 대한 모든 **preremove** 스크립트는 제품의 파일이 제거되기 직전에 실행됩니다.
- **preremove** 스크립트는 제거되는 파일에 대한 특정 준비 작업을 수행합니다. **swremove** 세션은 **preremove** 스크립트의 반환 값과는 상관없이 파일을 제거합니다. 예를 들면 **postinstall** 스크립트에서 작성한 파일을 제거합니다.
- **preremove** 스크립트는 기본 루트("/")나 대체 루트에 설치하는 경우 실행됩니다. **preremove** 스크립트 조치의 범위는 제품 자체(즉, 제품 디렉토리의 파일)로 국한됩니다.
- 제품/파일 세트를 범용으로 사용할 수 없도록 하기 위해 수행해야 하는 사용자 정의 해제 또는 구성 해제-구성 작업은 **preremove** 스크립트에서 수행되지 않고 **unconfigure** 스크립트(위에서 설명)에서 수행되어야 합니다.

## Postremove 스크립트

- **postremove** 스크립트는 **swremove** 세션의 제거 단계에서 실행됩니다. 실행되는 스크립트의 경로 이름은 다음과 같습니다.  

```
$(SW_CONTROL_DIRECTORY)postremove
```
- 제품의 모든 **postremove** 스크립트는 제품의 파일 세트 파일이 제거된 직후에 실행됩니다.
- **postremove** 스크립트는 방금 제거된 파일에 관련된 특정 작업을 수행합니다. **swremove** 세션은 **postremove** 스크립트의 반환 값과는 상관없이 세션의 나머지 부분을 수행합니다. 예를 들면 다음과 같은 조치를 수행합니다.

- `preremove` 이후에도 남아 있는 파일을 제거하고 `swremove` 파일 제거를 완료합니다.
- 전적으로 파일 세트에 포함되고 파일 제거에 의해 비워진 디렉토리를 제거합니다.
- `postremove` 스크립트는 기본 루트("/")와 대체 루트에 설치하는 경우 실행됩니다. `postremove` 스크립트 조치의 범위는 제품 자체(즉, 제품 디렉토리의 파일)로 국한됩니다.
- 제품/파일 세트를 범용으로 사용할 수 없도록 하기 위해 수행해야 하는 사용자 정의 해제 또는 구성 해제-구성 작업은 `postremove` 스크립트에서 수행되지 않고 `unconfigure` 스크립트(위에서 설명)에서 수행되어야 합니다.

## Request 스크립트

- `request` 스크립트는 소프트웨어 설치나 구성 단계의 일부로서 사용자의 응답을 요구하는 대화식 스크립트입니다. 실행되는 스크립트의 경로 이름은 다음과 같습니다.  

```
${SW_CONTROL_DIRECTORY}request
```
- `request` 스크립트는 `configure` 스크립트나 다른 스크립트가 이후에 사용할 수 있도록 응답 파일에 정보를 기록합니다. 선택 이후와 분석 단계 전에 `swask` 명령을 실행하거나 `swinstall` 또는 `swconfig` 명령을 `ask` 옵션과 함께 사용하여 `request` 스크립트를 실행할 수 있습니다.
- `request` 스크립트에 대한 POSIX 기본값은 셸 스크립트입니다. 셸 스크립트는 다음과 같은 작업을 수행할 수 있어야 합니다.
  - 사용자에게 질문합니다.
  - 사용자의 대답을 읽습니다.
  - 화면을 다시 표시하여 현재 사용자 응답을 모두 표시합니다.
  - 사용자에게 응답을 확인하고 계속 진행하거나 다시 돌아갈 것인지 묻습니다.
- `request` 스크립트는 응답 파일에 사용자 응답을 저장합니다. 응답 파일의 경로는 `SW_CONTROL_DIRECTORY` 환경 변수로 액세스할 수 있습니다.

- 응답 파일 형식에 대한 POSIX 권장 사항은 속성/값 쌍의 SVR4 모델입니다. 대답을 응답 파일에 `env_var=value` 형식으로 작성해야 다른 제어 스크립트가 응답 파일을 쉽게 사용할 수 있습니다.
- `request` 스크립트를 사용하여 설치 정보를 구하려는 경우 `checkinstall` 스크립트를 사용하여 `request` 스크립트가 올바르게 실행되었는지 확인하는 것이 좋습니다. `checkinstall` 스크립트는 다음과 같은 작업을 수행합니다.
  - 응답 파일이 존재하는지 확인합니다.
  - 다음과 같은 경우 `swinstall`이 “중지”되지 않도록 합니다.
    - 스크립트가 존재하지 않는 응답 파일을 읽으려고 하는 경우 또는
    - 설치 및 구성이 유실된 응답 파일의 정보에 의존하고 있는 경우

## 제어 스크립트에 의한 다른 명령 실행

제어 스크립트에 의해 실행되는 모든 명령은 대상 시스템에 해당 명령이 없는 경우 오류가 발생할 가능성이 있습니다. 스크립트에서 먼저 명령의 존재 여부와 실행 가능성을 확인하고 해당 명령이 없어도 오류가 발생하지 않는다면 조건부로 명령을 사용할 수 있습니다.

- 대상 시스템이 **POSIX 1003.2** 셸 및 유틸리티 표준을 따르는 경우 이 표준의 **Execution Environment Utilities**도 사용할 수 있습니다.
- 파일 세트가 다른 제품/파일 세트를 전제로 하는 종속성을 가지고 있다면 종속적 파일 세트에 대한 대부분의 제어 스크립트는 **\$ROOT\_DIRECTORY**가 /인 경우 필요한 제품/파일 세트의 명령을 사용할 수 있습니다. (모든 명령은 전제 조건 순서에 따라 작업을 수행합니다).
- 명령은 **PATH** 변수에 지정된 경로 구성 요소에 대한 상대 경로로 참조됩니다. (위의 **PATH**와 **SW\_PATH** 환경 변수에 대한 설명을 참조하십시오.)

## 제어 스크립트의 입력 및 출력

- `request` 스크립트를 제외한 나머지 모든 제어 스크립트는 대화식이 아닙니다. 여기에는 `Press return to continue`와 같은 메시지도 포함됩니다.
- `request` 스크립트를 제외한 모든 제어 스크립트는 대상 시스템에서 에이전트에 의해 실행됩니다. `request` 스크립트는 컨트롤러(`swinstall`, `swconfig` 또는 `swask`)에 의해 실행됩니다.
- `request` 스크립트를 제외하고는 제어 스크립트에 입력 방법이 지원되지 않습니다. `request` 스크립트는 사용자가 `swask` 명령을 사용하거나 `swinstall` 또는 `swconfig` 명령에 `ask` 옵션을 사용하여 입력합니다.
- 제어 스크립트는 오류와 경고 조건에 대한 메시지를 `stderr` (`echo &>2`)에 쓰고 기타 다른 메시지를 `stdout`에 써야 합니다. 제어 스크립트가 `/dev/console`에 직접 쓰거나 화면에 직접 쓰는 다른 방법을 시도해서는 안 됩니다.

제어 스크립트의 `stdout`와 `stderr`은 에이전트에 의해 작업이 수행되고 있는 기본 및 대체 루트 디렉토리 내 로그 파일(`var/adm/sw/swagent.log`)로 다시 지정됩니다.

`swinstall`과 `swremove`의 대화식 세션의 경우 이 로그 파일을 표시하고 찾아볼 수 있습니다.

- 최소한의 필수적인 정보만 제어 스크립트에 의해 출력합니다. 스크립트가 조치를 모두 정상적으로 수행하면 어떠한 출력도 표시되지 않는 것이 이상적입니다.
- 에이전트 로그 파일에서 각 제어 스크립트 실행 앞에 “`begin execution`” 메시지가 표시됩니다.

```
* Running "checkinstall" script for product "PRODUCT"  
* Running "checkinstall" script for fileset  
  "PRODUCT.FILESET".
```

스크립트에 의해 생성되는 모든 메시지는 다음과 같습니다. 스크립트가 `0(SUCCESS)`이 아닌 다른 값을 반환한 경우 다음과 같은 종료 메시지가 기록됩니다.

```
ERROR: The "unconfigure" script for "PRODUCT.FILESET"
failed (exit code "1"). The script location was
"/var/adm/sw/products/PRODUCT/FILESET/unconfigure".
* This script had errors but the execution of this
product will still proceed. Check the above output
from the script for further details.
```

```
WARNING: The "unconfigure" script for
"PRODUCT.FILESET" failed (exit code "2"). The script locat
ion was
"/var/adm/sw/products/PRODUCT/FILESET/unconfigure "
* This script had warnings but the execution of this
product will still proceed. Check the above output
from the script for further details.
```

- 제어 스크립트에 의해 기록되는 메시지는 가능한 다음 형식을 따라야 합니다.
  1. 공백 줄을 두지 마십시오.
  2. 모든 출력 줄은 다음 형식 중 하나를 따라야 합니다.

```
ERROR:      text
WARNING:    text
NOTE:       text
blank      text
```

각각의 경우에 키워드는 1열에서 시작해야 하고 *text*는 10열(공백 9칸을 들여쓰기)에서 시작해야 합니다.

3. 다음과 같이 키워드(ERROR, WARNING, NOTE 또는 공백)를 선택합니다.

ERROR:	작업을 계속 진행할 수 없으며 문제를 해결하기 위한 조치가 필요합니다.
WARNING:	작업을 계속 진행할 수는 있지만 문제가 있으며 조치가 필요합니다.
NOTE:	작업을 계속 진행할 수는 있지만 정상적이지 않거나 특별한 주의가 필요한 사항이 발생했습니다.(단순한 상태 메시지가 아닙니다.)
공백	일반적인 진행 및 상태 메시지입니다(가능한 한 최소한으로 유지하십시오).

별표(\*) 문자로 시작하지 마십시오. 이 문자는 에이전트에 의해 인쇄되는 작업 메시지를 위해 예약된 것으로 다른 메시지와 쉽게 구분하기 위한 것입니다.



4. 72자 길이의 한 줄을 초과하는 경우 72자 길이의 여러 줄로 나눕니다. 첫 줄 이후 나머지 줄은 모두 들여쓰기를 합니다. 예를 들면 다음과 같습니다.

```
NOTE: To install your new graphics package, you
      must turn on the lights in the next room.
      Please turn them off when you leave.
```

5. 메시지 안에 탭 문자를 사용하지 마십시오.

- 스크립트는 예기치 않은 오류가 발생할 수 있고 위에 설명한 형식이 아닌 다른 형식으로 출력을 표시할 수도 있는 다른 명령을 실행합니다. 오류가 발생할 가능성이 있는(그럴 이유가 충분하다면) 곳에서는 실행되는 명령의 표준 출력이나 오류를 /dev/null 또는 임시 파일로 재지정합니다. 그런 다음 반환 코드나 명령의 출력을 기초로 적절한 형식의 메시지를 내보냅니다. 예를 들면 다음과 같습니다.

```
/bin/grep bletch /etc/bagel 2c>/dev/null
if[ $?=1]
then
  echo "ERROR: Cannot find bletch in /etc/bagel." |&>2
fi
```

- 이러한 규칙을 따라서 제어 스크립트의 메시지가 에이전트 및 명령 자체에 의해 생성되는 메시지와 비슷한 형태와 느낌을 가질 수 있도록 합니다.

— 가능하다면 완전한 문장을 사용합니다. 간략한 문장은 피하도록 합니다.

— 문장과 구는 대문자로 시작하고 마침표로 끝냅니다.

— 마침표 뒤에는 공백을 두 개 두고 콜론, 세미콜론 및 쉼표 뒤에는 공백을 한 개 두십시오.

— 콜론 뒤에 나오는 절의 첫 번째 문자는 대문자를 사용합니다. (이렇게 하면 메시지를 이해할 수 있는 정보의 “단위”로 만드는 데 도움이 됩니다.)

— 제품, 파일 세트, 디렉토리 및 파일 이름과 기타 변수 값의 문자열 앞뒤로 인용 부호를 사용합니다. 예를 들면 다음과 같습니다.

```
echo "ERROR: Cannot open file \"${file}\"." &>2
```

— 현재 시제를 사용합니다. “would”, “will” 등 이와 유사한 시제는 사용하지 않습니다. 또한 필요한 경우를 제외하고는 과거 시제도 사용하지 않습니다.

— “can’t”, “could not”, “couldn’t”, “unable to”, “failed to” 등 이와 유사한 구문 대신 “cannot”을 사용합니다.

- 시스템 관리자와 사용자가 이해할 수 있도록 메시지를 작성합니다. 메시지 대상을 고려하십시오.

---

## 제어 스크립트에 의한 파일 관리

- `preinstall`, `postinstall` 또는 `configure` 스크립트에 의해 작성된 모든 파일은 각기 해당되는 `postremove`, `preremove` 또는 `unconfigure` 스크립트에 의해 제거됩니다.

스크립트에 의해 작성된 파일은 `swremove` 명령이 인식할 수 없으며 `swinstall` 명령으로 설치한 파일을 제거할 때 제거되지 않습니다. 스크립트로 작성된 파일을 **SD**로 제거하려는 경우에는 `swmodify` 명령이나 `control_utils` 함수 `IPD_addfile`을 사용하여 **IPD**에 파일을 추가해야 합니다.

- 이전 버전의 제품에 있는 파일의 이름이 바뀌었거나 오래된 파일이 있는 경우 제품의 새로운 버전에 대한 제품/파일 세트의 `preinstall`이나 `postinstall` 스크립트는 이전 파일을 제거해야 합니다. 에이전트는 기존 제품/파일 세트의 파일을 새로운 버전으로 업데이트하기 전에는 이를 제거하지 않습니다.

---

### 참고

새로운 버전으로 업데이트할 수 있는 이전 버전에 대해서는 정리 작업을 수행할 필요가 있습니다. 때때로 이 작업은 이전 버전에만 적용되는 것이 아닙니다.

## 제어 스크립트 테스트

다음 테스트 제안 사항은 모든 테스트 시나리오에 적용되는 것은 아닙니다. 이 테스트를 수행한 뒤에도 제어 스크립트에 문제가 있을 수 있습니다. 예를 들어 개별 파일 세트의 설치/제거를 테스트할 수 있습니다. 그러나 모든 파일 세트가 시스템에 설치되거나 시스템에서 제거된 후에만 표시되는 상호 작용이 있을 수도 있습니다.

마찬가지로 완전히 로드된 시스템의 제어 스크립트를 테스트하는 경우 기본(또는 핵심) 시스템에 속하지 않는 명령을 실행할 때 발생할 수 있는 문제를 놓칠 수도 있습니다. 대상 시스템에 특정 명령이 없는 경우 스크립트에 오류가 발생하게 됩니다.

## 설치 스크립트 테스트

checkinstall, preinstall 및 postinstall 스크립트의 경우 최소한 다음과 같은 테스트를 수행해야 합니다. 모든 테스트는 로컬 시스템(즉, 로컬 설치에 의한)에서 수행될 수 있습니다.

### 1. 기본 테스트:

- swinstall을 실행하여 제품 전체(즉 모든 파일 세트)를 설치합니다. configure 스크립트를 테스트하지 않으려면 제품에 포함시키지 않거나 defer\_configure 옵션을 “true”로 설정합니다.
- 설치가 완료된 후  
<\${SW\_ROOT\_DIRECTORY}var/adm/sw/swagent.log 파일에서 스크립트 또는 스크립트에 의해 생성된 메시지의 형식/내용에 문제가 없는지 확인합니다.
- 결과 파일 시스템을 조사하여 스크립트가 예상한 조치를 수행했는지 확인합니다.
- 같은 제품을 다시 설치하여 테스트를 다시 실행합니다.

### 2. 파일 로드 시간을 줄이려면 reinstall\_files 옵션을 “false”로 설정하고 reinstall\_files\_use\_cksum 옵션을 “false”로 설정합니다.

### 3. 제품의 이전 버전을 이 버전으로 업데이트할 수 있는 경우에는 이전 버전이 설치된 이 제품을 업데이트하여 테스트를 다시 실행합니다.

4. **checkinstall** 스크립트가 대상 시스템의 현재 활동이나 구성에 근거하여 오류나 경고 상태를 생성하는 경우 **checkinstall** 스크립트가 이러한 상태를 올바르게 검색할 수 있도록 그러한 상태를 사용 가능하게 하십시오.
5. 기본 루트 디렉토리("/") 대신 대체 루트 디렉토리(**swinstall -r**)에 설치하여 테스트를 다시 실행합니다. 스크립트가 대체 루트 디렉토리 안에서 모든 작업(있는 경우)을 수행하는지 확인합니다. (이는 스크립트에서 **\$(SW\_ROOT\_DIRECTORY)**를 올바르게 사용하는지 확인합니다.)
6. 제품이 검색 가능한 경우(즉 다른 위치에 설치할 수 있는 경우) 제품을 다른 위치에 설치하여(**swinstall product:new\_location**) 테스트를 다시 실행합니다. 스크립트가 기본 위치가 아닌 새로운 위치에서 작업을 모두 수행했는지 확인합니다. (이는 스크립트에서 **\$SW\_LOCATION**을 올바르게 사용하는지 확인합니다.)
7. 스크립트가 복잡한 경우 제품이 시스템에 올바르게 설치되었다고 자신할 수 있도록 추가적인 테스트를 실행하십시오. 예를 들어 제품 전체를 설치하는 대신 제품의 특정 부분만을 설치해 봅니다.

## 구성 스크립트 테스트

**configure**, **verify** 및 **unconfigure** 스크립트의 경우 최소한 다음 테스트를 수행해야 합니다. 모든 테스트는 로컬 시스템(즉, 로컬 설치에 의한)에서 수행될 수 있습니다.

1. **swinstall**을 실행하여 제품 전체(즉 모든 파일 세트)를 설치합니다. 설치 프로세스를 통해 구성 작업을 수행 및 구성 스크립트를 실행합니다.
  - 설치 및 구성이 완료된 후 **\$(SW\_ROOT\_DIRECTORY)var/adm/sw/swagent.log** 파일에서 **configure** 스크립트 또는 **configure** 스크립트에 의해 생성된 메시지의 형식/내용에 문제가 없는지 확인합니다.
  - 결과 파일 시스템을 조사하여 **configure** 스크립트가 예상된 조치를 수행했는지 확인합니다.
  - 제품 자체를 테스트하여 필요한 구성 작업이 수행되어 제품을 사용할 준비가 되었는지 확인합니다.

2. `swremove`를 실행하여 구성된 제품을 제거합니다.

- 구성 해제 및 제거가 완료된 후  
`#{SW_ROOT_DIRECTORY}var/adm/sw/swagent.log` 파일에서 `unconfigure` 스크립트 또는 `unconfigure` 스크립트에 의해 생성된 메시지의 형식/내용에 문제가 없는지 확인합니다.
- 결과 파일 시스템을 조사하여 `unconfigure` 스크립트가 예상된 “undo” 조치를 수행했는지 확인합니다.

3. `swinstall`을 실행하여 제품 전체를 다시 설치합니다. `configure` 스크립트를 실행하지 않으려면 `defer_configure` 옵션을 “false”로 설정합니다.

- 설치가 완료되면 `swconfig`를 실행시켜 제품을 구성합니다.
- 결과 파일 시스템을 조사하여 `configure` 스크립트가 예상된 조치를 수행했는지 확인합니다.
- 제품 자체를 테스트하여 필요한 구성 작업이 수행되어 제품을 사용할 준비가 되었는지 확인합니다.
- 이제 `swconfig -u`를 실행하여 제품 구성을 해제합니다.
- 결과 파일 시스템을 조사하여 `unconfigure` 스크립트가 예상된 “undo” 조치를 수행했는지 확인합니다.
- `swconfig`를 다시 실행시켜 제품을 다시 구성합니다.
- 결과 파일 시스템을 조사하여 예상된 조치를 수행했는지 확인합니다.

4. `swverify`를 실행하여 `verify` 스크립트를 실행합니다.

- 확인이 완료되면 `#{SW_ROOT_DIRECTORY}var/adm/sw/swagent.log` 파일에서 `verify` 스크립트 또는 `verify` 스크립트에서 생성되는 메시지의 형식/내용에 문제가 없는지 확인합니다.

5. 제품의 이전 버전을 이 버전으로 업데이트할 수 있는 경우에는 이전 버전이 설치되고 구성된 시스템으로 이 제품을 업데이트하여 첫 번째 테스트를 다시 실행합니다.

6. `#{SW_ROOT_DIRECTORY}`가 /가 아닌 경우에는 `configure`와 `unconfigure` 스크립트가 실행되지 않습니다. 그러나 `verify` 스크립트는 두 경우 모두에 실행됩니다.

7. 제품이 검색 가능한 경우(즉 다른 위치에 설치할 수 있다면) 제품을 다른 위치에 설치 및 구성하여 테스트를 다시 실행합니다. 스크립트가 기본 위치가 아닌 새로운 위치에서 작업을 모두 수행했는지 확인합니다. (이는 스크립트에서 `$SW_LOCATION`을 올바르게 사용하는지 확인합니다.)
8. 스크립트가 복잡한 경우 해당 제품이 시스템에 올바르게 설치되었다고 자신할 수 있도록 추가적인 테스트를 실행하십시오. 예를 들어 제품 전체를 설치하는 대신 제품의 특정 부분만을 설치해 봅니다.

## 제거 스크립트 테스트

`checkremove`, `preremove` 및 `postremove` 스크립트의 경우 최소한 다음 테스트를 수행해야 합니다. 모든 테스트는 로컬 시스템(즉, 로컬 설치에 의한)에서 수행될 수 있습니다. 원격 대상 시스템에 설치해서 스크립트를 테스트하는 경우 얻을 수 있는 장점이 없습니다.

1. `swinstall`을 실행하여 제품 전체(즉 모든 파일 세트)를 설치합니다.  
`defer_configure` 옵션을 `false`로 설정하여 구성을 방지합니다.
  - `swremove`를 실행하여 구성 해제된 제품을 제거합니다.
  - 제거가 완료되면 `${SW_ROOT_DIRECTORY}var/adm/sw/swagent.log` 파일에서 `removal` 스크립트 또는 스크립트에 의해 생성된 메시지의 형식/내용에 문제가 없는지 확인합니다.
  - 결과 파일 시스템을 조사하여 `removal` 스크립트가 예상된 조치를 수행했는지 확인합니다.
2. `swinstall`을 실행하여 제품 전체(즉 모든 파일 세트)를 설치합니다. 설치 프로세스를 통해 구성 작업을 수행 및 구성 스크립트를 실행합니다.
  - `swremove`를 실행하여 구성된 제품을 제거합니다.
  - 구성 해제 및 제거가 완료되면 `${SW_ROOT_DIRECTORY}var/adm/sw/swagent.log` 파일에서 `removal` 스크립트 또는 스크립트에 의해 생성된 메시지의 형식/내용에 문제가 없는지 확인합니다.
  - 결과 파일 시스템을 조사하여 `removal` 스크립트가 예상된 조치를 수행했는지 확인합니다.

3. **checkremove** 스크립트가 대상 시스템의 현재 활동이나 구성에 근거하여 오류나 경고 상태를 생성하는 경우 **checkremove** 스크립트가 이러한 상태를 올바르게 감지할 수 있도록 그러한 상태를 사용 가능하게 하십시오.
4. 기본 루트 디렉토리("/") 대신 대체 루트 디렉토리(`swinstall -r`)를 설치하여 첫 번째 테스트를 다시 실행합니다. 스크립트가 대체 루트 디렉토리 안에서 모든 작업(있는 경우)을 수행하는지 확인합니다. (이는 스크립트에서 `${SW_ROOT_DIRECTORY}`를 올바르게 사용하는지 확인합니다.)
5. 제품이 검색 가능한 경우(즉 다른 위치에 설치할 수 있다면) 제품을 다른 위치에 설치하여 테스트를 다시 실행하십시오. 제품을 제거할 때 **removal** 스크립트가 기본 위치가 아닌 새 위치에서 작업을 모두 수행하는지 확인하십시오. (이는 스크립트에서 `$SW_LOCATION`을 올바르게 사용하는지 확인합니다.)
6. 스크립트가 복잡한 경우 해당 제품이 시스템에 올바르게 설치되었다고 자신할 수 있도록 추가적인 테스트를 실행하십시오. 예를 들어 제품 전체를 설치하는 대신 제품의 특정 부분만을 설치한 다음 제거 작업을 수행합니다. (또는 전체 설치된 제품에서 일부만을 제거합니다.)



## 사용자 응답 요구(swask)

패키징된 SD-UX 응용 프로그램은 대화식 제어 스크립트를 사용하여 사용자를 조회하고 패키징할 때는 알 수 없었던 설치 또는 구성 정보를 얻을 수 있습니다. 예를 들어 서로 다른 하드웨어나 OS 버전의 경우 서로 다른 구성이 필요할 수도 있고 일부 소프트웨어에는 구성을 위한 특정 IP 주소 또는 호스트 이름이 필요할 수도 있습니다.

SD-UX는 swask 명령을 사용하거나 swinstall, swconfig 명령에 ask 기본 옵션을 사용하여 대화식 제어 스크립트를 실행합니다. (SD-UX는 사용자를 조회하지 않지만 제어 스크립트는 사용자를 조회합니다.)

### swask 사용

- swask 명령은 선택된 소프트웨어 객체에 대한 대화식 소프트웨어 request 스크립트를 실행합니다.
- 이 스크립트는 swinstall이나 swconfig 명령이 이후에 사용할 수 있도록 응답을 응답 파일(response)에 저장합니다. (swinstall과 swconfig는 ask 옵션을 사용하여 대화식 스크립트를 직접 실행할 수 있습니다.)
- 응답 파일은 해당 request 스크립트를 가지고 있는 선택된 소프트웨어 각각에 대해 생성됩니다.
- swask는 명령줄만 사용하며 그래픽 사용자 인터페이스가 없습니다.

### 구문

```
swask [-v] [-c catalog] [-C session_file] [-f software_file]
[-s source][-S session_file][-x option=value] [-X options_file]
[software_selections][@target_selections]
```

### 옵션 및 피연산자

- |            |   |
|------------|---|
| -v         | 상세 출력을 stdout로 설정하고 모든 작업을 화면에 표시합니다.   |
| -c catalog | request 스크립트가 작성한 응답 파일을 저장하는 내보낸 카탈로그의 경로 이름을 지정합니다.<br>swask는 카탈로그가 없는 경우 카탈로그를 작성합니다.<br><br>-c catalog 옵션이 누락되고 소스가 로컬인 경우, swask는 응답 파일을 소스 저장소인 <code>distribution.path/catalog</code> 로 복사합니다. |

## 제어 스크립트 사용 사용자 응답 요구(*swask*)

### *-C session\_file*

명령을 실행하고 현재 옵션과 피연산자 값을 다른 세션에 다시 사용할 수 있도록 *session\_file*에 저장합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.

### *-f software\_file*

명령줄 대신(또는 그에 추가하여) 별도의 파일에서 대상 선택 목록을 읽습니다. 자세한 내용은 60페이지의 “소프트웨어 파일”을 참조하십시오.

### *-S session\_file*

*session\_file*에 정의된 대로, 이전 설치 세션에서 저장된 값을 근거로 명령을 실행합니다. 자세한 내용은 63페이지의 “세션 파일”을 참조하십시오.

### *-s source*

기본값 */var/spool/sw* 대신 *source*로 정의한 소프트웨어 소스를 사용합니다. 구문은 다음과 같습니다.

```
[host:] [:] [/directory]
```

*host*는 호스트 이름이나 도메인 이름 또는 인터넷 주소(예: 15.1.48.23)입니다. *directory*는 절대 경로입니다.

### *-X option=value*

명령 *option*을 *value*로 설정하고 기본값 또는 옵션 파일의 값을 대체합니다. 자세한 내용은 387페이지의 “명령 옵션 변경”을 참조하십시오.

### *-X option\_file*

*option\_file*에서 세션 옵션 및 동작을 읽습니다. 자세한 내용은 387페이지의 “명령 옵션 변경”을 참조하십시오.

### *software\_selections*

**request** 스크립트가 실행될 소프트웨어 객체입니다. 자세한 내용은 58페이지의 “소프트웨어 선택 사항”을 참조하십시오.

### *target\_selections*

명령의 대상입니다. 자세한 내용은 60페이지의 “대상 선택 사항”을 참조하십시오.

## 명령 옵션 변경

명령을 호출할 때 명령줄 옵션을 추가로 지정(-x 옵션 사용)하거나 파일에서 미리 정의된 값을 읽어서 이 명령의 동작을 변경할 수 있습니다. 다음 표에서는 swconfig에 적용되는 기본값과 옵션을 보여줍니다.

표 11-3

swask 명령 옵션 및 기본값

<ul style="list-style-type: none"><li>• admin_directory=/var/adm/sw</li><li>• ask=true</li><li>• autoselect_dependencies=true</li><li>• autoselect_patches=true</li><li>• enforce_scripts=true</li><li>• installed_software_catalog=products</li><li>• log_msgid=0</li></ul>	<ul style="list-style-type: none"><li>• logdetail=false</li><li>• logfile=/var/adm/sw/swask.log</li><li>• loglevel=1</li><li>• patch_filter=*.*</li><li>• run_as_superuser=true</li><li>• verbose=1</li></ul>
--	---

## 추가 정보

각 기본값에 대한 자세한 설명은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.

---

## request 스크립트 작업 및 예

swinstall 또는 swconfig 명령에서 ask 옵션을 true로 설정하여 request 스크립트를 실행할 수 있습니다. 이렇게 설정하면 설치 및 구성 작업을 수행하는 일 외에도 request 스크립트(있는 경우)를 실행하도록 명령에 지시합니다. (swinstall과 swconfig 명령에서 ask 옵션의 값이 false이지만 swask에서는 true일 경우 ask 옵션의 값에 유의하십시오.)

### swask 예제

모든 request 스크립트를 기본 저장소(/var/spool/sw)에서 실행하고 같은 저장소에 응답 파일(response)을 씁니다.

```
swask -s /var/spool/sw \*
```

원격 호스트 swposix에 있는 저장소 /tmp/sample.depot.1의 Product1에 대해 request 스크립트를 실행하고 로컬 컨트롤러 시스템에서 카탈로그 /tmp/test1.depot를 작성한 다음 응답 파일(response)을 카탈로그에 저장합니다.

```
swask -s swposix:/tmp/sample.depot.1 \  
-c /tmp/test1.depot Product1
```

응답 파일이 없는 경우에만 호스트 swposix의 원격 저장소인 /tmp/sample.depot.1에서 request 스크립트를 실행하고 로컬 컨트롤러 시스템에서 카탈로그 /tmp/test1.depot를 작성한 다음 응답 파일(response)을 카탈로그에 저장합니다.

```
swask -s swposix:/tmp/sample.depot.1 \  
-c /tmp/test1.depot -x ask=as_needed \*
```

### swinstall 예제

request 스크립트로 생성된 응답 파일을 사용하여 로컬 저장소 tmp/sample.depot.1에 모든 소프트웨어를 설치하려면 다음을 수행합니다.

```
swinstall -s /tmp/sample.depot.1 -x ask=true \*
```

호스트 swposix의 원격 저장소 /tmp/sample.depot.1의 Product1을 설치하고 /tmp/bar.depot에 있는 기존 응답 파일(이전에 swask 명령으로 생성된)을 사용하면 다음을 수행합니다.

```
swinstall -s swposix:/tmp/sample.depot.1 \  
-c /tmp/bar.depot Product1
```

호스트 swposix의 원격 저장소인 /tmp/sample.depot.1에 모든 제품을 설치하고 request 스크립트로 생성된 응답을 사용하며 카탈로그 /tmp/bar.depot를 작성한 다음 모든 응답 파일을 새 카탈로그로 복사하려면 다음을 수행합니다.

```
swinstall -s swposix:/tmp/sample.depot.1 \  
-c /tmp/bar.depot -x ask=true \*
```

호스트 swposix의 원격 저장소 /tmp/sample.depot.1에 모든 제품을 설치하고 응답 파일을 사용하며 응답 파일이 없는 경우에만 request 스크립트를 실행하고 카탈로그 /tmp/bar.depot를 생성한 다음 모든 응답 파일을 새 카탈로그로 복사하려면 다음을 수행합니다.

```
swinstall -s swposix:/tmp/sample.depot.1 \  
-c swposix:/tmp/bar.depot -x ask=as_needed \*
```

## swconfig 예제

Product1을 구성하고 request 스크립트로 생성된 관련 응답 파일을 사용하며 응답 파일을 /tmp/resp1 아래에 저장하려면 다음을 수행합니다.

```
swconfig -x ask=true -c /tmp/resp1 Product1
```

제어 스크립트 사용  
request 스크립트 작업 및 예

# 12 비권한 SD

이 장에서는 Software Distributor가 비권한 모드에서 실행될 수 있도록 설정하는 방법에 대한 일반적인 지침을 제공합니다.

표 12-1

## 이 장의 항목

설명하는 항목
392페이지의 “개요”
394페이지의 “비권한 모드 설정”
396페이지의 “기본 구성”
397페이지의 “대체 구성”

## 개요

SD-UX의 비권한 모드를 통해 사용자는 SD-UX ACL에 의해 구현된 슈퍼유저 권한이 아닌 자신이 갖고 있는 파일 시스템 권한을 사용하여 응용 프로그램 소프트웨어에 액세스할 수 있습니다. 비권한 모드는 대부분의 SD 명령을 사용하여 지정할 수 있습니다. 응용 프로그램의 개발, 배포 및 관리에 이르는 모든 측면에 비권한 모드를 사용할 수 있습니다.

## 혜택을 받는 대상은?

비권한 SD-UX는 주로 슈퍼유저 권한이 없어도 내부 응용 프로그램을 관리해야 하는 대규모 데이터 센터의 관리자를 위한 것입니다. 비권한 모드를 활용하여 소프트웨어를 패키징할 수 있는 숙련된 사용자가 아니고 때때로 자신의 응용 프로그램을 관리하고자 하는 사용자라면 이 기능이 큰 도움이 되지 않을 수도 있습니다.

## 작동 방식?

비권한 모드에서 대부분의 SD-UX 작업은 사용자의 uid, gid 및 umask 호출에 따라 이루어집니다. 이 모드에서 주로 /var/adm/sw에 위치한 로그 파일과 설치된 소프트웨어 카탈로그는 사용자 고유의 관리 디렉토리인 /var/home/USER\_NAME/sw (USER\_NAME은 사용자의 로그인 이름)에 저장됩니다. 사용자의 관리 디렉토리와 설치된 소프트웨어 카탈로그의 위치는 기본 옵션을 사용하여 사용자 정의할 수 있습니다.

비권한 모드를 사용하는 동안 비권한 모드에서 사용되지 않는 응용 프로그램을 패키징하고 복사할 수도 있습니다. 그러나 그러한 응용 프로그램을 설치하려면 SD-UX 정상 모드(즉 run\_as\_superuser를 true로 설정하고 ACL이 권한을 부여해야 함)를 사용해야 합니다.

패키징할 때 설치 대상에서의 파일 시스템 액세스를 고려해야 합니다. 자세한 내용은 394페이지의 “비권한 모드에서 사용할 소프트웨어 패키징”을 참조하십시오.



## 제한 사항

- 원격 시스템에 대한 `swlist` 액세스 및 원격 저장소에 정상적으로 액세스할 수 있는 명령을 제외하고는 **SD-UX** 원격 작업에 원격 대상이 허용되지 않습니다. 그러한 원격 시스템에 대한 액세스는 원격 시스템 상의 **SD ACL**에 의해서 결정됩니다.
- **HP-UX** 운영 체제 소프트웨어나 그에 대한 패치를 관리하는 데에는 비권한 모드를 사용할 수 없습니다.
- 비권한 모드에서 `swinstall` 또는 `swcopy`는 슈퍼유저 권한으로 작성된 로컬 쓰기 가능한 파일 시스템의 소스 저장소를 읽을 수 없습니다(즉, 슈퍼유저가 작성하거나 `run_as_superuser` 옵션을 **true**로 설정하고 **ACL** 권한을 사용하여 슈퍼유저가 아닌 사용자가 작성하는 경우). 이 제한 사항은 테이프나 **CD-ROM** 소스 저장소에는 적용되지 않습니다.
- 비권한 모드에서 `swinstall` 및 `swcopy`는 **ACL**이 허용한 원격 소스 저장소는 모두 읽을 수 있고, 비권한 모드에서 호출한 사용자가 작성한 로컬 소스 저장소를 읽을 수 있으며, 다른 사용자의 `umask`에 따라 비권한 모드에서 다른 사용자가 작성한 로컬 소스 저장소를 읽을 수 있습니다.

## 비권한 모드 설정

비권한 SD는 다음 두 옵션으로 제어됩니다.

- `admin_directory`
- `run_as_superuser`

`run_as_superuser` 옵션은 비권한 모드를 켜거나 끌 수 있으며 기본 구성을 실행하기 위해서는 이 옵션으로 충분합니다(395페이지의 “비권한 모드 켜기” 및 396페이지의 “기본 구성” 참조).

`admin_directory` 옵션을 통해 대체 구성을 설정할 수 있습니다(397페이지의 “대체 구성” 참조).

## 비권한 모드에서 사용할 소프트웨어 패키징

이러한 옵션과 함께 비권한 모드에서 사용할 소프트웨어 응용 프로그램에는 특별한 패키징 요구 사항이 있습니다.

비권한 모드가 작동되도록 하려면 다음과 같이 합니다.

- 응용 프로그램을 패키징하고 설치하여 파일이 해당 응용 프로그램을 설치할 사용자가 쓰기 권한을 가지고 있는 위치에 설치되도록 합니다. 이는 다음과 같이 수행합니다.
  - 패키징 시 `PSF`에 디렉토리 키워드를 사용합니다.
  - 명령줄에서 명령을 호출할 때 소프트웨어 사양에 `location`을 추가합니다 (58페이지의 “소프트웨어 선택 사항” 참조).
- 응용 프로그램으로 패키징된 스크립트는 슈퍼유저 권한을 필요로 하지 않도록 설계되어야 합니다.

## 비권한 모드 켜기

SD는 `run_as_superuser` 옵션이 `false`이고 호출하는 사용자가 슈퍼유저가 아닌 경우에만 비권한 모드에서 작동합니다.

이 옵션은 `swagent`, `swagentd`, `swjob` 및 `install-sd`를 제외한 모든 SD-UX 명령에 적용됩니다. 이 옵션을 `false`로 설정하면 적용되는 모든 명령은 비권한 모드로 실행됩니다. 예를 들면 다음과 같습니다.

- 명령줄에 `-x run_as_superuser=false`를 포함시켜 입력하면 해당 명령에 대해서만 비권한 모드가 실행됩니다.
- `$HOME/.swdefaults` 디렉토리에 `-x run_as_superuser=false`를 포함시키면 실행하는 모든 SD-UX 명령에 대해 비권한 모드가 실행됩니다.
- `/var/adm/sw/defaults`에 `-x run_as_superuser=false`를 포함시키면 시스템상의 모든 SD-UX 명령에 대해 비권한 모드가 실행됩니다.

이러한 옵션 사용에 대한 전체 내용은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.

---

### 참고

호출한 사용자가 슈퍼유저인 경우 이 옵션은 무시됩니다(`true`로 처리).

## 비권한 모드로 SD-UX 작동 방식을 바꾸는 방법

`run_as_superuser` 옵션을 기본값인 `true`로 설정하면 SD-UX는 정상적으로 작동되며 작동 권한은 로컬 슈퍼유저에게 부여되거나 SD ACL에 의해 설정됩니다. (ACL에 대한 자세한 내용은 9장, 233페이지의 “SD-UX 보안”을 참조하십시오.)

`run_as_superuser`를 `false`로 설정하고 호출한 사용자가 슈퍼유저가 아닌 로컬 사용자인 경우 비권한 모드가 호출됩니다.

- 작동 권한은 사용자의 파일 시스템 권한에 근거합니다.
- SD ACL은 무시됩니다.
- SD에 의해 작성된 파일은 호출한 사용자의 `uid`와 `gid`를 갖게 되고 작성된 파일의 모드는 호출한 사용자의 `umask`에 따라 설정됩니다.

## 기본 구성

비권한 모드의 기본 구성에는 사용자가 설치한 소프트웨어 카탈로그에 대한 중앙 위치가 있어야 합니다.

`run_as_superuser` 옵션이 **false**이고 `admin_directory` 옵션을 설정하지 않으면 **SD-UX** 로그 파일과 설치된 소프트웨어 카탈로그는 사용자 고유 디렉토리인 `/var/home/USER_NAME/sw`(`USER_NAME`은 호출한 사용자 이름으로 바뀜)에 저장됩니다.

로그 파일과 설치된 소프트웨어 카탈로그를 중앙의 위치에 둬으로써 소프트웨어를 홈 디렉토리 외부의 시스템에 설치하고 사용자 홈 디렉토리가 여러 시스템에 걸쳐 마운트되어 있는 **NFS**인 경우 발생하는 문제점을 방지할 수 있습니다.

`/var/adm/sw/defaults`에서 `run_as_superuser` 옵션을 **false**로 설정함으로써 모든 사용자가 비권한 모드로 사용할 수 있도록 설정할 수 있습니다.

사용자들은 개별적으로 자신의 `$HOME/.swdefaults` 파일이나 명령줄에서 `run_as_superuser` 옵션을 **true**나 **false**로 설정하여 시스템 관리자가 선택한 기본 값을 바꿀 수 있습니다.

## 대체 구성

비권한 모드의 대체 구성은 각 사용자의 홈 디렉토리에 사용자가 설치한 소프트웨어 카탈로그를 설정합니다. `/var/adm/sw/defaults`의 `admin_directory` 옵션을 사용하여 `HOME` 또는 `/HOME`으로 시작되는 경로를 나타내어 비권한 모드가 실행되는 동안 **SD-UX**가 사용하는 기본 관리 디렉토리가 각 사용자의 홈 디렉토리에 있도록 합니다. (이러한 목적에는 `HOME/.sw` 값이 적절합니다.)

사용자는 개별적으로 `$HOME/.swdefaults` 파일이나 명령줄에서 이 값을 바꿀 수 있습니다.

## 관리 디렉토리 옵션 설정

이 옵션을 통해 로그 파일의 위치와 설치된 소프트웨어 카탈로그에 대한 기본 상위 디렉토리를 지정할 수 있습니다. 그 값은 다음과 같습니다.

**admin\_directory=/var/adm/sw**(정상 모드)

**admin\_directory=/var/home/LOGNAME/sw**(비권한 모드)

정상 작동의 경우 기본값은 `/var/adm/sw`입니다. 비권한 모드(즉, `run_as_superuser` 옵션을 `true`로 설정)의 경우 다음과 같습니다.

- 기본값은 `/var/home/LOGNAME/sw`로 강제 지정됩니다.
- 경로 요소인 `LOGNAME`은 **SD-UX**가 시스템 암호 파일에서 읽은 호출 사용자의 이름으로 대체됩니다.
- 이 옵션 값을 `HOME/path`로 설정하면 **SD-UX**는 `HOME`을 호출 사용자의 홈 디렉토리(시스템 암호 파일에서 읽음)로 대체하고 그 디렉토리에 대한 상대 경로로 해석합니다.

예를 들어 이 옵션에 `HOME/my_admin`을 지정한 경우 해당 위치는 사용자의 홈 디렉토리 내의 `my_admin` 디렉토리로 간주됩니다.

이 옵션은 `swinstall`, `swcopy`, `swremove`, `swconfig`, `swverify`, `swlist`, `swreg`, `swacl`, `swpackage`, `swmodify`에 적용됩니다.

비권한 SD  
대체 구성

# A 명령 옵션

이 부록에서는 SD-UX 명령 옵션 변경에 대한 기본 사항을 검토하고 모든 옵션과 기본 값을 알파벳 순서로 설명합니다.

표 A-1

이 장의 항목

설명하는 항목
400페이지의 “명령 옵션 변경”
402페이지의 “옵션(알파벳 순)”

## 명령 옵션 변경

옵션 값 변경을 통해 명령의 작동 방식을 바꾸고 사용자 요구에 맞게 **SD-UX** 정책을 조정할 수 있습니다. 사전 정의된 파일, 명령줄에서 직접 지정한 값 또는 **Options** 메뉴에서 선택한 **GUI Options Editor** 등을 사용하여 옵션을 변경할 수 있습니다. 파일을 사용하여 옵션 값을 변경하면 명령을 호출할 때마다 명령 작동 방식을 지정하고 싶지 않을 때 도움이 됩니다.

이 규칙은 옵션이 작동하는 방식을 제어합니다.

- `/var/adm/sw/defaults`에 지정된 옵션 값은 시스템상의 모든 **SD-UX** 명령에 영향을 미칩니다. 이 파일은 옵션이 적용되는 모든 명령에 대해서나 특정 명령에 대해서만 옵션을 변경할 수 있습니다.
- 사용자 개인의 `$HOME/.swdefaults` 파일에 있는 옵션 값은 해당 사용자에게만 영향을 미치며 시스템 전체에는 영향을 미치지 않습니다.
- 세션 파일로부터 읽은 옵션 값은 해당 세션에만 영향을 미칩니다.
- `-X option_file` 또는 `-x option=value` 인수를 사용하여 명령줄에서 옵션을 변경하면 시스템 전체 및 개인 기본 파일을 재정의하지만 실행된 해당 명령에 한해 영향을 미칩니다.

시스템 전체에 적용되는 정책을 설정하려면 `/var/adm/sw/defaults` 파일을 사용합니다. 그러나 사용자의 `$HOME/.swdefaults` 파일, 세션 파일 또는 명령줄 변경을 통해 이러한 값이 바뀔 수 있음을 명심하시기 바랍니다.

`/usr/lib/sw/sys.defaults` 템플릿 파일은 모든 옵션에 대한 설명을 제공하며 시스템 전체 또는 개인의 기본 파일을 쉽게 변경하기 위한 지침이 들어 있습니다.

템플릿 파일에는 모든 **SD-UX** 명령 옵션, 이 옵션이 적용되는 명령과 가능한 옵션 값 및 시스템 작동 결과가 주석의 형태로 나와 있습니다. 이 파일의 값을 시스템 기본 파일(`/var/adm/sw/defaults`), 사용자 개인 기본 파일(`$HOME/.swdefaults`) 또는 입력 파일로 복사하고 주석 표시를 제거하여 실제 시스템에 작동하도록 만들 수 있습니다.



옵션 파일은 다음과 같은 구문을 사용합니다.

`[command.]option=value`

- 선택 사항인 `command`는 **SD-UX** 명령의 이름입니다. 명령 이름을 지정하면 해당 명령의 기본 작동만 변경됩니다. 명령 이름 다음에는 반드시 마침표(.)가 있어야 합니다.
- `option`은 기본 옵션의 이름입니다. 옵션 이름 다음에 = 기호가 나와야 합니다.
- `value`는 해당 옵션에 대해 사용할 수 있는 값입니다.

---

### 참고

기본 옵션 값을 변경할 때는 주의가 필요합니다. 이러한 유연성은 유용하게 활용할 수 있지만 사용자 요구에 맞지 않는 값으로 변경하면 부정적인 결과를 가져올 수 있습니다.

---

### 참고

기본 파일의 옵션은 명령 초기화의 일부로 읽어들입니다. 데몬이 이미 실행 중이므로 데몬 옵션을 변경한 후에는 시스템이 이 옵션을 인식할 수 있도록 데몬을 다시 시작해야 합니다. 데몬을 다시 시작하려면 다음과 같이 입력합니다.

```
/usr/sbin/swagentd -r
```

---

### 관련 항목

예제를 보려면 61페이지의 “명령 옵션 사용”을 참조하십시오.

## 옵션(알파벳 순)

- **admin\_directory=/var/adm/sw**(정상 모드)  
**admin\_directory=/var/home/LOGNAME/sw**  
(비권한 모드)

로그 파일 및 설치된 소프트웨어 카탈로그의 기본 상위 디렉토리 위치입니다. 일반적인 작동의 경우 기본값은 /var/adm/sw입니다.

비권한 모드(즉, run\_as\_superuser 기본 옵션이 true인 경우)에서는 다음과 같습니다.

- 기본값은 /var/home/LOGNAME/sw입니다.
- 경로 요소인 LOGNAME은 SD-UX가 시스템 암호 파일에서 읽은 호출 사용자의 이름으로 대체됩니다.
- 이 옵션 값을 HOME/path로 설정하면 SD-UX는 HOME을 호출 사용자의 홈 디렉토리(시스템 암호 파일에서 읽음)로 대체하고 그 디렉토리에 대한 상대 경로로 해석합니다. 예를 들어 HOME/my\_admin을 사용자의 홈 디렉토리에 있는 my\_admin 디렉토리로 해석합니다.
- installed\_software\_catalog 옵션 값을 상대 경로로 설정하면 해당 경로가 이 옵션 값의 상대 경로로 해석됩니다.

비권한 모드는 특별하게 설계되고 패키징된 응용 프로그램 관리를 목적으로 하는 경우에 한해 사용됩니다. 이 모드는 HP-UX 운영 체제나 그에 대한 패치를 관리하는 데에 사용할 수 없습니다. 비권한 SD-UX에 대한 전체적인 설명을 보려면 12장, 391페이지의 “비권한 SD”를 참조하십시오.

installed\_software\_catalog 및 run\_as\_superuser 옵션도 참조하십시오.

swagent, swagentd 및 install-sd를 제외한 모든 명령에 적용됩니다.

- **agent=/usr/sbin/swagent**

이것은 에이전트 작업을 수행하기 위해 호출되는 실행 파일의 기본 위치입니다. swagentd에 적용됩니다.

- **agent\_auto\_exit=true**

실행 단계를 종료한 후 또는 분석 단계에서 실패한 후 자동으로 대상 에이전트가 종료되도록 합니다. 컨트롤러가 대화식 UI를 사용하고 있거나 -p(미리 보기)를 사용하는 경우 이 옵션은 **false**로 강제 설정됩니다.

네트워크 안정성과 성능을 향상시킵니다.

기본값은 **true**입니다. 해당되는 대상 에이전트가 자동으로 종료됩니다.

**false**로 설정하면 대상 에이전트는 컨트롤러가 명시적으로 세션을 종료할 때까지 종료되지 않습니다.

swconfig, swcopy, swinstall, swremove 및 swverify에 적용됩니다.

- **agent\_timeout\_minutes=10000**

지정된 시간 동안 활성화되지 않으면 대상 에이전트가 종료되도록 합니다.

이 기본값을 사용하여 대상 에이전트가 단절된 네트워크 연결을 더 신속하게 찾을 수 있도록 할 수 있습니다. RPC는 일반적으로 단절된 연결을 매우 신속하게 찾아내지만 130분까지 걸릴 수도 있습니다. 권장하는 값은 사용자 환경에서 가장 하다고 생각되는 가장 긴 비활성 시간 값입니다.

명령줄에서 실행할 때 값은 10분에서 60분 사이가 적합합니다. GUI를 사용할 때는 60분 이상이 권장됩니다.

기본값은 10,000분으로 7일에 조금 못 미치는 기간입니다.

swcopy, swinstall, swlist, swremove 및 swverify에 적용됩니다.

- **allow\_downdate=false**

일반적으로 **false**로 설정하여 이미 설치되어 있는 것보다 이전 버전의 소프트웨어는 설치할 수 없도록 합니다. 이 옵션을 통해 실수로 이전 버전을 설치하는 것을 방지할 수 있습니다. 또한 많은 소프트웨어 제품에서 이러한 “다운데이트”를 지원하지 않습니다.

**true**로 설정하면 이전 버전이 설치될 수 있지만 SD-UX에서 경고 메시지를 표시합니다.

swinstall에 적용됩니다.

- **allow\_incompatible=false**

일반적으로 **false**로 설정하여 로컬 호스트와 호환되는 소프트웨어만 설치 또는 구성될 수 있도록 합니다.

**true**로 설정하면 호환성 검사를 하지 않습니다.

swconfig, swinstall 및 swverify에 적용됩니다.

- **allow\_multiple\_versions=false**

일반적으로 **false**로 설정하면 여러 버전을 설치하거나 구성하는 작업(즉, 동일 제품의 다른 개정본을 다른 위치에 설치)이 허용되지 않습니다. 소프트웨어 위치를 검색할 수 있는 경우 여러 개의 소프트웨어 버전을 **설치**했다라도 제품에서 지원하지 않으면 여러 개의 **구성된** 버전은 작동하지 않습니다.

**true**로 설정하면 같은 소프트웨어의 여러 버전을 설치 및 관리할 수 있습니다.

swconfig, swinstall 및 swverify에 적용됩니다.

- **allow\_partial\_bundles=true**

로그 파일에 경고나 메시지를 보내지 않고 부분 번들 처리 여부를 결정합니다.

**true**(기본값)로 설정하면 swpackage는 유실되었거나 모호한 번들 내용은 무시하고 소스 PSF에서 사용할 수 있는 것만 패키징합니다. 번들은 제품을 싸고 있는 포장으로 번들 내용이 PSF에 없는 경우 결과 저장소에 들어 있는지 확인해야 합니다. 그렇지 않으면 유실되었거나 모호한 내용물이 설치 후 시스템에 영향을 미치지 않을 것입니다.

**false**로 설정하면 swpackage는 번들 내용이 소스 PSF에 모두 들어 있으며 고유하다고 가정합니다. 모호하거나 유실된 모든 내용은 메시지를 받게 되고 유실되었거나 모호한 내용이 들어 있는 번들은 모두 경고를 받게 됩니다.

swpackage에만 적용됩니다.

- **allow\_split\_patches=false**

패치의 일부를 설치하거나 복사하는 기능을 제어합니다. 이 옵션은 HP 지원 담당자의 도움을 받아 중요한 문제를 해결해야 하는 경우에만 사용하십시오.

기본값인 **false**로 설정하면 대상의 **ancestor** 속성을 바탕으로 여러 파일 세트 패치에서 하나의 파일 세트를 설치하거나 복사할 때 해당되는 다른 “형제” 파일 세트를 자동으로 포함합니다. 이러한 기능은 사용자가 직접 선택한 파일 세트와 패치 파일 세트의 종속성을 만족시키기 위해 자동으로 선택된 파일 세트에 적용됩니다. 마찬가지로 이 옵션이 **false**일 때 파일 세트를 제거하면 형제 파일 세트도 함께 제거됩니다.

이 옵션을 **true**로 설정하면 하나의 패치 파일 세트가 형제 파일 세트를 포함하지 않고 대상에 또는 대상으로부터 설치, 복사 또는 제거됩니다(즉, 이 옵션을 **false**로 설정하면 대상에 상위 파일 세트가 있는 파일 세트가 일반적으로 로드됩니다). 이를 통해 대상에는 구성 요소 파일 세트로 “분할된” 패치가 포함될 수 있습니다. 형제 그룹의 특정 파일 세트만 업데이트되고 다른 파일 세트는 이전 버전으로 남아 있는 경우 부정적인 결과가 발생할 수 있습니다. 이 옵션을 사용한 후 대상에서 **swlist**를 실행하면 활성화되어 있는 패치가 한 번에 두 개 이상 표시될 수 있습니다. 이 경우 시스템 유지 보수 및 문제 해결이 더 어려워집니다.

**swcopy**, **swinstall** 및 **swremove**에 적용됩니다.

- **alternate\_source=**

구문은 **host:path**이며 **use\_alternate\_source** 기본값이 **true**로 설정될 때 사용됩니다.

기본적으로 이 옵션은 정의하지 않습니다. 호스트 부분을 지정하지 않으면 로컬 호스트가 사용됩니다. 경로를 지정하지 않으면 명령에 지정한 경로가 사용됩니다. **use\_alternate\_source** 옵션을 참조하십시오.

에이전트가 대체 소스 저장소에 액세스하려고 할 때

**rpc\_binding\_info\_alt\_source** 옵션으로 제공되는 프로토콜 순서와 엔드 포인트가 사용됩니다.

**swagent**에만 적용됩니다.

- **ask=true**

사용자 응답을 요구하는 **request** 스크립트를 실행합니다.

**ask** 옵션에는 세 가지 값이 가능합니다.

**true** (swask의 경우 기본값) **request** 스크립트를 실행하고(선택된 소프트웨어에 대해 **request** 스크립트가 있는 경우) **response** 파일에 사용자 응답을 저장합니다.

- false** (swinstall 및 swconfig의 경우 기본값) request 스크립트를 실행하지 않습니다.
- as\_needed** request 스크립트를 실행하기 전에 swask는 먼저 응답 파일이 있는지 확인하고 응답 파일이 제어 디렉토리에 없는 경우에만 request 스크립트를 실행합니다.

swask 명령 및 request 스크립트 작성에 대한 자세한 내용은 385페이지의 “사용자 응답 요구(swask)”를 참조하십시오.

swask, swconfig 및 swinstall에 적용됩니다.

- **auto\_kernel\_build=true**

일반적으로 true로 설정합니다. 커널 파일 세트 제거 시 커널을 다시 빌드해야 하는지 여부를 지정합니다. 커널을 다시 빌드하면 시스템은 자동으로 다시 부팅됩니다. false로 설정하면 시스템은 현재의 커널을 계속 실행합니다.

auto\_kernel\_build 옵션이 true로 설정된 경우에는 autoreboot 옵션도 true로 설정되어야 합니다. auto\_kernel\_build 옵션이 false로 설정된 경우 autoreboot 옵션 값은 중요하지 않습니다.

swremove에만 적용됩니다.

- **autoreboot=false**

일반적으로 false로 설정되며, 이는 부팅을 다시 해야 하는 소프트웨어 설치 명령줄에서 수행할 수 없음을 나타냅니다.

true로 설정하면 이 옵션은 소프트웨어의 설치나 제거를 허용하며 자동으로 로컬 호스트를 다시 부팅합니다.

auto\_kernel\_build 옵션이 true로 설정된 경우에는 이 옵션도 true로 설정되어야 합니다.

swinstall 및 swremove에 적용됩니다.

- **autorecover=false**

이 옵션은 설치 오류가 발생할 경우 소스 파일 세트의 자동 복구를 허용합니다. 그에 대한 일시적인 디스크 사용 공간 증가 및 성능 저하가 뒤따릅니다. 기본값 false를 설정하면 파일 세트가 업데이트될 때 swinstall이 소스 파일을 덮어씁니다. 설치 시 오류가 발생하는 경우(예: 네트워크 장애) 소스 파일이 유실되면 파일 세트를 다시 설치해야 합니다.

**true**로 설정하면 현재 파일 세트의 로드가 완료될 때까지 모든 파일이 백업 사본으로 저장됩니다. 설치 시 오류가 발생하면 파일 세트의 소스 파일이 복원되고 **swinstall**은 제품의 다음 파일 세트를 계속 설치하거나 제품의 **postinstall** 스크립트를 실행합니다.

**true**로 설정하면 이 옵션도 스크립트에 영향을 미칩니다. 예를 들어 **preinstall** 스크립트에 오류가 발생하면 이 옵션을 통해 해당 **unpreinstall** 스크립트가 실행됩니다.

**swinstall**에만 적용됩니다.

- **autorecover\_product=false**

기본적으로 **swinstall**은 오래된 파일을 덮어씁니다. 로드 오류가 발생하면 제품을 “**corrupt**” 로 표시되고 설치를 다시 시도해야 합니다.

이 옵션을 **true**로 설정하면 **swinstall**은 제품이 정상적으로 로드를 완료할 때까지 모든 제품 파일을 백업으로 저장한 다음 로드를 완료하면 백업을 제거합니다. 로드 실패하면 **swinstall**은 자동으로 파일을 복구합니다. 그에 대한 일시적인 디스크 사용 공간 증가 및 성능 저하가 뒤따릅니다.

참고: 파일을 이동시키거나 제거하는 **preinstall** 스크립트를 갖고 있는 소프트웨어가 있는 경우 자동 복구 작업이 올바르게 작동하지 않습니다. 여기에는 **HP-UX** 운영체제 파일이 포함됩니다.

**swinstall**에만 적용됩니다.

- **autoremove\_job=false**

**SD-UX**는 각 작업에 대해 작은 양의 정보(작업 상태, 컨트롤러 또는 에이전트로 그 파일 등)만을 저장합니다. 이 정보를 작업 브라우저나 **swjob**을 사용하여 표시할 수 있습니다. 대량의 작업을 실행시키는 경우 디스크 공간을 많이 차지할 수 있습니다.

이 옵션을 **true**로 설정하면 **SD-UX**가 작업 정보를 저장하지 못하게 할 수 있습니다. 그 대신 더 이상 작업 정보를 표시할 수 없습니다.

**run\_as\_superuser**가 **true**로 설정되면 이 옵션도 자동으로 **true**로 설정됩니다.

**swconfig**, **swcopy**, **swinstall**, **swremove** 및 **swverify**에 적용됩니다.

- **autoselect\_dependencies=true**

소프트웨어가 선택될 때 필수 SD-UX가 자동으로 선택하도록 합니다. 기본값인 **true**로 설정하면 종속되는 소프트웨어는 필수 소프트웨어와 함께 소프트웨어를 선택할 때 자동으로 선택됩니다. **false**로 설정하면 자동 선택이 수행되지 않으므로 필수 소프트웨어가 함께 선택되지 않습니다.

swconfig, swcopy, swinstall 및 swverify에 적용됩니다.

- **autoselect\_dependents=false**

소프트웨어가 선택될 때 swconfig 및 swremove를 통해 자동으로 종속 소프트웨어를 선택하도록 합니다. 이 옵션을 **true**로 설정하고 다른 소프트웨어가 종속되는 소프트웨어를 선택한 경우 SD-UX에서는 그에 종속된 소프트웨어도 선택됩니다. 이미 선택되어 있는 경우가 아니라면 자동으로 선택됩니다. **false**로 설정하면 종속 소프트웨어는 자동으로 선택되지 않습니다.

종속되는 파일 세트는 선택된 파일 세트에 대한 전제 조건, 필수 조건 또는 제외 조건 중 하나가 됩니다. 기본값인 **false**를 설정하면 종속 소프트웨어의 자동 선택이 수행되지 않습니다. **true**로 설정하면 SD-UX는 종속 소프트웨어를 자동으로 선택합니다.

swconfig 및 swremove에 적용됩니다.

- **autoselect\_patches=true**

swinstall이나 swcopy 명령을 실행할 때 사용자가 선택한 소프트웨어 객체에 대해 대체 및 상위 속성을 바탕으로 자동으로 최신 패치를 선택합니다. **false**로 설정하면 선택된 객체에 해당되는 패치가 자동 선택되지 않습니다.

autoselect\_patches와 함께 patch\_filter= 옵션을 사용할 수 있습니다.

swask, swinstall 및 swcopy에 적용됩니다.

- **autoselect\_reference\_bundles=true**

**true**로 설정하면 참조된 번들이 참조한 소프트웨어와 함께 설치, 복사 또는 제거됩니다.

swcopy, swinstall 및 swremove에 적용됩니다.



- **check\_contents=true**  
일반적으로 **true**로 설정하며, 파일의 *mtime*, *size* 및 *cksum*을 확인합니다.  
**false**로 설정하면 이를 포함하는 번들 없이 소프트웨어가 설치될 수 있습니다.  
**swverify**에만 적용됩니다.
- **check\_contents\_uncompressed=false**  
파일이 압축되면, **SD-UX**가 이 파일을 **check\_contents** 및 **check\_contents\_use\_cksum**과 함께 사용하여 계산 여부를 결정하고 압축된 파일의 압축되지 않은 체크섬과 크기를 확인합니다. (파일이 압축되지 않으면 이 옵션은 무시됩니다.)  
압축되지 않은 내용에 대한 타임스탬프는 의미가 없으므로 이 옵션은 압축 여부에 상관없이 파일의 타임스탬프만 확인합니다.  
**true**로 설정하고 파일이 압축되어 있는 경우 **SD-UX**는 메모리에 압축된 파일을 풀고 압축되지 않은 내용의 체크섬과 크기를 계산합니다. 그런 다음 압축된 파일의 체크섬과 크기 및 압축되지 않은 내용의 체크섬과 크기를 확인합니다.  
**check\_contents\_use\_cksum=false**인 경우 압축된 크기와 압축되지 않은 크기만 확인되고 체크섬은 확인되지 않습니다.  
**swverify**에만 적용됩니다.
- **check\_contents\_use\_cksum=true**  
일반적으로 **true**로 설정하며 확인되는 파일의 체크섬을 계산하고 타임스탬프, 크기 및 체크섬을 검사합니다.  
**false**로 설정하면 체크섬을 계산하지 않습니다. **check\_contents**가 **true**로 설정된 경우, 타임스탬프와 크기 확인 작업이 여전히 수행됩니다.  
**swverify**에만 적용됩니다.
- **check\_permissions=true**  
일반적으로 **true**로 설정하며 파일의 소유자, *uid*, 그룹, *gid* 및 모드 속성을 확인합니다.  
**swverify**에만 적용됩니다.

- **check\_requisites=true**

일반적으로 **true**로 설정되며 파일 세트의 전제 조건 및 필수 조건이 만족되는지 확인합니다.

swverify에만 적용됩니다.

- **check\_scripts=true**

일반적으로 **true**로 설정하며 소프트웨어를 확인할 때 공급업체에서 제공한 verify 스크립트를 실행합니다.

swverify에만 적용됩니다.

- **check\_volatile=false**

**true**로 설정하면 swverify가 is\_volatile 속성이 설정되어 설치된 파일을 확인합니다. 기본적으로 설치되어 있는 활성 파일은 고객이 수정할 것이므로 속성을 확인할 수 없습니다.

swverify에만 적용됩니다.

- **codeword=**

사용자가 HP-UX 사용권 프로시저에 대한 코드워드를 입력할 수 있도록 합니다. 한 번 입력하면 코드워드를 다시 입력할 필요가 없습니다. 자세한 내용은 36페이지의 “보호되는 소프트웨어에 대한 작업”을 참조하십시오.

swcopy, swinstall 및 swlist에 적용됩니다.

- **compress\_cmd=/usr/contrib/bin/gzip**

설치, 복사 또는 패키징하기 전에 파일을 압축하기 위해 소스 에이전트에 의해 호출된 명령을 지정합니다.

compression\_type 옵션을 gzip 또는 compress 이외의 값으로 설정하는 경우 이 경로를 변경해야 합니다.

swagent 및 swpackage에 적용됩니다.

- **compress\_files=false**

전송 중에 파일 압축을 제어합니다. **false**로 설정하면 원격 소스로부터 전송되기 전에 파일이 압축되지 않습니다.

**true**로 설정하면 SD-UX는 이미 압축되어 있지 않은 한 네트워크 전송이 이루어지기 전에 파일을 압축합니다. swinstall의 경우 파일은 네트워크 전송 후 압축이 풀립니다.

swcopy 또는 swpackage 실행 중에 **true**로 설정하는 경우 `uncompress_files`도 **true**로 설정하지 않으면 저장소가 더 작게 만들어집니다.

swcopy, swinstall 및 swpackage에 적용됩니다.

- **compress\_index=false**

더 큰 IPD(Installed Products Database)와 저장소 카탈로그로 인해 디스크 사용량은 증가하지만 속도가 느린 네트워크에서는 성능을 향상시킵니다. 기본값인 **false**는 INDEX와 INFO 파일을 압축하지 않습니다. **true**로 설정하면 INDEX와 INFO 파일이 압축됩니다.

swinstall, swcopy, swpackage, swmodify, swconfig 및 swremove에 적용됩니다.

- **compression\_type=gzip**

전송 중에 또는 전송 후에 파일을 압축할 때 에이전트에 의해 사용되는(또는 swpackage에 의해 설정되는) 기본 압축 유형을 정의합니다.

`uncompress_files`를 **false**로 설정하면 압축된 각 파일에 대한 압축 유형이 기록되므로 swinstall을 실행하거나 `uncompress_files`를 **true**로 설정하여 swcopy를 실행할 때 올바르게 압축이 풀릴 수 있습니다.

`compress_cmd`를 지정하면 `compression_type`이 지정되어 파일이 생성됩니다.

형식이 내부 압축 해제 기능(`funzip`)으로 압축이 풀리는 **gzip**이 아닌 한

`uncompress_cmd`는 `compression_type`이 지정된 파일을 처리할 수 있어야 합니다. **gzip**을 사용하려면 SW-DIST.GZIP 파일 세트(선택 가능한 프리웨어)를 로드해야 합니다. SW-DIST.GZIP 파일 세트가 로드되면 다음과 같이 압축 옵션을 설정할 수 있습니다.

```
compress_cmd=/usr/contrib/bin/gzip
```

```
uncompress_cmd=/usr/contrib/bin/gunzip
```

```
compression_type=gzip
```

swpackage 및 swagent에 적용됩니다.

- **config\_cleanup\_cmd=/usr/sbin/sw/config\_clean**

릴리즈별 구성 클린업 단계를 수행하기 위해 에이전트가 호출하는 스크립트를 정의합니다.

swagent에만 적용됩니다.

- **control\_files=**

제어 파일 객체를 추가하거나 삭제할 때 이 옵션은 해당 제어 파일의 태그를 나열합니다. 기본값은 제공되지 않습니다. (추가되는 제어 파일 객체도 해당 제품 사양 파일에 지정할 수 있습니다.)

태그가 둘 이상 있는 경우 공백으로 구분하고 인용 부호 안에 넣어야 합니다.

swmodify에만 적용됩니다.

- **controller\_source=**

컨트롤러가 선택 사항을 해결하기 위해 액세스할 수 있도록 저장소의 위치를 지정합니다. 이 옵션을 설정하면 컨트롤러와 대상 사이의 네트워크 소통량을 줄일 수 있습니다. 다음과 같은 대상 선택 구문을 사용하여 위치를 지정합니다.

*[host][:][path]*

이 옵션은 대상이 사용하는 소스에는 영향을 미치지 않으며 대화식 사용자 인터페이스를 사용할 때에는 무시됩니다.

swconfig, swcopy, swinstall, swremove 및 swverify에 적용됩니다.

- **create\_target\_acls=true**

일반적으로 **true**로 설정하며, 이 기본값은 swpackage가 ACL(Access Control List)을 저장소에 작성하는지 여부를 결정합니다.

수퍼유저로서 이 옵션을 **false**로 설정하면 패키징되는 각각의 새로운 제품 및 저장소에 대한 ACL이 작성되지 않습니다.

다른 사용자가 swpackage를 호출할 때는 항상 배포 저장소에 ACL을 작성합니다. 이 기본값은 이미 저장소에 들어 있는 ACL에는 아무런 영향도 미치지 않습니다. swpackage 명령은 소프트웨어가 배포 테이프에 패키징될 때는 절대로 ACL을 작성하지 않습니다.

swpackage에만 적용됩니다.

- **create\_target\_path=true**

일반적으로 **true**로 설정하며 대상 디렉토리가 없으면 대상 디렉토리를 만듭니다. **false**로 설정하면 대상 디렉토리를 만들지 않습니다. 이 옵션은 실수로 저장소를 새로 작성하는 것을 방지하기 위해 사용할 수 있습니다.  
swcopy 및 swinstall에 적용됩니다.

- **create\_time\_filter=0**

누적 소스 저장소에 대한 시간 설정을 제어합니다. 기본값 **0**을 설정하면 소프트웨어 선택 사항 및 다른 옵션에 근거하여 선택 및 종속 소프트웨어와 패치를 자동 선택할 수 있는 대상으로 소스 저장소에 모든 번들, 제품, 하위 제품 및 파일 세트를 포함합니다. 시간(기준 시점으로부터 초 단위로 지정)으로 설정하면 **create\_time** 이 지정된 값 이하인 번들, 제품 및 파일 세트 및 제품 내 하위 제품만 선택(또는 자동 선택) 대상이 됩니다.

번들, 제품 및 파일 세트의 **create\_time**을 표시하려면 다음과 같이 입력합니다.

```
swlist -a create_time -a create_date
```

swlist, swcopy 및 swinstall에 적용됩니다.

- **customer\_id=**

소프트웨어 인증서에 인쇄된 이 번호는 보호된 소프트웨어의 “잠금을 해제”하고 특정 사이트나 소유자만 가능하도록 설치를 제한합니다. **-x customer\_id=** 옵션 또는 대화식 사용자 인터페이스를 사용하여 번호를 입력할 수 있습니다. 자세한 내용은 **codeword** 옵션을 참조하십시오.

swinstall, swcopy, swlist에 적용됩니다.

- **defer\_configure=false**

swinstall 소프트웨어 선택 사항이 설치된 후 **configure** 스크립트의 자동 실행을 제어합니다. 기본값인 **false**를 설정하면 swinstall은 자동으로 **configure** 스크립트를 실행합니다. **true**로 설정하면 swinstall은 **configure** 스크립트를 실행하지 않습니다. 나중에 소프트웨어를 구성하려면 **swconfig** 명령을 실행해야 합니다.

— 다른 버전이 이미 구성되어 있는 경우 한 제품의 여러 버전이 자동으로 구성되지 않습니다. **swconfig** 명령을 사용하여 여러 버전의 경우 각각 별도로 구성합니다.

— SD-UX는 시스템 재부팅이 필요한 소프트웨어를 설치하는 경우 이 옵션을 무시합니다(true로 간주).

— 대체 루트 디렉토리는 구성되지 않습니다.

swinstall에만 적용됩니다.

- **distribution\_source\_directory=/var/spool/sw**

*source\_type* 옵션 값이 *directory*인 경우 기본 소스 저장소를 정의합니다.

*host:path* 구문을 사용할 수도 있습니다. *-s* 옵션은 이 기본값을 바꿉니다.

swcopy, swinstall 및 swpackage에 적용됩니다.

- **distribution\_target\_directory= /var/spool/sw**

대상 저장소의 기본 배포 디렉토리를 정의합니다. *target\_selection* 명령 실행은 이 기본값을 바꿉니다.

swacl, swcopy, swlist, swmodify, swpackage, swreg, swremove 및 swverify에 적용됩니다.

- **distribution\_target\_serial=/dev/rmt/0m**

대상 테이프 장치 파일의 기본 위치를 정의합니다. *target\_selection* 작업은 이 기본값을 바꿉니다.

swpackage에만 적용됩니다.

- **enforce\_dependencies=true**

**true**로 설정하면 SD-UX는 종속성을 적용합니다. 필요한 종속성을 선택할 수 없거나 적절한 상태(설치됨, 구성됨 또는 사용 가능)로 이미 존재하지 않으면 swinstall, swcopy 및 swconfig가 처리되지 않습니다. 이 옵션으로 SD-UX가 유용하지 않은 소프트웨어를 설치하거나 복사하는 것을 방지할 수 있습니다. 또한 이 옵션으로 swremove가 종속 소프트웨어를 제거하는 것도 방지할 수 있습니다.

**false**로 설정하면 SD-UX는 종속성을 검사하지만 종속성을 적용하지는 않습니다. 필수 조건인 종속 소프트웨어가 적용되지 않는 경우에는 소프트웨어가 올바르게 작동하지 않을 수도 있습니다. 전체 조건인 종속 소프트웨어가 적용되지 않는 경우에는 설치나 구성에 오류가 발생할 수 있습니다.

swconfig, swcopy, swinstall, swremove 및 swverify에 적용됩니다.

- **enforce\_dsa=true**

기본값인 **true**로 설정하면 사용 가능한 여유 공간에 비해 소프트웨어 작동에 필요한 디스크 공간이 더 많이 필요한 경우 **SD-UX**는 실행을 중단합니다. 이 옵션은 최소 여유 공간에 설치하려는 경우 또는 디스크 용량이 적대 한계에 도달해서 오류가 발생하더라도 설치, 복사 또는 패키징 작업을 시도하려는 경우 사용할 수 있습니다.

**false**로 설정하면 공간 검사는 수행되지만 디스크의 최소 여유 공간 임계값이 초과되면 시스템을 사용할 수 없을 것이라는 내용의 경고를 내보냅니다. 디스크 공간이 부족하면 설치에 실패합니다.

swcopy, swinstall 및 swpackage에 적용됩니다.

- **enforce\_kernblid\_failure = true**

커널 빌드 단계 중 어느 하나(system\_prep 및 mk\_kernel)에 발생한 오류가 설치 세션에 치명적인 영향을 미칠 것인지 여부를 제어합니다. 커널 빌드 시 발생한 오류로 인해 대화식 모드가 아닌 경우 설치 프로세스가 종료하고 대화식 모드의 경우에는 일시 중지합니다.

**false**로 설정하면 커널 빌드 프로세스에서 반환된 오류는 무시되고 설치 세션은 계속됩니다. 현재 실행 중인 커널이 그 자리에 위치하게 됩니다.

swinstall에만 적용됩니다.

- **enforce\_locatable=true**

기본값인 **true**로 설정하면 명령이 재배치할 수 없는 파일 세트를 재배치하려고 시도할 때 오류가 발생합니다. (is\_relocatable 속성을 **true**로 설정하여 재배치 가능한 파일 세트를 패키징합니다.) **false**로 설정하면 일반적인 오류 처리 프로세스가 대체되고 **SD-UX**는 그 명령이 해당 파일 세트를 재배치하도록 허용합니다.

이 옵션이 swverify에 대해 정의되어도 이 옵션과 관련된 swverify 기능은 없으므로 유의하십시오.

swinstall 및 swverify에 적용됩니다.

- **enforce\_scripts=true**

스크립트에 의해 생성된 오류 처리를 제어합니다. **true**로 설정하고 스크립트에서 오류를 반환한 경우 명령은 중지되고 실행에 실패했음을 알리는 오류 메시지가 표시됩니다. **false**로 설정하면 스크립트에서 생성된 오류는 경고로 처리되고 명령은 계속 실행을 시도합니다. 명령이 실행에 성공했음을 알리는 경고 메시지가

표시됩니다. 해당되는 경우 메시지에 오류가 발생한 스크립트 (configure/unconfigure, preinstall/postinstall, preremove/postremove 등)가 표시됩니다.

swask, swconfig, swinstall 및 swremove에 적용됩니다.

- **files=**

파일 객체를 추가하거나 삭제할 때 이 옵션으로 해당 파일의 경로 이름을 나열합니다. 기본값은 제공되지 않습니다. 추가되는 파일 객체는 해당 제품 사양 파일에도 지정할 수 있습니다.

경로 이름이 둘 이상 있는 경우 공백으로 구분하고 인용 부호 안에 넣어 표기해야 합니다.

swmodify에만 적용됩니다.

- **follow\_symlinks=false**

패키징 소스에 존재하는 심볼릭 링크를 따르지 않고 심볼릭 링크로 패키징합니다.

swpackage에만 적용됩니다.

- **include\_file\_revisions=false**

일반적으로 false로 설정하며 swpackage가 패키징되는 제품에 각 소스 파일의 버전 속성을 포함시킬 것인지를 제어합니다. 이 작업은 시간이 매우 많이 소요되므로 기본적으로는 버전 속성이 포함되지 않습니다.

이 키워드에 true 값을 설정하면 swpackage는 what 및 ident 명령을 차례로 실행하여 파일 버전을 판별하려 합니다.

swpackage에만 적용됩니다.

- **install\_cleanup\_cmd= /usr/sbin/sw/install\_clean**

릴리즈별 설치 클린업을 수행하기 위해 에이전트가 호출한 스크립트는 마지막 postinstall 스크립트가 실행된 직후에 실행됩니다. OS 업데이트의 경우 이 스크립트는 적어도 install\_setup 스크립트가 저장한 명령은 제거해야 합니다.

swagent에만 적용됩니다.

- **installed\_software\_catalog=products**

IPD(Information Products Database, 설치된 소프트웨어에 대한 설명 정보)가 저장되어 있는 디렉토리 경로를 정의합니다. 절대 경로로 설정하면 이 옵션은 IPD의 위치를 정의합니다. 이 옵션에 상대 경로를 지정하면 컨트롤러는



`admin_directory` 옵션이 지정한 경로에 경로를 추가하여 IPD에 대한 경로를 결정합니다. 대체 루트의 경우 이 경로는 대체 루트의 위치에 대한 상대 경로로 해석됩니다. (이 옵션은 소프트웨어 설치 위치에는 영향을 미치지 않으며 IPD의 위치에만 영향을 미칩니다.)

이 옵션을 통해 서로 다른 IPD를 사용하는 응용 프로그램이나 응용 프로그램 그룹을 갖는 여러 사용자나 여러 프로세스에 의한 여러 소프트웨어 응용 프로그램의 동시 설치 및 제거가 가능합니다.

주의: 특정 응용 프로그램을 관리할 때는 특정 `installed_software_catalog`를 사용하십시오. **SD-UX**는 여러 IPD의 동일 응용 프로그램에 대한 복수 설명은 지원하지 않습니다.

`admin_directory` 옵션을 참조하십시오.

`swacl`, `swask`, `swconfig`, `swinstall`, `swlist`, `swmodify`, `swremove` 및 `swverify`에 적용됩니다.

- **`install_setup_cmd=/usr/sbin/sw/install_setup`**

릴리즈별 설치 준비 작업을 수행하기 위해 에이전트가 호출하는 스크립트를 정의합니다. OS 업데이트의 경우 OS가 시스템 명령을 업데이트하는 동안 이 스크립트는 `checkinstall`, `preinstall` 및 `postinstall` 스크립트에 필요한 명령을 액세스 가능한 위치로 복사해야 합니다.

커널 파일 세트가 로드되기 전에 이 스크립트가 실행됩니다.

`swagent`에만 적용됩니다.

- **`job_title=`**

대량의 작업을 실행할 때 특정 작업을 식별하는 데 도움이 되도록 정보를 추가하고자 할 수 있습니다. 이 옵션에 값을 설정함으로써 `swjob`이나 작업 브라우저를 호출할 때 ID와 기타 작업 속성과 함께 표시되는 ASCII 문자열을 추가할 수 있습니다.

`swconfig`, `swcopy`, `swinstall`, `swremove` 및 `swverify`에 적용됩니다.

- **`kernel_build_cmd=/usr/sbin/mk_kernel`**

이는 커널 빌드를 위해 에이전트가 호출하는 스크립트입니다.

`swagent`에만 적용됩니다.

- **kernel\_path=/stand/vmunix**

시스템의 부팅 가능한 커널로의 경로. SW\_KERNEL\_PATH 환경 변수를 통해 kernel\_build\_cmd로 전달됩니다.

swagent에만 적용됩니다.

- **layout\_version=1.0**

배포 및 swlist 출력을 작성할 때 SD-UX 명령이 따라야 하는 POSIX 레이아웃 버전을 지정합니다. 1.0(기본값)과 0.8이 지원됩니다. SD-UX for HP-UX 버전 11.10 이상에서는 어떤 버전이든 읽고 쓸 수 있습니다.

SD-UX 객체와 속성 구문은 *IEEE Standard 1387.2, Software Administration (POSIX)*의 레이아웃 버전 1.0을 준수합니다. SD-UX는 이전 레이아웃 버전과 연관된 키워드 이름을 받아들여긴 하지만 이전 버전의 SD-UX가 읽을 수 있는 배포 버전을 작성할 때에는 레이아웃 버전 0.8만 사용해야 합니다.

레이아웃 버전 1.0에는 버전 0.8만을 지원하는 시스템에서는 인식할 수 없는 다음과 같은 중요한 기능이 추가되었습니다.

- 범주 클래스 객체(이전에는 번들 또는 제품 클래스 내에 있는 category 및 category\_title 속성)
- applied\_patches, is\_patch 및 patch\_state를 포함한 패치 처리 속성
- 소프트웨어가 실행될 대상 시스템의 아키텍처를 지정할 수 있도록 해주는 파일 세트 수준의 파일 세트 architecture 속성

새로운 속성과 객체를 추가함과 동시에 레이아웃 버전 1.0에서는 다음과 같이 기존의 0.8 객체와 속성이 변경됩니다.

- 미디어 객체의 저장소 media\_sequence\_number 속성을 일련 번호 속성으로 대체합니다.
- 제품 및 번들 내 공급업체 정의를 vendor\_tag 속성 및 제품이나 번들 외부에서 정의된 해당 공급업체 객체로 대체합니다.
- corequisite 및 prerequisite 파일 세트 속성이 증가합니다.
- timestamp 속성을 mod\_time으로 변경합니다.

swcopy, swlist, swmodify 및 swpackage에 적용됩니다.

- **level=**

swacl, swlist 및 swreg에 대한 소프트웨어 수준을 지정합니다.

swlist의 경우 이 옵션은 지정된 수준까지 모든 객체를 나열합니다. 지정된 수준 및 지정된 *software\_selections* 깊이가 swlist 출력의 깊이를 제어합니다. 지원되는 소프트웨어 수준은 다음과 같습니다.

- **bundle** -- 번들 수준까지 모든 객체를 표시합니다.
- **product** -- 제품 수준까지 모든 객체를 표시합니다. 또한 **-l bundle -l product**를 사용하여 번들을 표시할 수도 있습니다.
- **subproduct** -- 하위 제품 수준까지 모든 객체를 표시합니다.
- **fileset** -- 파일 세트 수준까지 모든 객체를 표시합니다. 또한 **-l fileset -l subproduct**를 사용하여 하위 제품을 표시할 수도 있습니다.
- **file** -- 파일 수준까지 모든 객체(저장소, 제품, 파일 세트 및 파일)를 표시합니다.
- **control\_file** -- 제어 파일 수준까지 모든 객체를 표시합니다.
- **category** -- 사용 가능한 소프트웨어 객체의 범주를 모두 표시합니다.
- **patch** -- 적용된 패치를 모두 표시합니다. (*show\_superseded\_patches* 옵션도 참조하십시오.)

지원되는 저장소 및 루트 수준은 다음과 같습니다.

- **depot** -- 저장소 수준(지정된 대상 호스트에 존재하는 저장소)만을 표시합니다.
- **root** -- 대체 루트를 모두 표시합니다.
- **shroot** -- 등록된 공유 루트를 모두 표시합니다(HP-UX 10.X에 한함).
- **prroot** -- 개별 루트를 모두 표시합니다(HP-UX 10.X에 한함).

swacl의 경우 이 옵션은 보거나 변경할 ACL의 수준을 지정합니다.

- **host** -- *target\_selections*로 식별되는 호스트 시스템을 보호하며 ACL을 보거나 변경합니다.
- **depot** -- *target\_selections*로 식별되는 소프트웨어 저장소를 보호하며 ACL을 보거나 변경합니다.
- **root** -- *target\_selections*로 식별되는 루트 파일 시스템을 보호하며 ACL을 보거나 변경합니다.

- `product -- software_selection`으로 식별되는 소프트웨어 제품을 보호하며 **ACL**을 보거나 변경합니다. 저장소의 제품에만 적용되고 루트에 설치된 제품에는 적용되지 않습니다.
- `product_template -- target_selections`로 식별되는 소프트웨어 저장소에 추가된 차후 제품의 **ACL**을 초기화하는 데 사용되는 템플릿 **ACL**을 보거나 변경합니다.
- `global_soc_template -- target_selections`로 식별되는 호스트에 추가된 차후 소프트웨어 저장소나 루트 파일 시스템의 **ACL**을 초기화하는 데 사용되는 템플릿 **ACL**을 보거나 변경합니다.
- `global_product_template -- target_selections`로 식별되는 호스트에 추가된 차후 소프트웨어 저장소의 `product_template` **ACL**을 초기화하는 데 사용되는 템플릿 **ACL**을 보거나 변경합니다.

`swreg`의 경우 이 옵션은 등록 또는 등록 해제할 객체의 수준을 정의합니다.

- `depot` -- 지정된 대상 호스트에 존재하는 저장소입니다.
  - `root` -- 모든 대체 루트입니다.
  - `shroot` -- 등록된 모든 공유 루트(**HP-UX 10.X**에 한함)입니다.
  - `prroot` -- 등록된 모든 개별 루트(**HP-UX 10.X**에 한함)입니다.
- `swacl`, `swlist` 및 `swreg`에 적용됩니다.

- **log\_msgid=0**

다음과 같은 숫자 식별 번호를 **SD-UX** 로그 파일 메시지 처음에 추가합니다.

- 0(기본값)은 메시지에 식별자를 추가하지 않습니다.
- 1은 **ERROR** 메시지에만 식별자를 추가합니다.
- 2는 **ERROR** 및 **WARNING** 메시지에 식별자를 추가합니다.
- 3은 **ERROR**, **WARNING** 및 **NOTE** 메시지에 식별자를 추가합니다.
- 4는 **ERROR**, **WARNING**, **NOTE** 및 기타 특정 정보 메시지에 식별자를 추가합니다.

`swconfig`, `swcopy`, `swinstall`, `swmodify`, `swpackage`, `swreg`, `swremove` 및 `swverify`에 적용됩니다.

- **logdetail=false**

로그 파일에 기록되는 정보의 상세도를 제어합니다. 이 옵션을 **true**로 설정하면 지정된 옵션, 진행 상태 및 추가 요약 정보와 같은 자세한 작업 정보를 로그 파일에 추가합니다.

표 A-1에서는 `loglevel`과 `logdetail` 옵션의 가능한 조합을 보여줍니다.

표 A-2

**loglevel과 logdetail 조합**

로그 수준	로그 상세도	포함되는 정보
<code>loglevel=0</code>	(적용되지 않음)	로그 파일에 정보가 기록되지 않습니다.
<code>loglevel=1</code>	<code>logdetail=false</code>	주요 이벤트만 기록됩니다. 두 옵션의 기본 설정입니다.
<code>loglevel=1</code>	<code>logdetail=true</code>	위의 자세한 이벤트 내용과 작업 진행 메시지(1이 기본값이므로 <code>loglevel=1</code> 을 설정하는 것은 생략할 수 있습니다.)
<code>loglevel=2</code>	<code>logdetail=false</code>	이벤트 및 파일 수준의 메시지만 기록됨 ( <code>false</code> 가 기본값이므로 <code>logdetail=false</code> 를 설정하는 것은 생략할 수 있습니다.)
<code>loglevel=2</code> <sup>a</sup>	<code>logdetail=true</code>	모든 정보가 기록됩니다.

a. 이 조합은 HP-UX 10.x 릴리즈에서와 로그 파일에 기록됩니다.

`loglevel=2`와 `logdetail=true` 옵션을 모두 설정해야 합니다.

`swconfig`, `swcopy`, `swinstall`, `swreg`, `swremove` 및 `swverify`에 적용됩니다.

- **logfile=/var/adm/sw/<command>.log**

각 명령에 대한 기본 컨트롤러 로그 파일입니다. 에이전트 로그 파일은 /var/spool/sw/swagent.log 및 /var/adm/sw/swagent.log와 같이 항상 대상 저장소 또는 대상 루트의 상대적인 위치에 있습니다.

swacl, swlist 및 swjob을 제외한 모든 명령에 적용됩니다.

- **loglevel=1**

이 옵션은 SD-UX 로그 파일 메시지 앞에 식별 번호를 붙여서 명령 로그 파일, 대상 에이전트 로그 파일 및 소스 에이전트 로그 파일에 기록되는 이벤트의 로그 수준을 제어합니다. 이 정보는 logdetail 옵션으로 상세도를 제어할 수도 있습니다. 값은 다음과 같습니다.

— 0 -- 로그 파일에 정보를 제공하지 않습니다.

— 1 -- 로그 파일에 상세하게 기록합니다.

— 2 -- 로그 파일에 매우 상세하게 기록합니다.

swconfig, swcopy, swinstall, swmodify, swpackage, swremove 및 swverify에 적용됩니다.

- **match\_target=false**

true로 설정하면 대상 시스템에 이미 설치되어 있는 파일 세트와 일치하는 소스에서 파일 세트를 강제로 선택합니다.

설치된 파일 세트를 “상위”로 지정한 소스상의 파일 세트가 선택됩니다.

이 옵션은 다른 소프트웨어 선택 사항을 대체합니다.

모호하게 선택할 수 없습니다.

swinstall에만 적용됩니다.

- **max\_agents=-1**

동시에 실행할 수 있도록 허용되는 에이전트의 최대 수. -1 값은 제한이 없음을 나타냅니다.

swagentd에만 적용됩니다.

- **max\_targets=25**

이 옵션을 양의 정수로 설정하면 동시에 설치하거나 복사할 수 있는 횟수를 지정된 수로 제한합니다. 하나의 복사 또는 설치 작업이 완료되면 모든 대상이 완료될 때까지 다른 대상이 선택되어 시작됩니다.

서버 및 네트워크 성능을 고려하여 적절한 설정을 결정하는데 권장되는 시작점은 25(기본값)입니다. 이 옵션을 1 미만의 값으로 설정하면 SD-UX는 모든 대상을 한 번에 설치하거나 복사하려 합니다.

swcopy 및 swinstall에 적용됩니다.

- **media\_capacity=1330**

배포 테이프나 CD-ROM과 같은 여러 디렉토리 미디어를 작성할 때 이 키워드는 백만 바이트 단위로(MB 아님) 테이프의 용량을 지정합니다. 이 옵션은 미디어가 DDS 테이프나 디스크 파일이 아닌 경우 필요합니다. 이 옵션을 지정하지 않으면 swpackage는 테이프의 경우 기본값인 13억 바이트를 설정하고 디스크 파일의 경우 디스크상의 여유 공간 용량을 최대 minfree 값까지 설정합니다.

SD-UX는 파일 세트의 파티션 및 media\_sequence\_number 속성의 설정에 따라 올바른 크기를 계산하는 등 여러 시리얼 미디어에 대해서와 마찬가지로 여러 디렉토리 미디어에 대해서도 동일한 형식을 사용합니다.

swpackage에만 적용됩니다.

- **media\_type=directory**

작성할 배포 유형을 정의합니다. 인식할 수 있는 유형은 directory와 tape입니다. 이 옵션을 지정하지 않으면 swpackage는 기본값으로 배포 디렉토리(저장소)를 만듭니다.

swpackage에만 적용됩니다.

- **minimum\_job\_polling\_interval=1**

시작해야 할 예약된 작업이 있는지 또는 원격 대상 상태가 로컬로 캐싱되어야 하는 활성화된 작업이 있는지 여부를 판별하기 위해 데몬을 “활성화”시켜 작업 대기열을 점검할 빈도를 분 단위로 정의합니다.

0으로 설정하면 예약된 작업이 시작되지 않으며 활성화된 작업에 대한 캐싱도 발생하지 않습니다.

swagentd에만 적용됩니다.

- **mount\_all\_filesystems=true**

일반적으로 **true**로 설정하며, 명령은 분석 단계가 시작될 때 파일 시스템 표 (/etc/fstab)에 있는 파일 시스템을 모두 자동으로 마운트하려고 시도하고 계속 진행하기 전에 모든 파일 시스템이 마운트되었는지 확인합니다.

**false**로 설정하면 추가 파일 시스템이 마운트되지 않습니다.

swconfig, swcopy, swinstall, swremove, swverify에 적용됩니다.

- **mount\_cmd=/sbin/mount**

모든 파일 시스템을 마운트하기 위해 에이전트가 호출하는 명령을 지정합니다.

swagent에만 적용됩니다.

- **objects\_to\_register=**

등록하거나 등록을 해제할 기본 객체를 정의합니다. 객체가 둘 이상 있는 경우 공백으로 분리해야 합니다.

기본값은 제공되지 않습니다. select\_local도 참조하십시오.

swreg에만 적용됩니다.

- **one\_liner=<attributes>**

비상세 목록에 나열되는 속성을 정의합니다.

속성이 둘 이상 있는 경우 공백으로 구분하고 인용 부호 안에 넣어야 합니다.

**one\_liner="revision size title"**

기본 소프트웨어 목록이 사용해야 할 속성(즉, 버전, 크기, 제목 등)을 선택해야 합니다. 참고: tag 속성은 항상 번들, 제품, 하위 제품 및 파일 세트에 대해 표시되고 path는 항상 파일에 대해 표시됩니다.

어떤 속성이든 선택할 수 있지만 특정 속성은 모든 적용 가능한 소프트웨어 클래스(번들, 제품, 하위 제품, 파일 세트)에 대해 존재하지 않을 수도 있습니다. 예를 들어 title이라는 소프트웨어 속성은 번들, 제품, 하위 제품 및 파일 세트에 대해 사용할 수 있지만 architecture 속성은 제품과 파일 세트에 대해서만 사용할 수 있습니다.

-v나 -a 옵션을 지정하지 않으면 swlist는 각 소프트웨어 객체(번들, 제품, 하위 제품 및 파일 세트)에 대한 one\_liner 정보를 표시합니다.

swlist에만 적용됩니다.



- **os\_name**

HP-UX 업데이트를 위한 파일 세트 선택을 지정합니다. (이 옵션은 항상 `os_release` 옵션과 함께 사용해야 합니다.) 이 옵션은 명령줄에서 지정하거나 `swinstall GUI`를 호출할 때 지정해야 합니다.

이 옵션의 구문은 다음과 같습니다.

**os\_name=operating\_system:width**

- `operating_system`은 HP-UX와 같은 운영 체제 이름을 지정합니다. 자세한 내용은 `uname(1)` 매뉴얼 페이지를 참조하십시오.
  - `width`는 설치될 OS의 비트 단위로(32비트 또는 64비트 중 하나) 단어의 길이를 지정합니다.
  - `operating_system`과 `width`는 콜론(:)으로 구분해야 합니다.
- `swinstall`에만 적용됩니다.

- **os\_release**

HP-UX 업데이트를 위한 파일 세트 선택을 지정합니다. (이 옵션은 항상 `os_name` 옵션과 함께 사용해야 합니다.) 이 옵션은 명령줄에서 지정하거나 `swinstall GUI`를 호출할 때 지정해야 합니다.

이 옵션의 구문은 다음과 같습니다.

**os\_release=release**

`release`는 HP-UX 릴리즈를 지정합니다. 그 값은 다음과 같습니다.

B.10.01  
B.10.10  
B.10.20  
B.10.30  
B.11.00

`swinstall`에만 적용됩니다.

- **package\_in\_place=no**

이 옵션을 `yes`로 설정하면 `swpackage`가 제품을 빌드하고 저장소에는 제품을 구성하는 파일이 실제로 들어 가지 않습니다. 대신 저장소는 제품 빌드에 사용된 소스 파일을 참조합니다. 이를 통해 배포 저장소를 만드는 데 필요한 디스크 공간을 추가로 사용하지 않고 개발 또는 테스트 환경에서 제품을 패키징할 수 있습니다.

`swpackage`에만 적용됩니다.

- **patch\_commit=false**

패치 롤백을 위해 저장된 파일을 제거하여 패치를 확약합니다. 기본값은 `false`입니다. `true`로 설정하면 이 옵션은 해당 명령에 대한 소프트웨어 선택에 지정된 패치에 대해 저장된 파일을 제거합니다. 일단 이 옵션을 패치에 실행하면 해당 패치가 수정한 연관된 기본 소프트웨어를 제거하지 않는 한 패치를 제거할 수 없습니다. `swmodify`에만 적용됩니다.

- **patch\_filter=\*. \***

패치 필터에 대한 `software_specification`을 지정합니다.

이 옵션은 `autoselect_patches` 및 `patch_match_target` 옵션과 함께 사용하여 `software_specification`에 의해 지정된 기준을 충족시키기 위해 선택된 패치를 필터링하는 데 사용할 수 있습니다. 기본 `software_specification` 값은 `*. *`입니다.

명령을 사용하여 소프트웨어를 지정하거나 `\*` 와일드카드를 사용하여 소프트웨어를 선택하는 경우 패치 필터링이 바뀔 수 있으므로 유의하십시오.

`swask`, `swcopy` 및 `swinstall`에 적용됩니다.

- **patch\_match\_target=false**

`true`로 설정하면 이 옵션은 대상 루트나 저장소의 소프트웨어와 일치하는 최신 패치(`is_patch` 속성을 `true`로 설정하여 패키징한 소프트웨어)를 선택합니다.

`patch_filter=` 옵션은 `patch_match_target` 옵션과 함께 사용할 수 있습니다.

`swcopy` 및 `swinstall`에 적용됩니다.

- **patch\_one\_liner=title patch\_state**

이 명령을 사용하여 `-l` 패치 옵션을 호출할 때와 `-a` 또는 `-v` 옵션을 지정하지 않을 때 나열되는 각 객체에 대해 표시되는 속성을 지정합니다. 기본으로 표시되는 속성은 `title`과 `patch_state`입니다.

`swlist` 및 `swjob`에 적용됩니다.

- **patch\_save\_files=true**

패치된 파일을 저장하여 차후 패치 롤백이 가능하게 합니다. **false**로 설정하면 패치에 의해 수정된 기본 소프트웨어가 동시에 제거되지 않는 한 패치는 롤백(제거)할 수 없습니다.

swinstall에 적용됩니다.

- **polling\_interval=2**

대화식 세션에만 적용됩니다. 분석 및 실행 단계에서 SD-UX가 상태 정보를 위해 각 대상을 폴링하는 빈도를 초 단위로 지정합니다. WAN상에서 작동해야 하는 경우 네트워크 오버헤드를 줄이기 위해 폴링 간격을 증가시킬 수 있습니다. 이 옵션에 대해 높은 숫자를 지정하면 폴링 간격이 더 길어집니다.

swcopy, swinstall 및 swremove에 적용됩니다.

- **preserve\_create\_time=false**

저장소를 복사할 때 원래의 작성 시간을 보관하여 사본을 사용할 때 일관성 있는 결과를 산출합니다. 기본값인 **false**로 설정하면 번들, 제품 및 파일 세트의 **create\_time**이 **swcopy**가 새로운 저장소를 작성한 시간과 동일하게 설정됩니다. **true**로 설정하면 **create\_time**은 현재의 선택 사항을 복사해온 소스 저장소에 지정된 시간으로 설정됩니다. 마스터 저장소로 복사할 때 이 옵션을 사용하면 **create\_time\_filter** 옵션을 사용할 때 보이는 객체를 변경할 수 있습니다.

swcopy에 적용됩니다.

- **preview=false**

**true**로 설정하면 이 명령을 미리 보기 모드에서만 실행합니다(분석 단계를 끝내고 종료합니다). 이 옵션은 명령줄에서 **-p**를 지정하는 것과 동일한 효과를 갖습니다.

swcopy, swinstall, swremove 및 swconfig에 적용됩니다.

- **reboot\_cmd=/sbin/reboot**

이것은 시스템을 재부팅하기 위해 에이전트가 호출하는 명령입니다.

swagent에 적용됩니다.

- **reconfigure=false**

이 옵션은 이미 구성 상태에 있는 소프트웨어가 다시 구성되지 않도록 방지합니다. **true**로 설정하면 구성된 소프트웨어를 다시 구성할 수 있습니다.

swconfig에 적용됩니다.

- **register\_new\_depot=true**

일반적으로 **true**로 설정하며 새로 작성된 저장소는 호스트상에서 등록됩니다. 이를 통해 다른 명령이 자동으로 이 저장소를 볼 수 있습니다.

**false**로 설정하면 새로운 저장소는 등록되지 않습니다. 이 옵션을 통해 테스트하고 차후 swreg로 등록할 개별 저장소를 만들 수 있습니다.

swcopy에만 적용됩니다.

- **register\_new\_root=true**

swinstall이 로컬 swagentd를 사용하여 새로 작성된 대체 루트를 등록할 수 있도록 하며, 이를 통해 다른 SD-UX 명령이 이 루트를 볼 수 있습니다.

**false**로 설정하면 새로운 루트는 자동으로 등록되지 않습니다. (swreg 명령을 사용하여 나중에 저장소를 등록할 수 있습니다.)

swinstall에만 적용됩니다.

- **reinstall=false**

SD-UX가 기존 버전의 파일 세트를 다시 설치(덮어쓰기)하지 않도록 방지합니다. **true**로 설정하면 파일 세트는 다시 설치됩니다.

swcopy 및 swinstall에 적용됩니다.

- **reinstall\_files=false**

파일 덮어쓰기를 제어하며 이는 속도가 느린 네트워크나 디스크에서의 성능을 향상시킬 수 있습니다. 기본값인 **false**로 설정하면 SD-UX는 소스 파일 세트의 각 파일을 대상 시스템의 대응하는 파일과 비교합니다. SD-UX는 크기, 타임스탬프 및 체크섬(생략 가능) 등을 기준으로 파일을 비교합니다. 파일이 대상 시스템의 파일과 동일한 경우 덮어쓰지 않습니다.

**true**로 설정하면 SD-UX는 파일을 비교하지 않고 대상의 동일한 파일을 덮어씁니다.

reinstall 및 reinstall\_files\_use\_cksum 옵션도 참조하십시오.

swcopy, swinstall 및 swpackage에 적용됩니다.

- **reinstall\_files\_use\_cksum=true**

**reinstall\_files\_use\_cksum=false**(swpackage에만 해당)

reinstall\_files 옵션을 **false**로 설정하면 체크섬 비교의 사용을 제어합니다. 기본값인 **true**로 설정하면 SD-UX가 체크섬을 계산 및 비교하여 새 파일이 이전 파일을 덮어써야 하는지 결정합니다. 체크섬을 사용하면 비교 속도가 늦어지지만 크기나 타임스탬프에 비해 보다 확실한 검사를 수행하는 것입니다.

**false**로 설정하는 경우 SD-UX는 체크섬을 계산하지 않고 크기와 타임스탬프만으로 파일을 비교합니다.

swcopy, swinstall 및 swpackage에 적용됩니다.

- **remove\_empty\_depot=true**

저장소의 마지막 제품이나 번들이 제거되면 저장소 자체가 제거됩니다.

**false**로 설정하면 마지막 제품(또는 번들)이 제거될 때 저장소는 제거되지 않습니다. 이 때 해당 저장소의 ACL이 보존됩니다.

swremove에만 적용됩니다.

- **remove\_obsolete\_filesets=false**

이 명령은 swcopy가 대상 저장소의 대상 제품에서 오래된 파일 세트를 자동으로 제거할지 여부를 제어합니다. **true**로 설정하면 swcopy는 복사 프로세스를 실행하는 동안 대상 제품에서 오래된 파일 세트를 제거합니다. 제거는 복사가 완료된 후에 수행됩니다. 파일 세트는 소스 저장소에 있는 제품의 가장 최신 패키지징에 포함되어 있지 않으면 오래된 것으로 정의됩니다.

swcopy에만 적용됩니다.

- **remove\_setup\_cmd=/usr/sbin/sw/remove\_setup**

릴리즈별 제거 준비 작업을 수행하기 위해 에이전트가 호출하는 스크립트를 정의합니다. OS 업데이트를 위해 이 스크립트는 파일 세트가 제거될 때 tlink 명령을 호출합니다.

swagentd에만 적용됩니다.

- **retry\_rpc=1**

이 명령은 파일 전송 중에 끊어진 소스 연결이 재시도되는 횟수를 정의합니다. 1에서 9까지의 숫자로 설정하면 각 파일 세트 설치가 해당 숫자 횟수만큼 시도됨을 의미합니다. `reinstall_files` 옵션은 파일 세트 내에서 정상적으로 설치된 파일을 설치하지 못하도록 `false`로 설정해야 합니다.

이 옵션은 에이전트에 연결되는 컨트롤러에도 적용됩니다. 어떤 이유에서건 에이전트 세션이 정상적으로 시작되지 못한 경우 컨트롤러는 에이전트로의 연결을 시도한 후 다음 시도까지 얼마나 기다려야 하는지를 판별하기 위해 `retry_rpc_interval` 옵션의 값을 사용하여 `retry_rpc`에 지정된 횟수만큼 에이전트에 다시 연결하려고 시도합니다.

`swcopy` 및 `swinstall`에 적용됩니다.

- **retry\_rpc\_interval={0}**

최초의 시도가 실패한 후 대상으로의 연결 시도를 반복할 때 그 간격의 길이를 분단위로 지정합니다. `retry_rpc` 옵션과 함께 사용됩니다.

이 옵션 값 숫자가 `retry_rpc` 값과 동일한 경우 **SD-UX**는 `retry_rpc`에 지정된 횟수만큼 소스 연결을 재설정하려고 시도합니다. `retry_rpc_interval` 값의 숫자가 `retry_rpc` 값보다 작은 경우 **SD-UX**는 재시도 횟수가 `retry_rpc`와 같아질 때까지 마지막 간격 값을 반복합니다.

예를 들어 에이전트 세션이 정상적으로 시작하지 못하고 `retry_rpc`가 9로 설정되었으며 `retry_rpc_interval`을 {1 2 4 8 15}로 설정하여 일시적인 네트워크 장애를 해결하기에 충분히 긴 시간을 확보한 경우, 컨트롤러는 첫 번째 시도 후 1분 후에 에이전트에 재연결을 시도하고, 두 번째 시도 후 2분, 세 번째 시도 후 4분 그리고 8분 그리고 15분을 모두 9번의 재시도가 이루어질 때까지 계속 추가 재시도를 수행합니다. 이 값이 지정된 경우 파일 로드 장애가 발생하면 작동이 90분 (1+2+4+8+15+15+15+15+15) 동안 정지할 수 있습니다. 두 가지 옵션이 5로 설정된 경우 컨트롤러는 대상을 30분의 간격을 두고 다섯 번 대상에 연결하려고 시도합니다.

`swcopy` 및 `swinstall`에 적용됩니다.

- **reuse\_short\_job\_numbers=true**

작업 ID 번호를 지정할 때 SD-UX는 10,000보다 작은 숫자를 사용합니다. 일반적으로 오래된 작업은 작업 번호 9,999에 도달하기 훨씬 전에 제거되므로 작업 번호는 9999에서 1로 빠르게 돌아옵니다. 작업 ID 번호가 9,999에 도달하기 전에 제거되지 않은 많은 수의 작업을 실행할 때 SD-UX가 사용하지 않은 작업 번호를 검색하는 동안 성능이 저하될 수 있습니다.

reuse\_short\_job\_numbers를 false로 설정하면 SD-UX는 10,000 이상의 숫자를 사용하기 시작합니다. 이를 통해 검색 시 지연될 수 있는 가능성을 방지하고 작업 ID 번호가 필요에 따라 8자리(99,999,999)로 늘어날 수 있습니다. (이는 9999에서 다시 1로 돌아오는 것을 방지하므로 일반적으로 바람직하지 않습니다.)

autoremove\_job 옵션도 참조하십시오.

swconfig, swcopy, swinstall, swremove 및 swverify에 적용됩니다.

- **rpc\_binding\_info=ncacn\_ip\_tcp:[2121] ncadg\_ip\_udp:[2121]**

swagentd 데몬이 청취하는 프로토콜 순서와 엔드포인트를 결정합니다. 한 프로토콜 순서에 대한 연결이 실패하면 다음 프로토콜 순서에 대해 시도합니다.

SD-UX는 대부분의 플랫폼에서 tcp와 udp 프로토콜을 모두 지원합니다.

값은 다음과 같은 형식을 갖습니다.

- 프로토콜 순서와 엔드포인트(포트 번호)가 포함된 DCE 문자열 바인딩. 이 구문은 다음과 같습니다.

**protocol\_sequence: [ endpoint ]**

- 엔드포인트를 지정하지 않은 DCE 프로토콜 순서의 이름. 이 구문은 다음과 같습니다.

**ncadg\_ip\_udp** 또는 **ncacn\_ip\_tcp**

엔드포인트를 지정하지 않았기 때문에 DCE 엔드포인트 매핑인 *rpcd*가 실행되어 swagentd에 의해 등록된 엔드포인트를 찾는 데 사용되어야 합니다.

- 리터럴 문자열 *a11*. 이 항목은 DCE RPC(Remote Procedure Call) 런타임에 의해 지원되는 모든 프로토콜 순서를 사용함을 의미합니다. 이것은 목록에서 유일한 항목이어야 합니다. *rpcd*가 실행되고 있어야 합니다.

swask, swmodify 및 swpackage를 제외한 모든 명령에 적용됩니다.

- **rpc\_binding\_info\_alt\_source=ncadg\_ip\_udp:[2121]**

에이전트가 alternate\_source 옵션에 의해 지정된 대체 소스 저장소에 연결하려고 시도할 때 사용되는 프로토콜 순서와 엔드포인트를 정의합니다. SD-UX 는 udp(ncadg\_ip\_udp:[2121]) 및 tcp(ncacn\_ip\_tcp:[2121]) 프로토콜 순서/엔드포인트를 모두 지원합니다.

swagent에 적용됩니다.

- **rpc\_timeout=5**

통신 제한 시간의 상대적 길이. 이 값은 0에서 9 사이의 값으로 DCE RPC에 의해 해석됩니다. 값이 높으면 더 긴 시간을 나타내며 속도가 느리거나 처리량이 많은 네트워크에서는 더 높은 값을 지정해야 합니다. 낮은 값은 가동되지 않고 있거나 swagentd를 실행하고 있지 않은 호스트에 연결하려는 시도를 더 신속하게 감지합니다.

그보다 낮은 값의 약 두 배입니다. ncadg\_ip\_udp 프로토콜 순서의 경우 5 값은 약 30초입니다. ncacn\_ip\_tcp 프로토콜 순서를 사용할 때 이 옵션은 무시될 수 있습니다.

swmodify 및 swpackage를 제외한 모든 명령에 적용됩니다.

- **run\_as\_superuser=true**

이 옵션은 SD-UX의 비권한 모드를 제어합니다. 호출한 사용자가 슈퍼유저인 경우 이 옵션은 무시됩니다(true로 처리).

기본값인 true로 설정하면 작업 권한이 로컬 슈퍼유저에게 부여되거나 ACL에 의해 설정되어 SD-UX 기능이 정상적으로 수행됩니다. (ACL에 대한 자세한 내용은 9장, 233페이지의 “SD-UX 보안”을 참조하십시오.)

false로 설정하고 호출한 사용자가 로컬 사용자이고 슈퍼유저가 아닌 경우 비권한 모드가 호출됩니다.

— 작업 권한은 사용자의 파일 시스템 권한에 근거합니다.

— ACL은 무시됩니다.

— SD-UX에 의해 작성된 파일은 호출한 사용자의 uid와 gid를 갖게 되고 작성된 파일의 모드는 호출한 사용자의 umask에 따라 설정됩니다.



비권한 모드는 특별하게 설계되고 패키징된 응용 프로그램 관리를 목적으로 하는 경우에 한해 사용됩니다. 이 모드는 HP-UX 운영 체제나 그에 대한 패치를 관리하는 데에 사용할 수 없습니다. 이 옵션은 원격 작업에는 사용할 수 없습니다. 이 옵션을 **true**로 설정하면 `autoremove_job` 옵션이 **true**로 강제 설정됩니다. 비권한 SD-UX에 대한 자세한 설명은 12장, 391페이지의 “비권한 SD”를 참조하십시오.

`admin_directory` 옵션도 참조하십시오.

`swagent`, `swagentd` 및 `install-sd`를 제외한 모든 명령에 적용됩니다.

- **select\_local=true**

일반적으로 **true**로 설정하며 로컬 호스트의 기본 저장소나 설치 디렉토리를 명령의 대상으로 선택합니다.

`swacl`, `swconfig`, `swcopy`, `swinstall`, `swlist`, `swreg`, `swremove`, `swverify`에 적용됩니다.

- **show\_superseded\_patches=false**

**false**로 설정하면 `swlist`는 대체된 패치를 표시하지 않습니다. 대체된 패치를 보려면 이 옵션을 **true**로 설정해야 합니다. 대체된 패치를 `swlist`로 명시적으로 표시해도 이 옵션이 **true**가 아니면 표시되지 않습니다.

`swlist`에만 적용됩니다.

- **software=**

기본 `software_selections`를 정의합니다.

기본값은 제공되지 않습니다. `software_selection`이 둘 이상 있는 경우 중괄호 `{}`나 인용 부호 안에 넣어야 합니다. 소프트웨어는 일반적으로 명령줄이나 GUI 또는 TUI에서의 옵션처럼 `software_selections` 입력 파일에 지정됩니다.

`swreg`를 제외한 모든 명령에 적용됩니다.

- **software\_view=products**

GUI에서 사용할 소프트웨어 보기를 나타냅니다. `products`, `all_bundles` 또는 `bundle category`(해당 범주의 번들만 표시) 태그로 설정할 수 있습니다. 기본 보기는 `all_bundles`와 번들에 포함되지 않는 제품을 표시합니다.

`swcopy`, `swinstall`, `swlist` 및 `swremove`에 적용됩니다.

- **source=**

자동으로 GUI 및 CLI 소스 선택 대화 상자를 통과할 소스를 지정합니다. 이 옵션은 `-s source` 명령줄 옵션과 동일한 효과를 갖습니다. 다음과 같은 구문을 사용하여 소스를 지정하십시오.

**[path]**

swcopy 및 swinstall에 적용됩니다.

- **source\_cdrom=/SD\_CDROM**

소스 CD-ROM의 기본 위치를 정의합니다. 구문은 다음과 같습니다.

**[host] [:] [path]**

swinstall에만 적용됩니다.

- **source\_depot\_audit=true**

소스 및 대상 시스템이 모두 SD-UX 버전 B.11.00 이상으로 업데이트되는 경우 소스 저장소 시스템의 시스템 관리자는 이 옵션을 설정하여 어느 사용자가 소스 시스템 저장소에서 어떤 소프트웨어를 가져오는지 및 언제 소프트웨어를 가져오는지를 추적합니다.

대상 시스템에서 swinstall이나 swcopy를 실행하는 사용자는 이 옵션을 설정할 수 없으며 소스 저장소 시스템의 관리자만 설정할 수 있습니다.

source\_depot\_audit를 기본값인 true로 설정하면 swaudit.log 파일이 소스 저장소(쓰기 가능한 디렉토리 저장소의 경우)나 /var/tmp(tar 이미지, CD-ROM 또는 기타 쓰기 불가능한 저장소의 경우)에 작성됩니다.

감사 정보를 보거나 인쇄하거나 저장하려면 다음과 같이 입력하여 swlist 대화식 사용자 인터페이스를 실행합니다.

**swlist -i -d**

시스템에 해당 SD-UX 메시지 카탈로그 파일을 가지고 있는 경우 언어 기본 설정에 기초하여 감사 정보를 볼 수 있습니다. 예를 들어 swlist를 한 번 실행시킬 때 일본어로 소스 감사 정보를 보고, 다음에 실행시킬 때 영어로 동일한 정보를 볼 수 있습니다.

swagent에 적용됩니다.

- **source\_file=psf**

이 키워드는 입력으로 패키징 또는 swmodify 세션으로 읽어들이 기본 *product\_specification\_file*을 정의합니다. 상대 경로 또는 절대 경로일 수 있습니다.

swpackage 및 swmodify에 적용됩니다.

- **source\_tape=/dev/rmt/0m**

일반적으로 로컬 테이프 장치의 문자 특수 파일인 기본 테이프 위치를 정의합니다. **host:path** 구문을 사용할 수도 있지만 호스트는 로컬 호스트와도 일치해야 합니다. -s 옵션은 이 값을 바꿉니다.

swcopy 및 swinstall에 적용됩니다.

- **source\_type=directory**

다음 세 가지 옵션 중 하나를 가리키는 기본 소스 유형(cdrom, file, directory 또는 tape)입니다. -s *source file* 옵션에서 파생된 소스 유형이 이 기본값을 바꿉니다.

cdrom과 tape 값은 swcopy와 swinstall에 적용됩니다. file 값은 swpackage에만 적용됩니다.

- **system\_file\_path=/stand/system**

커널의 템플릿 파일에 대한 경로. 경로가 SW\_SYSTEM\_FILE\_PATH 환경 변수를 통해 system\_prep\_command에 전달됩니다.

swagent에만 적용됩니다.

- **system\_prep\_cmd=/usr/sbin/sysadm/system\_prep**

에이전트가 호출하는 커널 빌드 준비 스크립트. 이 스크립트는 모든 필요한 준비 작업을 수행하여 제어 스크립트가 빌드될 커널을 올바르게 구성할 수 있도록 합니다.

swagent에만 적용됩니다.

- **targets=**

기본값은 제공되지 않습니다(select\_local 참조). 대상이 둘 이상 있는 경우 공백으로 분리해야 합니다. 대상은 일반적으로 명령줄이나 GUI에서의 옵션처럼 대상 입력 파일에 지정됩니다.

모든 명령에 적용됩니다.

- **target\_type**

media\_type을 참조하십시오.

- **uncompress\_cmd=**

이것은 설치, 복사 또는 패키징할 때 파일 압축을 풀기 위해 소스 에이전트가 호출하는 명령입니다.

이 명령은 압축된 형식으로 미디어에 저장되었던 파일을 처리합니다. 파일의 compression\_type이 gzip인 경우 외부 압축 해제 명령 대신 내부 압축 (funzip)이 사용됩니다.

swagent 및 swpackage에 적용됩니다.

- **uncompress\_files=false**

true로 설정하면 파일은 대상 저장소에 저장되기 전에 현재의 uncompress\_cmd를 사용하여 압축이 풀립니다.

swpackage 세션 중에 uncompress\_files와 compress\_files 옵션 중 하나만 true로 설정할 수 있습니다.

package\_in\_place가 true이거나 media\_type이 tape인 경우 uncompress\_files 옵션을 true로 설정하지 못할 수도 있습니다.

swcopy 및 swpackage에 적용됩니다.

- **use\_alternate\_source=false**

기본값 false를 설정하면 swinstall이나 swcopy는 소프트웨어 소스로 사용할 로컬 호스트에 대한 소스 바인딩 및 저장소 경로를 설명하는 정보가 포함된 요청을 통해 분석 또는 작업을 시작합니다.

true인 경우 로컬 호스트는 자체 구성된 값을 사용합니다. 로컬 호스트에서 alternate\_source에 대한 에이전트의 구성된 값이 host:/path 형식으로 지정됩니다. 이 값에 경로 구성 요소(예: alternate\_source=:/path)만 포함되는 경우 에이전트는 이 경로를 자신의 로컬 호스트의 파일 시스템에 적용합니다.

호스트 구성 요소만 존재하는 경우(예: alternate\_source=host) 에이전트는 컨트롤러가 제공하는 경로를 이 호스트에 적용합니다. alternate\_source에 대해 구성된 값이 없는 경우 에이전트는 컨트롤러가 제공한 경로를 자신의 로컬 호스트에 적용합니다.

swcopy 및 swinstall에 적용됩니다.

- **verbose=**

기본적으로 이 명령은 출력을 stdout으로 보내 작업 요약 메시지를 작성합니다. 또는 세션 수준 메시지에 대해 상세 옵션을 0으로 설정(stdout으로 출력되지 않음)하거나 (swpackage 및 swmodify의 경우) 파일 수준 메시지에 대해 2를 설정할 수 있습니다.

오류 및 경고 메시지는 항상 stderr로 기록됩니다.

swlist 명령의 경우 상세 목록에는 각 *software\_selection* 피연산자의 적절한 수준에 대해 정의된 모든 속성을 포함합니다. 속성은 한 줄에 하나씩 표시되며 앞에 속성 키워드가 나옵니다.

-v 옵션을 0으로 설정하면 이 기본값을 바꿉니다.

모든 명령에 적용됩니다.

- **write\_remote\_files=false**

원격(NFS) 파일 시스템에서의 파일 작업을 방지합니다. 원격(NFS) 파일 시스템의 대상에서 설치, 복사, 제거 또는 패키징할 모든 파일은 생략됩니다.

true로 설정하고 슈퍼유저가 원격 파일 시스템에 대해 쓰기 권한을 가지고 있는 경우 원격 파일은 생략되지 않습니다.

swconfig, swcopy, swinstall, swpackage 및 swremove에 적용됩니다.

명령 옵션  
옵션(알파벳 순)

## B 문제 해결

이 부록에서는 SD-UX 오류 메시지가 사용되는 방법을 설명하고 SD-UX 오류 기록 프로세스를 검토하며, 발생할 수 있는 일반적인 문제점을 나열하고 해결 방법을 제시합니다.

표 B-1

### 이 장의 항목

설명하는 항목
440페이지의 “오류 기록”
442페이지의 “일반적인 문제점”

## 오류 기록

모든 SD-UX 명령(`swlist`와 `swacl` 제외)은 오류 메시지, 세션 요약 정보 및 작동 세부 사항을 `/var/adm/sw/<command>.log`에 기본적으로 위치한 명령별 로그 파일에 기록합니다. 예를 들어 `swinstall`의 로그 파일을 검토하려면 `/var/adm/sw/swinstall.log` 파일을 살펴봅니다. 또한 `swinstall`, `swcopy` 또는 `swremove` GUI에서 현재 세션의 대상 에이전트 로그 파일을 검토할 수도 있습니다. 대상 호스트에 로그인 액세스를 하는 경우 `agent` 로그 파일을 직접 볼 수 있습니다. 에이전트 로그 파일의 위치는 대상의 유형에 따라 달라집니다.

- 호스트의 기본 루트에서 작동하는 경우 `/var/adm/sw/swagent.log`
- 대체 루트의 경우 `<root_path>/var/adm/sw/swagent.log`
- 대상 또는 소스 저장소의 경우 `<depot_path>/swagent.log`

호스트의 데몬 로그 파일의 기본 위치는 `/var/adm/sw/swagentd.log`입니다. 이 로그 파일에는 에이전트를 시작할 때 발생하는 문제점, 특히 저장소나 루트에 대한 액세스가 거부되는 문제점에 대한 정보가 들어 있습니다.

---

### 참고

소스 및 대상 시스템이 모두 HP-UX 10.30 이상으로 업데이트되면 소스 저장소 시스템의 시스템 관리자는 어떤 사용자가 저장소에서 어떤 소프트웨어를 가져오고 언제 소프트웨어를 가져오는지를 추적할 수 있습니다. 부록 A, 399페이지의 “명령 옵션” 또는 162페이지의 “소스 저장소 감사”의 `source_depot_audit` 옵션을 참조하십시오.

---



## 오류 메시지

SD-UX 오류 메시지는 전반적인 작동 결과에 영향을 미치는 문제가 발생했음을 나타냅니다.

예를 들어 설치 세션에서 대상이 디스크 공간 부족으로 분석 단계에 실패한 경우 에이전트 로그 파일에서 다음과 같은 오류 메시지를 찾아볼 수 있습니다.

```
ERROR:The estimated disk space used on filesystem "/" is
14104 Kbyte blocks. This operation will exceed the
minimum free space for this disk. You should free up at
least 2280 Kbyte blocks to avoid installing beyond this
threshold of available user disk space. If you are
running interactive "swinstall", you must return to the
Selection Window and Unmark this target before using
"swremove" to free disk space.
```

## 경고 메시지

경고 메시지를 통해 부정적 결과를 가져올 수 있는 예상치 못한 무언가가 발생할 수 있음을 알려 줍니다. 경고가 발생해도 SD 세션은 계속 실행됩니다. 대화식 세션의 분석 단계 중에 발생하는 경고 메시지를 통해 계속 실행할 것인지 중단할 것인지를 결정할 수 있습니다.

예를 들어 SD-DATABASE.SD-DATABASE2 파일 세트를 대상 시스템의 여러 위치에 설치하는 경우 에이전트 로그 파일에서 다음과 같은 경고 메시지를 발견할 수 있습니다.

```
WARNING:A version of fileset
"SD-DATABASE.SD-DATABASE2,r=9.00.1C"
is already installed in another location (see previous
lines). Installing this version will create multiple
installed versions. This new multiple version will be
installed because the "allow_multiple_versions" option is
set to "true".
```

## 참고 사항

참고 사항은 오류도 아니고 예상치 못하거나 부정적이지도 않지만 사용자가 알고 있어야 할 이벤트에 대해 알려 주는 데 사용됩니다.

```
NOTE:The fileset "SD-DATABASE.SD-DATABASE1,r=9.00.1C" is
already installed. If you wish to reinstall this
fileset, change the "reinstall" option to "true".
```

## 일반적인 문제점

이 절에서는 발생할 수 있는 문제점 및 이를 해결할 수 있는 방법에 대해 설명합니다.

표 B-2

일반적인 문제점

문제점
대상 호스트의 데몬이나 에이전트에 연결할 수 없는 경우
GUI가 시작되지 않거나 지원 파일이 유실된 경우
객체에 대한 액세스가 거부되는 경우
네트워크 성능이 떨어지는 경우
연결 시간 제한 및 기타 WAN 문제점
디스크 공간 분석이 올바르지 않은 경우
패키징 프로그램이 실패한 경우
데몬 로그 파일이 너무 긴 경우
테이프 저장소를 읽을 수 없는 경우
설치에 실패한 경우
swinstall 또는 swremove가 실패하고 잠금 오류가 발생한 경우

## 대상 호스트의 데몬이나 에이전트에 연결할 수 없는 경우

다음과 같은 오류 메시지가 나타날 경우:

```
ERROR: Could not contact host <hostname>. Make sure the  
hostname is correct.
```

지정한 호스트 이름을 호스트 데이터베이스에서 찾을 수 없다는 의미입니다. 호스트 이름을 올바르게 입력했는지 확인하십시오(`nslookup` 명령을 사용하여 호스트 이름을 확인할 수 있습니다). 호스트 데이터베이스에 대상 호스트 이름이 없지만 호스트의 네트워크 주소를 알고 있는 경우 호스트 이름 대신 네트워크 주소(일반적으로 “점” 표기)를 사용할 수 있습니다.

다음과 같은 오류 메시지가 나타날 경우:

```
ERROR:Remote Procedure Call to a daemon has failed.  
Could not start a management session for <target>.  
Make sure the host is accessible from the network,  
and that its daemon, swagentd, is running. If the  
daemon is running see the daemon logfile  
on this target for more information.
```

SD-UX가 특정 대상 시스템의 데몬 프로그램에 연결할 수 없다는 의미입니다. 이는 예를 들어 로컬 호스트의 데몬이 실행되지 않고 있는 경우 대상 시스템을 지정하지 않아도 이 오류 메시지가 발생할 수 있음에 유의하십시오.

### 해결책

SD-UX 데몬/에이전트가 해당 대상 시스템에 설치되어 있지 않은 경우 이를 먼저 설치해야 SD-UX를 사용할 수 있습니다.

데몬/에이전트 구성 요소가 대상 시스템에 설치되어 있음을 확인했는데도 여전히 연결 문제가 발생하면 데몬이 실행 중인지 확인합니다.

1. 대상 시스템에서 다음과 같이 입력하십시오.

```
ps -e | grep swagentd
```

2. 데몬이 실행 중이 아니라면 대상 시스템의 `root` 권한으로 다음과 같이 입력하여 실행시킬 수 있습니다.

```
/usr/sbin/swagentd
```

3. 데몬이 실행 중인데 데몬 시작을 시도하는 경우 다른 데몬에 대한 메시지가 표시되는데, 이는 별 영향을 미치지 않습니다.

다음과 같이 입력하여 현재 실행 중인 데몬을 중지시킨 다음 다시 시작할 수 있습니다.

```
/usr/sbin/swagentd -r
```

이러한 문제가 발생하는 다른 가능한 원인은 450페이지의 “연결 시간 제한 및 기타 WAN 문제점” 절에 설명되어 있습니다.

---

**팁**

다음과 같이 입력하면 대상 시스템에 SD-UX 데몬이 설치되어 있고 실행 중인지 쉽게 판별할 수 있습니다.

```
/usr/sbin/swlist -l depot @ <one or more target hostnames>
```

이 명령은 각 대상에 연결하여 등록된 저장소 목록을 구하려고 시도합니다. SD-UX 데몬이 설치되어 있는 대상 시스템은 다음과 같은 메시지 중 하나를 표시합니다.

```
# Initializing...  
# Target <hostname> has the following depot(s):  
# <...insert list of depots...>
```

또는

```
# Initializing...  
WARNING: No depot was found for <hostname>.
```

데몬 작업에 대한 자세한 내용은 /var/adm/sw/swagentd.log의 데몬 로그 파일을 참조하십시오.

---

## GUI가 시작되지 않거나 지원 파일이 유실된 경우

GUI는 다음과 같은 방법으로 시작할 수 있습니다.

- `swinstall`, `swcopy` 또는 `swremove`의 경우 추가 옵션이나 인수를 지정하지 않고 명령을 입력합니다.
- 명령줄에 명령을 입력하는 경우 다른 옵션 및 인수와 함께 `-i` 옵션을 지정합니다 (`swlist`에 필수).
- 작업 브라우저의 경우 명령줄에 `sd`를 입력합니다.

GUI를 사용하는 경우 다음과 같은 문제점이 발생할 수 있습니다.

- 디스플레이를 열 수 없거나 올바르게 설정되지 않습니다.
- GUI 지원 파일이 없습니다.

### 해결책

원격 시스템에서 GUI를 호출한 경우 다음과 같은 오류 메시지가 표시됩니다.

```
Xlib: connection to <display> refused by server
Xlib: Client is not authorized to connect to Server
Error: Can't Open display.
```

사용자의 디스플레이를 식별할 수 있도록 원격 시스템에서 `$DISPLAY` 환경 변수를 올바르게 설정했는지 확인합니다. 올바르게 설정되어 있다면 `xhost(1)` 명령을 사용하거나 `/etc/X*.hosts` 파일을 수정하여 원격 호스트가 사용자의 **X** 서버에 연결되도록 합니다.

다음과 같은 오류 메시지가 나타날 경우:

```
swinstall: Error: cannot read file:
           /usr/lib/sw/ui/smc_install_copy.ui
```

— 또는 —

```
swremove: Error: cannot read file:
           /usr/lib/sw/ui/smc_remove.ui
```

`swinstall`이나 `swcopy`를 대화식으로 실행하기 위해서는 시스템에 `/usr/lib/sw/ui/smc_install_copy.ui` 파일이 설치되어야 하며 `swremove`를 실행하기 위해서는 `/usr/lib/sw/ui/smc_remove.ui` 파일이 설치되어야 함을 시스템이 알려줍니다. `/usr/lib/sw/ui` 디렉토리가 존재하는지 및 요청된 파일이 들어 있는지 확인합니다. 파일이 없는 경우 OS 미디어에서 SD-CMDS 파일 세트를 다시 설치해야 합니다.

## 객체에 대한 액세스가 거부되는 경우

SD-UX 객체에 대한 액세스가 거부되는 경우 다음과 같은 몇 가지 원인이 있을 수 있습니다.

- ACL 권한
- 호스트간 비밀 파일
- 저장소의 이미지 사본에 대한 작업

### 해결책

일반적으로 SD-UX가 객체에 대한 액세스를 거부하는 경우 사용자에게 필요한 액세스 권한이 없다는 메시지가 표시됩니다. 하지만 어떤 객체에 액세스할 수 없는지 확실하지 않을 수 있습니다. 예를 들어 `swcopy`를 사용하여 시스템 A에서 저장소로 제품을 복사할 때 SD-UX는 해당 ACL을 확인합니다.

1. 대상 저장소가 존재하는 경우 사용자에게 “삽입” 권한이 있는지 확인하기 위해 호스트 ACL을 검사합니다.
2. 대상 저장소가 존재하는 경우 사용자에게 쓰기 권한이 있는지 확인하기 위해 저장소 ACL을 검사합니다.
3. 사용자에게 소스 저장소에서 읽기 권한이 있는지 확인하기 위해 소스 저장소의 ACL을 검사합니다.
4. 또한 사용자 및 대상 시스템이 모두 제품에 대한 읽기 액세스를 할 수 있는지 확인하기 위해 소스 제품의 ACL도 검사합니다.

이러한 액세스 권한 중 하나라도 없는 경우 전체적인 작동이 허용되지 않으며 오류 메시지를 신중하게 읽고 정확한 원인을 이해해야 합니다. 어떤 종류의 보안 문제나 액세스 문제가 있는지 더 자세히 확인하려면 대상 시스템 데몬 로그 파일인

`/var/adm/sw/swagentd.log`를 참조하십시오.

### ACL 수정의 영향

기본 ACL을 통해 ACL을 매우 쉽게 관리할 수 있지만 항상 적절한 액세스 제어 수준을 제공하는 것은 아닙니다. 액세스를 제한하기 위해, 특히 `any_other` 읽기 권한을 제거하여 ACL을 변경하면 예상치 못한 방식으로 액세스가 제한될 수 있습니다.

`swcopy` 및 `swinstall`를 수행하기 위해서는 대상 시스템에 대한 호스트 항목이 필요합니다.

수행된 액세스 테스트나 각 작업에 대한 자세한 설명은 9장, 233페이지의 “SD-UX 보안”을 참조하십시오.

### swacl 명령을 통해서만 ACL 파일을 수정

SD-UX는 파일 시스템에 일반 텍스트 파일로 ACL을 저장하므로 일반적인 편집기를 사용하여 수정하려고 하는 경우가 있습니다. 그러나 이는 뜻하지 않게 ACL을 손상시킬 수 있습니다. 이렇게 손상되는 경우 대개 손상되었음을 알리는 메시지만을 내보내지만 `num_entries` 값을 업데이트하지 않고 ACL 파일에 추가 항목을 삽입하는 경우 보고되지 않은 문제점을 발생시킬 수 있으며 SD-UX가 액세스를 거부하는 원인이 될 수 있습니다. 예를 들어 ACL 파일에 사용자 항목을 삽입하는 경우 일반적인 장애가 발생할 수 있습니다. 이는 `any_other` 항목을 `num_entries` 한계 아래로 밀어냅니다. 따라서 ACL 관리자는 `any_other` 항목을 읽을 수 없으며 액세스 문제가 발생하게 됩니다. 이러한 상황이 발생하지 않도록 하기 위한 가장 좋은 방법은 ACL을 조작할 때 항상 `swacl` 명령을 사용하는 것입니다.

### 호스트간 비밀 파일

기본 `/var/adm/sw/security/secrets` 파일에는 다음과 같은 하나의 항목만 포함됩니다.

```
default      -sdu-
```

컨트롤러가 실행될 수 있는 모든 호스트를 명시적으로 지정하려면 `-sdu-`를 다른 기본 비밀 파일로 바꾸거나 항목 전체를 삭제합니다. 비밀 파일에 대한 자세한 설명은 9장, 233페이지의 “SD-UX 보안”을 참조하십시오.

컨트롤러(`swinstall`, `swcopy` 등의 경우)는 컨트롤러가 실행되는 시스템에서 비밀 파일을 찾아 이를 암호화된 형태로 에이전트로 전달합니다. 컨트롤러로부터 요청을 받은 에이전트는 호출이 들어온 호스트에서 비밀 파일을 찾아 암호화한 다음 이를 컨트롤러가 제공한 것과 비교합니다. 두 개의 비밀 파일이 일치하지 않으면 액세스는 거부됩니다. 이러한 메커니즘에 문제가 발생하는 경우 모든 시스템에 일치하는 항목이 있는지 확인하십시오. 모든 호스트에서 이전 비밀 파일(9.x에서는

`/etc/newconfig/sd/secrets`, 10.x에서는

`/usr/newconfig/var/adm/sw/security/secrets`)로 되돌릴 수도 있으며 단 순히 비밀 파일을 모든 호스트로 복사할 수도 있습니다.

### 저장소 이미지에 대한 작업

다른 시스템에서 사용하기 위해 저장소 이미지를 복사할 때 `cp`, `tar`, `cpio`, `dd` 및 기타 명령 사용 중에 문제가 발생할 수 있습니다. 이미지로 된 저장소와 제품 ACL에는 저장소가 만들어진 호스트에 대한 정보가 내장되어 있습니다. 특히 ACL 기본 영역은 잘못된 것일 수 있으며 로컬 사용자가 원래의 호스트 사용자와 혼동될 수 있습니다. 예를

들어 액세스 목록에 로컬 사용자를 추가하려고 하면 원격 사용자에 대한 액세스를 허용하게 됩니다. **ACL의 기본 영역이 작성될 때 해당 세트에서 이를 변경할 수 있는 방법은 없습니다.**

이러한 이미지를 ACL에 들어 있는 모든 호스트 이름을 해석하지 못하는 시스템으로 가져오는 경우 이러한 이미지에 대한 또 다른 문제가 발생합니다(*resolver(4)* 및 *nslookup(1)* 참조).

“단계별” 설치를 작성하는 것이 목적이라면 **swcopy**를 사용하여 저장소를 배포합니다. 이 명령은 로컬 템플릿을 바탕으로 저장소의 각 인스턴스에 대한 ACL을 새로 작성합니다.

저장소의 유일한 목적이 그러한 이미지를 배포하기 위한 것이라면 **swpackage create\_target\_acls** 옵션을 **false**로 설정하여 **swpackage** 명령 실행 중에 저장소와 제품에 ACL이 작성되는 것을 방지할 수 있습니다. 이 옵션은 테이프 및 CD-ROM 이미지를 만듭니다. ACL이 없는 저장소와 제품에 대해 로컬 슈퍼유저는 모든 권한을 갖게 되고 다른 모든 사용자와 시스템은 읽기 액세스 권한을 갖습니다. **swcopy**나 **swinstall**을 사용하여 ACL이 없는 저장소를 복사하거나 설치할 때 대상 호스트의 템플릿을 바탕으로 복사본(설치)이 자동으로 보호된다는 점에 유의하십시오.



## 네트워크 성능이 떨어지는 경우

네트워크 대역폭이 “병목”인 환경에서 `swinstall`이나 `swcopy`를 사용하는 경우 소스와 대상 시스템 간의 파일 전송 속도가 매우 느려질 수 있습니다.

### 해결책

`compress_files=true` 옵션은 소스 저장소에서 대상 시스템으로 전송되는 파일을 압축합니다. 이는 네트워크 사용량을 대략 50% 정도 감소시킬 수 있으며 정확한 압축률은 파일의 형식에 따라 다릅니다. 바이너리 파일은 50% 이하로 압축되고 텍스트 파일은 좀 더 압축됩니다.

속도가 느린 네트워크(약 초당 50KB 이하)에서 전송될 때 성능이 가장 큰 폭으로 개선되며 소스 저장소 서버는 한꺼번에 두 세 개의 대상 호스트를 전송합니다.

---

### 참고

이 옵션은 네트워크 대역폭이 전체 처리량을 명확하게 제한하는 경우에만 `true`로 설정해야 합니다. 고속 네트워크 또는 많은 대상 호스트가 동시에 연결되어 있는 저장소 서버에 이 옵션을 사용하는 경우, 소스 저장소가 압축되지 않은 상태라면 이 옵션으로 인해 전체 처리량이나 성능을 실제로 떨어질 수 있습니다.

---

이 옵션이 사용자의 상황에서 도움이 되는지 확실하지 않다면 이 옵션값을 변경하기 전에 몇 가지 설치나 복사 작업의 처리량을 비교하여 보십시오.

성능 옵션에 대한 자세한 내용은 8장, 213페이지의 “안정성 및 성능”을 참조하십시오.

## 연결 시간 제한 및 기타 WAN 문제점

처리 속도가 늦은 WAN에서 운용되는 경우, 다른 시스템에 있는 원격 에이전트와의 네트워크 연결을 설정하고 유지할 때 SD-UX에 시간 제한 문제가 발생할 수 있습니다.

다음과 같은 메시지가 나타날 경우:

```
ERROR:A Remote Procedure Call to a daemon has failed.
Could not start a management session for <target>.
Make sure the host is accessible from the network, and
that its daemon, swagentd, is running. If the daemon is
running see the daemon logfile on this target for more informa
tion.
```

또는

```
ERROR: Could not perform the requested operation for
<target>, possibly due to a network communications
failure. Check that the host is still accessible from
the network.
```

그리고 시스템이 가동 중이고 데몬 프로그램(**swagentd**)이 실행 중임을 확인했다면 네트워크 지연에 의해 연결 시간 제한이 발생했을 수 있습니다.

### 해결책

RPC(Remote Procedure Call)를 실행할 때 명령줄이나 기본값 파일에 `rpc_timeout` 옵션을 더 높은 값으로 지정하여 SD-UX가 사용하는 시간 제한 값을 증가시키십시오. RPC 시간 제한 값은 0에서 9까지 가능하며 9가 가장 긴 시간 제한 값입니다. 기본 RPC 시간 제한 값은 5입니다. 이 값이 어떤 특정 시간 단위를 나타내는 것은 아님에 유의하십시오. `rpc_timeout` 옵션에 대한 자세한 내용은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.

`rpc_timeout`을 증가시키면 설치 또는 복사 세션의 대상 에이전트가 소스 에이전트에 연결하려는 경우 시간 제한이 발생하는 상황에서도 도움이 될 수 있습니다. 이 문제는 에이전트 로그 파일에 나타나는 다음과 같은 오류 메시지를 통해 알 수 있습니다.

```
ERROR: Could not open remote depot/root <path> due to
an RPC or network I/O error.
```

```
ERROR: Cannot open source. Check above for errors, as
well as the daemon logfile on the source host (default
location: /var/adm/sw/swagentd.log).
```

```
ERROR: Cannot continue the Analysis Phase until the
previous errors are corrected.
```

속도가 느린 네트워크에서 RPC 시간 제한에 영향을 미치는 또 다른 요소는 네트워크 프로토콜 선택입니다. SD-UX는 UDP 및 TCP 기반 통신을 모두 지원합니다(기본값은 TCP). TCP의 경우 연결을 기반으로 하기 때문에 WAN에서 더 안정적입니다. 어떤 이유에서 TCP 연결에 실패하면 SD는 UDP 연결을 시도합니다. 기본 바인딩은 `-x rpc_binding_info` 옵션을 사용하여 설정할 수 있습니다.

기본적으로 데몬 프로그램(`swagentd`)은 UDP 기반 RPC와 TCP 기반 RPC를 모두 청취한다는 점에 유의하십시오. `rpc_binding_info` 옵션에 대한 자세한 내용은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.

마지막 WAN 관련 문제는 대화식 GUI를 사용할 때 발생할 수 있습니다. 대화식 세션의 분석 및 실행 단계에서 최신 상태 정보를 확보하기 위해 주기적으로 각 대상 에이전트를 폴링합니다. 해당 대상 시스템에 대한 상태 폴링 간격을 나타내는 시간(초 단위)은 `polling_interval` 옵션을 사용하여 제어할 수 있습니다. 아주 작은 양의 데이터 전송도 문제가 되는 네트워크에서는 이러한 폴링 간격을 증가시킴으로써 폴링 빈도를 감소시키고 대화식 세션이 네트워크에 가하는 전반적인 부담을 감소시킬 수 있습니다. `polling_interval` 옵션에 대한 자세한 내용은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.

## 디스크 공간 분석이 올바르지 않은 경우

디스크 공간 분석이 정상적으로 수행되었는데도 설치나 복사를 실행할 때 디스크 공간 부족이 발생하는 경우가 있습니다. 자세히 살펴보면 디스크 공간 분석 결과가 실제 사용 가능한 공간과 다르다는 것을 알 수 있습니다.

### 해결책

이 문제의 원인:

- 설치 관련 제어 스크립트가 분석 단계 중에 예기치 않은 파일을 추가로 작성하거나 복사함으로써 디스크 공간을 차지했습니다.
- 분석이 완료되었을 때 대상 시스템이 유틸 상태가 아니며 다른 작업(SD-UX와 무관)에서 디스크 공간을 사용하고 있었습니다.
- 제품이 설치되었거나 복사된 저장소를 작성할 때 `package_in_place` 옵션과 함께 `swpackage` 명령을 `true`로 설정하여 작성함으로써 제품이 패키징된 이후 소스 파일이 계속 수정되었습니다. `swverify` 명령으로 이 문제를 진단할 수 있습니다.

## 패키징 프로그램이 실패한 경우

PSF(Product Specification File)에서 `end` 키워드를 잘못 사용하여 `swpackage` 작업에 실패하는 경우가 있습니다.

### 해결책

`end` 키워드는 PSF에서 저장소, 공급업체, 제품, 하위 제품 또는 파일 세트 사양의 끝을 표시합니다. 값을 지정할 필요가 없으며 생략 가능한 키워드입니다. 그러나 키워드를 잘못된 위치에 사용하는 경우 이 사양에 오류가 발생할 것입니다. 이 키워드를 사용하는 경우 모든 객체 사양(특히 마지막에 나오는 사양)에 대해 `end` 키워드를 지정했는지 확인하십시오.

## 명령 로그 파일이 너무 커지는 경우

SD-UX 명령 로그 파일의 내용을 줄이려면 다음 과정을 수행합니다.

### 해결책

메시지를 최소한으로 줄이려면 옵션 파일 중 하나에서 verbose 명령 옵션을 0으로 설정하거나 명령줄에서 -x 옵션을 사용합니다. 예를 들어 swpackage를 실행할 때 명령줄에 -x swpackage.verbose=0을 입력하면 swpackage 로그에 대한 항목 수를 최소한으로 줄일 수 있습니다. 옵션 설정에 대한 자세한 내용은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.

## 데몬 로그 파일이 너무 긴 경우

SD-UX 데몬 로그 파일이 너무 길어서 이를 짧게 하려는(절단하려는) 경우 다음과 같은 과정을 수행합니다.

### 해결책

데몬이 현재 실행 중이라면 해당 로그 파일을 절대 제거하지 마십시오. 실행 중인 데몬은 로그 파일을 제거한 후에도 계속해서 로그 메시지를 자신의 로그 파일에 기록하려고 하기 때문에 이후 발생하는 정보가 유실될 수 있습니다. 데몬이 실행 중이면 로그 파일이 사용하는 디스크 공간도 사용 가능 공간으로 해제되지 않습니다.

또는 다음과 같이 입력하여(루트 계정으로) 로그 파일을 절단합니다.

```
echo > /var/adm/sw/swagentd.log
```

이는 로그 파일의 이전 데이터를 공백 문자열로 바꿉니다.

데몬이 실행 중일 때 실수로 데몬 로그 파일을 제거한 경우, 이후 발생하는 데몬 로그 메시지를 보고 로그 파일이 사용하고 있는 디스크 공간을 해제하려면 데몬을 중지시켰다 다시 시작해야 합니다. 다음과 같이 입력하여 데몬을 중지(kill)시킬 수 있습니다.

```
/usr/sbin/swagentd -k
```

다음과 같이 입력하여 현재 실행 중인 데몬을 중지시킨 다음 다시 시작할 수 있습니다.

```
/usr/sbin/swagentd -r
```

## 테이프 저장소를 읽을 수 없는 경우

테이프 저장소에 액세스하려고 할 때 데몬 로그 파일에 다음과 같은 오류 메시지가 있는 경우 이는 테이프가 손상되었거나 SD-UX 형식이 아님을 나타냅니다.

```
ERROR:The INDEX file on the source did not exist or could not be read.
```

```
ERROR:The target <depot_path> could not be opened.
```

### 해결책

테이프 저장소를 올바르게 지정했는지 및 올바른 테이프가 드라이브에 들어 있는지 확인합니다. SD-UX는 SD-UX 형식으로 되어 있는 테이프만을 읽습니다. 예를 들어 SD-UX는 업데이트 형식의 테이프는 읽지 못합니다.

## 설치에 실패한 경우

설치 프로세스 중에 일부만 완료된 상태에서 설치 장애가 발생할 수 있습니다.

### 해결책

SD-UX에서는 사용자에게 몇 가지 다시 시작할 수 있는 옵션을 제공합니다.

- 명령줄에서 같은 명령을 다시 실행합니다.
- 자동으로 저장되었던 `swinstall.last` 세션 파일을 다시 호출합니다(63페이지의 “세션 파일” 참조).
- 검사점 옵션을 다시 설정합니다.

기본적으로 SD-UX는 파일 세트 수준에서 검사점을 설정하는데, 이는 마지막으로 시도할 파일 세트에서 파일 전송을 시작한다는 의미입니다.

`reinstall_files` 옵션을 `false`로 설정함으로써, SD-UX는 마지막으로 시도되었던 파일에서부터 배포 및 설치를 다시 시작합니다. (SD-UX에서는 파일 수준 아래에서 검사점을 지원하지 않습니다.)

`reinstall`과 `reinstall_files` 옵션을 모두 `true`로 설정하여 검사점을 모두 재정의할 수 있습니다. 자세한 내용은 부록 A, 399페이지의 “명령 옵션”을 참조하십시오.

## swinstall 또는 swremove가 실패하고 잠금 오류가 발생한 경우

swinstall 또는 swremove가 실패하는 경우 다음과 같은 메시지가 나타납니다.

```
Cannot lock "/" because another command holds a conflicting  
lock. The process id of that command is #####.
```

### 해결책

swinstall이나 swremove 명령이 실행되지 못하도록 하는 다른 SD 명령이 실행 중입니다. 해당 명령의 실행이 완료될 때까지 기다렸다가 다시 시도하십시오.

문제 해결  
일반적인 문제점



# C SD-UX 교체 또는 업데이트

이 부록에서는 `install-sd` 명령을 사용하여 SD-UX를 대체하거나 업데이트하는 방법에 대해 설명합니다.

표 C-1

## 이 장의 항목

설명하는 항목
458페이지의 “SD 재설치”
460페이지의 “사용할 수 없는 SD-UX 버전 교체”
461페이지의 “새로운 버전의 SD-UX 설치”

## SD 재설치

SW-DIST라는 소프트웨어 제품은 모든 SD-UX 기능, 명령 및 도구를 제공합니다. 이 제품은 HP-UX 11i 미디어에 들어 있습니다.

SW-DIST에 포함된 파일이 삭제되거나 손상되는 경우 제품을 다시 설치해야 합니다. `install-sd` 명령을 통해 HP-UX 11i 미디어나 저장소에서 SD-UX 제품을 설치할 수 있습니다. 이 명령을 사용하여 소스 저장소에 있는 SD-UX 패치도 설치할 수 있습니다.

- `install-sd` 명령은 HP-UX 버전 10.20 또는 11.00에서는 지원되지 않습니다.
- 11i SD-UX 형식으로 패키징된 HP-UX 소프트웨어를 설치하거나 복사하기 위해서는 SW-DIST 11i 버전이 필요합니다.
- `update-ux` 명령은 OS 업데이트를 위해 SD-UX 이전 버전이 사용하는 `swgettools` 스크립트를 대체합니다.

### 추가 정보

HP-UX 업데이트를 위한 자세한 지침은 다음을 참조하십시오.

- **HP-UX 11i 설치 및 업데이트 설명서**
- `update-ux` (1M) 맨페이지

이러한 문서들은 HP-UX Instant Information CD-ROM과 다음 사이트의 HP-UX 11i 절에서 구할 수 있습니다.

<http://docs.hp.com>(영문)

<http://docs.hp.com/ko>(한글)

### 전제 조건

`install-sd` 명령과 그에 따르는 `swagent.z` 파일은 `/var/tmp` 디렉토리에 최소 2MB의 여유 공간을 필요로 합니다. 이 디렉토리에 공간이 충분하지 않으면 `install-sd`는 실패합니다. `/var/tmp`에 적절한 공간이 확보되어 있는지 확인하려면 다음과 같이 입력합니다.

```
bdf /var/tmp
```

## install-sd 사용

**구문** `install-sd -s source_depot_location`

**옵션 및 피연산자** `source_depot_location` 옵션은 소스 미디어 위치에 대한 절대 경로를 지정합니다. 가능한 미디어 위치는 다음과 같습니다.

- 로컬 디렉토리
- SD-UX 미디어 CD-ROM이 로드된 CD-ROM 마운트 지점
- 이 구문을 사용하여 지정해야 하는 원격 시스템(또는 호스트)과 저장소 조합

`system_name:/depot_path`

예를 들면 다음과 같습니다.

**swtest:/var/spool/sw**

### 명령 참고 사항

- 이 명령은 정상적으로 완료하면 0 값을 반환하고 오류가 발생하면 1 값을 반환합니다.
- `install-sd` 세션은 주요 작업 및 각 세션의 시작과 끝에 대해 메시지를 기록합니다. 모든 WARNING 및 ERROR 상태가 `stderr`에 기록됩니다.
- 자세한 이벤트가 `/var/adm/sw/install-sd.log`에 기록됩니다.

**예제** `install-sd -s swtest:/var/spool/sw`

## 사용할 수 없는 SD-UX 버전 교체

대상 시스템의 SD-UX 버전을 사용할 수 없는 경우 먼저 `install-sd` 및 `swagent.z` 파일을 사용자 시스템의 `/var/tmp`로 로드한 다음, `install-sd`를 사용하여 `SW-DIST`를 다시 설치합니다. `install-sd` 유틸리티는 `catalog/SW-DIST/pfiles` 디렉토리에 들어 있습니다.

`cp`(로컬 CD-ROM에서 복사하는 경우) 또는 `rcp`(원격 시스템의 소프트웨어 저장소에서 복사하는 경우)를 사용하여 `install-sd`를 사용자 시스템에 로드합니다.

예를 들어 `/SD_CDROM`에 마운트된 로컬 CD-ROM에서 `/var/tmp`로 `install-sd`를 로드하려면 다음을 입력합니다.

- 단계 1. CD-ROM에서 사용자 시스템으로 `install-sd`를 복사합니다.

```
cp /SD_CDROM/catalog/SW-DIST/pfiles/ \
install-sd /var/tmp
```

- 단계 2. CD-ROM에서 `swagent.z` 파일을 복사합니다.

```
cp /SD_CDROM/catalog/SW-DIST/pfiles/ \
swagent.Z /var/tmp
```

- 단계 3. `install-sd` 실행을 가능하게 만듭니다.

```
chmod +x /var/tmp/install-sd
```

- 단계 4. `install-sd`를 실행합니다.

```
/var/tmp/install-sd -s /SD_CDROM
```

그러면 `SW-DIST` 제품은 자신을 CD-ROM 으로부터 사용자 시스템에 설치합니다.

---

## 새로운 버전의 SD-UX 설치

시스템에 새로운 버전의 SD-UX를 설치하고자 하며 /usr/sbin/install-sd가 시스템상에 없는 경우 다음과 같이 하십시오.

(두 가지 단계 모두에서 *source\_depot\_location*은 새로운 버전의 SD-UX가 들어 있는 저장소나 미디어에 대한 절대 경로입니다.)

- 단계 1. root 권한으로 다음을 입력합니다.

```
/usr/sbin/swinstall -r -s \  
source_depot_location\SW-DIST.SD-UPDATE \  
& /var/adm/sw/install-sd.root 2>/dev/null
```

- 단계 2. 새로운 버전을 설치합니다.

```
/usr/sbin/install-sd -s source_depot_location
```

---

### 참고

이 install-sd 명령은 앞으로 출시되는 버전에서는 사용할 수 없습니다.

---

SD-UX 교체 또는 업데이트  
새로운 버전의 SD-UX 설치

# D Software Distributor 파일 및 파일 시스템 구조

이 장에서는 주요 Software Distributor 파일에 대한 정보를 제공합니다.

추가 정보는 다음과 같은 설명서 참조 페이지를 참조하십시오.

<i>sd</i> (5)	Software Distributor 파일에 대한 최신 정보
<i>sd</i> (4)	모든 Software Distributor 파일의 파일 레이아웃
<i>swpackage</i> (4)	패키징하는 중에 작성된 Software Distributor 파일의 파일 레이아웃

표 D-1

이 장의 항목

설명하는 항목
464페이지의 “에이전트 파일 시스템 구조”
467페이지의 “Software Distributor 컨트롤러 파일 시스템 구조”
468페이지의 “Installed Products Database”

## 에이전트 파일 시스템 구조

에이전트 구성 요소는 다음과 같이 구성됩니다.

표 D-2

### 에이전트 구성 요소

<code>/dev/rmt/Om</code>	대상 테이프 장치 파일의 기본 위치
<code>/usr/contrib/bin</code>	파일 압축에 사용되는 <b>gzip</b> 실행 파일의 위치
<code>/usr/sbin/swagent</code>	<b>SD-UX</b> 에이전트
<code>/usr/sbin/sw</code>	<code>swinstall</code> 과 <code>swremove</code> 에 의해 사용되는 유틸리티가 들어 있는 디렉토리
<code>/usr/sbin/sw/control_utils</code>	<b>SD</b> 제어 스크립트가 사용하는 일반적인 유틸리티가 들어 있는 파일
<code>/usr/sbin</code>	<b>Software Distributor</b> 데몬 및 기타 모든 실행 파일이 들어 있는 디렉토리
<code>/usr/lib/sw/examples</code>	다양한 패키지 예제 및 <b>PSF</b> 파일이 들어 있는 디렉토리
<code>/usr/lib/sw/sys.defaults</code>	모든 옵션 및 기본값이 들어 있는 파일
<code>/var/adm/sw</code>	<b>Software Distributor</b> 제품의 모든 데이터와 로그 파일의 기본 위치가 들어 있는 디렉토리
<code>/var/spool/sw</code>	로컬 <b>Software Distributor</b> 저장소의 기본 디렉토리
<code>/var/adm/sw/defaults</code>	<b>Software Distributor</b> 시스템 전체에 적용되는 기본값 파일
<code>/var/adm/sw/host_object</code>	로컬 호스트에 등록된 저장소 목록
<code>/var/adm/sw/host_object_np</code>	비권한 모드가 실행 중일 때 로컬 호스트에 등록된 저장소 목록



표 D-2

에이전트 구성 요소(계속)

<code>/var/adm/sw/products</code>	루트(/) 디렉토리 아래에 설치된 모든 제품에 대한 정보가 들어 있는 <b>IPD(Installed Products Database)</b> , 파일 및 하위 디렉토리
<code>/var/adm/sw/queue</code>	<b>Jobs</b> 데이터베이스가 들어 있는 디렉토리
<code>/var/adm/sw/save</code>	패치에 대한 <b>SD</b> 의 저장 영역인 디렉토리
<code>/var/adm/sw/save_custom</code>	패치에 대한 사용자 정의 저장 영역인 디렉토리
<code>/var/adm/sw/security</code>	호스트 <b>ACL(Access Control List)</b> , 모든 기본 <b>ACL</b> 및 비밀 파일이 들어 있는 디렉토리
<code>/var/adm/sw/swagent.log</code>	설치된 소프트웨어 작동에 대한 자세한 내용이 기록된 에이전트 로그 파일
<code>/var/adm/sw/swagentd.log</code>	호스트 및 보안 작동에 대한 자세한 내용이 기록된 데몬 로그 파일
<code>/var/adm/sw/sw&lt;task&gt;.log</code>	각 작업에 대한 요약 내용이 들어 있는 컨트롤러 로그 파일로서, <task>는 다음 값 중 하나입니다. <ul style="list-style-type: none"><li>• install</li><li>• remove</li><li>• config</li><li>• modify</li><li>• package</li><li>• reg</li><li>• verify</li></ul>

표 D-2

에이전트 구성 요소(계속)

`/var/adm/sw/tmp`

임시 파일을 위한 디렉토리

`/var/home/USER_NAME`

비권한 모드로 실행 중일 때  
**admin\_directory**의 기본 위치

`$HOME/.swdefaults`

사용자가 지정한 기본값이 들어 있는  
파일. 이 파일이 없으면 **Software  
Distributor**는  
`$HOME/.sw/defaults`에서 사용자  
지정 기본값을 찾습니다.

## Software Distributor 컨트롤러 파일 시스템 구조

컨트롤러 파일 시스템 구조는 에이전트의 모든 파일(464페이지의 “에이전트 파일 시스템 구조” 참조)과 다음 파일로 구성됩니다.

표 D-3

### 컨트롤러 파일 시스템 구조

<code>/usr/lib/sw/help</code>	온라인 도움말을 위한 도움말 파일이 들어 있는 디렉토리
<code>/usr/lib/sw/ui</code>	GUI가 사용하는 설정 파일이 들어 있는 디렉토리
<code>/usr/lib/X11/app-defaults</code>	GUI에 대한 X11 리소스 정의
<code>/usr/lib/nls/msg/\$LANG/sw*.cat</code>	데몬, 에이전트 및 공유 메시지에 대한 메시지 카탈로그
<code>/usr/newconfig/var/adm/sw</code>	<code>/var/adm/sw</code> 로 조건부 복사되는 데이터 파일
<code>/var/adm/sw/queue</code>	작업에 대한 모든 데이터가 들어 있는 디렉토리
<code>/var/adm/sw/sw&lt;task&gt;.log</code>	컨트롤러 로그 파일
<code>/var/adm/sw/defaults.hosts</code>	GUI에 대한 시스템 수준의 <code>defaults.hosts</code> 파일
<code>/var/adm/sw/.sdkey</code>	원격 작업 GUI를 가능하게 하는 키 파일
<code>/var/adm/sw/target_hosts</code>	각 대상 호스트에 대한 캐시 파일의 위치
<code>/var/adm/sw/ui/preferences</code>	GUI 사용자 보기 기본 설정이 저장된 디렉토리

---

## Installed Products Database

Software Distributor 명령은 IPD(Installed Product Database)가 설치되어 있는 시스템에서 설치, 제품, 및 파일 세트를 추적할 수 있습니다. IPD는

`/var/adm/sw/products` 디렉토리에 위치하며 루트 디렉토리(/) 아래에 설치된 모든 제품에 대한 정보를 담고 있는 일련의 파일 및 하위 디렉토리입니다. 이러한 정보에는 제품, 파일 세트 및 파일을 기술하는 모든 속성이 포함됩니다. `swinstall`, `swconfig` 및 `swremove` 작업은 명령이 실행될 때 자동으로 이러한 IPD 정보를 추가, 변경 및 삭제합니다.

IPD 파일은 수동으로 직접 수정할 수 없으며 `swmodify`를 사용하여 로컬 IPD와 로컬 저장소 카탈로그 정보를 변경할 수 있습니다.

저장소의 경우 IPD 파일에 해당되는 것이 카탈로그 파일입니다. `swcopy`를 사용하여 저장소를 작성하거나 수정하면 저장소와 그 내용을 설명하는 카탈로그 파일이 기본적으로 `/var/spool/sw/catalog`에 만들어집니다.

IPD에는 소프트웨어 객체와 ACL에 대한 읽기/쓰기 동시 액세스를 관리하는 `swlock` 파일도 포함됩니다.

### 추가 정보

- 117페이지의 “IPD 수정(`swmodify`)”

## 용어집

**참고:** 이 설명서에서 처음 정의되는 용어는 **굵은체**로 표시됩니다. **고딕체** 용어는 용어집의 다른 용어를 참조하는 경우를 나타냅니다.

### ㄱ

**개발자 호스트** 소프트웨어 응용 프로그램 파일이 차후 통합 및 배포 준비를 위해 위치할 시스템. 개발자 호스트를 사용하여 제품 테이프나 저장소를 조합, 구성 및 작성할 수 있습니다.

**객체 SD-UX**가 패키징, 배포, 설치 및 관리하는 소프트웨어. 소프트웨어(대상 루트에 설치되어 있거나 저장소에서 사용 가능), 컨테이너(저장소, 루트, 대체 루트) 및 **작업**의 세 가지 객체 클래스가 있습니다.

**검색 가능 제품** 설치되었을 때 대체 제품 디렉토리로 재배치할 수 있는 제품. 제품을 찾을 수 없는 경우 제품은 항상 정의된 제품 디렉토리 안에만 설치해야 합니다.

**경로** 파일에 대한 전체 경로 이름을 지정하는 속성

**공급업체** 공급업체 사양이 **PSF**에 포함되면 **swpackage**는 공급업체 및 태그 키워드를 필요로 합니다.

**공유 비밀 파일 SD-UX** 보안에서 보안을 강화하기 위해 분산된 통신을 암호화하고 암호화를 해제하는데 사용되는 암호가 들어 있는 파일

**관리 호스트 로컬 호스트**를 참조하십시오.

**권한 부여 SD-UX** 보안에서 사용자가 **액세스 제어 목록**에서 정의한 대로 특정 작업을 수행하기 위해 필요한 권한을 가지고 있는지 확인하는 기능

**그래픽 사용자 인터페이스(GUI)** **sd**, **swinstall**, **swcopy**, **swlist**, **swremove** 등의 명령에 제공되는 창과 풀다운 메뉴가 있는 **OSF/Motif™** 사용자 인터페이스. **명령줄 사용자 인터페이스(CLUI)** 및 **터미널 사용자 인터페이스(TUI)**도 참조하십시오.

**그룹 SD-UX** 보안에서 사용자 집합

**그룹 이름 SD-UX** 보안에서 사용자의 기본 그룹

**기본 루트** 소프트웨어가 설치되고 구성되는 시스템

**기본 소프트웨어** 패치에 의해 변경될 소프트웨어

**기본 옵션 SD-UX**의 명령 작동 방식과 정책에 영향을 미치는 변경 가능한 값. 기본 옵션은 **기본값 파일**에 들어 있습니다. 자세한 내용은 부록 **A, 399**페이지의 “명령 옵션”을 참조하십시오.

**기본 호스트 파일 SD-UX** 명령에 대한 기본 **호스트** 목록이 들어 있는 파일

(시스템 수준 기본값의 경우

`/var/adm/sw/defaults.hosts`, 사용자 수준 기본값의 경우 `$HOME/.sw/defaults.hosts`)

**기본값 파일** 각 **SD-UX** 명령에 대한 **기본 옵션**과 피연산자가 들어 있는 파일

(시스템 수준 기본값의 경우 `/var/adm/sw/defaults`, 사용자 수준 기본값의 경우 `$HOME/.sw/defaults`)

### ㄴ

**네트워크 소스** 하나의 호스트로부터 여러 개의 네트워크 소스가 있을 수 있으며 각각은 해당 호스트의 단일 **swagentd** 데몬에 의해 서로 다른 저장소에 대해 작동합니다. 네트워크 소스는 호스트 이름 및 저장소 디렉토리에 의해 식별됩니다.

노드

**노드** 클라이언트 호스트의 다른 이름. **클라이언트**를 참조하십시오.

**누적 패치 대체 패치**를 참조하십시오.

ㄷ

**다운데이트** 설치되어 있는 소프트웨어 버전을 이전 버전으로 덮어쓰는 작업

**단계 LAN**에서 각 대상 그룹에 대해 로컬인 중간 저장소를 설정하는 방식. 이를 통해 네트워크 처리량을 감소시킬 수 있습니다.

**단계별 설치 단계**를 참조하십시오.

**단일 지점 관리(SPA)** 단일 컨트롤러 시스템에서 여러 원격 대상으로 배포하고 이에 대한 관리나 모니터링도 동시에 수행하는 기능. **원격 작업**을 참조하십시오.

**대상 파일**이 설치될 경로

**대상 SD-UX**를 사용하여 소프트웨어를 설치 또는 관리할 시스템. 네트워크상에는 일반적으로 여러 대상 시스템이 있고 이들은 시스템 이름, 네트워크 주소, 사용자 이름 또는 사용자 그룹 등으로 식별합니다. 대상에는 **기본 루트**, **대체 루트** 또는 **저장소** 등이 포함될 수 있습니다. 대상 역시 **원격 작업**의 객체가 될 수 있습니다.

**대상 그룹** 대부분의 SD-UX 명령에서 -t *target\_file* 옵션을 명령에 대한 피연산자로 사용하여 이전에 정의한 **대상 선택** 목록을 읽을 수 있습니다. 또한 **원격 작업**이 가능한 경우 GUI 프로그램에서 대상 그룹 파일을 읽거나 저장할 수도 있습니다.

**대상 선택** 작동시킬 **대상**으로 선택한 시스템 또는 소프트웨어 객체 그룹. 차후 다시 사용하기 위해 선택한 내용을 저장할 수 있습니다. **대상 그룹**을 참조하십시오.

**대체 루트/대체 루트 디렉토리** 소프트웨어 설치를 위한 대상. 여기서 대상은 **기본 루트(/)**가 아니고 소프트웨어는 저장되거나 참조될 수는 있지만 구성되거나 사용될 수는 없습니다.

**대체 저장소 디렉토리** 기본 위치가 아닌 다른 곳에 위치한 저장소 디렉토리

**대체 패치** 해당 파일 세트에 대한 모든 이전 패치를 대체하는 패치

**데몬 에이전트**가 소프트웨어 관리 작업을 수행할 수 있도록 예약하는 SD-UX 프로그램. **SD-UX 컨트롤러**에서 데몬은 예약된 작업에 대한 작업 대기열을 폴링합니다.

**등록** 호스트에서 어떤 저장소를 사용할 수 있는지 판별하고 해당 저장소를 사용할 수 있도록 만드는 과정. 등록 정보는 저장소나 루트의 식별자(호스트 파일 시스템에서의 경로)로 구성됩니다. 이 정보는 시작할 때 자체 파일을 읽는 데몬에 의해 관리됩니다.

**등록 해제** *swreg* 명령을 사용하여 **저장소 등록**을 제거합니다. 그러면 네트워크에서 저장소에 액세스할 수 없게 됩니다.

**디렉토리 저장소** 저장소가 위치한 대상 호스트상의 디렉토리. 기본값은 /var/spool/sw입니다.

**디스크 공간 분석(DSA)** 호스트의 사용 가능한 디스크 공간이 선택된 제품을 설치하기에 충분한지 판별하는 프로세스

ㄹ

**로그** 각 SD-UX 명령은 로그 파일에 해당 작업을 기록합니다(swlist 명령은 예외). 다양한 로그 파일의 기본 위치는 /var/adm/sw/<command>.log입니다.

**로드/로드 단계** swinstall과 swcopy가 제품 파일을 호스트로 로드할 때 및 swinstall이 제품에 따른 사용자 정의를 수행할 때 소프트웨어의 설치나 복사 작업에서 이루어지는 세 번째 단계

**로컬 호스트 SD-UX** 명령이 실행되는 호스트. 관리 호스트라고도 합니다. **원격 작업**이 가능한 경우 로컬 호스트는 여러 원격 시스템에서 작업을 지시할 수 있는 **컨트롤러**를 실행합니다.

**루트 시스템의 루트 디렉토리(/)**. **루트 디렉토리를 참조하십시오.**

**루트 디렉토리** 선택된 제품의 모든 파일이 설치될 대상 호스트의 디렉토리. 기본값(/)은 다른 시스템에 대한 루트로 작동할 디렉토리로 설치되도록 변경할 수 있습니다. **대체 루트 디렉토리**를 참조하십시오.

ㅁ

**명령줄 사용자 인터페이스(CLI/CLUI)** HP-UX 명령줄 프롬프트에서 입력하거나 스크립트에 의해 실행되는 텍스트 형식의 명령과 옵션. SD-UX에는 또한 sd, swinstall, swcopy, swlist 및 swremove 명령에 대한 **그래픽 사용자 인터페이스(GUI)**와 **터미널 사용자 인터페이스(TUI)**가 있습니다.

**명령줄 옵션** HP-UX 명령줄 프롬프트에서 명령과 함께 입력하는 선택적 매개 변수. **기본 옵션**도 참조하십시오.

**미디어** 테이프, CD-ROM 또는 DVD와 같이 소프트웨어가 저장되는 실제 데이터 저장 미디어

ㅂ

**번들** 특정 목적을 위해 캡슐화된 파일 세트 모음. 번들을 지정함으로써 해당 번들에 포함되는 모든 제품이나 파일 세트는 자동으로 동일한 작업의 대상이 됩니다.

**보안** 소프트웨어 객체에 대한 액세스를 제어하는 작업. SD-UX에서 보안은 객체 및 명령과 관련된 액세스 제어 목록(ACL)과 소프트웨어가 저장된 파일 시스템 권한에 상속된 보안이 결합하여 이루어집니다. **액세스 제어 목록**을 참조하십시오.

**보호된 소프트웨어** 코드워드나 고객 ID(미디어 키트에 동봉된 소프트웨어 인증서에 명시)를 입력하지 않으면 설치하거나 복사할 수 없는 소프트웨어. 코드워드로 보호된 소프트웨어는 해당 소프트웨어에 대한 유효한 사용권을 가지고 있는 시스템에서만 사용할 수 있습니다.

**복수 아키텍처** 같은 파일 세트의 서로 다른 버전이 들어 있는 하나의 제품

**부분 패치** 동일한 기본 파일 세트의 별도의 부분에 적용되는 패치

**분석/분석 단계** 소프트웨어 설치, 복사 또는 제거 작업의 두 번째 단계로서, 이 단계에서 호스트는 일련의 검사를 실행하여 선택된 제품을 호스트에서 설치, 복사, 제거 또는 확인할 수 있는지 판별합니다. 이러한 검사에는 검사 스크립트 실행 및 **디스크 공간 분석(DSA)**이 포함됩니다.

비밀

**비밀 SD-UX** 보안에서 호출자의 호스트 자격을 확인하는 데 사용하는 암호. **SD-UX**는 네트워크상의 모든 **컨트롤러** 및 **대상** 호스트에서 기본 비밀을 제거하고 변경함으로써 호스트를 관리합니다. **공유 비밀 파일**을 참조하십시오.

**빌드 단계** 소스 파일과 정보를 제품으로 패키징하고 제품을 대상 저장소/미디어로 만들거나 병합하는 작업

人

**사용자 이름 RPC**(Remote Procedure Call) 호출을 시작하는 사용자(또는 다른 에이전트에 대한 **RPC**를 호출하는 에이전트의 경우 호스트 시스템)

**상세 목록** 제품, 하위 제품, 파일 세트 또는 파일에 대한 속성을 모두 표시하는 데 사용되는 목록

**상위** 파일 세트의 이전 버전을 지칭하는 **속성**. 대상 시스템에서 파일 세트를 일치시키기 위해 사용됩니다. **match\_target** 옵션을 **true**로 설정하면 **SD-UX**는 새로운 파일 세트 이름을 상위 파일 세트 이름과 일치시킵니다.

**상태** 파일 세트의 현재 상태를 나타내는 속성. 설치 중에 소프트웨어는 **non-existent**, **transient**, **installed** 및 **configured**와 같은 상태 변화를 차례로 거치게 됩니다. 제거 중에는 **configured**, **installed**, **transient** 및 **non-existent**와 같은 상태 변화를 차례로 거칩니다. 작업이 **transient** 상태에서 실패하면 상태는 **corrupt**로 설정됩니다.

**서버** 네트워크의 다른 시스템에 대한 소프트웨어 소스로 작동하는 네트워크상의 시스템

**선택, 선택 단계** 소프트웨어 설치, 복사, 제거 또는 확인 작업의 첫 번째 단계로서 이때 사용자는 호스트에서 설치, 복사 또는 제거될 소프트웨어 제품을 선택합니다.

**설명** 제품과 파일 세트에 대한 속성으로, 일반적으로 해당 제품이나 파일 세트에 대한 단락 형식의 설명

**설치된 제품** 호스트 파일 시스템상의 저장소에 있는 제품과 대조되는 개념으로 최종 사용자가 해당 파일을 사용할 수 있도록 호스트에 설치된 제품. 사용 가능한 제품이라고도 합니다.

**세션/세션 파일 SD-UX** 명령을 매번 호출할 때마다 세션이 정의됩니다. 대부분의 **SD** 명령에서 **-C session\_file** 옵션을 지정하여 명령 옵션, 소스 정보, 소프트웨어 선택 및 호스트 선택 사항을 저장하고 **-S session\_file** 옵션을 지정하여 이 정보를 다시 사용합니다. **GUI** 프로그램에서도 세션 정보를 저장하고 다시 사용할 수 있습니다.

**소유자** 파일의 소유자를 나타내는 속성(문자열)

**소프트웨어 객체 SD-UX**에 의해 패키징, 배포, 설치 또는 관리되는 객체. 소프트웨어 객체는 파일, 파일 세트, 번들 또는 제품일 수 있습니다. 대부분의 작업은 **파일 세트**를 대상으로 수행됩니다.

**소프트웨어 그룹 GUI** 프로그램으로부터 읽히거나 저장되는 소프트웨어 선택 사항 그룹

**소프트웨어 선택 사항** 작업을 위해 사용자가 선택한 소프트웨어 객체 그룹. 이러한 소프트웨어 선택 사항을 차후에 다시 사용하기 위해 저장할 수 있습니다. **소프트웨어 그룹**을 참조하십시오.



**소프트웨어 소스** `swinstall` 또는 `swcopy` 작업의 소스로 사용되는 저장소

**소프트웨어 저장소** 다른 시스템에 설치되거나 다른 저장소로 복사될 수 있는 하나 이상의 소프트웨어 제품이 포함된 **SD-UX 형식**의 구조

**소프트웨어 파일** 이전에 정의된 **소프트웨어 선택 사항** 입력 파일로 명령의 피연산자로 사용됩니다.  
-f *software\_file* 명령줄 옵션을 통해 소프트웨어 파일을 지정합니다.

**속성** 소프트웨어 객체의 특성을 설명하는 정보. 예를 들어 제품 속성에는 버전 번호, 태그(이름) 및 내용(파일 세트 목록)이 포함됩니다. 파일 세트 속성에는 태그, 버전, 커널 및 재부팅 등이 포함됩니다. 파일 속성에는 모드, 소유자 및 그룹이 포함됩니다. **제품 사양 파일** 속성의 핵심적인 부분에는 제품의 축약 이름이나 태그, 한 줄로 구성되는 전체 이름 또는 객체에 대한 한 단락으로 구성되는 설명 등이 포함됩니다. 다른 속성으로는 여러 단락으로 구성되는 **README** 파일, 저작권 정보 표시 등이 있습니다.

**시스템** 독립 실행형인지 다른 컴퓨터에 네트워크로 연결되어 있는지에 관계 없이 컴퓨터를 의미함. **로컬 호스트**를 참조하십시오.

○

**아카이브 파일** `ar` 명령을 통해 기존 아카이브에서 대체되어야 하는 .o 파일. 패치 파일에 사용됩니다.

**액세스 제어 목록(ACL)** 여러 사용자와 그룹에 대한 액세스 권한을 정의하는 소프트웨어 객체에 연결된 구조. **ACL**은 개인이나 그룹 중 하나에만 액세스 권한을

지정하는 대신 여러 개인과 그룹에 액세스 권한을 지정함으로써 **HP-UX** 파일 시스템 모드 비트의 개념을 확장시킵니다.

**업데이트** 시스템에 이미 설치되어 있는 소프트웨어 객체를 덮어쓰고 새로운 객체로 교체하는 작업

**에이전트** 에이전트(`swagent`)는 로컬 호스트에서 실행되며 모든 선택, 분석, 실행 및 상태 요청을 수행합니다. **데몬**에 의해 예약되고 **SD-UX 컨트롤러**의 지시를 따릅니다.

**영역 SD-UX** 보안에서 **주체**가 인증되는 권한의 범위

**요청** 사용자로부터 응답을 얻기 위해 **SD-UX**가 대화식 **request 스크립트**를 실행하는 작업. **request 스크립트**는 `swask`, `swconfig` 및 `swinstall` 명령으로 실행할 수 있습니다.

**원격 작업** 하나의 컨트롤러 시스템에서 원격 시스템상의 작업을 수행하는 것. 원격 작업이 활성화되어 있어야 합니다. (**중앙 집중 관리** 또는 **단일 지점 관리**라고도 합니다.) 자세한 내용은 6장, 167페이지의 “원격 작업 개요”를 참조하십시오.

**원격 프로시저 호출(RPC)** 원격 컴퓨터에서 에이전트와 관련된 작업을 의미합니다.

**원격 호스트 SD-UX** 명령이 실행되는 호스트가 아닌 다른 호스트

**소스 소프트웨어 소스**를 참조하십시오.

**위임 SD-UX**는 저장소에 들어 있는 제품에 대한 액세스를 제어합니다. **에이전트**가 실행 중인 **호스트**와 호출을 시작한 사용자(위임) 모두 읽기 액세스 권한을 가지고 있어야 합니다.

**응답 파일** 대화식 **request 스크립트**에 의해 생성되는 파일로서 사용자 응답이 들어 있습니다.

**입력 파일 SD-UX** 명령의 작동 방식을 변경하고 제어하는 기본값 파일, 옵션 파일, 소프트웨어 선택 파일, 대상 호스트 파일 및 세션 파일

## ㅈ

**작업 swinstall, swcopy, swremove, swverify** 또는 **swconfig** 명령에 의해 작성된 **SD-UX** 작업. **작업 브라우저**를 사용하여 작업을 작성, 모니터, 예약 및 삭제합니다. **swjob** 명령을 사용해서도 작업을 모니터링할 수 있습니다.

**작업 브라우저** 사용자가 작업을 작성, 모니터, 예약 및 삭제할 수 있도록 하는 **GUI** 프로그램. **GUI**는 **sd** 명령에 의해 활성화됩니다. **swjob** 명령을 사용해서도 작업을 모니터링할 수 있습니다.

**작업 ID** 작업을 식별하기 위해 **SD-UX**에 의해 생성된 고유 번호

**저장소 SD-UX가 소프트웨어 소스**로 사용할 수 있도록 구성된 소프트웨어 제품 및 **카탈로그**의 저장소. 저장소의 내용은 하나의 공통 루트를 갖는 디렉토리 구조에 저장됩니다. 저장소는 **SD-UX** 파일 시스템이나 **CD-ROM**상에서 디렉토리 트리로 존재하고 순차 미디어(테이프)에서는 **tar** 아카이브 형태로 존재합니다. 모든 저장소는 저장소가 존재하는 미디어의 종류에 관계없이 하나의 논리적 형식을 공유합니다. 저장소는 로컬 또는 원격 시스템상에 존재할 수 있습니다. 저장소로 소프트웨어를 직접 패키징할 수도 있고 패키징된 소프트웨어를 다른 곳에서 저장소로 복사할 수도 있습니다.

**저장소 소스 저장소**를 참조하십시오.

**전제 조건** 하나의 파일 세트를 설치하거나 구성하기 위해 다른 파일 세트를 먼저 설치하거나 구성해야 하는 **종속성**. 예를 들어 파일 세트 **A**를 설치하기 위해서는 먼저 파일 세트 **B**를 설치해야 합니다. 따라서 파일 세트 **B**는 파일 세트 **A**의 전제 조건입니다. **종속성, 필수 조건 및 제외 조건**을 참조하십시오.

**제목** 제목을 갖는 제품을 식별하는 한 줄로 구성된 전체 이름 속성

**제어 스크립트 IPD**를 변경하여 소프트웨어와 함께 패키징되거나 소프트웨어에 추가되는 선택적 스크립트. 제어 스크립트는 **swconfig, swinstall, swremove** 또는 **swverify** 작업 중에 실행됩니다. 제어 스크립트에는 **swconfig** 명령을 위한 **configure** 또는 **unconfigure** 스크립트, **swinstall** 명령을 위한 **checkinstall, preinstall, postinstall** 및 **configure** 스크립트, **swremove** 명령을 위한 **checkremove, unconfigure, preremove** 및 **postremove** 스크립트 그리고 **swverify** 명령을 위한 **fix** 또는 **verify** 스크립트 등이 있습니다.

**제외 조건** 하나의 파일 세트를 설치하거나 구성하기 위해 다른 파일 세트가 없어야 하는 **종속성**. 예를 들어 파일 세트 **A**는 파일 세트 **B**가 이미 설치되어 있는 경우 설치하거나 구성할 수 없다면 파일 세트 **B**는 파일 세트 **A**의 제외 조건입니다.

**제품** 하위 제품 및/또는 파일 세트의 집합

**제품 디렉토리** 대부분의 파일이 들어 있는 제품 객체의 루트 디렉토리. **검색 가능 제품**을 설치할 때 기본 제품 디렉토리를 변경(재배치)할 수 있습니다.

**제품 버전** 저장소에는 한 제품의 여러 버전이 들어 있을 수 있습니다. 제품의 여러 버전들은 동일한 태그 속성을 갖지만 버전 속성은 서로 다릅니다. **IPD(installed products database)**는 설치된 여러 제품 버전을 지원합니다. 설치된 버전은 동일한 태그 속성을 갖지만 버전 속성이나 제품 디렉토리는 서로 다릅니다.

**제품 사양 파일(PSF) swpackage**에 의해 패키징되는 파일의 구조와 속성을 정의하는 입력 파일

**제품 ACL 템플릿 SD-UX** 보안에서 호스트가 만든 저장소의 새로운 제품을 보호하는 ACL을 초기화하는 데 사용되는 ACL

**종속성** 특정 방식으로 하나의 파일 세트가 다른 파일 세트를 필요로 하는 파일 세트간의 관계. 예를 들어 파일 세트 A를 설치하기 전에 먼저 파일 세트 B를 설치해야 하는 경우가 있습니다. **SD-UX는 필수 조건, 제외 조건 및 전제 조건** 종속성을 지원합니다. 종속성을 참조하십시오.

**종속 파일 세트** 다른 파일 세트에 대한 **종속성**이 있는 파일 세트. 예를 들어 파일 세트 A가 파일 세트 B에 의존한다면 B는 A에 대한 종속 파일 세트이거나 종속성을 갖습니다.

**주체 SD-UX** 보안에서 다른 시스템에 대한 호출을 시작하는 사용자(또는 RPC를 수행하는 에이전트의 경우 호스트 시스템)

**중앙 집중 관리 원격 작업**을 참조하십시오.

## ㅌ

**체크섬 CRC(Cyclic Redundancy Check)**. 파일이 전송 중에 손상되었는지 알 수 있도록 저장된 데이터와 비교되는 계산된 값

## ㅋ

**카탈로그/카탈로그 디렉토리 SD-UX**가 저장소에 저장되어 있는 제품의 구성과 내용을 정의하는 데 필요한 모든 정보가 들어 있는 저장소 내의 한 영역. 여기에는 글로벌 **INDEX** 파일과 저장소의 각 제품 버전에 대한 정보 디렉토리가 포함됩니다. 카탈로그 디렉토리라고도 합니다.

**캐시 파일 swinstall** 또는 **swcopy**에 의해 선택한 대상의 이름과 속성이 들어 있는 파일

**커널 파일 세트** 운영 체제 커널을 생성하는 데 사용되는 파일이 들어 있는 파일 세트. 커널 파일 세트는 **swinstall** 로드 단계에서 로드되고 다른 파일 세트보다 먼저 사용자 정의됩니다.

**컨테이너 ACL 템플릿** 저장소와 루트에 대한 초기 ACL을 작성하는 데 사용되는 특수 ACL (global\_soc\_template). **제품 ACL 템플릿**도 참조하십시오.

**컨트롤러 로컬 호스트**에서 사용자에게 의해 호출되어 **SD-UX 에이전트**가 수행할 작업을 지시하는 **SD-UX 프로그램**이나 명령(**swinstall**, **swcopy** 등)

코드워드 참조 소프트웨어 무단 사용을 방지하기 위해 HP 및 기타 업체는 특별한 코드워드와 고객 식별 번호를 통해 소프트웨어를 특정 소유자만 사용할 수 있도록 잠급니다. 이러한 코드워드와 고객 ID는 소프트웨어를 구입하거나 업데이트 소프트웨어를 받을 때 제공됩니다.

클라이언트 일반적으로 디스크가 없는 서버 컴퓨터를 말함. 이전 버전의 SD-UX에서는 디스크가 없는 클라이언트가 지원되었습니다.

키워드 패키징할 때 사용자가 패키징하는 소프트웨어 객체의 구조나 내용에 대해 `swpackage`에게 알려주는 단어(또는 명령). 패키징 정보는 *PSF(Product Specification File)*를 통해 `swpackage`에 입력됩니다.

## ≡

테이프 미디어 `tar`를 사용하여 SD-UX 소프트웨어 제품 및 제어 파일을 저장하는 소프트웨어 미디어. `tar` 아카이브를 포함할 수 있는 일반 파일도 가능하지만 일반적으로는 DDS, 카트리지, 9 트랙 또는 기타 테이프와 같은 순차 미디어상에 존재합니다. `tar` 아카이브 내에서 디렉토리와 파일 항목은 다른 저장소와 동일한 구조를 사용하여 구성됩니다.

테이프 소스 테이프 저장소를 참조하십시오.

테이프 작성 단계 배포 테이프 소프트웨어를 패키징할 때 이 단계에서 실제로 임시 저장소의 내용을 테이프로 복사합니다.

테이프 저장소 `tar`(테이프 아카이브) 형식으로 저장된 소프트웨어 저장소. 테이프 저장소 내에서 아카이브, 디렉토리 및 파일 항목은 다른 SD-UX 형식 저장소와 동일한 구조를 사용하여 구성됩니다. 카트리지

테이프, DAT 및 9 트랙 테이프와 같은 테이프 저장소는 테이프 드라이브의 장치 파일에 대한 파일 시스템 경로로 참조됩니다.

## 표

파일 세트 파일 집합. 대부분의 SD-UX 작업은 파일 세트에 대해 수행됩니다.

패치 시스템에서 특정 번들, 제품, 하위 제품, 파일 세트 또는 파일을 업데이트하기 위해 설계된 소프트웨어. 부분 패치와 대체(누적) 패치가 있습니다. 정의에 의하면 패치 소프트웨어는 `is_patch` 속성을 `true`로 설정하여 패키징됩니다.

패키지 빌드 단계 `swpackage`가 소스 파일과 정보를 제품 객체로 빌드한 다음 그 제품을 기존의 저장소에 삽입하는 단계. 저장소가 없으면 `swpackage`가 저장소를 새로 만들지만 등록하지는 않습니다.

패키지 선택 단계 패키징할 때 제품, 하위 제품 및 파일 세트 구조, 각 파일 세트에 들어 있는 파일 및 이들 객체와 관련된 속성을 판별하기 위해 `product_specification_file`을 읽는 단계

패키지 `swpackage`로 작성된 설치 가능한 SD-UX 형식의 소프트웨어. 패키징된 소프트웨어는 배포를 위해 저장소에 배치할 수 있습니다.

패키징 패키지를 작성하는 작업

패키징 프로그램 차후 대상 시스템으로의 배포를 위해 소프트웨어를 패키징하는 `swpackage` 프로그램

푸시(Push) 중앙의 컨트롤러에서 여러 원격 대상 시스템상의 소프트웨어 관리 작업(주로 설치나 복사)을 수행하는 것. 원격 작업을 참조하십시오.

**풀(Pull)** 로컬 시스템에 설치하거나 복사하기 위해 **저장소**에서 소프트웨어 제품을 가져오는 작업. **푸시(Push)**도 참조하십시오.

**필수 조건** 특정 파일 세트 설치 시 다른 파일 세트를 동시에 설치하거나 구성해야 하는 **종속성**. 예를 들어 파일 세트 A를 설치하기 위해 파일 세트 B를 동시에 설치해야 한다면 파일 세트 B는 필수 조건입니다.

## ㅎ

**하위 제품** 많은 수의 파일 세트를 가지고 있는 제품을 나누거나 파일 세트에 대한 서로 다른 사용자 보기를 제공하기 위해 선택적으로 파일 세트를 그룹화한 것

**핵심 파일 세트** 호스트가 올바르게 작동하는 데 핵심적인 소프트웨어를 가지고 있는 파일 세트. 핵심 파일 세트는 재부팅 및/또는 커널 파일 세트 플래그가 있는 파일 세트입니다. **swinstall**의 로드 단계에서 핵심 파일 세트가 로드되어 다른 파일 세트에 앞서 사용자 정의됩니다.

**호스트 SD-UX** 작업이 수행되는 컴퓨터 시스템. **로컬 호스트** 및 **컨트롤러**를 참조하십시오.

**호스트 ACL** 호스트 객체에 연결되어 이에 대한 액세스를 제어하는 ACL

**호환되는 소프트웨어** 해당 하드웨어 시스템에서 작동되는 소프트웨어 제품. 로컬 호스트에 대한 **호환성 필터링**을 통과한 소프트웨어입니다. **호환되지 않는 소프트웨어**도 참조하십시오.

**호환되지 않는 소프트웨어** 소프트웨어 제품은 특정 컴퓨터 하드웨어와 운영 체제에서 실행되도록 만들어집니다. 같은 제품의 여러 버전이 존재할 수 있

며 각각은 서로 다른 하드웨어와 운영 체제상에서 실행됩니다. 호환되지 않는 소프트웨어는 호스트의 컴퓨터 하드웨어나 운영 체제로 인해 호스트에서 작동하지 않습니다. **swinstall**의 기본 조건은 호환되지 않는 소프트웨어의 선택 및 설치를 허용하지 않는 것입니다.

**호환성 필터링** 호스트의 **uname** 속성에 따라 소스에서 사용 가능한 소프트웨어를 필터링하는 **swinstall**의 기능. 소프트웨어 제품은 특정 컴퓨터 하드웨어와 운영 체제에서 실행되도록 만들어집니다. 같은 제품의 여러 버전이 존재할 수 있는데 이들 각각은 서로 다른 컴퓨터 하드웨어 및 운영 체제에서 실행됩니다. 기본적으로 **swinstall**에서는 호환되지 않는 소프트웨어를 선택하거나 설치할 수 없습니다.

## A

**Applied** 패치가 설치된 상태. 패치가 설치되면 기본적으로 **patch\_state**가 **Applied** 상태가 됩니다. 다른 패치 상태로는 **committed**, **superseded** 및 **committed/superseded** 등이 있습니다.

**Architecture** 제품이 실행되는 운영 체제 플랫폼을 나타내는 키워드

## C

**Category** 제품 객체에 대한 "범주" 속성을 정의하는 키워드. 패키징되는 소프트웨어의 종류를 나타냅니다.

**CD-ROM** CD-ROM(Compact Disc-Read Only Memory) 또는 CD-ROM 상에 있는 SD-UX 저장소

## Checkinstall 스크립트

**Checkinstall 스크립트** 제품이나 파일 세트와 관련된 선택적 스크립트. 분석 단계에서 **swinstall**에 의해 실행됩니다. 스크립트가 반환하는 결과를 통해 파일 세트가 설치되거나 업데이트될 수 있는지 판별할 수 있습니다.

**Checkremove 스크립트** 파일 세트와 관련된 선택적 스크립트. **swremove** 분석 단계에서 실행됩니다. 스크립트가 반환하는 결과를 통해 파일 세트가 제거될 수 있는지 판별할 수 있습니다.

**CLI 명령줄 인터페이스**. **명령줄 사용자 인터페이스**를 참조하십시오.

**CLUI 명령줄 사용자 인터페이스**를 참조하십시오. 모든 SD-UX 명령은 명령줄에서 실행할 수 있습니다. **GUI**, **TUI** 및 **IUI**도 참조하십시오.

**Committed** 패치가 적용되고 롤백 파일이 삭제된 상태. 다른 패치 상태로는 *applied*, *superseded* 및 *committed/superseded* 등이 있습니다.

**committed/superseded** 패치가 *committed*되고 *superseded*된 상태

**Configure 스크립트** 파일 세트와 관련된 선택적 스크립트. 파일 세트 설치가 완료된 후 **swinstall**에 의해 자동으로 실행되거나 **swconfig**에 의해 수동으로 실행됩니다.

**Contents** 파일 세트를 하위 제품에 지정하는 데 사용하는 키워드. 이를 통해 하나의 파일 세트가 여러 하위 제품에 포함될 수 있습니다.

**Copyright swpackage**에 의해 작성/변경되는 대상 저장소(미디어)에 대한 저작권 속성을 정의하는 키워드. 소프트웨어 제품에 대한 저작권 정보를 나타냅니다.

## D

**data\_model\_revision SD-UX INDEX** 파일 구문에 대한 내부 속성. **layout\_version 0.8**은 **data\_model\_revision 2.10**을 사용하는 반면 **Layout\_version 1.0**은 **data\_model\_revision 2.40**을 사용합니다.

**DCE** 분산 컴퓨팅 환경(Distributed Computing Environment). 분산 통신을 위해 SD-UX가 사용하는 기술. **컨트롤러**, **데몬** 및 **에이전트**는 **DCE RPC(Remote Procedure Call)**를 사용하여 통신합니다.

**Details 대화 상자 GUI** 또는 **TUI**에서 진행 상황을 모니터링하기 위해 특정 프로세스에 대한 더 자세한 정보를 확인할 수 있는 대화 상자

**Directory** 패키징할 때 제품 객체에 대한 디렉토리를 정의하는 키워드. 지정된 디렉토리는 제품이 설치될 디렉토리에 대한 기본적인 절대 경로 이름입니다.

**DSA 디스크 공간 분석**을 참조하십시오.

## E

**End PSF**에서 소프트웨어 객체 사양을 종료하는 선택적 키워드. 값은 필요하지 않습니다.

## G

**GUI 그래픽 사용자 인터페이스**를 참조하십시오.

**H**

**HOME** 현재 사용자의 로컬 로그인 디렉토리 경로가 들어 있는 변수

**I**

**INDEX/INDEX 파일** 패키징할 때 **INDEX** 파일은 객체(예: 저장소, 제품 또는 파일 세트)에 대한 속성 및 구성 정보를 정의합니다. **INDEX** 파일은 저장소 카탈로그와 **IPD(Installed Products Database)**에 존재하며 그 내용을 설명합니다.

**INFO INFO** 파일은 파일 세트 안에 들어 있는 파일에 대한 정보를 제공합니다. 이 정보에는 유형, 모드, 소유권, 체크섬, 크기 및 경로 이름 속성이 포함됩니다. **INFO** 파일은 저장소 카탈로그와 **IPD(Installed Products Database)**에 존재하며 각 기존 파일 세트에 들어 있는 파일을 설명합니다.

**install-sd** 미디어나 저장소에서 워크스테이션이나 서버로 **SD-UX** 제품을 설치할 수 있도록 하는 명령. 사용자 시스템상의 버전이 손상되었거나 삭제된 경우 **SD-UX**를 다시 설치해야 합니다. 이 명령은 **update-ux**와 함께 이전의 **swgettools** 명령을 대신합니다.

**Instance\_ID** 동일한 태그(이름)나 버전을 갖는 제품을 고유하게 식별하는 **IPD(Installed Products Database)**의 제품 속성

**IPD(Installed Products Database)** 해당 호스트상에(또는 대체 루트 안에) 설치된 제품을 설명합니다. 설치된 제품 정보는 **swinstall**에 의해 작성되고 **swmodify**에 의해 관리됩니다. **IPD**의 내용은 하나의 공통 루트를 갖는 디렉토리 구조에 저장됩니다.

**IPD 참조 IPD(Installed Products Database)**를 참조하십시오.

**Is\_Locatable** 패키징할 때 제품이 대체 제품 디렉토리에 설치될 수 있는지 여부를 정의하는 키워드. 이 키워드를 지정하면 속성은 **true** 값으로 설정됩니다. 지정하지 않으면 속성은 **false** 값으로 설정됩니다.

**IUI** 대화식 사용자 인터페이스로 **그래픽 사용자 인터페이스(GUI)**나 **터미널 사용자 인터페이스(TUI)**를 의미하는 일반적인 용어

**M**

**Machine\_Type** 패키징할 때 제품이 실행될 시스템 유형을 나타내는 키워드. 이 키워드를 지정하지 않으면 키워드에는 와일드카드 값인 **\***(모든 시스템에서 실행)가 지정됩니다. 여러 플랫폼의 경우 각 시스템을 | (세로줄)로 구분하여 지정해야 합니다.

**Minfree** 최소 여유 임계값을 나타내는 용어로, 패키징되는 제품을 저장하는 데 필요한 최소한의 여유 디스크 공간의 용량입니다.

**multi\_stream** 복수 아키텍처를 참조하십시오.

**N**

**Number** 패키징할 때 배포 미디어(CD나 테이프 저장소)의 부품 또는 제조 번호를 정의하는 키워드

**O**

**OS** 운영 체제

## P

**POSIX SD-UX**의 기반이 되는 **POSIX 1387.2-1995 IEEE** 표준

**Postinstall 스크립트** 파일 세트와 관련된 선택적 스크립트. 해당 파일 세트가 설치되거나 업데이트된 다음 **swinstall**에 의해 실행됩니다.

**Postremove 스크립트** 파일 세트와 관련된 선택적 스크립트. 해당 파일 세트가 제거된 다음 **swremove**에 의해 실행됩니다.

**Preinstall 스크립트** 파일 세트와 관련된 선택적 스크립트. 해당 파일 세트를 설치하거나 업데이트하기 전에 **swinstall**에 의해 실행됩니다.

**Preremove 스크립트** 파일 세트와 관련된 선택적 스크립트. 해당 파일 세트를 제거하기 전에 **swremove**에 의해 실행됩니다.

**PSF 참조 제품 사양 파일을 참조하십시오.**

## R

**Readme** 제품 객체에 대한 "readme" 속성을 정의하는 키워드. 제품에 대한 **README** 정보의 텍스트 파일로 텍스트 값 자체이거나 텍스트를 포함하는 파일 이름입니다.

**request 스크립트** 사용자로부터 응답을 받는 대화식 제어 스크립트. **request** 스크립트는 사용자의 응답을 기다리고 사용자의 응답을 읽어서 그 결과를 **응답 파일**에 저장합니다. **request** 스크립트는 **swask**, **swconfig** 및 **swinstall** 명령으로 실행할 수 있습니다.

**Revision** 제품 객체에 대한 "버전" 속성을 정의하는 키워드. 제품에 대한 버전 정보(릴리즈 번호, 버전)를 나타냅니다.

**RPC** 원격 프로시저 호출. 분산 통신 및 데이터 전송을 위한 **DCE** 기술

## S

**sd** 사용자가 작업을 작성, 모니터, 예약 및 삭제할 수 있도록 하는 **GUI** 프로그램인 **작업 브라우저**를 호출하는 명령. **swjob** 명령을 통해 명령줄에서 작업을 모니터링할 수 있습니다. 또한 **swjob -i** 명령을 사용하여 작업 브라우저를 활성화할 수도 있습니다.

**SD 형식 SD-UX 형식**을 참조하십시오.

**SD-UX 형식** 저장소에 있는 **SD-UX** 소프트웨어의 형식과 구분

**ServiceControl Manager(SCM)** 많은 시스템 관리 기능의 중앙 집중 관리를 허용하는 **HP** 프로그램. **SCM**에서 **SD-UX**를 실행할 수 있습니다.

**Software Selection 창** 설치, 복사 또는 제거하려는 소프트웨어 파일을 선택할 수 있는 **GUI** 창

**SPA 단일 지점 관리**를 참조하십시오.

**Superseded** 패치가 적용된 다음 **대체 패치**에 의해 대체된 상태. 다른 패치 상태로는 **applied**, **committed** 등이 있습니다.



**SW-DIST** 모든 SD-UX 기능을 제공하는 소프트웨어 제품. SW-DIST는 HP-UX 11i 미디어에 포함되어 있습니다. SW-DIST가 시스템에서 손상되거나 유실된 경우 새로운 SW-DIST 제품을 포함하여 **SD-UX 형식**으로 패키징된 모든 HP-UX 소프트웨어를 설치하거나 복사할 수 없습니다. *install-sd* 명령을 사용하여 SD-UX를 다시 설치할 수 있습니다.

**swacl** 소프트웨어 보안을 제공하는 액세스 제어 목록 권한을 변경할 수 있도록 해주는 SD-UX 명령

**swadm** SD-UX 보안에서 기본 사용자 식별 그룹

**swagent** 저장소와 루트를 변경하는 SD-UX 에이전트 프로그램. **컨트롤러**의 지시를 받고 **데몬**에 의해 예약됩니다.

**swagentd** 컨트롤러와 에이전트 간의 통신 시작, 원격 **호스트**상의 여러 **에이전트**가 요청하는 하나 이상의 **저장소** 처리 등을 포함하여 다양한 서비스를 제공하는 SD-UX 데몬

**swask** 사용자로부터의 응답을 받기 위해 대화식 **request 스크립트**를 실행하도록 하는 SD-UX 명령. Request 스크립트는 *swconfig*와 *swinstall* 명령으로도 실행할 수 있습니다.

**swconfig** 이전에 설치된 소프트웨어를 구성하고 해당 소프트웨어를 사용할 수 있도록 준비하는 SD-UX 명령

**swcopy** 소프트웨어 소스로부터 저장소로 또는 한 저장소에서 다른 저장소로 소프트웨어를 복사하는 SD-UX 명령. *swcopy* 명령은 기존 저장소에 제품을 추가하고 이미 저장소에 있는 제품을 바꾸거나 새로운 저장소를 만듭니다.

**swgettools** 미디어에서 새로운 SW-DIST 제품을 설치하기 위해 이전 HP-UX 릴리즈에서 사용되었던 SD-UX 명령. 이 명령은 *install-sd* 및 *update-ux*로 대체되었습니다.

**swinstall** 소프트웨어를 설치하는 SD-UX 명령. *swinstall*은 소프트웨어 구성도 수행합니다.

**swlist** 소프트웨어 객체, 속성 및 구성을 나열하는 SD-UX 명령. 설치된 소프트웨어와 저장소 안에 들어 있는 소프트웨어를 모두 나열합니다.

**swlock** 소프트웨어 객체 및 ACL에 대한 읽기 또는 쓰기 액세스를 가지고 있는 파일

**swmodify** *IPD(installed products database)* 또는 **저장소 카탈로그** 파일의 정보를 변경하도록 해주는 SD-UX 명령

**swpackage** **제품 사양 파일(PSF)**을 사용하여 소프트웨어 제품을 구성하고 이를 **저장소**로 **패키징**하는 SD-UX 명령. 저장소는 SD-UX가 직접 액세스하거나 CD-ROM이나 테이프에 마스터할 수 있습니다.

**swreg** 저장소를 **등록** 또는 **등록 해제**하는 데 사용하는 SD-UX 명령

**swremove** 이전에 설치된 소프트웨어를 제거하거나 저장소에서 패키징된 소프트웨어를 제거하는 SD-UX 명령

**swverify** 설치된 소프트웨어나 저장소 소프트웨어가 올바르게 완전한지 확인하는 SD-UX 명령

## T

**Tag** 패키징할 때 대상 저장소(미디어)에 대한 배포 태그나 소프트웨어 객체의 이름 속성을 정의하는 키워드

**TUI** 터미널 사용자 인터페이스. **ASCII** 터미널에서 작동하는 창과 풀다운 메뉴로 구성되는 문자 기반 디스플레이. **TUI**는 키보드를 사용하여 탐색합니다(마우스를 사용하지 않음). **명령줄 사용자 인터페이스** 및 **그래픽 사용자 인터페이스**도 참조하십시오.

**TUI** 터미널 사용자 인터페이스를 참조하십시오.

## U

**UDP/IP** 사용자 데이터그램 프로토콜. **TCP/IP**와 종종 비교되며, 연결 성능은 조금 떨어지지만 보다 안정적인 네트워크 환경(**LAN**)에서 사용되도록 설계되었습니다.

**Uname 속성** 소프트웨어 관리 작업을 위해 대상 시스템에 연결할 때 시스템의 네 가지 **uname** 속성(운영 체제 이름, 릴리즈, 버전 및 하드웨어 시스템 종류)을 얻을 수 있습니다. 해당 호스트와의 소프트웨어 호환성을 판별하는 데 사용됩니다.

**Unconfigure 스크립트** **configure** 스크립트에 의해 이루어진 구성을 취소하는 선택적 스크립트. **unconfigure** 스크립트는 파일 세트와 관련되며 파일 세트 제거가 시작되기 전에 **swremove**에 의해 자동으로 실행됩니다. **swconfig**를 통해서도 **unconfigure** 스크립트를 실행할 수 있습니다.

**update-ux** **HP-UX** 업데이트 프로세스의 일부를 자동화하는 명령. **SD-UX** 이전 버전에서 사용된 **swgettools** 스크립트를 대신합니다. **install-sd**는 **OS**는 업데이트하지 않고 **SD-UX** 제품을 업데이트합니다.

**UUID** 패키징할 때 공급업체 객체에 대한 키워드. **NetLS** 공급업체 및 동일한 **vendor\_tag**를 갖는 두 공급업체 중에서 제품을 선택하려는 사용자들에게 유용합니다.

## V

**Vendor\_tag** 해당 객체에 일치하는 **태그** 속성이 있는 경우 제품이나 번들을 마지막으로 정의된 공급업체 객체와 연관시킵니다.

## 기호

\*systemFont, 55  
 \*userFont, 55  
 /(루트 디렉토리), 66  
 @("at") 기호, 60

## ㄱ

가져오기(pull) 배포, 보안, 274  
 개발자, 보안, 275  
 개요, 명령, 27  
 객체 권한, 254  
 객체 목록, 39  
 객체, 소프트웨어, 30  
 검사점 지정, 221  
 고객 식별 번호, 413  
 고객 ID, 사용, 80, 109, 152  
 공간 파일, 353  
 공급업체  
 사양, 301  
 정의 속성, 299  
 키워드, 301  
 공급업체 사양, 301  
 공유 가능 파일, 281  
 공유 비밀 파일, 267  
 관리  
 여러 버전, 82  
 관리 책임, 지정, 274  
 구문  
 install-sd, 459  
 swask, 385  
 swcopy, 149  
 swinstall, 75  
 swjob, 207  
 swlist, 100  
 swmodify, 119  
 swremove, 130  
 구성  
 단계, 85  
 샘플, 90  
 구성 단계, 87  
 구성 클린업, 411  
 구성 해제  
 제거된 소프트웨어, 124  
 구조  
 소프트웨어, 281  
 제품 결정, 281  
 구조, 소프트웨어 제품, 30  
 권한 비트, 263  
 권한 사양, 기본, 317

권한이 부여된 기능, 263  
 권한, 저장소, 154  
 권한, RPC, 269  
 그래픽 사용자 인터페이스(GUI), 27, 37, 67, 125, 140  
 swlist, 98, 188  
 그룹  
 액세스, 265  
 글꼴  
 가변 너비, 56  
 고정 너비, 56  
 기능, swpackage, 330  
 기본 비밀 파일, 대체, 447  
 기본 옵션, 목록, 95, 104, 122, 132, 151, 156, 237, 332, 387  
 기본 템플릿 ACL 항목, 261  
 기본값  
 값, 변경(swask), 387  
 값, 변경(swconfig), 89, 332  
 값, 변경(swcopy), 150  
 값, 변경(swinstall), 76, 208  
 값, 변경(swlist), 104  
 값, 변경(swmodify), 122  
 값, 변경(swreg), 156  
 값, 변경(swremove), 132, 237  
 값, 변경(swverify), 95  
 옵션 값, 61, 400  
 우선권, 61, 400  
 정책 설정, 61, 400  
 swlist, 107  
 기본값, 변경, 61, 400  
 기본값 수정, 61, 400

## ㄴ

나열  
 대화식 swlist, 98, 188  
 소프트웨어, 98  
 내부 권한 인증, SD, 265  
 너무 제한적인 권한, 336  
 네트워크  
 문제, 449  
 오류, 450  
 프로토콜, 451  
 네트워크 서버, 137  
 네트워크 소스, 137  
 네트워크 연결 요구 사항, 24  
 네트워크 요구 사항, 24  
 네트워크 저장소, 만들기, 159  
 노드, 정의, 29

# 색인

## ㄷ

다시 부팅, 시스템, 80  
다중 사용자 모드, 24  
단계, 224  
단계적 설치, 224  
단일 대상 설치, 179  
단일 사용자 모드, 24  
대상  
구문, 60  
변경, 126  
선택 사항, 60  
선택, swmodify, 119  
원격, 168  
원격 작업에 대한 정의, 168  
작업 브라우저에서 재사용, 205  
정의, 29  
파일, 61  
대상 그룹, 176, 216  
대상 그룹 수정, 175, 176  
대상 그룹 추가, 176  
대상 디렉토리, 413  
대상 변경, 126  
대상 선택  
원격 작업, 173, 180  
대상 호스트 추가, 246  
대체 루트  
디렉토리, 66  
설치, 82  
소프트웨어 제거, 134  
옵션 -r, 75  
대체 소스, 사용, 224  
대화식 옵션, -i, 75, 98, 149, 188, 207  
데몬, 30  
다시 시작, 62  
재시작, 401  
데몬 다시 시작하기, 401  
데몬 로그 파일, 453  
데몬/에이전트, 168  
권한, 266  
도움말  
메뉴, 54  
온라인, 54  
f1 키, 54  
두 번 클릭, 50  
등록되지 않은 저장소, 154, 271  
디렉토리  
/var/tmp, 458  
디렉토리 구조, 281  
디렉토리 매핑, 318  
디렉토리 저장소, 137

디스크 공간  
공간 파일, 353  
단추, 71, 73, 145, 147  
분석 대화 상자, 72  
요구 사항 지정, 353  
장애, 73, 147  
swpackage에 의한 분석, 327  
디스크 공간 분석, 452  
디스크 없는 클러스터, 24

## ㄹ

로그 파일 메시지, 420  
로그 파일, swpackage, 333  
로컬 슈퍼유저, 271  
로컬 호스트  
정의, 29  
롤백, 79  
루트  
디렉토리, 66  
루트 ACL  
사용 권한, 257  
제어, 254

## ㅁ

마스터 및 중간 저장소, 이들 간의 일관성, 226  
마스터 저장소, 224  
마우스, 클릭, 39  
맨페이지, 28  
메뉴 표시줄, 39  
메뉴, 폴다운, 39  
명령  
개요, 27  
설명, 27  
SD-UX, 28  
명령 옵션  
변경, 61, 400  
알파벳 목록, 402  
우선권, 400  
작업 관련, 211  
편집기, 48  
명령줄, 실행, 57  
모드 비트, 269  
목록  
간단, 105  
다른 명령에 대한 입력으로, 98  
상세, 113  
저장소, 113  
목차, 98  
문제점 해결, 439

미리 보기, 427  
미리 보기 옵션, -p, 75, 88, 130, 149

## ㅂ

배포 디렉토리, 414  
배포 저장소 사양, 299  
배포 테이프, 423  
배포 테이프, 만들기, 330  
배포 테이프 형식, 329  
버전, 230  
버전 상태, 92  
버전 속성, 416  
번들, 30, 281  
변경  
  기본 옵션, -x, 75, 88, 94, 103, 119, 130, 149, 154, 207, 330  
  명령 옵션, 61, 400  
  소스, 69  
  소스 대화 상자, 69  
  IPD 또는 카탈로그 파일, 117  
보기 기본 설정, 변경, 44  
보기 정보 저장, 48  
보안  
  "가져오기" 설치, 273  
  가져오기 배포, 274  
  개발자, 275  
  기본값, 234  
  액세스 거부, 446  
  저장소, 154  
  패키징, 337  
  UNIX, 263  
보안 작업, 239  
보안 점검  
  구성, 278  
  나열, 276  
  복사, 276  
  설치, 277  
  작업 브라우저, 276  
  저장소 등록, 278  
  제거, 277  
  패키징, 276  
  확인, 278  
보호되는 소프트웨어, 36  
보호된 소프트웨어, 설치 예, 81  
복구, 227  
복사  
  대화 상자, 148  
복사/설치 다시 시작, 74  
복사/설치 중단, 74

복사할 소프트웨어 선택, 143  
분석  
  제거, 127  
  진행 및 결과, swremove, 128  
비권한 SD, 391  
  개요, 392  
  설정, 394  
  제한 사항, 393  
  패키징 요구 사항, 394  
비밀  
  기본값, 267  
  보안, 267  
  일치, 447  
  호스트간, 447

## ㅅ

사용 가능  
  직접 액세스, 153, 156  
사용 중인 파일, swremove, 124  
사용자  
  액세스, 265  
  ACL 일치, 250  
  사용자 정의 목록, 108  
  사용자에게 ACL 일치, 250  
  사전 지정된 선택 사항, 75, 149, 207  
  삽입 권한, 154  
상세  
  목록, 113  
  목록, 샘플, 116  
  옵션, -v, 75, 83, 94, 130, 149, 154, 207  
상세 옵션, 330  
새로 고침 간격, 작업 브라우저, 201  
샘플  
  복사, 152  
  설치, 78  
서버, 정의, 29  
선택할 소프트웨어 표시, 69, 143  
설명 파일, 69, 126, 143  
설명서  
  매펀페이지, 28  
설치  
  단계적, 224  
  대화 상자, 74, 129  
  복구, 227  
  분석, 69, 126, 143  
  작업 브라우저의 아이콘, 196  
  장애, 454  
  재시도, 218, 220  
  호환성 필터링, 82

# 색인

- PC, 단계, 224
- UNIX, 단계, 224
- UNIX, 대체 소스 사용, 224
- UNIX, 별도의 구성으로, 229
- 설치 기본 설정, 184
- 설치 제한, 274
- 설치 후 구성, 229
- 성능, 214
- 세션
  - 파일, 63
  - 파일, 예제, 64
- 세션 파일 옵션, -S, 75, 88, 94, 103, 119, 130, 149, 154, 207, 330, 385
- 세션 파일 저장 옵션, -C, 75, 88, 94, 101, 119, 130, 149, 154, 207, 330, 385
- 셸, 제어 스크립트, 356
- 소스, 434
  - 네트워크, 137
  - 작업 브라우저에서 재사용, 205
  - 저장소 경로, 52, 68, 142
  - 추가, 52, 68, 142
  - 호스트 이름, 52, 68, 142
- 소스 옵션, 216
- 소스 추가, 52, 68, 142
- 소프트웨어
  - 객체, 30
  - 선택 파일, 60, 61
  - 선택 파일 옵션, -f, 75, 88, 94, 101, 103, 119, 130, 149, 154, 207, 330, 385
  - 소스 옵션, -s, 75, 103, 149, 385
  - 종속성, 35, 84, 91
- 소프트웨어 그룹, 216
  - 저장, 126
  - 추가, 126
- 소프트웨어 그룹 저장, 126
- 소프트웨어 그룹 추가, 126
- 소프트웨어 나열
  - 등록된 저장소, 160
  - UNIX 저장소 내용, 161
- 소프트웨어 보기, 433
- 소프트웨어 보기 기본값, 433
- 소프트웨어 복사
  - 작업 브라우저, 205
  - 작업 브라우저의 아이콘, 196
  - 저장소, 140
- 소프트웨어 선택
  - 원격 작업, 182
- 소프트웨어 설명 표시, 69, 126, 143
- 소프트웨어 설치, 402
- 소프트웨어 수준, 419
  - 소프트웨어 인증서, 36
  - 소프트웨어 제거
    - 작업 브라우저, 205
    - 작업 브라우저의 아이콘, 198
  - 소프트웨어 호환성, 402
  - 소프트웨어를 다시 패키징, 340
  - 속성
    - 샘플, 114
    - 정의, 98
  - 속성 나열, 424
  - 수준
    - 상세, swlist, 98
    - 지정, swlist, 101
  - 수퍼유저
    - 권한, 263, 271
    - ACL 액세스, 250
    - swpackage, 337
  - 스크립트
    - 인터프리터, 354
    - fix, 349
    - request, 349, 357, 372
  - 스크립트, 기타, 349
  - 시간 제한, 432
    - 문제 해결, 450
    - 연결, 450
    - 옵션, 218
  - 시스템
    - 정의, 29
  - 신뢰 입증, 267
  - 실행 수준, 24
  - 실행 수준 요구 사항, 24
  - 심볼릭 링크, 416
    - 값, swverify, 92
    - swremove, 124
  - 쓰기 가능 저장소, 137
  - 쓰기 권한, 154
- - 아키텍처 필드, 117
  - 안정성, 214
  - 압축, 222
    - 성능, 449
    - compress\_index 옵션, 411
  - 액세스 거부, 문제점 해결, 446
  - 액세스, 권한 부여, 243
  - 액세스 제어, 154, 263
  - 언어 환경 변수, 359
  - 업데이트
    - 네트워크 저장소 만들기, 159

- SD-UX, 457
  - 업데이트된 파일 복구, 79
  - 에이전트, 30
    - 권한, 266
    - 보안, 264
    - 컨트롤러 요청 처리, 269
  - UNIX, 대체 소스 사용, 225
  - 에이전트 폴링, 451
  - 여러 버전, 230
    - 설치, 81
    - 저장소, 152
    - 제거, 133
    - swconfig, 86
  - 여러 저장소, 137
  - 여러 테이프, 쓰기, 345
  - 여러 테이프에 파일 세트 나누기, 345
  - 영역, 265
  - 예
    - 명령 옵션, 62
    - 세션 파일, 64
    - swconfig, 90
    - swmodify, 122
    - swremove, 132
    - swverify, 96
  - 예약
    - 작업 브라우저의 아이콘, 197
  - 예제
    - request 스크립트, 388
    - swask, 388
  - 오류
    - 네트워크, 450
    - 데몬 로그 파일, 453
    - 디스크 공간 분석이 올바르지 않은 경우, 452
    - 메시지, 439
    - 문제 해결, 439
    - 설치 장애, 454
    - 액세스 거부, 446
    - 작업 브라우저에 표시됨, 197, 202
    - 테이프 저장소를 읽을 수 없는 경우, 454
    - 파일 세트 설치, 221
    - 해결, 439
    - ACL, 247
    - GUI가 시작되지 않음, 445
    - PSF 구분, 326
    - PSF 읽기, 289
    - RPC 시간 제한, 450
    - swpackage, 320, 322
    - UNIX 패키징, 452
    - WAN 연결 시간 제한, 450
  - 오류 메시지, 제어 스크립트, 356
  - 온라인 도움말, 54
  - 옵션
    - 메뉴, 48
    - 및 기본값, swconfig, 122
    - 및 기본값, swremove, 132
    - 변경, 61, 400
    - 알파벳 목록, 402
    - 우선권, 400
    - 작업 관련, 211
    - 편집기, 48
    - compress\_index, 411
    - create\_time\_filter, 413
    - preserve\_create\_time, 427
  - 와일드 카드 사용, 부분적, 321
  - 와일드카드, 344
  - 용어, 469
  - 용어집, 469
  - 원격 작업, 167
    - 개요, 168
    - 기본 설정, 184
    - 대상 선택, 180
    - 소프트웨어 선택, 182
    - 소프트웨어 설치, 186
    - 작업 결과 모니터링, 187
    - swlist, 188
  - 원격 프로시저 호출(RPC), 263
  - 위임, 271
  - 위치 검색 가능 제품, 81
  - 응답 파일, 372, 385
  - 응용 프로그램 기본값 파일, 56
  - 이 설명서의 독자, 17
  - 이미지, 저장소, 448
  - 인터프리터, 스크립트, 354
  - 읽기 권한, 154
  - 입력 파일, 78
- ㅈ
- 자격 증명, 265
  - 자동 복구, 406
  - 자동 화면 이동, 71, 128, 145
  - 자습서 전체 조건, 178
  - 작동 실패, 441
  - 작성 시간, 427
  - 작업
    - 명령줄에서 모니터링, 207
    - 옵션, 211
    - 작업 정보의 재사용, 205
    - swjob을 사용하여 제거, 207
  - 작업 결과, 202

## 색인

- 작업 결과 모니터링, 187
- 작업 고유 권한, 276
- 작업 로그, 205
- 작업 메뉴, 69, 126
- 작업 브라우저
  - 경고가 발생한 작업 아이콘, 197
  - 보안 점검, 276
  - 복사 아이콘, 196
  - 설명, 195
  - 설치 아이콘, 196
  - 소스 또는 대상의 재사용, 205
  - 수행 가능한 조치, 202
  - 아이콘 설명, 195
  - 예약된 작업 아이콘, 197
  - 오류 표시, 197, 202
  - 작업 복사, 205
  - 작업 제거, 205
  - 제거 작업 아이콘, 198
  - 호출, 194
- 작업 브라우저 등록 정보, 200
- 작업 브라우저의 열 편집기, 44
- 작업 설명, 203
- 작업 작성, 202, 205
- 잠긴 소프트웨어, 80, 109, 152
- 저장소
  - 공시, 153
  - 관리, 137
  - 권한, 154
  - 나열, 160
  - 내용 나열, 161
  - 등록, 153
  - 등록 해제, 153
  - 등록되지 않은, 271
  - 등록되지 않음, 154
  - 디렉토리, 137
  - 목록, 113
  - 배포, 29, 137
  - 복사, 140
  - 소프트웨어 제거, 125, 164
  - 여러, 137
  - 원격 파일 시스템, 343
  - 정의, 29, 137
  - 타이프, 138
  - ACL 권한, 257
  - ACL 제어, 254
  - CD-ROM, 336
  - swreg 명령, 154, 156
- 저장소 공시, 153
- 저장소 등록, 153
- 저장소 등록, 안전, 271
- 저장소 등록 해제, 153
- 저장소 사용, 135
- 저장소 제품을 관리하는 사용자, 243
- 저장소를 CD-ROM에 마스터, 336
- 저장소에 대한 액세스 제한, 244
- 전제 조건, 36
  - 정의, 36
- 정렬, 47
  - 정의
  - 노드, 29
  - 대상, 29
  - 로컬 호스트, 29
  - 번들, 30
  - 서버, 29
  - 세션 파일, 63
  - 소프트웨어 객체, 30
  - 소프트웨어 선택 파일, 60, 61
  - 속성, 98
  - 시스템, 29
  - 아키텍처 필드, 117
  - 용어, 469
  - 위치 검색 가능 제품, 81
- 저장소, 29, 137
- 전제 조건, 36
- 제품, 30
  - 카탈로그 파일, 32, 117
  - 클라이언트, 29
  - 태그, 117
  - 파일 세트, 30
  - 필수 조건, 36
  - 하위 제품, 30
  - 호스트, 29
- IPD(Installed Products Database), 32, 117
- uname 속성, 117
- 정책 설정, 61, 400
- 제거
  - 간단, 132
  - 대체 루트에서 소프트웨어 제거, 134
  - 소프트웨어, 124
  - 작업, 207
  - 저장소에서 소프트웨어, 164
- 창, 129
- 제거할 소프트웨어 선택, 126
- 제어 스크립트, 283
- 다른 명령의 실행, 374
- 세부 사항, 365
- 셀, 356
- 수퍼유저로 실행, 263
- 위치 및 실행, 365
- 입력 및 출력, 375



- 작성, 348
  - 종류, 349
  - 지침, 356
  - 테스트, 380
  - 파일 관리, 379
  - 형식, 355
  - 환경 변수, 359
  - request, 349, 357, 372
  - swask 명령, 385
  - unpostinstall, 227
  - unpreinstall, 227
  - 제어 스크립트 위치, 358
  - 제품, 30, 281
    - 설명, swremove, 128
    - 수준, 지정(swlist), 110
    - description 단추, 71, 128, 145
    - summary 단추, 71, 128, 145
  - 제품 ACL
    - 사용 권한, 258
    - 제어, 255
    - 템플릿, 260, 262
  - 제품 사양, 304
  - 제품 사양 파일(PSF), 435
  - 제품 요약, swremove, 128
  - 제품-위치 디렉토리 쌍, 133
  - 중속 대상, 408
  - 중속성, 35, 84, 91, 414
    - swconfig, 86
    - swcopy, 139
  - 중간 저장소, 224
  - 중요한 용어, 469
  - 지원 파일
    - install-sd, 458
- ㅈ
- 참조된 번들, 408
  - 창
    - GUI 구성 요소, 39
    - Software Selection, 69, 143
    - swremove, 129
  - 추가
    - 디스크 공간 요구 사항, 353
- ㅋ
- 카탈로그 파일, 32, 117, 118
  - 편집, 32, 118
  - 커널
    - 재구축, 66
    - swcopy를 위한 재구축, 139
  - 커널 빌드, 415
  - 커널 파일 세트, 402
  - 컨테이너 ACL 템플릿, 260, 262
  - 컨트롤러, 168
  - 컨트롤러 권한, 266
  - 컨트롤러 로그 파일, 422
  - 코드워드, 36
  - 코드워드, 사용, 69, 80, 109, 143, 152
  - 클라이언트, 정의, 29
  - 클라이언트/서버, 24
  - 키워드
    - 값, 제어 스크립트, 357
    - checkinstall 스크립트, 세부 사항, 366
    - checkremove 스크립트, 세부 사항, 370
    - configure 스크립트, 세부 사항, 368
    - postinstall 스크립트, 세부 사항, 367
    - postremove 스크립트, 세부 사항, 371
    - preinstall 스크립트, 세부 사항, 367
    - preremove 스크립트, 세부 사항, 371
    - unconfigure 스크립트, 세부 사항, 369
    - verify 스크립트, 세부 사항, 369, 370
  - 키워드 구문, PSF, 289
- ㅌ
- 태그, 117
  - 터미널 사용자 인터페이스(TUI), 27, 37, 67, 125, 140
  - 테스트
    - 구성 스크립트, 381
    - 설치 스크립트, 380
    - 제거 스크립트, 383
  - 테스트용 개발 저장소, 275
  - 테이프
    - 변경, 74
    - 저장소, 138
  - 테이프 만들기(기존 저장소), 346
  - 테이프, 여러 개에 파일 세트 나누기, 345
  - 테이프 장치, 414
  - 테이프 형식
    - cpio, 329
    - tar, 329
  - 테이프에 쓰기 명령 종료, 345
  - 테이프에 쓰기, 종료, 345
  - 템플릿 ACL, 260
    - 기본 항목, 261
  - 트로이 목마, 271

# 색인

## 표

### 파일

- 공간, 353
- 공유 가능, 281
- 대상, 61
- 비밀, 267
- 세션, 63
- 소프트웨어, 60
- 수준 지정(swlist), 112
- 수준 확인, swverify, 92
- 압축, 222
- 응답, 372, 385
- 카탈로그, 32, 117
- defaults.hosts, 173
- exclude, 322
- include, 322
- 파일 구조, 281
- 파일 사양, 316
  - 명시적, 319
  - 재귀적(암시적), 321
- 파일 세트, 30, 281
  - 수준, 지정(swlist), 112
- 파일 세트 사양, 309
- 파일 시스템 구조
  - SD-UX 에이전트, 464
  - SD-UX 컨트롤러, 467
- 파일 시스템 마운트, 424
- 파일 시스템 보호, 263
- 파일에 대한 액세스, 263
- 패치, 408, 426
  - 관리, 165
- 패치 관리, 165
- 패키지 다시 사용, 340
- 패키징
  - 개요, 280, 325
  - 다시 패키징, 340
  - 보안, 337
  - 비권한 SD용, 394
  - 원격 파일 시스템, 343
  - 장애, 452
  - 저장소 등록, 335
  - 테이프 만들기, 346
  - 테이프에 쓰기, 345
  - ACL, 337
  - CD-ROM, 336
- 패키징 명령, 324
- 폴링 간격, 증가, 451
- 표시, 39
- 폴다운 메뉴, 39
- 프로토콜 순서, 402, 431

### 플래그

- "yes", 39
- Marked?, 69, 126, 143
- 필수 조건, 36
  - 정의, 36
- 필요한 권한, 문제점 해결, 446
- 필터, 45

## ㅎ

- 하위 제품, 30, 281
  - 수준, 지정(swlist), 111
- 하위 제품 사양, 308
- 항목 열기, 50
- 호스트
  - 작업 브라우저에서 재사용, 205
  - 정의, 29
- 호스트 ACL, 261
  - 사용 권한, 256
- 호스트간 비밀 파일, 447
- 호환성 필터링, 82
- 확인
  - 버전 상태, 92
  - 분석 단계, 92
  - 설치, 91
  - 스크립트, 세부 사항, 370
  - 스크립트, 실행, 92
  - 작업, 샘플, 96
  - 종속성, 92
- 확장성, 217
- 환경 변수
  - LANG, 359
  - LC\_ALL, 359
- 환경 변수, 제어 스크립트, 359
- 활성, 410

## A

- ACL, 234
  - 경고, 247
  - 관리자, 269
  - 기본 템플릿 항목, 261
  - 루트, 254, 257
  - 머리말, 238
  - 명령 옵션, 236
  - 사용 권한, 253
  - 샘플, 248
  - 수정의 영향, 446
  - 수퍼유저 액세스, 250
  - 액세스 거부, 446
  - 오류, 247

일치, 250  
 작성, swpackage, 339  
 저장소, 254, 257  
 정의, 251  
 제품, 255, 258  
 키 값, 252  
 템플릿, 260  
 패키징, 337  
 편집, 247  
 항목 필드, 251  
 호스트, 256  
 any\_other, 251  
 depot, 448  
 group, 251  
 object\_group, 251  
 object\_owner, 251  
 swacl 명령, 236  
 swacl을 사용하지 않고 ACL 파일 수정, 447  
 user, 251  
 agent=, 402  
 agent\_auto\_exit=, 403  
 agent\_timeout\_minutes=1000, 403  
 allow\_downdate=, 403  
 allow\_incompatible 기본값  
   swconfig, 84  
 allow\_incompatible 옵션, 82  
 allow\_incompatible=, 404  
 allow\_multiple\_versions default, 81  
 allow\_multiple\_versions 옵션, 152, 231  
 allow\_multiple\_versions=, 404  
 allow\_partial\_bundles, 404  
 allow\_split\_patches, 404  
 alternate\_source=, 405  
 Analysis 대화 상자, 71, 127, 145  
 ask 옵션, 405  
 ask=, 405  
 auto\_kernel\_build=, 406  
 autoreboot=, 406  
 autorecover, 80, 406  
 autorecover\_product=, 407  
 autorecover\_product= default, 79  
 autoremove\_job, 212, 407  
 autoselect\_dependencies=, 408  
 autoselect\_dependencies=true 옵션, 35  
 autoselect\_dependents=, 408  
 autoselect\_patches=, 408  
 autoselect\_reference\_bundles=, 408

## C

-C 옵션, 75, 88, 94, 101, 119, 130, 149, 154,  
   207, 330, 385  
 CD-ROM  
   저장소, 336  
 CD-ROM, 마스터, 336  
 check\_volatile=, 410  
 check\_contents\_uncompressed, 409  
 check\_contents\_use\_chksum, 409  
 check\_permissions=, 409  
 check\_requisites=, 410  
 check\_scripts=, 410  
 check\_volatile 옵션, 319  
 checkinstall 스크립트, 349  
   세부 사항, 366  
 checkremove 스크립트, 349  
   세부 사항, 370  
 codeword=, 410  
 communication failure, 73, 147  
 compress\_cmd=, 410  
 compress\_files 옵션, 222, 449  
 compress\_files=, 410  
 compress\_index 옵션, 411  
 compression\_type=, 411  
 config\_cleanup\_cmd=, 411  
 configure 스크립트, 349  
   세부 사항, 368  
   실행, 87  
 CONFIGURED 상태, 84  
 control\_files=, 412  
 controller\_source=, 412  
 corequisite(필수 조건), 314  
 CORRUPT 상태, 86  
 cpio 테이프 형식, 329  
 create\_target\_acls 옵션, 336, 448  
 create\_target\_acls=, 412  
 create\_target\_path=, 413  
 create\_time\_filter 옵션, 413  
 crwit, 253  
 customer\_id, 사용, 80, 109, 152  
 customer\_id=, 413  
 customer\_ids, 36

## D

DCE 런타임 라이브러리, 265  
 DCE 없이 작업, 265  
 defaults.hosts 파일, 173  
 defer\_configuration=, 413  
 defer\_configure 기본값, 84

---

## 색인

defer\_configure 옵션, 229  
depot  
  이미지, 448  
  읽을 수 없음, 454  
distribution\_source\_directory=, 414  
distribution\_target\_directory=, 414  
distribution\_target\_serial=, 414

### E

enforce\_dependencies 기본값, 35  
  swconfig, 84  
enforce\_dependencies=, 414  
enforce\_dsa=, 415  
enforce\_kernbld\_failure=, 415  
enforce\_locatable, 415  
enforce\_scripts=, 415  
exclude 파일, 322  
excluded  
  due to errors, 73, 147  
  from task, 73, 147  
exerequisite(제외 조건), 314

### F

-f 옵션, 60, 75, 88, 94, 101, 103, 119, 130, 149, 154, 207, 330, 385  
f1 키, 도움말, 54  
file 메뉴, 199  
files=, 416  
fix 스크립트, 349  
follow\_symlinks=, 416

### G

global\_product\_template, 239, 246, 260  
global\_soc\_template, 239, 260  
go up, 50  
group  
  ACL, 251  
GUI  
  시작되지 않음, 445  
GUI 및 TUI  
  swlist, 98

### H

\$HOME/.sw/sessions/ directory, 63  
hosts 키워드, 173  
HP-UX SD 컨트롤러  
  정의, 168

HP-UX 업데이트, 458

### I

-i 옵션, 75, 149, 207  
include 파일, 322  
include\_file\_revisions=, 416  
install\_cleanup\_cmd=, 416  
install\_setup\_cmd=, 417  
INSTALLED 상태, 84  
installed\_software\_catalog, 416  
install-sd  
  구문, 459  
  옵션, 459  
  지원 파일, 458  
  SD 업데이트, 457  
IPD, 32, 117, 468  
  편집, 32, 118  
IPD(Installed Products Database), 32, 117, 468  
is\_kernel 속성, 309  
is\_locatable 속성, 305, 309  
is\_reboot 속성, 80, 309

### J

job\_title, 211, 417

### K

kernel\_build\_cmd=, 417  
kernel\_path=, 418

### L

LANG 환경 변수, 359  
layout\_version=, 418  
LC\_ALL 환경 변수, 359  
level  
  지정, swlist, 109  
level=, 419  
level= default, 107, 110  
log\_msgid=, 420  
logdetail=, 421  
logfile, 71, 128, 145  
  너무 긴 경우, 453  
  단추, 73, 147  
  swremove, 128  
logfile=, 422  
loglevel 옵션, 220  
loglevel=, 422

## M

mark  
 for copy, 126  
 for install, 69, 126  
 for remove, 126  
 Marked? 플래그, 69, 126, 143  
 match\_target 옵션, 309  
 match\_target=, 422  
 match\_target= 옵션, 69  
 Match-What-Target-Has, 69  
 max\_agents=, 422  
 max\_target 옵션, 217  
 max\_targets, 423  
 media\_capacity 옵션, 345  
 media\_capacity=, 423  
 media\_type=, 423  
 minimum\_job\_polling\_interval, 423  
 mount\_all\_filesystems=, 424  
 mount\_cmd=, 424

## N

num\_entries 값, 447

## O

objects\_to\_register=, 424  
 one\_liner=, 424  
 one\_liner= default, 107, 110  
 option 메뉴, 201  
 OS 업데이트, 417, 429  
 os\_name=, 425  
 os\_release=, 425

## P

-p 옵션, 75, 88, 130, 149  
 package\_in\_place=, 425  
 patch\_commit=, 426  
 patch\_filter=, 426  
 patch\_match\_target=, 426  
 patch\_one\_liner=, 426  
 patch\_save\_files=, 427  
 polling\_interval 옵션, 451  
 polling\_interval=, 427  
 postinstall 스크립트, 349  
 세부 사항, 367  
 postremove 스크립트, 349  
 세부 사항, 371  
 preinstall 스크립트, 349

세부 사항, 367  
 preremove 스크립트, 349  
 세부 사항, 371  
 prerequisite(전제 조건), 314  
 preserve\_create\_time option, 427  
 product specification file, PSF, 285  
 product\_template, 260  
 Products Ready 열, 73, 147  
 Projected Actions  
 swremove, 128  
 PSF, 285  
 공급업체 클래스, 301  
 구분, 289  
 권한 사양 예, 317  
 디렉토리 매핑, 318  
 만들기, 285  
 명시적 파일 사양, 319  
 및 swmodify, 119  
 예제, 286  
 인용 부호, 289  
 재귀적 파일 사양, 321  
 저장소 클래스, 299  
 제품 클래스, 304  
 종속성 클래스, 314  
 주석 줄, 289  
 키워드, 289  
 키워드 값, 289  
 파일 사양의 예, 321  
 파일 세트 클래스, 309  
 파일 클래스, 316  
 하위 제품 클래스, 308  
 확장, 322  
 exclude 파일, 322  
 include 파일, 322  
 PSF(Packaging Specification File)  
 및 swmodify, 119  
 pushAgent, 설치 시 ACL을 변경, 240

## R

-r 옵션, 75  
 README, 108  
 ready, 73, 147  
 with errors, 73, 147  
 with warnings, 73, 147  
 reboot\_cmd=, 427  
 reconfigure=, 428  
 reconfigure=true/false 옵션, 86  
 register\_new\_depot=, 428  
 register\_new\_root=, 428

---

## 색인

- reinstall 옵션, 152, 220
  - reinstall=, 428
  - reinstall\_files 옵션, 220
  - reinstall\_files=, 428
  - reinstall\_files\_use\_cksum 옵션, 220
  - reinstall\_files\_use\_cksum=, 429
  - remove\_empty\_depot=, 429
  - remove\_obsolete\_filesets=, 429
  - remove\_setup\_cmd=, 429
  - request 스크립트, 349, 372
    - 예제, 388
    - 응답 파일, 372
    - 키워드, 357
    - swask 명령, 385
    - swinstall 또는 swconfig로부터 실행, 388
  - resolver 명령, 448
  - Resume 단추, 74
  - retry\_interval, 219
  - retry\_rpc, 218, 219
  - retry\_rpc 옵션, 218
  - retry\_rpc=, 430
  - retry\_rpc\_interval, 430
  - reuse\_short\_job\_numbers, 431
  - RPC 권한, 269
  - RPC 시간 제한, 450
  - rpc\_binding\_info 옵션, 451
  - rpc\_binding\_info=, 431
  - rpc\_binding\_info\_alt\_source, 432
  - rpc\_timeout 옵션, 218, 450
  - rpc\_timeout=, 432
- S**
- S 옵션, 75, 88, 94, 103, 119, 130, 149, 154, 207, 330, 385
  - s 옵션, 75, 103, 149, 385
  - sd
    - 보안 점검, 276
    - 호출, 194
  - SD 객체 보호, 254
  - SD 내부 권한 인증, 265
  - SD 문제점 해결, 439
  - SD 재설치, 457
  - SD-UX
    - 맨페이지, 28
    - 명령, 28
  - SD-UX 컨트롤러, 168
  - select\_local=, 433
  - setuid root, 337
  - show\_superseded\_patches, 433
  - Software Distributor
    - 소개, 24
  - Software Selection 창, 69, 143
  - software=, 433
  - software\_view=, 433
  - source\_cdrom=, 434
  - source\_depot\_audit=, 434
  - source\_file=, 435
  - source\_tape=, 435
  - source\_type=, 435
  - Specify Source 대화 상자, 68, 142
  - stty, 문자 매핑 판별에 사용, 60, 189
  - Support Plus로 원격 액세스, 153, 156
  - Support Plus로 직접 액세스, 153, 156
  - SW\_CONTROL\_DIRECTORY, 360
  - SW\_DEFERRED\_KERNBLD, 363
  - SW\_INITIAL\_INSTALL, 363
  - SW\_KERNEL\_PATH, 363
  - SW\_LOCATION, 361
  - SW\_PATH, 361
  - SW\_ROOT\_DIRECTORY, 362
  - SW\_SYSTEM\_FILE\_PATH, 364
  - swacl, 278
    - 개요, 25
    - 사용자 액세스 나열, 240
    - D 옵션, 247
    - F 옵션, 247
    - l depot 옵션, 240
    - l product 옵션, 240
    - l root 옵션, 240
    - l 옵션, 240
    - M 옵션, 247
  - swacl 명령, 236
    - 옵션, 236
  - swadm 그룹, 238
  - swagent, 30, 168
  - swagentd, 30, 168
    - 개요, 25
  - swask, 278, 385
    - 구문, 385
    - 예제, 388
  - swconfig
    - 명령, 84
    - 보안 점검, 278
  - swcopy
    - 개요, 25
    - 보안 점검, 276
    - 종속성, 139
    - GUI 개요, 140

## SW-DIST

- 새 버전 로드, 457
- 손상된 경우 다시 로드, 457
- swgettools, 458
- swinstall
  - 개요, 25
  - 디스크 공간 분석, 353
  - 보안 점검, 277
- swjob, 191
  - 명령 정보, 207
  - 보안 점검, 276
- swlist
  - 개요, 25
  - 구문, 100
  - 등록된 저장소 나열, 160
  - 명령, 98
  - 보안 점검, 276
  - 원격 작업을 위한 -i 옵션, 188
  - 저장소 내용 나열, 161
  - a(속성) 옵션, 101
  - d 옵션, 101
  - examples, 105
  - i 옵션, 98, 101
  - l depot 옵션, 160
  - l 옵션, 101, 109
  - R(축약) 옵션, 101
  - v(상세) 옵션, 101
- swlock 파일, 118, 468
- swmodify, 278
  - 개요, 25
  - 구문, 119
  - a 옵션, 119
  - d 옵션, 119
  - P 옵션, 119
  - p 옵션, 119
  - r 옵션, 119
  - s 옵션, 119
  - u 옵션, 119
  - V 옵션, 119
  - v 옵션, 119
- swmodify용 PSF(product\_specification\_file), 119
- swpackage
  - 개요, 25, 325
  - 구문, 330
  - 보안 점검, 276
  - 옵션, 330
  - logfile, 333
- swreg
  - 개요, 25, 153
  - 보안 점검, 278

## swremove

- 구문, 130
- 보안 점검, 277
- d 옵션, 130, 164
- i 옵션, 130
- r 옵션, 130

## swverify

- 개요, 25
- 명령, 91
- 보안 점검, 278
- 저장소 확인, 163
- d 옵션, 163

system\_file\_path=, 435

system\_prep\_cmd=, 435

## T

- t 옵션, 61
- tag 속성, 항상 나열됨, 108
- tape is ready 응답, 345
- tar 아카이브, 138
- tar 테이프 형식, 329
- target\_type, 436
- target\_type 옵션, 330
- targets=, 435
- TCP/IP
  - 프로토콜, 451

## U

- u 옵션
  - swconfig, 88
- UDP 통신, 451
- umask 값, 317
- uname 속성, 82, 117
- uncompress\_cmd=, 436
- uncompress\_files 옵션, 222
- uncompress\_files=, 436
- unconfigure 스크립트, 349
  - 세부 사항, 369
- UNCONFIGURED 상태, 87
- UNIX
  - 사용자 모드, 24
  - 실행 수준, 24
- UNIX 실행 수준, 24
- unpostinstall 스크립트, 349
  - 자동 복구용, 227
- unpreinstall 스크립트, 349
  - 자동 복구용, 227
- use\_alternate\_source 옵션, 224, 225
- use\_alternate\_source=, 436

---

## 색인

user  
ACL, 251

### V

/var/adm/sw/defaultstor \$HOME/.sw/defaults 파일  
, 61, 400  
/var/adm/sw/products file, 117  
/var/adm/sw/software/ directory, 60, 61  
var/spool/sw, 137  
/var/spool/sw/catalog 파일, 117, 118  
/var/tmp 디렉토리, 458  
verbose=, 437  
verify  
스크립트, 349  
스크립트, 세부 사항, 369  
verify 스크립트, 410  
view 메뉴, 199

### W

WAN, 218, 224  
WAN 연결 시간 제한, 450  
WAN(wide area network), 224  
write\_remote\_files 옵션, 343  
write\_remote\_files=, 437

### X

-x codeword=, 81  
-x customer\_id=, 81  
-x 옵션, 75, 88, 94, 103, 119, 130, 149, 154,  
207, 330  
XToolkit  
-fn 옵션, 55  
-font 옵션, 55