



Sun Java™ System
Directory Server 5.2
관리 설명서

2005Q1

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

부품 번호: 819-2012

Copyright © 2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 모든 권리는 저작권자의 소유입니다.

Sun Microsystems, Inc.는 이 문서에 설명된 제품의 기술 관련 지적 재산권을 소유합니다. 특히 이 지적 재산권에는 <http://www.sun.com/patents>에 나열된 하나 이상의 미국 특허권이 포함될 수 있으며, 미국 및 다른 국가에서 하나 이상의 추가 특허권 또는 출원 중인 특허권이 제한 없이 포함될 수 있습니다.

이 제품에는 Sun Microsystems, Inc.의 기밀 정보 및 무역 비밀이 포함되어 있습니다. Sun Microsystems, Inc.의 명시된 사전 서면 승인 없이는 해당 기밀의 사용, 공개 또는 복제가 금지됩니다.

미국 정부의 권리 - 상용 소프트웨어. 정부 사용자는 Sun Microsystems, Inc. 표준 사용권 계약과 해당 FAR 규정 및 보충 규정을 준수해야 합니다.

이 배포에는 타사에서 개발한 자료가 포함되어 있을 수 있습니다.

제품 중에는 캘리포니아 대학에서 허가한 Berkeley BSD 시스템에서 파생된 부분이 포함되어 있을 수 있습니다. UNIX는 미국 및 다른 국가에서 X/Open Company, Ltd.를 통해 독점적으로 사용권이 부여되는 등록 상표입니다.

Sun, Sun Microsystems, Sun 로고, Java, Solaris, JDK, Java Naming 및 Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, Duke 로고, Java Coffee Cup logo, Solaris 로고, SunTone Certified 로고 및 Sun ONE 로고는 미국 및 다른 국가에서 Sun Microsystems, Inc.의 상표 또는 등록 상표입니다.

SPARC 상표는 사용 허가를 받았으며 미국 및 다른 국가에서 SPARC International, Inc.의 상표 또는 등록 상표입니다. SPARC 상표를 사용하는 제품은 Sun Microsystems, Inc.가 개발한 구조에 기반을 두고 있습니다.

Legato 및 Legato 로고는 Legato Systems, Inc.의 등록 상표이고 Legato NetWorker는 Legato Systems, Inc.의 상표 또는 등록 상표입니다. Netscape Communications Corp 로고는 Netscape Communications Corporation의 상표 또는 등록 상표입니다.

OPEN LOOK 및 Sun(TM) 그래픽 사용자 인터페이스(GUI)는 Sun Microsystems, Inc.가 자사의 사용자 및 정식 사용자용으로 개발했습니다. Sun은 컴퓨터 업계를 위한 시각적 또는 그래픽 사용자 인터페이스(GUI)의 개념을 연구 개발한 Xerox사의 선구적인 노력을 높이 평가하고 있습니다. Sun은 Xerox와 Xerox 그래픽 사용자 인터페이스(GUI)에 대한 비독점적 사용권을 보유하고 있습니다. 이 사용권은 OPEN LOOK GUI를 구현하는 Sun의 정식 사용자에게도 적용되며 그렇지 않은 경우에는 Sun의 서면 사용권 계약을 준수해야 합니다.

이 서비스 설명서에서 다루는 제품과 수록된 정보는 미국 수출 관리법에 의해 규제되며 다른 국가의 수출 또는 수입 관리법의 적용을 받을 수도 있습니다. 이 제품과 정보를 직간접적으로 핵무기, 미사일 또는 생화학 무기에 사용하거나 핵과 관련하여 해상에서 사용하는 것은 엄격하게 금지됩니다. 미국 수출 금지 국가 또는 금지된 개인과 특별히 지정된 국민 목록을 포함하여 미국 수출 금지 목록에 지정된 대상으로의 수출이나 재수출은 엄격하게 금지됩니다.

설명서는 "있는 그대로" 제공되며, 법률을 위반하지 않는 범위 내에서 상품성, 특정 목적에 대한 적합성 또는 비침해에 대한 묵시적인 보증을 포함하여 모든 명시적 또는 묵시적 조건, 표현 및 보증을 배제합니다.

목차

그림 목차	19
표 목차	21
절차 목록	23
코드 예 목록	25
머리말	27
규칙	27
관련 설명서	30
Directory Server 설명서	30
Administration Server 설명서	30
Directory Proxy Server 설명서	30
관련된 Java Enterprise System 설명서	30
설명서, 지원 및 교육	31
타사 웹 사이트 관련 참조 사항	31
사용자 의견 환영	31
1장 Directory Server 관리 개요	33
Directory Server 관리 개요	34
Directory Server 시작 및 중지	35
명령줄에서 서버 시작 및 중지	35
콘솔에서 서버 시작 및 중지	35
SSL을 활성화하여 서버 시작	36
1024보다 적은 포트에서 비루트 사용자로 서버 시작	37
Directory Server 콘솔 사용	37
Directory Server 콘솔 시작	38

Directory Server 콘솔 시작	39
태스크 탭	40
구성 탭	41
디렉토리 탭	42
상태 탭	44
콘솔에서 현재 바인드 DN 보기	45
로그인 아이디 변경	45
온라인 도움말 사용	45
콘솔 클립보드	46
콘솔 설정	47
시각적 구성 기본 설정	47
디렉토리 트리 보기 옵션	47
LDAP 매개 변수 구성	49
디렉토리 관리자 구성	49
Directory Server의 포트 번호 변경	49
받는 LDAP 요청에 대해 Directory Server에서 수신하는 포트 또는 보안 포트를 수정하려면	50
전역 읽기 전용 모드 설정	51
디렉토리 항목에 대한 수정 추적	52
플러그인 서명 확인	53
플러그인 서명 확인 구성	54
플러그인의 상태 보기	55
DSML 구성	55
DSML 요청 사용	56
DSML 보안 구성	57
DSML 아이디 매핑	59
2장 디렉토리 항목 관리	61
구성 항목	62
콘솔에서 구성 수정	62
명령줄에서 구성 수정	63
dse.ldif 파일 수정	63
콘솔에서 항목 관리	64
디렉토리 항목 작성	64
사용자 정의 편집기에서 항목 작성	65
다른 유형의 항목 작성	67
사용자 정의 편집기에서 항목 수정	68
사용자 정의 편집기 호출	69
언어 지원 속성 설정	70
일반 편집기에서 항목 수정	70
일반 편집기 호출	70
디렉토리 항목의 일반 편집기를 호출하려면	70
속성 값 수정	72
속성 값을 수정하려면	72

여러 값을 갖는 속성 편집	73
속성 추가	73
속성 제거	74
객체 클래스 관리	75
디렉토리 항목 삭제	76
콘솔을 사용한 대량 작업	77
명령줄에서 항목 관리	77
LDIF 입력 제공	78
명령줄에서 LDIF 입력 종료	78
특수 문자 사용	79
속성 OID 사용	79
스키마 검사	79
LDIF 항목 정렬	79
대규모 항목 관리	80
오류 처리	81
ldapmodify를 사용한 항목 추가	81
ldapmodify를 사용한 항목 수정	83
속성 값 추가	84
이진 속성 하위 유형 사용	84
언어 하위 유형이 지정된 속성 추가	85
속성 값 수정	85
속성 값 삭제	86
여러 값을 갖는 속성의 값 하나만 수정	86
ldapdelete를 사용한 항목 삭제	87
ldapmodify를 사용한 항목 삭제	87
항목 이름 바꾸기 및 이동	88
DN 수정 작업 소개	88
이름 바꾸기 및 이동 작업 사이의 차이	88
DN 수정 작업 사용에 대한 지침과 제한 사항	89
콘솔에서 항목 이름 바꾸기 또는 이동	92
콘솔에서 DN 수정 작업 활성화	92
콘솔에서 ACI 권한을 확장하려면	92
콘솔에서 DN 수정 작업을 전역적으로 활성화 또는 비활성화하려면	93
콘솔에서 지정된 접미어에 대해 DN 수정 작업을 활성화하려면	94
콘솔에서 항목 이름 바꾸기	95
콘솔에서 항목의 이름을 바꾸려면	95
콘솔에서 항목 이동	96
콘솔에서 항목을 이동하려면	96
콘솔에서 항목 이동 및 이름 바꾸기	96
ldapmodify 명령을 사용하여 항목 이름 바꾸기 또는 이동	96
ldapmodify 명령을 사용하여 DN 수정 작업 활성화	97
ldapmodify 명령을 사용하여 DN 수정 작업을 전역적으로 활성화하려면	97
ldapmodify 명령을 사용하여 접미어에 대해 DN 수정 작업을 활성화하려면	97

ldapmodify 명령을 사용하여 항목 이름 바꾸기	98
ldapmodify 명령을 사용하여 리프 항목의 이름을 바꾸려면	98
ldapmodify 명령을 사용하여 항목 이동	98
ldapmodify 명령을 사용하여 항목을 이동하려면	98
ldapmodify 명령을 사용하여 항목의 이름을 바꾸고 이동하려면	99
참조 설정	99
기본 참조 설정	100
콘솔에서 기본 참조 설정	100
명령줄에서 기본 참조 설정	100
스마트 참조 작성	101
콘솔에서 스마트 참조 작성	101
명령줄에서 스마트 참조 작성	102
속성 값 암호화	103
콘솔에서 속성 암호화 구성	105
명령줄에서 속성 암호화 구성	106
참조 무결성 유지	108
참조 무결성 작동 방식	108
참조 무결성 구성	109
콘솔에서 참조 무결성 구성	109
복제에 참조 무결성 사용	110
레거시 복제에 참조 무결성 사용	110
디렉토리 검색	111
ldapsearch를 사용한 디렉토리 검색	112
ldapsearch 명령줄 형식	112
특수 문자 사용	113
자주 사용되는 ldapsearch 옵션	113
ldapsearch 예	115
모든 항목 반환	115
명령줄에서 검색 필터 지정	115
루트 DSE 항목 검색	115
스키마 항목 검색	116
LDAP_BASEDN 사용	116
속성의 하위 집합 표시	117
여러 값을 갖는 속성 검색	117
검색 시 클라이언트 인증 사용	118
LDAP 검색 필터	118
검색 필터 구문	119
검색 필터에 속성 사용	119
검색 필터에 연산자 사용	120
검색 필터에 OID 사용	121
복합 검색 필터 사용	121
파일을 사용하여 검색 필터 지정	122
검색 필터에서 7비트 ASCII 이외의 문자 지정	123

검색 필터 예	123
자동 가능 속성 검색	124
국가별 디렉토리 검색	124
일치 규칙 필터 구분	125
일치 규칙 형식	125
일치 규칙 필터에 와일드카드 사용	127
지원되는 검색 유형	127
국제적 검색 예	128
보다 작음 예	128
작거나 같음 예	129
동일 예	129
크거나 같음 예	129
보다 큼 예	130
하위 문자열 예	130
DSMLv2를 사용한 디렉토리 액세스	130
빈 익명 DSML "Ping" 요청	131
DSML 요청 발급 및 특정 사용자로 바인드	134
DSML 검색 요청	136

3장 디렉토리 트리 작성 141

접미어 작성	141
콘솔에서 새 루트 접미어 작성	142
콘솔에서 새 하위 접미어 작성	144
명령줄에서 접미어 작성	147
접미어 관리	150
접미어 비활성화 또는 활성화	150
콘솔에서 접미어 비활성화 또는 활성화	150
명령줄에서 접미어 비활성화 또는 활성화	151
액세스 권한 및 참조 설정	151
콘솔에서 액세스 권한 및 참조 설정	152
명령줄에서 액세스 권한 및 참조 설정	153
접미어 삭제	153
콘솔에서 접미어 삭제	154
명령줄에서 접미어 삭제	154
연결 접미어 작성	155
프록시 아이디 작성	155
콘솔에서 프록시 아이디 작성	156
명령줄에서 프록시 아이디 작성	156
기본 연결 매개 변수 설정	157
클라이언트 반환 매개 변수	157
계단식 연결 매개 변수	158
연결 관리 매개 변수	158
오류 감지 매개 변수	159

콘솔에서 기본 연결 매개 변수 설정	159
명령줄에서 기본 연결 매개 변수 설정	160
콘솔에서 연결 접미어 작성	160
명령줄에서 연결 접미어 작성	162
연결 접미어를 통한 액세스 제어	166
SSL을 사용한 연결	167
연결 접미어 관리	168
연결 정책 구성	168
LDAP 컨트롤의 연결 정책	168
서버 구성 요소의 연결 정책	170
콘솔에서 연결 정책 수정	171
명령줄에서 연결 정책 수정	172
연결 접미어 비활성화 또는 활성화	173
콘솔에서 연결 접미어 비활성화 또는 활성화	173
명령줄에서 접미어 비활성화 또는 활성화	173
액세스 권한 및 참조 설정	174
콘솔에서 액세스 권한 및 참조 설정	174
콘솔에서 액세스 권한 및 참조 설정	175
연결 매개 변수 수정	176
콘솔에서 연결 매개 변수 수정	176
명령줄에서 연결 매개 변수 수정	178
스레드 사용 최적화	180
콘솔에서 스레드 자원 설정	180
명령줄에서 스레드 자원 설정	180
연결 접미어 삭제	181
콘솔에서 연결 접미어 삭제	181
명령줄에서 접미어 삭제	181
계단식 연결 구성	182
계단식 매개 변수 설정	183
콘솔에서 계단식 매개 변수 설정	183
명령줄에서 계단식 매개 변수 설정	184
계단식 연결을 위한 LDAP 컨트롤 전송	184
4장 데이터 백업 및 복원	187
접미어 읽기 전용 모드 설정	188
데이터 가져오기	188
LDIF 파일 가져오기	189
콘솔에서 LDIF 가져오기	189
명령줄에서 LDIF 가져오기	190
접미어 초기화	191
콘솔에서 접미어 초기화	191
ldif2db 명령을 사용하여 접미어 초기화	192
ldif2db-task를 사용하여 접미어 초기화	193

데이터 내보내기	194
콘솔에서 전체 디렉토리를 LDIF로 내보내기	195
콘솔에서 개별 접미어를 LDIF로 내보내기	196
명령줄에서 LDIF로 내보내기	196
데이터 백업	197
콘솔에서 서버 백업	198
명령줄에서 서버 백업	198
dse.ldif 구성 파일 백업	199
백업을 사용한 데이터 복원	199
복제된 접미어 복원	200
단일 마스터 시나리오에서 공급자 복원	200
다중 마스터 시나리오에서 공급자 복원	200
허브 복원	201
전용 소비자 복원	202
콘솔에서 서버 복원	203
명령줄에서 서버 복원	203
bak2db 명령 사용	203
bak2db-task 사용	204
dse.ldif 구성 파일 복원	204
5장 아이디 및 역할 관리	207
그룹 관리	208
새 정적 그룹 추가	208
새 동적 그룹 추가	209
그룹 정의 수정	210
그룹 정의 제거	210
역할 할당	210
역할	211
nsRole 속성 검색	212
nsRole 속성에 대한 권한	212
콘솔에서 역할 할당	213
관리된 역할 만들기	213
필터링된 역할 만들기	213
중첩된 역할 만들기	215
항목의 역할 보기 및 편집	215
역할 항목 수정	216
역할 삭제	216
명령줄에서 역할 관리	217
관리된 역할 정의 예	217
필터링된 역할 정의 예	218
중첩된 역할 정의 예	219
서비스 클래스(CoS) 정의	220
CoS	220

CoS 제한	222
콘솔에서 CoS 관리	223
새 CoS 만들기	223
기존 CoS 편집	225
CoS 삭제	226
명령줄에서 CoS 관리	226
명령줄에서 CoS 정의 항목 작성	226
실제 속성 값 무시	229
여러 값을 갖는 CoS 속성	229
Cos 속성 우선 순위	230
명령줄에서 CoS 템플릿 항목 작성	230
포인터 CoS 예	231
간접 CoS 예	232
클래식 CoS 예	233
역할 기반의 속성 작성	234
CoS 플러그 인 모니터	235

6장 액세스 제어 관리	237
액세스 제어 원칙	238
ACI 구조	238
ACI 배치	239
ACI 평가	239
ACI 제한	240
기본 ACI	241
ACI 구문	242
대상 정의	243
대상 디렉토리 항목 지정	245
대상 속성 지정	246
대상 항목과 속성 모두 지정	247
LDAP 필터를 사용한 대상 항목 또는 속성 지정	247
LDAP 필터를 사용한 대상 속성 값 지정	248
한 개의 디렉토리 항목을 대상으로 지정	249
매크로를 사용하여 대상 정의	250
권한 정의	250
액세스 허용 또는 거부	250
권한 지정	250
LDAP 작업에 필요한 권한	252
권한 구문	253
바인드 규칙	254
바인드 규칙 구문	254
사용자 액세스 정의 - userdn 키워드	256
익명 액세스(anyone 키워드)	256
일반 액세스(all 키워드)	257

자체 액세스(self 키워드)	257
부모 액세스(parent 키워드)	257
LDAP URL	257
와일드카드	258
LDAP URL의 논리적 OR	258
특정 LDAP URL 제외	258
그룹 액세스 정의 - groupdn 키워드	259
단일 LDAP URL	259
LDAP URL의 논리적 OR	259
역할 액세스 정의 - roledn 키워드	260
값 일치에 따른 액세스 정의	260
userattr 키워드 사용	260
상속 기능과 함께 userattr 키워드 사용	263
userattr 키워드를 사용한 추가 권한 부여	264
특정 IP 주소로부터의 액세스 정의	265
특정 도메인으로부터의 액세스 정의	266
특정 시간 또는 요일의 액세스 정의	267
인증 방법에 따른 액세스 정의	268
예	269
부울 바인드 규칙 사용	269
명령줄에서 ACI 작성	270
ACI 속성 값 보기	270
콘솔에서 ACI 작성	271
항목의 ACI 보기	272
새 ACI 작성	274
ACI 편집	275
ACI 삭제	276
액세스 제어 사용 예	276
익명 액세스 부여	277
개인 항목에 대한 쓰기 액세스 권한 부여	279
중요 역할에 대한 액세스 제한	282
그룹에 접미어에 대한 전체 액세스 권한 부여	284
그룹 항목 추가 및 삭제 권한 부여	285
그룹이나 역할에 조건부 액세스 권한 부여	288
액세스 거부	290
필터링을 사용한 대상 설정	292
사용자가 그룹에 자신을 추가 또는 제거할 수 있도록 허용	292
쉽표가 있는 DN에 대한 권한 정의	294
프록시 인증 ACI 예	294
유효 권한 보기	295
유효 권한 보기 컨트롤에 대한 액세스 제한	296
유효 권한 보기 컨트롤 사용	296
유효 권한 결과 이해	301

고급 액세스 제어: 매크로 ACI 사용	305
매크로 ACI 예	305
매크로 ACI 구분	308
대상의 (\$dn) 일치	308
주제의 (\$dn) 대체	309
주제의 [\$dn] 대체	309
(\$attr.attrName)의 매크로 일치	311
액세스 제어 및 복제	311
액세스 제어 및 연결	312
액세스 제어 로깅 정보	312
이전 릴리스와의 호환성	313

7장 사용자 계정 및 비밀번호 관리 315

비밀번호 정책에 대한 개요	316
전역 비밀번호 정책 구성	316
콘솔에서 비밀번호 정책 구성	316
명령줄에서 비밀번호 정책 구성	318
개별 비밀번호 정책 관리	319
콘솔에서 정책 정의	319
명령줄에서 정책 정의	321
비밀번호 정책 할당	322
콘솔 사용	322
명령줄 사용	323
역할 및 CoS 사용	324
개별 비밀번호 정책 보호	325
사용자 비밀번호 재설정	325
사용자와 역할 비활성화 및 활성화	326
콘솔에서 사용자 및 역할 활성화 설정	326
명령줄에서 사용자 및 역할 활성화 설정	327
개별 자원 제한 설정	327
콘솔에서 자원 제한 설정	328
명령줄에서 자원 제한 설정	329

8장 복제 관리 331

소개	332
구성 복제 단계 요약	334
복제 관리자 선택	336
전용 소비자 구성	337
소비자 복제본에 대한 접미어 작성	338
소비자 복제본 활성화	338
고급 소비자 구성	339
허브 구성	340

허브 복제본에 대한 접미어 작성	340
허브 복제본 활성화	340
고급 허브 구성	341
마스터 복제본 구성	343
마스터 복제본에 대한 접미어 정의	344
마스터 복제본 활성화	344
고급 다중 마스터 구성	345
복제 계약 작성	347
단편 복제 구성	349
단편 복제 시 고려 사항	349
속성 집합 정의	350
단편 복제 활성화	352
복제본 초기화	352
초기화 시기	353
다중 마스터 복제 시 복제본 초기화	353
계단식 복제 시 복제본 초기화	354
다중 마스터 초기화 후의 수렴	354
콘솔을 통해 업데이트 허용	355
명령줄을 통해 업데이트 허용	356
자동 참조 지연 설정	356
콘솔에서 복제본 초기화	357
온라인 복제본 초기화	357
명령줄에서 복제본 초기화	358
LDIF로 복제본 내보내기	359
단편 복제를 위한 LDIF 파일 필터링	359
소비자 복제본으로 LDIF 파일 가져오기	360
이진 복사를 사용한 복제본 초기화	360
이진 복사 제한	360
서버를 중지하지 않는 이진 복사	361
최소 디스크 공간을 사용한 이진 복사	362
참조 무결성 플러그인 활성화	363
SSL을 통한 복제	363
WAN을 통한 복제	364
네트워크 매개 변수 구성	365
복제 작업 예약	367
데이터 압축	367
복제 토폴로지 수정	368
복제 계약 관리	368
복제 관리자 변경	369
복제 계약 복사	369
복제 계약 비활성화	370
복제 계약 활성화	371
복제 계약 삭제	371

복제본 수준 올리기 또는 내리기	372
복제본 비활성화	373
변경 로그 이동	373
복제본을 동기화 상태로 유지	374
복제 재시도 알고리즘	375
콘솔에서 복제 업데이트 강제	375
명령줄에서 복제 업데이트 강제	376
이전 릴리스를 사용한 복제	377
Directory Server 5.2 및 Directory Server 5.1 간의 복제	377
Directory Server 5.2 및 Directory Server 4.x 간의 복제	378
Directory Server 5.2를 Directory Server 4.x의 소비자로 구성하려면	378
Directory Server 5.1 스키마 업데이트	380
Directory Server 5.1 스키마를 업데이트하려면	380
레트로 변경 로그 플러그인 사용	381
레트로 변경 로그 플러그인 설명	381
레트로 변경 로그 추가 정보	381
레트로 변경 로그 플러그인 활성화	382
Directory Server 콘솔을 사용하여 레트로 변경 로그 플러그인을 활성화하려면	382
명령줄을 사용하여 레트로 변경 로그 플러그인을 활성화하려면	383
지정된 접미어에 업데이트를 기록하도록 레트로 변경 로그 구성	383
Directory Server 콘솔을 사용하여 지정된 접미어에 대한 업데이트를 기록하도록 레트로 변경 로 그를 구성하려면	383
명령줄을 사용하여 지정된 접미어에 대한 업데이트를 기록하도록 레트로 변경 로그를 구성하려면	384
삭제된 항목의 속성을 기록하도록 레트로 변경 로그 구성	384
Directory Server 콘솔을 사용하여 삭제된 항목의 속성을 기록하도록 레트로 변경 로그를 구성하 려면	384
명령줄을 사용하여 삭제된 항목의 속성을 기록하려면 레트로 변경 로그를 구성하려면	385
레트로 변경 로그 지우기	385
레트로 변경 로그 액세스	385
복제 상태 모니터	386
명령줄 도구	386
복제 상태 탭	387
일반적인 복제 충돌 해결	388
이름 지정 충돌 해결	389
여러 값을 갖는 이름 지정 속성이 있는 항목의 이름 변경	389
한 개의 값을 갖는 이름 지정 속성이 있는 항목의 이름 변경	390
고아 항목 충돌 해결	391
잠재적 상호 운용성 문제 해결	391

9장 디렉토리 스키마 확장	393
스키마 검사	393
콘솔에서 스키마 검사 설정	395
명령줄에서 스키마 검사 설정	395
스키마 확장에 대한 개요	395
스키마 파일 수정	396
명령줄에서 스키마 수정	397
콘솔에서 스키마 수정	397
속성 정의 관리	398
속성 보기	398
속성 작성	400
속성 편집	401
속성 삭제	401
객체 클래스 정의 관리	402
객체 클래스 보기	402
객체 클래스 작성	403
객체 클래스 편집	404
객체 클래스 삭제	405
스키마 정의 복제	405
복제된 스키마 파일 수정	406
스키마 복제 제한	407
10장 디렉토리 데이터 색인화	409
색인화에 대한 개요	409
시스템 색인	411
기본 색인	411
속성 이름 빠른 참조 테이블	413
색인 관리	414
콘솔에서 색인 관리	415
명령줄에서 색인 관리	416
색인 구성 항목 작성	417
색인 구성 항목 수정	418
db2index-task 실행	419
특정 속성에 대한 모든 색인 삭제	419
접미어 다시 색인화	420
접미어 다시 색인화	420
접미어 다시 초기화	421
기본 색인 집합 수정	421
찾아보기 색인 관리	422
콘솔에 대한 찾아보기 색인	423
클라이언트 검색에 대한 찾아보기 색인	424
찾아보기 색인 항목 지정	424
vlvindex 명령 실행	426

11장 인증 및 암호화 관리	427
Directory Server의 SSL 소개	428
SSL 활성화 단계 요약	429
서버 인증서 얻기 및 설치	429
인증서 데이터베이스 작성	430
콘솔 사용	430
명령줄 사용	430
인증서 요청 생성	431
콘솔 사용	431
명령줄 사용	432
서버 인증서 설치	432
콘솔 사용	433
명령줄 사용	434
인증 기관 트러스트	434
콘솔 사용	435
명령줄 사용	435
SSL 활성화	436
암호화 암호 선택	438
클라이언트 인증 허용	440
클라이언트 인증 구성	440
DIGEST-MD5를 통한 SASL 인증	441
DIGEST-MD5 기법 구성	441
DIGEST-MD5 아이디 매핑	442
GSSAPI를 통한 SASL 인증(Solaris에만 해당)	443
커버로스 시스템 구성	443
GSSAPI 기법 구성	444
GSSAPI 아이디 매핑	444
아이디 매핑	446
LDAP 클라이언트에서 보안을 사용하도록 구성	448
클라이언트에 서버 인증 구성	450
Mozilla를 통한 클라이언트 인증서 관리	450
명령줄을 통한 클라이언트 인증서 관리	451
서버 인증에 대한 SSL 옵션 지정	452
클라이언트에 인증서 기반의 인증 구성	452
사용자 인증서 얻기 및 설치	452
인증서 기반의 클라이언트 인증에 대한 SSL 옵션 지정	455
클라이언트에 SASL DIGEST-MD5 사용	455
영역 지정	455
환경 변수 지정	456
ldapsearch 명령 예	456
클라이언트에 커버로스 SASL GSSAPI 사용	457
클라이언트 호스트에 Kerberos V5 구성	457
커버로스 인증에 대한 SASL 옵션 지정	457

SASL이 있는 GSSAPI를 사용하여 커버로스 인증 구성: 절차 예	458
12장 Pass Through Authentication 구현	473
Directory Server의 PTA 사용 방법	473
PTA 플러그 인 구성	475
플러그 인 구성 항목 작성	475
보안 연결을 사용하도록 PTA 구성	476
연결 매개 변수(선택 사항) 설정	476
여러 개의 서버 및 하위 트리 지정	477
PTA 플러그 인 구성 수정	478
13장 로그 파일을 사용하여 Directory Server 모니터	479
로그 파일 정책 정의	480
로그 파일 순환 정책 정의	480
로그 파일 삭제 정책 정의	480
수동 로그 파일 순환	481
액세스 로그	481
액세스 로그 보기	481
액세스 로그 구성	483
오류 로그	485
오류 로그 보기	485
오류 로그 구성	486
감사 로그	487
감사 로그 구성	487
감사 로그 보기	487
서버 작업 모니터	488
콘솔에서 서버 모니터	488
명령줄에서 서버 모니터	493
14장 SNMP를 사용하여 Directory Server 모니터	495
Sun Java System 서버의 SNMP	496
Directory Server MIB의 개요	497
SNMP 설정	498
Solaris 플랫폼의 경우	498
snmpdx SNMP 마스터 에이전트로 플러그 - Solaris 예	498
Linux 플랫폼의 경우	501
Linux의 SNMP 관련 힌트	502
Directory Server에서 SNMP 구성	506
SNMP 하위 에이전트 시작 및 중지	507
15장 속성 값 고유성 강화	509
개요	509

uid 속성에 대한 고유성 실행	510
콘솔에서 플러그인 구성	510
명령줄에서 플러그인 구성	511
다른 속성에 대한 고유성 실행	513
복제 시 고유성 플러그인 사용	514
단일 마스터 복제 시나리오	514
다중 마스터 복제 시나리오	515
16장 Directory Server 문제 해결	517
설치 및 로그인 문제 해결	517
부록 A Sun Crypto Accelerator 보드 사용	521
시작 전 주의 사항	521
토큰 작성	522
보드 바인딩 생성	523
인증서 가져오기	524
SSL 구성	524
부록 B 타사 사용권 내용	527
용어집	531

그림 목차

그림 1-1	Sun Java System 서버 콘솔	39
그림 1-2	Directory Server 콘솔의 태스크 탭	40
그림 1-3	Directory Server 콘솔의 구성 탭	41
그림 1-4	Directory Server 콘솔의 디렉토리 탭	43
그림 1-5	Directory Server 콘솔의 상태 탭	44
그림 2-1	Directory Server 콘솔 - 사용자 항목의 사용자 정의 편집기	66
그림 2-2	Directory Server 콘솔 - 일반 편집기	71
그림 3-1	3대의 서버를 사용한 계단식 연결	182
그림 6-1	상속 기능과 함께 userattr 키워드 사용	264
그림 6-2	액세스 제어 관리 대화 상자	272
그림 6-3	ACI 편집기 대화 상자	273
그림 6-4	매크로 ACI의 디렉토리 트리 예	306
그림 8-1	단일 마스터 복제	333
그림 8-2	허브를 사용한 계단식 복제	333
그림 8-3	다중 마스터 복제	334
그림 13-1	로그 내용 보기	482
그림 13-2	로그 파일 순환 및 삭제 구성 패널	484
그림 13-3	접미어 모니터 패널	491

표 목차

표 1	활자체 규약	27
표 2	자리 표시자 규칙	28
표 3	기호 규칙	28
표 4	웹 프롬프트	29
표 5	설명서, 지원 및 교육 링크	31
표 2-1	항목 템플릿 및 해당 객체 클래스	65
표 2-2	검색 필터 연산자	120
표 2-3	검색 필터 부울 연산자	121
표 2-4	검색 필터의 특수 문자	123
표 2-5	검색 유형, 연산자 및 접미어	128
표 3-1	기본적으로 연결이 허용되는 LDAP 컨트롤	168
표 3-2	연결 가능한 LDAP 컨트롤	169
표 4-1	데이터 가져오기와 접미어 초기화 비교	189
표 4-2	위의 예에 사용된 ldif2db 옵션에 대한 설명	193
표 4-3	위의 예에 사용된 ldif2db-task 옵션에 대한 설명	193
표 4-4	위의 예에 사용된 db2ldif 옵션에 대한 설명	197
표 5-1	역할 정의에 사용되는 객체 클래스 및 속성	217
표 5-2	CoS 정의 항목의 객체 클래스 및 속성	227
표 5-3	CoS 정의 항목 속성	227
표 6-1	LDIF target 키워드	244
표 6-2	LDIF 바인드 규칙 키워드	255
표 6-3	유효 권한 상호 종속성	302
표 6-4	유효 권한 로깅 정보 이유 및 설명	304
표 6-5	ACI 키워드의 매크로	308

표 8-1	Directory Server 콘솔 상태 탭의 복제 매개 변수	387
표 9-1	속성 탭의 테이블 열	398
표 9-2	속성 구문 정의	398
표 9-3	객체 클래스 탭의 필드	402
표 10-1	모든 접미어의 시스템 색인	411
표 10-2	모든 새 접미어의 기본 색인	412
표 10-3	속성의 기본 이름 및 별명	413
표 10-4	위의 예에 사용된 <code>vlindex</code> 옵션에 대한 설명	426
표 11-1	Directory Server에서 제공하는 암호	438
표 13-1	자원 요약 테이블	489
표 13-2	현재 자원 사용	489
표 13-3	연결 상태 테이블	490
표 13-4	항목 캐시 사용	491
표 13-5	데이터베이스 캐시 액세스	492
표 16-1	솔루션 사용시 일반 설치 및 로그인 문제	517
표 A-1	보드 사용을 위한 필수 사항	521

절차 목록

받는 LDAP 요청에 대해 Directory Server에서 수신하는 포트 또는 보안 포트를 수정하려면	50
디렉토리 항목의 일반 편집기를 호출하려면	70
속성 값을 수정하려면	72
콘솔에서 ACI 권한을 확장하려면	92
콘솔에서 DN 수정 작업을 전역적으로 활성화 또는 비활성화하려면	93
콘솔에서 지정된 접미어에 대해 DN 수정 작업을 활성화하려면	94
콘솔에서 항목의 이름을 바꾸려면	95
콘솔에서 항목을 이동하려면	96
ldapmodify 명령을 사용하여 DN 수정 작업을 전역적으로 활성화하려면	97
ldapmodify 명령을 사용하여 접미어에 대해 DN 수정 작업을 활성화하려면	97
ldapmodify 명령을 사용하여 리프 항목의 이름을 바꾸려면	98
ldapmodify 명령을 사용하여 항목을 이동하려면	98
ldapmodify 명령을 사용하여 항목의 이름을 바꾸고 이동하려면	99
Directory Server 5.2를 Directory Server 4.x의 소비자로 구성하려면	378
Directory Server 5.1 스키마를 업데이트하려면	380
Directory Server 콘솔을 사용하여 레트로 변경 로그 플러그 인을 활성화하려면	382
명령줄을 사용하여 레트로 변경 로그 플러그 인을 활성화하려면	383
Directory Server 콘솔을 사용하여 지정된 접미어에 대한 업데이트를 기록하도록 레트로 변경 로그를 구성하려면	383
명령줄을 사용하여 지정된 접미어에 대한 업데이트를 기록하도록 레트로 변경 로그를 구성하려면	384
Directory Server 콘솔을 사용하여 삭제된 항목의 속성을 기록하도록 레트로 변경 로그를 구성하려면	384
명령줄을 사용하여 삭제된 항목의 속성을 기록하려면 레트로 변경 로그를 구성하려면	385
snmpdx SNMP 마스터 에이전트로 플러그 - Solaris 예	498

코드 예 목록

코드 예 2-1	빈 익명 DSML 요청	131
코드 예 2-2	빈 익명 DSML 응답	133
코드 예 2-3	DSML 확장 작업: 특정 사용자로 바인드	134
코드 예 2-4	DSML 확장 작업에 대한 응답	135
코드 예 2-5	DSML 검색 요청	136
코드 예 2-6	DSML 검색 응답	138
코드 예 11-1	편집된 커버로스 클라이언트 구성 파일 /etc/krb5/krb5.conf	461
코드 예 11-2	편집된 Administration Server ACL 구성 파일	462
코드 예 11-3	편집된 KDC 서버 구성 파일 /etc/krb5/kdc.conf	462
코드 예 11-4	KDC 데이터베이스를 만드는 명령	463
코드 예 11-5	관리 사용자 및 키 탭을 만드는 명령	463
코드 예 11-6	Solaris 9에서 커버로스 데몬을 시작하는 명령	464
코드 예 11-7	Solaris 10에서 커버로스 데몬을 시작하는 명령	464
코드 예 11-8	호스트 사용자를 추가하는 명령	464
코드 예 11-9	KDC에 LDAP 사용자를 추가하는 명령	465
코드 예 11-10	KDC에 테스트 사용자를 추가하는 명령	465
코드 예 11-11	gssapi.ldif 파일 내용	466
코드 예 11-12	GSSAPI를 활성화하도록 Directory Server를 업데이트하는 명령	467
코드 예 11-13	Directory Server에 대한 키 탭을 만드는 명령	468
코드 예 11-14	키 탭 사용 권한 및 소유권을 바꾸는 명령	468
코드 예 11-15	사용자 정의 커버로스 키 탭을 사용하도록 업데이트된 start-slapd 스크립트	468
코드 예 11-16	Directory Server를 다시 시작하는 명령	469
코드 예 11-17	새 파일 testuser.ldif	469

코드 예 11-18	서버에 테스트 항목을 추가하는 명령	470
코드 예 11-19	커버로스 티켓을 받는 명령	470
코드 예 11-20	테스트 티켓에 대한 정보를 보는 명령	470
코드 예 11-21	SASL_PATH 환경 변수를 설정하는 명령	471
코드 예 11-22	GSSAPI를 통한 인증 후 dc=example,dc=com 항목 검색	471
코드 예 11-23	471
코드 예 14-1	SNMP Linux 스크립트 directorysnmp	502
코드 예 A-1	보드를 사용하여 SSL을 활성화하기 위한 수정 작업(ssl.ldif)	525

이 설명서에서는 Directory Server에 기반한 디렉토리 서비스를 구성하고 관리하는 데 필요한 절차를 설명합니다. 필요한 경우 콘솔과 명령줄을 사용하여 모든 Directory Server 기능을 구성하는 절차도 포함되어 있습니다.

Sun™ 설명서에 액세스하는 방법과 Sun 설명서를 사용하는 방법에 대한 자세한 내용은 다음 내용을 참조하십시오.

- [규칙](#)
- [관련 설명서](#)
- [설명서, 지원 및 교육](#)
- [타사 웹 사이트 관련 참조 사항](#)
- [사용자 의견 환영](#)

규칙

표 1은 본 설명서에 사용된 활자체 규약에 대해 설명합니다.

표 1 활자체 규약

글꼴	의미	예
AaBbCc123 (고정 폭 글꼴)	API 및 언어 요소, HTML 태그, 웹 사이트 URL, 명령 이름, 파일 이름, 디렉토리 경로 이름, 화면상의 컴퓨터 출력, 샘플 코드	.login 파일을 편집합니다. ls -a를 사용하면 모든 파일이 열거됩니다. % You have mail.
AaBbCc123 (굵은 고정 폭 글꼴)	화면 상의 컴퓨터 출력과는 반대로 사용자가 직접 입력하는 사항	% su Password:

표 1 활자체 규약(계속)

글꼴	의미	예
<i>AaBbCc123</i> (기울임꼴)	책 제목 새 단어 또는 용어 강조되는 단어 실제 이름이나 값으로 대체되는 명령 줄 변수	<i>Developer's Guide</i> 의 6장을 읽어 보십시오. 이것을 <i>클래스</i> 옵션이라고 합니다. 이 작업을 수행하려면 <i> 반드시 </i> 수퍼유저여야 합니다. 파일은 <i>ServerRoot</i> 디렉토리에 있습니다.

표 2에서는 본 설명서에 사용된 자리 표시자 규칙에 대해 설명합니다.

표 2 자리 표시자 규칙

항목	의미	예
<code>install-dir</code>	설치 후 소프트웨어 이진 파일이 위치하는 디렉토리 접두어를 나타내는 자리 표시자	Solaris 시스템의 기본 <code>install-dir</code> 접두어는 /입니다. Red Hat 시스템의 기본 <code>install-dir</code> 접두어는 /opt/sun입니다.
<code>ServerRoot</code>	서버 인스턴스와 데이터가 위치하는 디렉토리를 나타내는 자리 표시자 클라이언트측 서버 콘솔을 통해 <code>ServerRoot</code> 아래의 각 서버를 원격으로 관리할 수 있습니다. 서버 콘솔은 서버측 Administration Server를 사용하여 서버측 시스템에서 직접 실행해야 하는 작업을 수행합니다.	기본 <code>ServerRoot</code> 디렉토리는 /var/opt/mps/serverroot입니다.
<code>slapd-serverID</code>	특정 서버 인스턴스 및 관련 데이터가 <code>ServerRoot</code> 아래에 기본적으로 위치하는 디렉토리를 나타내는 자리 표시자	기본 <code>serverID</code> 는 호스트 이름입니다.

표 3에서는 본 설명서에 사용된 기호 규칙에 대해 설명합니다.

표 3 기호 규칙

기호	의미	표기법	예
[]	선택 사항인 명령 옵션이 포함됩니다.	<code>O[n]</code>	<code>O4, O</code>
{ }	필수 명령 옵션에 대한 선택 사항이 포함됩니다.	<code>d{y n}</code>	<code>dy</code>
	명령 옵션 선택 사항을 구분합니다.		

표 3 기호 규칙(계속)

기호	의미	표기법	예
+	그래픽 사용자 인터페이스에서 사용되는 단축 키의 동시 키 누르기를 결합합니다.		Ctrl+A
-	그래픽 사용자 인터페이스에서 사용되는 단축 키의 연속 키 누르기를 결합합니다.		Esc-S
>	그래픽 사용자 인터페이스의 메뉴 선택을 나타냅니다.		파일 > 새로 만들기 파일 > 새로 만들기 > 템플리트

표 4에서는 본 설명서에 사용된 셸 프롬프트 규칙에 대해 설명합니다.

표 4 셸 프롬프트

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

Directory Server 명령의 입력 및 출력은 대개 LDIF(LDAP Data Interchange Format)[RFC 2849]를 사용하여 표시합니다. 가독성을 높이기 위해 자동으로 줄바꿈됩니다.

관련 설명서

다음 설명서는 <http://www.sun.com/documentation/>에서 HTML 및 PDF 형식으로 확인할 수 있습니다.

Directory Server 설명서

Directory Server 릴리스 노트

Directory Server Technical Overview

Directory Server Deployment Planning Guide

Directory Server Installation and Migration Guide

Directory Server Performance Tuning Guide

Directory Server 관리 설명서

Directory Server Administration Reference

Directory Server Plug-in Developer's Guide

Directory Server Plug-in Developer's Reference

Directory Server Man Page Reference

Administration Server 설명서

Administration Server 릴리스 노트

Administration Server Administration Guide

Administration Server Man Page Reference

Directory Proxy Server 설명서

Directory Proxy Server 릴리스 노트

Directory Proxy Server 관리 설명서

관련된 Java Enterprise System 설명서

Java Enterprise System 설치 설명서

Java Enterprise System 업그레이드 및 마이그레이션 설명서

Java Enterprise System 용어집

설명서, 지원 및 교육

표 5는 Sun 설명서와 지원, 교육 정보에 대한 링크를 제공합니다.

표 5 설명서, 지원 및 교육 링크

정보	링크	설명
설명서	http://www.sun.com/documentation/	PDF 및 HTML 설명서를 다운로드하고 인쇄본 설명서를 주문합니다.
지원 및 교육	http://www.sun.com/supporttraining/	기술 지원을 받고 패치를 다운로드하며 Sun 교육 과정을 학습합니다.

타사 웹 사이트 관련 참조 사항

Sun은 이 설명서에 언급된 타사 웹 사이트의 가용성에 대해 책임지지 않습니다. Sun은 이러한 사이트나 자원을 통해 사용할 수 있는 내용, 광고, 제품 또는 기타 자료에 대해서는 보증하지 않으며 책임지지 않습니다. Sun은 해당 사이트 또는 자원을 통해 사용 가능한 내용, 제품 또는 서비스의 사용과 관련해 발생하거나 발생했다고 간주되는 손해나 손실에 대해 책임이나 의무를 지지 않습니다.

사용자 의견 환영

Sun은 설명서의 내용을 개선하기 위해 노력하고 있으며 사용자의 의견 및 제안을 환영합니다. 웹 기반 양식을 사용하여 의견을 보내 주십시오.

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

해당 필드에 전체 설명서 제목과 부품 번호를 기입해 주십시오. 부품 번호는 해당 설명서의 제목 페이지나 문서 맨 위에 있으며 일반적으로 7자리 또는 9자리 숫자입니다. 예를 들어, 이 설명서의 부품 번호는 819-2012입니다.

사용자의 의견을 제출할 때 해당 양식에 영문 설명서의 제목과 부품 번호를 입력해야 할 수도 있습니다. 본 설명서의 영문 제목과 부품 번호는 Sun Java(TM) System Directory Server 5.2 2005Q1 Administration Guide(817-7613)입니다.

사용자 의견 환영

Directory Server 관리 개요

Directory Server 제품은 Directory Server, 여러 디렉토리를 관리하는 Administration Server, 그래픽 인터페이스를 통해 두 서버를 모두 관리하는 서버 콘솔 등으로 구성되어 있습니다. 이 장에서는 Directory Server에 대한 개요를 제공하고 디렉토리 서비스 관리에 필요한 가장 기본적인 작업에 대해 설명합니다.

플러그인 서명 확인은 추가 보안 기능으로, 서버는 이 기능을 사용하여 권한 없는 플러그인을 감지하고 로드되지 않도록 차단할 수 있습니다. DSML(Directory Server Markup Language)은 요청을 디렉토리 서버로 보내기 위한 XML 기반의 새로운 형식입니다.

이 장은 다음 내용으로 구성되어 있습니다.

- [Directory Server 관리 개요](#)
- [Directory Server 시작 및 중지](#)
- [SSL을 활성화하여 서버 시작](#)
- [Directory Server 콘솔 사용](#)
- [LDAP 매개 변수 구성](#)
- [플러그인 서명 확인](#)
- [DSML 구성](#)

주 두 가지 버전 이상의 Directory Server를 실행할 경우 이 장의 모든 예에서는 Directory Server 5.2를 기본 버전으로 가정합니다. 그렇지 않은 경우 다음 명령을 한 번 실행하여 5.2를 기본 버전으로 설정해야 합니다.

```
# /usr/sbin/directoryserver -d 5.2
```

또는 directoryserver 명령을 실행할 때마다 `-useversion` 옵션을 사용하여 버전을 지정하십시오. 예를 들면 다음과 같습니다.

```
# /usr/sbin/directoryserver -useversion 5.2 start
```

Directory Server 관리 개요

Directory Server는 기업 전체의 사용자 및 자원 디렉토리를 관리하기 위한 강력하고 확장성이 뛰어난 서버로, LDAP(Lightweight Directory Access Protocol)라는 오픈 시스템 서버 프로토콜에 기반을 두고 있습니다. Directory Server는 사용자 시스템에서 `ns-slapd` 프로세스 또는 서비스로 실행됩니다. 디렉토리 내용을 관리하고 클라이언트의 요청에 응답합니다.

대부분의 Directory Server 관리 작업은 Sun Java System에서 Directory Server 및 기타 모든 Sun Java System 서버의 관리용 보조 서버로 제공되는 관리 서버를 통해 수행됩니다. 서버 콘솔은 관리 서버에 대한 그래픽 인터페이스입니다. *Directory Server* 콘솔은 특히 Directory Server용으로 설계된 서버 콘솔의 한 구성 요소입니다.

대부분의 Directory Server 관리 작업은 Directory Server 콘솔에서 수행할 수 있습니다. 구성 파일을 편집하거나 명령줄 유틸리티를 사용하여 수동으로 관리 작업을 수행할 수도 있습니다. 서버 콘솔에 대한 자세한 내용은 *Administration Server Administration Guide*를 참조하십시오.

주 Directory Server 데이터 서비스용 Sun Cluster HA를 사용할 경우 Directory Server를 관리할 때 명령줄에서 `directoryserver(1M)` 명령 및 해당 하위 명령을 사용해야 합니다.

독립형 스크립트와 이진을 직접 사용하지 마십시오.

Directory Server 시작 및 중지

SSL(Secure Sockets Layer)을 사용하지 않는 경우 여기에 소개된 방법으로 Directory Server를 시작하고 중지할 수 있습니다. SSL을 사용 중이면 [36페이지의 "SSL을 활성화하여 서버 시작"](#)을 참조하십시오.

명령줄에서 서버 시작 및 중지

명령줄에서 서버를 시작하거나 중지하려면 다음을 수행합니다. 다음 명령을 실행합니다.

```
# /usr/sbin/directoryserver -useversion 5.2 start
```

또는

```
# /usr/sbin/directoryserver -useversion 5.2 stop
```

Directory Server 5.2가 기본 버전이 아닐 경우에만 useversion 옵션이 필요합니다. directoryserver 명령의 전체 구문을 보려면 *Directory Server Man Page Reference*를 참조하십시오.

Directory Server에 대한 UID로 루트를 지정한 경우 이 명령은 루트로서 실행되어야 합니다.

그렇지 않으면 두 명령은 루트로 실행되거나 Directory Server와 동일한 UID와 GID를 사용하여 실행되어야 합니다. 예를 들어, Directory Server가 nobody로 실행되면 start 유틸리티와 stop 유틸리티도 nobody로 실행해야 합니다.

이전 버전의 Directory Server 사용자의 경우 더 이상 참조 모드에서 서버를 시작할 수 없습니다. Directory Server 콘솔을 사용하여 전역 참조를 설정할 수 있습니다. 이 절차에 대해서는 [100페이지의 "기본 참조 설정"](#)에서 설명합니다.

콘솔에서 서버 시작 및 중지

Directory Server 콘솔을 실행할 경우 그래픽 인터페이스를 통해 Directory Server를 시작, 중지 및 다시 시작할 수 있습니다. 콘솔 실행에 대한 자세한 내용은 [38페이지의 "Directory Server 콘솔 시작"](#)을 참조하십시오.

1. Directory Server 콘솔의 최상위 태스크 탭에서 "Directory Server 시작", "Directory Server 중지" 또는 "Directory Server 다시 시작" 버튼을 누릅니다.

Directory Server 콘솔에서 성공적으로 Directory Server를 시작하거나 중지하면 서버 시작 또는 서버 종료를 알리는 메시지 대화 상자가 콘솔에 표시됩니다. 오류가 발생한 경우에는 해당 오류에 관한 모든 메시지가 콘솔에 표시됩니다.

SSL을 활성화하여 서버 시작

SSL을 활성화하기 전에 서버에 인증서를 설치 및 구성해야 합니다. 인증서를 관리하고 SSL을 활성화하는 방법에 대한 지침은 [11장, "인증 및 암호화 관리"](#)를 참조하고 인증서와 인증서 데이터베이스, 서버 인증서를 얻는 방법에 대한 자세한 내용은 *Administration Server Administration Guide*를 참조하십시오.

SSL을 활성화한 상태에서 서버를 시작하려면 명령줄에서 서버를 시작하고 서버의 인증서를 보호하는 비밀번호를 제공해야 합니다.

또는 인증서 비밀번호를 저장할 비밀번호 파일을 작성할 수 있습니다. 인증서 데이터베이스 비밀번호를 파일에 저장하면 서버 콘솔에서 서버를 시작할 수 있을 뿐만 아니라 무인 작동 시 서버가 자동으로 다시 시작될 수 있습니다.

주의 이 비밀번호는 일반 텍스트로 비밀번호 파일에 저장되기 때문에 보안상 상당한 위험을 초래할 수 있습니다. 안전하지 않은 환경에서 서버를 실행하는 경우에는 비밀번호 파일을 사용하지 마십시오.

비밀번호 파일은 아래 위치에 저장해야 합니다.

```
ServerRoot/alias/slaped-serverID-pin.txt
```

여기서 *serverID*는 서버를 설치할 때 지정한 서버 식별자입니다.

다음과 같은 형식으로 보안 토큰 이름과 비밀번호를 파일에 추가해야 합니다.

```
tokenName:password
```

이 예의 경우, 다음과 같이 내부 인증서 데이터베이스의 장치 이름이 표시됩니다. 대소문자와 공백까지 표시된 것과 정확히 일치해야 합니다.

```
Internal (Software) Token:password
```

대체 장치에 인증서를 저장하는 경우 인증서 관리 대화 상자의 맨 위에 있는 드롭다운 메뉴에 표시된 장치 이름을 사용합니다. 인증서 데이터베이스를 작성하려면 관리 서버와 인증서 설정 마법사를 사용해야 합니다.

SSL을 활성화하여 서버를 시작하지만 비 SSL 포트에서의 서버 수신을 방지하려면 `nssldap-listenhosts`를 루프백 주소 127.0.0.1로 설정할 수 있습니다. 또는 `nsslapd-port` 속성을 0으로 설정할 수도 있는데, 이 경우 `monitor`, `db2bak.p1` 및 `ldif2db.p1`과 같은 관리 스크립트를 손상시킬 수 있습니다.

Directory Server에서 SSL을 사용하는 방법은 [11장, "인증 및 암호화 관리"](#)를 참조하십시오.

1024보다 적은 포트에서 비루트 사용자로 서버 시작

일반적으로 포트가 1024보다 적을 때 Directory Server를 시작하려면 루트 사용자에게서 권한을 넘겨야 합니다. 다음 절차를 사용하면 지정된 비루트 사용자를 대신 사용할 수 있습니다.

1. Directory Server 및 Administration Server를 설치합니다. 서버를 구성할 때는 두 서버의 사용자가 루트인지 확인하도록 합니다.

*Java Enterprise System 설치 설명서*의 설치 지침을 사용하십시오.

2. Directory Server를 중지합니다. [35페이지의 "Directory Server 시작 및 중지"](#)를 참조하십시오.
3. *ServerRoot* 디렉토리에서 다음 명령을 실행하여 디렉토리 및 파일 소유권을 필요한 *userID*로 바꿉니다.

```
chown -R userID:groupID slapd-hostname
```

```
chown -R userID:groupID alias/slapd-hostname-*.db
```

4. *dse.ldif* 파일을 편집하여 *nsslapd-localuser* 값을 루트에서 필요한 *userID*로 바꿉니다.
5. Directory Server를 다시 시작합니다. [35페이지의 "Directory Server 시작 및 중지"](#)를 참조하십시오.

Directory Server 콘솔 사용

Directory Server 콘솔은 별도의 서버 콘솔 창으로 액세스되는 인터페이스입니다. 다음 절차에 설명된 것처럼 Directory Server 콘솔은 서버 콘솔에서 시작합니다.

Directory Server 콘솔 시작

1. Directory Server 데몬인 `slapd-serverID`가 실행되고 있는지 확인합니다. 실행되고 있지 않으면 루트 또는 관리 권한이 있는 사용자로 아래 명령을 실행하여 시작합니다.

```
# /usr/sbin/directoryserver -useversion 5.2 start
```

2. 관리 서버 데몬인 `ns-httpd`가 실행되고 있는지 확인합니다. 실행되고 있지 않으면 루트 또는 관리 권한이 있는 사용자로 아래 명령을 실행하여 시작합니다.

```
# /usr/sbin/directoryserver -useversion 5.2 start-admin
```

3. 아래 명령을 입력하여 서버 콘솔을 시작합니다.

```
# /usr/sbin/directoryserver -useversion 5.2 startconsole
```

서버 콘솔과 Administration Server를 각각 다른 시스템에서 실행하는 경우 *Administration Server Administration Guide*에 설명된 것처럼 Administration Server에서 연결 제한을 구성해야 할 수도 있습니다.

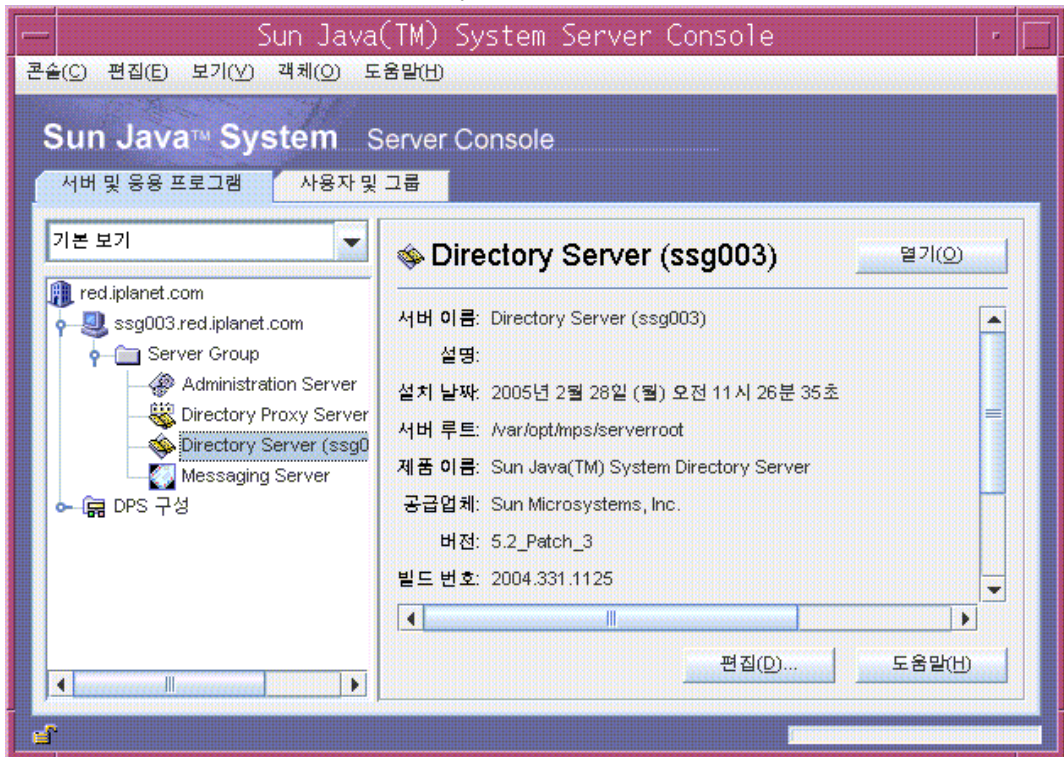
콘솔 로그인 창이 표시됩니다. 구성 디렉토리(`o=NetscapeRoot` 접미어가 포함된 디렉토리)가 Directory Server의 개별 인스턴스에 저장되는 경우에는 해당 디렉토리 서버의 관리 권한이 있는 사용자 DN과 비밀번호, 그리고 관리 서버 URL을 요청하는 창이 표시됩니다.

4. 수행할 작업에 대한 충분한 액세스 권한이 있는 사용자의 바인드 DN과 비밀번호를 사용하여 로그인합니다.

서버 콘솔이 표시됩니다.

5. 왼쪽 패널의 트리를 탐색하여 Directory Server를 호스트하는 시스템을 찾은 다음, 해당 이름이나 아이콘을 눌러 일반 등록 정보를 표시합니다.

그림 1-1 Sun Java System 서버 콘솔



Directory Server의 이름과 설명을 편집하려면 편집 버튼을 누른 다음 텍스트 상자에 새 이름과 설명을 입력합니다. 확인을 눌러 새 이름과 설명을 설정합니다. 앞의 그림과 같이 왼쪽 트리에 새 이름이 표시됩니다.

6. 트리에서 Directory Server 이름을 두 번 누르거나 열기버튼을 눌러 이 디렉토리 서버를 관리하기 위한 Directory Server 콘솔을 표시합니다.

Directory Server 콘솔 시작

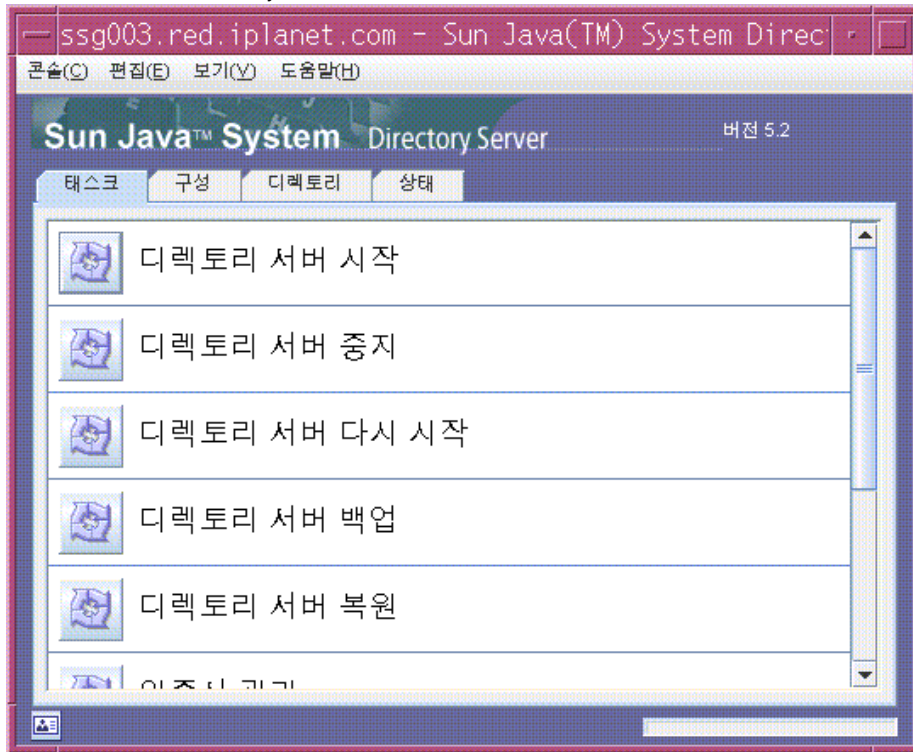
Directory Server 콘솔은 Directory Server 인스턴스에서 탐색 및 관리 작업을 수행하기 위한 인터페이스를 제공합니다. 콘솔에는 네 개의 탭이 있으며, 이 탭을 사용하여 모든 Directory Server 기능에 액세스할 수 있습니다.

- 태스크 탭 - 서버 다시 시작과 같은 관리 작업 버튼이 포함되어 있습니다.
- 구성 탭 - 서버 관리를 위한 모든 매개 변수에 대한 액세스를 제공합니다.
- 디렉토리 탭 - 디렉토리에 있는 모든 데이터 항목을 표시하고 편집합니다.
- 상태 탭 - 서버의 통계, 로그 및 복제 상태를 표시합니다.

태스크 탭

태스크 탭은 Directory Server 콘솔을 열 때 표시되는 첫 인터페이스로, 아래 그림과 같이 Directory Server 시작 또는 중지와 같은 중요한 관리 작업을 위한 모든 버튼이 포함되어 있습니다. 모든 작업과 해당 버튼을 보려면 목록을 스크롤해야 할 수도 있습니다.

그림 1-2 Directory Server 콘솔의 태스크 탭



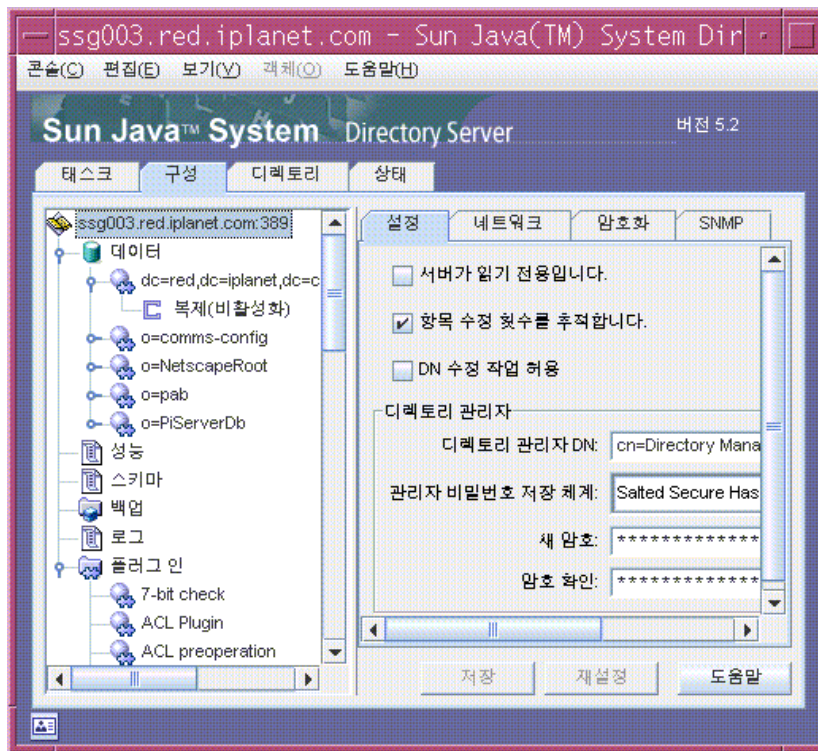
이 작업을 수행하려면 관리자 권한이 있는 사용자로 로그인해야 합니다. 권한이 충분하지 않은 사용자에게는 작업 버튼이 표시되지 않습니다.

구성 탭

Directory Server 콘솔의 구성 탭은 접미어, 복제, 스키마, 로그, 플러그인 등과 같은 모든 디렉토리 설정을 보고 수정할 수 있는 인터페이스와 대화 상자를 제공합니다. 대화 상자는 관리자 권한이 있는 사용자로 로그인한 경우에만 사용할 수 있거나 적용됩니다.

이 탭의 왼쪽에는 모든 구성 기능이 표시된 트리があり 오른쪽에는 각 기능을 관리하기 위한 특정 인터페이스가 표시됩니다. 인터페이스에 다른 탭이나 대화 상자 또는 팝업 창이 표시되는 경우도 있습니다. 예를 들어 아래 그림은 전체 디렉토리의 일반 설정을 보여줍니다.

그림 1-3 Directory Server 콘솔의 구성 탭



왼쪽 트리에서 구성 가능한 항목을 선택하면 오른쪽 패널에 있는 한 개 이상의 탭에 이 항목의 현재 설정이 표시됩니다. 설정에 대한 자세한 설명과 기능은 본 설명서에서 해당 기능을 설명하는 장을 참조하십시오. 설정에 따라 저장 후 즉시 적용되는 변경 사항도 있고 서버를 다시 시작해야만 적용되는 변경 사항도 있습니다. 서버를 다시 시작해야 하는 경우에는 이를 알려주는 대화 상자가 콘솔에 표시됩니다.

탭에 저장되지 않은 변경 사항이 있으면 탭 이름 옆에 빨간색 표시가 나타납니다. 변경 사항을 저장하지 않고 다른 항목을 구성하거나 다른 탭 중 하나를 표시해도 이 변경 사항은 해당 탭에 그대로 남아 있습니다. 저장 버튼과 재설정 버튼은 구성 가능한 특정 항목의 모든 탭에 적용되지만 다른 항목의 저장되지 않은 설정에는 영향을 주지 않습니다.

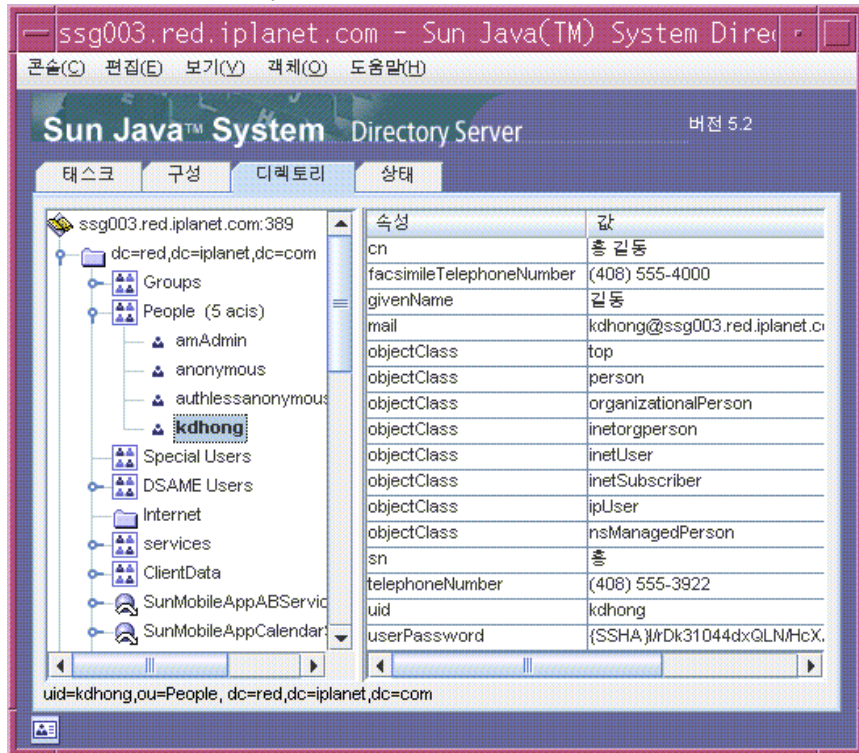
대부분의 텍스트 필드에는 해당 설정에 적합한 구문의 값만 입력할 수 있습니다. 올바른 구문으로 값을 입력할 때까지 기본적으로 설정 레이블과 값은 빨간색으로 강조 표시됩니다. 모든 설정의 구문이 정확해야만 저장 버튼이 활성화됩니다. [47페이지의 "시각적 구성 기본 설정"](#)에 설명된 것처럼 잘못된 값을 기울임꼴로 강조 표시하도록 선택할 수도 있습니다.

디렉토리 탭

콘솔의 디렉토리 탭은 쉽게 탐색할 수 있도록 디렉토리 항목을 트리로 표시합니다. 이 탭에서 모든 항목과 해당 속성을 탐색하거나 표시하고 편집할 수 있습니다.

주 수천 개의 항목 목록을 찾아보려면 더 빠른 액세스를 위해 찾아보기 색인을 만듭니다. 자세한 내용은 [423페이지의 "콘솔에 대한 찾아보기 색인"](#)을 참조하십시오.

그림 1-4 Directory Server 콘솔의 디렉토리 탭



로그인 중에 입력한 바인드 DN에 충분한 액세스 권한이 있으면 구성 항목은 일반 항목으로 표시되고 직접 수정할 수 있습니다. 하지만 안전하게 구성 설정을 변경하려면 반드시 구성 탭을 눌렀을 때 나타나는 대화 상자를 사용해야 합니다.

보기 메뉴에서 디렉토리 탭의 레이아웃과 내용을 변경하는 여러 가지 옵션을 사용할 수 있습니다. 새 레이아웃 옵션에는 리프 항목을 비롯한 모든 항목을 단일 트리에서 보기 및 오른쪽 패널에 속성 표시가 있습니다. 왼쪽이 아닌 오른쪽 트리에 리프 항목이 표시되는 것이 기본값입니다.

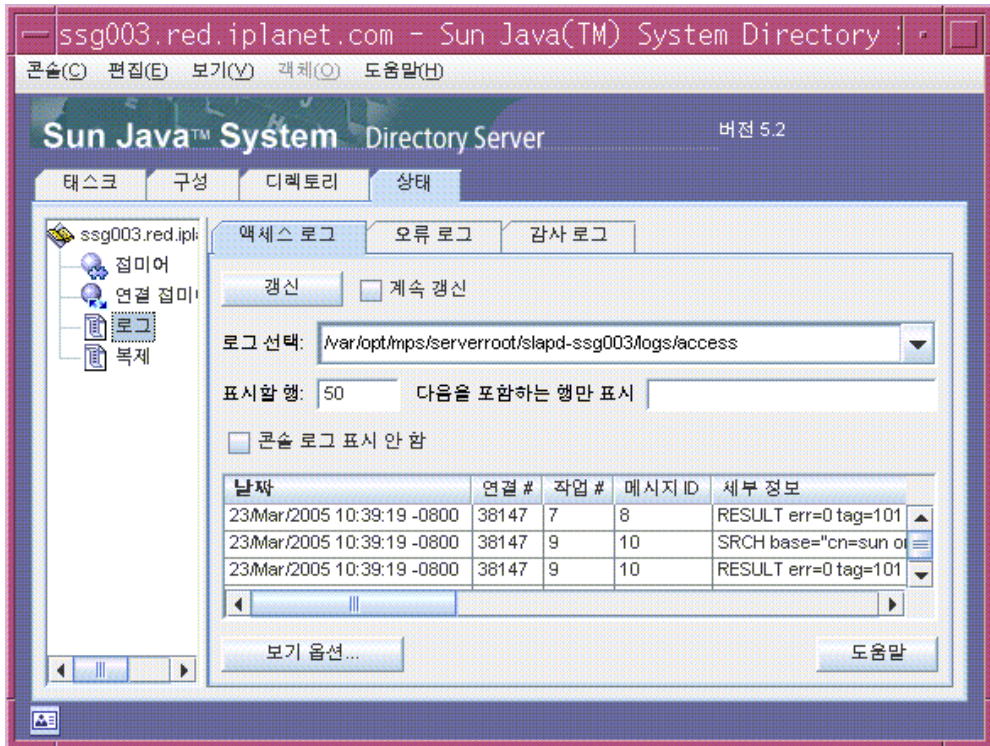
보기 > 표시 옵션을 사용하면 디렉토리 트리의 모든 항목에 대한 ACI 개수, 역할 개수 및 비활성화 상태 아이콘이 활성화됩니다. 그림 1-4에서는 왼쪽 트리에 ACI 개수와 리프 항목이 표시되고 오른쪽 패널에 선택한 항목의 속성 값이 표시됩니다. 자세한 내용은 47페이지의 "디렉토리 트리 보기 옵션"을 참조하십시오.

주 디렉토리 탭의 왼쪽 창에서 작업할 때 노란색 삼각형이 표시되는 경우는 오류가 발생했다는 뜻입니다. 발생한 오류에 대한 자세한 내용을 보려면 Directory Server 콘솔의 상태 탭에서 액세스 로그를 확인하십시오. 오른쪽 창에서 작업하는 동안 오류가 발생하는 경우에는 오류 경고가 표시되지 않습니다.

상태 탭


상태 탭에는 서버 통계와 로그 메시지가 표시됩니다. 왼쪽 트리에는 모든 상태 항목이 표시되고 오른쪽 패널에는 선택한 각 항목의 내용이 표시됩니다. 예를 들어 아래 그림은 로그 항목 테이블을 보여줍니다.

그림 1-5 Directory Server 콘솔의 상태 탭



콘솔에서 현재 바인드 DN 보기

디스플레이의 왼쪽 아래에 있는 로그인 아이콘을 누르면 Directory Server 콘솔에 로그인할 때 사용한 바인드 DN을 볼 수 있습니다. 여기에 표시된 것처럼 로그인 아이콘 옆에 현재 바인드 DN이 표시됩니다.



로그인 아이디 변경

Directory Server 콘솔에서 항목을 작성하거나 관리할 때, 그리고 서버 콘솔에 처음 액세스할 때 바인드 DN과 비밀번호를 입력하여 로그인할 수 있습니다. 이러한 로그인을 통해 누가 디렉토리 트리에 액세스하는지, 그리고 작업을 수행하는 데 필요한 액세스 권한이 있는지 확인됩니다.

처음 서버 콘솔을 시작할 때 디렉토리 관리자 DN으로 로그인한 다음 콘솔을 중지하고 다시 시작할 필요 없이 언제든지 다시 다른 사용자로 로그인할 수 있습니다.

서버 콘솔에서 로그인을 변경하려면 다음을 수행합니다.

1. Directory Server 콘솔에서 태스크 탭을 선택하고 "새 사용자로 Directory Server에 로그인" 레이블 옆의 버튼을 누릅니다. 또는 다른 콘솔 탭에 있는 경우 콘솔 > 새 사용자로 로그인 메뉴 항목을 선택합니다.

로그인 대화 상자가 표시됩니다.

2. 새로운 DN과 비밀번호를 입력하고 확인을 누릅니다.

서버에 바인드할 때 사용할 항목의 전체 DN을 입력하십시오. 예를 들어, 디렉토리 관리자로 바인드하려면 고유 이름 텍스트 상자에 아래 DN을 입력합니다.

```
cn=Directory Manager
```

온라인 도움말 사용

온라인 도움말은 Directory Server 콘솔의 탭과 대화 상자에 대한 상황에 맞는 정보를 제공합니다. 도움말 버튼은 대체로 이러한 인터페이스의 오른쪽 아래에 있습니다. 모든 화면에서 상황에 맞는 도움말을 실행할 때는 항상 Alt-P를 단축키로 사용할 수 있습니다.

온라인 도움말을 실행하면 콘솔의 기본 브라우저에 HTML 기반의 페이지가 표시됩니다. 여기서 브라우저 시작 버튼을 눌러 동일한 페이지를 Mozilla와 같은 외부 브라우저에서 열 수도 있습니다. 온라인 도움말 내에서 추가 정보에 대한 링크를 눌러도 외부 브라우저 창이 열립니다.

각 온라인 도움말 페이지에는 해당 탭이나 대화 상자에 있는 필드 및 버튼에 대한 설명이 나와 있습니다. 콘솔을 통해 값을 평가하거나 입력 또는 수정하는 경우 이 정보를 참조하십시오.

Directory Server의 도움말 시스템은 Administration Server에 기반을 두고 있습니다. Administration Server의 원격 시스템에서 Directory Server 콘솔을 실행 중이면 다음과 같은 사항을 확인해야 합니다.

- 컴퓨터의 액세스를 허용하기 위해 Administration Server에 적용되는 연결 제한을 구성해야 할 경우도 있습니다. 이는 *Administration Server Administration Guide*에 설명되어 있습니다.
- 브라우저에 프록시 사용이 구성되어 있는 경우, 외부 브라우저를 사용하여 온라인 도움말 페이지를 보려면 다음 중 하나를 수행해야 합니다.
 - 브라우저 구성에서 프록시를 비활성화합니다. Mozilla에서는 편집 > 환경 설정 메뉴 항목을 선택합니다. 그런 다음, 고급 > 프록시 범주를 선택하여 프록시 구성에 액세스합니다. Internet Explorer에서는 도구 메뉴에서 인터넷 옵션을 선택합니다.
 - Administration Server에서 프록시 서버의 액세스를 허용하도록 연결 제한을 구성합니다.

주의 Administration Server에서 프록시 서버의 액세스를 허용하도록 구성하면 시스템에 잠재적 보안 허점이 발생합니다.

콘솔 클립보드

Directory Server 콘솔은 시스템 클립보드를 사용하여 텍스트를 복사하거나 잘라내어 붙여넣습니다. 입력을 줄이려면 디렉토리 탭에서 탐색할 때 항목의 DN이나 URL을 클립보드로 복사할 수 있습니다.

DN 또는 URL을 붙여넣어야 하는 텍스트 필드가 있는 대화 상자나 다른 탭을 열기 전에 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 트리를 탐색하여 해당 DN이나 URL을 복사하려는 항목을 선택합니다.
2. 메뉴에서 편집 > DN 복사 또는 편집 > URL 복사를 선택합니다.

콘솔 설정

Directory Server 콘솔은 구성 및 디렉토리 탭에 정보를 표시하는 방법을 사용자 정의하기 위한 다양한 설정을 제공합니다.

시각적 구성 기본 설정

구성 매개 변수를 수정하고 최상위 구성 탭에 있는 필드에 값을 입력하는 경우 Directory Server 콘솔은 컬러 텍스트를 사용하여 입력이 유효한지 여부를 표시합니다. 예를 들어, 추가 구성 값을 입력해야 하는 기능을 활성화하면 필수 필드의 레이블이 빨간색으로 표시되었다가 유효한 값을 입력한 후에 다시 파란색으로 바뀝니다.

기본적으로 콘솔은 빨간색과 파란색을 사용하지만 다음과 같이 이 동작을 수정할 수도 있습니다.

1. Directory Server 콘솔의 임의의 탭에서 편집 > 기본 설정 메뉴 항목을 선택합니다. 콘솔 기본 설정 대화 상자에서 기타 탭을 선택합니다.
2. 원하는 시각적 구성 표시기에 해당하는 라디오 버튼을 선택합니다. 텍스트 색상이나 글꼴, 또는 둘 모두를 선택할 수 있습니다.
3. 콘솔 기본 설정 대화 상자의 다른 탭 설정에 대한 설명을 보려면 *Administration Server Administration Guide*를 참조하십시오.
확인을 눌러 변경 사항을 저장합니다.
4. 서버 콘솔의 모든 창을 종료한 후 콘솔을 다시 시작합니다.

디렉토리 트리 보기 옵션

Directory Server 콘솔의 최상위 디렉토리 탭에서 보기 메뉴 항목을 사용하여 디렉토리 트리에 추가 정보를 표시하고 오른쪽 패널에 표시할 내용을 선택할 수 있습니다.

디렉토리 탭의 내용은 다음과 같은 보기 옵션에 영향을 받습니다.

- 참조 따름 - 이 확인란을 선택하면 참조 대상의 모든 자식과 항목이 실제 디렉토리에 있는 것처럼 디렉토리 트리에 표시됩니다. 확인란을 선택 취소하면 참조는 참조 항목으로 표시됩니다. 자세한 내용은 [101페이지의 "스마트 참조 작성"](#)을 참조하십시오.
- 객체 정렬 - 이 확인란을 선택 취소하면 항목은 서버에서 반환된 순서대로 표시됩니다. 확인란을 선택하면 디렉토리 트리에서 같은 수준에 있는 항목은 아래에 설명된 해당 표시 속성에 따라 정렬됩니다. 서버 성능에 영향을 주지 않고 대규모 하위 트리를 정렬하는 방법은 [423페이지의 "콘솔에 대한 찾아보기 색인"](#)을 참조하십시오.

cn, givenname, o, ou, sn, uid 속성으로 표시되는 항목이 정렬됩니다. 기타 속성으로 표시되는 항목은 정렬되지 않습니다.

- 표시 > ACI 개수 - 항목의 aci 속성에 하나 이상의 ACI(Access Control Instruction)가 포함되어 있으면 디렉토리 트리에서 해당 항목 옆에 개수가 표시됩니다. 자세한 내용은 [6장, "액세스 제어 관리"](#)를 참조하십시오.
- 표시 > 역할 개수 - 항목이 하나 이상의 역할의 구성원이면 디렉토리 트리에서 해당 항목 옆에 개수가 표시됩니다. 자세한 내용은 [210페이지의 "역할 할당"](#)을 참조하십시오.
- 표시 > 비활성화 상태 - 사용자 또는 그룹 항목이 비활성화되어 서버에 바인드할 수 없으면 디렉토리 트리에서 해당 항목의 아이콘에 빨간색 상자와 선이 표시됩니다. 자세한 내용은 [326페이지의 "사용자와 역할 비활성화 및 활성화"](#)를 참조하십시오.
- 레이아웃 > 자식 보기 - 이 레이아웃 옵션을 선택하면 왼쪽 패널의 트리에 디렉토리의 리프 항목이 표시되지 않고, 왼쪽 패널에서 부모 노드를 선택하면 오른쪽 패널에 리프 항목을 비롯한 모든 자식이 표시됩니다. 두 패널에서 모두 항목을 선택할 수 있습니다.
- 레이아웃 > 트리만 보기 - 이 레이아웃 옵션을 선택하면 디렉토리 탭에는 디렉토리의 모든 항목이 포함된 트리가 있는 한 개의 패널만 표시됩니다.
- 레이아웃 > 속성 보기 - 이 레이아웃에서는 왼쪽 패널에 디렉토리의 모든 항목이 포함된 트리가 표시되고 오른쪽 패널에는 트리에서 선택한 항목에 저장된 속성과 값이 표시됩니다.
- 표시 속성 - 이 메뉴 항목을 누르면 "표시 속성" 대화 상자가 열리고 디렉토리 탭에 표시된 속성의 레이블을 선택할 수 있습니다. 기본적으로 항목의 첫 RDN 속성 값(예: People)이 레이블로 지정됩니다. RDN이 없는 기본 항목의 레이블은 전체 DN(예: dc=example, dc=com)입니다.

다른 속성을 사용하여 디렉토리 트리의 항목을 표시하려면 다른 라디오 버튼을 선택한 다음 원하는 속성을 선택합니다. 선택한 속성이 없는 항목은 계속해서 첫 RDN 속성을 사용합니다. 기본적으로 레이블에는 속성 값만 사용됩니다. "속성 이름 표시" 확인란을 선택하면 레이블은 ou=People과 같이 표시됩니다.
- 갱신 - 특정 작업 후에 새 값을 표시하려면 디렉토리 트리 디스플레이를 갱신해야 합니다. 이 항목을 선택하면 서버의 전체 디렉토리 트리가 다시 로드됩니다.

LDAP 매개 변수 구성

LDAP 매개 변수는 디렉토리 관리자의 고유 이름(DN), 전역 읽기 전용 속성, 포트 구성, 모든 디렉토리 수정 횟수를 추적하는 기능 등 디렉토리 서버의 기본 설정입니다.

디렉토리 관리자 구성

*디렉토리 관리자*는 UNIX에서의 루트 사용자에게 해당하는 권한이 있는 서버 관리자입니다. 디렉토리 관리자로 정의한 항목에는 액세스 제어가 적용되지 않습니다. 처음에 설치 도중 이 항목을 정의했습니다. 기본값은 `cn=Directory Manager`입니다.

디렉토리 관리자의 DN은 `cn=config` 분기의 `nsslapd-rootDN` 속성에, 비밀번호는 `nsslapd-rootpw` 속성에 저장됩니다.

Directory Server 콘솔을 사용하면 디렉토리 관리자 DN과 비밀번호, 이 비밀번호에 사용된 저장소 체계를 변경할 수 있습니다.

1. 디렉토리 관리자로 콘솔에 로그인합니다.
 - 이미 콘솔에 로그인되어 있으면 [45페이지의 "로그인 아이디 변경"](#)에서 다른 사용자로 로그인하는 방법을 참조하십시오.
2. 최상위 구성 탭에 있는 탐색 트리의 루트에서 서버 노드를 선택한 다음 오른쪽 패널에서 설정 탭을 선택합니다.
3. 디렉토리 관리자 DN 필드에 새 고유 이름(DN)을 입력합니다. 기본값은 설치 중에 정의한 DN입니다.
4. 관리자 비밀번호 암호화 풀다운 메뉴에서 디렉토리 관리자 비밀번호를 저장하는 데 사용할 서버의 저장소 체계를 선택합니다.
5. 제공된 텍스트 필드를 사용하여 새 비밀번호를 입력하고 확인합니다.
6. 저장을 누릅니다.

Directory Server의 포트 번호 변경

Directory Server 콘솔을 사용하거나 `cn=config` 항목의 `nsslapd-port` 또는 `nssldap-secureport` 속성의 값을 변경하여 사용자 디렉토리 서버의 포트 또는 보안 포트 번호를 수정할 수 있습니다.

Sun Java System 구성 정보(o=NetscapeRoot 하위 트리)가 저장되는 Directory Server 포트 또는 보안 포트를 수정하려면 Directory Server 콘솔을 사용합니다.

구성 디렉토리 또는 사용자 디렉토리 포트(또는 보안 포트) 번호를 변경하는 경우 다음과 같은 점에 주의해야 합니다.

- 관리 서버에 구성된 사용자 디렉토리 포트 또는 보안 포트 번호나 구성을 변경해야 합니다. *Administration Server Administration Guide*를 참조하십시오.
- 이 구성 디렉토리나 사용자 디렉토리를 가리키는 다른 Sun Java System 서버가 설치되어 있으면 해당 서버도 새 포트 번호를 가리키도록 업데이트해야 합니다.
- Directory Server가 다른 서버에 정의되어 있는 복제 계약에 언급되어 있는 경우에는 해당 복제 계약을 업데이트하여 새 포트 또는 보안 포트 번호를 사용하도록 해야 합니다.
- 권한이 없는 포트 번호를 설정하고, Directory Server를 다른 사용자가 액세스 권한을 갖고 있는 시스템에 설치한 경우 다른 응용 프로그램에서 해당 포트를 임의로 사용할 수 있는 위험이 있습니다. 즉 다른 응용 프로그램에서 동일한 주소/포트 쌍을 바인드할 수 있습니다. 그렇게 될 경우 이 응용 프로그램에서 Directory Server에 대한 요청을 처리할 수 있게 되며, 이 응용 프로그램을 사용하여 인증 프로세스에서 사용된 비밀번호를 수집하여 클라이언트 요청이나 서버 응답을 변경하거나 서비스 거부 공격을 수행할 수 있습니다. 이러한 보안 위험을 방지하려면 nsslapd-listenhost 속성을 사용하여 Directory Server가 수신하는 인터페이스(주소)를 지정합니다. 이 속성에 대한 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.
- 콘솔을 통해 포트 번호를 바꾼다고 해서 일부 스크립트에 필요한 변경 작업이 수행되는 것은 아닙니다. bak2db.pl, schema_push.pl, db2bak.pl, check-slapd, db2index.pl, db2ldif.pl, monitor, ldif2db.pl, ns-accountstatus.pl, ldif2ldap, ns-activate.pl, ns-inactivate.pl 스크립트는 원래 포트 번호에 대해 하드 코드가 유지되므로 수동으로 수정해야 합니다.

여기에 주어진 스크립트 이름은 독립형 도구 이름이며 check-slapd 명령은 공개적으로 노출된 API의 일부가 아니므로 문서화되지 않았습니다. 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.

▶ 받는 LDAP 요청에 대해 Directory Server에서 수신하는 포트 또는 보안 포트를 수정하려면

DSML 요청을 수신하는 포트를 수정하려면 55페이지의 "DSML 구성"을 참조하십시오.

1. Directory Server 콘솔의 최상위 구성 탭에서 이 서버 이름의 루트 노드를 선택한 다음 오른쪽 패널에서 네트워크 탭을 선택합니다.
LDAP 프로토콜에 대한 서버의 현재 포트 설정이 탭에 표시됩니다.
2. 서버에서 비SSL 통신에 사용할 포트 번호를 포트 필드에 입력합니다. 기본값은 389입니다.
3. 11장, "인증 및 암호화 관리"에 설명된 것처럼 이 서버에서 SSL을 활성화한 경우 다음과 같이 보안 포트에서 연결을 허용할 수도 있습니다.
 - a. 보안 포트와 비보안 포트를 모두 사용하는 옵션을 선택합니다.
 - b. 서버에서 SSL 통신에 사용할 포트 번호를 보안 포트 필드에 입력합니다. 기본값은 636입니다.
일반적인 LDAP 통신에 사용 중인 포트 번호를 암호화된 포트 번호로 지정해서는 안 됩니다.
4. 저장을 누른 후 서버를 다시 시작합니다.
자세한 내용은 35페이지의 "Directory Server 시작 및 중지"를 참조하십시오.

전역 읽기 전용 모드 설정

디렉토리의 각 접미어는 읽기 전용 모드로 설정할 수 있으며 정의된 특정 참조를 반환할 수 있습니다. 또한 Directory Server는 모든 접미어에 적용되는 전역 읽기 전용 모드를 제공하며 정의된 전역 참조를 반환할 수도 있습니다.

전역 읽기 전용 모드를 사용할 경우 관리자는 접미어를 다시 색인화하는 등의 작업을 수행하는 동안 디렉토리 내용이 수정되는 것을 방지할 수 있습니다. 이 때문에 다음과 같은 구성 분기에는 전역 읽기 전용 모드가 적용되지 않습니다.

- cn=config
- cn=monitor
- cn=schema

이러한 분기는 읽기 전용 설정과 상관없이 관리자 이외의 사용자가 수정할 수 없도록 항상 ACI(Access Control Instruction)를 사용하여 보호해야 합니다(6장, "액세스 제어 관리" 참조). 전역 읽기 전용 모드는 디렉토리 관리자가 시작한 업데이트 작업을 포함하여 디렉토리의 다른 모든 접미어에 대한 업데이트 작업을 방지합니다.

읽기 전용 모드를 사용하면 접미어에 대한 복제도 중단됩니다. 읽기 전용 모드를 사용하기 전의 변경 사항은 계속 복제되지만 더 이상 마스터 복제본에 복제할 변경 사항이 추가되지 않습니다. 읽기 전용 모드를 비활성화할 때까지 소비자 복제본도 업데이트를 받지 못합니다. 다중 마스터 복제 시나리오에서는 마스터에 복제할 변경 사항이 추가되지 않을 뿐만 아니라 다른 마스터의 업데이트를 받을 수도 없습니다.

전역 읽기 전용 모드를 활성화하거나 비활성화하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에 있는 구성 트리에서 루트 노드를 선택한 다음 오른쪽 패널에서 설정 탭을 선택합니다.
2. "서버가 읽기 전용입니다" 확인란을 선택하거나 선택 취소합니다.
3. 저장을 누릅니다. 변경 사항이 즉시 적용됩니다.

개별 접미어를 읽기 전용 모드로 설정하는 방법은 [188페이지의 "접미어 읽기 전용 모드 설정"](#)을 참조하십시오.

디렉토리 항목에 대한 수정 추적

서버에서 새로 작성하거나 수정한 항목에 대해 다음과 같은 특수 속성을 유지 관리하도록 구성할 수 있습니다.

- `creatorsName` — 처음에 항목을 작성한 사람의 고유 이름(DN)
- `createTimestamp` — 항목이 작성된 시간의 타임스탬프(GMT 형식)
- `modifiersName` — 마지막으로 항목을 수정한 사람의 고유 이름(DN)

- `modifyTimestamp` — 항목이 마지막으로 수정된 시간의 타임스탬프(GMT 형식)

주 클라이언트 응용 프로그램에서 연결 접미어에 항목을 작성하거나 수정하는 경우에는 `creatorsName` 속성과 `modifiersName` 속성이 항목의 실제 작성자나 수정자를 반영하지 않습니다. 두 속성에는 원격 서버에 바인드할 때 필요한 연결 프록시 이름이 포함됩니다. 프록시 인증에 대한 자세한 내용은 [155페이지의 "프록시 아이디 작성"](#)을 참조하십시오.

복제된 접미어의 수정 횟수를 추적하는 경우 이름 속성과 타임스탬프 속성이 일반 속성처럼 복제되기 때문에 두 속성은 항목이 소비자에 복제된 시간이 아닌, 마스터 서버의 원래 항목이 수정된 시간을 반영합니다.

Directory Server에서 이 정보를 추적하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에 있는 구성 트리에서 루트 노드를 선택한 다음 오른쪽 패널에서 설정 탭을 선택합니다.
2. "항목 수정 횟수를 추적합니다" 확인란을 선택합니다.
서버는 새로 작성하거나 수정한 모든 항목에 대해 `creatorsName`, `createTimestamp`, `modifiersName` 및 `modifyTimestamp` 속성을 추가합니다. 기존 항목에는 작성 속성이 포함되지 않습니다.
3. 저장을 누른 후 서버를 다시 시작합니다.
자세한 내용은 [35페이지의 "Directory Server 시작 및 중지"](#)를 참조하십시오.

플러그인 서명 확인

플러그인 서명 확인은 Directory Server 5.2의 새 기능입니다. Directory Server와 함께 제공된 각 플러그인에는 시작 시 서버에서 확인할 수 있는 디지털 서명이 있습니다. 기본적으로 서버는 플러그인 서명을 확인하지만, 서명의 존재 또는 유효성 여부에 관계 없이 모든 플러그인을 로드합니다.

서명 확인은 다음과 같은 이점을 제공합니다.

- Directory Server와 함께 제공된 플러그인의 서명은 이 플러그인이 엄격하게 테스트 되었으며 공식적으로 지원된다는 것을 나타냅니다.
- 서명 확인은 플러그인 이진 자체의 체크섬을 사용하여 플러그인의 무단 변경을 감지할 수 있습니다. 따라서 서명은 서버에서 실행되는 중요 코드를 보호합니다.
- 서버에서 서명된 플러그인만 로드하도록 구성하여 서명되지 않은 플러그인 및 지원되지 않는 플러그인의 문제를 감지할 수 있습니다.

플러그인 서명 확인 구성

1. Directory Server 콘솔의 최상위 구성 탭에 있는 구성 트리에서 플러그인 노드를 선택합니다. 오른쪽 패널에 현재 서명 확인 정책이 표시됩니다.
2. 다음 옵션 중 하나를 선택합니다.

- 플러그인 서명을 확인하지 않습니다 - 서명에 관계 없이 서버 구성에 정의된 모든 플러그인이 로드되며 플러그인 서명으로 인한 경고나 오류는 표시되지 않습니다.
- 잘못된 서명으로 플러그인에 플래그를 지정합니다 - 서버 구성에 정의된 모든 플러그인이 로드되지만 서버에서 각 플러그인의 서명을 확인합니다. 플러그인이 이전이 어떤 식으로든 변경된 경우에는 서명이 더 이상 유효하지 않으며, 서버에서 시작 시 및 오류 로그에 경고 메시지를 표시합니다. 서명이 없는 플러그인에도 플래그가 지정됩니다.

서명되지 않은 사용자 정의 플러그인이 있는 경우 이 옵션을 선택하는 것이 좋습니다. 모든 플러그인이 로드되지만 여전히 서명된 플러그인의 상태를 볼 수 있습니다.

- 잘못된 서명이 있는 플러그인을 거부합니다 - 서버에서 구성에 정의된 모든 플러그인의 서명을 확인하지만 유효한 서명이 있는 플러그인만 로드합니다. 서버는 잘못된 서명이 있거나 서명이 없는 플러그인을 나타내는 경고 메시지를 시작 시 및 오류 로그에 표시합니다.

이 옵션은 가장 안전하지만 서명되지 않은 사용자 정의 플러그인을 로드할 수 없는 단점이 있습니다.

3. 저장을 누른 다음 [35페이지](#)의 "**Directory Server 시작 및 중지**"에서 설명한 대로 Directory Server를 다시 시작합니다.

플러그 인의 상태 보기

1. Directory Server 콘솔의 최상위 구성 탭에 있는 구성 트리에서 플러그 인 노드를 확장 다음 확인할 플러그 인을 선택합니다. 오른쪽 패널에 플러그 인의 현재 구성이 표시됩니다.
2. "서명 상태" 필드에는 플러그 인의 서명 확인 상태가 다음 값 중 하나로 표시됩니다.
 - 알 수 없음 - 서버에서 플러그 인 서명을 확인하지 않도록 구성하면 모든 플러그 인에 이 서명 상태가 표시됩니다. 다음과 같은 상태는 플러그 인 서명을 확인하는 경우에만 표시됩니다.
 - 유효한 서명 - 플러그 인 구성에 플러그 인 이진의 체크섬에 일치하는 서명이 있습니다. 이 플러그 인은 공식적으로 지원됩니다. 다음과 같은 상태는 잘못된 서명을 거부하지 않고 플래그를 지정하는 경우에만 표시됩니다.
 - 잘못된 서명 - 플러그 인 구성에 플러그 인 이진의 체크섬에 일치하지 않는 서명이 있습니다. 이 상태는 플러그 인이 무단 변경되었을 수 있음을 나타냅니다.
 - 서명 없음 - 플러그 인 구성에 서버에서 확인할 서명이 없습니다.

DSML 구성

Directory Server는 LDAP(Lightweight Directory Access Protocol) 요청을 처리하는 동시에 Directory Service Markup Language 버전 2(DSMLv2)로 전송된 요청에도 응답합니다. DSML은 클라이언트에서 디렉토리 작업을 인코딩하는 또 다른 방법이지만, 서버는 다른 요청과 마찬가지로 모든 액세스 제어 및 보안 기능을 사용하여 DSML을 처리합니다. 결과적으로 DSML 처리는 많은 유형의 클라이언트가 디렉토리 내용에 액세스할 수 있도록 도와줍니다.

Directory Server는 HTTP(Hypertext Transfer Protocol/1.1)를 통해 DSMLv2의 사용을 지원하며 SOAP(Simple Object Access Protocol) 버전 1.1을 DSML 내용 전송을 위한 프로그래밍 프로토콜로 사용합니다. 이 프로토콜에 대한 자세한 내용과 DSML 요청의 예는 [130페이지의 "DSMLv2를 사용한 디렉토리 액세스"](#)를 참조하십시오.

DSML 요청 사용

디렉토리에 액세스하는 표준 프로토콜은 LDAP이기 때문에 Directory Server를 설치해도 DSML 요청은 기본적으로 사용되지 않습니다. 서버에서 HTTP/SOAP를 통한 DSML 요청에 응답하게 하려면 명시적으로 이 기능을 활성화해야 합니다.

콘솔을 통해 서버에서 DSML 요청을 활성화하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에 있는 구성 트리에서 루트 노드를 선택한 다음 오른쪽 패널에서 네트워크 탭을 선택합니다.
2. DSML 사용 확인란을 선택하고 다음 보안 옵션 중 하나를 선택합니다. 보안 포트 옵션은 11장, "인증 및 암호화 관리"에 설명된 것처럼 SSL을 활성화한 경우에만 사용할 수 있습니다.
 - 비보안 포트만 - 암호화되지 않은 HTTP를 통한 DSML 요청만 비보안 포트에서 승인됩니다.
 - 보안 포트만 - HTTPS를 통한 DSML 요청만 보안 포트에서 승인됩니다.
 - 보안 및 비보안 포트 모두 - 두 포트가 모두 활성화되며 클라이언트에서 둘 중 하나를 선택할 수 있습니다.
3. 다음 필드 중에서 원하는 필드를 편집합니다.
 - 포트 - DSML 요청을 수신하는 HTTP 포트
 - 암호화된 포트 - SSL을 사용하여 암호화된 DSML 요청을 수신하는 HTTPS 포트
 - 상대 URL - 호스트 및 포트에 추가되어 클라이언트에서 DSML 요청을 보낼 때 필요한 전체 URL을 구성하는 상대 URL

기본적으로 서버는 아래 URL로 보내진 요청을 처리합니다.

```
http://host:80/dsml
```

4. 저장을 누르면 서버를 다시 시작해야만 DSML 요청에 응답한다는 알림 메시지가 표시됩니다.

명령줄을 통해 DSML 요청을 활성화하려면 다음을 수행합니다.

1. 아래의 ldapmodify 명령을 사용하여 DSML 프런트엔드 플러그 인을 활성화하고 해당 설정을 수정합니다. 선택 사항으로 ds-hdsml-port, ds-hdsml-secureport 및 ds-hdsml-rooturl 속성을 수정할 수 있습니다.


```
% ldapmodify -h host -p LDAPport -D "cn=Directory Manager" -w passwd
dn:cn=DSMLv2-SOAP-HTTP,cn=frontends,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
-
replace: ds-hdsml-port:
ds-hdsml-port: DSMLport
-
add: ds-hdsml-secureport
ds-hdsml-port: secureDSMLport
-
replace: ds-hdsml-rooturl
ds-hdsml-root: relativeURL
-
^D
```

정의한 매개 변수와 속성 값에 따라 DSML 클라이언트는 아래 URL을 사용하여 이 서버로 요청을 보낼 수 있습니다.

`http://host:DSMLport/relativeURL`

`https://host:secureDSMLport/relativeURL`

2. DSML 프런트엔드 플러그 인을 수정한 경우 이 변경 사항을 적용하려면 서버를 다시 시작해야 합니다. 서버를 다시 시작하기 전에 다음 절에 설명된 것처럼 DSML 인증을 위한 보안 및 아이디 매핑을 구성할 수 있습니다.

DSML 보안 구성

이전 절에서 설명한 보안 포트 설정 외에도 DSML 요청을 승인하는 데 필요한 보안 수준을 구성할 수 있습니다. DSML 프런트엔드 플러그 인의 `ds-hdsml-clientauthmethod` 속성은 클라이언트에 필요한 인증 방법을 지정하며 다음과 같은 값을 가질 수 있습니다.

- `httpBasicOnly` - 서버에서 HTTP Authorization 헤더의 내용을 사용하여 디렉토리 항목에 매핑할 수 있는 사용자 이름을 찾습니다. 이 프로세스와 해당 구성에 대해서는 59페이지의 "DSML 아이디 매핑"에서 자세히 설명합니다. 이렇게 설정하면 클라이언트 인증은 사용되지 않지만 보안 HTTPS 포트에 대한 DSML 요청이 SSL을 통해 암호화됩니다.

- `clientCertOnly` - 서버에서 클라이언트 인증서의 자격 증명을 사용하여 클라이언트를 확인합니다. 이렇게 설정하면 모든 DSML 클라이언트가 보안 HTTPS 포트를 통해 DSML 요청을 보내고 인증서를 제공해야 합니다. 서버는 클라이언트 인증서가 디렉토리 항목과 일치하는지 확인합니다. 자세한 내용은 [11장, "인증 및 암호화 관리"](#)를 참조하십시오.
- `clientCertFirst` - 서버에서 먼저 제공된 클라이언트 인증서를 사용하여 클라이언트 인증을 시도합니다. 클라이언트 인증서가 제공되지 않은 경우 서버는 `Authorization` 헤더의 내용을 사용하여 클라이언트를 인증합니다.

HTTP 요청에 인증서와 `Authorization` 헤더가 모두 없으면 서버는 익명 바인드를 사용하여 DSML 요청을 처리합니다. 익명 바인드는 다음과 같은 경우에도 사용됩니다.

- `clientCertOnly` 값을 지정할 때 클라이언트에서 `Authorization` 헤더만 제공하고 인증서는 제공하지 않는 경우
- `httpBasicOnly` 값을 지정할 때 클라이언트에서 인증서만 제공하고 `Authorization` 헤더는 제공하지 않는 경우

인증서가 제공되었지만 항목에 일치시킬 수 없는 경우 또는 HTTP `Authorization` 헤더가 지정되었지만 사용자 항목에 매핑할 수 없는 경우에는 `ds-hdsml-clientauthmethod` 속성 값에 상관없이 오류 메시지 403: "금지"가 표시되며 요청이 거부됩니다.

콘솔을 통해 DSML 보안 요구 사항을 설정하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에 있는 구성 트리에서 루트 노드를 선택한 다음 오른쪽 패널에서 암호화 탭을 선택합니다.

[11장, "인증 및 암호화 관리"](#)에 설명된 것처럼 SSL이 구성 및 활성화되어 있어야 합니다.

2. DSML 클라이언트 인증 필드의 드롭다운 메뉴에서 원하는 옵션을 선택합니다.
3. 저장을 누른 후 서버를 다시 시작하여 새로운 보안 설정을 실행합니다.

명령줄을 통해 DSML 보안 요구 사항을 설정하려면 다음을 수행합니다.

1. 아래의 `ldapmodify` 명령을 실행하여 DSML 프런트엔드 플러그인의 속성을 편집합니다.

```
% ldapmodify -h host -p LDAPport -D "cn=Directory Manager" -w passwd
dn:cn=DSMLv2-SOAP-HTTP,cn=frontends,cn=plugins,cn=config
changetype: modify
replace: ds-hdsml-clientauthmethod:
ds-hdsml-clientauthmethod: httpBasicOnly 또는
clientCertOnly 또는 clientCertFirst
^D
```

2. DSML 프런트엔드 플러그 인을 수정한 경우 새로운 보안 설정을 실행하려면 서버를 다시 시작해야 합니다.

DSML 아이디 매핑

인증서 없이 기본 인증을 수행하는 경우 Directory Server는 *아이디 매핑*이라는 기법을 사용하여 DSML 요청을 승인할 때 사용할 바인드 DN을 지정합니다. 이 기법은 HTTP 요청의 Authorization 헤더에서 필요한 정보를 추출하여 바인드에 사용할 아이디를 확인합니다. 이 기법에 대한 자세한 내용은 [446페이지의 "아이디 매핑"](#)을 참조하십시오.

DSML-over-HTTP에 대한 기본 아이디 매핑은 다음과 같은 서버 구성 항목을 통해 지정됩니다.

```
dn:cn=default,cn=HTTP-BASIC, cn=identity mapping, cn=config
objectclass: top
objectclass: nsContainer
objectclass: dsIdentityMapping
cn: default
dssearchbasedn: ou=People, userRoot
dssearchfilter: (uid=${Authorization})
```

이 매핑은 *ou=People, userRoot* 하위 트리를 검색하여 uid 속성이 Authorization 헤더에 포함된 사용자 이름과 일치하는 항목을 찾습니다. *userRoot*는 디렉토리 설치 중에 정의한 접미어입니다(예: *dc=example, dc=com*).

매핑 항목 속성에 *\${header}* 형식의 자리 표시자를 사용할 수도 있습니다. 여기서 *header*는 HTTP 헤더의 이름입니다. DSML 매핑에 가장 일반적으로 사용되는 헤더는 다음과 같습니다.

- *\${Authorization}* - 이 문자열은 HTTP Authorization 헤더에 포함된 사용자 이름으로 바뀝니다. Authorization 헤더에는 사용자 이름과 비밀번호가 모두 포함되어 있지만 이 자리 표시자는 사용자 이름으로만 바뀝니다.
- *\${From}* - 이 문자열은 HTTP From 헤더에 포함된 전자 메일 주소로 바뀝니다.
- *\${host}* - 이 문자열은 DSML 요청의 URL에 포함된 서버 자체의 호스트 이름 및 포트 번호로 바뀝니다.

DSML 요청에 다른 아이디 매핑을 사용하려면 다음과 같이 HTTP 헤더에 대한 새 아이디 매핑을 정의합니다.

1. 기본 DSML-over-HTTP 아이디 매핑을 편집하거나 이 프로토콜에 대한 사용자 정의 매핑을 작성합니다. ID 매핑 항목의 속성 정의에 대해서는 [446페이지의 "아이디 매핑"](#)을 참조하십시오. 이러한 매핑 항목은 다음 항목 아래에 위치해야 합니다.
`cn=HTTP-BASIC, cn=identity mapping, cn=config`

다음 두 가지 방법 중 하나로 새 매핑 항목을 작성할 수 있습니다.

- [64페이지의 "콘솔에서 항목 관리"](#)에 설명된 것처럼 Directory Server 콘솔의 최상위 디렉토리 탭에서 적절한 객체 클래스가 있는 새 항목을 작성합니다.
- [81페이지의 "ldapmodify를 사용한 항목 추가"](#)에 설명된 것처럼 ldapmodify 도구를 사용하여 명령줄에서 이 항목을 추가합니다.

2. Directory Server를 다시 시작하여 새 매핑을 적용합니다.

먼저 사용자 정의 매핑이 평가되고, 사용자 정의 매핑이 성공하지 못하면 기본 매핑이 평가됩니다. 모든 매핑을 평가한 후에도 DSML 요청에 대한 바인드 DN을 확인하지 못하면 이 DSML 요청은 금지되어 거부됩니다(오류 403).

디렉토리 항목 관리

이 장에서는 Directory Server 콘솔 및 명령줄 유틸리티를 사용하여 디렉토리 내용을 관리하는 방법에 대해 설명합니다. 또한 속성 암호화 기능을 사용하여 속성이 저장되는 방법 및 DSML을 사용하여 디렉토리에 액세스하는 방법에 대해서도 설명합니다. 디렉토리 배포를 계획할 때 디렉토리에 저장할 데이터 유형을 결정해야 합니다. 항목을 만들고 기본 스키마를 수정하기 전에 먼저 *Directory Server Deployment Planning Guide*의 관련 장을 읽어 보십시오.

이 장에서는 사용자가 LDAP 스키마와 이 스키마에서 정의하는 객체 클래스 및 속성에 대해 어느 정도 알고 있다고 가정합니다. Directory Server에서 제공하는 모든 객체 클래스와 속성 정의 및 스키마에 대한 소개는 *Directory Server Administration Reference*를 참조하십시오. 또한 디렉토리를 수정하려면 적절한 ACI(Access Control Instruction)가 정의되어 있어야 합니다. 자세한 내용은 6장, "액세스 제어 관리"를 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

- 구성 항목
- 콘솔에서 항목 관리
- 명령줄에서 항목 관리
- 항목 이름 바꾸기 및 이동
- 참조 설정
- 속성 값 암호화
- 참조 무결성 유지
- 디렉토리 검색
- 국가별 디렉토리 검색
- DSMLv2를 사용한 디렉토리 액세스

구성 항목

Directory Server는 모든 구성 정보를 다음 파일에 저장합니다.

```
ServerRoot/slapd-serverID/config/dse.ldif
```

이 파일은 LDIF(LDAP Data Interchange Format) 형식입니다. LDIF는 항목, 속성 및 해당 값을 텍스트로 표현하며, RFC2849(<http://www.ietf.org/rfc/rfc2849>)에 설명된 표준 형식입니다. dse.ldif 파일의 Directory Server 구성은 다음과 같은 항목으로 이루어져 있습니다.

- cn=config 항목의 속성과 값
- cn=config 아래의 하위 트리에 있는 모든 항목 및 해당 속성과 값. 항목 또는 속성이 있는지 여부가 중요할 수 있습니다.
- 루트 항목("") 및 cn=monitor 항목의 객체 클래스와 ACI(Access Control Instruction). 이러한 항목의 다른 속성은 서버에서 생성됩니다.

Directory Server에서는 LDAP를 통해 모든 구성 설정을 읽고 쓸 수 있습니다. 기본적으로 디렉토리의 cn=config 분기는 관리 서버에 정의된 디렉토리 관리자(directory administrator) 및 디렉토리 관리자(directory manager)만 액세스할 수 있습니다. 관리 권한이 있는 사용자는 구성 항목을 다른 디렉토리 항목처럼 보고 수정할 수 있습니다.

cn=config 항목 아래에 새 항목을 작성할 경우 일반 항목보다 확장성이 낮은 데이터베이스인 dse.ldif 파일에 항목이 저장되므로 바람직하지 않습니다. 많은 항목, 특히 자주 업데이트되는 항목을 cn=config에 저장하면 성능이 저하됩니다. 하지만 구성 정보를 중앙 집중화하기 위해 복제 관리자(공급자 바인드 DN) 항목과 같은 특수 사용자 항목을 cn=config 아래에 저장하는 것은 도움이 될 수 있습니다.

콘솔에서 구성 수정

Directory Server 콘솔의 최상위 구성 탭을 사용하여 구성을 수정하는 것이 가장 좋습니다. 이 탭의 패널과 대화 상자는 신속하고 효율적으로 구성을 설정할 수 있도록 도와주는 작업 기반의 컨트롤을 제공합니다. 또한 콘솔 인터페이스는 사용자를 대신하여 복잡하고 상호 종속된 구성을 관리합니다.

콘솔의 구성 인터페이스에 대해서는 이 설명서 내의 "콘솔에서..."로 시작되는 절차에서 설명합니다. 이 절차에서는 구성 탭의 패널과 대화 상자를 사용하여 특정 관리 작업을 수행하는 방법을 소개합니다. 변경 사항을 적용하기 위해 서버를 다시 시작해야 하는 경우 및 구성 저장 방법은 인터페이스 자체에서 명확히 지정합니다.

명령줄에서 구성 수정

LDAP를 통해 `cn=config` 하위 트리에 액세스할 수 있으므로 `ldapsearch`, `ldapmodify` 및 `ldapdelete` 명령을 사용하여 서버 구성을 보고 수정할 수 있습니다. `cn=config` 항목 및 모든 하위 항목은 77페이지의 "명령줄에서 항목 관리"에 설명된 절차와 LDIF 형식을 사용하여 수정할 수도 있습니다.

하지만 각 항목의 의미, 속성 용도 및 허용되는 값을 명백히 이해해야 합니다. 이러한 중요 고려 사항에 대해서는 설명서 내의 "명령줄에서..."로 시작되는 절차에서 설명합니다. 이 절차에서는 설정할 수 있는 구성 항목 및 속성의 예를 소개합니다. 허용되는 값의 범위를 비롯한 모든 구성 항목 및 속성에 대한 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.

따라서 콘솔보다 명령줄에서 구성을 수정하는 것이 더 복잡하지만 콘솔에서 사용할 수 없는 구성 설정의 경우에는 명령줄 절차만 제공됩니다. 명령줄 도구를 사용하는 스크립트를 작성함으로써 명령줄 절차를 사용하여 구성 작업을 자동화할 수도 있습니다.

dse.ldif 파일 수정

`dse.ldif` 파일에는 서버를 시작하거나 다시 시작할 때 서버에서 읽고 사용하는 구성이 저장되어 있습니다. 이 파일의 LDIF 내용은 `cn=config` 항목과 해당 하위 트리이며, 사용자가 설치 중에 정의한 시스템에서만 파일을 읽고 쓸 수 있습니다.

파일 내용을 직접 편집하여 구성을 수정할 경우 오류 발생 가능성이 커지므로 바람직하지 않습니다. 다음과 같은 동작에 주의해야 합니다.

- `dse.ldif` 파일은 시작 시 한 번만 읽혀집니다. 이후의 서버 구성은 메모리에 저장된 구성 항목의 LDAP 이미지를 사용하므로 서버를 실행하는 동안의 파일 수정 사항은 지워집니다.
- 콘솔이나 명령줄에서 구성을 수정하면 구성의 LDAP 이미지가 변경됩니다. 일부 디렉토리 기능은 호출 시 현재 구성을 읽기 때문에 서버를 다시 시작할 필요가 없습니다.

- 서버는 구성된 LDAP 이미지가 변경될 때마다 `dse.ldif` 파일을 작성합니다. 서버를 시작할 때만 구성을 읽는 디렉토리 기능도 있으므로 이렇게 파일을 새로 작성하여 변경 사항이 저장되도록 보장합니다.
기존의 `dse.ldif` 파일은 `dse.ldif.bak`로 복사되어 기존의 `dse.ldif.bak` 파일을 덮어씁니다. 따라서 서버를 다시 시작하기 전에 LDAP를 통해 구성을 변경한 경우 `dse.ldif` 파일의 모든 수동 변경 사항은 손실됩니다.
- 디렉토리를 성공적으로 시작한 후 `dse.ldif` 파일은 같은 위치의 `dse.ldif.startOK`로 복사됩니다. 잘못된 구성 변경으로 인해 서버를 시작할 수 없는 경우에는 이 파일을 사용하여 `dse.ldif`를 복원해야 합니다.

콘솔에서 항목 관리

Directory Server 콘솔의 디렉토리 탭과 항목 편집기 대화 상자를 사용하여 개별 항목을 추가, 수정 또는 삭제할 수 있습니다. 동시에 여러 개의 항목에 대해 작업하려면 [77페이지](#)의 "콘솔을 사용한 대량 작업"을 참조하십시오.

Directory Server 콘솔을 시작하고 사용자 인터페이스를 탐색하는 방법은 [37페이지](#)의 "Directory Server 콘솔 사용"을 참조하십시오.

디렉토리 항목 작성

Directory Server 콘솔은 디렉토리 항목을 작성하기 위한 다양한 사용자 정의 템플리트를 제공합니다. 각각의 템플리트는 특정 객체 클래스 유형의 사용자 정의 편집기입니다. [표 2-1](#)에서는 각 사용자 정의 편집기에 사용된 객체 클래스를 보여줍니다.

표 2-1 항목 템플릿 및 해당 객체 클래스

템플릿	객체 클래스
사용자	inetOrgPerson(작성 및 편집용) organizationalPerson(편집용) person(편집용)
그룹	groupOfUniqueNames 및 동적 그룹과 인증서 그룹에 사용할 수 있는 기타 객체 클래스
조직 구성 단위	organizationalUnit
역할	nsRoleDefinition 및 관리된 역할, 필터링된 역할 또는 중첩된 역할에 따른 기타 객체 클래스
서비스 클래스	cosSuperDefinition 및 서비스 클래스 유형에 따른 기타 객체 클래스
비밀번호 정책	passwordPolicy
참조	referral

사용자 정의 편집기에는 해당 객체 클래스의 자주 사용되는 일부 선택 사항 속성과 모든 필수 속성을 나타내는 필드가 포함되어 있습니다. 이런 템플릿 중 하나를 사용하여 항목을 작성하려면 [65페이지의 "사용자 정의 편집기에서 항목 작성"](#)에 설명된 지침에 따라 수행합니다. 다른 유형의 항목을 작성하려면 [67페이지의 "다른 유형의 항목 작성"](#)을 참조하십시오.

사용자 정의 편집기에서 항목 작성

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 디렉토리 트리를 확장하여 새 항목의 부모가 될 항목을 표시합니다.
2. 부모 항목을 마우스 오른쪽 버튼으로 누르고 새로 만들기 메뉴 항목을 선택한 다음 하위 메뉴에서 사용자, 그룹, 조직 구성 단위, 역할, 서비스 클래스, 비밀번호 정책, 참조 등의 항목 유형을 선택합니다. 또는, 부모 항목을 마우스 왼쪽 버튼으로 눌러 선택한 다음 객체 > 새로 만들기 메뉴에서 항목 유형을 선택할 수도 있습니다. 선택한 항목 유형의 사용자 정의 편집기 대화 상자가 표시됩니다.

사용자 정의 편집기의 왼쪽 열에는 탭 목록이 있고 오른쪽에는 각 탭의 필드가 표시됩니다. 기본적으로 모든 사용자 정의 편집기는 새 항목의 이름 및 설명 필드가 있는 최상위 사용자 또는 일반 탭이 선택된 상태로 열립니다.

예를 들어 아래 그림은 사용자 항목의 사용자 정의 편집기를 보여줍니다.

그림 2-1 Directory Server 콘솔 - 사용자 항목의 사용자 정의 편집기

The screenshot shows a web-based form titled "새 사용자 만들기" (Create New User) within the Directory Server console. On the left, there is a sidebar menu with options: "사용자" (User), "계정" (Account), "언어" (Language), "NT 사용자" (NT User), "Posix 사용자" (Posix User), and "계정" (Account). The main form area contains the following fields:

- * 이름: 김동 (Name)
- * 성: 총 (Sex)
- * 일반 이름: 총 김동 (Full Name)
- 사용자 아이디: kdhong (User ID)
- 암호: [Redacted] (Password)
- 암호 확인: [Redacted] (Confirm Password)
- 전자 우편: kdhong@ssg003.red.iplanet.com (예: user@company.com) (Email)
- 전화: (408) 555-3922 (Phone)
- 팩스: (408) 555-4000 (Fax)

At the bottom of the form, there is a note: "*는 필수 필드를 나타냅니다." (Asterisk indicates required fields). Below the form are three buttons: "액세스 권한 도움말" (Access Rights Help), "확인" (OK), "취소" (Cancel), and "도움말" (Help).

3. 사용자 정의 편집기 필드에 제공할 속성 값을 입력합니다. 필드 이름 옆에 별표(*)로 표시된 모든 필수 속성에 값을 입력해야 합니다. 다른 필드는 비워둘 수 있습니다. 여러 값을 허용하는 필드에서는 Enter를 입력하여 값을 구분할 수 있습니다.

각 항목 유형의 사용자 정의 편집기에 있는 특정 필드에 대한 자세한 내용을 보려면 도움말 버튼을 누릅니다. 사용자 및 조직 구성 단위 편집기의 언어 탭에 대한 자세한 내용은 70페이지의 "언어 지원 속성 설정"을 참조하십시오.

그룹, 역할 및 서비스 클래스 항목 작성에 대한 자세한 내용은 5장, "아이디 및 역할 관리"를 참조하십시오. 비밀번호 정책 작성에 대한 자세한 내용은 7장, "사용자 계정 및 비밀번호 관리"를 참조하십시오. 참조 작성에 대한 자세한 내용은 99페이지의 "참조 설정"을 참조하십시오.

4. "확인"을 눌러 새 항목을 작성하고 사용자 정의 편집기 대화 상자를 닫습니다. 디렉토리 트리에서 새 항목이 표시됩니다.
5. 사용자 정의 편집기 대화 상자에서 각 객체 클래스의 선택 사항 속성 필드를 모두 제공하지는 않습니다. 사용자 정의 편집기에 표시되지 않은 선택 사항 속성을 추가하려면 70페이지의 "일반 편집기에서 항목 수정"에 설명된 지침에 따라 수행합니다.

다른 유형의 항목 작성

65페이지의 표 2-1에 열거된 객체 클래스 이외의 객체 클래스 항목을 작성하려면 아래 단계에 따라 수행합니다. 이 절차를 사용하여 디렉토리 스키마에 정의한 사용자 정의 객체 클래스 항목을 작성할 수도 있습니다.

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 디렉토리 트리를 확장하여 새 항목의 부모가 될 항목을 표시합니다.
2. 부모 항목을 마우스 오른쪽 버튼으로 누르고 하위 메뉴에서 새로 만들기 > 기타 항목을 선택합니다. 또는, 부모 항목을 마우스 왼쪽 버튼으로 눌러 선택한 다음 객체 > 새로 만들기 > 기타 메뉴 항목을 선택할 수도 있습니다.

새 객체 대화 상자가 표시됩니다.

3. 새 객체 대화 상자의 객체 클래스 목록에서 새 항목을 정의하는 객체 클래스를 선택한 다음 확인을 누릅니다.

65페이지의 표 2-1에 열거된 객체 클래스를 선택한 경우 해당 사용자 정의 편집기가 표시됩니다(65페이지의 "사용자 정의 편집기에서 항목 작성" 참조). 다른 모든 경우에는 일반 편집기가 표시됩니다.

4. 새 항목을 작성하는 경우 일반 편집기에는 선택한 객체 클래스의 각 필수 속성 필드가 포함되어 있습니다. 모든 필수 속성에 값을 입력해야 합니다. 일부 필드에는 일반 자리 표시자 값(예: New)이 포함되어 있으므로 항목에 적합한 값으로 바뀌어야 합니다.
5. 선택한 객체 클래스에 허용되는 기타 속성을 정의하려면 원하는 속성을 명시적으로 추가해야 합니다. 선택 사항 속성에 값을 제공하려면 다음을 수행합니다.
 - a. 속성 추가 버튼을 눌러 허용되는 속성 목록을 표시합니다.
 - b. 속성 추가 대화 상자에서 하나 이상의 속성을 선택하고 확인을 누릅니다.
 - c. 일반 편집기에서 새 항목 이름 옆에 값을 입력합니다.

이 대화 상자의 그밖에 다른 컨트롤에 대한 자세한 내용은 70페이지의 "일반 편집기에서 항목 수정"을 참조하십시오.

6. 기본적으로 필수 속성 중 하나가 이름 지정 속성으로 선택되어 일반 편집기의 항목 DN에 표시됩니다. 이름 지정 속성을 변경하려면 다음을 수행합니다.
 - a. 변경 버튼을 눌러 이름 지정 속성 변경 대화 상자를 표시합니다.
 - b. 속성 테이블에서 새 항목의 DN에 사용할 하나 이상의 속성 옆에 있는 확인란을 선택합니다.
 - c. 이름 지정 속성 변경 대화 상자에서 확인을 누릅니다. 선택한 이름 지정 속성을 사용한 새 DN이 일반 편집기의 DN으로 표시됩니다.
7. 일반 편집기에서 확인을 눌러 새 항목을 저장합니다.

디렉토리 트리에서 새 항목은 부모 항목의 자식으로 표시됩니다.

사용자 정의 편집기에서 항목 수정

65페이지의 표 2-1에 열거된 객체 클래스의 경우 해당 사용자 정의 편집기나 일반 편집기를 사용하여 항목을 편집할 수 있습니다. 사용자 정의 편집기를 사용하면 가장 일반적인 필드를 쉽게 액세스할 수 있으며 인터페이스를 통해 역할 또는 서비스 클래스 정의 속성과 같은 복잡한 속성 값을 간편하게 정의할 수 있습니다.

일반 편집기에서는 객체 클래스 추가, 허용되는 속성 추가, 여러 값을 갖는 속성 처리 등 항목에 대한 고급 작업을 수행할 수 있습니다. 일반 편집기를 사용하여 항목을 편집하려면 70페이지의 "일반 편집기에서 항목 수정"을 참조하십시오.

주 사용자 정의 편집기는 오직 65페이지의 표 2-1에 열거된 객체 클래스를 편집할 때만 사용할 수 있습니다. 다른 구조적 객체 클래스(예: inetorgperson에서 상속한 사용자 정의 클래스)가 포함된 항목을 편집하려면 일반 편집기를 사용해야 합니다.

열거된 객체 클래스 중 하나와 보조 객체 클래스가 포함된 항목도 사용자 정의 편집기에서 관리할 수 있습니다. 하지만 보조 클래스로 정의된 속성은 사용자 정의 편집기에서 볼 수 없습니다. 보조 객체 클래스의 정의에 대해서는 *Directory Server Administration Reference*를 참조하십시오.

사용자 정의 편집기 호출

65페이지의 표 2-1에 열거된 객체 클래스를 가진 항목을 편집하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 디렉토리 트리를 확장하여 편집할 항목을 표시합니다.
2. 항목을 두 번 누릅니다. 다음과 같은 대체 방법으로 항목의 사용자 정의 편집기를 호출할 수도 있습니다.
 - 항목을 마우스 오른쪽 버튼으로 누르고 "사용자 정의 편집기로 편집" 항목을 선택합니다.
 - 항목을 마우스 왼쪽 버튼으로 눌러 선택한 다음 객체 > 사용자 정의 편집기로 편집 메뉴 항목을 선택합니다.
 - 항목을 마우스 왼쪽 버튼으로 눌러 선택한 다음 Ctrl-P 단축키를 사용합니다.

항목의 객체 클래스에 대한 사용자 정의 편집기가 표시됩니다. 예를 들어, 66페이지의 그림 2-1은 사용자 항목의 사용자 정의 편집기를 보여줍니다.

3. 기본적으로 모든 사용자 정의 편집기는 새 항목의 이름 및 설명 필드가 있는 최상위 사용자 또는 일반 탭이 선택된 상태로 열립니다. 사용자 정의 편집기 필드에서 수정할 속성 값을 편집하거나 제거합니다. 필드 이름 옆에 별표(*)로 표시된 필수 속성 값은 수정할 수는 있지만 제거할 수 없습니다. 다른 필드는 비워둘 수 있습니다. 여러 값을 허용하는 필드에서는 Return을 입력하여 값을 구분할 수 있습니다.

왼쪽 열에서 다른 탭을 선택하여 해당 패널의 값을 수정합니다. 각 항목 유형의 사용자 정의 편집기에 있는 특정 필드에 대한 자세한 내용을 보려면 도움말 버튼을 누릅니다.

사용자 및 조직 구성 단위 편집기의 언어 탭에 대한 자세한 내용은 70페이지의 "언어 지원 속성 설정"을 참조하십시오. 사용자 및 그룹 항목의 계정 탭에 있는 필드에 대해서는 7장, "사용자 계정 및 비밀번호 관리"에서 설명합니다. Directory Server 동기화 서비스를 위해 NT 사용자 및 Posix 사용자 탭이 제공됩니다. 자세한 내용은 Sun 담당자에게 문의하십시오.

그룹, 역할 및 서비스 클래스 항목 수정에 대한 자세한 내용은 5장, "아이디 및 역할 관리"를 참조하십시오. 비밀번호 정책 수정에 대한 자세한 내용은 7장, "사용자 계정 및 비밀번호 관리"를 참조하십시오. 참조 수정에 대한 자세한 내용은 99페이지의 "참조 설정"을 참조하십시오.

4. 확인을 눌러 항목의 변경 사항을 저장하고 사용자 정의 편집기 대화 상자를 닫습니다. 이름 지정 속성(예: 사용자 항목의 일반 이름)을 수정한 경우 해당 변경 사항이 디렉토리 트리에 반영됩니다.

언어 지원 속성 설정

사용자 및 조직 구성 단위 항목의 사용자 정의 편집기는 다국어 디렉토리에 대한 언어 지원을 제공합니다.

1. 69페이지의 "사용자 정의 편집기 호출"에 설명된 것처럼 항목의 사용자 정의 편집기를 엽니다.
2. 왼쪽 열의 언어 탭을 누릅니다.
3. 사용자 항목의 경우 드롭다운 목록에서 기본 설정 언어를 선택할 수 있습니다.
4. 사용자 및 조직 구성 단위 항목의 경우 목록에 표시된 언어의 특정 필드에 현지화된 값을 입력할 수 있습니다. 언어를 선택한 다음 해당 언어에 하나 이상의 값을 입력합니다. 현지화된 값을 정의하면 목록의 언어 이름이 굵게 표시됩니다.
일부 언어에는 현지화된 값의 발음 표시를 입력할 수 있는 발음 필드도 있습니다.
5. 확인을 눌러 항목의 변경 사항을 저장하고 사용자 정의 편집기 대화 상자를 닫습니다.

일반 편집기에서 항목 수정

일반 편집기를 사용하면 콘솔에 로그인할 때 사용한 바인드 DN에 따라 읽기 가능한 모든 항목 속성을 보고, 쓰기 가능한 항목 속성을 편집할 수 있습니다. 또한, 속성을 추가하거나 제거하고 여러 값을 갖는 속성을 설정하며 항목의 객체 클래스를 관리할 수 있습니다. 속성을 추가하는 경우 이진 속성 및 언어 지원을 위한 하위 유형을 정의할 수도 있습니다.

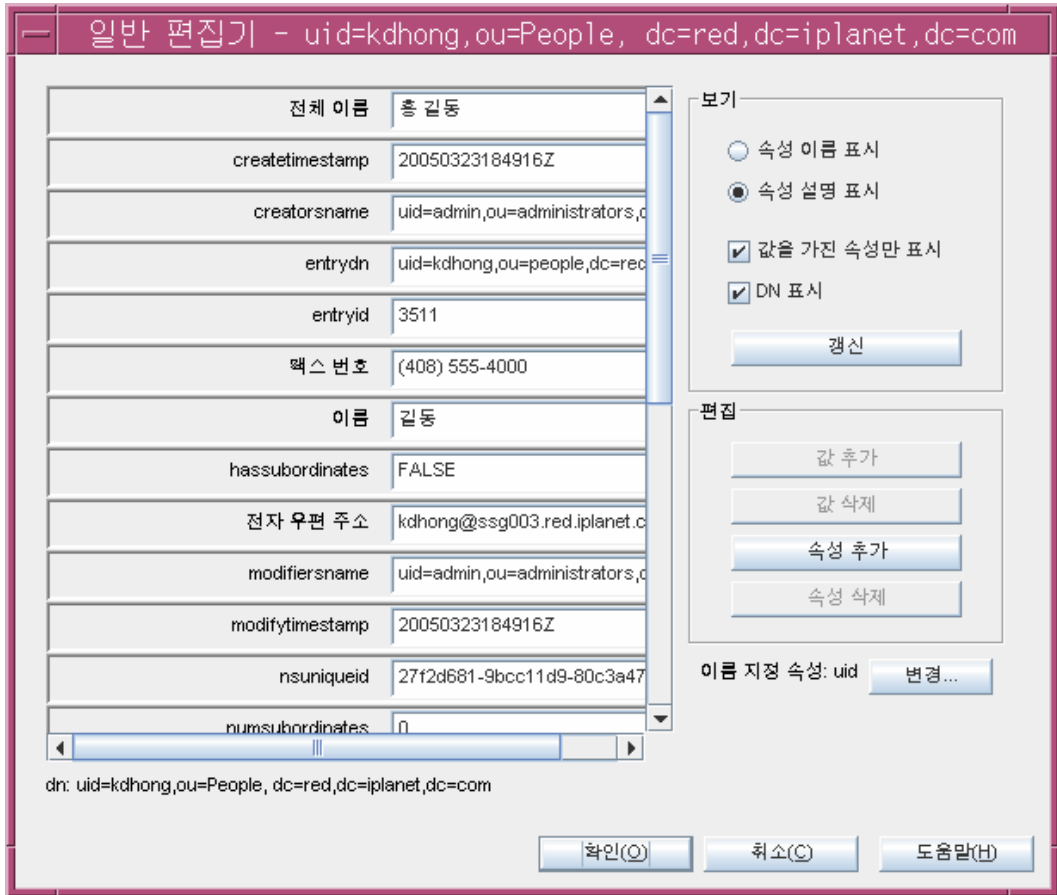
일반 편집기 호출

▶ 디렉토리 항목의 일반 편집기를 호출하려면

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 디렉토리 트리를 확장하여 편집할 항목을 표시합니다.
2. 항목을 마우스 오른쪽 버튼으로 누르고 "일반 편집기로 편집" 항목을 선택합니다. 다음과 같은 대체 방법으로 일반 편집기를 호출할 수도 있습니다.
 - 항목을 마우스 왼쪽 버튼으로 눌러 선택한 다음 객체 > 일반 편집기로 편집 메뉴 항목을 선택합니다.
 - 65페이지의 표 2-1에 열거되지 않은 객체 클래스를 가진 항목을 두 번 누릅니다. 객체 클래스에 사용자 정의 편집기가 없는 경우 기본적으로 일반 편집기가 사용 됩니다.

아래 그림과 같은 일반 편집기가 표시됩니다.

그림 2-2 Directory Server 콘솔 - 일반 편집기



일반 편집기에서 항목 속성은 알파벳순으로 열거되며 각 텍스트 상자에 속성 값이 표시됩니다. 읽기 전용 및 작동 가능 속성을 비롯한 모든 속성이 표시됩니다. 오른쪽의 컨트롤을 사용하여 편집기의 디스플레이를 수정하고 속성 목록을 편집할 수 있습니다.

3. 선택 사항으로, 보기 상자의 컨트롤을 사용하여 일반 편집기의 디스플레이를 수정할 수 있습니다.
 - 처음 스키마에 정의한 대로 속성 이름을 보려면 속성 이름 표시 옵션을 선택합니다. 속성 목록은 이름 알파벳순으로 다시 정렬됩니다.

- 스키마에 정의한 대체 이름별로 속성을 열거하려면 속성 설명 표시 옵션을 선택합니다. 일반적으로 대체 이름은 속성을 보다 명시적으로 설명합니다. 속성 목록은 설명 알파벳순으로 다시 정렬됩니다.
- 스키마에서 항목의 객체 클래스에 대해 명시적으로 허용하는 모든 속성을 열거하려면 "값을 가진 속성만 표시" 확인란을 선택 취소합니다. 항목에 extensibleObject 객체 클래스가 포함되어 있으면 모든 속성이 암묵적으로 허용되지만 표시되지는 않습니다. 기본적으로 값이 정의된 속성만 표시됩니다.
- 속성 목록 아래에 항목의 고유 이름을 표시하거나 표시하지 않으려면 "DN 표시" 확인란을 선택하거나 선택 취소합니다.
- "갱신" 버튼을 누르면 서버에 액세스하여 항목의 현재 내용에 따라 모든 속성 값을 업데이트합니다.

주의 "갱신" 버튼을 누르면 저장하지 않은 일반 편집기의 수정 사항이 즉시 제거됩니다.

다음 절에서는 속성 값 설정, 객체 클래스 관리, 항목의 이름 지정 속성 변경 등의 컨트롤에 대해 설명합니다.

속성 값 수정

▶ 속성 값을 수정하려면

1. 70페이지의 "일반 편집기 호출"에 설명된 것처럼 일반 편집기를 엽니다.
2. 속성 목록을 스크롤하여 수정할 값을 누릅니다.
선택한 속성이 강조 표시되고 선택한 값이 포함된 텍스트 필드에 편집 커서가 나타납니다.
3. 마우스와 키보드를 사용하여 텍스트를 원하는 값으로 편집합니다. 시스템 클립보드를 사용하여 이 필드의 텍스트를 복사하거나 잘라내어 붙여넣을 수도 있습니다.
속성이 읽기 전용이거나 속성을 수정할 수 있는 쓰기 권한이 없는 경우에는 텍스트 필드를 편집할 수 없습니다.
4. 다른 값을 편집하거나 항목을 원하는 대로 수정한 다음 확인을 눌러 변경 사항을 저장하고 일반 편집기를 닫습니다.

여러 값을 갖는 속성 편집

디렉토리 스키마에서 여러 값을 갖는 것으로 정의된 속성의 경우 일반 편집기에 두 개 이상의 값 필드가 표시될 수 있습니다. 자세한 내용은 9장, "디렉토리 스키마 확장"을 참조하십시오.

여러 값을 갖는 속성에 새 값을 추가하려면 다음을 수행합니다.

1. 70페이지의 "일반 편집기 호출"에 설명된 것처럼 일반 편집기를 엽니다.
2. 속성 목록을 스크롤하여 해당 속성이나 속성 값 중 하나를 누릅니다. 선택한 속성이 강조 표시되고 값 추가 버튼이 활성화됩니다. 선택한 속성이 여러 값을 갖는 것으로 정의되지 않았거나 읽기 전용인 경우 또는 속성을 수정할 수 있는 쓰기 권한이 없는 경우에는 버튼이 활성화되지 않습니다.
3. 값 추가 버튼을 누릅니다. 목록에 있는 속성 이름 옆에 새로운 빈 텍스트 필드가 표시됩니다.
4. 새 텍스트 필드에 새로운 속성 값을 입력합니다. 시스템 클립보드를 사용하여 이 필드의 텍스트를 복사하거나 잘라내어 붙여넣을 수도 있습니다.
5. 다른 값을 편집하거나 항목을 원하는 대로 수정한 다음 확인을 눌러 변경 사항을 저장하고 일반 편집기를 닫습니다.

여러 값을 갖는 속성의 값을 제거하려면 다음을 수행합니다.

1. 70페이지의 "일반 편집기 호출"에 설명된 것처럼 일반 편집기를 엽니다.
2. 속성 목록을 스크롤하여 제거할 특정 값을 누릅니다. 선택한 속성이 강조 표시되고 값 삭제 버튼이 활성화됩니다. 선택한 속성이 읽기 전용이거나 속성을 수정할 수 있는 쓰기 권한이 없는 경우에는 버튼이 활성화되지 않습니다.
3. 값 삭제 버튼을 누릅니다. 선택한 값이 포함된 텍스트 필드가 제거됩니다.
4. 다른 값을 편집하거나 항목을 원하는 대로 수정한 다음 확인을 눌러 변경 사항을 저장하고 일반 편집기를 닫습니다.

속성 추가

항목에 속성을 추가하려면 이 속성이 필요하거나 허용되는 객체 클래스가 항목에 포함되어 있어야 합니다. 자세한 내용은 75페이지의 "객체 클래스 관리" 및 9장, "디렉토리 스키마 확장"을 참조하십시오.

항목에 속성을 추가하려면 다음을 수행합니다.

1. 70페이지의 "일반 편집기 호출"에 설명된 것처럼 일반 편집기를 엽니다.
2. 값을 가진 속성만 표시 옵션이 선택되어 있는지 확인합니다.
3. 속성 추가 버튼을 눌러 속성 목록이 있는 대화 상자를 표시합니다. 이 목록에는 항목에 정의된 객체 클래스에서 허용하는 속성만 포함되어 있습니다.
4. 속성 추가 대화 상자에서 추가할 속성을 하나 이상 선택합니다.
5. 선택 사항으로, 대화 상자의 맨 위에 있는 드롭다운 목록에서 다음 하위 유형 중 한 개나 둘 모두를 선택할 수 있습니다.
 - 언어 하위 유형 - 속성 값에 사용된 언어를 표시하려면 이 하위 유형을 사용합니다. 한 속성을 다른 언어로 여러 번 추가하여 디렉토리에 현지화 정보를 저장할 수 있습니다.

선택 사항으로, 언어 이외에 발음 하위 유형을 선택하여 이 속성 값에 해당 언어 값의 발음이 포함되어 있음을 나타낼 수도 있습니다.
 - 이진 하위 유형 - 속성에 이진 하위 유형을 지정하면 실제 구문에 관계없이 속성 값이 이진 데이터(불투명한 데이터 체크)로서 LDAP를 통해 전송됩니다. 이 옵션은 주의해서 사용해야 하며, userCertificate와 같이 LDAP 문자열 표현이 없는 복잡한 구문에 사용됩니다. 속성 값이 이미 이진으로 간주되는 속성에는 이진 하위 유형을 사용하지 마십시오.
6. 속성과 하위 유형(선택 사항)을 선택했으면 확인을 누릅니다. 속성은 일반 편집기의 목록에 알파벳순으로 추가됩니다.
7. 새로운 속성 이름 옆의 빈 텍스트 필드에 새 속성 값을 입력합니다. 시스템 클립보드를 사용하여 이 필드의 텍스트를 복사하거나 잘라내어 붙여넣을 수도 있습니다.
8. 다른 값을 편집하거나 항목을 원하는 대로 수정한 다음 확인을 눌러 변경 사항을 저장하고 일반 편집기를 닫습니다.

속성 제거

항목의 속성과 해당 값을 모두 제거하려면 다음을 수행합니다.

1. 70페이지의 "일반 편집기 호출"에 설명된 것처럼 일반 편집기를 엽니다.

- 속성 목록을 스크롤하여 제거할 속성 이름을 누릅니다. 선택한 속성이 강조 표시되고 속성 삭제 버튼이 활성화됩니다. 선택한 속성이 읽기 전용이거나 속성을 수정할 수 있는 쓰기 권한이 없는 경우에는 버튼이 활성화되지 않습니다.

주 일반 편집기를 사용하면 이 속성에 정의할 수 있는 객체 클래스에 필요한 속성을 제거할 수 있습니다. 필요한 속성 없이 항목을 저장하려고 하면 서버에서 이에 대한 응답으로 객체 클래스 위반을 반환합니다. 항목에서 정의하는 모든 객체 클래스의 필수 속성이 항목에 포함되어 있어야 합니다.

- 속성 삭제 버튼을 누릅니다. 속성과 해당 텍스트 필드 값이 모두 제거됩니다.
- 다른 값을 편집하거나 항목을 원하는 대로 수정한 다음 확인을 눌러 변경 사항을 저장하고 일반 편집기를 닫습니다.

객체 클래스 관리

항목의 객체 클래스는 여러 값을 갖는 `objectclass` 속성으로 정의됩니다. 정의된 객체 클래스의 관리를 돕기 위해 이 속성을 수정하는 경우 일반 편집기는 특수 대화 상자를 제공합니다.

항목에 객체 클래스를 추가하려면 다음을 수행합니다.

- 70페이지의 "[일반 편집기 호출](#)"에 설명된 것처럼 일반 편집기를 엽니다.
- 속성 목록을 스크롤하여 `objectclass` 속성을 선택합니다. 값 추가 버튼이 활성화됩니다. 이 항목의 객체 클래스를 수정할 수 있는 권한이 없는 경우에는 버튼이 활성화되지 않습니다.
- 값 추가 버튼을 누릅니다.
항목에 추가할 수 있는 객체 클래스 목록이 포함된 객체 클래스 추가 대화 상자가 표시됩니다.
- 이 항목에 추가할 객체 클래스를 하나 이상 선택하고 확인을 누릅니다. 선택한 객체 클래스가 `objectclass` 속성 값 목록에 표시됩니다.
- 항목에 없는 속성이 새 객체 클래스에 필요하면 일반 편집기에서 해당 속성을 자동으로 추가합니다. 모든 필수 속성에 값을 입력해야 합니다.
- 다른 값을 편집하거나 항목을 원하는 대로 수정한 다음 확인을 눌러 변경 사항을 저장하고 일반 편집기를 닫습니다.

항목의 객체 클래스를 제거하려면 다음을 수행합니다.

1. 70페이지의 "일반 편집기 호출"에 설명된 것처럼 일반 편집기를 엽니다.
2. 속성 목록을 스크롤하여 제거할 `objectclass` 속성의 특정 값을 누릅니다. 스키마에서 선택한 객체 클래스의 제거를 허용하고 이 항목의 객체 클래스를 수정할 수 있는 권한이 있으면 "값 삭제" 버튼이 활성화됩니다.
3. 값 삭제 버튼을 누릅니다. 특정 객체 클래스가 제거됩니다.
 객체 클래스를 제거하면 일반 편집기에서 자동으로 나머지 객체 클래스가 허용하지 않거나 필요로 하지 않는 속성을 제거합니다. 이름 지정 속성 중 하나를 제거하면 다른 속성이 자동으로 선택되며, 콘솔에서 이 변경 사항을 알려줍니다.
4. 다른 값을 편집하거나 항목을 원하는 대로 수정한 다음 확인을 눌러 변경 사항을 저장하고 일반 편집기를 닫습니다.

디렉토리 항목 삭제

Directory Server 콘솔에서 항목을 삭제하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 디렉토리 트리를 확장하여 제거할 항목을 표시합니다.
 하위 트리의 루트 노드를 선택하여 디렉토리의 전체 분기를 삭제할 수도 있습니다.
2. 항목을 마우스 오른쪽 버튼으로 누르고 삭제 항목을 선택합니다. 다음과 같은 대체 방법으로 항목을 삭제할 수도 있습니다.
 - 항목을 마우스 왼쪽 버튼으로 눌러 선택한 다음 편집 > 삭제 메뉴 항목을 선택합니다. 이 항목을 디렉토리의 다른 위치에 붙여넣으려면 편집 > 잘라내기 메뉴 항목을 사용할 수도 있습니다.
 - 항목을 마우스 왼쪽 버튼으로 눌러 선택한 다음 Ctrl-D 단축키를 사용합니다.
 보기 > 레이아웃 옵션을 선택하여 Directory Server 콘솔의 오른쪽 패널에 자식을 표시한 경우 Ctrl+누름 또는 Shift+누름 조합을 사용하여 삭제할 항목을 여러 개 선택할 수 있습니다.
3. 항목 또는 하위 트리와 해당 내용을 모두 삭제할 것을 확인합니다.
 선택한 항목은 즉시 삭제되며 실행 취소할 수 없습니다. 두 개 이상의 항목을 삭제하면 콘솔에서 삭제한 항목 수와 발생한 오류를 알리는 정보 대화 상자를 표시합니다.

콘솔을 사용한 대량 작업

LDIF 파일을 사용하여 여러 항목을 추가하거나 혼합된 작업을 수행하거나 전체 접미어를 가져올 수 있습니다. LDIF 파일과 Directory Server 콘솔을 사용하여 항목을 추가하려면 다음을 수행합니다.

1. 이전 절에서 설명한 구문을 사용하여 LDIF 파일에 항목 또는 작업을 정의합니다. 단순히 항목을 추가하거나 접미어를 초기화하는 경우 `changetype` 키워드를 사용할 필요 없이 LDIF 파일에 항목만 추가해도 됩니다. 혼합된 작업을 수행하는 경우에는 모든 DN 뒤에 `changetype` 및 필요에 따라 특정 작업이나 속성 값이 있어야 합니다.
2. Directory Server 콘솔에서 LDIF 파일을 가져옵니다. 자세한 내용은 [189페이지](#)의 "LDIF 파일 가져오기"를 참조하십시오.

혼합된 작업을 수행하는 경우 서버에서 모든 LDIF 작업을 수행하도록 "LDIF 가져오기" 대화 상자에서 "추가만" 확인란을 선택 취소합니다.

명령줄에서 항목 관리

`ldapmodify` 및 `ldapdelete` 명령줄 유틸리티는 디렉토리 내용을 추가, 편집 및 삭제하는 모든 기능을 제공합니다. 두 유틸리티를 사용하여 서버의 구성 항목과 사용자 항목의 데이터를 모두 관리할 수 있으며, 두 개 이상의 디렉토리를 대량으로 관리하는 스크립트를 작성할 수도 있습니다.

`ldapmodify` 및 `ldapdelete`는 본 설명서의 절차에서 주로 사용하는 명령입니다. 다음 절에서는 이러한 관리 절차를 수행하는 데 필요한 모든 기본 작업에 대해 설명합니다.

`ldapmodify` 및 `ldapdelete` 명령에 대한 자세한 내용은 *Directory Server Man Page Reference*를 참조하십시오.

명령줄 유틸리티에 대한 입력은 반드시 명령줄 또는 입력 파일을 통해 직접 제공할 수 있는 LDIF 형식이어야 합니다. 아래 절에서는 LDIF 입력에 대해 설명하고 이후 절에서는 각 수정 유형에 대한 LDIF를 소개합니다.

LDIF 입력 제공

모든 디렉토리 데이터는 유니코드의 UTF-8 인코딩을 사용하여 저장되므로 제공하는 모든 LDIF 입력도 UTF-8로 인코딩해야 합니다. LDIF 형식에 대해서는 *Directory Server Administration Reference*의 "LDAP Data Interchange Format Reference"에서 자세히 설명합니다.

LDIF 입력을 제공할 때는 다음을 고려하십시오.

- 객체는 빈 행으로서 뒤에 dn:으로 시작하는 행이 나옵니다. 이 행은 객체의 고유 이름이며 다른 행은 모두 객체의 속성입니다.
- 주석은 #으로 시작(및 EOL로 끝남)합니다.
- 앞에 공백이 하나 있는 행은 이전 행에서 계속되는 행입니다.
- 이전 값은 base-64로 인코딩된 값으로 속성 이름 뒤에 이중 콜론(::)이 함께 표시됩니다.
- 캐리지 리턴과 줄바꿈은 LDIF 값에서 안전하지 않으므로 base-64로 인코딩되어야 합니다.
- ldapmodify 명령을 사용하여 속성 값을 바꿀 때는 해당 속성 값의 끝에 실수로 후행 공백을 두지 않도록 하십시오. 자세한 내용은 [85페이지의 "속성 값 수정"](#)을 참조하십시오.

명령줄에서 LDIF 입력 종료

ldapmodify 유틸리티와 ldapdelete 유틸리티는 파일을 읽는 것과 같은 방법으로 명령 뒤에 입력된 LDIF 명령문을 읽습니다. 입력이 끝나면 셸 프로그램에서 EOF 이스케이프 시퀀스로 인식하는 문자를 입력하십시오.

- 일반적으로 EOF 이스케이프 시퀀스로 Control-D(^D)가 사용됩니다.

아래 예는 ldapmodify 명령에 대한 입력이 어떻게 종료되는지 보여줍니다.

```
prompt> ldapmodify -h host -p port -D bindDN -w password
dn: cn=Barry Nixon,ou=People,dc=example,dc=com
changetype: modify
delete: telephonenumber
^D
prompt>
```

단순성과 이식 가능성을 위해 본 설명서의 예에서는 프롬프트나 EOF 시퀀스를 표시하지 않습니다.

특수 문자 사용

명령줄에 명령 옵션을 입력할 때 명령줄 해석기에 특별한 의미가 있는 문자(예: 공백(), 별표[*], 역슬래시[\] 등)를 이스케이프해야 할 수도 있습니다. 예를 들어, DN에는 공백이 포함되어 있는 경우가 많으므로 대부분의 UNIX 셸에서 DN 값을 다음과 같이 큰따옴표 ("")로 묶어야 합니다.

```
-D "cn=Barbara Jensen,ou=Product Development,dc=example,dc=com"
```

사용하는 명령줄 해석기에 따라 작은따옴표나 큰따옴표를 이스케이프 문자로 사용해야 합니다. 자세한 내용은 운영 체제 설명서를 참조하십시오.

또한, 쉼표가 포함된 DN을 사용하는 경우 역슬래시(\)를 사용하여 쉼표를 이스케이프해야 합니다. 예를 들면 다음과 같습니다.

```
-D "cn=Patricia Fuentes,ou=People,o=example.com Bolivia\S.A."
```

ldapmodify 명령 뒤의 LDIF 명령문은 셸이 아닌 명령에서 해석되므로 특별히 주의할 필요가 없습니다.

속성 OID 사용

기본적으로 속성 OID는 속성 이름에서 지원되지 않습니다. Directory Server의 이전 버전 중 일부에서는 지원되는 경우도 있었습니다. 이 경우는 그에 해당되지 않습니다. Directory Server의 이전 버전에서 속성 이름으로서 속성 OID를 사용한 경우 해당 속성 OID를 허용하려면 nsslapd-attribute-name-exceptions 속성을 on으로 설정해야 합니다.

스키마 검사

항목을 추가하거나 수정하는 경우 이 항목의 객체 클래스에서 필요로 하거나 허용하는 속성을 사용해야 하며 속성 값이 정의된 구문과 일치해야 합니다.

항목을 수정하면 Directory Server는 수정되는 항목만이 아닌 전체 항목에 대해 스키마 검사를 수행합니다. 따라서 항목의 객체 클래스 또는 속성이 스키마에 맞지 않으면 작업이 실패할 수 있습니다. 자세한 내용은 [393페이지의 "스키마 검사"](#)를 참조하십시오.

LDIF 항목 정렬

명령줄 또는 파일의 항목 추가용 LDIF 텍스트 시퀀스에서 부모 항목은 항상 자식보다 앞에 나와야 합니다. 이렇게 하면 서버에서 LDIF 텍스트를 처리할 때 자식 항목보다 먼저 부모 항목을 작성하게 됩니다.

예를 들어, 디렉토리에 없는 항목을 **People** 하위 트리에 작성하려면 다음과 같이 **People** 컨테이너를 나타내는 항목이 이 하위 트리의 항목보다 앞에 나와야 합니다.

```
dn: dc=example,dc=com
dn: ou=People,dc=example,dc=com
...
People subtree entries
...
dn: ou=Group,dc=example,dc=com
...
Group subtree entries
...
```

`ldapmodify` 명령줄 유틸리티를 사용하여 디렉토리에 어떤 항목이든 작성할 수 있지만 접미어 루트나 하위 접미어는 필수 구성 항목과 연결해야 하는 특수 항목입니다. 새로운 루트 접미어나 하위 접미어 및 관련된 구성 항목을 추가하려면 [147페이지의 "명령줄에서 접미어 작성"](#)을 참조하십시오.

대규모 항목 관리

대규모 속성 값을 가진 항목을 추가하거나 수정하려면 먼저 서버에서 이 값을 허용하도록 구성해야 합니다. 서버의 오버로드를 방지하기 위해 클라이언트는 기본적으로 **2MB**보다 큰 데이터를 보낼 수 없도록 제한됩니다.

이보다 큰 항목을 추가하거나 속성 값을 이보다 큰 값으로 수정하면 서버에서 작업을 거부하고 즉시 연결을 닫습니다. 예를 들어, 하나 이상의 항목 속성에 멀티미디어 내용과 같은 이진 데이터가 있으면 이 제한을 초과할 수 있습니다.

또한 대규모 정적 그룹을 정의하는 항목에는 많은 구성원이 포함될 수 있으므로 해당 표시가 이 제한을 초과합니다. 하지만 이러한 그룹은 성능상 바람직하지 않으며 디렉토리 구조를 다시 설계하는 것이 좋습니다. 자세한 내용은 [208페이지의 "그룹 관리"](#)를 참조하십시오.

클라이언트에서 보내는 데이터에 대한 서버의 크기 제한을 수정하려면 다음을 수행합니다.

1. `cn=config` 항목의 `nsslapd-maxbersize` 속성에 새 값을 설정합니다.
 - 콘솔에서 이 작업을 수행하려면 관리자 또는 디렉토리 관리자로 로그인하여 [70페이지의 "일반 편집기에서 항목 수정"](#)에 설명된 지침에 따라 `cn=config` 항목을 편집합니다. `nsslapd-maxbersize` 속성을 클라이언트에서 한 번에 보낼 수 있는 최대 바이트 수로 설정하십시오.
 - 명령줄에서 이 작업을 수행하려면 아래 명령을 실행합니다.


```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=config
changetype: modify
replace: nsslapd-maxbersize
nsslapd-maxbersize: sizeLimitInBytes
^D
```

자세한 내용은 *Directory Server Administration Reference*의 "nsslapd-maxbersize"를 참조하십시오.

2. 35페이지의 "Directory Server 시작 및 중지"에 설명된 것처럼 서버를 다시 시작합니다.

오류 처리

명령줄 도구는 LDIF 입력의 모든 항목 또는 수정 사항을 순차적으로 처리합니다. 기본 동작은 처음 오류가 발생할 때 처리를 중지하는 것입니다. 오류에 관계 없이 모든 입력을 계속 처리하려면 `-c` 옵션을 사용하십시오. 도구 출력에 오류 조건이 표시됩니다.

위에 열거된 주의 사항 외에 일반적으로 발생하는 오류는 다음과 같습니다.

- 작업에 대한 적절한 액세스 권한이 없는 경우
- 디렉토리에 있는 DN을 가진 항목을 추가하는 경우
- 존재하지 않는 부모 아래에 항목을 추가하는 경우

`ldapmodify` 및 `ldapdelete` 명령에서 발생하는 오류 조건과 이를 피하는 방법에 대한 자세한 내용은 *Directory Server Man Page Reference*를 참조하십시오.

ldapmodify를 사용한 항목 추가

`ldapmodify`의 `-a` 옵션을 사용하여 하나 이상의 항목을 디렉토리에 추가할 수 있습니다. 아래 예에서는 사용자가 포함될 구조적 항목을 작성한 후에 사용자 항목을 작성합니다.

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: ou=People,dc=example,dc=com
objectclass: top
objectclass: organizationalUnit
ou: People
description: Container for user entries

dn: uid=bjensen,ou=People,dc=example,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetorgPerson
```

```
uid: bjensen
givenName: Barbara
sn: Jensen
cn: Babs Jensen
telephoneNumber: (408) 555-3922
facsimileTelephoneNumber: (408) 555-4000
mail: bjensen@example.com
userPassword: clearPassword
```

-D 옵션과 -w 옵션은 각각 이 항목을 작성할 수 있는 권한이 있는 사용자의 바인드 DN과 비밀번호를 제공합니다. -a 옵션은 LDIF의 모든 항목이 추가된다는 것을 나타냅니다. 그런 후에 각 항목이 해당 DN과 속성 값으로 지정되고 항목 사이에는 빈 줄이 들어갑니다. ldapmodify 유틸리티는 입력된 항목을 작성하고 오류가 발생하면 이를 보고합니다.

관례상, 항목의 LDIF는 다음과 같은 순서로 속성을 열거합니다.

- 객체 클래스 목록
- 이름 지정 속성. 이 속성은 DN에 사용되며, 반드시 필수 속성일 필요는 없습니다.
- 모든 객체 클래스의 필수 속성 목록
- 허용되는 속성 중에서 추가할 모든 속성

userpassword 속성 값을 입력할 때는 일반 텍스트로 비밀번호를 지정하십시오. 서버에서 이 값을 암호화하여 암호화된 값만 저장합니다. LDIF 파일에 표시되는 일반 텍스트 비밀번호를 보호하려면 읽기 권한을 제한해야 합니다.

명령줄에서 -a 옵션을 지정할 필요가 없는 다른 형식의 LDIF를 사용할 수도 있습니다. 이 형식은 다음 절에서 설명하는 항목 수정 명령문과 항목 추가 명령문을 결합할 수 있다는 이점이 있습니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: ou=People,dc=example,dc=com
changetype: add
objectclass: top
objectclass: organizationalUnit
ou: People
description: Container for user entries

dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: add
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetorgPerson
uid: bjensen
givenName: Barbara
```

```

sn: Jensen
cn: Barbara Jensen
telephoneNumber: (408) 555-3922
facsimileTelephoneNumber: (408) 555-4000
mail: bjensen@example.com
userPassword: clearPassword

```

`changetype: add` 키워드는 지정된 DN의 항목이 이후의 모든 속성을 사용하여 작성되어야 함을 나타냅니다. 다른 모든 옵션과 LDIF 규정은 동일합니다.

두 예에서 모두 `-f filename` 옵션을 사용하여 단말기 입력이 아닌 파일에서 LDIF를 읽을 수도 있습니다. `-a` 옵션의 사용에 따라 LDIF 파일에는 단말기 입력과 동일한 형식이 포함되어야 합니다.

Idapmodify를 사용한 항목 수정

속성 및 해당 값을 기존 항목에 추가하거나 대체 또는 제거하려면 `changetype: modify` 키워드를 사용합니다. `changetype: modify`를 지정하는 경우 항목의 수정 방법을 나타내는 하나 이상의 변경 작업도 함께 제공해야 합니다. 아래 예에서는 가능한 LDIF 변경 작업 중 세 개를 보여줍니다.

```

dn: entryDN
changetype: modify
add: attribute
attribute: value
...
-
replace: attribute
attribute: newValue
...
-
delete: attribute
[attribute: value]
...

```

같은 항목에 대한 작업을 구분하려면 하이픈(-)을 사용하고 다른 항목에 대한 작업 그룹을 구분하려면 빈 줄을 사용합니다. 각 작업에 여러 개의 `attribute: value` 쌍을 지정하여 동시에 추가, 교체 또는 삭제할 수도 있습니다.

속성 값 추가

아래 예에서는 어떻게 동일한 add LDIF 구문을 사용하여 여러 값을 갖는 기존 속성 및 존재하지 않는 속성에 값을 추가할 수 있는지 보여줍니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: cn
cn: Babs Jensen
-
add: mobile
mobile: (408) 555-7844
mobile: (408) 555-7845
```

다음과 같은 조건에 부합될 경우 이 작업은 실패할 수 있으며 서버에서 오류를 반환합니다.

- 지정된 값이 해당 속성에 이미 있는 경우
- 값이 속성에 정의된 구문과 일치하지 않는 경우
- 항목의 객체 클래스에서 해당 속성 유형을 필요로 하지 않거나 허용하지 않는 경우
- 속성 유형이 여러 값을 갖지 않으며 이미 값이 있는 경우

이진 속성 하위 유형 사용

`attribute;binary` 하위 유형을 지정하면 실제 구문에 관계없이 속성 값이 이진 데이터로서 LDAP를 통해 전송됩니다. 이진 하위 유형은 `userCertificate`와 같이 LDAP 문자열 표현이 없는 복잡한 구문에 사용되며, 이외의 용도로 사용해서는 안 됩니다.

`ldapmodify` 명령과 함께 사용된 모든 LDIF 명령문의 속성 이름에 적절한 하위 유형을 추가할 수 있습니다.

이진 값을 입력하려면 LDIF 텍스트에 직접 입력하거나 다른 파일에서 읽어올 수 있습니다. 아래 예에서는 파일의 값을 읽어오는 LDIF 구문을 보여줍니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
version: 1
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: userCertificate;binary
userCertificate;binary:< file:///path/certFile
```

< 구문을 사용하여 파일 이름을 지정하려면 LDIF 명령문을 `version: 1`로 시작해야 합니다. `ldapmodify`는 이 명령문을 처리할 때 속성을 지정된 파일의 전체 내용에서 읽은 값으로 설정합니다.

언어 하위 유형이 지정된 속성 추가

속성의 언어 및 발음 하위 유형은 현지화된 값을 지정합니다. 속성에 언어 하위 유형을 지정하면 다음과 같이 속성 이름에 하위 유형이 추가됩니다.

```
attribute;lang-CC
```

여기서 `attribute`는 기존 속성 유형이고 `CC`는 언어를 지정하는 두 글자 국가 코드입니다. 선택 사항으로 언어 하위 유형에 발음 하위 유형을 추가하여 현지화된 값의 발음을 지정할 수도 있습니다. 이 경우 속성 이름은 아래와 같이 표시됩니다.

```
attribute;lang-CC;phonetic
```

하위 유형이 지정된 속성에 작업을 수행하려면 해당 하위 유형을 명시적으로 일치시켜야 합니다. 예를 들어, `lang-fr` 언어 하위 유형이 지정된 속성 값을 수정하려면 다음과 같이 수정 작업에 `lang-fr`를 추가해야 합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
replace: homePostalAddress;lang-fr
homePostalAddress;lang-fr: 34\, avenue des Champs-Elysées
```

속성 값 수정

다음 예에서는 LDIF의 `replace` 구문을 사용하여 속성 값을 바꾸는 방법을 보여 줍니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
replace: sn
sn: Morris
-
replace: cn
cn: Barbara Morris
cn: Babs Morris
```

지정된 속성의 현재 값이 모두 제거되고 제공된 모든 값이 추가됩니다.

속성 값을 바꾼 후에는 `ldapsearch` 명령을 사용하여 변경 사항을 확인할 수 있습니다.

후행 공백

속성 값을 수정할 때는 값의 끝에 실수로 후행 공백을 포함하지 않도록 하십시오. 후행 공백으로 인해 값이 base-64 인코딩(예: 34xy57eg)으로 표시될 수도 있습니다.

속성 값의 끝에 후행 공백이 있으면 이 공백은 해당 속성 값의 일부로 인코딩됩니다. 콘솔이나 `ldapsearch` 명령을 사용하여 변경 사항을 확인할 때는 보이는 값이 일반 텍스트일 수도 있으나 base-64로 인코딩된 텍스트로 표시될 수도 있습니다. 이는 사용하는 Directory Server 클라이언트에 따라 달라집니다.

속성 값 삭제

아래 예에서는 속성을 완전히 삭제하는 방법과 여러 값을 갖는 속성의 값 하나만 삭제하는 방법을 보여줍니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
delete: facsimileTelephoneNumber
-
delete: cn
cn: Babs Morris
```

`attribute: value` 쌍을 지정하지 않고 `delete` 구문을 사용하면 이 속성의 모든 값이 제거됩니다. `attribute: value` 쌍을 지정하면 해당 값만 제거됩니다.

여러 값을 갖는 속성의 값 하나만 수정

`ldapmodify` 명령을 사용하여 여러 값을 갖는 속성의 값 하나만 수정하려면 아래 예와 같이 두 가지 작업을 수행해야 합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
delete: mobile
mobile: (408) 555-7845
-
add: mobile
mobile: (408) 555-5487
```

ldapdelete를 사용한 항목 삭제

디렉토리에서 항목을 삭제하려면 ldapdelete 명령줄 유틸리티를 사용합니다. 이 유틸리티는 디렉토리 서버에 바인드하여 해당 DN으로 지정된 하나 이상의 항목을 삭제합니다. 지정된 항목을 삭제할 수 있는 권한이 있는 바인드 DN을 제공해야 합니다.

자식이 있는 항목은 삭제할 수 없습니다. LDAP 프로토콜은 자식 항목의 부모가 없는 상황을 허용하지 않습니다. 예를 들어, 조직 구성 단위 항목을 삭제하려면 먼저 조직 구성 단위에 속해 있는 모든 항목을 삭제해야 합니다.

주의 o=NetscapeRoot 접미어를 삭제하지 마십시오. Administration Server는 설치된 Sun Java System 서버에 대한 정보를 저장할 때 이 접미어를 사용합니다. 이 접미어를 삭제하면 Directory Server를 비롯한 모든 Sun Java System 서버를 다시 설치해야 할 수 있습니다.

아래 예에서는 조직 구성 단위의 항목이 한 개뿐이므로 이 항목을 삭제한 후에 부모 항목을 삭제할 수 있습니다.

```
ldapdelete -h host -p port -D "cn=Directory Manager" -w password
uid=bjensen,ou=People,dc=example,dc=com
ou=People,dc=example,dc=com
```

ldapmodify를 사용한 항목 삭제

changetype: delete 키워드를 ldapmodify 유틸리티와 함께 사용하여 항목을 삭제할 수도 있습니다. 이 경우 ldapdelete를 사용할 때와 같은 제한이 모두 적용됩니다. 항목을 삭제하는 LDIF 구문은 한 개의 LDIF 파일로 혼합된 작업을 수행할 수 있다는 이점이 있습니다.

아래 예에서는 이전 예와 동일한 삭제 작업을 수행합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: delete

dn: ou=People,dc=example,dc=com
changetype: delete
```

항목 이름 바꾸기 및 이동

이 절에서는 DN 수정 작업을 소개하고 DN 수정 작업 사용에 관한 지침을 제공하며 콘솔 및 명령줄을 사용하여 DN 수정 작업을 수행하는 방법에 대해 설명합니다.

DN 수정 작업 소개

Directory Server 5.2 2005Q1 이전의 Directory Server 버전에서는 항목의 이름을 바꾸는 것이 가능했습니다. 반면, Directory Server 5.2 2005Q1 및 향후 버전에서는 항목의 이름을 바꾸는 것과 이동하는 것이 가능합니다.

다음 작업에 대해서는 DN 수정 작업을 사용할 수 없습니다.

- 한 접미어에서 다른 접미어로 항목을 이동
- 루트 접미어의 이름을 바꾸거나 이동

이름 바꾸기 및 이동 작업 사이의 차이

이 절에서는 항목의 이름을 바꾸는 것과 항목을 이동하는 것의 차이에 대해 설명합니다.

항목 이름 바꾸기

항목의 이름을 바꾸면 해당 항목의 DN에서 가장 왼쪽(가장 덜 중요한) 속성 *type=value* 쌍이 바뀝니다. 이 속성 *type=value* 쌍은 항목의 RDN입니다. 속성 유형 또는 속성 값, 또는 둘 다의 이름이 바뀔 수 있습니다. 이름 바꾸기 작업이 성공하려면 새 DN이 이미 존재하지 않아야 합니다.

다음 예는 속성 유형과 속성 값이 어떻게 바뀔 수 있는지 보여줍니다.

예 1: 다음 DN에서 속성 유형을 cn에서 uid로 바꾸려면

```
dn: cn=john,dc=california,dc=sun,dc=com
```

위의 내용이 다음과 같이 됩니다.

```
dn: uid=john,dc=california,dc=sun,dc=com
```

uid는 항목의 유효한 속성이어야 합니다.

예 2: 다음 DN에서 속성 값을 john에서 bob으로 바꾸려면

```
dn: cn=john,dc=california,dc=sun,dc=com
```

위의 내용이 다음과 같이 됩니다.

```
dn: cn=bob,dc=california,dc=sun,dc=com
```

항목 이동

항목을 이동하면 해당 항목의 DN에서 맨 오른쪽(가장 중요한) 속성 *type=value* 쌍이 바뀝니다. 이 작업에 따라 항목이 다른 하위 트리로 옮겨집니다. 이동 작업이 성공하려면 새 위치에 해당하는 DN이 같은 접미어에 있어야 합니다.

예 3: 다음 DN에서 항목 john을 california에서 france로 이동하려면

```
dn: cn=john,dc=california,dc=sun,dc=com
```

위의 내용은 다음과 같이 됩니다.

```
dn: cn=john,dc=france,dc=sun,dc=com
```

DN 수정 작업 사용에 대한 지침과 제한 사항

DN 수정 작업을 사용하도록 계획하는 경우 다음 절에 제공되어 있는 지침을 따르십시오.

- "DN 수정 작업 사용에 대한 일반 지침"
- "복제와 함께 DN 수정 작업 사용에 대한 지침"

DN 수정 작업 사용에 대한 일반 지침

DN 수정 작업을 사용할 때는 다음 권장 사항을 준수하십시오.

- 다음 작업에 대해서는 DN 수정 작업을 사용하지 마십시오.
 - 한 접미어에서 다른 접미어로 항목 이동
 - 루트 접미어의 이름 바꾸기 또는 이동

- 현재 Directory Server 5.2 2005Q1 이상의 버전을 실행하고 있는지 확인합니다. DN 수정 작업은 Directory Server 5.2 2005Q1 이전의 Directory Server 버전에서는 사용할 수 없습니다. 복제된 토폴로지를 사용하는 경우에는 해당 토폴로지의 모든 서버가 Directory Server 5.2 2005Q1 이상을 실행하고 있는지 확인합니다.
- 응용 프로그램에서 entryid 작동 가능 속성을 사용하지 마십시오. 이 속성은 내부 사용을 위해서만 예약되어 있습니다. 항목의 entryid 속성은 항목을 이동할 때 변경할 수 있습니다.
- DN 수정 작업을 서버의 모든 접미어에 대해 전역적으로 활성화하거나 작업을 실행할 접미어마다 개별적으로 활성화합니다. 기본적으로 DN 수정 작업은 비활성화 상태입니다. 복제된 토폴로지를 실행하는 경우에는 해당 토폴로지의 모든 서버에서 DN 수정 작업을 활성화합니다. DN 수정 작업의 활성화 방법에 대한 자세한 내용은 92페이지의 "콘솔에서 DN 수정 작업 활성화" 또는 97페이지의 "ldapmodify 명령을 사용하여 DN 수정 작업 활성화"를 참조하십시오.
- DN 수정 작업을 실행하려는 접미어마다 ACI 권한을 확장합니다. Import 액세스 권한을 사용하면 항목을 지정된 DN으로 가져올 수 있습니다. Export 액세스 권한을 사용하면 항목을 지정된 DN에서 내보낼 수 있습니다. ACI 권한의 확장 방법에 대한 자세한 내용은 92페이지의 "콘솔에서 DN 수정 작업 활성화" 또는 97페이지의 "ldapmodify 명령을 사용하여 DN 수정 작업 활성화"를 참조하십시오.
- DN 수정 작업을 수행하기 전에 먼저 해당 작업으로 인해 클라이언트 인증이 손상되지 않는지 확인합니다. 클라이언트 인증서를 참조하는 항목을 이동하면 클라이언트 인증이 손상됩니다. 항목을 이동한 다음에는 인증서를 검증하십시오.
- DN 수정 작업을 수행하기 전에 먼저 해당 작업으로 인해 응용 프로그램이 손상되지 않는지 확인합니다. 항목의 이름을 바꾸거나 이동하는 작업은 일부 접미어에 영향을 줄 수도 있고 항목의 다음 특성을 바꿀 수도 있습니다.
 - 항목의 필터링된 역할 범위
 - 항목의 중첩된 역할(중첩된 역할에 필터링된 역할이 포함됨)
 - 항목의 동적 그룹 구성원

복제와 함께 DN 수정 작업 사용에 대한 지침

복제와 함께 DN 수정 작업을 사용하는 경우에는 복제 토폴로지가 다음의 요구 사항을 준수해야 합니다.

주의 다음 요구 사항을 준수하지 않고 DN 수정 작업을 사용하면 복제에 손상을 일으켜 디렉토리 서비스가 종료될 수 있습니다.

- 복제 토폴로지의 모든 서버에서 Directory Server 5.2 2005Q1 이상을 실행하고 있는지 확인합니다. Directory Server 5.2 2005Q1 이전의 Directory Server 버전에서는 DN 수정 작업을 사용할 수 없습니다.
- 복제 토폴로지의 모든 서버에서 DN 수정 작업을 활성화합니다. DN 수정 작업이 마스터 서버에서는 지원되지만 소비자 서버에서는 지원되지 않는 경우에는 제대로 복제되지 않습니다. 다음과 같은 메시지가 공급자 서버의 오류 로그에 작성됩니다.

Unable to start a replication session with MODDN enabled

복제를 다시 시작하려면 다음 단계를 수행합니다.

1. 복제 토폴로지를 재구성하여 모든 서버에서 DN 작업을 활성화합니다.
2. 다음 중 한 가지 방법으로 복제 세션을 시작합니다.
 - [375페이지의 "콘솔에서 복제 업데이트 강제"](#) 또는 [376페이지의 "명령줄에서 복제 업데이트 강제"](#)의 지침을 따르는 방법
 - 공급자 서버에서 항목을 바꾸는 방법. 이 변경 사항은 소비자 서버에 복제됩니다.
- 토폴로지의 모든 마스터 복제에서 참조 무결성 플러그 인을 활성화하고 구성합니다. 이 작업으로 서버는 그룹과 역할에 대한 참조 무결성을 유지할 수 있습니다. 참조 무결성 플러그 인을 활성화 및 구성하는 방법에 대한 자세한 내용은 [109페이지의 "참조 무결성 구성"](#)을 참조하십시오.
- DN 수정 작업을 수행한 후 참조 무결성 플러그 인이 이러한 변경 사항을 복제하는 데는 약간의 시간이 걸립니다.

콘솔에서 항목 이름 바꾸기 또는 이동

이 절에서는 콘솔에서 항목의 이름을 바꾸고 항목을 이동하는 방법에 대해 설명합니다.

콘솔에서 DN 수정 작업 활성화

ACI 권한이 사용자에게 사용 권한을 주도록 구성되어 있지 않은 경우에는 접미어에 대해 DN 수정 작업을 수행할 수 없습니다. DN 수정 작업은 서버의 모든 접미어에 대해 전역적으로 활성화 또는 비활성화될 수도 있고 지정된 접미어에 대해 개별적으로 활성화 또는 비활성화될 수도 있습니다.

다음 절차는 서로 다른 ACI 권한을 구성하는 방법에 대한 예입니다. 이러한 절차가 사용자에게 가장 적합한 ACI 권한을 구성하는 것은 아닙니다. 다른 ACI 권한을 구성하는 방법에 대한 자세한 내용은 *Administration Server Administration Guide*에서 액세스 제어 지침 작업에 대한 정보를 참조하십시오.

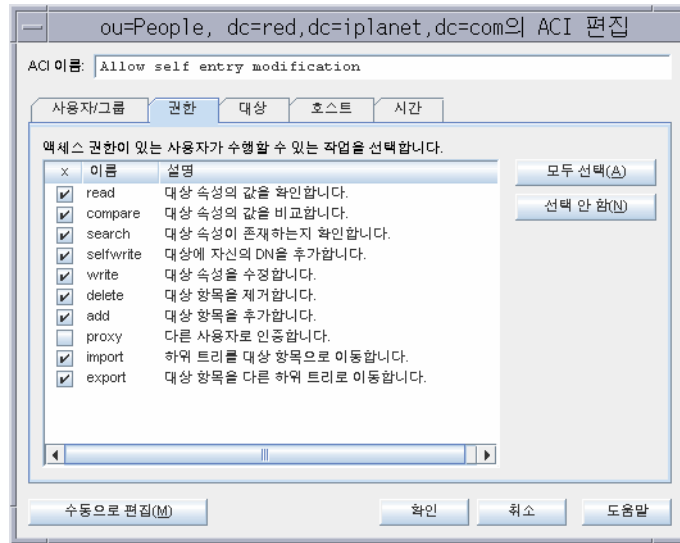
▶ 콘솔에서 ACI 권한을 확장하려면

이 절차는 모든 사용자가 전체 접미어에 대해 DN 수정 작업을 수행할 수 있도록 ACI 권한을 구성합니다.

1. 디렉토리 탭에서 Directory Server 콘솔을 엽니다.
2. 왼쪽에서 ACI 사용 권한을 확장하려는 접미어를 선택합니다.
3. 해당 접미어를 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 "액세스 권한 설정"을 선택합니다.

액세스 제어 관리자 창이 표시됩니다. 이 창에는 해당 항목에 속하는 ACI 목록이 포함되어 있습니다.

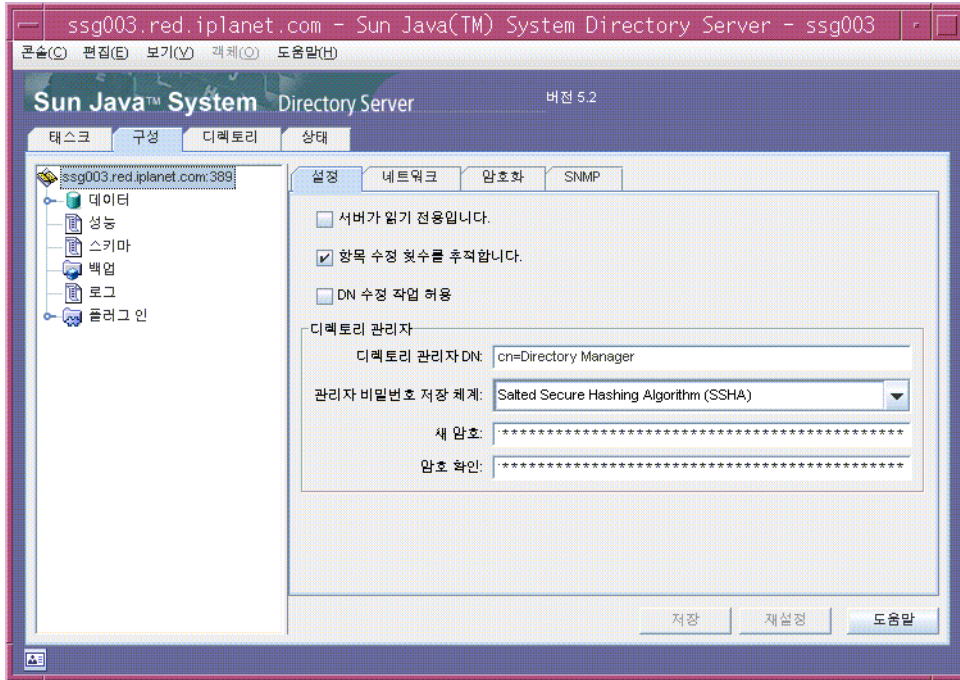
4. 액세스 제어 관리자 창에서 익명 액세스 ACI를 강조 표시하고 편집을 누릅니다. 액세스 제어 편집기가 표시됩니다.



5. import라는 레이블이 있는 확인란을 선택하여 지정된 접미어로 자식 항목을 가져올 수 있도록 합니다.
 6. export라는 레이블이 있는 확인란을 선택하여 지정된 접미어 아래에서 다른 위치로 항목을 이동할 수 있도록 합니다.
 7. ACI 편집 창에서 확인을 누릅니다. 창이 닫힙니다.
 8. "액세스 제어 관리자" 창에서 확인을 누릅니다. 창이 닫힙니다.
- ▶ 콘솔에서 DN 수정 작업을 전역적으로 활성화 또는 비활성화하려면
1. 구성 탭 및 설정 탭에서 Directory Server 콘솔을 엽니다.

2. DN 수정 작업을 전역적으로 허용하려면 "DN 수정 작업 허용"이라는 레이블이 있는 확인란을 선택합니다.

DN 수정 작업을 전역적으로 허용하지 않으려면 "DN 수정 작업 허용"이라는 레이블이 있는 확인란을 선택 취소합니다.

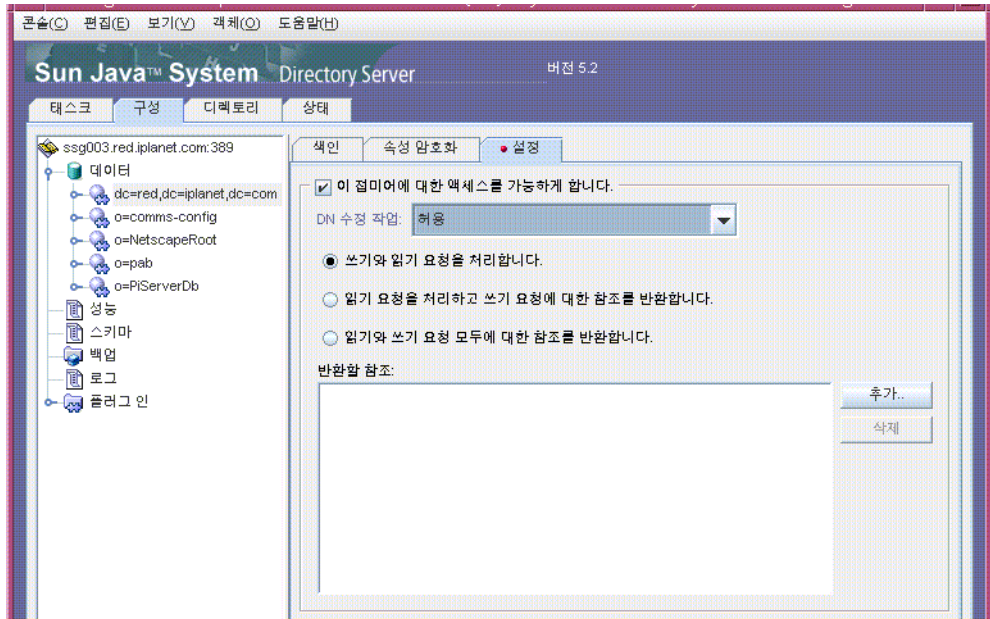


3. 저장을 누릅니다.
4. 확인 팝업 창에서 예를 누르면 선택 사항이 확정됩니다.

▶ 콘솔에서 지정된 접미어에 대해 DN 수정 작업을 활성화하려면

1. 구성 탭 및 설정 탭에서 Directory Server 콘솔을 엽니다.
2. 왼쪽에서 DN 수정 작업을 허용 또는 허용하지 않으려는 접미어를 선택합니다.
3. 설정 탭을 선택합니다.

4. "DN 수정 작업"이라는 레이블이 있는 상자의 드롭다운 메뉴에서 앞서 지정한 접미어에 대해 작업을 선택하여 DN 수정 작업을 허용 또는 허용하지 않습니다.



5. 저장 버튼을 누릅니다.
 DN 수정 작업이 서버에 대해 전역적으로 활성화되어 있으면 확인하는 팝업 창이 나타납니다.
6. 이 창에서 예를 누르면 선택 사항이 확정됩니다.
7. 필요에 따라 각 접미어에 대해 단계 4부터 단계 6까지 반복합니다.

콘솔에서 항목 이름 바꾸기

이 절에서는 항목의 이름을 바꾸는 방법에 대해 설명합니다. 이 작업은 Directory Server 5.2 2005Q1 이전의 Directory Server 버전에서 지원됩니다. 이 작업을 수행하기 위해 DN 수정 작업을 활성화할 필요는 없습니다.

▶ 콘솔에서 항목의 이름을 바꾸려면

1. 디렉토리 탭에서 Directory Server 콘솔을 엽니다.
2. 디렉토리 트리를 확장하여 편집하려는 항목을 표시합니다.

- 이름을 바꾸려는 항목을 선택하고 마우스 오른쪽 단추로 누른 다음 "일반 편집기로 편집"을 선택합니다.
일반 편집기 창이 열립니다. 일반 편집기 창은 [71페이지의 그림 2-2](#)에서 볼 수 있습니다.
- 바꾸려는 RDN에 해당하는 속성을 선택합니다.
- 항목의 현재 이름에서 새 이름으로 속성 상자의 텍스트를 바꿉니다.
- 확인을 누릅니다.

콘솔에서 항목 이동

이 절차는 Directory Server 5.2 2005Q1 이전의 Directory Server 버전에서는 지원되지 않습니다.

▶ 콘솔에서 항목을 이동하려면

- DN 수정 작업이 이동하려는 항목을 포함하고 있는 집미사에 대해 전역적으로 활성화 되어 있는지 확인합니다. 자세한 내용은 [92페이지의 "콘솔에서 DN 수정 작업 활성화"](#)를 참조하십시오.
- 디렉토리 탭에서 Directory Server 콘솔을 엽니다.
- 디렉토리 트리를 확장하여 편집하려는 항목을 표시합니다.
- 이동하려는 항목을 선택합니다.
- 항목을 새 부모로 끌어 놓습니다.
- 경고 팝업 상자에서 계속을 누릅니다.

콘솔에서 항목 이동 및 이름 바꾸기

항목의 이름을 바꾸고 이동하려면 다음 두 가지를 모두 수행해야 합니다.

- [95페이지의 "콘솔에서 항목의 이름을 바꾸려면"](#)
- [96페이지의 "콘솔에서 항목을 이동하려면"](#)

ldapmodify 명령을 사용하여 항목 이름 바꾸기 또는 이동

이 절에서는 ldapmodify 명령을 사용하여 항목의 이름을 바꾸고 항목을 이동하는 방법을 설명합니다.

다음 속성은 LDIF 명령문과 함께 사용됩니다.

- `dn` - 이름을 바꾸거나 이동하려는 항목을 지정합니다.
- `changetype: modrdn` - DN 수정 작업이 사용되도록 지정합니다.
- `newrdn` - 새로운 이름 지정 속성을 제공합니다.
- `deleteoldrdn` - 이전의 이름 지정 속성을 항목에서 제거할지 여부를 나타냅니다(1은 예, 0은 아니오).
이름 지정 속성이 항목 정의에서 의무적으로 사용되어야 할 속성일 경우 항목에서 이 속성을 제거할 수 없습니다.
- `newsuperior` - 항목의 새로운 상위 속성을 지정합니다.

LDIF 명령문과 함께 사용되는 속성에 대한 자세한 내용은 *Directory Server Administration Reference*의 "Attribute Reference"를 참조하십시오. `ldapmodify` 명령과 그 옵션에 대한 자세한 내용은 *Directory Server Man Page Reference*를 참조하십시오.

ldapmodify 명령을 사용하여 DN 수정 작업 활성화

DN 수정 작업을 사용할 수 있으려면 먼저 ACI 권한을 확장하고 DN 수정 작업을 활성화해야 합니다.

ACI 권한의 확장 방법에 대한 자세한 내용은 270페이지의 "명령줄에서 ACI 작성"을 참조하십시오.

이 절에서는 DN 수정 작업을 활성화하는 방법을 설명합니다.

▶ ldapmodify 명령을 사용하여 DN 수정 작업을 전역적으로 활성화하려면

`ldapmodify` 명령을 실행합니다. 예를 들어 다음 예를 사용하면 DN 수정 작업을 활성화할 수 있습니다.

```
ldapmodify -h <hostname> -p <port> -D <user> -w <user_password>
dn: cn=config,cn=ldb database,cn=plugins,cn=config
changetype: modify
replace: nsslapd-moddn-enabled
nsslapd-moddn-enabled: on
```

▶ ldapmodify 명령을 사용하여 접미어에 대해 DN 수정 작업을 활성화하려면

`ldapmodify` 명령을 실행합니다. 다음 예를 사용하면 `suffix-name`이라는 이름의 접미어에 대해 DN 수정 작업을 활성화할 수 있습니다.

```
ldapmodify -h <hostname> -p <port> -D <user> -w <user_password>
dn: cn=<suffix-name>,cn=ldb database,cn=plugins,cn=config
changetype: modify
replace: nsslapd-moddn-enabled
nsslapd-moddn-enabled: on
```

ldapmodify 명령을 사용하여 항목 이름 바꾸기

이 절에서는 항목의 이름을 바꾸는 방법을 설명합니다. 이 작업은 Directory Server 5.2 2005Q1 이전의 Directory Server 버전에서 지원됩니다. 이 작업을 수행하기 위해 DN 수정 작업을 활성화할 필요는 없습니다.

▶ ldapmodify 명령을 사용하여 리프 항목의 이름을 바꾸려면

ldapmodify 명령을 실행합니다. 다음 예에서는 항목의 이름을 john에서 bob으로 바꿉니다.

```
ldapmodify -h <hostname> -p <port> -D <user> -w <user_password>
dn: cn=john,dc=california,dc=sun,dc=com
changetype: modrdn
newrdn: cn=bob
deleteoldrdn: 1
```

ldapmodify 명령을 사용하여 항목 이동

이 절에서는 접미어의 다른 부분으로 항목을 이동하는 방법을 설명합니다. 이 작업은 Directory Server 5.2 2005Q1 이전의 Directory Server 버전에서는 지원되지 않습니다.

▶ ldapmodify 명령을 사용하여 항목을 이동하려면

1. DN 수정 작업에 대해 ACI 권한이 확장되어 있는지 확인합니다. 자세한 내용은 97페이지의 "ldapmodify 명령을 사용하여 DN 수정 작업 활성화"를 참조하십시오.
2. 이름 바꾸기 및 이동 작업에 의해 영향을 받는 접미어에 대해 DN 수정 작업이 활성화되어 있는지 확인합니다. 자세한 내용은 97페이지의 "ldapmodify 명령을 사용하여 DN 수정 작업 활성화"를 참조하십시오.
3. ldapmodify 명령을 실행합니다. 예를 들어 다음 명령은 john 항목을 California 하위 트리에서 France 하위 트리로 이동합니다.

```
ldapmodify -h <hostname> -p <port> -D <user> -w <user_password>
dn: cn=john,dc=california,dc=sun,dc=com
changetype: modrdn
newrdn: cn=john
deleteoldrdn: 0
newsuperior: dc=france,dc=france,dc=sun,dc=com
```

ldapmodify 명령을 사용하여 항목 이름 바꾸기 및 이동

이 절에서는 항목의 이름을 바꾸고 접미어의 다른 부분으로 이동하는 방법을 설명합니다. 이 작업은 Directory Server 5.2 2005Q1보다 앞선 Directory Server 버전에서는 지원되지 않습니다.

▶ ldapmodify 명령을 사용하여 항목의 이름을 바꾸고 이동하려면

1. DN 수정 작업에 대해 ACI 권한이 확장되어 있는지 확인합니다. 자세한 내용은 [97페이지의 "ldapmodify 명령을 사용하여 DN 수정 작업 활성화"](#)를 참조하십시오.
2. 이름 바꾸기 및 이동 작업에 의해 영향을 받는 접미사에 대해 DN 수정 작업이 활성화되어 있는지 확인합니다. 자세한 내용은 [97페이지의 "ldapmodify 명령을 사용하여 DN 수정 작업 활성화"](#)를 참조하십시오.
3. ldapmodify 명령을 실행합니다. 예를 들어 다음 명령은 이름 바꾸기 작업과 이동 작업을 하나의 작업으로 결합합니다.

```
ldapmodify -h <hostname> -p <port> -D <user> -w <user_password>
dn: cn=john,dc=california,dc=sun,dc=com
changetype: modrdn
newrdn: dc=bob
deleteoldrdn: 1
newsuperior: dc=france,dc=france,dc=sun,dc=com
```

참조 설정

참조를 사용하여 클라이언트 응용 프로그램에서 로컬에 없는 정보를 구하기 위해 연결할 서버를 알려줄 수 있습니다. 참조란 Directory Server에서 작업 결과 대신 클라이언트로 반환하는 원격 접미어 또는 항목에 대한 포인터입니다. 이 경우 클라이언트는 참조에 지정된 원격 서버에 대해 다시 작업을 수행해야 합니다. 리디렉션은 다음 세 가지 경우에 발생합니다.

- 클라이언트 응용 프로그램에서 로컬 서버에 없는 항목을 요청하면 서버는 기본 참조를 반환합니다.
- 유지 관리나 보안상의 이유로 전체 접미어가 오프라인 상태인 경우에는 서버에서 해당 접미어에 정의된 참조를 반환합니다. 접미어 수준의 참조에 대해서는 [151페이지의 "액세스 권한 및 참조 설정"](#)에서 설명합니다. 또한 클라이언트에서 쓰기 작업을 요청하면 접미어의 읽기 전용 복제본은 마스터 서버에 대한 참조를 반환합니다.

- 스마트 참조 항목을 작성할 수 있습니다. 클라이언트에서 명시적으로 스마트 참조에 액세스하면 서버는 정의된 참조를 대신 반환합니다. Directory Server 콘솔에서 자동으로 스마트 참조를 따르도록 구성하여 스마트 참조가 최상위 디렉토리 탭에 로컬 항목처럼 표시되게 할 수 있습니다.

참조는 항상 다른 서버의 호스트 이름, 포트 번호 및 DN(선택 사항)이 포함된 LDAP URL 형식으로 지정됩니다. 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오. 디렉토리 배포 시 참조를 사용하는 방법에 대한 개념 정보는 *Directory Server Deployment Planning Guide*를 참조하십시오.

다음 절에서는 디렉토리의 기본 참조 및 스마트 참조를 정의하는 절차에 대해 설명합니다.

기본 참조 설정

클라이언트 응용 프로그램이 사용자 디렉토리의 접미어에 존재하지 않는 DN에 대한 작업을 제출하면 기본 참조가 해당 클라이언트 응용 프로그램으로 반환됩니다. 기본 참조는 디렉토리의 모든 접미어에 적용되기 때문에 전역 참조라고도 불립니다. 서버는 정의된 모든 참조를 반환하지만 반환 순서는 정의되어 있지 않습니다.

콘솔에서 기본 참조 설정

1. Directory Server 콘솔의 최상위 구성 탭에 있는 구성 트리의 루트에서 서버 노드를 선택한 다음 오른쪽 패널에서 네트워크 탭을 선택합니다.
2. 참조 반환 확인란을 선택하고 텍스트 필드에 LDAP URL을 입력합니다. 또는 URL 생성을 눌러 LDAP URL의 정의를 도와줄 대화 상자를 표시합니다. 보안 포트에 대한 LDAP URL의 예는 아래와 같습니다.

```
ldaps://east.example.com:636/dc=example,dc=com
```

다음과 같이 여러 개의 참조 URL을 공백과 인용 부호로 구분하여 입력할 수도 있습니다.

```
"ldap://east.example.com:389" "ldap://backup.example.com:389"
```

3. 저장을 누르면 변경 사항이 즉시 적용됩니다.

명령줄에서 기본 참조 설정

ldapmodify 명령줄 유틸리티를 사용하여 디렉토리 구성 파일의 cn=config 항목에 하나 이상의 기본 참조를 추가 또는 교체합니다. 예를 들면 다음과 같습니다.

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=config
changetype: modify
replace: nsslapd-referral
nsslapd-referral: ldap://east.example.com:389
nsslapd-referral: ldap://backup.example.com:389
```

서버를 다시 시작할 필요는 없습니다.

스마트 참조 작성

스마트 참조를 사용하면 디렉토리 항목이나 디렉토리 트리를 특정 LDAP URL에 매핑할 수 있습니다. 스마트 참조를 통해 클라이언트 응용 프로그램에서 특정 서버나 특정 서버의 특정 항목을 참조하도록 설정할 수 있습니다.

스마트 참조는 동일한 DN을 가진 다른 서버의 실제 항목을 가리키는 경우가 많습니다. 하지만 동일한 서버나 다른 서버의 모든 항목에 대해 스마트 참조를 정의할 수 있습니다. 예를 들어 다음과 같은 DN을 가진 항목을 정의할 수 있습니다.

```
uid=bjensen,ou=People,dc=example,dc=com
```

이 항목을 east.example.com 서버의 다른 항목을 가리키는 스마트 참조로 정의하려면 다음과 같이 입력합니다.

```
cn=Babs Jensen,ou=Sales,o=east,dc=example,dc=com
```

디렉토리는 RFC 2251 섹션 4.1.11에 지정된 표준에 따라 스마트 참조를 사용합니다 (<http://www.ietf.org/rfc/rfc2251.txt> 참조).

콘솔에서 스마트 참조 작성

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 디렉토리 트리를 확장하여 스마트 참조의 부모가 될 항목을 표시합니다.
2. 부모 항목을 마우스 오른쪽 버튼으로 누르고 새로 만들기 > 참조 메뉴 항목을 선택합니다. 또는, 부모 항목을 마우스 왼쪽 버튼으로 눌러 선택한 다음 객체 > 새로 만들기 > 참조 메뉴 항목을 선택할 수도 있습니다.

참조 항목의 사용자 정의 편집기 대화 상자가 표시됩니다.

3. 편집기의 일반 탭에서 참조 이름을 입력하고 드롭다운 목록에서 이름 지정 속성을 선택합니다. 이 이름은 선택한 이름 지정 속성 값이 됩니다. 선택 사항으로, 이 참조에 대한 설명 문자열을 입력할 수도 있습니다.

4. 편집기의 URL 탭에서 구성 버튼을 눌러 스마트 참조의 URL을 정의합니다. 표시되는 대화 상자에 LDAP URL의 요소를 입력합니다.

URL은 참조 항목이 저장된 디렉토리 서버의 호스트 이름과 LDAP 포트 번호, 서버에 있는 대상 항목의 DN 등으로 구성됩니다. 기본적으로 대상 DN은 스마트 참조 항목의 DN과 같으며 접미어, 하위 트리 또는 리프 항목 중 하나일 수 있습니다.

5. LDAP URL 구성 대화 상자에서 확인을 누릅니다. 새 참조 텍스트 상자에 이 URL이 표시됩니다.
6. 새 참조 텍스트 상자 옆에 있는 추가를 눌러 목록에 참조를 추가합니다.
7. 이 항목에 대한 참조로 반환할 URL을 두 개 이상 정의할 수 있습니다. 구성, 추가, 삭제 및 변경 버튼을 사용하여 참조 목록을 작성하고 관리합니다.
8. 참조 인증 버튼을 눌러 Directory Server 콘솔에서 원격 서버에 대한 참조를 따르면서 바인드할 때 사용할 자격 증명을 설정할 수 있는 대화 상자를 표시합니다. 서버에 액세스할 때 사용할 바인드 DN과 비밀번호를 정의할 수도 있습니다. 같은 서버에 대한 참조는 모두 동일한 자격 증명을 사용합니다.
9. 추가, 편집 및 삭제 버튼을 사용하여 서버 목록 및 해당 자격 증명을 관리합니다. 모두 마쳤으면 확인을 누릅니다.
10. 참조의 사용자 정의 편집기에서 확인을 눌러 스마트 참조 항목을 저장합니다.

콘솔의 디렉토리 트리에서 스마트 참조 항목이 있던 위치에 대상 하위 트리나 항목이 표시되어야 합니다. 노란색 경고 아이콘, URL 또는 자격 증명이 표시된 스마트 참조 항목은 잘못된 것입니다. 항목을 두 번 눌러 참조 오류가 표시되면 계속을 누른 다음 URL 인증 또는 참조 인증을 수정하여 오류를 수정합니다.

명령줄에서 스마트 참조 작성

스마트 참조를 작성하려면 `referral` 및 `extensibleObject` 객체 클래스가 포함된 항목을 작성합니다. 참조 객체 클래스는 LDAP URL이 포함되는 `ref` 속성을 허용합니다. `extensibleObject` 객체 클래스를 사용하면 이름 지정 속성과 같은 스키마 속성을 이용하여 대상 항목과 일치시킬 수 있습니다.

예를 들어, `uid=bjensen` 항목 대신 스마트 참조를 반환하려면 아래 항목을 정의합니다.

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: referral
uid: bjensen
ref: ldap://east.example.com/cn=Babs%20Jensen,ou=Sales,
o=east,dc=example,dc=com
```

주 LDAP URL에서 공백 뒤의 정보는 서버에서 무시되므로 참조로 사용할 LDAP URL에는 공백 대신 %20을 사용해야 합니다. 다른 특수 문자는 이스케이프해야 합니다.

스마트 참조를 정의한 후에 uid=bjensen 항목을 수정하면 이 변경 사항이 실제로 다른 서버의 cn=Babs Jensen 항목에 적용됩니다. ldapmodify 명령은 다음과 같이 자동으로 참조를 따릅니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: replace
replace: telephoneNumber
telephoneNumber: (408) 555-1234
```

스마트 참조 항목을 수정하려면 다음과 같이 ldapmodify의 -M 옵션을 사용해야 합니다.

```
ldapmodify -M -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: replace
replace: ref
ref: ldap://east.example.com/cn=Babs%20Jensen,ou=Marketing,
o=east,dc=example,dc=com
```

속성 값 암호화

속성 암호화는 디렉토리에 저장된 중요한 데이터를 보호합니다. 속성 암호화를 사용하여 항목의 특정 속성을 암호화 형식으로 저장하도록 지정하면 데이터베이스 파일, 백업 파일 및 내보낸 LDIF 파일에 저장된 데이터를 읽을 수 없습니다.

이 기능을 사용할 경우 속성 값은 Directory Server 데이터베이스에 저장되기 전에 암호화되고 원래 값으로 암호가 해독된 후 클라이언트로 반환됩니다. 액세스 제어를 사용하여 클라이언트가 권한 없이 이러한 속성에 액세스하지 못하도록 제한하고 SSL을 사용하여 클라이언트와 Directory Server 간에 전송되는 속성 값을 암호화해야 합니다. 일반적인 데이터 보안의 구조적 개요와 특별한 속성 암호화에 대해서는 *Directory Server Deployment Planning Guide*를 참조하십시오.

서버에 SSL이 구성 및 활성화되어 있는 경우에만 속성 암호화를 사용할 수 있으며 기본적으로 속성은 암호화되지 않습니다. 속성 암호화는 접미어 수준에서 구성되므로 속성은 해당 속성이 포함된 각 접미어 항목에서 암호화됩니다. 전체 디렉토리에서 특정 속성을 암호화하려면 모든 접미어에서 해당 속성을 암호화해야 합니다.

주의 속성 암호화는 접미어와 관련된 모든 데이터 및 색인 파일에 영향을 줍니다. 기존 접미어의 암호화 구성을 수정하는 경우 반드시 구성 내용을 먼저 내보내서 원하는 대로 수정한 후에 다시 가져와야 합니다. 콘솔에서 단계별 작업에 대한 도움을 받을 수 있습니다.

또한, 암호화를 사용하는 경우 암호화되지 않은 값이 있는 데이터베이스 캐시 파일을 수동으로 삭제해야 합니다.

가능하면 새로운 접미어에 데이터를 로드하거나 작성하기 전에 암호화된 모든 속성을 활성화하는 것이 좋습니다.

일부 항목에서 이름 지정 속성으로 사용하는 속성을 암호화하도록 선택하면 DN에 표시된 값이 아닌 항목에 저장된 값이 암호화됩니다.

userPassword 속성을 암호화하도록 선택할 수도 있지만, DIGEST-MD5 SASL 인증과 같이 비밀번호를 단순 텍스트로 저장해야 하는 경우가 아니면 이 설정은 실제 보안에 전혀 도움이 되지 않습니다. 비밀번호 정책에 정의된 암호화 기법이 이미 비밀번호에 적용되어 있으면 추가 암호화를 적용해도 보안은 강화되지 않고 모든 바인드 작업의 성능만 저하됩니다.

저장 시 암호화 속성 앞에 사용된 암호화 알고리즘을 나타내는 암호화 태그가 표시됩니다. DES 암호화 알고리즘으로 암호화된 속성은 다음과 같이 나타납니다.

```
{CKM_DES_CBC}3hac&jla+=snda%
```


콘솔에서 속성 암호화 구성

1. Directory Server 콘솔의 구성 탭을 선택하고 데이터 노드를 확장한 다음, 암호화할 속성 값이 있는 접미어를 선택합니다. 오른쪽 패널에서 속성 암호화 탭을 선택합니다.

이 탭에는 현재 접미어에 대해 암호화된 모든 속성의 이름과 암호화 체계가 열거된 테이블이 포함되어 있습니다.

2. 속성에 대한 암호화를 활성화하려면 다음을 수행합니다.
 - a. 속성 추가 버튼을 눌러 속성 목록을 표시합니다.
 - b. 목록에서 암호화할 속성을 선택하고 확인을 누릅니다. 테이블의 속성 이름 옆에 해당 속성이 추가됩니다.
 - c. 속성 이름 옆의 드롭다운 목록에서 이 속성의 암호화 체계를 선택합니다.
3. 속성에 대한 암호화를 해제하려면 테이블에서 속성 이름을 선택하고 속성 삭제 버튼을 누릅니다.
4. 저장을 누릅니다. 구성을 변경하기 전에 접미어 내용을 LDIF 파일로 내보내라는 메시지가 표시됩니다.
5. 접미어 내보내기를 눌러 내보내기 대화 상자를 열거나, 속성 암호화 구성을 내보내지 않고 수정하려면 계속을 누릅니다. 그런 후에 새 구성이 저장됩니다.

아직 접미어를 내보내지 않은 경우에는 이때 접미어를 내보내야만 해당 내용을 저장할 수 있습니다. 접미어에 암호화된 속성이 포함되어 있고 다음 단계에서 이 LDIF 파일을 사용하여 접미어를 다시 초기화할 계획이면 내보낸 LDIF에서 접미어 암호화를 그대로 유지할 수 있습니다.

LDIF 파일에서 접미어를 초기화하라는 메시지가 표시됩니다.

6. 지금 접미어 초기화를 눌러 초기화 대화 상자를 열고 디렉토리에 로드할 LDIF 파일 이름을 입력합니다.

이전 단계에서 암호화된 속성이 포함된 접미어를 내보낸 경우, 암호화된 값은 접미어를 다시 초기화한 후에 복구할 수 없으므로 이 파일을 사용하여 지금 초기화해야 합니다. 파일을 로드하고 색인이 작성되는 동안 지정된 속성 값이 모두 암호화됩니다.

이때 접미어를 초기화하지 않으려면 단기를 누릅니다. 188페이지의 "데이터 가져오기"에 설명된 것처럼 나중에 데이터를 가져올 수 있습니다.

7. 하나 이상의 속성을 암호화하도록 구성을 변경한 경우 가져오기 작업 전에 지정된 암호화되지 않은 일부 속성 값은 계속 데이터베이스 캐시에 표시될 수 있습니다. 데이터베이스 캐시를 지우려면 다음을 수행합니다.

- a. 35페이지의 "Directory Server 시작 및 중지"에 설명된 것처럼 Directory Server를 중지합니다.
- b. 관리자 권한이 있는 루트로 아래 명령을 실행하여 파일 시스템의 데이터베이스 캐시 파일을 삭제합니다.

```
ServerRoot/slapd-serverID/db/_db.*
```

- c. Directory Server를 다시 시작합니다. 서버에서 자동으로 새 데이터베이스 캐시 파일을 작성합니다.

명령줄에서 속성 암호화 구성

1. 속성 암호화를 구성할 접미어에 항목이 있으면 먼저 이 접미어의 내용을 LDIF 파일로 내보내야 합니다. 자세한 내용은 194페이지의 "데이터 내보내기"를 참조하십시오.

접미어에 암호화된 속성이 포함되어 있고 단계 5에서 이 LDIF 파일을 사용하여 접미어를 다시 초기화할 계획이면 내보낸 LDIF에서 접미어 암호화를 그대로 유지할 수 있습니다.

2. 속성에 대한 암호화를 활성화하려면 `ldapmodify` 명령을 사용하여 아래 구성 항목을 추가합니다.

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=attributeName, cn=encrypted attributes, cn=databaseName,
    cn=ldbm database, cn=plugins, cn=config
objectclass: top
objectclass: dsAttributeEncryption
cn: attributeName
dsEncryptionAlgorithm: cipherName
```

여기서 *attributeName*은 암호화할 속성의 유형 이름이고, *databaseName*은 접미어에 해당하는 데이터베이스의 상징적인 이름이며, *cipherName*은 다음 중 하나입니다.

- o `ckm_des_cbc` - DES 블록 암호
- o `ckm_des3_cbc` - Triple-DES 블록 암호
- o `ckm_rc2_cbc` - RC2 블록 암호
- o `ckm_rc4` - RC4 스트림 암호

3. 속성에 대한 암호화를 해제하려면 `ldapmodify` 명령을 사용하여 아래 구성 항목을 수정합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=attributeName, cn=encrypted attributes, cn=databaseName,
    cn=ldbm database, cn=plugins, cn=config
changetype: modify
replace: dsEncryptionAlgorithm
dsEncryptionAlgorithm: clearText
```

여기서 *attributeName*은 암호화할 속성의 유형 이름이고, *databaseName*은 접미어에 해당하는 데이터베이스의 상징적인 이름입니다.

주 속성 암호화 구성 항목은 삭제하지 마십시오. 다음에 접미어를 초기화하면 자동으로 삭제됩니다.

4. 하나 이상의 속성을 암호화하도록 구성을 변경한 경우 가져오기 작업 전에 지정된 암호화되지 않은 일부 속성 값은 계속 데이터베이스 캐시에 표시될 수 있습니다. 데이터베이스 캐시를 지우려면 다음을 수행합니다.

- a. 35페이지의 "**Directory Server 시작 및 중지**"에 설명된 것처럼 Directory Server를 중지합니다.
- b. 관리자 권한이 있는 루트로 아래 명령을 실행하여 파일 시스템의 데이터베이스 캐시 파일을 삭제합니다.

```
ServerRoot/slapd-serverID/db/__db.*
```

- c. Directory Server를 다시 시작합니다. 서버에서 자동으로 새 데이터베이스 캐시 파일을 작성합니다. 캐시가 다시 채워질 때까지 해당 접미어의 작업 성능이 다소 느려질 수도 있습니다.
5. 188페이지의 "**데이터 가져오기**"에 설명된 것처럼 LDIF 파일을 사용하여 접미어를 초기화합니다.

파일을 로드하고 해당 색인이 작성되는 동안 지정된 속성 값이 모두 암호화됩니다.

참조 무결성 유지

*참조 무결성*이란 관련된 항목 간의 관계를 유지하는 플러그인 기법입니다. 그룹 구성원의 속성과 같이 다른 항목의 DN이 포함된 속성 유형이 있습니다. 참조 무결성을 사용하면 항목을 제거할 때 해당 DN이 포함된 모든 속성도 함께 제거됩니다.

예를 들어, 참조 무결성이 활성화된 상태에서 디렉토리의 사용자 항목을 제거하면 서버는 이 사용자가 구성원으로 속해 있는 모든 그룹에서 해당 사용자를 제거합니다. 참조 무결성을 사용하지 않으면 관리자가 수동으로 이 사용자를 그룹에서 제거해야 합니다. 이 기능은 디렉토리를 사용하여 사용자 및 그룹을 관리하는 다른 Sun Java System 제품과 Directory Server를 통합할 때 특히 유용합니다.

참조 무결성 작동 방식

활성화된 참조 무결성 플러그인은 삭제, 이름 바꾸기 또는 이동 작업 후 즉시 지정된 속성에 대해 무결성 업데이트를 수행합니다. 기본적으로 참조 무결성 플러그인은 사용되지 않습니다.

디렉토리에서 사용자 또는 그룹 항목을 삭제하거나 이름을 바꾸거나 이동하면 해당 작업은 다음과 같이 참조 무결성 로그 파일에 기록됩니다.

```
ServerRoot/slapd-serverID/logs/referint
```

*업데이트 간격*이라는 지정된 시간 후에 서버는 참조 무결성이 활성화된 모든 속성에 대해 검색을 수행하고 검색 결과에 표시된 항목과 로그 파일에 있는 삭제 또는 수정된 항목의 DN을 비교합니다. 로그 파일에 항목이 삭제되었다고 표시되면 해당 속성은 삭제됩니다. 로그 파일에 항목이 변경되었다고 표시되면 해당 속성 값도 이에 따라서 수정됩니다.

활성화된 참조 무결성 플러그인의 기본 구성은 삭제, 이름 바꾸기 또는 이동 작업 후 즉시 member, uniquemember, owner, seeAlso 및 nsroledn 속성에 대해 무결성 업데이트를 수행합니다. 하지만 사용자 요구에 맞게 다음과 같이 참조 무결성 플러그인의 동작을 구성할 수 있습니다.

- 참조 무결성 업데이트를 다른 파일에 기록합니다.
- 업데이트 간격을 수정합니다. 참조 무결성 업데이트로 인한 시스템 영향을 줄이려면 업데이트 간격을 증가시키는 것이 좋습니다.
- 참조 무결성을 적용할 속성을 선택합니다. DN 값이 포함된 속성을 사용하거나 정의하는 경우 참조 무결성 플러그인에서 이러한 속성을 모니터하도록 설정하는 것이 좋습니다.

참조 무결성 구성

참조 무결성을 활성화 또는 비활성화하거나 Directory Server 콘솔에서 이 플러그 인을 구성하려면 아래 절차를 사용합니다.

주 참조 무결성 플러그 인에서 사용하는 모든 데이터베이스의 속성은 모두 색인화되어 있어야 합니다. 모든 데이터베이스의 구성에서 색인을 작성해야 합니다. 레트로 변경 로그가 활성화되어 있을 때는 cn=changeLog 접미어가 색인화되어 있어야 합니다. 자세한 내용은 [10장, "디렉토리 데이터 색인화"](#)를 참조하십시오.

콘솔에서 참조 무결성 구성

1. Directory Server 콘솔의 최상위 구성 탭에서 플러그 인 노드를 확장하여 "Referential Integrity Postoperation" 플러그 인을 선택합니다.
오른쪽 패널에 플러그 인 설정이 표시됩니다.
2. 플러그 인을 활성화하려면 플러그 인 사용 확인란을 선택하고 플러그 인을 비활성화하려면 플러그 인 사용 확인란을 선택 취소합니다.
3. 인수 1의 값을 설정하여 업데이트 간격(초)을 수정합니다. 일반적인 값은 다음과 같습니다.
 - 0 - 모든 작업 후 즉시 업데이트. 이것이 기본값이며 모든 삭제 및 수정 작업 후에 즉시 참조 무결성 검사를 수행하면 서버 성능이 크게 저하될 수 있습니다.
 - 90 - 90초마다 업데이트
 - 3600 - 매 시간마다 업데이트
 - 10,800 - 3시간마다 업데이트
 - 28,800 - 8시간마다 업데이트
 - 86,400 - 매일 한 번 업데이트
 - 604,800 - 매주 한 번 업데이트
 무결성과 전체 성능을 고려하여 적절한 양수 값을 설정합니다.
4. 인수 2의 값을 사용하려는 참조 무결성 로그 파일의 절대 경로로 설정합니다.
인수 3은 사용되지 않지만 반드시 있어야 합니다.

5. 참조 무결성이 모니터되는 속성은 인수 4부터 열거됩니다. 추가 및 삭제 버튼을 눌러 이 목록을 관리하고 자신의 속성을 추가합니다.
6. 저장을 눌러 변경 사항을 저장합니다.
7. 변경 사항을 적용하려면 Directory Server를 다시 시작해야 합니다.

복제에 참조 무결성 사용

복제 환경에서 참조 무결성 플러그 인을 사용하는 경우 다음과 같은 몇 가지 제한 사항이 있습니다.

- 마스터 복제본이 포함된 모든 서버에 대해 활성화해야 합니다.
- 모든 마스터에서 동일한 구성을 사용하여 활성화해야 합니다.
- 허브나 소비자 복제본만 포함된 서버에서 활성화하는 것은 도움이 되지 않습니다.

복제 토폴로지에 참조 무결성 플러그 인을 구성하려면 다음을 수행합니다.

1. 모든 복제본이 구성되고 모든 복제 계약이 정의되어 있는지 확인합니다.
2. 참조 무결성을 유지할 속성 집합을 지정합니다. 마스터 서버에서 사용할 업데이트 간격도 지정합니다.
3. 모든 마스터 서버에서 동일한 속성 집합과 업데이트 간격을 사용하여 참조 무결성 플러그 인을 활성화합니다. 이 절차에 대해서는 [109페이지의 "참조 무결성 구성"](#)에서 설명합니다.
4. 모든 소비자 서버에서 참조 무결성 플러그 인을 비활성화합니다.

레거시 복제에 참조 무결성 사용

참조 무결성을 사용하여 4.x 마스터에서 5.x 소비자로 복제하는 경우 4.x 마스터에서 참조 무결성 플러그 인을 다시 구성하여 참조 무결성 변경 사항을 4.x 변경 로그에 써야 합니다. 이렇게 하면 참조 무결성 변경 사항이 복제됩니다. *플러그 인을 다시 구성하지 않으면 참조 무결성이 올바르게 작동하지 않습니다.*

이 환경에서 참조 무결성 플러그 인을 다시 구성하려면 다음을 수행합니다.

1. 4.x 서버를 중지합니다.
2. `ServerRoot/slapd-ServerID/config/`에 있는 `slapd.ldbm.conf` 파일을 엽니다.

- 다음과 같이 시작되는 줄을 찾습니다.

```
plugin postoperation on "referential integrity postoperation"
```

- 속성 목록 바로 앞에 있는 인수를 **0**에서 **1**로 변경하여 이 줄을 수정합니다.

예를 들어, 다음과 같은 줄이 있다고 가정합니다.

```
plugin postoperation on "referential integrity postoperation"
"ServerRoot/lib/referint-plugin.dll" referint_postop_init 0
"ServerRoot/slapd-serverID/logs/referint" 0 "member" "uniquemember"
"owner" "seeAlso"
```

이 줄을 다음과 같이 변경합니다.

```
plugin postoperation on "referential integrity postoperation"
"ServerRoot/lib/referint-plugin.dll" referint_postop_init 0
"ServerRoot/slapd-serverID/logs/referint" 1 "member" "uniquemember"
"owner" "seeAlso"
```

- slapd.ldbm.conf 파일을 저장합니다.
- 서버를 다시 시작합니다.
- 4.x 공급자에서 5.x 소비자를 다시 초기화합니다.

디렉토리 검색

LDAP 클라이언트를 사용하여 디렉토리에서 항목을 찾을 수 있습니다. 대부분의 클라이언트는 디렉토리를 검색하고 항목 정보를 검색할 수 있는 특정 형식의 검색 인터페이스를 제공합니다.

디렉토리에 설정된 액세스 제어에 따라 검색 결과가 결정됩니다. 일반 사용자에게는 디렉토리의 상당 부분이 "표시"되지 않지만, 디렉토리 관리자는 구성을 포함한 모든 데이터에 액세스할 수 있습니다.

ldapsearch를 사용한 디렉토리 검색

ldapsearch 명령줄 유틸리티를 사용하여 디렉토리 항목을 찾고 검색할 수 있습니다. 이 절에서 설명하는 ldapsearch 유틸리티는 Solaris 플랫폼과 함께 제공되는 유틸리티가 아닌, Directory Server Resource Kit의 일부입니다. 이 유틸리티에 대한 자세한 내용은 *Directory Server Resource Kit Tools Reference*를 참조하십시오.

이 유틸리티는 지정된 사용자 아이디(일반적으로 고유 이름)와 비밀번호를 사용하여 서버에 연결한 다음, 검색 필터를 기반으로 항목을 찾습니다. 단일 항목, 항목의 직계 하위 항목, 전체 트리 또는 하위 트리 등을 검색 범위로 지정할 수 있습니다.

검색 결과는 LDIF 형식으로 반환됩니다.

ldapsearch 명령줄 형식

ldapsearch를 사용하는 경우 다음 형식으로 명령을 입력해야 합니다.

```
ldapsearch [optional_options] [search_filter] [optional_list_of_attributes]
```

여기서

- *optional_options*는 일련의 명령줄 옵션을 나타냅니다. 이러한 옵션은 검색 필터 앞에 지정해야 합니다(있는 경우).
- *search_filter*는 118페이지의 "LDAP 검색 필터"에 설명된 것과 같은 LDAP 검색 필터를 나타냅니다. -f 옵션을 사용하여 검색 필터를 파일로 제공하지 않는 경우 검색 필터를 지정해야 합니다.
- *optional_list_of_attributes*는 공백으로 분리된 속성 목록을 나타냅니다. 속성 목록을 지정하면 검색 결과로 반환되는 속성 수가 줄어듭니다. 이 속성 목록은 검색 필터 뒤에 와야 합니다. 예를 보려면 117페이지의 "속성의 하위 집합 표시"를 참조하십시오. 속성 목록을 지정하지 않으면 디렉토리에 설정된 액세스 제어에서 허용하는 모든 속성 값이 검색 결과로 반환합니다(작동 가능 속성 제외).

주 작동 가능 속성이 검색 결과로 반환되게 하려면 검색 명령에서 명시적으로 지정해야 합니다. 명시적으로 지정된 작동 가능 속성과 일반 속성을 모두 검색하려면 ldapsearch 명령의 속성 목록에 별표(*)를 사용합니다.

특수 문자 사용

ldapsearch 명령줄 유틸리티를 사용할 때 명령줄 해석기에 특별한 의미가 있는 특정 문자(예: 공백[, 별표*, 역슬래시\ 등)가 포함된 값을 지정해야 할 수도 있습니다. 이런 경우에는 특수 문자를 지정할 때 값을 인용 부호("")로 묶어서 입력합니다. 예를 들면 다음과 같습니다.

```
-D "cn=Charlene Daniels,ou=People,dc=example,dc=com"
```

사용하는 명령줄 해석기에 따라 작은따옴표나 큰따옴표를 이스케이프 문자로 사용합니다. 자세한 내용은 셸 설명서를 참조하십시오.

자주 사용되는 ldapsearch 옵션

아래에는 자주 사용되는 ldapsearch 명령줄 옵션이 열거되어 있습니다. 공백 []이 포함된 값을 지정하는 경우 -b "ou=groups, dc=example,dc=com"과 같이 값을 큰따옴표로 묶어야 합니다.

-b 검색 시작점을 지정합니다. 현재 데이터베이스에 있는 고유 이름을 값으로 지정해야 합니다. LDAP_BASEDN 환경 변수가 기본 DN으로 설정되어 있으면 이 옵션은 선택 사항입니다.

이 옵션에 값을 지정할 때는 큰따옴표로 묶어야 합니다. 예를 들면 다음과 같습니다.

```
-b "cn=Charlene Daniels, ou=People, dc=example,dc=com"
```

-D 인증할 때 사용할 고유 이름을 서버에 지정합니다. 서버에서 익명 액세스를 지원하면 이 옵션은 선택 사항입니다. 옵션을 사용하려면 Directory Server에서 인식되는 DN으로, 항목을 검색할 수 있는 권한이 있는 값을 지정해야 합니다. 예를 들면 다음과 같습니다.

```
-D "uid=cdaniels, dc=example,dc=com"
```

-h Directory Server가 설치된 시스템의 IP 주소나 호스트 이름을 지정합니다. 호스트를 지정하지 않으면 ldapsearch는 localhost를 사용합니다. 예를 들어, -h myServer와 같이 지정합니다.

-l 검색 요청이 완료될 때까지 기다리는 최대 초 수를 지정합니다. 여기에 지정한 값과 관계없이 `ldapsearch`는 결코 서버의 `nsslapd-timelimit` 속성이 허용하는 시간보다 오래 기다리지 않습니다(영구 검색 제외). 지속적인 검색에 대한 자세한 내용은 *Directory Server Resource Kit Tools Reference*를 참조하십시오.

예를 들어, `-l 300`과 같이 지정합니다. `nsslapd-timelimit` 속성의 기본값은 3,600초(1시간)입니다.

-p `Directory Server`에서 사용하는 TCP 포트 번호를 지정합니다. 예를 들어, `-p 5201`과 같이 지정합니다. 기본값은 389이고, SSL 옵션을 사용하는 경우 636입니다.

-s 검색 범위를 지정합니다. 범위는 다음 중 하나일 수 있습니다.

- `base` — `-b` 옵션에 지정된 항목이나 `LDAP_BASEDN` 환경 변수로 정의된 항목만 검색합니다.
- `one` — `-b` 옵션에 지정된 항목의 직계 하위 항목만 검색합니다. 하위 항목만 검색되고 `-b` 옵션에 지정된 실제 항목은 검색되지 않습니다.
- `sub` — `-b` 옵션에 지정된 항목과 모든 하위 항목을 검색합니다. 즉, `-b` 옵션에 지정된 지점에서 시작하여 하위 트리 검색을 수행합니다. 이것이 기본값입니다.

-w `-D` 옵션에 지정된 고유 이름에 해당하는 비밀번호를 지정합니다. 이 옵션을 지정하지 않으면 익명 액세스가 사용됩니다. 예를 들어, `-w diner892`와 같이 지정합니다.

-x 검색 결과가 클라이언트가 아닌 서버에서 정렬되도록 지정합니다. 이 옵션은 국제적인 검색과 같이 일치 규칙에 따라 정렬하려는 경우에 유용합니다. 서버측 정렬은 여러 자원을 사용하지만 일반적으로 클라이언트보다 서버에서 정렬하는 것이 더 빠릅니다.

-z 검색 요청에 응답하여 반환하는 최대 항목 수를 지정합니다. 예를 들어, `-z 1000`과 같이 지정합니다.

일반적으로 여기에 지정한 값과 관계없이 `ldapsearch`는 결코 서버의 `nsslapd-sizelimit` 속성에서 허용하는 수보다 많은 항목을 반환하지 않습니다. 그러나 이 명령줄 인수를 사용할 때 루트 DN으로 바인드하여 이 제한을 무시할 수 있습니다. 루트 DN으로 바인드하면 이 옵션은 기본적으로 0으로 설정됩니다. `nsslapd-sizelimit` 속성의 기본값은 2,000개의 항목입니다.

모든 `ldapsearch` 유틸리티 옵션에 대한 자세한 내용은 *Directory Server Resource Kit Tools Reference*를 참조하십시오.

ldapsearch 예

아래 예에서는 다음과 같이 가정합니다.

- 디렉토리에 있는 모든 항목을 검색하려고 합니다.
- 서버는 호스트 이름 **myServer**에 있습니다.
- 서버는 포트 번호 **5201**을 사용합니다.
- **password** 비밀번호를 사용하는 **Directory Manager**로 디렉토리에 바인드합니다.
- 서버 포트 **636**(기본 SSL 포트 번호)에 대해 SSL을 사용합니다.
- 모든 데이터가 저장되는 접미어는 **dc=example,dc=com**입니다.

모든 항목 반환

앞에서 가정한 내용에 따라 다음 호출은 디렉토리에 있는 모든 항목을 반환합니다.

```
ldapsearch -h myServer -p 5201 -D "cn=directory manager" -w password
-b "dc=example,dc=com" -s sub "(objectclass=*)"
```

"(objectclass=*)"은 디렉토리에 있는 모든 항목과 일치하는 검색 필터입니다.

명령줄에서 검색 필터 지정

명령줄에서 직접 검색 필터를 지정할 수 있습니다. 이렇게 하려면 필터를 인용 부호로 묶고("filter"), `-f` 옵션을 지정하지 마십시오.

예를 들면 다음과 같습니다.

```
ldapsearch -h myServer -p 5201 -D "cn=directory manager" -w password
-b "dc=example,dc=com" "(cn=Charlene Daniels)"
```

루트 DSE 항목 검색

루트 DSE는 지원되는 접미어 목록, 사용 가능한 인증 기법 등 현재 서버 인스턴스와 관련된 정보가 포함된 특수 항목입니다. 검색 기준을 ""로 지정하여 이 항목을 검색할 수 있습니다. 또한, 검색 범위를 `base`로, 필터를 "(objectclass=*)"로 지정해야 합니다.

예를 들면 다음과 같습니다.

```
ldapsearch -h myServer -p 5201 -D "cn=directory manager" -w password
-b "" -s base "(objectclass=*)"
```

스키마 항목 검색

Directory Server는 모든 디렉토리 서버 스키마를 특수 `cn=schema` 항목에 저장합니다. 이 항목에는 디렉토리 서버에 정의된 모든 객체 클래스와 속성에 대한 정보가 포함됩니다.

다음과 같이 이 항목의 내용을 확인할 수 있습니다.

```
ldapsearch -h myServer -p 5201 -D "cn=directory manager" -w password
-b "cn=schema" -s base "(objectclass=*)"
```

주 엄격한 준수를 위해 지정된 항목에 대한 스키마 하위 항목의 위치는 `subschemaSubentry` 작동 가능 속성으로 지정됩니다. 이 버전의 Directory Server에서는 속성 값이 항상 `cn=schema`로 설정됩니다.

LDAP_BASEDN 사용

편리한 검색을 위해 `LDAP_BASEDN` 환경 변수를 사용하여 검색 기준을 설정할 수 있습니다. 이렇게 하면 `-b` 옵션을 사용하여 검색 기준을 지정하지 않아도 됩니다. 환경 변수 설정 방법에 대한 자세한 내용은 운영 체제 설명서를 참조하십시오.

일반적으로 디렉토리의 접미어 값에 `LDAP_BASEDN`을 설정합니다. 디렉토리 접미어는 루트, 즉 디렉토리의 최상위 항목과 같기 때문에 이렇게 하면 모든 검색이 디렉토리의 루트 항목에서 시작합니다.

예를 들어, `LDAP_BASEDN`을 `dc=example,dc=com`으로 설정한 경우 다음 명령줄 호출을 사용하여 디렉토리에서 (`cn=Charlene Daniels`)를 검색할 수 있습니다.

```
ldapsearch -h myServer -p 5201 -D "cn=directory manager" -w password
"(cn=Charlene Daniels)"
```

이 예에서는 `-s` 옵션을 사용하여 범위를 지정하지 않았기 때문에 기본 범위인 `sub`가 사용됩니다.

속성의 하위 집합 표시

ldapsearch 명령은 모든 검색 결과를 LDIF 형식으로 반환합니다. 기본적으로 ldapsearch는 항목의 고유 이름과 읽을 수 있는 모든 속성을 반환합니다. 지정된 디렉토리 항목에 대한 속성의 하위 집합만 읽을 수 있도록 디렉토리 액세스 제어를 설정할 수 있습니다. 작동 가능 속성만 반환되지 않습니다. 작동 가능 속성이 검색 결과로 반환되게 하려면 검색 명령에서 명시적으로 지정해야 합니다. 작동 가능 속성에 대한 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.

모든 속성이 검색 결과로 반환되는 것을 원하지 않는다고 가정합니다. 검색 필터 바로 뒤의 명령줄에서 원하는 속성을 지정하여 반환되는 속성을 몇 개의 특정 속성으로만 제한할 수 있습니다. 예를 들어, 디렉토리에 있는 모든 항목의 cn 속성과 sn 속성을 표시하려면 아래 명령을 실행합니다.

```
ldapsearch -h myServer -p 5201 -D "cn=directory manager" -w password
"(objectclass=*)" sn cn
```

이 예에서는 검색 기준을 LDAP_BASEDN으로 설정했다고 가정합니다.

여러 값을 갖는 속성 검색

검색 중에 Directory Server에서 여러 값을 갖는 속성을 반드시 정렬하여 반환하지는 않습니다. 예를 들어, 변경 사항이 적용되려면 서버를 다시 시작해야 하는 cn=config에 대한 구성 속성을 검색한다고 가정합니다.

```
ldapsearch -h myServer -p 5201 -D "cn=directory manager" -w password
-b cn=config "(objectclass=*)" nsslapd-requiresrestart
```

다음 결과가 반환됩니다.

```
dn: cn=config
nsslapd-requiresrestart: cn=config:nsslapd-port
nsslapd-requiresrestart: cn=config:nsslapd-secureport
nsslapd-requiresrestart: cn=config:nsslapd-plugin
nsslapd-requiresrestart: cn=config:nsslapd-changelogdir
nsslapd-requiresrestart: cn=config:nsslapd-changelogsuffix
nsslapd-requiresrestart: cn=config:nsslapd-changelogmaxentries
nsslapd-requiresrestart: cn=config:nsslapd-changelogmaxage
nsslapd-requiresrestart: cn=config:nsslapd-db-locks
nsslapd-requiresrestart: cn=config:nsslapd-return-exact-case
nsslapd-requiresrestart: cn=config,cn=ldb database,cn=plugins,
cn=config:nsslapd-allidsthreshold
nsslapd-requiresrestart: cn=config,cn=ldb database,cn=plugins,
cn=config:nsslapd-dbcachesize
nsslapd-requiresrestart: cn=config,cn=ldb database,cn=plugins,
cn=config:nsslapd-dbncache
nsslapd-requiresrestart: cn=config,cn=ldb database,cn=plugins,
cn=config:nsslapd-directory
```

```
nsslapd-requiresrestart:
cn=encryption,cn=config:nssslsessiontimeout
nsslapd-requiresrestart: cn=encryption,cn=config:nssslclientauth
nsslapd-requiresrestart: cn=encryption,cn=config:nssslserverauth
nsslapd-requiresrestart: cn=encryption,cn=config:nsssl2
nsslapd-requiresrestart: cn=encryption,cn=config:nsssl3
...
```

여기에 표시된 것처럼 `nsslapd-requiresrestart` 속성은 여러 값을 갖지만 값이 정렬되어 있지 않습니다. 여러 값을 갖는 속성을 정렬해서 표시하는 응용 프로그램을 개발하려면 응용 프로그램에서 정렬을 수행하도록 해야 합니다.

검색 시 클라이언트 인증 사용

이 예에서는 클라이언트 인증을 사용하여 디렉토리를 검색하는 사용자 `cdaniels`를 보여줍니다.

```
ldapsearch -h myServer -p 636 -b "dc=example,dc=com"
-N "cdanielsscrtname" -Z -W certdbpassword
-P /home/cdaniels/certdb/cert.db "(givenname=Richard)"
```

LDAP 검색 필터

검색 필터는 검색 작업에 대해 반환되는 항목을 선택하며 `ldapsearch` 명령줄 유틸리티에서 자주 사용됩니다. `ldapsearch`를 사용할 때 여러 개의 검색 필터를 각 줄에 하나씩 입력하여 한 파일에 저장하거나 명령줄에서 직접 검색 필터를 지정할 수 있습니다.

예를 들어, 다음 필터는 일반 이름 `Lucie Du Bois`에 대한 검색을 지정합니다.

```
(cn=Lucie Du Bois)
```

이 검색 필터는 일반 이름 `Lucie Du Bois`가 포함된 모든 항목을 반환합니다. 일반 이름 값 검색은 대소문자를 구분하지 않습니다.

일반 이름 속성에 언어 태그와 관련된 값이 있으면 모든 값이 반환됩니다. 따라서 다음 두 속성 값은 이 필터와 모두 일치합니다.

```
cn: Lucie Du Bois
cn;lang-fr: Lucie Du Bois
```

검색 필터 구문

검색 필터의 기본 구문은 다음과 같습니다.

(attribute operator value)

예를 들면 다음과 같습니다.

(buildingname>=alpha)

이 예에서 `buildingname`은 속성, `>=`은 연산자, `alpha`는 값입니다. 여러 속성을 부울 연산자로 결합하여 필터를 정의할 수도 있습니다.

검색 필터에 대해서는 다음 절에서 자세히 설명합니다.

- [검색 필터에 속성 사용](#)
- [검색 필터에 연산자 사용](#)
- [검색 필터에 OID 사용](#)
- [복합 검색 필터 사용](#)
- [파일을 사용하여 검색 필터 지정](#)
- [검색 필터에서 7비트 ASCII 이외의 문자 지정](#)
- [검색 필터 예](#)

검색 필터에 속성 사용

항목을 검색할 때 해당 항목 유형과 관련된 속성을 지정할 수 있습니다. 예를 들어, `people` 항목을 검색할 때 `cn` 속성을 사용하여 특정 일반 이름을 가진 사용자를 검색할 수 있습니다.

`people` 항목에는 다음과 같은 속성이 포함될 수 있습니다.

- `cn`(사용자의 일반 이름)
- `sn`(사용자의 성)
- `telephoneNumber`(사용자의 전화 번호)
- `buildingName`(사용자가 상주하는 건물 이름)
- `l`(사용자를 찾을 수 있는 구/군/시)

각 항목 유형과 관련된 속성 목록은 *Directory Server Administration Reference*를 참조하십시오.

검색 필터에 연산자 사용

표 2-2에는 검색 필터에 사용할 수 있는 연산자가 나와 있습니다.

표 2-2 검색 필터 연산자

검색 유형	연산자	설명
동일	=	지정된 값과 정확히 일치하는 속성 값을 포함한 항목을 반환합니다. 예를 들어, cn=Bob Johnson과 같이 지정합니다.
하위 문자열	=string* string	지정된 하위 문자열이 포함된 속성을 가진 항목을 반환합니다. 예를 들면 다음과 같습니다. cn=Bob* cn=*Johnson cn=*John* cn=B*John 별표(*)는 0개 이상의 문자를 나타냅니다.
크거나 같음	>=	지정된 값보다 크거나 같은 속성 값을 포함한 항목을 반환합니다. 예를 들면 다음과 같습니다. buildingname >= alpha
작거나 같음	<=	지정된 값보다 작거나 같은 속성 값을 포함한 항목을 반환합니다. 예를 들면 다음과 같습니다. buildingname <= alpha
있음	=*	지정된 속성에 대한 값이 하나 이상 포함된 항목을 반환합니다. 예를 들면 다음과 같습니다. cn=* telephonenumber=* manager=*
근사	~=	검색 필터에서 지정한 값과 대략 같은 값을 가진 지정된 속성을 포함한 항목을 반환합니다. 예를 들면 다음과 같습니다. cn~=suret l~=san francisco could return cn=sarette l=san francisco 근사 연산자는 아직 시험 단계이며 영문 문자열에서만 작동합니다. Ja 또는 Zn과 같은 비ASCII 기반의 문자열에서는 작동하지 않습니다.

확장 연산자는 cn:dn:=John과 같이 검색을 dn 속성으로 확장하고 국제적인 검색을 지원합니다.

검색 필터에 OID 사용

LDAPv3을 사용하면 특정 속성에 대한 일치 연산자와 규칙을 만들 수 있습니다. 일치 규칙은 속성 값과 특정 구문을 비교하는 방법을 정의합니다. 다시 말해서 잠재적으로 일치하는 속성을 비교하는 방법을 정의하는 것입니다. 예를 들어 일치 규칙은 속성 비교 시 텍스트 대소문자를 적용할 것인지 여부를 정의할 수 있습니다.

규칙이 만들어질 때 검색 필터에서 이를 참조할 수 있습니다.

예를 들어 다음 검색 필터는 OID 2.5.13.5에서 설계한 일치 규칙을 사용하여 "jensen"이라는 성을 포함하는 항목을 비교합니다.

```
(sn:2.5.13.5:=Jensen)
```

다음 예는 비교 시 OID 2.5.13.5가 사용되어야 한다는 것과, 일치를 평가할 때 항목의 고유 이름 속성을 해당 항목의 일부로 고려해야 한다는 사실을 나타내기 위해 ":dn" 표기를 사용한 예를 보여줍니다.

```
(sn:dn:2.5.13.5:=Jensen)
```

복합 검색 필터 사용

다음과 같이 부울 연산자를 접두어에 사용하여 여러 개의 검색 필터 구성 요소를 결합할 수 있습니다.

```
(Boolean-operator (filter) (filter) (filter) ...)
```

여기서 *Boolean-operator*는 표 2-3에 열거된 부울 연산자 중 하나입니다.

부울 연산자를 결합하고 중첩하여 다음과 같은 복잡한 표현식을 작성할 수 있습니다.

```
(Boolean-operator (filter) (Boolean-operator (filter) (filter) ) )
```

검색 필터에 사용할 수 있는 부울 연산자는 다음과 같습니다.

표 2-3 검색 필터 부울 연산자

연산자	기호	설명
AND	&	명령문이 참이 되려면 지정된 모든 필터가 true여야 합니다. 예를 들면 다음과 같습니다. (&(filter) (filter) (filter) ...)

표 2-3 검색 필터 부울 연산자(계속)

연산자	기호	설명
OR		명령문이 참이 되려면 지정된 필터 중 적어도 한 개는 true여야 합니다. 예를 들면 다음과 같습니다. ((filter) (filter) (filter) ...)
NOT	!	명령문이 참이 되려면 지정된 명령문이 true가 아니어야 합니다. NOT 연산자는 한 개의 필터에만 적용됩니다. 예를 들면 다음과 같습니다. (!(filter))

부울 표현식은 다음 순서로 평가됩니다.

- 가장 안쪽의 괄호 표현식에서 가장 바깥쪽 괄호 표현식으로 평가
- 왼쪽에서 오른쪽으로 평가

파일을 사용하여 검색 필터 지정

명령줄에서 입력하는 대신 검색 필터를 파일에 입력할 수 있습니다. 이렇게 하려면 각 줄에 한 개씩 파일에 검색 필터를 입력합니다. `ldapsearch` 명령은 파일에 열거된 순서대로 각 검색을 실행합니다.

예를 들어 다음 줄이 포함된 파일을 가정합니다.

```
(sn=Daniels)
(givenname=Charlene)
```

이 경우 `ldapsearch`는 먼저 성이 **Daniel**인 모든 항목을 찾은 후 이름이 **Charlene**인 모든 항목을 찾습니다. 두 검색 조건에 모두 일치하는 항목은 두 번 반환됩니다.

예를 들어, 앞의 검색 필터를 `searchdb` 파일에 지정하고 `LDAP_BASEDN`을 사용하여 검색 기준을 설정했다고 가정합니다. 아래 명령은 각 검색 필터에 일치하는 모든 항목을 반환합니다.

```
ldapsearch -h myServer -p 5201 -D "cn=directory manager" -w password
-f searchdb
```

검색 줄 끝에 원하는 속성 이름을 지정하여 반환되는 속성 집합을 제한할 수 있습니다. 예를 들어, 아래 `ldapsearch` 명령은 두 검색을 모두 수행하지만 각 항목의 `DN`과 `givenname` 및 `sn` 속성만 반환합니다.

```
ldapsearch -h myServer -p 5201 -D "cn=directory manager" -w password
-f searchdb sn givenname
```

검색 필터에서 7비트 ASCII 이외의 문자 지정

검색 필터에서 7비트 ASCII 이외의 문자는 UTF-8 인코딩의 각 바이트 앞에 역슬래시(\)가 오도록 해당 문자의 표시를 대체해야 합니다. UTF-8에서 문자는 각 바이트에 해당하는 16진수 코드로 대체됩니다.

예를 들어 é 문자에 UTF-8 표시 c3a9가 있다고 할 때 검색 필터에서는 é를 \c3\a9로 표시합니다. 따라서 cn=Véronique Martin을 검색하려면 다음과 같이 표시합니다.

```
ldapsearch -h myServer -b "dc=example,dc=com" "(cn=V\c3\a9ronique Martin) "
```

표 2-4에 있는 특수 문자도 검색 필터에서 사용될 경우 이와 같은 방식으로 표시되어야 합니다.

표 2-4 검색 필터의 특수 문자

특수 문자	특수 문자가 있는 값	필터 예
*	Five*Star	(cn=Five\2aStar)
\	c:\File	(cn=\5cFile)
()	John(2nd)	(cn=John \282nd\29)
null	0004	(bin=\00\00\00\04)

검색 필터 내 고유 이름의 제어 문자

Directory Server에서 DN을 사용할 때는 역슬래시(\)를 사용하여 기타 특수 문자와 쉼표를 이스케이프해야 합니다. 검색 필터에서 DN을 사용하는 경우에는 DN에서 특수 문자를 이스케이프하는 데 사용한 역슬래시(\)가 \5c로 표시되어야 합니다. 예를 들면 다음과 같습니다.

```
DN: cn=Julie Fulmer,ou=Marketing\,Bolivia,dc=example,dc=com
```

```
검색 필터의 DN: ldapsearch -h myServer -b "dc=example,dc=com"
"(manager=cn=Julie Fulmer,ou=Marketing\5c,Bolivia,dc=example,dc=com) "
```

검색 필터 예

다음 필터는 manager 속성 값이 하나 이상 포함된 항목을 검색하며, 있음 검색이라고도 합니다.

```
(manager=*)
```

다음 필터는 일반 이름 Ray Kultgen이 포함된 항목을 검색하며, 동일 검색이라고도 합니다.

```
(cn=Ray Kultgen)
```

다음 필터는 하위 문자열 X.500이 포함된 설명 속성을 가진 모든 항목을 반환합니다.

```
(description=*X.500*)
```

다음 필터는 조직 구성 단위가 Marketing이고 설명 필드에 하위 문자열 X.500이 포함되지 않은 모든 항목을 반환합니다.

```
(&(ou=Marketing) (!(description=*X.500*)))
```

다음 필터는 조직 구성 단위가 Marketing이고 Julie Fulmer 또는 Cindy Zwaska가 관리자 인 모든 항목을 반환합니다.

```
(&(ou=Marketing) (|(manager=cn=Julie Fulmer,ou=Marketing,dc=example,dc=com)(manager=cn=Cindy Zwaska,ou=Marketing,dc=example,dc=com)))
```

다음 필터는 사용자를 나타내지 않는 모든 항목을 반환합니다.

```
(!(objectClass=person))
```

앞의 필터를 사용하면 성능이 저하되므로 복잡한 검색의 일부로 사용해야 합니다. 다음 필터는 사용자를 나타내지 않고 일반 이름이 printer3b와 유사한 모든 항목을 반환합니다.

```
(&(cn~=printer3b) (!(objectClass=person)))
```

작동 가능 속성 검색

작동 가능 속성이 검색 결과로 반환되게 하려면 검색 명령에서 명시적으로 지정해야 합니다.

```
ldapsearch -h myServer -p 5201 -D "cn=directory manager" -w password "(objectclass=*)" aci
```

명시적으로 지정된 작동 가능 속성과 일반 속성을 모두 검색하려면 작동 가능 속성 외에 "*"를 지정합니다. 예를 들면 다음과 같습니다.

```
ldapsearch -h myServer -p 5201 -D "cn=directory manager" -w password "(objectclass=*)" aci *
```

국가별 디렉토리 검색

검색 작업을 수행할 때는 디렉토리에서 지원하는 조합 순서가 있는 서버의 언어를 기준으로 결과를 정렬하도록 요청할 수 있습니다. 디렉토리에서 지원하는 조합 순서의 목록은 *Directory Server Administration Reference*를 참조하십시오.

이 절에서는 `ldapsearch` 구문의 일치 규칙 필터 부분을 중점적으로 다룹니다. 일반 `ldapsearch` 구문에 대한 자세한 내용은 [118페이지의 "LDAP 검색 필터"](#)를 참조하십시오. `Directory Server` 콘솔의 사용자 및 그룹 부분을 사용하여 국가별 디렉토리를 검색하는 방법에 대한 자세한 내용은 온라인 도움말이나 *Administration Server Administration Guide*를 참조하십시오.

이 절은 다음 내용으로 구성되어 있습니다.

- [일치 규칙 필터 구문](#)
- [지원되는 검색 유형](#)
- [국제적 검색 예](#)

일치 규칙 필터 구문

일치 규칙은 검색 작업 중 디렉토리가 문자열을 비교하는 방법에 대해 특별한 지침을 제공합니다. 국제적인 검색의 경우 일치 규칙에서 시스템에게 검색 작업 수행 시 사용할 조합 순서 및 연산자를 알려줍니다. 일치 규칙 필터의 구문은 다음과 같습니다.

`attr:matchingRule:=value`

여기서,

- `attr`은 검색 중인 항목(예: `cn` 또는 `mail`)에 속한 속성입니다.
- `matchingRule`은 기본 설정한 형식에 따라 조합 순서를 식별하거나 조합 순서 및 관계 연산자를 식별하는 문자열입니다. 일치 규칙 형식에 대한 자세한 내용은 [125페이지의 "일치 규칙 형식"](#)을 참조하십시오.
- `value`는 검색할 속성 값이거나 관계 연산자에 검색할 속성 값을 더한 값입니다. 필터의 값 부분 구문은 사용하는 일치 규칙 형식에 따라 달라집니다.

일치 규칙 형식

검색 필터의 일치 규칙 부분은 다음 방식으로 표시될 수 있습니다.

- 검색의 기준으로 삼을 로케일에 대한 조합 순서의 OID로서
- 검색의 기준으로 삼을 조합 순서와 관련된 언어 태그로서
- 관계 연산자를 나타내는 접미어와 조합 순서의 OID로서
- 관계 연산자를 나타내는 접미어 및 조합 순서와 관련된 언어 태그로서

이러한 작업의 각 구문은 다음 절에서 다룹니다.

- 일치 규칙에 **OID** 사용
- 일치 규칙에 언어 태그 사용
- 일치 규칙에 **OID** 및 접미어 사용
- 일치 규칙에 언어 태그 및 접미어 사용

일치 규칙에 OID 사용

Directory Server에서 지원하는 각 로캘에는 연관된 조합 순서 **OID**가 있습니다. 지원되는 로캘과 관련 **OID**의 목록은 *Directory Server Administration Reference*를 참조하십시오.

다음과 같이 일치 규칙 필터의 일치 규칙 부분에서 조합 순서 **OID**를 사용할 수 있습니다.

```
attr:OID:=(relational_operator value)
```

관계 연산자는 문자열의 값 부분에 포함되어 있으며 공백 하나로 값과 구별됩니다. 예를 들어 스웨덴어 조합 순서의 N4709 이후에 있는 departmentNumber 속성을 모두 검색하려면 다음 필터를 사용합니다.

```
departmentNumber:1.3.6.1.4.1.42.2.27.9.4.129.1:== N4709
```

일치 규칙에 언어 태그 사용

Directory Server에서 지원하는 각 로캘마다 관련 언어 태그가 있습니다. 지원되는 로캘 및 관련 언어 목록은 *Directory Server Administration Reference*를 참조하십시오.

다음과 같이 일치 규칙 필터의 일치 규칙 부분에서 언어 태그를 사용할 수 있습니다.

```
attr:language-tag:=(relational_operator value)
```

관계 연산자는 문자열의 값 부분에 포함되어 있으며 공백 하나로 값과 구별됩니다. 예를 들어 스페인어 조합 순서를 사용하여 estudiante 값이 있는 모든 설명 속성을 검색하려면 다음 필터를 사용합니다.

```
description:es:== estudiante
```

일치 규칙에 OID 및 접미어 사용

관계 연산자 값 쌍을 사용하는 대신 필터의 일치 규칙 부분에서 OID에 특정 연산자를 표시하는 접미어를 추가할 수 있습니다. 다음과 같이 OID와 접미어를 조합합니다.

attr:OID+suffix:=value

예를 들어 독일어 조합 순서에서 Softwareprodukte 값이 있는 businessCategory 속성을 검색하려면 다음 필터를 사용합니다.

businessCategory:1.3.6.1.4.1.42.2.27.9.4.28.1.3:=Softwareprodukte

이전 예에서 **.3**은 동일 접미어입니다.

일치 규칙에 언어 태그 및 접미어 사용

관계 연산자 값 쌍을 사용하는 대신 필터의 일치 규칙 부분에서 언어 태그에 특정 연산자를 표시하는 접미어를 추가할 수 있습니다. 다음과 같이 언어 태그와 접미어를 조합합니다.

attr:language-tag+suffix:=value

예를 들어 프랑스어 조합 순서에서 La Salle 이후 나오는 모든 성을 검색하려면 다음 필터를 사용합니다.

sn:fr.4:=La Salle

일치 규칙 필터에 와일드카드 사용

일치 규칙 필터를 사용하여 하위 문자열 검색을 수행할 때 0개 이상의 문자를 나타내는 와일드카드로서 별표(*) 문자를 사용할 수 있습니다.

예를 들어 k로 시작하고 n으로 끝나는 속성 값을 검색하려면 검색 필터의 값 부분에 k*n을 입력합니다. 마찬가지로 u로 시작하는 모든 속성 값을 검색하려면 검색 필터의 값 부분에 u*을 입력하면 됩니다.

별표(*) 문자가 포함된 값을 검색하려면 별표 문자를 이스케이프해야 합니다.

지원되는 검색 유형

디렉토리 서버는 다음 유형의 국제적 검색을 지원합니다.

- 동일(=)
- 하위 문자열(*)
- 보다 큼(>)
- 크거나 같음(>=)

- 보다 작음(<)
- 작거나 같음(<=)

영어에서만 근사, 발음, 있음 검색이 지원됩니다.

보통의 ldapsearch 검색 작업과 같이 국제적 검색에서도 검색 유형을 정의하는 연산자를 사용합니다. 그러나 국제적 검색을 실행할 때는 검색 문자열의 값 부분에 표준 연산자(=, >=, >, <, <=)를 사용할 수도 있고 필터의 일치 규칙 부분에 접미어(디렉토리 접미어와 다름)라는 특정 연산자 유형을 사용할 수도 있습니다. [128페이지의 표 2-5](#)에는 검색 유형과 연산자, 동일 접미어가 요약되어 있습니다.

표 2-5 검색 유형, 연산자 및 접미어

검색 유형	연산자	접미어
보다 작음	<	.1
작거나 같음	<=	.2
동일	=	.3
크거나 같음	>=	.4
보다 큼	>	.5
하위 문자열	*	.6

국제적 검색 예

다음 절에서는 디렉토리 데이터에서 국제적 검색을 수행하는 방법의 예를 보여줍니다. 각 예에는 가능한 일치 규칙 필터 형식을 모두 제공하고 있으므로 사용자는 형식을 익히고 자신에게 가장 적합한 형식을 선택할 수 있습니다.

보다 작음 예

보다 작음 연산자(<) 또는 접미어(.1)를 사용하여 로컬별 검색을 수행할 때는 특정 조합 순서에서 제공된 속성 이전에 오는 속성 값을 모두 검색합니다.

예를 들어 스페인어 조합 순서에서 *Marquez*라는 성 이전에 오는 모든 성을 검색하려면 다음 일치 규칙 필터 중 하나를 사용합니다.

```
sn:1.3.6.1.4.1.42.2.27.9.4.49.1:=< Marquez
sn:es:=< Marquez
sn:1.3.6.1.4.1.42.2.27.9.4.49.1.1:=Marquez
sn:es.1:=Marquez
```


작거나 같음 예

작거나 같음 연산자(<=) 또는 접미어(.2)를 사용하여 로캘별 검색을 수행할 때는 특정 조합 순서에서 제공된 속성 이전에 오는 모든 속성 값을 검색합니다.

예를 들어 헝가리어 조합 순서에서 CZ422라는 방 번호 이전에 오는 모든 방 번호를 검색하려면 다음 일치 규칙 필터 중 하나를 사용합니다.

```
roomNumber:1.3.6.1.4.1.42.2.27.9.4.88.1: <= CZ422
roomNumber:hu: <= CZ422
roomNumber:1.3.6.1.4.1.42.2.27.9.4.88.1.2: =CZ422
roomNumber:hu.2: =CZ422
```

동일 예

동일 연산자(=) 또는 접미어(.3)를 사용하여 로캘별 검색을 수행할 때는 특정 조합 순서에서 제공된 속성과 일치하는 모든 속성 값을 검색합니다.

예를 들어 독일어 조합 순서에서 Softwareprodukte라는 값이 있는 businessCategory 속성을 모두 검색하려면 다음 일치 규칙 필터 중 하나를 사용합니다.

```
businessCategory:1.3.6.1.4.1.42.2.27.9.4.28.1: = Softwareprodukte
businessCategory:de: = Softwareprodukte
businessCategory:1.3.6.1.4.1.42.2.27.9.4.28.1.3: =Softwareprodukte
businessCategory:de.3: =Softwareprodukte
```

크거나 같음 예

크거나 같음 연산자(>=) 또는 접미어(.4)를 사용하여 로캘별 검색을 수행할 때는 특정 조합 순서에서 제공된 속성 이후에 나오는 모든 속성 값을 검색합니다.

예를 들어 프랑스어 조합 순서에서 Québec 이후에 나오는 모든 지역을 검색하려면 다음 일치 규칙 필터 중 하나를 사용합니다.

```
locality:1.3.6.1.4.1.42.2.27.9.4.76.1: >= Québec
locality:fr: >= Québec
locality:1.3.6.1.4.1.42.2.27.9.4.76.1.4: =Québec
locality:fr.4: =Québec
```

보다 큼 예

보다 큼 연산자(>) 또는 접미어(.5)를 사용하여 로컬별 검색을 수행할 때는 특정 조합 순서에서 제공된 속성 이후에 오는 모든 속성 값을 검색합니다.

예를 들어 체코어 조합 순서에서 *schranka4* 호스트 뒤에 오는 모든 메일 호스트를 검색하려면 다음 일치 규칙 필터 중 하나를 사용합니다.

```
mailHost:1.3.6.1.4.1.42.2.27.9.4.26.1 :=> schranka4
mailHost:cs:=> schranka4
mailHost:1.3.6.1.4.1.42.2.27.9.4.26.1.5:=schranka4
mailHost:cs.5:=schranka4
```

하위 문자열 예

국제적 하위 문자열 검색을 수행할 때는 지정된 조합 순서에서 제공된 패턴과 일치하는 모든 값을 검색합니다.

예를 들어 중국어 조합 순서에서 *ming*으로 끝나는 모든 사용자 아이디를 검색하려면 다음 일치 규칙 필터 중 하나를 사용합니다.

```
uid:1.3.6.1.4.1.42.2.27.9.4.143.1:=* *ming
uid:zh:=* *ming
uid:1.3.6.1.4.1.42.2.27.9.4.143.1.6:=*ming
uid:zh.6:=*ming
```

DSMLv2를 사용한 디렉토리 액세스

다음 예는 DSML 요청을 사용하여 디렉토리에 액세스하고 검색하는 방법을 보여줍니다. DSML 관련 속성의 전체 목록과 DSMLv2 표준에 대한 정보를 보려면 *Directory Server Administration Reference*를 참조하십시오.

이 절은 다음 예로 구성되어 있습니다.

- 빈 익명 DSML "Ping" 요청
- DSML 요청 발급 및 특정 사용자로 바인드
- DSML 검색 요청

이러한 예에서 `content-length`: 헤더는 DSMLv2 요청의 정확한 길이를 포함합니다. 예가 올바르게 작동하려면 사용하는 편집기가 이러한 내용 길이를 지원하는지 확인하거나 적절하게 수정해야 합니다.

빈 익명 DSML "Ping" 요청

DSML 프런트엔드는 기본적으로 비활성화됩니다. 활성화하는 방법은 [56페이지](#)의 "DSML 요청 사용"을 참조하십시오. DSML 프런트엔드가 활성화되었는지 확인하려면 [코드 예 2-1](#)에 표시된 것처럼 빈 DSML 배치 요청을 보냅니다.

코드 예 2-1 빈 익명 DSML 요청

```
POST /dsml HTTP/1.1
content-length: 451
HOST: hostMachine
SOAPAction: ""
Content-Type: text/xml
Connection: close

<?xml version='1.0' encoding='UTF-8'?>
<soap-env:Envelope
  xmlns:xsd='http://www.w3.org/2001/XMLSchema'
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:soap-env='http://schemas.xmlsoap.org/soap/envelope/'>

  <soap-env:Body>
    <batchRequest
      xmlns='urn:oasis:names:tc:DSML:2:0:core'
      requestID='Ping!'>
      <!-- empty batch request -->
    </batchRequest>
  </soap-env:Body>
</soap-env:Envelope>
```

이 DSML 요청의 첫 번째 섹션에는 HTTP 메소드 줄(POST /dsml HTTP/1.1)이 포함되어 있으며 뒤에 HTTP 헤더 수가 표시됩니다. HTTP 메소드 줄은 DSML 프런트엔드에서 사용할 URL과 HTTP 메소드 요청을 지정합니다. DSML 프런트엔드에서 허용하는 유일한 HTTP 메소드 요청은 POST입니다. Directory Server의 기본 URL은 /dsml이지만 다른 유효한 URL을 사용하여 구성할 수 있습니다. 뒤에 나오는 HTTP 헤더는 DSML 요청의 나머지 세부 정보를 지정합니다.

- content-length: 451
SOAP/DSML 요청의 정확한 길이를 지정합니다.
- HOST: hostMachine
연결되는 호스트 Directory Server의 이름을 지정합니다.
- SOAPAction:
필수 사항이며, HTTP/SOAP 스택에서 DSML 요청을 수행할 것을 디렉토리에 알립니다. 그러나 비워둘 수도 있습니다.
- Content-Type: text/xml
내용을 XML로 정의하는 text/xml 값이 있어야 합니다.
- Connection: close
요청이 만족되면 연결을 단도록 지정합니다(기본 HTTP/1.1 동작은 연결을 열어둠).

요청의 나머지 부분은 SOAP/DSML 섹션입니다. DSML 요청은 XML 프롤로그 헤더로 시작합니다.

```
<?xml version='1.0' encoding='UTF-8'?>
```

이 헤더는 요청을 UTF-8 문자 집합으로 인코딩해야 함을 지정합니다. 이 헤더 뒤에는 XML 스키마의 필수 포함, XML 스키마 인스턴스 및 SOAP 이름 공간이 포함된 SOAP 봉투 및 본문 요소가 있습니다.

DSML 배치 요청은 DSML 배치 요청 요소로 시작하며, 바로 뒤에 DSMLv2 이름 공간의 필수 포함이 옵니다.

```
xmlns='urn:oasis:names:tc:DSML:2:0:core'.
```

요청은 선택 사항인 다음 요청 아이디로 식별됩니다.

```
requestID='Ping! '>
```

빈 배치 요청은 다음과 같이 표시됩니다.

```
<!-- empty batch request -->
```

이것은 XML 주석으로 처리되며, SOAP/DSML 배치 요청은 배치 단기 요청, SOAP 본문 단기 및 SOAP 봉투 요소 단기를 사용하여 단합니다.

DSML 프런트엔드를 활성화하면 코드 예 2-2에 나와 있는 바와 같이 빈 DSML 요청이 반환됩니다.

코드 예 2-2 빈 익명 DSML 응답

```
HTTP/1.1 200 OK
Cache-control: no-cache
Connection: close
Date: Mon, 09 Sep 2002 13:56:49 GMT
Accept-Ranges: none
Server: Sun-ONE-Directory/5.2
Content-Type: text/xml; charset="utf-8"
Content-Length: 500

<?xml version='1.0' encoding='UTF-8'?>
<soap-env:Envelope
  xmlns:xsd='http://www.w3.org/2001/XMLSchema'
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:soap-env='http://schemas.xmlsoap.org/soap/envelope/'
  >
  <soap-env:Body>
  <batchResponse
    xmlns:xsd='http://www.w3.org/2001/XMLSchema'
    xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='urn:oasis:names:tc:DSML:2:0:core'
    requestID='Ping!'
    >
  </batchResponse>
  </soap-env:Body>
</soap-env:Envelope>
```

아무 것도 반환되지 않으면 프런트엔드가 비활성화되었다고 확신할 수 있습니다.

동시에 디렉토리에 연결하는 클라이언트 수와 DSML 요청 크기에 대한 최대 제한이 있습니다. 클라이언트 수에 대한 제한은 ds-dsml-poolsize 속성과 ds-dsml-poolmaxsize 속성으로 지정되고 요청 크기 제한은 ds-dsml-requestmaxsize 속성으로 지정됩니다.

DSML 관련 속성에 대한 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.

DSML 요청 발급 및 특정 사용자로 바인드

DSML 요청을 발급하기 위해 지정된 사용자 또는 익명으로 디렉토리에 바인드할 수 있습니다. 지정된 사용자로 바인드하려면 코드 예 2-3에 나와 있는 바와 같이 요청에 DN으로 매핑되는 UID와 비밀번호가 포함된 HTTP 인증 헤더가 있어야 합니다.

코드 예 2-3 DSML 확장 작업: 특정 사용자로 바인드

```
POST /dsml HTTP/1.1
content-length: 578
Content-Type: text/xml; charset="utf-8"
HOST: hostMachine
Authorization: Basic ZWFzdGVyOmVnZw==
SOAPAction: ""
Connection: close

<?xml version='1.0' encoding='UTF-8'?>
<soap-env:Envelope
  xmlns:xsd='http://www.w3.org/2001/XMLSchema'
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:soap-env='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap-env:Body>
    <batchRequest
      xmlns='urn:oasis:names:tc:DSML:2:0:core'>
      <extendedRequest>
        <requestName>1.3.6.1.4.1.4203.1.11.3</requestName>
      </extendedRequest>
    </batchRequest>
  </soap-env:Body>
</soap-env:Envelope>
```

이 예에서 HTTP 인증 헤더는 uid를 더 쉽게 전송하며, 일반 텍스트에서 `easter:egg`로 표시되는 비밀번호 `egg`는 `Authorization: Basic ZWFzdGVyOmVnZw==`와 같이 base64로 인코딩됩니다.

`<extendedRequest>` 태그는 LDAP 확장 작업을 지정하는 데 사용됩니다. `<requestName>` 태그는 확장 작업의 OID를 지정하는 데 사용됩니다. 이 예에서 `OID 1.3.6.1.4.1.4203.1.11.3`은 `whoami` 확장 작업을 나타냅니다.

DSML 확장 작업에 대한 응답으로서 바인드 요청을 수행한 사용자의 DN이 표시됩니다.

코드 예 2-4에서는 다음과 같이 whoami 응답(DN 포함)이 행에 표시됩니다.

```
<response>dn:uid=easter,ou=people,dc=france,dc=sun,dc=com</response>.
```

whoami 확장 작업에 대한 자세한 내용은

<http://www.ietf.org/internet-drafts/draft-zeilenga-ldap-authzid-08.txt>를 참조하십시오.

코드 예 2-4 DSML 확장 작업에 대한 응답

```
HTTP/1.1 200 OK
Cache-control: no-cache
Connection: close
Date: Fri, 30 Jul 2004 09:15:09 GMT
Accept-Ranges: none
Server: Sun-ONE-Directory/5.2
Content-Type: text/xml; charset="utf-8"
Content-length: 697

<?xml version='1.0' encoding='UTF-8'?>
<soap-env:Envelope
  xmlns:xsd='http://www.w3.org/2001/XMLSchema'
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:soap-env='http://schemas.xmlsoap.org/soap/envelope/'
  >
  <soap-env:Body>
  <batchResponse
    xmlns:xsd='http://www.w3.org/2001/XMLSchema'
    xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='urn:oasis:names:tc:DSML:2:0:core' .
    >
    <extendedResponse>
      <resultCode code='0' descr='success' />
      <responseName>1.3.6.1.4.1.4203.1.11.3</responseName>
      <response>dn:uid=easter,ou=people,dc=france,dc=sun,dc=com</response>
    </extendedResponse>
  </batchResponse>
</soap-env:Body>
</soap-env:Envelope>
```

익명 액세스에 엄격한 액세스 제어가 적용되는 경우도 많고 데이터 액세스 제한이 적용될 수도 있지만, 익명 액세스에는 HTTP 인증 헤더가 필요 없습니다. 이와 유사하게, DSML 요청을 발급하여 LDAP 프록시로 LDAP 작업을 수행할 수 있습니다.

DSML 요청은 배치별로 관리되기 때문에 LDAP 프록시로 요청을 발급하는 경우 필수 DSML 프록시 인증 요청이 지정된 요청 배치에서 첫 번째 항목이어야 합니다.

DSML 검색 요청

코드 예 2-5는 루트 DSE 항목에 대한 DSML 기본 객체 검색 요청을 보여줍니다.

코드 예 2-5 DSML 검색 요청


```

POST /dsml HTTP/1.1
HOST: hostMachine
Content-Length: 1081
Content-Type: text/xml
SOAPAction: ""
Connection: close

<?xml version='1.0' encoding='UTF-8'?>
<soap-env:Envelope
  xmlns:xsd='http://www.w3.org/2001/XMLSchema'
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:soap-env='http://schemas.xmlsoap.org/soap/envelope/'
  >
  <soap-env:Body>
    <batchRequest
      xmlns='urn:oasis:names:tc:DSML:2:0:core'
      requestID='Batch of search requests'
      >
      <searchRequest
        dn=""
        requestID="search on Root DSE"
        scope="baseObject"
        derefAliases="neverDerefAliases"
        typesOnly="false"
        >
        <filter>
          <present name="objectClass"/>
        </filter>
        <attributes>
          <attribute name="namingContexts"/>
          <attribute name="supportedLDAPversion"/>
          <attribute name="vendorName"/>
          <attribute name="vendorVersion"/>
          <attribute name="supportedSASLMechanisms"/>
        </attributes>
      </searchRequest>
    </batchRequest>
  </soap-env:Body>
</soap-env:Envelope>

```

이 예에서 각 항목은 다음을 나타냅니다.

- dn=""
requestID="search on Root DSE"
검색 작업이 루트 DSE 항목(빈 DN) 아래의 데이터를 요청하며 선택 사항인 요청 아 이디 속성으로 식별됨을 지정합니다.
- scope="baseObject"
검색이 기본 객체 검색임을 지정합니다.
- derefAliases="neverDerefAliases"
해당 기본 객체를 검색하거나 찾는 동안 별칭 참조가 취소되지 않도록 지정합니다. Directory Server에서 지원하는 유일한 derefAliases 값입니다.
- typesOnly="false"
속성 이름과 해당 값을 모두 반환하도록 지정합니다. typesOnly="true"는 속성 이름 만 반환합니다. 이 속성의 기본값은 false입니다.

필터와 일치하는 항목에 대해 objectclass 필터는 다음과 같이 사용됩니다.

```
<filter>
  <present name="objectClass"/>
</filter>
```

이 필터는 LDAP 필터 문자열(objectclass=*)과 같으며, 뒤에 원하는 속성 목록이 옵니다.

```
<attributes>
  <attribute name="namingContexts"/>
  <attribute name="supportedLDAPversion"/>
  <attribute name="vendorName"/>
  <attribute name="vendorVersion"/>
  <attribute name="supportedSASLMechanisms"/>
</attributes>
```

DSML 검색 응답의 예가 코드 예 2-6에 나와 있습니다.

코드 예 2-6 DSML 검색 응답

```
HTTP/1.1 200 OK
Cache-control: no-cache
Connection: close
Date: Fri, 30 Jul 2004 09:21:43 GMT
Accept-Ranges: none
Server: Sun-ONE-Directory/5.2
Content-Type: text/xml; charset="utf-8"
```

Content-length: 1287

```
<?xml version='1.0' encoding='UTF-8'?>
<soap-env:Envelope
  xmlns:xsd='http://www.w3.org/2001/XMLSchema'
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:soap-env='http://schemas.xmlsoap.org/soap/envelope/'
  >
<soap-env:Body>
<batchResponse
  xmlns:xsd='http://www.w3.org/2001/XMLSchema'
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns='urn:oasis:names:tc:DSML:2:0:core'
  requestID='Batch of search requests'
  >
<searchResponse requestID='search on Root DSE'>
<searchResultEntry>
  <attr name='namingContexts'>
    <value>dc=france,dc=sun,dc=com</value>
    <value>o=NetscapeRoot</value>
  </attr>
  <attr name='supportedLDAPVersion'>
    <value>2</value>
    <value>3</value>
  </attr>
  <attr name='vendorName'>
    <value>Sun Microsystems, Inc.</value>
  </attr>
  <attr name='vendorVersion'>
    <value>Sun-ONE-Directory/5.2</value>
```

```
    </attr>
    <attr name='supportedSASLMechanisms'>
      <value>EXTERNAL</value>
      <value>GSSAPI</value>
      <value>DIGEST-MD5</value>
    </attr>
  </searchResultEntry>
</searchResultDone>
  <resultCode code='0' descr='success' />
</searchResultDone>
</searchResponse>
</batchResponse>
</soap-env:Body>
</soap-env:Envelope>
```

디렉토리 트리 작성

디렉토리 트리는 서버 내의 고유 이름(DN)으로 식별되는 모든 항목으로 구성됩니다. DN의 계층 구조적 특성으로 인해 트리의 데이터를 구조화하는 분기와 리프가 작성됩니다. 디렉토리 트리는 관리 편의상 접미어, 하위 접미어 및 연결 접미어로 정의됩니다. Directory Server 콘솔에서 이러한 모든 요소를 작성하고 관리하기 위한 컨트롤을 제공하지만 명령줄 도구를 사용할 수도 있습니다.

디렉토리 데이터 구성과 일반적인 접미어에 대한 개념 정보는 *Directory Server Deployment Planning Guide*를 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

- 접미어 작성
- 접미어 관리
- 연결 접미어 작성
- 연결 접미어 관리
- 계단식 연결 구성

접미어 작성

Directory Server 콘솔이나 명령줄을 사용하여 루트 접미어와 하위 접미어를 모두 작성할 수 있습니다.

콘솔에서 새 루트 접미어 작성

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드를 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 새 접미어를 선택합니다.

또는 데이터 노드를 선택하고 객체 메뉴에서 새 접미어를 선택할 수도 있습니다.

"새 접미어" 대화 상자가 표시됩니다.

2. 접미어 DN 필드에 접미어 고유 이름을 입력합니다. 두 개 이상의 속성 값 쌍이 쉼표로 구분된 고유 이름 형식을 사용하여 이름을 입력해야 합니다.

관례상, 루트 접미어는 도메인 구성 요소(dc) 이름 지정 속성을 사용합니다. 예를 들어 새로운 접미어 DN을 `dc=example,dc=org`와 같이 입력할 수 있습니다.

주 접미어 이름에는 DN 형식의 속성 값 쌍이 사용되지만 하나의 문자열로 처리됩니다. 따라서 모든 공백은 접미어 이름의 일부로 중요한 의미를 가 집니다.

3. 기본적으로 이 접미어의 데이터베이스 파일 위치는 서버에서 자동으로 선택합니다. 또한 기본적으로 접미어는 시스템 색인만 유지 관리하고 속성이 암호화되거나 복제가 구성되지 않습니다.

기본값을 수정하려면 옵션 버튼을 눌러 새 접미어 옵션을 표시합니다.

- a. 데이터베이스 이름은 데이터베이스 파일이 저장된 디렉토리의 이름이기도 합니다. 기본 데이터베이스 이름은 접미어 DN의 이름 지정 속성 중에서 첫 번째 속성 값으로, 고유하게 숫자가 추가될 수도 있습니다. 다른 이름을 사용하려면 사용자 정의 사용 라디오 버튼을 선택하고 새로운 데이터베이스 고유 이름을 입력합니다.

데이터베이스 이름에는 ASCII(7비트) 영숫자 문자, 하이픈(-) 및 밑줄(_)만 사용할 수 있습니다. 예를 들어 새 데이터베이스의 이름을 `example_2`와 같이 지정할 수 있습니다.

- b. 데이터베이스 파일을 저장할 디렉토리 위치를 선택할 수도 있습니다. 기본적으로 데이터베이스 파일은 아래 경로의 하위 디렉토리에 저장됩니다.

`ServerRoot/slapd-serverID/db`

새 경로를 입력하거나 찾아보기를 눌러 새로운 데이터베이스 디렉토리 위치를 찾습니다. 디렉토리 서버 호스트에서 액세스할 수 있는 경로를 지정해야 합니다.

- c. 신속하게 새 접미어를 구성하기 위해 기존 접미어를 복제할 수도 있습니다. 접미어 구성 복제를 선택하고 드롭다운 메뉴에서 복제할 접미어를 선택합니다. 다음과 같은 복제 구성 중 하나를 선택합니다.
 - 색인 구성 복제 - 새 접미어가 복제된 접미어와 같은 속성 색인을 유지 관리합니다.
 - 속성 암호화 구성 복제 - 새 접미어가 복제된 접미어와 같은 속성 목록에 대해 동일한 암호화 체계의 암호화를 사용합니다.
 - 복제 구성 복제 - 새 접미어가 복제된 접미어와 같은 복제본 유형으로 지정되고, 공급자인 경우 모든 복제 계약이 복사되며, 복제가 활성화됩니다.
- d. 새 접미어 옵션이 모두 구성되면 확인을 누릅니다. 새 접미어 대화 상자에 선택한 모든 옵션이 표시됩니다.

4. 새 접미어 대화 상자에서 확인을 눌러 새 루트 접미어를 작성합니다.

데이터 분기에 이 루트 접미어가 자동으로 표시됩니다. 새 접미어의 추가 구성 방법은 [150페이지의 "접미어 관리"](#)를 참조하십시오.

새 루트 접미어에는 접미어 DN 항목을 비롯하여 항목이 하나도 없으므로 초기화되어 적절한 액세스 권한이 할당될 때까지 디렉토리에서 액세스할 수 없으며 콘솔의 디렉토리 탭에도 표시되지 않습니다.

LDIF 파일을 사용하여 접미어를 초기화하는 경우 나머지 단계는 무시해도 됩니다. 단, 배포에 필요한 액세스 제어 명령(ACI)이 LDIF 파일의 루트 항목에 포함되어 있어야 합니다.

- 5. 콘솔의 최상위 디렉토리 탭을 선택합니다. 새 접미어는 아직 디렉토리 트리에 표시되지 않습니다.
- 6. 디렉토리 관리자만 접미어의 최상위 항목을 만들 수 있는 권한을 갖고 있습니다. 디렉토리 관리자로 로그인하지 않은 경우, 콘솔 > 새 사용자로 로그인 메뉴 항목을 선택하여 디렉토리 관리자로 로그인합니다. 디렉토리 관리자의 DN과 비밀번호를 입력하여 로그인합니다. 기본적으로 디렉토리 관리자 DN은 cn=Directory Manager입니다.
- 7. 디렉토리 트리에서 서버 호스트 이름과 포트가 있는 루트 노드를 마우스 오른쪽 버튼으로 누릅니다. 팝업 메뉴에서 새 루트 객체 항목을 선택하고 새 루트 접미어의 DN을 선택합니다.

또는 디렉토리 트리의 루트 노드를 선택한 다음 객체 메뉴에서 새 루트 객체 항목을 선택합니다.

8. 새 객체 대화 상자에서 루트 객체의 객체 클래스를 하나 선택합니다. 선택한 객체 클래스에 따라 루트 항목에 추가할 수 있는 속성이 결정됩니다.

관계상, dc 이름 지정 속성이 있는 접미어 DN의 루트 객체는 domain 객체 클래스에 속합니다. 일반적으로 루트 객체는 단순 객체로, 데이터가 거의 포함되어 있지 않습니다.

9. 객체 클래스를 선택했으면 새 객체 대화 상자에서 확인을 누릅니다.

이제 콘솔에 새 루트 객체의 일반 편집기가 표시되고 기본 ACI 집합이 자동으로 새 객체에 추가됩니다. 자세한 내용은 [241페이지의 "기본 ACI"](#)를 참조하십시오. ACI 집합을 수정하는 등 토폴로지에 필요한 속성 값을 추가하고 편집합니다.

10. 항목 편집이 끝나면 일반 편집기에서 확인을 눌러 새 접미어의 루트 객체를 작성합니다.

이제 디렉토리 트리에 새 접미어가 표시되며 ACI로 할당된 권한에 따라 콘솔에서 관리할 수 있습니다.

콘솔에서 새 하위 접미어 작성

아래 절차에서는 기존의 루트 접미어 또는 하위 접미어 아래에 새 하위 접미어를 작성하는 방법에 대해 설명합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드와 접미어 노드를 확장하여 부모 접미어를 표시합니다.
2. 부모 접미어 노드를 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 새 하위 접미어를 선택합니다.

또는 부모 접미어 노드를 선택하고 객체 메뉴에서 새 하위 접미어를 선택할 수도 있습니다.

새 하위 접미어 대화 상자가 표시됩니다.

3. 하위 접미어 RDN 필드에 고유 이름을 입력합니다. 두 개 이상의 속성 값 쌍이 쉼표로 구분된 상대적인 고유 이름 형식으로 이름을 입력해야 합니다(예: ou=Contractors).

텍스트 상자 아래 줄에는 RDN 뒤에 부모 접미어 DN이 추가된, 이 하위 접미어의 전체 DN이 표시됩니다.

주 하위 접미어 이름에는 RDN 형식의 속성 값 쌍이 사용되지만 하나의 문자 열로 처리됩니다. 따라서 모든 공백은 접미어 DN의 일부로 중요한 의미를 가집니다.

4. 기본적으로 이 접미어의 데이터베이스 파일 위치는 서버에서 자동으로 선택합니다. 또한 기본적으로 접미어는 시스템 색인만 유지 관리하고 속성이 암호화되거나 복제가 구성되지 않습니다.

기본값을 수정하려면 옵션 버튼을 눌러 새 접미어 옵션을 표시합니다.

- a. 데이터베이스 이름은 데이터베이스 파일이 저장된 디렉토리의 이름이기도 합니다. 기본 데이터베이스 이름은 RDN의 이름 지정 속성 중에서 첫 번째 속성 값으로, 고유하게 숫자가 추가될 수도 있습니다. 다른 이름을 사용하려면 사용자 정의 사용 라디오 버튼을 선택하고 새로운 데이터베이스 고유 이름을 입력합니다.

데이터베이스 이름에는 ASCII(7비트) 영숫자 문자, 하이픈(-) 및 밑줄(_)만 사용할 수 있습니다. 예를 들어 새 데이터베이스의 이름을 `temps-US`와 같이 지정할 수 있습니다.

- b. 데이터베이스 파일을 저장할 디렉토리 위치를 선택할 수도 있습니다. 기본적으로 데이터베이스 파일은 아래 경로의 하위 디렉토리에 저장됩니다.

`ServerRoot/slapd-serverID/db`

새 경로를 입력하거나 찾아보기를 눌러 새로운 데이터베이스 디렉토리 위치를 찾습니다. 디렉토리 서버 응용 프로그램에서 액세스할 수 있는 경로를 지정해야 합니다.

- c. 신속하게 새 하위 접미어를 구성하기 위해 부모 접미어나 다른 기존 접미어를 복제할 수도 있습니다. 접미어 구성 복제를 선택하고 드롭다운 메뉴에서 복제할 접미어를 선택합니다. 다음과 같은 복제 구성 중 하나를 선택합니다.

- 색인 구성 복제 - 새 접미어가 복제된 접미어와 같은 속성 색인을 유지 관리합니다.
- 속성 암호화 구성 복제 - 새 접미어가 복제된 접미어와 같은 속성 목록에 대해 동일한 암호화 체계의 암호화를 사용합니다.

- 복제 구성 복제 - 새 접미어가 복제된 접미어와 같은 복제본 유형으로 지정되고, 공급자인 경우 모든 복제 계약이 복사되며, 복제가 활성화됩니다.

d. 새 접미어 옵션이 모두 구성되면 확인을 누릅니다. 새 하위 접미어 대화 상자에 선택한 모든 옵션이 표시됩니다.

5. 새 하위 접미어 대화 상자에서 확인을 눌러 하위 접미어를 작성합니다.

구성 탭의 부모 접미어 아래에 이 하위 접미어가 자동으로 표시됩니다. 새 접미어의 추가 구성 방법은 [150페이지의 "접미어 관리"](#)를 참조하십시오.

새 하위 접미어에는 RDN 항목을 비롯하여 항목이 하나도 없으므로 초기화되어 적절한 액세스 권한이 할당될 때까지 디렉토리에서 액세스할 수 없으며 콘솔의 디렉토리 탭에도 표시되지 않습니다.

LDIF 파일을 사용하여 접미어를 초기화하는 경우 나머지 단계는 무시해도 됩니다. 단, 배포에 필요한 ACI(Access Control Instruction)가 LDIF 파일의 부모 접미어 및 새 항목에 포함되어 있어야 합니다.

6. 콘솔의 최상위 디렉토리 탭에서 디렉토리 트리를 확장하여 하위 접미어의 부모를 표시합니다. 새 하위 접미어는 아직 표시되지 않습니다.

7. 디렉토리 관리자만 접미어 및 하위 접미어의 최상위 항목을 만들 수 있는 권한(ACI)을 갖고 있습니다. 디렉토리 관리자로 로그인하지 않은 경우, 콘솔 > 새 사용자로 로그인 메뉴 항목을 선택하여 디렉토리 관리자로 로그인합니다. 디렉토리 관리자의 DN과 비밀번호를 입력하여 로그인합니다. 기본적으로 디렉토리 관리자 DN은 `cn=Directory Manager`입니다.

8. 하위 접미어의 부모를 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 새 항목을 선택합니다. 새 객체 목록에서 이 하위 접미어의 RDN에 해당하는 객체 유형을 선택합니다. 예를 들어 `ou=Contractors` 하위 접미어를 작성한 경우 `OrganizationalUnit` 항목을 선택합니다. 작성한 하위 접미어의 객체 클래스가 표시되지 않으면 기타를 선택하여 새 객체 대화 상자의 목록에서 해당 객체 클래스를 선택합니다. 또는 하위 접미어의 부모를 선택한 다음 객체 메뉴에서 새 항목을 선택합니다.

9. 이제 콘솔에 새 객체의 사용자 정의 또는 일반 편집기가 표시됩니다. ACI 집합을 수정하는 등 토폴로지에 필요한 속성 값을 추가하고 편집합니다.

10. 항목 편집이 끝나면 편집기에서 확인을 눌러 새 하위 접미어 항목을 작성합니다.

이제 디렉토리 트리에 새 하위 접미어가 표시되며 ACI로 할당된 권한에 따라 콘솔에서 관리할 수 있습니다.

명령줄에서 접미어 작성

ldapmodify 명령줄 유틸리티를 사용하여 디렉토리에 접미어를 작성할 수도 있습니다. 루트 접미어와 하위 접미어는 서버에서 똑같이 내부적으로 관리되기 때문에 명령줄에서 작성하는 절차도 거의 흡사합니다.

다음 예에서는 cn=Directory Manager가 사용되었지만 구성 항목은 모든 관리 사용자가 만들 수 있습니다. 하지만 접미어의 최상위 항목은 반드시 디렉토리 관리자가 만들어야 합니다.

1. 루트 접미어의 경우 아래 명령을 실행하여 cn=mapping tree, cn=config에 접미어 구성 항목을 작성합니다.

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn="suffixDN",cn=mapping tree,cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsMappingTree
cn: suffixDN
nsslapd-state: backend
nsslapd-backend: databaseName
^D
```

하위 접미어의 경우 동일한 명령에 아래 속성만 추가하여 사용합니다.
nsslapd-parent-suffix: "parentSuffixDN"

suffixDN은 새 접미어의 전체 DN입니다. 관례상, 루트 접미어는 도메인 구성 요소(dc) 이름 지정 속성을 사용합니다(예: dc=example, dc=org). 하위 접미어의 suffixDN에는 하위 접미어의 RDN과 부모 접미어의 DN이 포함됩니다(예: ou=Contractors, dc=example, dc=com).

databaseName은 이 접미어와 관련된, 내부적으로 관리되는 데이터베이스의 이름입니다. 모든 접미어의 databaseName은 고유해야 하며, 관례상 suffixDN의 이름 지정 구성 요소 중에서 첫 구성 요소의 값입니다. databaseName은 접미어의 데이터베이스 파일이 저장된 디렉토리 이름이기도 하므로 ASCII(7비트) 영숫자 문자, 하이픈(-) 및 밑줄(_)만 사용할 수 있습니다.

하위 접미어의 경우 parentSuffixDN은 부모 접미어의 DN과 정확하게 일치합니다.

2. 아래 명령을 실행하여 데이터베이스 구성 항목을 작성합니다.

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=databaseName,cn=ldbm database,cn=plugins,cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsBackendInstance
cn: databaseName
nsslapd-suffix: suffixDN
^D
```

여기서 *databaseName*과 *suffixDN*은 이전 단계에서 사용한 값이어야 합니다.

디렉토리에 이 항목이 추가되면 서버의 데이터베이스 모듈이 자동으로 아래 디렉토리에 데이터베이스 파일을 작성합니다.

ServerRoot/*slapd-serverID*/*db/databaseName*

다른 위치에 데이터베이스 파일을 작성하게 하려면 아래 속성을 사용하여 데이터베이스 구성 항목을 작성합니다.

nsslapd-directory: *path/databaseName*

데이터베이스 파일의 저장 위치에 *databaseName* 디렉토리가 자동으로 작성됩니다.

3. 루트 접미어 또는 하위 접미어의 기본 항목을 작성합니다.

예를 들어 아래 명령을 실행하여 *dc=example,dc=org* 루트 접미어의 기본 항목을 작성할 수 있습니다.

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: dc=example,dc=org
objectclass: top
objectclass: domain
dc: example
^D
```

DN의 첫 이름 지정 속성과 해당 값을 추가해야 합니다. 또한 기본 항목의 객체 클래스 스키마에 필요한 모든 속성도 추가해야 합니다. 관례상, 도메인 구성 요소(*dc*)를 사용하는 루트 접미어 DN은 *domain* 객체 클래스에 속하므로 다른 속성을 추가할 필요가 없습니다.

액세스 정책을 실행하려면 루트 접미어에 액세스 제어 명령(ACI)도 추가해야 합니다. 다음과 같은 *aci* 속성 값을 추가하면 익명 읽기, 보안 자체 수정, 관리자용 전체 액세스 권한 등을 허용할 수 있습니다.

```

aci: (targetattr != "userPassword") (version 3.0; aci
  "Anonymous access";
  allow (read, search, compare)userdn = "ldap:///anyone");
aci: (targetattr != "nsroledn || aci || nsLookThroughLimit ||
  nsSizeLimit || nsTimeLimit || nsIdleTimeout ||
  passwordPolicySubentry || passwordExpirationTime ||
  passwordExpWarned || passwordRetryCount || retryCountResetTime
  || accountUnlockTime || passwordHistory ||
  passwordAllowChangeTime") (version 3.0; aci "Allow self entry
  modification except for nsroledn, aci, resource limit
  attributes, passwordPolicySubentry and password policy state
  attributes"; allow (write)userdn = "ldap:///self");
aci: (targetattr = "*")(version 3.0; aci
  "Configuration Administrator";
  allow (all) userdn = "ldap:///uid=admin,ou=Administrators,
  ou=TopologyManagement, o=NetscapeRoot");
aci: (targetattr = "*")(version 3.0;aci
  "Configuration Administrators Group";
  allow (all) (groupdn =
  "ldap:///cn=Configuration Administrators, ou=Groups,
  ou=TopologyManagement, o=NetscapeRoot");)

```

하위 접미어의 예로, 아래 명령을 실행하여
ou=Contractors,dc=example,dc=com의 기본 항목을 작성할 수 있습니다.

```

ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: ou=Contractors,dc=example,dc=com
objectclass: top
objectclass: organizationalUnit
description: base of separate subsuffix for contractor identities
^D

```

DN의 이름 지정 속성과 해당 값을 추가해야 합니다. 또한 기본 항목의 객체 클래스 스키마에 필요한 모든 속성을 추가해야 하며, 허용되는 다른 속성도 추가할 수 있습니다. 새 하위 접미어가 ACI 범위에 포함되면 부모에 정의된 ACI 액세스 제어가 이 하위 접미어에도 적용됩니다. 하위 접미어에 다른 액세스 정책을 정의하려면 기본 항목을 작성할 때 원하는 aci 속성을 지정합니다.

접미어 관리

접미어를 작성하여 해당 내용을 모두 동시에 관리할 수 있습니다. 이 절에서는 모든 작업 비활성화, 읽기 전용 접미어 설정, 접미어 수준의 참조 작성 등 접미어에 대한 액세스 관리 방법에 대해 설명합니다.

접미어 수준에서 구성되는 다양한 디렉토리 관리 작업에 대해서는 본 설명서의 다른 장에서 설명합니다.

- 188페이지의 "데이터 가져오기"
- 194페이지의 "데이터 내보내기"
- 414페이지의 "색인 관리"
- 103페이지의 "속성 값 암호화"
- 331페이지의 "복제 관리"

접미어 비활성화 또는 활성화

유지 관리를 위해 접미어를 비활성화하거나 보안상 해당 접미어 내용을 사용할 수 없도록 설정해야 하는 경우가 있습니다. 접미어를 비활성화하면 서버에서 접미어에 액세스하는 클라이언트 작업에 응답하여 접미어 내용을 읽거나 쓸 수 없습니다. 기본 참조가 정의되어 있으면 클라이언트가 비활성화된 접미어에 액세스할 때 해당 참조가 반환됩니다.

콘솔에서 접미어 비활성화 또는 활성화

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드를 확장하여 비활성화할 접미어를 선택합니다.
2. 오른쪽 패널에서 설정 탭을 선택합니다. 작성된 모든 접미어는 기본적으로 활성화됩니다.

이 접미어에 대한 복제를 허용하면 해당 탭의 내용이 자동으로 업데이트될 수 있음을 알리는 메시지가 표시됩니다. 복제되는 접미어를 비활성화하면 이 접미어에 대한 복제도 중단됩니다. 복제 중단 기간이 복구 설정보다 짧은 경우 접미어를 다시 활성화하면 복제 기법에서 이 복제본에 대한 업데이트를 계속합니다. 복제 복구 설정은 이 소비자 복제본의 지연 제거와 공급자 변경 로그의 최대 크기 및 수명으로 구성됩니다 (339페이지의 "고급 소비자 구성" 참조).

3. 접미어를 비활성화하려면 "이 접미어에 대한 액세스를 가능하게 합니다" 확인란을 선택 취소하고 접미어를 활성화하려면 확인란을 선택합니다.

4. 저장을 눌러 변경 사항을 적용하면 접미어가 즉시 비활성화 또는 활성화됩니다.
5. 선택 사항으로, 접미어가 비활성화될 경우 이 접미어에 대한 모든 작업에 반환할 전역 기본 참조를 설정할 수도 있습니다. 이 설정은 최상위 구성 탭에서 루트 노드의 네트워크 탭에 있습니다. 자세한 내용은 [100페이지의 "콘솔에서 기본 참조 설정"](#)을 참조하십시오.

명령줄에서 접미어 비활성화 또는 활성화

1. 아래 명령을 실행하여 접미어 구성 항목의 `nsslapd-state` 속성을 편집합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn="suffixDN",cn=mapping tree,cn=config
changetype: modify
replace: nsslapd-state
nsslapd-state: disabled or backend
^D
```

여기서 `suffixDN`은 정의된 바와 같이 공백까지 포함하는 접미어 DN의 전체 문자열입니다. 접미어를 비활성화하려면 `nsslapd-state` 속성을 `disabled` 값으로 설정하고 전체 액세스 권한을 허용하려면 `backend` 값으로 설정합니다.

명령이 성공하면 접미어가 즉시 비활성화됩니다.

2. 선택 사항으로, 접미어가 비활성화될 경우 이 접미어에 대한 모든 작업에 반환할 전역 기본 참조를 설정할 수도 있습니다. 자세한 내용은 [100페이지의 "명령줄에서 기본 참조 설정"](#)을 참조하십시오.

액세스 권한 및 참조 설정

접미어를 완전히 비활성화하지 않고 액세스를 제한하려면 읽기 전용 액세스를 허용하도록 액세스 권한을 수정할 수 있습니다. 이 경우 쓰기 작업을 위해 다른 서버에 대한 참조를 정의해야 합니다. 읽기 및 쓰기 액세스를 모두 거부하고 접미어에 대한 모든 작업에 반환할 참조를 정의할 수도 있습니다.

또한 참조를 사용하여 일시적으로 클라이언트 응용 프로그램을 다른 서버로 연결할 수 있습니다. 예를 들어 접미어 내용을 백업하는 동안 접미어가 다른 서버를 가리키도록 참조를 추가할 수 있습니다.

복제 기법은 쓰기 권한과 참조를 이용하여 복제할 접미어를 구성합니다. 복제를 활성화하거나, 복제본 수준을 올리거나 내리면 참조 설정이 수정됩니다.

주의 접미어가 복제될 경우 참조를 수정하면 이 접미어의 복제된 동작에 영향을 줄 수 있습니다.

콘솔에서 액세스 권한 및 참조 설정

1. **Directory Server** 콘솔의 최상위 구성 탭에서 데이터 노드를 확장하여 참조를 설정할 접미어를 선택합니다.
2. 오른쪽 패널에서 설정 탭을 선택합니다. 연결 접미어를 활성화한 경우에만 권한과 참조를 설정할 수 있습니다. 이 접미어에 대한 복제를 허용하면 해당 탭의 내용이 자동으로 업데이트될 수 있음을 알리는 메시지가 표시됩니다.
3. 다음 라디오 버튼 중 하나를 선택하여 이 접미어 항목에 대한 모든 쓰기 작업에 반환할 응답을 설정합니다.
 - 쓰기와 읽기 요청을 처리합니다 - 기본적으로 이 라디오 버튼이 선택되며 접미어의 정상적인 동작을 나타냅니다. 참조도 정의할 수 있지만 반환되지는 않습니다.
 - 읽기 요청을 처리하고 쓰기 요청에 대한 참조를 반환합니다 - 접미어를 읽기 전용으로 설정하고 쓰기 요청에 대한 참조로 반환할 LDAP URL을 목록에 하나 이상 입력하려면 이 라디오 버튼을 선택합니다.
 - 읽기와 쓰기 요청 모두에 대한 참조를 반환합니다 - 읽기와 쓰기 액세스를 모두 거부하려면 이 라디오 버튼을 선택합니다. 전역 기본 참조를 사용하지 않고 특별히 이 접미어에 대한 참조를 정의할 수 있다는 점을 제외하면 이 동작은 접미어에 대한 액세스를 비활성화하는 것과 유사합니다.
4. 추가 및 제거 버튼을 사용하여 참조 목록을 편집합니다. 추가 버튼을 누르면 새 참조의 LDAP URL을 작성할 수 있는 대화 상자가 표시됩니다. 원격 서버 분기의 DN에 대한 참조를 작성할 수도 있습니다. LDAP URL의 구조에 대한 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.
 여러 개의 참조를 입력할 수 있습니다. 디렉토리는 클라이언트 응용 프로그램의 요청에 응답하여 이 목록에 있는 모든 참조를 반환합니다.
5. 저장을 눌러 변경 사항을 적용하면 새로운 권한 및 참조 설정이 즉시 실행됩니다.

명령줄에서 액세스 권한 및 참조 설정

참조란 레이블이 지정된 URL, 즉 뒤에 공백 문자 및 레이블이 올 수 있는 LDAP URL입니다. 공백 문자가 중요하기 때문에 참조의 URL 부분에 있는 모든 공백 문자는 %20을 사용하여 이스케이프시켜야 합니다.

아래 명령에서 *suffixDN*은 정의된 바와 같이 공백까지 포함하는 접미어 DN의 전체 문자열입니다. *LDAPURL*은 아래 예와 같이 대상의 호스트 이름, 포트 번호 및 DN이 포함된 유효한 URL입니다.

```
ldap://phonebook.example.com:389/ou=All%20People,dc=example,dc=com
```

1. 아래 명령을 실행하여 접미어의 구성 항목을 편집합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn="suffixDN",cn=mapping tree,cn=config
changetype: modify
replace: nsslapd-state
nsslapd-state: referral on update or referral
-
add: nsslapd-referral
nsslapd-referral: LDAPURL
^D
```

마지막 변경 명령문을 반복하여 다른 LDAP URL을 *nsslapd-referral* 속성에 원하는 대로 추가할 수도 있습니다.

nsslapd-state 속성 값이 *referral on update*이면 접미어는 읽기 전용으로 설정되고 쓰기 작업에 대한 참조로 모든 LDAP URL이 반환됩니다. 속성 값이 *referral*이면 읽기와 쓰기 작업이 모두 거부되고 모든 요청에 대해 참조가 반환됩니다.

2. 명령이 성공하면 접미어가 즉시 읽기 전용으로 설정되거나 액세스할 수 없게 되고 참조를 반환할 수 있습니다.

접미어 삭제

접미어를 삭제하면 전체 분기가 디렉토리에서 제거됩니다. 디렉토리에서 부모 접미어만 삭제하고 해당 하위 접미어를 새로운 루트 접미어로 유지할 수도 있습니다.

주의	접미어를 삭제하면 모든 항목이 영구적으로 디렉토리에서 제거되며 복제 구성을 비롯한 모든 접미어 구성도 제거됩니다.
-----------	---

콘솔에서 접미어 삭제

1. Directory Server 콘솔의 구성 탭에서 데이터 노드를 확장합니다.
2. 제거할 접미어를 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 삭제를 선택합니다.
또는 접미어 노드를 선택하고 객체 메뉴에서 삭제를 선택할 수도 있습니다.
3. 모든 접미어 항목이 디렉토리에서 제거될 것이라고 알려주는 확인 대화 상자가 표시됩니다.

부모 접미어의 경우 모든 하위 접미어를 재귀적으로 삭제할 수 있습니다. 전체 분기를 제거하려면 "이 접미어와 해당 하위 접미어를 모두 삭제합니다"를 선택합니다. 디렉토리에서 특정 접미어만 제거하고 해당 하위 접미어를 그대로 유지하려면 "이 접미어만 삭제합니다"를 선택합니다.

4. 확인을 눌러 접미어를 삭제합니다.
콘솔의 작업 완료 상황을 단계별로 보여주는 진행률 대화 상자가 표시됩니다.

명령줄에서 접미어 삭제

명령줄에서 접미어를 삭제하려면 `ldapdelete` 명령을 사용하여 해당 구성 항목을 디렉토리에서 제거합니다.

하위 접미어를 포함한 전체 분기를 삭제하려면 삭제된 부모의 하위 접미어를 찾아서 각 하위 접미어 및 해당 하위 접미어에 대해 같은 절차를 반복해야 합니다.

1. 아래 명령을 실행하여 접미어 구성 항목을 제거합니다.

```
ldapdelete -h host -p port -D "cn=Directory Manager" -w password \
-v 'cn=suffixDN',cn=mapping tree,cn=config'
```

이 명령은 *suffixDN*에 지정된 기준 항목부터 시작하여 접미어를 서버에서 제거합니다. 이제 디렉토리에서 이 접미어를 보거나 액세스할 수 없습니다. `-v` 옵션은 세부 정보 표시 모드를 지정합니다. 즉, 삭제 작업에 대한 추가 정보가 표시됩니다.

2. `cn=databaseName,cn=ldbm database,cn=plugins,cn=config`에 있는 해당 데이터베이스 구성 항목과 하위 항목을 모두 제거합니다. 이러한 재귀적 삭제를 수행하기 위해서는 `ldapsubtdel` 명령을 Directory Server Resource Kit(DSRK)에서 다운로드하여 사용해야 합니다. 이 소프트웨어는

<http://www.sun.com/software/download/>에서 다운로드할 수 있습니다.

```
% ./ldapsubtdel -h hostname -p port -D "cn=Directory Manager" -w - -b
"ou=test,dc=example,dc=com" -r -v
```

```
Enter bind password: password
```

```
Processing subtree ou=test,dc=example,dc=com
```

```
Deleting entry uid=test0,ou=test,dc=example,dc=com
```

```

Deleting entry uid=test1,ou=test,dc=example,dc=com
Deleting entry uid=test2,ou=test,dc=example,dc=com
Deleting entry uid=test3,ou=test,dc=example,dc=com
[...]
Deleting entry ou=test,dc=example,dc=com
Successfully deleted subtree ou=test,dc=example,dc=com
%

```

이 명령은 데이터베이스와 관련된 모든 색인 구성 항목을 출력하며, 이렇게 출력된 항목은 모두 제거해야 합니다. 데이터베이스 구성이 완전히 삭제되면 서버는 이 접미어와 관련된 모든 데이터베이스 파일과 디렉토리를 제거합니다.

연결 접미어 작성

루트 접미어와 하위 접미어는 다른 서버에 연결할 수 있으며 콘솔이나 명령줄에서 두 절차를 모두 수행할 수 있습니다.

하지만 연결 접미어를 작성하기 전에 먼저 원격 서버에서 프록시 아이디를 작성해야 합니다. 로컬 서버는 연결 접미어를 통해 작업을 전달할 때 이 프록시 아이디를 사용하여 원격 서버에 바인드합니다.

동일한 매개 변수를 사용하여 많은 연결 접미어를 구성하는 경우 새로운 연결 접미어의 연결 매개 변수에도 기본값을 설정해야 합니다. [168페이지의 "연결 정책 구성"](#)에 설명된 것처럼, 연결 접미어를 작성하기 전이나 작성한 후에 LDAP 컨트롤과 서버 구성 요소에 연결 정책을 설정할 수도 있습니다.

프록시 아이디 작성

프록시 아이디는 로컬 서버에서 연결 작업을 바인드하고 전달하기 위해 사용하는 원격 서버의 사용자입니다. 보안상, 디렉토리 관리자나 관리 사용자(admin)를 프록시에 사용해서는 안 됩니다.

지정된 서버의 연결 작업에만 사용할 새 아이디를 작성하십시오. [160페이지의 "콘솔에서 연결 접미어 작성"](#) 또는 [171페이지의 "콘솔에서 연결 정책 수정"](#)에 정의된 모든 페일오버 서버 및 연결할 모든 서버에서 프록시 아이디를 작성합니다. 프록시 아이디는 연결 접미어에 완전한 액세스 권한을 가져야 합니다.

콘솔에서 프록시 아이디 작성

아래 절차는 Directory Server 콘솔이 연결 접미어의 대상인 원격 서버에 연결되어 있는 경우에만 적용됩니다.

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 디렉토리 트리를 확장합니다.
2. `cn=config` 항목을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 새로 만들기 > 사용자를 선택합니다. 또는 `cn=config` 항목을 선택하고 객체 메뉴에서 새로 만들기 > 사용자 항목을 선택합니다.
3. 아래 예와 같이 새 사용자 만들기 대화 상자의 필드에 프록시 아이디를 설명하는 값을 입력합니다.

이름:	<code>proxy</code>
성:	<code>host1</code>
일반 이름:	<code>host1 chaining proxy</code>
사용자 아이디:	<code>host1_proxy</code>
비밀 번호:	<code>password</code>
암호 확인:	<code>password</code>

여기서 `host1`은 연결 접미어가 있는 서버 이름입니다. 이 서버에 연결된 접미어가 있는 각각의 서버에서 다른 프록시 아이디를 사용해야 합니다.

4. 확인을 눌러 새 프록시 아이디를 저장합니다.

명령줄에서 프록시 아이디 작성

아래 절차에서는 `host1`과 `host2`를 사용하여 각각 연결 접미어가 있는 로컬 서버와 연결 접미어의 대상인 원격 서버를 나타냅니다.

1. 아래 명령을 실행하여 `host2`에서 프록시 아이디를 작성합니다.

```
ldapmodify -a -h host2 -p port2 -D "cn=Directory Manager" -w password2
dn: uid=host1_proxy,cn=config
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetorgperson
uid: host1_proxy
cn: host1 chaining proxy
sn: host1
userpassword: password
description: proxy entry to be used for chaining from host1
^D
```

주의 일반 텍스트 비밀번호 전송을 방지하려면 암호화된 포트를 통해 `ldapmodify` 명령을 실행해야 합니다.

기본 연결 매개 변수 설정

연결 매개 변수는 서버가 어떻게 연결 서버에 연결하여 해당 연결 접미어에 대한 작업을 처리하는지 결정하며, 각 연결 접미어에 구성됩니다. **Directory Server**는 연결 접미어를 작성할 때 사용되는 기본값을 제공합니다. 이 기본값을 편집하여 새로운 모든 연결 접미어의 연결 매개 변수를 지정할 수 있습니다.

기본 매개 변수를 수정한 후에 작성된 새 연결 접미어에는 변경된 값이 포함됩니다. 하지만 접미어가 작성된 후에는 [168페이지의 "연결 접미어 관리"](#)에 설명된 방식으로만 매개 변수를 수정할 수 있습니다.

아래에는 연결 매개 변수의 속성과 기본값이 설명되어 있습니다. 허용되는 값에 대한 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.

클라이언트 반환 매개 변수

- `nsReferralOnScopedSearch - on`(기본값)으로 설정하면 연결 접미어 내에서만 검색하는 클라이언트 검색 시 원격 서버에 대한 참조를 받게 되므로 검색 결과가 두 번 전송되는 것을 방지할 수 있습니다. `off`로 지정할 경우 크기 및 시간 제한 매개 변수를 설정하여 연결 접미어 검색 시간을 제한해야 합니다.

- `nsslapd-sizelimit` - 이 매개 변수는 연결 검색 작업에 응답하여 반환할 항목 수를 지정합니다. 기본 크기 제한은 2000개입니다. 연결 접미어에 대한 광범위한 검색을 제한하려면 이 매개 변수 값을 낮게 설정하십시오. 항상 원격 서버의 크기 설정에 따라 작업이 제한됩니다.
- `nsslapd-timelimit` - 이 매개 변수는 연결 작업 시간을 제어합니다. 기본 시간 제한은 3600초(1시간)입니다. 연결 접미어에 대한 작업 시간을 제한하려면 이 매개 변수 값을 낮게 설정하십시오. 항상 원격 서버의 시간 설정에 따라 작업이 제한됩니다.

계단식 연결 매개 변수

- `nsCheckLocalACI` - 단일 수준 연결에서는 로컬 서버가 아닌 원격 서버가 연결 접미어에 바인드된 사용자의 액세스 권한을 확인하므로 기본값은 `off`입니다. 그러나 계단식 연결의 중간 서버는 연결 작업을 전달하는 서버에서 사용된 프록시 DN의 액세스 권한을 확인 및 제한하기 위해 이 매개 변수를 `on`으로 설정해야 합니다.
- `nsHopLimit` - 루프 감지 기능은 이 매개 변수를 사용하여 허용되는 최대 홉 수를 정의합니다. 연결 작업이 정의된 최대 홉 수에 도달할 경우 계단식 토폴로지에 잘못된 루프가 있다는 가정 하에 이 연결 작업은 전달되지 않고 중단됩니다.

연결 관리 매개 변수

- `nsOperationConnectionsLimit` - 연결 접미어와 원격 서버 사이에 구성할 수 있는 동시 LDAP 작업 연결의 최대 개수입니다. 기본값은 10개입니다.
- `nsBindConnectionsLimit` - 연결 접미어와 원격 서버 간에 구성할 수 있는 최대 동시 바인드 연결 수입니다. 기본값은 3개입니다.
- `nsConcurrentBindLimit` - LDAP 연결 당 동시에 허용되는 최대 바인드 작업 수입니다. 기본값은 연결 당 아직 해결되지 않은 바인드 작업 10개입니다.
- `nsBindRetryLimit` - 오류 시 연결 접미어가 원격 서버로 재바인드를 시도하는 횟수입니다. 값을 0으로 설정하면 연결 접미어는 바인드를 한 번만 시도합니다. 기본값은 3회입니다.

- nsConcurrentOperationsLimit - LDAP 연결 당 동시에 허용되는 최대 작업 수입니다. 기본값은 연결 당 10개입니다.
- nsBindTimeout - 연결 접미어에 대한 바인드 시도가 시간 초과될 때까지의 시간(초)입니다. 기본값은 15초입니다.
- nsAbandonedSearchCheckInterval - 서버에서 작업 중단 여부를 확인할 때까지의 시간(초)입니다. 기본값은 2초입니다.
- nsConnectionLife - 연결 접미어와 원격 서버 간의 연결을 다시 사용할 수 있도록 열어 두는 시간입니다. 연결을 열어 두면 속도는 더 빠르지만 많은 자원이 사용됩니다. 예를 들어, 전화 접속 연결을 사용하는 경우 연결 시간을 제한하는 것이 좋습니다. 기본값은 0으로, 연결 시간이 제한되지 않습니다.

오류 감지 매개 변수

- nsmaxresponedelay - 원격 서버가 LDAP의 연결 작업 요청에 응답할 때까지의 최대 시간입니다. 시간은 초 단위로 설정됩니다. 로컬 서버는 이 지연 시간 후에 연결을 테스트합니다. 기본 지연 시간은 60초입니다.
- nsmaxtestresponedelay - 원격 서버의 응답 여부를 확인하기 위한 테스트 기간입니다. 테스트는 존재하지 않는 항목을 검색하는 단순한 요청을 이용합니다. 시간은 초 단위로 설정됩니다. 테스트 지연 동안 응답이 없으면 연결 접미어는 원격 서버가 다운되었다고 가정합니다. 기본 테스트 응답 지연 시간은 15초입니다.

이 연결 접미어에 한 개의 원격 서버만 정의한 경우에는 오버로드를 방지하기 위해 원격 서버에 대한 모든 연결 작업이 30초 동안 차단됩니다. 페일오버 서버를 정의한 경우 연결 작업은 정의된 그 다음 대체 서버에서 시작됩니다.

콘솔에서 기본 연결 매개 변수 설정

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 디렉토리 트리를 확장하고 `cn=default instance config,cn=chaining database,cn=plugins,cn=config` 항목을 선택합니다.
2. 항목을 두 번 누르거나 객체 > 일반 편집기로 편집 메뉴 항목을 선택합니다. 위의 목록에서 원하는 속성 값을 수정합니다.
3. 일반 편집기 대화 상자에서 저장을 누르면 변경 사항이 즉시 적용됩니다.

명령줄에서 기본 연결 매개 변수 설정

1. ldapmodify 명령을 사용하여 cn=default instance config,cn=chaining database,cn=plugins,cn=config 항목을 편집합니다. 이 항목의 모든 속성은 새 연결 접미어 매개 변수의 기본값이 됩니다.

예를 들어 아래 명령은 새 연결 접미어의 기본 크기 제한을 5000개 항목으로 늘리고 기본 시간 제한을 10분으로 단축합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=default instance config,cn=chaining database,
   cn=plugins,cn=config
changetype: modify
replace: nsslapd-sizelimit
nsslapd-sizelimit: 5000
-
replace: nsslapd-timelimit
nsslapd-timelimit: 600
^D
```

항목에 대한 변경 사항이 즉시 적용됩니다.

콘솔에서 연결 접미어 작성

아래 절차에 따라 연결 루트 접미어와 연결 하위 접미어를 모두 작성할 수 있습니다.

1. Directory Server 콘솔의 구성 탭을 선택합니다.
 - 연결 루트 접미어의 경우 데이터 노드를 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 새 연결 접미어를 선택합니다. 또는, 데이터 노드를 선택하고 객체 메뉴에서 새 연결 접미어를 선택할 수도 있습니다.
 - 연결 하위 접미어의 경우 데이터 노드와 접미어 노드를 확장하여 부모 접미어를 표시합니다. 부모 접미어 노드를 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 새 연결 하위 접미어를 선택합니다. 또는 부모 접미어 노드를 선택하고 객체 메뉴에서 새 연결 하위 접미어를 선택할 수도 있습니다.

"새 연결 (하위) 접미어" 대화 상자가 표시됩니다.
2. 연결할 원격 서버 항목의 DN을 입력합니다. 원격 항목이 원격 접미어의 기본 항목일 필요는 없습니다.

- 루트 접미어의 경우 접미어 DN 필드에 원격 항목의 전체 DN을 입력합니다. 원격 디렉토리 트리에 있는 어떤 항목의 DN이든 입력할 수 있습니다. 이 항목이 연결 루트 접미어의 기본 항목이 되고, 연결 접미어를 통해 모든 하위 항목을 사용할 수 있습니다.
 - 하위 접미어의 경우 연결할 항목의 하위 접미어 RDN을 입력합니다. 이 항목이 연결 하위 접미어의 기본 항목이 됩니다. 텍스트 필드 아래에 표시되는 전체 하위 접미어 이름은 원격 서버 항목이어야 합니다.
3. 필요한 경우, 접미어 데이터가 저장된 원격 서버의 호스트 이름을 도메인과 함께 입력합니다.
 4. 원격 서버에 액세스하기 위한 포트 번호를 입력하고, 보안 포트인 경우 확인란을 선택합니다. 보안 포트를 사용하면 연결 작업이 SSL상에서 암호화됩니다. 자세한 내용은 167페이지의 "SSL을 사용한 연결"을 참조하십시오.
대화 상자 아래쪽에는 원격 서버의 전체 URL이 표시됩니다.
 5. 원격 서버에 지정된 프록시 아이디의 바인드 DN과 비밀번호를 입력합니다. 로컬 서버는 원격 서버에 있는 접미어 내용에 액세스할 때 이 DN을 프록시로 사용합니다. 예를 들어, 155페이지의 "프록시 아이디 작성"에서 정의한 `uid=host1_proxy, cn=config` DN을 사용합니다.
원격 서버의 디렉토리 관리자 DN은 사용할 수 없습니다. 연결 접미어를 통해 수행된 작업은 `creatorsName` 속성과 `modifiersName` 속성에 이 프록시 아이디를 사용합니다. 프록시 DN은 생략할 수 있으며, 이 경우에는 로컬 서버가 원격 서버에 액세스할 때 익명으로 바인드합니다.
 6. 확인을 눌러 연결 접미어를 작성합니다. 새 접미어가 연결 아이콘과 함께 구성 트리에 표시됩니다.
 7. 새 연결 접미어를 눌러 선택하고 오른쪽 패널에서 원격 서버 탭을 선택합니다.
 8. 선택 사항으로, 이 연결 접미어에 대한 페일오버 서버를 하나 이상 정의할 수도 있습니다. 원격 서버에 연결할 수 없는 경우 서버는 응답하는 서버가 나타날 때까지 페일오버 서버를 정의된 순서대로 하나씩 시도합니다. 페일오버 서버에는 원격 서버의 연결 접미어와 똑같은 접미어가 있어야 하며 프록시에 지정된 바인드 DN도 같아야 합니다.

페일오버 서버를 정의하려면 원격 서버 URL 필드에 다른 호스트 이름 및 포트 번호 쌍을 공백으로 구분하여 입력합니다. 필드 형식은 다음과 같습니다.

```
ldap[s]://hostname[:port][ hostname[:port]].../
```

9. 원격 서버 탭의 아래쪽 텍스트 상자에 연결을 통한 프록시 작업을 허용하는 데 필요한 ACI가 표시됩니다. *suffixDN*에 해당하는 원격 서버 항목에 이 ACI를 추가해야 합니다. 페일오버 서버를 정의한 경우 페일오버 서버에도 ACI를 추가하십시오. ACI 복사 버튼을 사용하여 ACI 텍스트를 시스템 클립보드에 복사한 다음 붙여넣습니다.

원격 서버의 기본 항목에 ACI를 추가하면 로컬 서버의 디렉토리 트리에 연결 접미어가 표시됩니다.

주의 연결을 통해 공개된 원격 서버에 대한 액세스를 제한하기 위해 동일한 항목에 다른 ACI를 정의해야 할 수도 있습니다. [166페이지의 "연결 접미어를 통한 액세스 제어"](#)를 참조하십시오.

10. 서버 구성 요소에 대한 연결 정책을 구성한 경우 이 구성 요소가 원격 서버에 액세스할 수 있도록 허용하는 ACI도 추가해야 합니다. 예를 들어 참조 무결성 플러그 인의 연결을 허용할 경우 [단계 2](#)에서 지정한 DN의 기본 항목에 아래 ACI를 추가해야 합니다.

```
aci: (targetattr "*"
(target="ldap:///suffixDN")
(version 3.0; acl "RefInt Access for chaining"; allow
(read,write,search,compare) userdn = "ldap:///cn=referential
integrity postoperation,cn=plugins,cn=config";)
```

명령줄에서 연결 접미어 작성

ldapmodify 명령줄 유틸리티를 사용하여 디렉토리에 연결 접미어를 작성할 수도 있습니다. 연결 루트 접미어와 연결 하위 접미어는 서버에서 똑같이 내부적으로 관리되기 때문에 명령줄에서 작성하는 절차도 거의 흡사합니다.

1. 연결 루트 접미어의 경우 아래 명령을 실행하여 cn=mapping tree,cn=config에 연결 접미어 항목을 작성합니다.

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=suffixDN,cn=mapping tree,cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsMappingTree
```

```

cn: suffixDN
nsslapd-state: backend
nsslapd-backend: databaseName
^D

```

연결 하위 접미어의 경우 아래 속성만 추가하여 같은 명령을 사용합니다.

```
nsslapd-parent-suffix: parentSuffixDN
```

연결 하위 접미어의 *suffixDN*에는 하위 접미어의 RDN과 부모 접미어의 DN이 포함됩니다(예: *l=Europe, dc=example, dc=com*). *suffixDN*은 원격 서버를 통해 사용할 수 있는 항목의 DN이어야 하지만 원격 접미어의 기본 항목일 필요는 없습니다.

접미어 이름은 DN 형식을 사용하지만 하나의 문자열로 처리됩니다. 따라서 모든 공백은 접미어 이름의 일부로 중요한 의미를 가집니다. 서버에서 원격 항목에 액세스하려면 원격 접미어에 사용된 공백까지 정확하게 *suffixDN* 문자열에 입력해야 합니다.

*databaseName*은 연결 플러그인 구성 요소에서 이 연결 접미어를 식별하기 위해 사용됩니다. 모든 접미어의 *databaseName*은 고유해야 하며, 관례상 *suffixDN*의 이름 지정 구성 요소 중에서 첫 구성 요소의 값입니다. 로컬 접미어와 달리 연결 접미어는 로컬 서버에 데이터베이스 파일을 저장하지 않습니다.

하위 접미어의 경우 *parentSuffixDN*은 부모 접미어의 DN과 정확하게 일치합니다. 부모는 로컬 접미어이거나 연결 접미어입니다.

2. 아래 명령을 실행하여 연결 구성 항목을 작성합니다.

```

ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=databaseName,cn=chaining database,cn=plugins,cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsBackendInstance
cn: databaseName
nsslapd-suffix: suffixDN
nsfarmserverurl: LDAPURL
nsmultiplexorbinddn: proxyDN
nsmultiplexorcredentials: ProxyPassword
^D

```

여기서 *databaseName*과 *suffixDN*은 이전 단계에서 사용한 값이어야 합니다. *LDAPURL*은 원격 서버의 URL이지만 접미어 정보는 포함되지 않습니다. 파일오버서버가 아래 형식으로 URL에 열거될 수도 있습니다.

```
ldap[s]://hostname[:port][ hostname[:port] ].../
```

LDAP URL에 열거된 모든 원격 서버에는 *suffixDN*이 있어야 합니다. 보안 포트 지정에 대한 자세한 내용은 [167페이지의 "SSL을 사용한 연결"](#)을 참조하십시오.

*proxyDN*은 원격 서버에 지정된 프록시 아이디의 DN입니다. 로컬 서버는 원격 서버에 있는 접미어 내용에 액세스할 때 이 DN을 프록시로 사용합니다. 연결 접미어를 통해 수행된 작업은 *creatorsName* 속성과 *modifiersName* 속성에 이 프록시 아이디를 사용합니다. 프록시 DN을 지정하지 않으면 로컬 서버는 원격 서버에 액세스할 때 익명으로 바인드합니다.

*ProxyPassword*는 프록시 DN의 암호화되지 않은 비밀번호 값으로, 비밀번호는 구성 파일에 저장될 때 암호화됩니다. 예를 들면 다음과 같습니다.

```
nsmultiplexorbinddn: uid=host1_proxy,cn=config
nsmultiplexorcredentials: secret
```

주의 일반 텍스트 비밀번호 전송을 방지하려면 보안 포트를 통해 `ldapmodify` 명령을 실행해야 합니다.

`cn=default instance config,cn=chaining database,cn=plugins,cn=config`의 기본값으로 설정된 모든 연결 매개 변수가 자동으로 새 항목에 포함됩니다. 연결 구성 항목을 작성할 때 다른 속성 값을 설정하여 이 값을 무시할 수 있습니다. 값을 정의할 수 있는 속성 목록은 [157페이지의 "기본 연결 매개 변수 설정"](#)을 참조하십시오.

- 아래 명령을 실행하여 원격 항목에 ACI를 작성합니다. 이 ACI는 연결을 통한 프록시 작업을 허용하는 데 필요합니다. ACI에 대한 자세한 내용은 [6장, "액세스 제어 관리"](#)를 참조하십시오.

```
ldapmodify -h host2 -p port2 -D "cn=Directory Manager" -w password2
dn: suffixDN
changetype: modify
add: aci
aci: (targetattr=*)(target = "ldap:///suffixDN")(version 3.0;acl
  "Allows use of admin for chaining"; allow (proxy)
  (userdn="ldap:///proxyDN");)
^D
```

주의 이 서버를 통해 공개된 원격 서버에 대한 액세스를 제한하기 위해 동일한 항목에 다른 ACI를 정의해야 할 수도 있습니다. [166페이지의 "연결 접미어를 통한 액세스 제어"](#)를 참조하십시오.

4. 서버 구성 요소에 대한 연결 정책을 구성한 경우 이 구성 요소가 원격 서버에 액세스할 수 있도록 허용하는 ACI도 추가해야 합니다. 예를 들어 참조 무결성 플러그인의 연결을 허용할 경우 *suffixDN*에 해당하는 기본 항목에 아래 ACI를 추가해야 합니다.

```
aci: (targetattr "*" )
(target="ldap:///suffixDN")
(version 3.0; acl "RefInt Access for chaining"; allow
(read,write,search,compare) userdn = "ldap:///cn=referential
integrity postoperation,cn=plugins,cn=config");)
```

아래 명령은 연결 하위 접미어를 작성하는 예를 보여줍니다. *suffixDN*의 쉼표는 DN의 이름 지정 속성에 표시될 때만 역슬래시(\)로 이스케이프해야 합니다.

```
ldapmodify -a -h host1 -p port1 -D "cn=Directory Manager" -w password1
dn: cn=l=Europe\,dc=example\,dc=com,cn=mapping tree,cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsMappingTree
cn: l=Europe,dc=example,dc=com
nsslapd-state: backend
nsslapd-backend: Europe
nsslapd-parent-suffix: dc=example,dc=com
```

```
dn: cn=Europe,cn=chaining database,cn=plugins,cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsBackendInstance
cn: Europe
nsslapd-suffix: l=Europe,dc=example,dc=com
nsfarmserverurl: ldap://host2:port2/
nsmultiplexorbinddn: uid=host1_proxy,cn=config
nsmultiplexorcredentials: proxyPassword
^D
```

```
ldapmodify -h host2 -p port2 -D "cn=Directory Manager" -w password2
dn: l=Europe,dc=example,dc=com
changetype: modify
changetype: modify
aci: (targetattr=*)(target =
"ldap:///l=Europe,dc=example,dc=com")(version 3.0;acl
"Allows use of admin for chaining"; allow (proxy)
(userdn="ldap:///uid=host1_proxy,cn=config");)
^D
```

연결 접미어를 통한 액세스 제어

인증된 사용자가 연결 접미어에 액세스하면 서버는 이 사용자의 아이디를 원격 서버로 보냅니다. 액세스 제어는 항상 원격 서버에서 평가됩니다. 원격 서버에서 평가되는 모든 LDAP 작업은 프록시 인증 제어를 통해 전달된 클라이언트 응용 프로그램의 원래 아이디를 사용합니다. 사용자에게 원격 서버의 하위 트리에 대한 올바른 액세스 제어가 있을 경우에만 원격 서버 작업이 제대로 수행됩니다. 따라서 다음과 같은 몇 가지 제한을 고려하여 원격 서버에 일반적인 액세스 제어를 추가할 필요가 있습니다.

- 모든 유형의 액세스 제어를 사용할 수는 없습니다.
예를 들어, 역할 기반이나 필터 기반의 ACI는 사용자 항목에 액세스해야 하지만 연결 접미어를 통해 데이터에 액세스하기 때문에 프록시 제어에 있는 데이터만 확인할 수 있습니다. 사용자 항목이 사용자 데이터와 같은 접미어에 저장되도록 디렉토리를 구성하십시오.
- 클라이언트의 원래 도메인이 연결 중에 손실되기 때문에 클라이언트의 IP 주소나 DNS 도메인에 기반을 둔 모든 액세스 제어는 제대로 작동하지 않을 수도 있습니다.
원격 서버는 클라이언트 응용 프로그램이 연결 접미어와 같은 IP 주소 및 DNS 도메인에 있다고 가정합니다.

연결 접미어에 사용할 ACI를 작성하는 경우 다음과 같은 제한이 적용됩니다.

- ACI는 사용되는 그룹과 같은 서버에 있어야 합니다. 그룹이 동적이면 이 그룹의 모든 사용자가 ACI 및 그룹과 같은 위치에 있어야 합니다. 정적 그룹은 원격 사용자를 참조할 수 있습니다.
- ACI는 사용되는 역할 정의 및 이 역할이 할당될 모든 사용자와 같은 서버에 있어야 합니다.
- 원격 사용자의 경우 사용자 항목 값을 참조하는 ACI(예: userattr 주제 규칙)가 작동합니다.

액세스 제어는 항상 원격 서버에서 평가되지만 연결 접미어가 있는 서버와 원격 서버에서 모두 평가하도록 설정할 수도 있습니다. 이 경우 다음과 같은 여러 가지 제한이 따릅니다.

- 액세스 제어를 평가하는 동안 사용자 항목 내용을 사용하지 못할 수도 있습니다(예: 연결 접미어가 있는 서버에서 액세스 제어가 평가되고 원격 서버에 항목이 있는 경우). 성능상의 이유로 클라이언트는 원격 조회 및 액세스 제어 평가를 수행할 수 없습니다.

- 연결 접미어가 클라이언트 응용 프로그램에서 수정하는 항목에 액세스하지 못할 수도 있습니다.
수정 작업을 할 때 연결 접미어에는 원격 서버에 저장된 전체 항목에 대한 액세스 권한이 없습니다. 삭제 작업의 경우 연결 접미어에서 항목의 DN만 알기 때문에 액세스 제어에서 특정 속성을 지정하면 연결 접미어를 통한 삭제 작업이 제대로 수행되지 않습니다.

기본적으로 연결 접미어가 있는 서버에서 설정된 액세스 제어는 평가되지 않습니다. 이 기본값을 무시하려면 `cn=databaseName, cn=chaining database, cn=plugins, cn=config` 항목의 `nsCheckLocalACL` 속성을 사용합니다. 그러나 계단식 연결을 사용하는 경우가 아니면 연결 접미어가 있는 서버에서 액세스 제어를 평가하는 것은 바람직하지 않습니다. 자세한 내용은 [182페이지의 "계단식 연결 구성"](#)을 참조하십시오.

SSL을 사용한 연결

연결 접미어에 대한 작업을 수행할 때 SSL을 사용하여 원격 서버와 통신하도록 서버를 구성할 수 있습니다. SSL을 연결에 사용하려면 다음과 같은 단계를 수행해야 합니다.

1. 원격 서버에서 SSL을 활성화합니다.
2. 연결 접미어가 있는 서버에서 SSL을 활성화합니다.
SSL 활성화에 대한 자세한 내용은 [11장, "인증 및 암호화 관리"](#)를 참조하십시오.
3. 연결 접미어를 작성 또는 수정하는 동안 SSL과 원격 서버의 보안 포트를 지정합니다.

콘솔을 사용하는 경우 연결 접미어를 작성 또는 구성하는 동안 보안 포트 확인란을 선택합니다. [160페이지의 "콘솔에서 연결 접미어 작성"](#) 또는 [171페이지의 "콘솔에서 연결 정책 수정"](#)을 참조하십시오.

명령줄 절차를 사용하는 경우 LDAPS URL과 원격 서버의 보안 포트를 지정합니다. 예를 들어 `ldaps://example.com:636/`과 같이 입력합니다. [162페이지의 "명령줄에서 연결 접미어 작성"](#) 또는 [172페이지의 "명령줄에서 연결 정책 수정"](#)을 참조하십시오.

연결 접미어와 원격 서버에서 SSL을 사용하여 통신하도록 구성한다고 해서 작업 요청을 하는 클라이언트 응용 프로그램도 SSL을 사용해야 하는 것은 아닙니다. 클라이언트는 LDAP 또는 DSML 프로토콜의 어떤 포트라도 사용할 수 있습니다.

연결 접미어 관리

이 절에서는 기존의 연결 접미어를 업데이트 및 삭제하고 연결 기법을 제어하는 방법에 대해 설명합니다.

연결 정책 구성

서버의 연결 정책은 연결 서버에 전파할 LDAP 컨트롤 및 연결 접미어에 대한 액세스를 허용할 서버 구성 요소를 결정합니다. 이러한 설정과 각 설정이 연결 접미어에 대한 작업에 미치는 영향을 명확히 이해해야 합니다. 연결 정책은 서버에 있는 모든 연결 접미어에 적용됩니다.

기본 설정은 정상적인 작업이 투명하게 완료될 수 있도록 구성되었습니다. 그러나 LDAP 컨트롤이 필요한 작업이거나 참조 무결성 플러그 인과 같은 서버 구성 요소를 사용하는 경우 사용자 요구에 맞게 연결 정책을 구성해야 합니다.

연결 접미어를 작성하기 전에 먼저 연결 정책을 구성하여 연결 접미어가 활성화되면 즉시 정책이 적용될 수 있도록 하는 것이 가장 좋지만 나중에 정책을 수정할 수도 있습니다.

LDAP 컨트롤의 연결 정책

LDAP 컨트롤은 클라이언트에서 특정 방식으로 작업이나 결과를 수정하라고 요청할 때 이 요청의 일부로 전송됩니다. 서버 연결 정책은 작업과 더불어 서버에서 연결 접미어로 전달할 컨트롤을 결정합니다. 기본적으로 다음과 같은 컨트롤이 연결 접미어의 원격 서버로 전달됩니다.

표 3-1 기본적으로 연결이 허용되는 LDAP 컨트롤

컨트롤의 OID	컨트롤 이름 및 설명
1.2.840.113556.1.4.473	서버측 정렬 - 검색과 연결되어 결과로 표시된 항목을 해당 속성 값에 따라 정렬합니다. ¹
1.3.6.1.4.1.1466.29539.12	연결 루프 감지 - 서버에서 다른 서버와 연결하는 횟수를 추적합니다. 이 횟수가 지정된 값에 도달하면 작업이 중단되고 클라이언트 응용 프로그램에 이를 알려줍니다. 자세한 내용은 184페이지의 "계단식 연결을 위한 LDAP 컨트롤 전송" 을 참조하십시오.
2.16.840.1.113730.3.4.2	스마트 참조의 관리된 DSA - 참조를 따르지 않고 스마트 참조를 항목으로 반환하므로 스마트 참조 자체를 변경하거나 삭제할 수 있습니다.

표 3-1 기본적으로 연결이 허용되는 LDAP 컨트롤(계속)

컨트롤의 OID	컨트롤 이름 및 설명
2.16.840.1.113730.3.4.9	가상 목록 보기(VLV) - 모든 결과 항목을 동시에 반환하지 않고 일부 검색 결과를 제공합니다. ¹

1. 서버측 정렬과 VLV 컨트롤은 검색 범위가 단일 접미어인 경우에만 연결에서 지원됩니다. 클라이언트 응용 프로그램이 여러 접미어에 요청하면 연결 접미어에서 VLV 컨트롤을 지원할 수 없습니다.

아래 표에는 연결 정책을 구성하여 연결에 허용할 수 있는 기타 LDAP 컨트롤이 열거되어 있습니다.

표 3-2 연결 가능한 LDAP 컨트롤

컨트롤의 OID	컨트롤 이름 및 설명
1.3.6.1.4.1.42.2.27.9.5.2	유효 권한 요청 받기 - 항목과 속성에 관한 액세스 권한 및 ACI 정보를 결과로 반환하도록 서버에 요청합니다.
2.16.840.1.113730.3.4.12	프록시 인증(이전 사양) - 클라이언트에서 요청 기간 동안 다른 항목의 아이디를 사용할 수 있게 합니다. ¹
2.16.840.1.113730.3.4.14	특정 데이터베이스에 대한 검색 - 검색 작업과 함께 사용되어 컨트롤에 지정된 데이터베이스에 대해 검색하도록 지정합니다.
2.16.840.1.113730.3.4.16	인증 아이디 요청 제어 - 클라이언트가 연결 바인드 중에 설정된 바인드 DN 또는 userAuthID를 요청할 수 있게 합니다. 예를 들어, SASL External 및 SSL을 사용할 때 클라이언트가 인증서 확보에 필요한 항목의 DN을 알 수 있도록 허용합니다.
2.16.840.1.113730.3.4.17	실제 속성만 요청 - 서버에서 가상 속성을 확인할 필요 없이 실제로 반환된 항목에 있는 속성만 반환하도록 지정합니다.
2.16.840.1.113730.3.4.18	프록시 인증(새 사양) - 클라이언트에서 요청 기간 동안 다른 아이디를 사용할 수 있게 합니다. ¹
2.16.840.1.113730.3.4.19	가상 속성만 요청 - 서버에서 서비스 기능의 역할 및 클래스에 의해 생성된 속성만 반환하도록 지정합니다.

1. 응용 프로그램은 두 컨트롤 중 하나를 프록시 인증에 사용할 수 있습니다. 두 OID에 대한 연결 정책은 같아야 합니다. 자세한 내용은 184페이지의 "계단식 연결을 위한 LDAP 컨트롤 전송"을 참조하십시오.

서버 구성 요소의 연결 정책

구성 요소는 내부 작업을 사용하는 서버의 모든 기능 또는 기능 단위입니다. 예를 들어 플러그인도 구성 요소로 간주됩니다. 대부분의 구성 요소는 작업을 수행하기 위해 디렉토리에 저장된 구성 데이터 또는 사용자 데이터 중 하나인 디렉토리 내용에 액세스해야 합니다.

기본적으로 서버 구성 요소는 연결이 허용되지 않습니다. 서버 구성 요소에서 연결 접미어에 액세스할 수 있게 하려면 명시적으로 연결을 허용해야 합니다. 연결 데이터에 액세스할 수 있는 구성 요소는 아래에 해당 DN으로 열거되어 있습니다.

160페이지의 "콘솔에서 연결 접미어 작성"에 설명된 것처럼 원격 서버에서 ACI에 특권 권한을 부여하여 연결을 허용해야 합니다. 서버 구성 요소를 연결하는 경우 이 ACI에서 검색, 읽기 및 비교를 허용해야만 서버 구성 요소에서 이러한 작업을 수행할 수 있습니다. 목록에 설명된 것처럼 원격 서버에 대한 쓰기 권한이 필요한 구성 요소도 있습니다.

- `cn=ACL Plugin,cn=plugins,cn=config-ACL` 플러그인은 액세스 제어 기능을 구현합니다. 로컬 및 원격 ACI 속성을 혼합하는 것은 안전하지 않으므로 ACI 속성을 검색 및 업데이트하는 작업은 연결되지 않습니다. 하지만 사용자 항목에 액세스하기 위한 요청은 연결할 수 있습니다. ACI 및 연결 제한에 대한 자세한 내용은 240페이지의 "ACI 제한"을 참조하십시오.
- `cn=old plugin,cn=plugins,cn=config-` 이 플러그인은 모든 Directory Server 4.x 플러그인과 각각의 연결 허용 여부를 나타냅니다. 4.x 플러그인은 모두 같은 연결 정책을 사용합니다. 4.x 플러그인에서 수행한 작업에 따라 원격 서버의 ACI를 설정해야 할 수도 있습니다.
- `cn=resource limits,cn=components,cn=config-` 이 구성 요소는 사용자 바인드 DN에 따라 자원 사용 제한을 설정합니다. 이 구성 요소의 연결이 허용되면 연결 접미어에 아이디가 저장되어 있는 사용자에게 대해 자원 제한을 실행할 수 있습니다.
- `cn=certificate-based authentication,cn=components,cn=config-` 이 구성 요소는 SASL 외부 바인드 방법과 함께 사용되며 원격 서버에서 사용자 인증서를 검색합니다.

주의 연결 접미어에서 인증서 기반의 인증을 허용하면 보안 허점이 생길 수 있습니다. 다른 접미어가 신뢰할 수 없는 원격 서버에 연결되어 있으면 신뢰할 수 없는 서버의 인증서가 인증에 사용될 수 있습니다.

- `cn=referential integrity postoperation,cn=plugins,cn=config` - 이 플러그 인을 사용하면 항목이 제거될 경우 해당 DN을 참조하는 다른 항목(예: 그룹 구성원 목록)으로 이 작업이 전파됩니다. 그룹 구성원이 연결 접미어에 있으면 이 플러그 인을 연결에 사용하여 정적 그룹의 관리를 간소화할 수 있습니다. 이 플러그 인은 원격 서버에 대한 쓰기 권한이 있어야만 연결 접미어에 액세스할 수 있습니다.
- `cn=uid uniqueness,cn=plugins,cn=config` - UID 고유성 플러그 인은 지정된 속성의 모든 새 속성 값이 서버에서 고유하도록 제어합니다. 이 플러그 인의 연결을 허용하면 전체 디렉토리 트리에서 고유성이 보장됩니다.

주 다음과 같은 구성 요소는 연결할 수 없습니다.

- 역할 플러그 인
 - 비밀번호 정책 구성 요소
 - 복제 플러그 인
-

콘솔에서 연결 정책 수정

1. Directory Server 콘솔의 구성 탭에서 데이터 노드를 선택하고 오른쪽 패널에서 연결 탭을 선택합니다.
2. 오른쪽 목록에서 LDAP 컨트롤을 하나 이상 선택하고 추가를 눌러 연결을 허용합니다. 추가 및 삭제 버튼을 사용하여 연결을 허용할 컨트롤 목록을 작성합니다.
LDAP 컨트롤은 해당 OID로 표시됩니다. 각 컨트롤의 이름과 자세한 설명은 [168페이지의 "LDAP 컨트롤의 연결 정책"](#)을 참조하십시오.
3. 동일한 탭의 아래쪽에 연결이 허용되는 서버 구성 요소가 표시됩니다. 오른쪽 목록에서 구성 요소 이름을 하나 이상 선택하고 추가를 눌러 연결을 허용합니다. 추가 및 삭제 버튼을 사용하여 연결을 허용할 구성 요소 목록을 작성합니다.
각 구성 요소에 대한 자세한 내용은 [170페이지의 "서버 구성 요소의 연결 정책"](#)을 참조하십시오.
4. 저장을 눌러 연결 정책을 저장합니다.
5. 서버를 다시 시작하여 변경 사항을 적용합니다.

명령줄에서 연결 정책 수정

cn=config,cn=chaining database,cn=plugins,cn=config 항목에는 연결 정책 구성의 속성이 포함되어 있습니다. 이 항목을 편집하려면 `ldapmodify` 명령을 사용합니다.

1. 여러 값을 갖는 `nsTransmittedControls` 속성에 연결을 허용할 모든 LDAP 컨트롤의 OID가 포함되도록 수정합니다. 연결할 수 있는 모든 컨트롤의 OID는 [168페이지](#)의 "[LDAP 컨트롤의 연결 정책](#)"을 참조하십시오.

예를 들어 아래 명령은 연결 컨트롤 목록에 효과적인 권한 제어를 추가합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=config,cn=chaining database,cn=plugins,cn=config
changetype: modify
add: nsTransmittedControls
nsTransmittedControls: 1.3.6.1.4.1.42.2.27.9.5.2
^D
```

클라이언트 응용 프로그램에서 사용자 정의 컨트롤을 사용하는 경우 이 컨트롤의 연결을 허용하기 위해 해당 OID를 `nsTransmittedControls` 속성에 추가할 수도 있습니다.

2. 연결을 허용할 모든 서버 구성 요소의 DN이 여러 값을 갖는 `nsActiveChainingComponents` 속성에 포함되도록 속성을 수정합니다. 각 구성 요소에 대한 자세한 내용은 [170페이지](#)의 "[서버 구성 요소의 연결 정책](#)"을 참조하십시오.

예를 들어 아래 명령은 연결 구성 요소 목록에 참조 무결성 구성 요소를 추가합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=config,cn=chaining database,cn=plugins,cn=config
changetype: modify
add: nsActiveChainingComponents
nsActiveChainingComponents: cn=referential integrity
postoperation,cn=components,cn=config
^D
```

3. 연결 정책 구성 항목의 수정이 끝나면 서버를 다시 시작하여 변경 사항을 적용해야 합니다.

연결 접미어 비활성화 또는 활성화

유지 관리나 보안을 위해 연결 접미어를 비활성화해야 하는 경우가 있습니다. 접미어를 비활성화하면 서버는 접미어에 액세스하는 클라이언트 작업에 응답하여 원격 서버에 연결할 수 없습니다. 기본 참조가 정의되어 있으면 클라이언트가 비활성화된 접미어에 액세스할 때 해당 참조가 반환됩니다.

콘솔에서 연결 접미어 비활성화 또는 활성화

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드를 확장하여 비활성화할 연결 접미어를 선택합니다.
2. 오른쪽 패널에서 설정 탭을 선택합니다. 작성된 모든 연결 접미어는 기본적으로 활성화됩니다.
3. 접미어를 비활성화하려면 "이 접미어에 대한 액세스를 가능하게 합니다" 확인란을 선택 취소하고 접미어를 활성화하려면 확인란을 선택합니다.
4. 저장을 눌러 변경 사항을 적용하면 접미어가 즉시 비활성화 또는 활성화됩니다.
5. 선택 사항으로, 접미어가 비활성화될 경우 이 접미어에 대한 모든 작업에 반환할 전역 기본 참조를 설정할 수도 있습니다. 이 설정은 최상위 구성 탭에서 루트 노드의 네트워크 탭에 있습니다. 자세한 내용은 [100페이지의 "콘솔에서 기본 참조 설정"](#)을 참조하십시오.

명령줄에서 접미어 비활성화 또는 활성화

1. 아래 명령을 실행하여 연결 접미어 항목의 `nsslapd-state` 속성을 편집합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=suffixDN,cn=mapping tree,cn=config
changetype: modify
replace: nsslapd-state
nsslapd-state: disabled 또는 backend
^D
```

여기서 `suffixDN`은 정의된 바와 같이 접미어 DN의 전체 문자열이며, 값에 쉼표가 있을 때 이스케이프하는 역슬래시(\)나 공백을 포함합니다. 접미어를 비활성화하려면 `nsslapd-state` 속성을 `disabled` 값으로 설정하고 전체 액세스 권한을 허용하려면 `backend` 값으로 설정합니다.

명령이 성공하면 접미어가 즉시 비활성화됩니다.

2. 선택 사항으로, 접미어가 비활성화될 경우 이 접미어에 대한 모든 작업에 반환할 전역 기본 참조를 설정할 수도 있습니다. 자세한 내용은 100페이지의 "명령줄에서 기본 참조 설정"을 참조하십시오.

액세스 권한 및 참조 설정

연결 접미어를 완전히 비활성화하지 않고 액세스를 제한하려면 읽기 전용 액세스를 허용하도록 액세스 권한을 수정할 수 있습니다. 이 경우 쓰기 작업을 위해 다른 서버에 대한 참조를 정의해야 합니다. 읽기 및 쓰기 액세스를 모두 거부하고 접미어에 대한 모든 작업에 반환할 참조를 정의할 수도 있습니다.

참조에 대한 일반적인 내용은 *Directory Server Deployment Planning Guide*를 참조하십시오.

콘솔에서 액세스 권한 및 참조 설정

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드를 확장하여 참조를 설정할 연결 접미어를 선택합니다.
2. 오른쪽 패널에서 설정 탭을 선택합니다. 연결 접미어를 활성화한 경우에만 권한과 참조를 설정할 수 있습니다.
3. 다음 라디오 버튼 중 하나를 선택하여 이 접미어 항목에 대한 모든 쓰기 작업에 반환할 응답을 설정합니다.
 - 쓰기 및 읽기 요청을 처리합니다 - 기본적으로 이 라디오 버튼이 선택되며 정상적인 동작을 나타냅니다. 읽기와 쓰기 작업이 모두 원격 서버로 전달되고 그 결과가 클라이언트로 반환됩니다. 참조도 정의할 수 있지만 클라이언트로 반환되지는 않습니다.
 - 읽기 요청을 처리하고 쓰기 요청에 대한 참조를 반환합니다 - 서버에서 읽기 요청만 전달하고 그 결과를 클라이언트로 반환합니다. 쓰기 요청에 대한 참조로 반환할 LDAP URL을 목록에 하나 이상 입력합니다.
 - 읽기와 쓰기 요청 모두에 대한 참조를 반환합니다 - 모든 작업에 대한 참조로 반환할 LDAP URL을 목록에 하나 이상 입력합니다. 전역 기본 참조를 사용하지 않고 특별히 이 접미어에 대한 참조를 정의할 수 있다는 점을 제외하면 이 동작은 접미어에 대한 액세스를 비활성화하는 것과 유사합니다.

4. 추가 및 제거 버튼을 사용하여 참조 목록을 편집합니다. 추가 버튼을 누르면 새 참조의 LDAP URL을 작성할 수 있는 대화 상자가 표시됩니다. 원격 서버 분기의 DN에 대한 참조를 작성할 수도 있습니다. LDAP URL의 구조에 대한 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.
 여러 개의 참조를 입력할 수 있습니다. 디렉토리는 클라이언트 응용 프로그램의 요청에 응답하여 이 목록에 있는 모든 참조를 반환합니다.
5. 저장을 눌러 변경 사항을 적용하면 새로운 권한 및 참조 설정이 즉시 실행됩니다.

콘솔에서 액세스 권한 및 참조 설정

아래 명령에서 *suffixDN*은 정의된 바와 같이 공백까지 포함하는 연결 접미어의 전체 문자열입니다. *LDAPURL*은 아래 예와 같이 대상의 호스트 이름, 포트 번호 및 DN이 포함된 유효한 URL입니다.

```
ldap://alternate.example.com:389/ou=People,dc=example,dc=com
```

1. 아래 명령을 실행하여 연결 접미어 항목을 편집합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=suffixDN,cn=mapping tree,cn=config
changetype: modify
replace: nsslapd-state
nsslapd-state: referral on update or referral
-
add: nsslapd-referral
nsslapd-referral: LDAPURL
^D
```

마지막 변경 명령문을 반복하여 다른 LDAP URL을 *nsslapd-referral* 속성에 원하는 대로 추가할 수도 있습니다.

nsslapd-state 속성 값이 *referral on update*이면 접미어는 읽기 전용으로 설정되고 쓰기 작업에 대한 참조로 모든 LDAP URL이 반환됩니다. 속성 값이 *referral*이면 읽기와 쓰기 작업이 모두 거부되고 모든 요청에 대해 참조가 반환됩니다.

2. 명령이 성공하면 접미어가 즉시 읽기 전용으로 설정되거나 액세스할 수 없게 되고 참조를 반환할 수 있습니다.

연결 매개 변수 수정

연결 접미어가 정의되면 연결을 제어하는 매개 변수를 수정할 수 있습니다. 원격 서버에 액세스하는 방법을 지정하거나 프록시에 사용된 DN을 변경할 수 있으며 원격 서버를 변경할 수도 있습니다. 또한 서버의 연결 서버와의 연결 및 유지 관리를 제어하는 성능 매개 변수를 수정할 수 있습니다.

콘솔에서 연결 매개 변수 수정

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드를 확장하여 수정할 연결 접미어를 선택합니다.
2. 오른쪽 패널에서 원격 서버 탭을 선택합니다.
3. 원격 서버의 이름이나 포트를 변경하려면 원격 서버 URL 필드를 수정합니다. URL에는 원격 서버 하나 이상의 호스트 이름과 포트 번호(선택 사항)가 아래 형식으로 포함되어 있습니다.

```
ldap[s]://hostname[:port][hostname[:port]].../
```

이 URL에 접미어 정보는 포함되지 않습니다. 보안 포트 지정에 대한 자세한 내용은 [167페이지의 "SSL을 사용한 연결"](#)을 참조하십시오. 첫 서버가 연결 요청에 응답하지 않으면 URL에 표시된 순서대로 각 서버에 연결합니다. LDAP URL에 열거된 모든 원격 서버에는 연결 접미어의 기본 항목인 *suffixDN*이 있어야 합니다.

4. 프록시 사용자의 DN을 변경하려면 바인드 DN 필드에 새 값을 입력합니다. 이 DN에 해당하는 비밀번호를 비밀번호 필드에 입력하고 다시 확인합니다.

*proxyDN*은 원격 서버 사용자의 DN입니다. 로컬 서버는 원격 서버에 있는 접미어 내용에 액세스할 때 이 DN을 프록시로 사용합니다. 연결 접미어를 통해 수행된 작업은 *creatorsName* 속성과 *modifiersName* 속성에 이 프록시 아이디를 사용합니다. 프록시 DN을 지정하지 않으면 로컬 서버는 원격 서버에 액세스할 때 익명으로 바인드합니다.

5. 탭 아래쪽의 텍스트 상자에 이 접미어의 연결을 허용하는 데 필요한 ACI가 표시됩니다. 원격 서버 URL을 변경한 경우 새 원격 서버에서 *suffixDN*에 해당하는 항목에 이 ACI를 추가해야 합니다. 프록시 DN을 수정한 경우에는 모든 연결 서버에서 ACI를 업데이트해야 합니다. ACI 복사 버튼을 사용하여 ACI 텍스트를 시스템 클립보드에 복사한 다음 붙여넣습니다.

6. 제한 및 제어 탭을 선택하여 연결 요청에 대한 매개 변수를 구성합니다. 계단식 연결 매개 변수에 대해서는 [182페이지의 "계단식 연결 구성"](#)에서 설명합니다.
7. 클라이언트 반환 제어 매개 변수를 설정하여 연결 작업의 크기와 시간을 제한합니다.
 - 범위 검색에서 참조 반환 - 검색 범위가 연결 접미어에만 제한되는 검색은 결과가 두 번 전송되기 때문에 비효율적입니다. 기본적으로 서버는 참조를 연결 서버로 반환하여 클라이언트가 연결 서버에서 직접 검색하도록 강제합니다. 이 옵션을 선택 취소하는 경우 다음과 같은 매개 변수를 설정하여 연결할 결과의 크기를 제한해야 합니다.
 - 크기 제한 또는 크기 제한 없음 - 이 매개 변수는 연결 검색 작업에 응답하여 반환할 항목 수를 지정합니다. 기본 크기 제한은 2000개입니다. 연결 접미어에 대한 광범위한 검색을 제한하려면 이 매개 변수 값을 낮게 설정하십시오. 항상 원격 서버의 크기 설정에 따라 작업이 제한됩니다.
 - 시간 제한 또는 시간 제한 없음 - 이 매개 변수는 연결 작업 시간을 제어합니다. 기본 시간 제한은 3600초(1시간)입니다. 연결 접미어에 대한 작업 시간을 제한하려면 이 매개 변수 값을 낮게 설정하십시오. 항상 원격 서버의 시간 설정에 따라 작업이 제한됩니다.
8. 연결 관리 매개 변수를 설정하여 서버에서 네트워크 연결 및 원격 서버와의 바인드를 관리하는 방법을 제어합니다.
 - 최대 LDAP 연결 수. 연결 접미어와 원격 서버 사이에 구성할 수 있는 동시 LDAP 작업 연결의 최대 개수입니다. 기본값은 10개입니다.
 - 최대 바인드 연결 수. 연결 접미어와 원격 서버 간에 구성할 수 있는 최대 동시 바인드 연결 수입니다. 기본값은 3개입니다.
 - 연결 당 최대 바인드 수. LDAP 연결 당 동시에 허용되는 최대 바인드 작업 수입니다. 기본값은 연결 당 아직 해결되지 않은 바인드 작업 10개입니다.
 - 최대 바인드 다시 시도 횟수. 오류 시 연결 접미어가 원격 서버로 재바인드를 시도하는 횟수입니다. 값을 0으로 설정하면 연결 접미어는 바인드를 한 번만 시도합니다. 기본값은 3회입니다.

- 연결 당 최대 작업 수. LDAP 연결 당 동시에 허용되는 최대 작업 수입니다. 기본값은 연결 당 10개입니다.
- 바인드 시간 초과 또는 바인드 시간 초과 없음. 연결 접미어에 대한 바인드 시도가 시간 초과될 때까지의 시간(초)입니다. 기본값은 15초입니다.
- 중단 전 시간 초과 또는 중단 전 시간 초과 없음. 서버에서 작업 중단 여부를 확인할 때까지의 시간(초)입니다. 기본값은 2초입니다.
- 연결 수명(초) 또는 수명 제한 없음. 연결 접미어와 원격 서버간의 연결을 다시 사용할 수 있도록 열어 두는 시간입니다. 연결을 열어 두면 속도는 더 빠르지만 많은 자원이 사용됩니다. 예를 들어, 전화 접속 연결을 사용하는 경우 연결 시간을 제한하는 것이 좋습니다. 기본적으로 연결 수명은 제한되지 않습니다.

오류 감지 매개 변수는 콘솔에서 사용할 수 없습니다. [178페이지의 "명령줄에서 연결 매개 변수 수정"](#)을 참조하십시오.

9. 연결 매개 변수가 모두 설정되면 저장을 누릅니다.

명령줄에서 연결 매개 변수 수정

명령줄을 사용하면 콘솔에서 설정 가능한 모든 매개 변수는 물론, [159페이지의 "오류 감지 매개 변수"](#)에 설명된 추가 매개 변수까지 구성할 수 있습니다.

1. 아래 명령을 실행하여 수정할 접미어에 해당하는 연결 구성 항목을 편집합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=databaseName,cn=chaining database,cn=plugins,cn=config
changetype: modify
replace: attributeName
attributeName: attributeValue
-
replace: attributeName2
attributeName2: attributeValue2
...
^D
```

사용할 수 있는 속성 이름과 값에 대해서는 이후 단계에서 설명합니다. 한 명령에 여러 개의 변경 명령문을 사용하여 많은 매개 변수를 동시에 변경할 수도 있습니다.

2. nsfarmserverURL 속성을 수정하여 원격 서버의 이름 또는 포트를 변경합니다. 속성 값은 하나 이상의 원격 서버 호스트 이름과 포트 번호(선택 사항)가 아래 형식으로 포함되어 있는 URL입니다.

```
ldap[s]://hostname[:port][ hostname[:port]].../
```

이 URL에 접미어 정보는 포함되지 않습니다. 보안 포트 지정에 대한 자세한 내용은 [167페이지의 "SSL을 사용한 연결"](#)을 참조하십시오. 첫 서버가 연결 요청에 응답하지 않으면 URL에 표시된 순서대로 각 서버에 연결합니다. LDAP URL에 열거된 모든 원격 서버에는 연결 접미어의 기본 항목인 *suffixDN*이 있어야 합니다.

3. nsmultiplexorBindDN 속성과 nsmultiplexorCredentials 속성을 수정하여 원격 서버에 대한 프록시 액세스에 사용된 DN을 변경합니다.

로컬 서버는 원격 서버에 있는 접미어 내용에 액세스할 때 이 DN을 프록시로 사용합니다. 연결 접미어를 통해 수행된 작업은 creatorsName 속성과 modifiersName 속성에 이 프록시 아이디를 사용합니다. 프록시 DN을 지정하지 않으면 로컬 서버는 원격 서버에 액세스할 때 익명으로 바인드합니다.

4. 프록시 DN이나 자격 증명을 수정한 경우 아래 명령을 실행하여 원격 서버에 해당 ACI를 작성해야 합니다. 이 ACI는 연결을 통한 프록시 작업을 허용하는 데 필요합니다.

```
ldapmodify -h host2 -p port2 -D "cn=Directory Manager" -w password2
dn: suffixDN
changetype: modify
add: aci
aci: (targetattr=*)(target = "ldap:///suffixDN")(version 3.0;acl
  "Allows use of admin for chaining"; allow (proxy)
  (userdn="ldap:///proxyDN");)
^D
```

5. [157페이지의 "기본 연결 매개 변수 설정"](#)에 설명된 속성을 설정하여 원격 서버의 연결 및 작업 처리를 제어합니다. 계단식 매개 변수에 대해서는 [182페이지의 "계단식 연결 구성"](#)에서 자세히 설명합니다.

스레드 사용 최적화

연결에 사용된 스레드 자원을 고려하여 서버에서 전역으로 사용되는 스레드 수를 설정할 수도 있습니다. 연결 작업은 원격 서버로 전달되어야 하기 때문에 오랜 시간이 소요될 수 있으며, 원격 서버에서 작업을 처리하는 동안 해당 스레드는 유휴 상태로 있습니다. 연결 서버에서 지연이 증가하면 이 시간 동안 더 많은 스레드가 로컬 작업 처리에 사용될 수 있도록 전역 스레드 수를 늘려야 합니다.

기본적으로 서버에서 사용되는 스레드 수는 30개입니다. 그러나 연결 접미어를 사용하는 경우 작업 처리에 사용할 수 있는 스레드 수를 늘려 성능을 향상시킬 수 있습니다. 필요한 스레드 수는 연결 접미어 개수, 연결 접미어에 대한 작업 유형, 원격 서버의 평균 작업 처리 시간 등에 따라 결정됩니다.

일반적으로 로컬 접미어 개수만큼 연결 접미어에 대한 작업이 구성된다고 가정하여 연결 접미어 당 스레드 수를 5-10개 정도 늘려야 합니다.

콘솔에서 스레드 자원 설정

1. Directory Server 콘솔의 최상위 구성 탭에서 성능 노드를 누르고 오른쪽 패널에서 기타 탭을 선택합니다.
2. 최대 스레드 수 필드에 새 값을 입력합니다.
3. 확인을 눌러 변경 사항을 저장하고 서버를 다시 시작해야만 변경 사항이 적용된다는 메시지를 확인합니다.
4. 디렉토리 서버를 다시 시작하여 변경된 개수의 스레드를 사용합니다.

명령줄에서 스레드 자원 설정

1. 아래 명령을 실행하여 전역 구성 항목의 스레드 수를 수정합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=config
changetype: modify
replace: nsslapd-threadnumber
nsslapd-threadnumber: newThreadNumber
^D
```

2. 디렉토리 서버를 다시 시작하여 변경된 개수의 스레드를 사용합니다.

연결 접미어 삭제

연결 접미어를 삭제하면 로컬 디렉토리 트리에서 해당 접미어를 액세스할 수 없지만 연결 서버에 있는 항목이나 접미어는 삭제되지 않습니다. 디렉토리에서 부모 접미어만 삭제하고 해당 하위 접미어를 새로운 루트 접미어로 유지할 수도 있습니다.

콘솔에서 연결 접미어 삭제

1. Directory Server 콘솔의 구성 탭에서 데이터 노드를 확장합니다.
2. 제거할 접미어를 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 삭제를 선택합니다. 또는 접미어 노드를 선택하고 객체 메뉴에서 삭제를 선택할 수도 있습니다.
3. 이 연결 접미어를 통해 액세스할 수 있는 항목은 원격 디렉토리에서 제거되지 *않는다*고 알려주는 확인 대화 상자가 표시됩니다.
부모 접미어의 경우 모든 하위 접미어를 재귀적으로 삭제할 수 있습니다. 전체 분기를 제거하려면 "이 접미어와 해당 하위 접미어를 모두 삭제합니다"를 선택합니다. 디렉토리에서 특정 접미어만 제거하고 해당 하위 접미어를 그대로 유지하려면 "이 접미어만 삭제합니다"를 선택합니다.
4. 확인을 눌러 접미어를 삭제합니다.
콘솔의 작업 완료 상황을 단계별로 보여주는 진행률 대화 상자가 표시됩니다.

명령줄에서 접미어 삭제

명령줄에서 접미어를 삭제하려면 `ldapdelete` 명령을 사용하여 해당 구성 항목을 디렉토리에서 제거합니다.

하위 접미어를 포함한 전체 분기를 삭제하려면 삭제된 부모의 하위 접미어를 찾아서 각 하위 접미어 및 해당 하위 접미어에 대해 같은 절차를 반복해야 합니다.

1. 아래 명령을 실행하여 접미어 구성 항목을 제거합니다.

```
ldapdelete -h host -p port -D "cn=Directory Manager" -w password
cn=suffixDN,cn=mapping tree,cn=config
```

이 명령을 실행하면 연결 접미어와 해당 원격 항목이 더 이상 디렉토리에 표시되지 않습니다.

- 아래 명령을 실행하여 `cn=databaseName, cn=chaining database, cn=plugins, cn=config`에 있는 해당 데이터베이스 구성 항목과 하위 모니터 항목을 제거합니다.

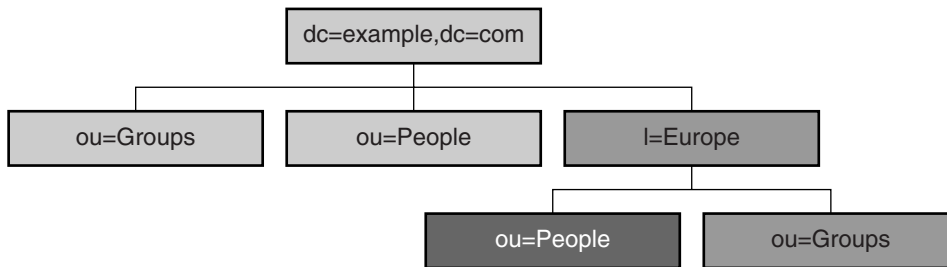
```
ldapdelete -h host -p port -D "cn=Directory Manager" -w password
cn=monitor, cn=dbName, cn=chaining database, cn=plugins, cn=config
cn=dbName, cn=chaining database, cn=plugins, cn=config
```

계단식 연결 구성

계단식 연결의 경우 특정 서버로부터 연결된 하위 트리 자체가 연결 접미어이거나 연결 하위 접미어를 포함할 수 있습니다. 특정 서버의 연결 접미어가 작업에 사용되면 이 작업은 중간 서버로 전달되고, 다시 이 서버가 세 번째 서버에 연결합니다. 계단식 연결은 디렉토리 트리의 모든 데이터를 액세스하기 위해 서버간의 홉이 두 번 이상 필요한 경우에 발생합니다.

예를 들어 아래 다이어그램은 `ou=People, l=Europe, dc=example, dc=com` 항목에 대한 액세스가 어떻게 서버 A에서 서버 B로 연결되었다가 다시 서버 C로 연결되는지 보여줍니다. 서버 A에는 루트 접미어 `dc=example, dc=com`이 있으며 `l=Europe, dc=example, dc=com` 분기에 서버 B에 대한 연결 하위 접미어가 있습니다. 서버 B에는 항목 `l=Europe, dc=example, dc=com`이 포함되지만, 분기 `ou=People, l=Europe, dc=example, dc=com`은 서버 C에 대한 연결 하위 접미어입니다. 서버 C에 항목 `ou=People, l=Europe, dc=example, dc=com`이 실제로 포함됩니다.

그림 3-1 3대의 서버를 사용한 계단식 연결



- 서버 A의 루트 접미어
- 서버 A에서 서버 B로 연결된 하위 접미어

계단식 매개 변수 설정

계단식 연결에서는 다음 두 개의 연결 매개 변수를 구성해야 합니다.

- 연결 토폴로지에 잘못된 루프가 있을 경우 이를 감지할 수 있도록 모든 서버에서 루프 감지를 구성해야 합니다. 루프 감지를 사용하지 않으면 루프에 있는 서버에서 오버로드가 발생할 때까지 계속해서 작업을 전달합니다.
- 모든 중간 연결 접미어에서 로컬 ACI를 평가하도록 구성해야 합니다. 일반적으로 연결 접미어의 첫 수준에서는 이 작업이 수행되지 않습니다.

콘솔에서 계단식 매개 변수 설정

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드를 확장하여 수정할 연결 접미어를 선택합니다.
2. 오른쪽 패널에서 계단식 연결 매개 변수를 수정할 수 있는 제한 및 제어 탭을 선택합니다.
3. 계단식 연결의 모든 중간 서버에서 로컬 ACI 확인란을 선택합니다.
 사용자의 액세스 권한은 첫 번째 서버에서 평가되지 않고 프록시를 통해 두 번째 서버에서 평가되기 때문에 단일 수준 연결에서는 이 확인란을 선택하지 않습니다. 하지만 계단식 연결의 중간 서버에서는 작업을 다시 전달하기 전에 액세스 제어를 수행하도록 ACI 확인 기능을 사용해야 합니다.
4. 계단식 연결의 모든 서버에서 토폴로지의 모든 연결 작업에 허용되는 최대 홉 수를 설정합니다. 같은 작업이 다른 연결 접미어로 전달될 때마다 홉 수가 하나씩 가산되며, 이 제한에 도달할 경우 연결 접미어는 더 이상 작업을 전달하지 않습니다.
 가장 긴 계단식 연결의 홉 수보다 큰 수를 설정해야 합니다. 제한에 도달할 경우 서버에서 토폴로지에 잘못된 루프가 있다고 가정하기 때문에 해당 작업이 중단됩니다.
 또한, 184페이지의 "계단식 연결을 위한 LDAP 컨트롤 전송"에 설명된 것처럼 연결 구성에서 루프 감지 제어를 허용하도록 설정해야 합니다.
5. 계단식 매개 변수가 모두 설정되면 저장을 누릅니다.

명령줄에서 계단식 매개 변수 설정

1. 모든 중간 서버에서 아래 명령을 실행하여 계단식 접미어에 대한 연결 구성 항목을 편집합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=databaseName,cn=chaining database,cn=plugins,cn=config
changetype: modify
replace: nsCheckLocalACI
nsCheckLocalACI: on
-
changetype: modify
replace: nsHopLimit
nsHopLimit: maximumHops
^D
```

*maximumHops*를 가장 긴 계단식 연결의 홉 수보다 큰 값으로 설정해야 합니다. 제한에 도달할 경우 서버에서 토폴로지에 잘못된 루프가 있다고 가정하기 때문에 해당 작업이 중단됩니다. 또한 [184페이지의 "계단식 연결을 위한 LDAP 컨트롤 전송"](#)에 설명된 것처럼 연결 구성에서 루프 감지 제어를 허용하도록 설정해야 합니다.

2. 계단식 연결의 다른 모든 서버에서 아래 명령을 실행하여 계단식 접미어에 대한 연결 구성 항목을 편집합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=databaseName,cn=chaining database,cn=plugins,cn=config
replace: nsHopLimit
nsHopLimit: maximumHops
^D
```

여기서 *maximumHops*의 정의는 이전 단계의 정의와 같습니다.

계단식 연결을 위한 LDAP 컨트롤 전송

기본적으로 연결 접미어는 프록시 인증 컨트롤을 전송하지 않습니다. 그러나 연결 접미어가 다른 연결 접미어에 연결하는 경우 이 컨트롤을 사용하여 원격 서버의 액세스 제어에 필요한 사용자 인증을 전송해야 합니다. 중간 연결 접미어도 이 컨트롤의 연결을 허용해야 합니다.

프록시 인증 컨트롤에 대한 두 번째 프로토콜이 최근에 정의되었습니다. 서버 버전이 틀리면 사용되는 컨트롤도 다를 수 있으므로 모든 계단식 서버에서 이전 프록시 인증 컨트롤과 새로운 프록시 인증 컨트롤의 연결을 모두 허용하도록 구성해야 합니다.

또한, 계단식 연결 중에 루프를 방지하려면 루프 감지 컨트롤이 필요합니다. 기본적으로 이 컨트롤은 연결 작업과 함께 전달되지만 만약을 위해 구성을 확인해야 합니다. 서버에서 이 컨트롤의 연결을 허용하지 않으면 해당 서버와 관련된 모든 루프는 감지되지 않습니다.

168페이지의 "연결 정책 구성"에 설명된 단계에 따라 다음 세 가지 컨트롤의 연결이 허용되는지 확인합니다.

- 2.16.840.1.113730.3.4.12 - 프록시 인증 컨트롤(이전 사양)
- 2.16.840.1.113730.3.4.18 - 프록시 인증 컨트롤(새 사양)
- 1.3.6.1.4.1.1466.29539.12 - 루프 감지 컨트롤

데이터 백업 및 복원

디렉토리 서버에서 관리하는 데이터는 대량으로 가져오는 경우가 많습니다. Directory Server는 전체 접미어를 가져오거나 내보내기 위한 도구를 제공합니다. 모든 접미어를 동시에 백업하고, 이 백업을 사용하여 모든 데이터를 복원할 수 있는 도구도 제공됩니다.

일반 텍스트 LDIF(LDAP Data Interchange Format)를 사용하여 데이터를 백업 및 복원할 수 있습니다.

이 장에서는 디렉토리 데이터의 백업 및 복원을 위한 다음 절차를 설명합니다.

- 접미어 읽기 전용 모드 설정
- 데이터 가져오기
- 데이터 내보내기
- 데이터 백업
- 백업을 사용한 데이터 복원

주 두 가지 이상 버전의 Directory Server를 실행할 경우 이 장의 모든 예에서 Directory Server 5.2를 기본 버전으로 가정합니다. 그렇지 않을 경우 다음 명령을 한 번 실행하여 5.2를 기본 버전으로 설정해야 합니다.

```
# /usr/sbin/directoryserver -d 5.2
```

또는 directoryserver 명령을 실행할 때마다 -useversion 옵션을 사용하여 버전을 지정하십시오. 예를 들면, 다음과 같습니다.

```
# /usr/sbin/directoryserver -useversion 5.2 ldif2db
```

접미어 읽기 전용 모드 설정

Directory Server에서 내보내기 또는 백업 작업을 수행하기 전에 임의의 접미어에 대해 읽기 전용 모드를 활성화하면 특정 시간에 접미어 내용을 안전하게 유지할 수 있습니다. 또한, 가져오기 또는 복원 작업을 수행하기 전에 영향을 받는 접미어가 읽기 전용 모드로 설정되어 있지 않은지 확인해야 합니다.

Directory Server 콘솔과 명령줄 유틸리티는 내보내기 또는 백업 작업 전에 디렉토리를 읽기 전용 모드로 자동 설정하지 않습니다. 이렇게 하면 디렉토리를 업데이트할 수 없게 되기 때문입니다. 하지만 다중 마스터 구성에서 서버 한 대만 읽기 전용 모드로 설정하는 경우 다른 마스터에서는 계속 데이터를 쓸 수 있습니다.

접미어를 읽기 전용으로 설정하려면 151페이지의 "액세스 권한 및 참조 설정"에 설명된 절차에 따라 수행합니다. 또는 51페이지의 "전역 읽기 전용 모드 설정"에 설명된 것처럼 전체 디렉토리 서버를 쓸 수 없게 설정할 수도 있습니다.

데이터 가져오기

Directory Server는 데이터를 가져오는 두 가지 방법을 제공합니다.

- LDIF 파일을 가져와서 디렉토리 접미어의 항목을 대량으로 추가, 수정 및 삭제합니다.
- LDIF 파일을 사용하여 접미어를 초기화함으로써 접미어의 현재 데이터를 삭제하고 LDIF 파일의 내용으로 바꿉니다.

두 방법 모두 Directory Server 콘솔과 명령줄 유틸리티를 통해 사용할 수 있습니다.

주

UTF-8 문자 집합 인코딩을 사용하는 LDIF 파일만 가져올 수 있습니다.

LDIF를 가져올 경우 부모 항목이 디렉토리에 있거나 먼저 이 파일을 사용하여 부모 항목을 추가해야 합니다. 접미어를 초기화할 경우에는 LDIF 파일에 해당 접미어의 모든 디렉토리 트리 노드와 루트 항목이 포함되어 있어야 합니다.

아래 표에는 가져오기와 초기화의 차이점이 설명되어 있습니다.

표 4-1 데이터 가져오기와 접미어 초기화 비교

비교 영역	데이터 가져오기	접미어 초기화
내용 덮어쓰기	아니요	예
LDAP 작업	추가, 수정 및 삭제	추가만
성능	느림	빠름
서버 장애에 대한 응답	최상의 노력(장애가 발생하기 전의 모든 변경 사항 그대로 유지)	자동(장애 후에 모든 변경 사항 손실됨)
LDIF 파일 위치	클라이언트 시스템	로컬 콘솔 또는 로컬 서버
구성 정보(cn=config) 가져오기	예	아니요

LDIF 파일 가져오기

가져오기 작업을 수행하면 Directory Server 콘솔은 `ldapmodify` 작업을 수행하여 새 항목을 디렉토리에 추가합니다. 항목은 LDIF 파일에 지정되어 있으며, 이 파일에는 가져오기 작업의 일부로 기존 항목을 수정하거나 삭제하는 업데이트 명령문도 포함될 수 있습니다.

Directory Server에서 관리하는 모든 접미어 및 구성에 정의된 연결 접미어 또는 연결 하위 접미어를 대상으로 항목을 가져올 수 있습니다. 항목을 추가하는 다른 모든 작업과 마찬가지로 서버는 가져오는 새 항목을 모두 색인화합니다.

콘솔에서 LDIF 가져오기

가져오기를 수행하려면 디렉토리 관리자 또는 관리자로 로그인해야 합니다.

1. Directory Server 콘솔의 최상위 태스크 탭에서 아래쪽으로 스크롤하여 "LDIF에서 파일 가져오기" 옆에 있는 버튼을 누릅니다.
LDIF 가져오기 대화 상자가 표시됩니다.
2. LDIF 가져오기 대화 상자의 LDIF 파일 필드에 가져올 LDIF 파일의 전체 경로를 입력하거나 찾아보기를 눌러 로컬 파일 시스템에서 해당 파일을 선택합니다.

원격 시스템의 디렉토리에 액세스하는 경우 필드 이름이 "LDIF 파일(콘솔 시스템)"로 표시됩니다. 이 레이블은 사용자가 원격 디렉토리 서버 시스템이 아닌 로컬 파일 시스템을 탐색하고 있음을 나타냅니다.

3. 다음 옵션을 원하는 대로 설정합니다.
 - a. "추가만" - LDIF 파일에는 기본 추가 지침 외에도 수정 및 삭제 지침이 포함될 수 있습니다. 콘솔에서 추가 지침만 수행하고 LDIF 파일의 다른 모든 지침을 무시하려면 이 확인란을 선택합니다.
 - b. "오류 발생 시 계속" - 오류가 발생한 경우에도 서버에서 가져오기 작업을 계속 수행하려면 이 확인란을 선택합니다. 예를 들어, 일부 항목이 접미어에 이미 존재하는 LDIF 파일을 가져오는 경우 이 옵션을 사용할 수 있습니다. 콘솔은 가져오기 작업을 수행하는 동안 이미 존재하는 항목과 같은 오류를 거부용 파일에 기록합니다.

이 확인란이 선택되어 있지 않으면 가져오기 작업은 첫 번째 오류가 발생한 후에 중단됩니다. LDIF 파일에서 오류 전의 항목은 성공적으로 가져오며 디렉토리에 남아 있습니다.
4. "거부용 파일" 필드에 콘솔에서 가져올 수 없는 모든 항목을 기록할 파일의 전체 경로를 입력하거나 찾아보기를 눌러 로컬 파일 시스템에서 해당 파일을 선택합니다.

예를 들어 디렉토리에 이미 존재하는 항목이나 부모 객체가 없는 항목은 가져올 수 없습니다. 콘솔은 서버에서 보낸 오류 메시지를 거부용 파일에 씁니다.

이 필드를 비워 두면 서버에서 거부된 항목을 기록하지 않습니다.
5. 확인을 눌러 가져오기 작업을 시작합니다.

Directory Server 콘솔에서 작업 상태와 발생한 오류 텍스트가 포함된 대화 상자를 표시합니다. "거부용 파일" 필드에 파일이 지정되어 있으면 해당 파일에도 모든 오류 메시지가 기록됩니다.

명령줄에서 LDIF 가져오기

`directoryserver ldif2ldap` 명령은 LDAP를 통해 LDIF 파일을 가져오며 포함된 모든 작업을 수행합니다. 이 스크립트를 사용하여 데이터를 모든 디렉토리 접미어에 동시에 가져올 수 있습니다. `ldif2ldap`를 사용하여 가져오려면 서버를 실행해야 합니다.

이 명령의 전체 경로는 다음과 같습니다.

```
# /usr/sbin/directoryserver -s serverID ldif2ldap
```

아래 예에서는 `ldif2ldap` 명령을 사용하여 가져오기를 수행합니다. 디렉토리 관리자 같이 루트 권한이 있는 사용자로 인증되면 루트 권한이 없어도 이 명령을 실행할 수 있습니다. 마지막 매개 변수는 가져올 LDIF 파일 이름을 지정합니다.

```
# /usr/sbin/directoryserver -s example ldif2ldap \
-D "cn=Directory Manager" -w password \
-f /var/opt/mps/serverroot/slapd-example//ldif/demo.ldif
```

이 명령의 사용에 대한 자세한 내용은 *Directory Server Man Page Reference*를 참조하십시오.

접미어 초기화

접미어를 초기화하면 추가할 항목만 포함된 LDIF 파일의 내용이 접미어의 기존 데이터를 덮어씁니다.

주의 관리하는 서버가 Configuration Directory Server일 경우 데이터를 복원하지 않으려면 LDIF 파일의 접미어를 초기화할 때 o=NetscapeRoot 접미어를 덮어쓰지 마십시오. 이 접미어를 덮어쓰면 정보가 삭제되어 모든 Sun Java System 서버를 다시 설치해야 합니다.

접미어를 초기화하려면 디렉토리 관리자 또는 관리자로 인증되어야 합니다. 보안상, 디렉토리 관리자 및 관리자만 접미어의 루트 항목(예: dc=example, dc=com)에 액세스할 수 있으므로 루트 항목이 포함된 LDIF 파일을 가져오려면 이러한 아이디를 사용해야 합니다.

콘솔에서 접미어 초기화

주의 이 절차에서는 접미어에 있는 데이터를 덮어씁니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드를 확장하여 초기화할 접미어를 표시합니다.
2. 접미어 노드를 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 초기화를 선택합니다. 또는 접미어 노드를 선택하고 객체 메뉴에서 초기화를 선택할 수도 있습니다. 접미어 초기화 대화 상자가 표시됩니다.
3. "LDIF 파일" 필드에 초기화에 사용할 LDIF 파일의 전체 경로를 입력하거나 찾아보기를 눌러 사용자 시스템에서 해당 파일을 찾습니다.

- 가져올 파일이 있는 로컬 시스템에서 콘솔을 실행하는 경우 단계 6으로 넘어갑니다. LDIF 파일이 있는 서버의 원격 시스템에서 콘솔을 실행하는 경우에는 다음 옵션 중 하나를 선택합니다.

콘솔. LDIF 파일이 콘솔을 실행 중인 시스템에 있음을 나타냅니다. 이 경우 파일을 찾아볼 수 있습니다.

서버. LDIF 파일이 원격 서버에 있음을 나타냅니다. 이 경우 찾아보기 버튼이 비활성화됩니다. 기본적으로 콘솔은 아래 디렉토리에서 이 파일을 찾습니다.

ServerRoot/slapd-*serverID*/ldif

- 확인을 누릅니다.
- 접미어에 있는 데이터를 덮어쓸 것을 확인합니다. 접미어 초기화가 수행되며 오류가 발생하면 대화 상자로 보고됩니다.

ldif2db 명령을 사용하여 접미어 초기화

directoryserver ldif2db 명령은 접미어를 초기화하고 기존 데이터를 덮어씹습니다. 이 명령을 사용하여 가져오기를 수행하려면 서버를 종료해야 합니다.

기본적으로 명령은 기존의 o=NetscapeRoot 구성 정보를 먼저 저장한 다음, 가져오는 파일에 있는 o=NetscapeRoot 구성 정보와 병합합니다.

주의 이 명령은 접미어에 있는 데이터를 덮어씹습니다.

서버를 중지한 상태에서 LDIF 파일을 가져오려면

- 명령줄에서 루트로 아래 명령을 실행하여 서버를 중지합니다.
/usr/sbin/directoryserver -s *serverID* stop
- 가져오기 명령을 실행합니다.
/usr/sbin/directoryserver -s *serverID* ldif2db ...
- 다음과 같이 서버를 시작합니다.
/usr/sbin/directoryserver -s *serverID* start

아래 예에서는 ldif2db 명령을 사용하여 LDIF 파일 두 개를 하나의 접미어로 가져옵니다.


```
/usr/sbin/directoryserver -s example ldif2db -n Database1 \  
-i /var/opt/mps/serverroot/slapd-example/ldif/demo.ldif \  
-i /var/opt/mps/serverroot/slapd-example/ldif/demo2.ldif
```

표 4-2 위의 예에 사용된 ldif2db 옵션에 대한 설명

옵션	설명
-n	데이터를 가져오는 데이터베이스의 이름을 지정합니다. 주의: LDIF 파일에 있는 접미어에 해당하지 않는 데이터베이스를 지정하면 데이터베이스에 있는 모든 데이터가 삭제되고 가져오기 작업이 실패합니다. 데이터베이스 이름을 잘못 입력하지 않도록 주의하십시오.
-i	가져올 LDIF 파일의 전체 경로 이름을 지정합니다. 이 옵션은 필수입니다. -i 인수를 여러 개 사용하여 두 개 이상의 LDIF 파일을 동시에 가져올 수 있습니다. 여러 파일을 가져올 경우 서버는 명령줄에 지정된 순서대로 LDIF 파일을 가져옵니다.

이 명령의 사용에 대한 자세한 내용은 *Directory Server Man Page Reference*를 참조하십시오.

ldif2db-task를 사용하여 접미어 초기화

ldif2db 명령과 마찬가지로 directoryserver ldif2db-task 명령은 지정한 접미어에 있는 데이터를 덮어씁니다. 이 스크립트를 사용하여 가져오기를 수행하려면 서버를 실행해야 합니다.

이 스크립트에 대한 명령은 다음과 같습니다.

```
# /usr/sbin/directoryserver -s serverID ldif2db-task ...
```

다음 예에서는 ldif2db-task를 사용하여 LDIF 파일을 가져옵니다. 디렉토리 관리자 같은 루트 권한을 가진 사용자로 인증되면 루트 권한이 없어도 명령을 실행할 수 있습니다.

```
/usr/sbin/directoryserver -s example ldif2db-task \  
-D "cn=Directory Manager" -w password -n Database1 \  
-i /var/opt/mps/serverroot/slapd-example/ldif/demo.ldif
```

아래 표에는 예에 사용된 ldif2db-task 옵션이 설명되어 있습니다.

표 4-3 위의 예에 사용된 ldif2db-task 옵션에 대한 설명

옵션	설명
-D	디렉토리 관리자 같이 루트 권한을 가진 사용자 DN을 지정합니다.

표 4-3 위의 예에 사용된 `ldif2db-task` 옵션에 대한 설명

옵션	설명
<code>-w</code>	사용자의 비밀번호를 지정합니다.
<code>-n</code>	데이터를 가져오는 데이터베이스의 이름을 지정합니다. 주의: LDIF 파일에 있는 접미어에 해당하지 않는 데이터베이스를 지정하면 데이터베이스에 있는 모든 데이터가 삭제되고 가져오기 작업이 실패합니다. 데이터베이스 이름을 잘못 입력하지 않도록 주의하십시오.
<code>-i</code>	가져올 LDIF 파일의 전체 경로 이름을 지정합니다. 이 옵션은 필수입니다. <code>-i</code> 인수를 여러 개 사용하여 두 개 이상의 LDIF 파일을 동시에 가져올 수 있습니다. 여러 파일을 가져올 경우 서버는 명령줄에 지정된 순서대로 LDIF 파일을 가져옵니다.

이 명령의 사용에 대한 자세한 내용은 *Directory Server Man Page Reference*를 참조하십시오.

데이터 내보내기

LDIF를 사용하여 디렉토리의 내용을 내보낼 수 있습니다. 데이터 내보내는 다음과 같은 작업에 도움이 됩니다.

- 서버에 있는 데이터 백업
- 다른 디렉토리 서버로 데이터 복사
- 다른 응용 프로그램으로 데이터 내보내기
- 디렉토리 토폴로지를 변경한 후에 접미어 다시 채우기

내보내기 작업 시 구성 정보(`cn=config`)는 내보내지 않습니다.

주의 내보내기 작업을 진행하는 중에는 서버를 중지하지 마십시오.

콘솔에서 전체 디렉토리를 LDIF로 내보내기

내보낸 최종 파일의 위치에 따라 디렉토리 데이터 전체 또는 일부를 LDIF로 내보낼 수 있습니다. LDIF 파일이 서버에 있으면 서버의 로컬 접미어에 저장된 데이터만 내보낼 수 있습니다. LDIF 파일이 서버의 원격 시스템에 있으면 접미어와 연결 접미어를 모두 내보낼 수 있습니다.

서버가 실행되는 동안 디렉토리 데이터를 Directory Server 콘솔에서 LDIF로 내보내려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 태스크 탭에서 아래쪽으로 스크롤하여 "LDIF로 파일 내보내기" 옆에 있는 버튼을 누릅니다.

내보내기 대화 상자가 표시됩니다.

2. 서버의 원격 시스템에서 콘솔을 실행 중이면 LDIF 파일 필드 아래에 두 개의 라디오 버튼이 표시됩니다. 콘솔을 실행하는 시스템에 있는 LDIF 파일로 내보내기를 나타내려면 "콘솔 시스템"을 선택합니다. 서버 시스템에 있는 LDIF 파일로 내보내기를 나타내려면 "서버 시스템"을 선택합니다.

3. "LDIF 파일" 필드에 LDIF 파일의 전체 경로와 파일 이름을 입력하거나 찾아보기를 눌러 해당 파일을 찾습니다.

"서버 시스템"을 선택할 경우 찾아보기 버튼이 비활성화됩니다. 찾아보기 버튼이 활성화되지 않을 경우 이 파일은 기본적으로 아래 디렉토리에 저장되어 있습니다.

`ServerRoot/slapd-serverID/ldif`

4. 전체 디렉토리를 내보내려면 "모든 접미어" 라디오 버튼을 선택합니다.

디렉토리의 개별 하위 트리만 내보내려면 "하위 트리" 라디오 버튼을 선택하고 텍스트 상자에 하위 트리의 기본 항목 DN을 입력합니다.

또는 찾아보기를 눌러 하위 트리를 선택할 수도 있습니다.

5. 확인을 눌러 디렉토리 내용을 파일로 내보냅니다.

콘솔에서 개별 접미어를 LDIF로 내보내기

서버가 실행되는 동안 접미어를 Directory Server 콘솔에서 LDIF로 내보내려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드를 확장하여 내보낼 접미어를 표시합니다.
2. 접미어 노드를 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 내보내기를 선택합니다. 또는 접미어 노드를 선택하고 객체 메뉴에서 내보내기를 선택할 수도 있습니다.
접미어 내보내기 대화 상자가 표시됩니다.
3. "LDIF 파일" 필드에 LDIF 파일의 전체 경로를 입력하거나 찾아보기를 눌러 사용자 시스템에서 해당 파일을 찾습니다.
찾아보기 버튼이 활성화되지 않을 경우 이 파일은 기본적으로 아래 디렉토리에 저장되어 있습니다.
`ServerRoot/slaped-serverID/ldif`
4. 접미어가 복제되면 복제 정보를 내보냅니다. 확인란을 선택할 수 있습니다. 이 기능은 내보낸 LDIF를 사용하여 이 접미어의 다른 복제본을 초기화할 경우에만 필요합니다.
5. 이 접미어에 속성 암호화가 활성화되어 있으면 속성 비밀번호 해독 확인란을 선택할 수 있습니다. 이렇게 하려면 서버의 인증서 데이터베이스를 보호하는 비밀번호를 제공해야 합니다. 비밀번호를 입력하거나 비밀번호가 포함된 파일 이름을 입력하는 옵션을 선택합니다. 속성 값의 비밀번호를 해독하는 비밀번호를 제공하지 않으면 LDIF 출력에 암호화된 값이 표시됩니다.
6. 확인을 눌러 접미어 내용을 파일로 내보냅니다.

명령줄에서 LDIF로 내보내기

`directoryserver db2ldif` 명령을 사용하여 디렉토리의 모든 접미어 또는 하위 트리를 LDIF로 내보낼 수 있습니다. 이 스크립트는 서버의 실행 여부에 관계 없이 접미어 내용의 전체 또는 일부를 내보냅니다.

데이터베이스 내용을 LDIF 파일로 내보내려면 아래 명령을 실행합니다.

```
# /usr/sbin/directoryserver -s serverID db2ldif ...
```

아래 예에서는 두 개의 접미어를 LDIF 파일 하나로 내보냅니다.

```
/usr/sbin/directoryserver -s example db2ldif \
-a /var/opt/mps/serverroot/slaped-example/output.ldif \
-s "dc=example,dc=com" -s "o=NetscapeRoot"
```

아래 표에는 예에 사용된 db2ldif 옵션이 설명되어 있습니다.

표 4-4 위의 예에 사용된 db2ldif 옵션에 대한 설명

옵션	설명
-a	서버에서 내보낸 LDIF를 저장하는 출력 파일의 이름을 정의합니다. 이 파일은 기본적으로 <i>ServerRoot/slaped-serverID</i> 디렉토리에 저장됩니다.
-s	내보내기에 포함할 접미어 또는 하위 트리를 지정합니다. -s 인수를 여러 개 사용하여 여러 접미어 또는 하위 트리를 지정할 수 있습니다.

db2ldif 명령을 -r 옵션과 함께 사용하여 복제된 접미어를 LDIF 파일로 내보낼 수도 있습니다. 결과로 작성된 LDIF에는 복제 기법에서 사용하는 속성 하위 유형이 포함되어 있습니다. 그런 다음 [352페이지의 "복제본 초기화"](#)에 설명된 것처럼 소비자 서버에서 이 LDIF 파일을 가져와 소비자 복제본을 초기화할 수 있습니다.

db2ldif 명령을 -r 옵션과 함께 사용하는 경우 서버가 실행 중이면 안 됩니다. 먼저 서버를 중지한 다음 명령을 시작하거나 서버를 중지할 필요가 없는 db2ldif.pl 스크립트를 -r 옵션과 함께 사용해야 합니다.

이 스크립트의 사용에 대한 자세한 내용은 *Directory Server Man Page Reference*를 참조하십시오.

데이터 백업

데이터 백업은 나중에 데이터베이스 파일이 손상되거나 삭제될 경우에 대비하여 디렉토리 내용의 스냅샷을 저장합니다. Directory Server 콘솔이나 명령줄 스크립트를 사용하여 접미어를 백업할 수 있습니다.

주의 백업 작업 중에는 서버를 중지하지 마십시오.

기본적으로 여기에 설명된 모든 백업 절차는 서버 파일의 복사본을 동일한 호스트에 저장합니다. 보안을 강화하려면 이 백업을 복사하여 다른 시스템이나 파일 시스템에 저장해야 합니다.

주 이러한 백업 방법으로 원격 서버에 있는 연결 접미어를 백업할 수는 없습니다. 이 경우 각각의 서버를 별도로 백업해야 합니다.

콘솔에서 서버 백업

Directory Server 콘솔에서 서버를 백업하는 경우 서버는 모든 데이터베이스 내용 및 관련 색인 파일을 백업 위치에 복사합니다. 서버가 실행되는 동안 백업을 수행할 수 있습니다.

콘솔에서 서버를 백업하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 태스크 탭에서 "디렉토리 서버 백업" 옆에 있는 버튼을 누릅니다.

디렉토리 서버 백업 대화 상자가 표시됩니다.

2. 디렉토리 텍스트 상자에 백업을 저장할 디렉토리의 전체 경로를 입력합니다. 디렉토리와 콘솔을 동일한 시스템에서 실행하는 경우 찾아보기를 눌러 로컬 디렉토리를 찾습니다.

또는 "기본값 사용"을 눌러 아래 디렉토리에 백업을 저장합니다.

`ServerRoot/slapd-serverID/bak/YYYY_MM_DD_lh_mm_ss`

여기서 `serverID`는 디렉토리 서버의 이름이며, 디렉토리 이름은 백업이 작성된 날짜와 시간을 사용하여 생성됩니다.

3. 확인을 눌러 백업을 작성합니다.

명령줄에서 서버 백업

`directoryserver db2bak` 명령을 사용하여 명령줄에서 서버를 백업할 수 있습니다. 이 명령은 서버의 실행 여부에 관계 없이 작동합니다.

이 백업 방법으로 구성 정보를 백업할 수는 없습니다. 구성 정보 백업에 대한 자세한 내용은 [199페이지의 "dse.ldif 구성 파일 백업"](#)을 참조하십시오.

디렉토리를 백업하려면 아래 명령을 실행합니다.

```
# /usr/sbin/directoryserver -s serverID db2bak backupDir
```

backupDir 매개 변수는 백업을 저장할 디렉토리를 지정합니다. 기본 백업 디렉토리 이름은 현재 날짜인 *YYYY_MM_DD_hh_mm_ss*로 만들어집니다. 이 명령의 사용에 대한 자세한 내용은 *Directory Server Man Page Reference*를 참조하십시오. 배포를 위한 백업 전략 설계에 대한 자세한 내용은 *Directory Server Deployment Planning Guide*를 참조하십시오.

dse.ldif 구성 파일 백업

서버를 복원할 때 *dse.ldif* 구성 파일에는 해당 서버를 백업했을 당시와 동일한 구성이 있어야 합니다. 이렇게 하려면 서버의 백업을 만들 때마다 안전한 장소에 *dse.ldif* 구성 파일의 복사본을 저장하여 백업합니다.

아울러, 다음 작업을 수행할 때 Directory Server는 자동으로 *dse.ldif* 구성 파일을 *ServerRoot/slapd-serverID/config* 디렉토리에 백업합니다.

- Directory Server를 시작하면 *dse.ldif.startOK* 파일에 자동으로 *dse.ldif* 파일의 백업이 작성됩니다.
- *cn=config* 분기가 수정되면 서버는 먼저 이 파일을 *config* 디렉토리의 *dse.ldif.bak* 파일에 백업한 다음 *dse.ldif* 파일에 수정 사항을 씁니다.

백업을 사용한 데이터 복원

아래 절차에서는 Directory Server 콘솔이나 명령줄을 사용하여 디렉토리에 있는 접미어를 복원하는 방법에 대해 설명합니다. 서버는 [197페이지의 "데이터 백업"](#)에 설명된 것처럼 백업되어 있어야 합니다. 복제 계약에 사용된 접미어를 복원하는 경우 먼저 [200페이지의 "복제된 접미어 복원"](#)을 읽어 보십시오.

주의 백업 또는 복원 작업 중에는 서버를 중지하지 마십시오.

서버를 복원하면 기존의 모든 데이터베이스 파일을 덮어쓰므로 백업 이후의 데이터 변경 사항이 모두 손실됩니다.

복제된 접미어 복원

공급자 서버와 소비자 서버 간에 복제된 접미어를 복원하려면 몇 가지 주의해야 할 사항이 있습니다. 가능하면 백업을 사용하여 접미어를 복원하는 대신 복제 기법을 통해 접미어를 업데이트해야 합니다.

공급자나 허브 인스턴스를 복원할 때 서버 구성은 백업을 만들었을 당시의 구성과 같아야 합니다. 이렇게 하려면 Directory Server 데이터를 복원하기 전에 먼저 dse.ldif를 복원해야 합니다. [204페이지의 "dse.ldif 구성 파일 복원"](#)을 참조하십시오.

이 절에서는 복제본의 복원 방법과 시기, 그리고 작업 후에 다른 복제본과 동기화하는 방법에 대해 설명합니다. 백업 및 복원 기능을 사용하여 복제본을 초기화하는 방법은 [350페이지의 "복제본 초기화"](#)를 참조하십시오.

단일 마스터 시나리오에서 공급자 복원

단일 마스터 공급자로 설정된 접미어에는 전체 복제 토폴로지에 대한 신뢰할 수 있는 데이터가 저장되어 있습니다. 따라서 이 접미어를 복원하면 전체 토폴로지의 모든 데이터를 다시 초기화하는 것과 같습니다. 복원할 백업 내용을 사용하여 모든 데이터를 다시 초기화하려는 경우에만 단일 마스터를 복원해야 합니다.

오류가 발생하여 단일 마스터 데이터를 복원할 수 없는 경우, 소비자 데이터가 백업보다 최신 버전일 수 있으므로 소비자 중 하나에 있는 데이터를 사용하는 방법을 고려해 볼 수 있습니다. 이 경우 소비자 복제본의 데이터를 LDIF 파일로 내보낸 다음 이 LDIF 파일을 사용하여 마스터를 다시 초기화해야 합니다.

마스터 복제본에서 백업을 복원하거나 LDIF 파일을 가져온 경우 이 복제본에서 업데이트를 받는 모든 허브와 소비자 복제본을 다시 초기화해야 합니다. 공급자 서버의 로그 파일에 소비자를 다시 초기화해야 한다는 메시지가 기록됩니다.

다중 마스터 시나리오에서 공급자 복원

다중 마스터 복제에서는 복제된 데이터의 신뢰할 수 있는 복사본이 각각의 마스터에 저장되어 있습니다. 현재의 복제본 내용으로 업데이트되면서 만료된 이전 백업은 복원할 수 없습니다. 가능하면 복제 기법에서 다른 마스터의 내용을 사용하여 마스터를 최신 상태로 유지할 수 있도록 해야 합니다.

이렇게 할 수 없는 경우에만 다음 방법 중 하나로 다중 마스터 복제본을 복원해야 합니다.

- 가장 간단한 방법은 백업을 복원하지 않고 다른 마스터 중 하나를 사용하여 해당 마스터를 다시 초기화하는 것입니다. 이렇게 하면 최신 데이터가 해당 마스터로 전송되어 복제할 수 있도록 준비됩니다. [357페이지의 "콘솔에서 복제본 초기화"](#) 또는 [358페이지의 "명령줄에서 복제본 초기화"](#)를 참조하십시오.
- 수백만 개의 항목이 있는 복제본의 경우, 새로운 이진 복사 기능을 사용하면 다른 마스터 중 하나에서 받은 최신 백업을 보다 신속하게 복원할 수 있습니다. [360페이지의 "이진 복사를 사용한 복제본 초기화"](#)를 참조하십시오.
- 마스터의 백업이 다른 마스터 중 *어느 하나*에 있는 변경 로그 내용의 최대 수명보다 오래되지 않은 경우 이 백업을 사용하여 마스터를 복원할 수 있습니다. 변경 로그 수명에 대한 자세한 내용은 [345페이지의 "고급 다중 마스터 구성"](#)을 참조하십시오. 이전 백업을 복원하면 다른 마스터는 자신의 변경 로그를 사용하여 백업이 저장된 이후에 처리된 모든 수정 사항을 마스터에 업데이트합니다.

복원 또는 다시 초기화하는 방법에 관계 없이 마스터 복제본은 초기화 후에 읽기 전용 모드로 남아 있습니다. [354페이지의 "다중 마스터 초기화 후의 수렴"](#)에 설명된 것처럼 복제본은 이 동작을 통해 다른 마스터와 동기화할 수 있으며, 그 후에 쓰기 작업이 허용됩니다.

모든 복제본이 동기화된 후에 복원 또는 다시 초기화된 마스터에 대한 쓰기 작업을 허용하는 경우 허브나 소비자 서버를 다시 초기화할 필요가 없다는 이점이 있습니다.

허브 복원

이 절의 내용은, 예를 들어 데이터베이스 파일이 손상되었거나 복제가 장시간 중단된 경우와 같이 복제 기법에서 자동으로 허브 복제본을 최신 상태로 유지할 수 없는 경우에만 적용됩니다. 이 경우 다음 방법 중 하나를 사용하여 허브 복제본을 복원하거나 다시 초기화해야 합니다.

- 가장 간단한 방법은 백업을 복원하지 않고 마스터 복제본 중 하나를 사용하여 허브를 다시 초기화하는 것입니다. 이렇게 하면 최신 데이터가 허브로 전송되어 바로 복제할 수 있는 상태가 됩니다. [357페이지의 "콘솔에서 복제본 초기화"](#) 또는 [358페이지의 "명령줄에서 복제본 초기화"](#)를 참조하십시오.
- 수백만 개의 항목이 있는 복제본의 경우, 새로운 이진 복사 기능을 사용하면 다른 허브 복제본에서 받은 최신 백업을 보다 신속하게 복원할 수 있습니다. [360페이지의 "이진 복사를 사용한 복제본 초기화"](#)를 참조하십시오. 복사할 다른 허브 복제본이 없을 경우, 가능하면 앞 단락에 설명된 것처럼 허브를 다시 초기화하거나 다음 단락에 설명된 것처럼 허브를 복원해야 합니다.

- 허브의 백업이 해당 공급자 중 *어느 하나*(허브 또는 마스터 복제본)에 있는 변경 로그 내용의 최대 수명보다 오래되지 않은 경우 이 백업을 사용하여 허브를 복원할 수 있습니다. 변경 로그 수명에 대한 자세한 내용은 [345페이지의 "고급 다중 마스터 구성"](#)을 참조하십시오. 이전 백업을 복원하면 해당 공급자는 자신의 변경 로그를 사용하여 백업이 저장된 이후에 처리된 모든 수정 사항을 허브에 업데이트합니다.

주 허브 복제본을 복원 또는 다시 초기화하는 방법에 관계 없이 다른 모든 수준의 허브를 비롯한 이 허브의 모든 소비자를 *받드시* 다시 초기화해야 합니다.

전용 소비자 복원

이 절의 내용은, 예를 들어 데이터베이스 파일이 손상되었거나 복제가 장시간 중단된 경우와 같이 복제 기법에서 자동으로 전용 소비자 복제본을 최신 상태로 유지할 수 없는 경우에만 적용됩니다. 이 경우 다음 방법 중 하나를 사용하여 소비자를 복원하거나 다시 초기화해야 합니다.

- 가장 간단한 방법은 백업을 복원하지 않고 공급자(마스터 또는 허브 복제본) 중 하나를 사용하여 소비자를 다시 초기화하는 것입니다. 이렇게 하면 최신 데이터가 소비자 로 전송되어 바로 복제할 수 있는 상태가 됩니다. [357페이지의 "콘솔에서 복제본 초기화"](#) 또는 [358페이지의 "명령줄에서 복제본 초기화"](#)를 참조하십시오.
- 수백만 개의 항목이 있는 복제본의 경우, 새로운 이진 복사 기능을 사용하면 다른 소비자 복제본에서 받은 최신 백업을 보다 신속하게 복원할 수 있습니다. [360페이지의 "이진 복사를 사용한 복제본 초기화"](#)를 참조하십시오. 복사할 다른 소비자가 없을 경우, 가능하면 앞 단락에 설명된 것처럼 허브를 다시 초기화하거나 다음 단락에 설명된 것처럼 복제본을 복원해야 합니다.
- 소비자의 백업이 해당 공급자 중 *어느 하나*(허브 또는 마스터 복제본)에 있는 변경 로그 내용의 최대 수명보다 오래되지 않은 경우 이 백업을 사용하여 소비자를 복원할 수 있습니다. 변경 로그 수명에 대한 자세한 내용은 [345페이지의 "고급 다중 마스터 구성"](#)을 참조하십시오. 이전 백업을 복원하면 해당 공급자는 자신의 변경 로그를 사용하여 백업이 저장된 이후에 처리된 모든 수정 사항을 허브에 업데이트합니다.

콘솔에서 서버 복원

Directory Server 콘솔을 사용하여 이전에 생성된 백업에서 손상된 디렉토리 데이터를 복원할 수 있습니다. 콘솔을 사용하여 데이터를 복원하려면 Directory Server가 실행 중이어야 합니다. 그러나 복원 중에는 해당 접미어를 사용하여 작업을 처리할 수 없습니다.

이전에 작성한 백업을 사용하여 서버를 복원하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 태스크 탭에서 "디렉토리 서버 복원" 옆에 있는 버튼을 누릅니다.
디렉토리 서버 복원 대화 상자가 표시됩니다.
2. 사용 가능한 백업 목록에서 백업을 선택하거나 "디렉토리" 텍스트 상자에 유효한 백업의 전체 경로를 입력합니다.
사용 가능한 백업 목록에는 아래의 기본 디렉토리에 있는 모든 백업이 표시됩니다.
`ServerRoot/slapd-serverID/bak`
3. 확인을 눌러 서버를 복원합니다.

명령줄에서 서버 복원

다음과 같은 스크립트를 사용하여 명령줄에서 서버를 복원할 수 있습니다.

- `directoryserver bak2db` 명령 사용. 이 명령을 사용하려면 서버를 종료해야 합니다.
- `directoryserver bak2db-task` 명령 사용. 이 명령을 사용하려면 서버를 실행해야 합니다.

bak2db 명령 사용

서버가 종료된 동안 명령줄에서 디렉토리를 복원하려면 다음을 수행합니다.

1. 명령줄에서 루트로 아래 명령을 실행하여 서버를 중지합니다.
`# /usr/sbin/directoryserver -s serverID stop`
2. 백업 디렉토리의 전체 경로를 사용하여 bak2db 명령을 실행합니다.
`# /usr/sbin/directoryserver -s serverID bak2db backupDir`
3. 다음과 같이 서버를 시작합니다.
`# /usr/sbin/directoryserver -s serverID start`

아래 예에서는 기본 백업 디렉토리의 백업을 복원합니다.

```
/usr/sbin/directoryserver -s example bak2db \  
/var/opt/mps/serverroot/slaped-example/bak/2003_07_01_11_34_00
```

bak2db 명령에 대한 자세한 내용은 *Directory Server Man Page Reference*를 참조하십시오.

bak2db-task 사용

서버가 실행되는 동안 명령줄에서 디렉토리를 복원하려면 다음 명령을 사용합니다.

```
# /usr/sbin/directoryserver -s serverID bak2db-task ...
```

다음 예에서는 bak2db-task 명령을 사용하여 백업을 복원합니다. -a 옵션은 백업 디렉토리의 전체 경로를 제공합니다.

```
/usr/sbin/directoryserver -s example bak2db-task\  
-D "cn=Directory Manager" -w password \  
-a /var/opt/mps/serverroot/slaped-example/bak/2003_07_01_11_34_00
```

자세한 내용은 *Directory Server Man Page Reference*를 참조하십시오.

dse.ldif 구성 파일 복원

디렉토리는 아래 디렉토리에 dse.ldif 파일의 백업 복사본 두 개를 작성합니다.

```
ServerRoot/slaped-serverID/config
```

dse.ldif.startOK 파일은 서버를 시작할 때 dse.ldif 파일의 복사본을 기록합니다. dse.ldif.bak 파일에는 dse.ldif 파일에 대한 최신 변경 사항의 백업이 저장되어 있습니다. 최신 변경 사항이 저장된 파일을 디렉토리에 복사합니다.

dse.ldif 구성 파일을 복원하려면 다음을 수행합니다.

1. 명령줄에서 루트로 아래 명령을 실행하여 서버를 중지합니다.
/usr/sbin/directoryserver -s serverID stop
2. 구성 파일이 포함된 다음과 같은 디렉토리로 변경합니다.
cd /var/mps/serverroot/slaped-serverID/config
3. 유효한 백업 구성 파일을 사용하여 dse.ldif 파일을 덮어씁니다. 예를 들어 다음과 같이 입력할 수 있습니다.
cp dse.ldif.startOK dse.ldif

4. 다음 명령을 실행하여 서버를 시작합니다.

```
# /usr/sbin/directoryserver -s serverID start
```

백업을 사용한 데이터 복원

아이디 및 역할 관리

디렉토리의 계층적 데이터 구조에 제한 받지 않고 자유롭게 사용자 항목을 관리하기 위해 그룹을 작성하여 공통 속성 값을 공유해야 하는 경우가 있습니다. Directory Server는 그룹, 역할 및 서비스 클래스(CoS)를 통해 이러한 고급 항목 관리 기능을 제공합니다.

그룹은 구성원 목록이나 구성원 필터로 다른 항목의 이름을 지정하는 항목입니다. 역할도 특정 역할의 각 구성원에 `nsrole` 속성을 생성하는 기법을 통해 동일한 기능을 제공하지만, 훨씬 더 기능이 다양합니다. CoS는 가상 속성을 생성함으로써 각 항목에 속성 값을 저장할 필요 없이 여러 항목이 공통 속성 값을 공유할 수 있게 합니다.

Directory Server에서는 역할 및 CoS 가상 속성 값에 따라 검색을 수행할 수 있는 새 기능을 소개합니다. `nsRole` 속성이나 CoS 정의에서 생성된 속성이 포함된 필터 문자열을 작업에 사용하고 이 속성 값에 대한 비교 작업을 수행할 수 있습니다. 그러나 가상 CoS 속성은 색인화할 수 없으므로 CoS에서 생성된 속성을 사용하는 검색도 색인화되지 않습니다. 시간과 메모리 자원을 많이 소비할 수 있습니다.

역할과 서비스 클래스에서 제공하는 기능을 최대한 활용하기 위해 디렉토리 토폴로지는 디렉토리 배포의 계획 단계에서 결정해야 합니다. 이러한 기법 및 기법이 토폴로지를 단순화하는 방법에 대한 설명은 *Directory Server Deployment Planning Guide*를 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

- 그룹 관리
- 역할 할당
- 서비스 클래스(CoS) 정의

그룹 관리

그룹을 사용하면 ACI 정의와 같은 관리 작업의 편의를 위한 항목을 연결할 수 있습니다. 그룹 정의는 해당 구성원의 이름을 정적 목록으로 지정하거나 동적 항목 집합을 정의하는 필터를 제공하는 특수 항목입니다.

그룹 정의 항목이 저장된 위치에 관계 없이 전체 디렉토리가 그룹 구성원이 될 수 있습니다. 관리를 간소화하기 위해 모든 그룹 정의 항목은 대체로 한 위치(루트 접미어 아래의 `ou=Groups`)에 저장됩니다.

정적 그룹을 정의하는 항목은 `groupOfNames` 또는 `groupOfUniqueNames` 객체 클래스에서 상속됩니다. 그룹 구성원은 DN에 따라 `member` 또는 `uniqueMember` 속성의 여러 값으로 나열됩니다.

동적 그룹을 정의하는 항목은 `groupOfURLs` 객체 클래스에서 상속됩니다. 그룹 구성원은 여러 값을 갖는 `memberURL` 속성에 지정된 하나 이상의 필터로 정의됩니다. 동적 그룹의 구성원은 각각의 평가에서 필터 중 하나에 일치하는 항목입니다.

다음 절에서는 콘솔을 사용하여 정적 그룹과 동적 그룹을 작성 및 수정하는 방법에 대해 설명합니다.

새 정적 그룹 추가

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 새 그룹을 추가할 디렉토리 트리 항목을 마우스 오른쪽 버튼으로 누르고 새로 만들기 > 그룹 항목을 선택합니다.
또는 항목을 선택하고 객체 메뉴에서 새로 만들기 > 그룹 항목을 선택합니다.
2. 새 그룹 만들기 대화 상자에서 "그룹 이름" 필드에 새 그룹의 이름을 입력해야 하며, 선택 사항으로 "설명" 필드에 그룹에 대한 설명을 추가할 수 있습니다. 그룹 이름은 새 그룹 항목의 `cn`(일반 이름) 속성 값이 되며 해당 DN에 표시됩니다.
3. 대화 상자의 왼쪽 목록에서 구성원을 누릅니다. 오른쪽 패널에는 기본적으로 정적 그룹 탭이 선택되어 있습니다.
4. 추가를 눌러 새 구성원을 그룹에 추가합니다. 표준 "사용자 및 그룹 검색" 대화 상자가 표시됩니다.

5. 검색 드롭다운 목록에서 사용자를 선택하고 검색할 문자열을 입력한 다음 검색을 누릅니다. 특정 속성이나 특정 속성 값을 검색하려면 고급 버튼을 누릅니다.
결과로 표시된 항목을 하나 이상 선택하고 확인을 누릅니다. 이 단계를 반복하여 이 정적 그룹에 추가할 모든 구성원을 추가합니다.

주 정적 그룹 구성원은 원격으로 연결될 수도 있습니다. 참조 무결성 플러그인을 사용하여 삭제된 구성원 항목이 정적 그룹 항목에서 자동으로 삭제되도록 설정할 수 있습니다. 연결에서 참조 무결성을 사용하는 방법은 [168페이지의 "연결 정책 구성"](#)을 참조하십시오.

6. 다른 언어로 작성된 이름과 설명 문자열을 그룹에 제공하려면 왼쪽 목록에서 언어를 누릅니다. 이러한 문자열은 콘솔에서 해당 로케일을 사용할 때 표시됩니다.
7. 확인을 눌러 새 그룹을 작성합니다. 새 그룹은 그룹이 작성된 위치에 해당하는 항목의 자식 중 하나로 표시됩니다.

새 동적 그룹 추가

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 새 그룹을 추가할 디렉토리 트리 항목을 마우스 오른쪽 버튼으로 누르고 새로 만들기 > 그룹 항목을 선택합니다.
또는 항목을 선택하고 객체 메뉴에서 새로 만들기 > 그룹 항목을 선택합니다.
2. 새 그룹 만들기 대화 상자의 "그룹 이름" 필드에 새 그룹의 이름을 입력합니다. "설명" 필드에 그룹의 설명(선택 사항)을 추가할 수 있습니다. 그룹 이름은 새 그룹 항목의 cn(일반 이름) 속성 값이 되며 해당 DN에 표시됩니다.
3. 대화 상자의 왼쪽 목록에서 구성원을 누르고 오른쪽 패널에서 동적 그룹 탭을 선택합니다.
4. 추가를 눌러 그룹 구성원을 정의할 필터 문자열이 있는 LDAP URL을 작성합니다. 표준 "LDAP URL 생성 및 테스트" 대화 상자가 표시됩니다.

5. 텍스트 필드에 LDAP URL을 입력하거나 구성을 선택하여 그룹용 필터가 있는 LDAP URL의 구성을 도와줄 대화 상자를 표시합니다. 테스트를 눌러 이 필터에서 반환되는 항목 목록을 확인합니다.
URL이 구성되면 확인을 누릅니다. 이 단계를 반복하여 동적 그룹 정의 필터가 있는 모든 URL을 추가합니다.
6. 다른 언어로 작성된 이름과 설명 문자열을 그룹에 제공하려면 왼쪽 목록에서 언어를 누릅니다. 이러한 문자열은 콘솔에서 해당 로케일을 사용할 때 표시됩니다.
7. 확인을 눌러 새 그룹을 작성합니다. 새 그룹은 그룹이 작성된 위치에 해당하는 항목의 자식 중 하나로 표시됩니다.

그룹 정의 수정

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 수정할 그룹을 나타내는 항목을 두 번 누릅니다.
또는 항목을 선택하고 객체 메뉴에서 열기를 선택합니다.
2. 항목 편집 대화 상자에서 일반, 구성원 또는 언어 범주에 있는 그룹 정보를 원하는 대로 수정합니다. 정적 그룹의 구성원을 추가 또는 제거하거나 동적 그룹용 필터가 있는 URL을 추가, 편집 또는 제거할 수 있습니다.
3. 그룹 정의 수정이 끝나면 확인을 누릅니다.
콘솔에서 변경 사항을 확인하려면 보기 메뉴에서 갱신을 선택합니다.

그룹 정의 제거

그룹을 정의하는 항목을 삭제하면 어떤 유형의 그룹이든 제거할 수 있습니다.

역할 할당

역할은 응용 프로그램에서 보다 편리하고 효율적으로 사용할 수 있도록 설계된 대체 그룹화 기법입니다. 역할은 그룹과 유사하게 정의 및 관리되지만 구성원 항목에도 참여하는 역할을 나타내는 생성된 속성이 있습니다. 예를 들어, 응용 프로그램은 그룹을 선택하여 구성원 목록을 탐색할 필요 없이 간단하게 항목의 역할을 읽을 수 있습니다.

기본적으로 역할의 범위는 해당 역할이 정의된 하위 트리로 제한됩니다. Directory Server 5.2에서는 중첩된 역할의 확장 범위를 지원하므로 다른 하위 트리에 있는 역할을 중첩하고 디렉토리의 어느 곳이든 구성원을 가질 수 있습니다. 역할 범위의 확장에 대한 자세한 내용은 [219페이지의 "중첩된 역할 정의 예"](#)를 참조하십시오.

역할

각각의 역할에는 해당 역할을 소유하는 항목인 **구성원**이 있습니다. 디렉토리에서 항목이 검색되면 역할 기법은 역할의 구성원인 모든 항목에 자동으로 nsRole 속성을 생성합니다. 여러 값을 갖는 이 속성에는 해당 항목이 구성원으로 속해 있는 모든 역할 정의 DN이 포함됩니다. nsRole 속성은 항목에 저장되지 않고 작업 결과의 일반 속성으로 클라이언트 응용 프로그램에 반환되는 계산된 속성입니다.

Directory Server는 다음 세 가지 유형의 역할을 지원합니다.

- 관리된 역할 - 관리자가 원하는 구성원 항목에 nsRoleDN 속성을 추가하여 관리된 역할을 할당합니다. 이 속성 값은 역할 정의 항목의 DN입니다. 관리된 역할은 구성원이 역할 정의 항목이 아닌 각 항목에 정의된다는 점만 제외하고 정적 그룹과 유사합니다.
- 필터링된 역할 - 동적 그룹에 해당합니다. nsRoleFilter 속성에 필터 문자열을 정의합니다. 필터링된 역할의 범위는 해당 정의 항목의 부모에서 시작하는, 현재 위치해 있는 하위 트리입니다. 서버가 필터링된 역할 범위에서 해당 필터에 일치하는 항목을 반환하면 이 역할을 식별하는 nsRole 속성이 생성되어 항목에 추가됩니다.
- 중첩된 역할 - 이 역할은 다른 중첩된 역할을 비롯한 다른 역할 정의의 이름을 지정합니다. 중첩된 역할의 구성원 집합은 포함된 역할에 지정된 모든 구성원의 합집합입니다. 중첩된 역할은 확장 범위를 정의하여 다른 하위 트리에 있는 역할의 구성원을 포함할 수도 있습니다.

역할을 사용하면 클라이언트 응용 프로그램이 항목의 nsRole 속성을 직접 읽어 이 항목의 모든 역할 구성원을 확인할 수 있습니다. 따라서 클라이언트 처리가 간소화되어 디렉토리 사용을 최적화할 수 있습니다. 역할을 CoS 기법과 함께 사용하여 역할 구성원의 다른 속성을 생성할 수 있습니다([234페이지의 "역할 기반의 속성 작성"](#) 참조). 역할은 액세스 제어 정의에 사용할 수 있으며([260페이지의 "역할 액세스 정의 - roledn 키워드"](#) 참조), 다른 모든 구성원을 동시에 활성화 또는 비활성화하는 등의 기타 기능도 지원합니다([326페이지의 "사용자와 역할 비활성화 및 활성화"](#) 참조).

nsRole 속성 검색

Directory Server에서는 모든 검색 필터에 nsRole 속성을 사용할 수 있습니다. 비교 연산자를 사용하여 이 속성의 특정 값을 검색할 수도 있지만 nsRole 속성을 검색할 때는 다음 사항을 고려해야 합니다.

- nsRole 속성을 사용한 검색은 항목을 필터링하기 전에 모든 역할을 평가해야 하므로 많은 시간이 소요될 수 있습니다.
- Directory Server는 관리된 역할의 특정 구성원에 대한 동일 검색에 가장 적합합니다. 예를 들어 다음과 같은 검색은 실제 속성에 대한 검색만큼 속도가 빠릅니다.

```
(&(objectclass=person)
(nsRole=cn=managersRole,ou=People,dc=example,dc=com))
```
- 관리된 역할 구성원의 정의에 사용된 nsRoleDN 속성은 기본적으로 모든 접미어에서 색인화됩니다. 이 속성에 대한 색인화를 비활성화하면 관리된 역할 구성원의 검색 최적화 이점이 사라집니다.
- 필터링된 역할이 포함된 항목을 검색하려면 역할 필터를 사용한 내부 검색이 필요합니다. 내부 작업은 역할 필터에 있는 모든 속성이 역할 범위의 모든 접미어에서 색인화되어 있을 때 가장 속도가 빠릅니다.

nsRole 속성에 대한 권한

nsRole 속성은 역할 기법에서만 지정되고 디렉토리 사용자가 쓰거나 수정할 수 없지만 다음과 같은 점에 주의해야 합니다.

- nsRole 속성은 잠재적으로 모든 디렉토리 사용자가 읽을 수 있지만, 액세스 제어를 정의하여 읽을 수 없도록 차단할 수 있습니다.
- nsRoleDN 속성은 관리된 역할 구성원을 정의하며, 사용자가 자신을 역할에 추가하거나 제거할 수 있도록 할지 지정해야 합니다. 사용자가 자신의 역할을 수정할 수 없도록 차단하는 ACI에 대해서는 [217페이지의 "관리된 역할 정의 예"](#)를 참조하십시오.
- 필터링된 역할은 사용자 항목에 있는 속성이나 속성 값에 기반을 둔 필터를 통해 구성원을 지정합니다. 필터링된 역할의 구성원을 정의할 수 있는 사람을 제어할 수 있도록, 이러한 속성의 사용자 권한은 신중하게 정의해야 합니다.

디렉토리의 역할을 사용하는 방법에 대한 자세한 내용은 *Directory Server Deployment Planning Guide*를 참조하십시오.

콘솔에서 역할 할당

이 절에서는 Directory Server 콘솔을 통해 역할을 만들고 수정하는 절차에 대해 설명합니다.

관리된 역할 만들기

관리된 역할에는 역할 정의 항목이 있으며 구성원은 각 구성원 항목에 nsRoleDN 속성을 추가하여 지정합니다. 콘솔을 사용하여 관리된 역할에 구성원을 작성 및 추가하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 새 역할 정의를 추가할 디렉토리 트리 항목을 마우스 오른쪽 버튼으로 누르고 새로 만들기 > 역할 항목을 선택합니다.
또는 항목을 선택하고 객체 메뉴에서 새로 만들기 > 역할 항목을 선택합니다.
2. 새 역할 만들기 대화 상자에서 "역할 이름" 필드에 새 역할의 이름을 입력해야 하며, 선택 사항으로 "설명" 필드에 역할에 대한 설명을 추가할 수 있습니다. 역할 이름은 새 역할 항목의 cn(일반 이름) 속성 값이 되며 해당 DN에 표시됩니다.
3. 대화 상자의 왼쪽 목록에서 구성원을 누릅니다. 오른쪽 패널에는 기본적으로 관리된 역할 라디오 버튼이 선택되어 있습니다.
4. 구성원 목록 아래에 있는 추가를 눌러 새 구성원을 역할에 추가합니다. 표준 "사용자 및 그룹 검색" 대화 상자가 표시됩니다.
5. 검색 드롭다운 목록에서 사용자를 선택하고 검색할 문자열을 입력한 다음 검색을 누릅니다. 특정 속성이나 특정 속성 값을 검색하려면 고급 버튼을 누릅니다.
결과로 표시된 항목을 하나 이상 선택하고 확인을 누릅니다. 이 단계를 반복하여 이 관리된 역할에 추가할 모든 구성원을 추가합니다.
6. 역할에 항목이 모두 추가되면 확인을 누릅니다. 새 역할이 관리된 역할 아이콘과 함께 디렉토리 트리에 표시되고, 모든 구성원 항목에 이 역할 항목의 DN 값이 포함된 nsRoleDN 속성이 추가됩니다.
7. 역할이 작성되면 이 역할의 DN 값이 포함된 nsRoleDN 속성을 항목에 추가하여 역할을 할당할 수도 있습니다.

필터링된 역할 만들기

필터링된 역할의 구성원은 역할 정의의 LDAP 필터에서 선택한 속성 또는 속성 값이 포함된 항목입니다.

주 필터링된 역할의 필터 문자열은 CoS 기법에서 생성된 가상 속성을 제외한 모든 속성에 기반을 둘 수 있습니다(220페이지의 "CoS" 참조).

콘솔을 사용하여 필터링된 역할에 구성원을 작성 및 추가하려면 다음을 수행합니다.

1. **Directory Server** 콘솔의 최상위 디렉토리 탭에서 새 역할 정의를 추가할 디렉토리 트리 항목을 마우스 오른쪽 버튼으로 누르고 새로 만들기 > 역할 항목을 선택합니다.
또는 항목을 선택하고 객체 메뉴에서 새로 만들기 > 역할 항목을 선택합니다.
2. 새 역할 만들기 대화 상자에서 "역할 이름" 필드에 새 역할의 이름을 입력해야 하며, 선택 사항으로 "설명" 필드에 역할에 대한 설명을 추가할 수 있습니다. 역할 이름은 새 역할 항목의 cn(일반 이름) 속성 값이 되며 해당 DN에 표시됩니다.
3. 대화 상자의 왼쪽 목록에서 구성원을 누르고 오른쪽 패널에서 필터링된 역할 라디오 버튼을 선택합니다.
4. 텍스트 상자에 LDAP 필터를 입력하여 역할 구성원을 결정할 필터를 정의합니다. 또는 구성을 눌러 LDAP 필터의 구성을 안내할 대화 상자를 표시합니다.
5. 구성을 누르면 LDAP 필터 구성 대화 상자가 표시됩니다. 필터링된 역할 정의에서는 지정할 수 없으므로 LDAP 서버 호스트, 포트, 기본 DN 및 검색 범위 필드는 무시하십시오.
 - a. 필터링된 역할의 사용자만 검색합니다. 이렇게 하면 필터에 (objectclass=person) 구성 요소가 추가됩니다. 이 구성 요소를 사용하지 않으려면 새 역할 만들기 대화 상자의 텍스트 필드에서 LDAP 필터를 편집해야 합니다.
 - b. "조건" 드롭다운 목록에서 속성을 선택하고 일치 조건을 설정하여 필터를 구체적으로 정의합니다. 다른 필터를 추가하려면 추가를 누릅니다. 불필요한 필터를 제거하려면 삭제를 누릅니다.
 - c. 확인을 눌러 작성된 필터를 필터링된 역할 정의에 사용합니다. 그런 후에 텍스트 필드에서 필터를 편집하여 구성 요소를 수정할 수도 있습니다.
6. 테스트를 눌러 필터를 적용합니다. 테스트 결과 필터링 대화 상자에 현재 필터에 일치하는 항목이 표시됩니다.
7. 확인을 눌러 새 역할 항목을 작성합니다. 새 역할이 필터링된 역할 아이콘과 함께 디렉토리 트리에 표시됩니다.

중첩된 역할 만들기

중첩된 역할을 사용하면 다른 역할을 포함하는 역할을 작성하여 기존 역할의 범위를 확장할 수 있습니다. 중첩된 역할을 작성하려면 다른 역할이 이미 존재해야 합니다. 중첩된 역할을 작성하는 경우 콘솔에 중첩할 수 있는 역할 목록이 표시됩니다. 중첩된 역할에는 다른 중첩된 역할이 최대 30개 중첩 수준까지 포함될 수 있습니다. 이 제한을 초과하면 서버에서 역할을 평가할 때 오류를 기록합니다.

콘솔을 사용하여 중첩된 역할에 구성원을 작성 및 추가하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 새 역할 정의를 추가할 디렉토리 트리 항목을 마우스 오른쪽 버튼으로 누르고 새로 만들기 > 역할 항목을 선택합니다.
또는 항목을 선택하고 객체 메뉴에서 새로 만들기 > 역할 항목을 선택합니다.
2. 새 역할 만들기 대화 상자에서 "역할 이름" 필드에 새 역할의 이름을 입력해야 하며, 선택 사항으로 "설명" 필드에 역할에 대한 설명을 추가할 수 있습니다. 역할 이름은 새 역할 항목의 cn(일반 이름) 속성 값이 되며 해당 DN에 표시됩니다.
3. 대화 상자의 왼쪽 목록에서 구성원을 누르고 오른쪽 패널에서 중첩된 역할 라디오 버튼을 선택합니다.
4. 추가를 눌러 기존 역할을 중첩된 역할 목록에 추가합니다. 역할 선택기 대화 상자의 사용 가능한 역할 목록에서 하나 이상의 역할을 선택하고 확인을 누릅니다.
5. 확인을 눌러 중첩된 역할 항목을 작성합니다. 새 역할이 중첩된 역할 아이콘과 함께 디렉토리에 표시됩니다.
6. 중첩된 역할 범위를 수정하려면 [219페이지의 "중첩된 역할 정의 예"](#)에 설명된 명령줄 절차를 사용합니다.

항목의 역할 보기 및 편집

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 디렉토리 트리를 탐색하여 역할을 보거나 편집하려는 항목을 표시합니다.
2. 항목을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 역할 설정을 선택합니다. 또는 항목을 왼쪽 마우스 버튼으로 눌러 선택한 다음 객체 메뉴에서 역할 설정을 선택할 수도 있습니다.
역할 설정 대화 상자가 표시됩니다.
3. 관리된 역할 탭을 선택하여 이 항목이 속하는 관리된 역할을 표시합니다. 다음과 같은 작업을 수행할 수 있습니다.

- 새 관리된 역할을 추가하려면 추가를 눌러 역할 선택기 창에서 사용 가능한 역할을 선택합니다. 역할 선택기 창에서 확인을 누릅니다.
 - 관리된 역할을 제거하려면 해당 역할을 선택하고 제거를 누릅니다.
 - 항목에 연결된 관리된 역할을 편집하려면 테이블에서 편집할 역할을 선택하고 편집을 누릅니다. 역할의 사용자 정의 편집기에 해당 역할이 표시됩니다. 원하는 대로 역할을 변경하고 확인을 눌러 새 역할 정의를 저장합니다.
4. 기타 역할 탭을 선택하여 이 항목이 속해 있는 필터링된 역할이나 중첩된 역할을 표시합니다. 필터링된 역할이나 중첩된 역할의 역할 구성원을 변경하려면 다음과 같이 역할 정의를 편집해야 합니다.
- 역할을 선택하고 편집을 눌러 역할의 사용자 정의 편집기를 표시합니다. 원하는 대로 역할을 변경하고 확인을 눌러 새 역할 정의를 저장합니다.
5. 역할 수정이 끝나면 확인을 눌러 변경 사항을 저장합니다.

역할 항목 수정

1. Directory Server 콘솔에서 디렉토리 탭을 선택합니다.
2. 탐색 트리를 검색하여 기존 역할의 정의 항목을 찾습니다. 역할은 작성된 위치에 해당하는 항목의 자식입니다. 역할을 두 번 누릅니다.
항목 편집 대화 상자가 표시됩니다.
3. 왼쪽 패널에서 일반을 눌러 역할 이름과 설명을 변경합니다.
4. 왼쪽 패널에서 구성원을 눌러 관리된 역할과 중첩된 역할의 구성원을 변경하거나 필터링된 역할의 필터를 변경합니다.
5. 확인을 눌러 변경 사항을 저장합니다.

역할 삭제

역할을 삭제하면 해당 구성원이 아닌 역할 정의 항목만 삭제됩니다.

역할을 삭제하려면 다음을 수행합니다.

1. Directory Server 콘솔에서 디렉토리 탭을 선택합니다.
2. 탐색 트리를 검색하여 역할의 정의 항목을 찾습니다. 역할은 작성된 위치에 해당하는 항목의 자식입니다.
3. 역할을 마우스 오른쪽 버튼으로 누르고 삭제를 선택합니다.
삭제를 확인하는 대화 상자가 표시됩니다. 예를 누릅니다.
4. 역할이 성공적으로 삭제되었음을 알려주는 항목 삭제 대화 상자가 표시됩니다. 확인을 누릅니다.

주 역할을 삭제하면 역할 항목만 삭제되고 각 역할 구성원의 nsRoleDN 속성은 삭제되지 않습니다. 이 속성을 삭제하려면 참조 무결성 플러그인을 활성화하여 nsRoleDN 속성을 관리하도록 구성합니다. 자세한 내용은 [108 페이지의 "참조 무결성 유지"](#)를 참조하십시오.

명령줄에서 역할 관리

역할은 디렉토리 관리자가 명령줄 유틸리티를 통해 액세스할 수 있는 항목에 정의됩니다. 역할이 작성되면 다음과 같이 구성원을 역할에 지정합니다.

- 관리된 역할의 구성원은 해당 항목에 nsRoleDN 속성이 있습니다.
- 필터링된 역할의 구성원은 nsRoleFilter 속성에 지정된 필터와 일치하는 항목입니다.
- 중첩된 역할의 구성원은 중첩된 역할 정의 항목의 nsRoleDN 속성에 지정된 역할의 구성원입니다.

모든 역할 정의는 LDAPsubentry 및 nsRoleDefinition 객체 클래스로부터 상속됩니다. 아래 표에는 역할 유형별 추가 객체 클래스 및 관련 속성이 나와 있습니다.

표 5-1 역할 정의에 사용되는 객체 클래스 및 속성

역할 유형	객체 클래스	속성
관리된 역할	nsSimpleRoleDefinition nsManagedRoleDefinition	Description(선택 사항)
필터링된 역할	nsComplexRoleDefinition nsFilteredRoleDefinition	nsRoleFilter Description(선택 사항)
중첩된 역할	nsComplexRoleDefinition nsNestedRoleDefinition	nsRoleDN Description(선택 사항)

관리된 역할 정의 예

모든 마케팅 직원에게 할당할 역할을 작성하려면 아래의 ldapmodify 명령을 실행합니다.

```

ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=Marketing,ou=marketing,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsSimpleRoleDefinition
objectclass: nsManagedRoleDefinition
cn: Marketing
description: managed role for marketing staff

```

nsManagedRoleDefinition 객체 클래스는 LDAPsubentry, nsRoleDefinition 및 nsSimpleRoleDefinition 객체 클래스로부터 상속됩니다.

아래의 ldapmodify 명령으로 마케팅 직원 구성원인 Bob의 항목을 업데이트하여 Bob에게 역할을 할당합니다.

```

ldapmodify -h host -p port -D "cn=Directory Manager" -w secret
dn: cn=Bob Arnold,ou=marketing,ou=People,dc=example,dc=com
changetype: modify
add: nsRoleDN
nsRoleDN: cn=Marketing,ou=marketing,ou=People,dc=example,dc=com

```

항목에 nsRoleDN 속성이 있으면 이 항목이 해당 역할 정의의 DN에 해당하는 관리된 역할의 구성원임을 나타냅니다. 사용자가 자신의 nsRoleDN 속성은 수정할 수 있지만 nsManagedDisabledRole을 추가하거나 제거하지는 못하도록 하려면 다음 ACI를 추가합니다.

```

aci: (targetattr="nsRoleDN") (targetattrfilters="add=nsRoleDN:
(! (nsRoleDN=cn=AdministratorRole,dc=example,dc=com)),
del=nsRoleDN: (! (nsRoleDN=cn=nsManagedDisabledRole,dc=example,
dc=com) ) (version3.0;aci "allow mod of nsRoleDN by self except
for critical values"; allow(write) userdn="ldap:///self";)

```

필터링된 역할 정의 예

영업 책임자에 대한 필터링된 역할을 설정하려면 모든 책임자에게 isManager 속성이 있다는 전제 하에 아래의 ldapmodify 명령을 실행합니다.

```

ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=ManagerFilter,ou=sales,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition

```

```

objectclass: nsFilteredRoleDefinition
cn: ManagerFilter
nsRoleFilter: (isManager=True)
Description: filtered role for sales managers

```

nsFilteredRoleDefinition 객체 클래스는 LDAPsubentry, nsRoleDefinition 및 nsComplexRoleDefinition 객체 클래스로부터 상속됩니다. nsRoleFilter 속성은 다음과 같이 ou=sales 조직에서 부하 직원이 있는 모든 직원을 찾는 필터를 지정합니다.

```

ldapsearch -h host -p port -D "cn=Directory Manager" -w password \
-b "ou=People,dc=example,dc=com" -s sub "(cn=*Fuentes)"
dn: cn=Carla Fuentes,ou=sales,ou=People,dc=example,dc=com
cn: Carla Fuentes
isManager: TRUE
...
nsRole: cn=ManagerFilter,ou=sales,ou=People,dc=example,dc=com

```

주 필터링된 역할의 필터 문자열은 CoS 기법에 의해 생성된 가상 속성을 제외한 모든 속성에 기반을 둘 수 있습니다(220페이지의 "CoS" 참조).

필터링된 역할 구성원이 사용자 항목인 경우 필터링된 속성을 ACI로 보호하여 해당 사용자가 자신을 역할에 추가하거나 제거할 수 없도록 제한할 수도 있습니다.

중첩된 역할 정의 예

중첩된 역할 내에 중첩된 역할은 nsRoleDN 속성을 사용하여 지정합니다. 앞의 예에서 작성된 역할의 마케팅 직원과 영업 책임자를 모두 포함하는 역할을 작성하려면 아래 명령을 실행합니다.

```

ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=MarketingSales,ou=marketing,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsNestedRoleDefinition
cn: MarketingSales
nsRoleDN: cn=ManagerFilter,ou=sales,ou=People,dc=example,dc=com
nsRoleDN: cn=Marketing,ou=marketing,ou=People,dc=example,dc=com
nsRoleScopedDN: ou=sales,ou=People,dc=example,dc=com

```

nsNestedRoleDefinition 객체 클래스는 LDAPsubentry, nsRoleDefinition 및 nsComplexRoleDefinition 객체 클래스로부터 상속됩니다. nsRoleDN 속성에는 마케팅 관리된 역할과 영업 책임자 필터링된 역할의 DN이 포함됩니다. 앞의 예에서 설정된 두 사용자 Bob과 Carla는 새 중첩된 역할의 구성원이 됩니다.

이 필터의 범위에는 필터가 위치해 있는 하위 트리와 nsRoleScopeDN 속성 값 아래의 하위 트리로 구성된 기본 범위가 포함됩니다. 이 예에서 ManagerFilter는 ou=sales, ou=People, dc=example, dc=com 하위 트리에 있으므로 이 하위 트리를 필터 범위에 추가해야 합니다.

서비스 클래스(CoS) 정의

서비스 클래스(CoS) 기법은 클라이언트 응용 프로그램에 대한 항목이 검색될 때 가상 속성을 생성합니다. CoS를 사용하면 항목 관리를 간소화하고 필요한 저장 공간을 줄일 수 있습니다.

그룹 및 역할과 마찬가지로 CoS는 디렉토리에 있는 도우미 항목을 이용하며 콘솔이나 명령줄을 통해 구성할 수 있습니다. 다음 절에서는 CoS에 대해 설명하고 두 가지 방법으로 CoS를 관리하는 절차를 소개합니다.

주	모든 검색 작업은 CoS에서 생성된 속성의 존재 여부를 테스트하거나 해당 값을 비교할 수 있습니다. 클라이언트 검색 작업 또는 필터링된 역할에 사용된 내부 필터의 모든 필터 문자열에 가상 속성의 이름을 사용할 수 있습니다. Directory Server는 VLV(가상 목록 보기) 작업과 서버측 정렬 컨트롤에서도 가상 속성을 실제 속성과 똑같이 지원합니다.
---	---

CoS

CoS는 CoS 범위에 있는 모든 항목에 가상 속성과 해당 값을 정의하며, 이러한 항목을 *대상 항목*이라고 합니다. 각각의 CoS는 다음과 같은 디렉토리 항목으로 구성됩니다.

- CoS 정의 항목 - 사용하는 CoS 유형과 생성할 CoS 속성 이름을 식별합니다. 역할 정의 항목과 마찬가지로 CoS 정의 항목은 LDAPsubentry 객체 클래스로부터 상속됩니다. CoS의 범위는 CoS 정의 항목의 부모 아래에 있는 전체 하위 트리입니다. 한 개의 CoS 속성에 여러 가지 정의가 존재할 수 있으므로 이 속성은 여러 값을 갖습니다.

- 템플릿 항목 - 한 개 이상의 가상 속성 값이 포함됩니다. CoS 범위에 있는 모든 항목은 여기에 정의된 값을 사용합니다. 여러 개의 템플릿 항목이 존재할 수 있으므로 생성된 속성은 여러 값을 갖습니다.

CoS 유형은 CoS 정의와 템플릿 항목 간의 상호 작용에 따라 다음 세 가지 유형으로 나뉘어집니다.

- 포인터 CoS - 이 CoS 정의의 항목은 템플릿 항목을 해당 DN으로 직접 식별합니다. 모든 대상 항목의 CoS 속성 값은 템플릿에서 지정된 값과 같습니다.
- 간접 CoS - 이 CoS 정의는 간접 지정자 속성을 식별하며 대상 항목의 속성 값에는 템플릿 DN이 포함되어야 합니다. 간접 CoS에서 대상 항목은 각각 다른 템플릿을 사용할 수 있으므로 CoS 속성 값도 서로 다릅니다.
- 클래식 CoS - 이 CoS 정의는 템플릿의 기본 DN과 대상 항목의 속성 이름인 지정자를 식별합니다. 지정자 속성에는 템플릿 기본 DN과 함께 사용되어 CoS 값이 있는 템플릿을 지정하는 RDN(상대적인 도메인 이름)이 있어야 합니다.

CoS 정의 항목은 `cosSuperDefinition` 객체 클래스의 개별 인스턴스로, CoS 유형을 지정하기 위해 다음 객체 클래스 중 하나로부터 상속됩니다.

- `cosPointerDefinition`
- `cosIndirectDefinition`
- `cosClassicDefinition`

CoS 정의 항목에는 필요한 경우 가상 CoS 속성, 템플릿 DN 및 대상 항목의 지정자 속성 이름을 지정하는 CoS 유형별 속성이 포함됩니다. 기본적으로 CoS 기법은 CoS 속성과 동일한 이름의 기존 속성 값을 무시하지 않지만 CoS 정의 항목 구문을 사용하여 이 동작을 제어할 수 있습니다.

CoS 템플릿 항목은 `cosTemplate` 객체 클래스의 개별 인스턴스입니다. CoS 템플릿 항목에는 CoS 기법에서 생성된 속성 값이 포함됩니다. 특정 CoS에 대한 템플릿 항목은 디렉토리 트리에서 CoS 정의와 동일한 수준에 저장됩니다.

정의 항목과 템플릿 항목을 편리하게 관리하려면 두 항목이 같은 위치에 있어야 합니다. 또한 제공하는 기능을 쉽게 확인할 수 있도록 항목 이름을 지정해야 합니다. 예를 들어, `cn=C1CosGenerateEmployeeType,ou=People,dc=example,dc=com`과 같은 정의 항목의 DN이 `cn=ClassicCos1,ou=People,dc=example,dc=com`보다 쉽게 식별됩니다.

*Directory Server Deployment Planning Guide*는 CoS의 각 유형을 자세히 설명하고 배포 고려 사항 및 예를 제공합니다. CoS 유형별 객체 클래스와 속성에 대한 자세한 내용은 [226페이지의 "명령줄에서 CoS 관리"](#)를 참조하십시오.

CoS 제한

CoS 정의 항목과 템플릿 항목의 작성 및 관리에는 다음과 같은 제한이 따릅니다. CoS 가상 속성의 배포와 관련된 제한 사항은 *Directory Server Deployment Planning Guide*에 자세히 설명되어 있습니다.

CoS에서 생성된 속성을 사용한 검색은 색인화되지 않습니다. 모든 검색 필터에서 가상 속성의 존재 여부를 테스트하고 속성 값을 비교할 수 있습니다. *하지만* 가상 속성은 색인화할 수 없으므로 CoS에서 생성된 속성을 사용하는 모든 필터 구성 요소는 색인화되지 않은 검색을 초래하여 성능을 크게 저하시킵니다. *Directory Server 5.2*에서는 10개 이상의 클래식 CoS 템플릿 항목으로 검색 속도를 높여주는 해시 테이블을 제공합니다. 이 기능은 기본적으로 활성화되어 있습니다. 이 기능을 비활성화하는 방법에 대한 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.

제한된 하위 트리. cn=config 및 cn=schema 하위 트리에는 CoS 정의를 작성할 수 없으므로 이들 항목은 가상 속성을 가질 수 없습니다.

제한된 속성 유형. 다음과 같은 속성 유형은 같은 이름의 실제 속성과 동일한 동작을 수행하지 않으므로 CoS 기법에서 생성해서는 안 됩니다.

- userPassword - CoS에서 생성된 비밀번호 값은 디렉토리 서버에 바인드하는 데 사용할 수 없습니다.
- aci - 디렉토리 서버는 CoS에서 정의된 가상 ACI 값의 내용에 따른 액세스 제어를 적용하지 않습니다.
- objectclass - 디렉토리 서버는 CoS에서 정의된 가상 객체 클래스 값에 대해 스키마 검사를 수행하지 않습니다.
- nsRoleDN - CoS에서 생성된 nsRoleDN 값은 서버에서 역할을 생성할 때 사용되지 않습니다.

속성 하위 유형은 지원되지 않습니다. CoS 기법은 언어나 ;binary와 같은 하위 유형이 있는 속성을 생성하지 않습니다.

실제 속성 값과 가상 속성 값. CoS 기법에서는 항목에 정의된 "실제" 값과 CoS 템플릿에서 정의된 "가상" 값이 모두 포함된 여러 값을 갖는 속성을 생성하지 않습니다. 속성 값은 "실제 속성 값 무시" 및 229페이지의 "여러 값을 갖는 CoS 속성"에 설명된 것처럼 항목에 저장된 값이나 CoS 기법에서 생성된 값 중 하나입니다.

모든 템플릿은 로컬에 있어야 합니다. CoS 정의나 대상 항목의 지정자에 포함된 템플릿 항목 DN은 디렉토리 서버의 로컬 항목을 참조해야 합니다. 템플릿과 템플릿에 포함된 값은 디렉토리 연결이나 참조를 통해 검색할 수 없습니다.

콘솔에서 CoS 관리

이 절에서는 Directory Server 콘솔을 사용하여 CoS 정의를 작성 및 편집하는 방법에 대해 설명합니다.

또한, CoS 값의 보안을 유지해야 하는 경우 CoS 정의 항목 및 템플릿 항목과 대상 항목의 지정자 속성에 대한 ACI를 정의해야 합니다. CoS 보안 고려 사항에 대해서는 *Directory Server Deployment Planning Guide*를 참조하고, 콘솔을 사용하는 ACI 작성 절차는 6장, "액세스 제어 관리"를 참조하십시오.

새 CoS 만들기

포인터 CoS와 클래식 CoS의 경우, 정의 항목을 작성하기 전에 먼저 템플릿 항목을 작성해야 합니다.

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 새 템플릿 항목을 추가할 디렉토리 트리 항목을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 새로 만들기 > 기타 항목을 선택합니다.
또는 부모 항목을 선택하고 객체 메뉴에서 새로 만들기 > 기타 항목을 선택합니다.
2. 새 객체 대화 상자의 객체 클래스 목록에서 `costemplate`를 선택합니다. 새 템플릿의 특정 속성에 대한 기본값이 표시된 일반 편집기 대화 상자가 열립니다.
3. 다음과 같은 방법으로 새 템플릿 객체를 편집합니다.
 - a. `objectclass` 속성에 `LDAPsubentry` 및 `extensibleobject` 값을 추가합니다.

- b. cn 속성을 추가하고 템플리트를 식별할 값(예: `cosTemplateForHeadquartersFax`)을 지정합니다.
 - c. 이름 지정 속성을 새로운 cn 속성으로 변경합니다.
다른 속성을 추가하여 이름 지정 속성으로 사용할 수도 있지만 일반적으로 cn이 사용됩니다.
 - d. `cosPriority` 속성을 정수 값으로 설정하여 수정하거나, 필요 없는 경우 우선 순위 속성을 제거합니다. 자세한 내용은 [230페이지의 "Cos 속성 우선 순위"](#)를 참조하십시오.
 - e. CoS 기법에서 대상 항목에 생성할 속성과 해당 값을 추가합니다.
4. 일반 편집기 대화 상자에서 확인을 눌러 템플리트 항목을 작성합니다.
 5. 이 템플리트에 대한 포인터 CoS를 정의하려면 디렉토리 트리에서 새 템플리트 항목을 선택하고 메뉴에서 편집 > DN 복사를 선택합니다.

정의 항목을 작성하는 절차는 모든 CoS 유형에서 동일합니다.

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 새 CoS 정의를 추가할 디렉토리 트리 항목을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 새로 만들기 > 서비스 클래스 항목을 선택합니다.
또는 부모 항목을 선택하고 객체 메뉴에서 새로 만들기 > 서비스 클래스 항목을 선택합니다.
서비스 클래스 항목의 사용자 정의 편집기가 표시됩니다.
2. 새 서비스 클래스의 이름과 설명(선택 사항)을 입력합니다. CoS 정의 항목의 cn 이름 지정 속성에 이 이름이 표시됩니다.
3. 왼쪽 목록에서 속성 탭을 누릅니다. CoS 기법에서 대상 항목에 생성할 속성 목록이 대화 상자에 표시됩니다.
추가를 눌러 가능한 속성 목록을 탐색하고 목록에 속성을 추가합니다.
4. 속성이 목록에 추가되면 "서비스 클래스 동작" 열에 드롭다운 목록이 표시됩니다. 셀 내부를 눌러 다음과 같은 무시 동작을 선택합니다.
 - 대상 항목 속성을 무시하지 않습니다 - 대상 항목에 해당 속성 값이 저장되어 있지 않은 경우에만 CoS 속성 값이 생성됩니다.
 - 대상 항목 속성을 무시합니다 - CoS에서 생성된 속성 값이 대상 항목의 해당 속성 값을 무시합니다.

- **대상 항목 속성을 무시하며 작동합니다** - 속성이 모든 대상 값을 무시하고 작동하므로, 명시적으로 요청되지 않은 경우 클라이언트 응용 프로그램에 표시되지 않습니다.

주 스키마에서 작동 가능으로 정의된 속성만 작동 가능으로 설정할 수 있습니다.

5. 병합 열에는 병합 체계를 위한 확인란이 있습니다. 이 CoS 속성이 동일한 속성의 다른 CoS 값과 병합되도록 하려면 이 확인란을 선택합니다. 그러나 CoS 속성은 기존 값과 병합되지 않고 이를 대체하거나, 이전 열에서 정의한 대로 생성되지 않습니다.
6. 왼쪽 목록에서 템플릿 탭을 누릅니다. 템플릿 항목의 식별 방법을 선택한 다음 해당 필드에 입력합니다. 이렇게 하면 정의할 CoS 유형이 결정됩니다.
 - **해당 DN 사용** - 이 옵션은 포인터 CoS를 정의하므로 "템플릿 DN" 필드에 템플릿 항목의 DN을 입력합니다. 찾아보기를 눌러 디렉토리에서 템플릿 DN을 선택하거나 Ctrl-V를 입력하여 템플릿 항목이 작성된 후에 복사한 DN을 붙여넣습니다.
 - **대상 항목의 속성들 중 하나의 값 사용** - 이 옵션은 간접 CoS를 정의하므로 "속성 이름" 필드에 지정자 속성의 이름을 입력합니다. 이 경우에는 DN 값이 포함된 속성을 선택해야 합니다. 변경을 눌러 목록에서 속성을 선택합니다.
 - **대상 항목의 속성들 중 하나의 값과 DN을 모두 사용** - 이 옵션은 클래식 CoS를 정의하므로 템플릿의 기본 DN과 속성 이름을 모두 입력합니다. 찾아보기를 눌러 잠재적 대상 항목의 부모 항목을 선택한 다음 변경을 눌러 목록에서 속성을 선택합니다.
7. 확인을 눌러 CoS 정의 항목을 작성합니다.

기존 CoS 편집

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 CoS 정의 항목을 두 번 누르거나 마우스 오른쪽 버튼으로 누른 다음 팝업 메뉴에서 사용자 정의 편집기로 편집을 선택합니다.
서비스 클래스 항목의 사용자 정의 편집기가 표시됩니다.
2. 이름 및 설명 필드를 원하는 대로 편집합니다.

3. 왼쪽 목록에서 속성 탭을 눌러 CoS 기법에서 생성할 가상 속성을 추가하거나 제거합니다.
4. 왼쪽 목록에서 템플릿 탭을 눌러 템플릿 지정자 속성의 이름이나 템플릿 항목 DN을 다시 정의합니다. 이 대화 상자에서 CoS 정의 유형을 다시 정의할 수도 있습니다.
5. 확인을 눌러 변경 사항을 저장합니다.

CoS 삭제

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 디렉토리 트리를 탐색하여 CoS 정의 항목을 표시합니다.
2. CoS 항목을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 삭제를 선택합니다. 삭제를 확인하는 대화 상자가 표시됩니다. 예를 누릅니다.

명령줄에서 CoS 관리

모든 구성 정보와 템플릿 데이터는 디렉토리 항목으로 저장되기 때문에 LDAP 명령줄 도구를 사용하여 CoS 정의를 구성 및 관리할 수 있습니다. 이 절에서는 명령줄에서 CoS 정의와 템플릿 항목을 작성하는 방법에 대해 설명합니다.

또한, CoS 값의 보안을 유지해야 하는 경우 CoS 정의 항목 및 템플릿 항목과 대상 항목의 지정자 속성에 대한 ACI를 정의해야 합니다. CoS 보안 고려 사항에 대해서는 *Directory Server Deployment Planning Guide*를 참조하고, 명령줄에서 ACI를 작성하는 절차는 [6장, "액세스 제어 관리"](#)를 참조하십시오.

명령줄에서 CoS 정의 항목 작성

모든 CoS 정의 항목에는 LDAPsubentry 객체 클래스가 있으며 cosSuperDefinition 객체 클래스로부터 상속됩니다. 또한, 각각의 CoS 유형은 특정 객체 클래스로부터 상속되고 해당 속성을 포함합니다. 아래 표에는 CoS 정의 항목의 유형별 객체 클래스와 속성이 나와 있습니다.

표 5-2 CoS 정의 항목의 객체 클래스 및 속성

CoS 유형	CoS 정의 항목
포인터 CoS	objectclass: top objectclass: LDAPsubentry objectclass: cosSuperDefinition objectclass: cosPointerDefinition cosTemplateDN: DN cosAttribute: <i>attributeName override merge</i>
간접 CoS	objectclass: top objectclass: LDAPsubentry objectclass: cosSuperDefinition objectclass: cosIndirectDefinition cosIndirectSpecifier: <i>attributeName</i> cosAttribute: <i>attributeName override merge</i>
클래식 CoS	objectclass: top objectclass: LDAPsubentry objectclass: cosSuperDefinition objectclass: cosClassicDefinition cosTemplateDN: DN cosSpecifier: <i>attributeName</i> cosAttribute: <i>attributeName override merge</i>

cosAttribute는 항상 여러 값을 가지며, 각각의 값은 CoS 기법에서 생성할 속성을 정의합니다.

CoS 정의 항목에는 다음과 같은 속성을 사용할 수 있습니다. 이러한 속성에 대한 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.

표 5-3 CoS 정의 항목 속성

속성	CoS 정의 항목에서의 용도
cosAttribute: <i>attributeName override merge</i>	값을 생성할 가상 속성 이름을 정의합니다. 이 속성은 여러 값을 가지며, 각각의 값은 템플릿에서 값이 생성될 속성 이름을 지정합니다. <i>override</i> 및 <i>merge</i> 지정자는 표 아래에 설명된 특별한 경우에 CoS 속성 값이 어떻게 계산되는지 지정합니다. <i>attributeName</i> 에는 하위 유형이 포함될 수 없습니다. 하위 유형이 있는 속성 이름은 무시되지만 cosAttribute의 다른 값은 정상적으로 처리됩니다.
cosIndirectSpecifier: <i>attributeName</i>	간접 CoS에서 값이 사용되어 템플릿 항목을 식별하는 대상 항목의 속성 이름을 정의합니다. 이름이 지정된 속성을 지정자라고 하며 각 대상 항목의 전체 DN 문자열이 포함되어야 합니다. 이 속성은 한 개의 값을 갖지만 지정자 속성은 많은 항목을 지정하기 위해 여러 값을 가질 수 있습니다.

표 5-3 CoS 정의 항목 속성(계속)

속성	CoS 정의 항목에서의 용도
cosSpecifier: attributeName	클래식 CoS에서 값이 사용되어 템플릿 항목을 식별하는 대상 항목의 속성 이름을 정의합니다. 이름이 지정된 속성을 지정자라고 하며 템플릿 항목의 RDN에 있는 문자열이 포함되어야 합니다. 이 속성은 한 개의 값을 갖지만 지정자 속성은 많은 항목을 지정하기 위해 여러 값을 가질 수 있습니다.
cosTemplateDN: DN	포인터 CoS 정의에 템플릿 항목의 전체 DN을 제공하거나 클래식 CoS 정의에 템플릿 항목의 기본 DN을 제공합니다.

cosAttribute 속성은 CoS 속성 이름 뒤에 두 개의 한정자를 허용합니다. *override* 한정자는 다음 값 중 하나를 갖습니다.

- default(한정자 없음) - 항목에 저장된 실제 속성 값이 가상 속성과 같은 유형이면 서버에서 실제 속성 값을 무시하지 않음을 나타냅니다.
- override - 항목에 저장된 값이 있어도 항상 CoS에서 생성된 값이 반환됨을 나타냅니다.
- operational - 검색에서 명시적으로 요청된 경우에만 속성이 반환됨을 나타냅니다. 작동 가능 속성은 스키마 검사를 통과하지 않아도 반환될 수 있으며 *override* 한정자와 같은 동작을 합니다.

스키마에서 작동 가능으로 정의된 속성만 작동 가능으로 설정할 수 있습니다. 예를 들어, CoS에서 *description* 속성 값을 생성한 경우 이 속성은 스키마에서 작동 가능으로 정의되지 않았으므로 *operational* 한정자를 사용할 수 없습니다.

merge 한정자는 비어 있거나 아래 값이 지정됩니다.

- merge-schemes - 가상 CoS 속성이 많은 템플릿나 CoS 정의의 여러 값을 가질 수 있도록 허용합니다. 자세한 내용은 [229페이지의 "여러 값을 갖는 CoS 속성"](#)을 참조하십시오.

실제 속성 값 무시

아래 명령을 실행하여 `override` 한정자가 있는 포인터 CoS 정의 항목을 작성할 수 있습니다.

```
dn: cn=pointerCoS,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn: cn=exampleUS,cn=data
cosAttribute: postalCode override
```

이 포인터 CoS 정의 항목은 `postalCode` 속성 값을 생성하는 템플리트 항목 `cn=exampleUS,cn=data`에 연결되어 있음을 나타냅니다. 또한 `override` 한정자는 이 값이 대상 항목에 있는 `postalCode` 속성 값보다 우선함을 나타냅니다.

주 CoS 속성이 `operational` 한정자나 `override` 한정자로 정의된 경우에는 CoS 범위의 항목에서 해당 속성의 "실제" 값에 대해 쓰기 작업을 수행할 수 없습니다.

여러 값을 갖는 CoS 속성

`merge-schemes` 한정자를 지정하면 생성된 CoS 속성이 여러 값을 가질 수 있습니다. CoS 속성이 여러 값을 갖게 되는 경우는 다음 두 가지입니다.

- 간접 CoS나 클래식 CoS에서 대상 항목의 지정자 속성은 여러 값을 가질 수 있습니다. 이 경우, 각각의 값은 템플리트를 지정하고 각 템플리트의 값이 생성된 값에 포함됩니다.
- 모든 유형에서 `cosAttribute`에 동일한 속성 이름이 포함된 여러 개의 CoS 정의 항목이 존재할 수 있습니다. 이 경우, 모든 정의에 `merge-schemes` 한정자가 포함되어 있으면 생성된 속성에는 각 정의에서 계산된 값이 모두 포함됩니다.

두 경우가 함께 발생하여 더 많은 값을 정의할 수도 있습니다. 하지만 중복 값은 항상 생성된 속성에 한 번만 반환됩니다.

`merge-schemes` 한정자가 없는 경우, 다음 절에 설명된 것처럼 템플리트 항목의 `cosPriority` 속성을 사용하여 생성된 이 속성에 대해 모든 템플리트에 지정되는 한 개의 값을 결정합니다.

merge-schemes 한정자는 대상에서 정의된 "실제" 값과 템플릿에서 생성된 값을 병합하지 않습니다. merge 한정자는 override 한정자와 별개로 작동하므로 모든 쌍이 가능하며 각 한정자의 동작은 상호 보충됩니다. 또한 순서에 상관 없이 속성 이름 뒤에 한정자를 지정할 수 있습니다.

주 한 속성에 대한 여러 개의 CoS 정의가 있는 경우 모두 동일한 override 및 merge 한정자를 사용해야 합니다. CoS 정의에 여러 개의 한정자 쌍이 있으면 한 개의 쌍이 임의로 선택되어 모든 정의에서 사용됩니다.

Cos 속성 우선 순위

여러 개의 CoS 정의나 여러 값을 갖는 지정자가 있지만 merge-schemes 한정자가 없는 경우 Directory Server는 우선 순위 속성을 사용하여 가상 속성 값 하나를 정의하는 한 개의 템플릿을 선택합니다.

cosPriority 속성은 해당되는 모든 템플릿 중에서 특정 템플릿의 전역 우선 순위를 나타냅니다. 우선 순위 0(제로)이 가장 높은 우선 순위입니다. cosPriority 속성이 없는 템플릿은 가장 낮은 우선 순위로 간주됩니다. 두 개 이상의 템플릿이 속성 값을 제공하지만 우선 순위가 같은 경우(혹은 없는 경우)에는 임의로 값이 선택됩니다.

merge-schemes 한정자를 사용하는 경우 템플릿 우선 순위는 고려되지 않습니다. 병합하는 경우에는 정의된 우선 순위에 관계 없이 해당되는 모든 템플릿이 값을 정의합니다. 다음 절에 설명된 것처럼 cosPriority 속성은 CoS 템플릿 항목에 정의됩니다.

주 cosPriority 속성에 음수 값을 지정할 수 없습니다. 또한 간접 CoS에서 생성된 속성은 우선 순위를 지원하지 않습니다. 간접 CoS 정의의 템플릿 항목에는 cosPriority를 사용하지 마십시오.

명령줄에서 CoS 템플릿 항목 작성

포인터 CoS나 클래식 CoS를 사용하는 경우에는 템플릿 항목에 LDAPsubentry 및 cosTemplate 객체 클래스가 포함됩니다. 이 항목은 특별히 CoS 정의에 대해 작성되어야 합니다. CoS 템플릿 항목을 LDAPsubentry 객체 클래스의 한 인스턴스로 만들면 구성 항목에 영향 받지 않고 일반 검색을 수행할 수 있습니다.

간접 CoS 기법의 템플리트는 디렉토리에 있는 임의의 기존 항목으로, 대상을 미리 식별하거나 LDAPsubentry 객체 클래스를 제공할 필요는 없지만 보조 cosTemplate 객체 클래스가 있어야 합니다. 간접 CoS 템플리트는 가상 속성과 해당 값을 생성하기 위해 CoS를 평가할 때만 액세스됩니다.

CoS 템플리트에는 대상 항목의 CoS에서 생성된 속성과 값이 항상 포함되어 있어야 합니다. 속성 이름은 CoS 정의 항목의 cosAttribute 속성에 지정됩니다.

아래 예에서는 postalCode 속성을 생성하는 포인터 CoS에 대한 가장 높은 우선 순위의 템플리트 항목을 보여줍니다.

```
dn: cn=ZipTemplate,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
postalCode: 95054
cosPriority: 0
```

다음 절에서는 템플리트 항목 예 및 CoS 정의 항목의 유형별 예를 제공합니다.

포인터 CoS 예

아래 명령은 cosPointerDefinition 객체 클래스가 있는 포인터 CoS 정의 항목을 작성합니다. 이 정의 항목은 위에서 지정된 CoS 템플리트를 사용하여 ou=People,dc=example,dc=com 트리의 모든 항목에서 일반 우편 번호를 공유합니다.

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=pointerCoS,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn: cn=ZipTemplate,ou=People,dc=example,dc=com
cosAttribute: postalCode
```

CoS 템플리트 항목(cn=ZipTemplate,ou=People,dc=example,dc=com)은 postalCode 속성에 저장된 값을 ou=People,dc=example,dc=com 접미어에 있는 모든 항목에 제공합니다. 아래 명령을 실행하여 같은 하위 트리에서 우편 번호가 없는 항목을 검색하면 생성된 속성 값이 표시됩니다.

```
ldapsearch -h host -p port -D "cn=Directory Manager" -w password \
-b "ou=People,dc=example,dc=com" -s sub "(cn=*Jensen)"
dn: cn=Babs Jensen,ou=People,dc=example,dc=com
cn: Babs Jensen
...
postalCode: 95054
```

간접 CoS 예

간접 CoS는 `cosIndirectSpecifier` 속성에 속성 이름을 지정하여 각 대상별 템플리트를 찾습니다. 이 예에서 간접 CoS는 대상 항목의 `manager` 속성을 사용하여 CoS 템플리트 항목을 식별합니다. 템플리트 항목은 관리자의 사용자 항목이며 생성할 속성 값이 포함되어 있어야 합니다.

아래 명령은 `cosIndirectDefinition` 객체 클래스가 있는 간접 CoS 정의 항목을 작성합니다.

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=generateDeptNum,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosIndirectDefinition
cosIndirectSpecifier: manager
cosAttribute: departmentNumber
```

그런 다음, `cosTemplate` 객체 클래스를 템플리트 항목에 추가하고 이 항목이 생성될 속성을 정의하는지 확인합니다. 이 예에서는 모든 관리자 항목이 템플리트입니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=Carla Fuentes,ou=People,dc=example,dc=com
changetype: modify
add: objectclass
objectclass: cosTemplate
-
add: departmentNumber
departmentNumber: 318842
```

이 CoS를 사용하면 `manager` 속성이 포함된 대상 항목

(`ou=People,dc=example,dc=com` 아래의 항목)에 자동으로 해당 관리자의 부서 번호가 지정됩니다. `departmentNumber` 속성은 서버에 없기 때문에 대상 항목의 가상 속성이 되지만 대상 항목의 일부로 반환됩니다. 예를 들어, `Babs Jensen`의 관리자가 `Carla Fuentes`로 정의되어 있으면 부서 번호는 다음과 같이 생성됩니다.


```
ldapsearch -h host -p port -D "cn=Directory Manager" -w password \
-b "ou=People,dc=example,dc=com" -s sub "(cn=*Jensen)"
dn: cn=Babs Jensen,ou=People,dc=example,dc=com
cn: Babs Jensen
...
manager: cn=Carla Fuentes,ou=People,dc=example,dc=com
departmentNumber: 318842
```

클래식 CoS 예

이 예에서는 클래식 CoS를 사용하여 우편 주소를 생성하는 방법을 보여줍니다. 생성된 속성은 CoS 정의의 `cosTemplateDn`과 대상 항목의 `cosSpecifier` 속성을 조합하여 찾은 템플릿 항목에 제공됩니다. 아래 명령은 `cosClassicDefinition` 객체 클래스를 사용하여 정의 항목을 작성합니다.

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=classicCoS,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosClassicDefinition
cosTemplateDn: ou=People,dc=example,dc=com
cosSpecifier: building
cosAttribute: postalAddress
```

같은 명령을 사용하여 각 건물의 우편 주소를 제공하는 템플릿 항목을 작성합니다.

```
dn: cn=B07,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
postalAddress: 7 Old Oak Street$Anytown, CA 95054
```

이 CoS를 사용하면 `building` 속성이 포함된 대상 항목 (`ou=People,dc=example,dc=com` 아래의 항목)에 자동으로 해당 우편 주소가 지정됩니다. CoS 기법은 RDN에 지정자 속성 값이 있는 템플릿 항목을 검색합니다. 이 예에서 Babs Jensen이 건물 B07에 지정되어 있으면 우편 주소는 다음과 같이 생성됩니다.

```
ldapsearch -h host -p port -D "cn=Directory Manager" -w password \
-b "ou=People,dc=example,dc=com" -s sub "(cn=*Jensen)"
dn: cn=Babs Jensen,ou=People,dc=example,dc=com
cn: Babs Jensen
...
building: B07
postalAddress: 7 Old Oak Street$Anytown, CA 95054
```

역할 기반의 속성 작성

항목이 소유한 역할에 따라 항목의 속성 값을 생성하는 클래식 CoS 체계를 작성할 수 있습니다. 예를 들어, 역할 기반의 속성을 사용하여 서버에서 항목별 제한을 조회하도록 설정할 수 있습니다.

역할 기반의 속성을 작성하려면 `nsRole` 속성을 클래식 CoS의 CoS 정의 항목에 있는 `cosSpecifier`로 사용합니다. `nsRole` 속성은 여러 값을 가질 수 있으므로 두 개 이상의 템플릿 항목이 있는 CoS 체계를 정의할 수 있습니다. 사용할 템플릿 항목을 명확히 지정하기 위해 CoS 템플릿 항목에 `cosPriority` 속성을 추가할 수 있습니다.

예를 들어, 관리자 역할의 구성원이 표준 우편함 할당량을 초과할 수 있도록 허용하는 CoS를 작성할 수 있습니다. 관리자 역할은 다음과 같습니다.

```
dn: cn=ManagerRole,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: ManagerRole
nsRoleFilter: (isManager=True)
Description: filtered role for managers
```

클래식 CoS 정의 항목은 다음과 같이 작성될 것입니다.

```
dn: cn=generateManagerQuota,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosClassicDefinition
cosTemplateDn: cn=managerCOS,ou=People,dc=example,dc=com
cosSpecifier: nsRole
cosAttribute: mailboxquota override
```

CoS 템플릿 이름은 `cosTemplateDn`과 `nsRole` 값(역할 DN)의 조합이어야 합니다. 예를 들면 다음과 같습니다.

```
dn:cn="cn=ManagerRole,ou=People,dc=example,dc=com",cn=managerCOS,
ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
mailboxquota: 1000000
```

CoS 템플릿 항목은 mailboxquota 속성 값을 제공합니다. override 한정자를 추가하면 CoS는 대상 항목에 있는 기존의 mailboxquota 속성 값을 모두 무시합니다. 역할 구성원인 대상 항목에는 다음과 같이 역할 및 CoS에서 생성된 가상 속성이 지정됩니다.

```
ldapsearch -h host -p port -D "cn=Directory Manager" -w password \
-b "ou=People,dc=example,dc=com" -s sub "(cn=*Fuentes)"
dn: cn=Carla Fuentes,ou=People,dc=example,dc=com
cn: Carla Fuentes
isManager: TRUE
...
nsRole: cn=ManagerRole,ou=People,dc=example,dc=com
mailboxquota: 1000000
```

주 역할 항목과 CoS 정의 항목은 해당 범위에 동일한 대상 항목이 포함되도록 디렉토리 트리에서 같은 위치에 있어야 합니다. CoS 대상 항목도 같은 위치에 있어야만 검색 및 유지 관리가 용이합니다.

CoS 플러그인 모니터

Directory Server 5.2에서는 CoS 플러그인 인의 특정 측면을 모니터링할 수 있습니다. CoS 모니터링 속성은 cn=monitor,cn=Class of Service,cn=plugins,cn=config 항목 아래에 있습니다. 이러한 속성과 이 속성이 제공하는 정보에 대한 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.

서비스 클래스(CoS) 정의

액세스 제어 관리

디렉토리 내용에 대한 액세스를 제어하는 것은 디렉토리 보안에 있어 핵심적인 부분입니다. 이 장에서는 디렉토리에 액세스하는 사용자에게 부여할 권한을 결정하는 액세스 제어 명령(ACI)에 대해 설명합니다. Directory Server는 지정된 항목에 대한 특정 사용자의 유효 권한을 볼 수 있는 기능을 제공합니다. 이 기능을 사용하면 복잡하면서도 강력한 액세스 제어 기법을 간단하게 관리할 수 있습니다.

전체 보안 정책을 제공하는 액세스 제어 전략은 디렉토리 배포의 계획 단계에서 정의해야 합니다. 액세스 제어 전략의 계획에 대한 팁은 *Directory Server Deployment Planning Guide*를 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

- 액세스 제어 원칙
- 기본 ACI
- ACI 구문
- 바인드 규칙
- 명령줄에서 ACI 작성
- 콘솔에서 ACI 작성
- 액세스 제어 사용 예
- 유효 권한 보기
- 고급 액세스 제어: 매크로 ACI 사용
- 액세스 제어 및 복제
- 액세스 제어 로깅 정보
- 이전 릴리스와의 호환성

액세스 제어 원칙

액세스를 정의하는 기법을 *액세스 제어*라고 합니다. 요청이 수신되면 서버는 바인드 작업 시 사용자가 제공한 인증 정보와 서버에 정의된 액세스 제어 명령(ACI)을 사용하여 디렉토리 정보에 대한 액세스를 허용하거나 거부합니다. 서버는 읽기, 쓰기, 검색 또는 비교 권한을 허용하거나 거부할 수 있습니다. 사용자에게 부여되는 권한 수준은 제공된 인증 정보에 따라 달라집니다.

액세스 제어를 사용하면 전체 디렉토리, 디렉토리의 하위 트리, 디렉토리의 특정 항목(구성 작업을 정의하는 항목 포함), 항목 속성의 특정 집합 또는 특정 항목 속성 값에 대한 액세스를 제어할 수 있습니다. 특정 사용자, 특정 그룹이나 역할에 속한 모든 사용자 또는 디렉토리에 있는 모든 사용자의 권한을 설정할 수 있습니다. 마지막으로 IP 주소나 DNS 이름으로 식별된 특정 클라이언트에 대해서도 액세스를 정의할 수 있습니다.

ACI 구조

액세스 제어 명령은 항목 속성으로 디렉토리에 저장됩니다. aci 속성은 작동 가능 속성으로, 항목의 객체 클래스에 정의되어 있지 않아도 디렉토리의 모든 항목에서 사용할 수 있습니다. 클라이언트로부터 LDAP 요청이 수신되면 Directory Server는 이 속성을 사용하여 부여하거나 거부할 권한을 평가합니다. 특별한 요청이 있을 경우 aci 속성은 ldapsearch 작업으로 반환됩니다.

ACI 명령문은 크게 세 부분으로 구성됩니다.

- 대상 - 권한을 적용할 항목이나 속성을 지정합니다.
- 권한 - 허용 또는 거부되는 작업을 정의합니다.
- 바인드 규칙 - 바인드 DN을 기준으로 ACI에 적용되는 사용자를 지정합니다.

ACI의 권한 부분과 바인드 규칙 부분은 한 쌍으로 설정되며 액세스 제어 규칙(ACR)이라고도 합니다. 대상에 대한 액세스 권한은 해당 규칙의 평가 결과에 따라 부여되거나 거부됩니다. 자세한 내용은 [242페이지의 "ACI 구문"](#)을 참조하십시오.

ACI 배치

ACI가 포함된 항목에 자식 항목이 없으면 해당 항목에만 ACI가 적용됩니다. 항목에 자식 항목이 있으면 해당 항목 및 모든 하위 항목에 ACI가 적용됩니다. 따라서 서버는 항목에 대한 액세스 권한을 평가할 때 요청된 항목과 루트 접미어의 기본 항목 사이에 있는 모든 항목에 대해 ACI를 확인합니다.

aci 속성은 여러 값을 가지므로 동일한 항목 또는 하위 트리에 대해 여러 개의 ACI를 정의할 수 있습니다.

해당 항목에 직접 적용되지는 않지만 하위 트리의 일부 또는 모든 항목에 적용되는 ACI를 항목에 작성할 수 있습니다. 이 경우, 실제로 디렉토리 트리의 하위 수준에 있는 항목에 적용되는 일반 ACI를 디렉토리 트리의 상위 수준에 배치할 수 있다는 이점이 있습니다. 예를 들어, organizationalUnit 항목이나 locality 항목 수준에서 inetorgperson 객체 클래스가 있는 항목을 대상으로 하는 ACI를 작성할 수 있습니다.

이 기능을 사용하면 상위 수준 분기점에 일반 규칙을 배치하여 디렉토리 트리의 ACI 수를 최소화할 수 있습니다. 특정 규칙의 범위를 제한하려면 최대한 리프 항목 가까이 규칙을 배치해야 합니다.

주 DN=""을 가진 루트 DSE 항목에 배치된 ACI는 해당 항목에만 적용됩니다.

ACI 평가

서버는 특정 항목에 대한 액세스 권한을 평가하기 위해 항목 루트 접미어의 기본 항목에 이르기까지 항목 자체 및 부모 항목에 있는 ACI 목록을 작성합니다. 평가 중에 서버는 이 순서대로 ACI를 처리합니다. ACI는 항목과 루트 접미어의 기본 항목 사이에 있는 모든 접미어 및 하위 접미어에서 평가되지만 다른 서버에 있는 연결 접미어에서는 평가되지 않습니다.

주 디렉토리 관리자는 액세스 제어가 적용되지 않는 권한을 가진 유일한 사용자입니다. 클라이언트가 디렉토리 관리자로 디렉토리에 바인드하면 서버는 작업 수행 전에 ACI를 평가하지 않습니다.

따라서 디렉토리 관리자로 LDAP 작업을 수행하면 다른 사용자에 비해 성능이 훨씬 증가하기 때문에 항상 일반 사용자 아이디로 디렉토리 성능을 테스트해야 합니다.

항목에 적용되는 ACI가 없으면 기본적으로 디렉토리 관리자를 제외한 모든 사용자의 액세스가 거부됩니다. 사용자가 서버 항목에 액세스하려면 ACI에서 명시적으로 액세스 권한을 부여해야 합니다. 기본 ACI는 익명 읽기 액세스를 정의하며 사용자가 보안에 필요한 속성을 제외한 자신의 항목을 수정할 수 있도록 허용합니다. 자세한 내용은 [241페이지의 "기본 ACI"](#)를 참조하십시오.

서버는 대상 항목 가까이 있는 ACI부터 먼저 처리하지만 항목에 적용되는 모든 ACI의 결과는 누적됩니다. 특정 ACI에서 액세스를 거부하지 않으면 ACI에서 부여한 액세스는 모두 허용됩니다. 목록 어디에 있건 동일 자원에 대해 액세스를 허용하는 ACI보다 액세스를 거부하는 ACI가 우선적으로 적용됩니다.

예를 들어, 디렉토리의 루트 수준에서 쓰기 권한을 거부하면 사용자는 자신에게 부여된 특정 권한에 관계 없이 디렉토리에 쓸 수 없습니다. 특정 사용자에게 디렉토리에 대한 쓰기 권한을 부여하려면 이 사용자가 포함되지 않도록 쓰기 권한에 대한 기존 거부 범위를 제한해야 합니다.

ACI 제한

디렉토리 서비스에 대한 액세스 제어 정책을 작성하는 경우 다음과 같은 제한에 주의해야 합니다.

- 연결 기능을 사용하여 여러 서버에 디렉토리 트리를 배포하는 경우 액세스 제어 명령문에서 사용할 수 있는 키워드에 몇 가지 제한이 적용됩니다.
 - 그룹 항목을 사용하는 ACI(groupdn 키워드)는 그룹 항목과 같은 서버에 있어야 합니다. 동적 그룹일 경우 그룹의 모든 구성원 항목도 이 서버에 있어야 합니다. 정적 그룹일 경우에는 원격 서버에 구성원 항목이 위치할 수 있습니다.
 - 역할 정의를 사용하는 ACI(roledn 키워드)는 역할 정의 항목과 같은 서버에 있어야 합니다. 해당 역할을 지정할 모든 항목도 동일한 서버에 있어야 합니다.

하지만 userattr 키워드 등을 사용하여 대상 항목에 저장된 값을 바인드 사용자 항목에 저장된 값과 일치시킬 수 있으므로 ACI가 포함된 서버에 바인드 사용자 항목이 없어도 정상적으로 액세스 평가가 이루어집니다.

액세스 제어 평가를 연결하는 방법은 [166페이지의 "연결 접미어를 통한 액세스 제어"](#)를 참조하십시오.

- CoS에서 생성된 속성은 일부 ACI 키워드에서만 사용할 수 있습니다. 특히 CoS에서 생성된 속성을 `userattr` 키워드 및 `userdnattr` 키워드와 함께 사용하면 액세스 제어 규칙이 작동하지 않습니다. 자세한 내용은 260페이지의 "[userattr 키워드 사용](#)"을 참조하십시오. CoS에 대한 자세한 내용은 5장, "[아이디 및 역할 관리](#)"를 참조하십시오.
- 액세스 제어 규칙은 항상 로컬 서버에서 평가됩니다. ACI 키워드에 사용된 LDAP URL에는 서버의 호스트 이름이나 포트 번호를 지정할 수 없습니다. 이 경우 LDAP URL이 적용되지 않습니다. 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.
- 프록시 권한을 부여하는 경우 사용자에게 디렉토리 관리자로서의 프록시 권한을 부여하거나 디렉토리 관리자에게 프록시 권한을 부여할 수 없습니다.

기본 ACI

Directory Server를 설치하면 구성 중에 지정한 루트 접미어에 다음과 같은 기본 ACI가 정의됩니다.

- 모든 사용자는 검색, 비교 및 읽기 작업을 위해 디렉토리에 대한 익명 액세스 권한을 갖습니다(`userpassword` 속성 제외).
- 바인드된 사용자는 디렉토리에 있는 자신의 항목을 수정할 수 있지만 삭제할 수는 없습니다. 또한 `aci`, `nsroledn` 및 `passwordPolicySubentry` 속성과 자원 제한 속성, 비밀번호 정책 상태 속성, 계정 잠금 상태 속성 등은 수정할 수 없습니다.
- 구성 관리자(기본적으로 `uid=admin,ou=Administrators,ou=TopologyManagement,o=NetscapeRoot`)는 프록시 권한을 제외한 모든 권한을 갖습니다.
- 구성 관리자 그룹의 모든 구성원은 프록시 권한을 제외한 모든 권한을 갖습니다.
- 디렉토리 관리자 그룹의 모든 구성원은 프록시 권한을 제외한 모든 권한을 갖습니다.
- SIE(Server Instance Entry) 그룹의 모든 구성원은 프록시 권한을 제외한 모든 권한을 갖습니다. SIE 그룹은 Administration Server에서 이 디렉토리 서버 그룹의 관리자 그룹입니다.

디렉토리에 새 루트 접미어를 작성하면 해당 기본 항목에는 위에 열거된 기본 ACI(자체 수정 ACI 제외)가 포함됩니다. 보안을 강화하려면 142페이지의 "[콘솔에서 새 루트 접미어 작성](#)"에 설명된 것처럼 이 ACI를 추가해야 합니다.

관리 서버의 NetscapeRoot 하위 트리에는 자체 기본 ACI 집합이 있습니다.

- 구성 관리자 그룹의 모든 구성원은 프록시 권한을 제외하고 NetscapeRoot 하위 트리에 대한 모든 권한을 가지므로 구성 관리자 그룹에 새 구성원을 추가할 수 있습니다.
 - 모든 사용자는 검색 및 읽기 작업을 위해 NetscapeRoot 하위 트리에 대한 익명 액세스 권한을 갖습니다.
 - 관리 그룹의 구성원은 그룹 확장 ACI를 사용하여 그룹 정의에 액세스할 수 있습니다.
- 다음 절에서는 이러한 기본 설정을 조직의 요구에 맞게 수정하는 방법에 대해 설명합니다.

ACI 구문

ACI는 다양한 변형이 가능한 복잡한 구조로 이루어져 있습니다. 콘솔 또는 명령줄에서 ACI를 작성하고 수정하는 경우 LDIF 형식의 ACI 구문을 명확히 이해해야 합니다. ACI 구문에 대해서는 다음 절에서 자세히 설명합니다.

팁 ACI는 매우 복잡하기 때문에 Directory Server 콘솔에서 모든 ACI를 시각적으로 편집할 수는 없습니다. 또한 다수의 디렉토리 항목에 대한 액세스 제어를 설정하는 경우 명령줄을 사용하는 것이 훨씬 빠릅니다. 따라서 ACI 구문을 명확히 이해해야만 효과적인 액세스 제어가 설정된 보안 디렉토리를 작성할 수 있습니다.

aci 속성 구문은 다음과 같습니다.

```
aci: (target) (version 3.0;acl "name";permission bindRules;)
```

여기서,

- *target*은 액세스를 제어할 항목, 속성 또는 항목과 속성 집합을 지정합니다. 고유 이름이나 하나 이상의 속성, 또는 한 개의 LDAP 필터를 대상으로 지정할 수 있습니다. 대상은 선택 사항으로, 대상을 지정하지 않으면 ACI가 정의된 전체 항목과 모든 자식 항목에 해당 ACI가 적용됩니다.
- *version 3.0*은 ACI 버전을 식별하는 필수 문자열입니다.

- *name*은 ACI의 이름입니다. ACI를 식별하는 모든 문자열은 이름이 될 수 있습니다. ACI 이름은 필수이며 ACI의 결과를 설명하는 이름이어야 합니다.

팁 이름에 대한 제한은 없지만 ACI에 고유 이름을 사용하는 것이 좋습니다. 고유 이름을 지정하면 "유효 권한 보기" 컨트롤을 사용하여 적용된 ACI를 확인할 수 있습니다.

- *permission*은 허용 또는 거부되는 권한(예: 읽기 권한이나 검색 권한)을 구체적으로 열거합니다.
- *bindRules*는 액세스 권한을 부여 받기 위해 사용자가 제공해야 하는 자격 증명과 바인드 매개 변수를 지정합니다. 바인드 규칙은 사용자나 그룹 구성원 자격 또는 클라이언트의 연결 등록정보를 기준으로 할 수도 있습니다.

여러 개의 대상과 권한 바인드 규칙 쌍을 지정할 수 있으므로 대상으로 지정되는 항목과 속성을 모두 구체화하여 지정된 대상에 대해 다양한 액세스 제어를 효율적으로 설정할 수 있습니다. 예를 들면 다음과 같습니다.

```
aci: (target)...(target) (version 3.0;acl "name"; permission bindRule;
permission bindRule; ...; permission bindRule;)
```

완성된 LDIF ACI의 예는 다음과 같습니다.

```
aci: (target="ldap:///uid=bjensen,dc=example,dc=com"
(targetattr="*")(version 3.0; acl "example"; allow (write)
userdn="ldap:///self";)
```

이 예에서 ACI는 사용자 bjensen이 자신의 디렉토리 항목에 있는 모든 속성을 수정할 수 있는 권한이 있음을 나타냅니다.

다음 절에서는 ACI 각 부분의 구문에 대해 자세히 설명합니다.

대상 정의

대상은 ACI가 어디에 적용되는지 식별합니다. 클라이언트에서 항목의 속성에 대한 작업을 요청하면 서버는 ACI를 평가하여 이 작업을 허용해야 할지 거부해야 할지를 확인하기 위해 대상을 평가합니다. 대상을 지정하지 않으면 ACI는 *aci* 속성이 있는 항목과 하위 항목의 모든 속성에 적용됩니다.

일반적인 대상 구문은 다음 중 하나를 사용합니다.

```
(keyword = "expression")
(keyword != "expression")
```

여기서,

- *keyword*는 대상 유형을 나타냅니다. 244페이지의 표 6-1의 키워드는 다음과 같은 대상 유형을 정의합니다.
 - 디렉토리 항목이나 하위 트리
 - 항목 속성
 - LDAP 필터와 일치하는 항목이나 속성 집합
 - LDAP 필터와 일치하는 속성 값이나 값 조합
- 같음(=)은 대상이 *expression*에 지정된 객체임을 나타내며, 같지 않음(!=)은 대상이 *expression*에 지정된 객체가 아님을 나타냅니다.

주 *targattrfilters* 키워드에 대해서는 "같지 않음" 연산자를 사용할 수 없습니다.

- *expression*은 키워드를 사용하여 대상을 식별하며, 현재 구현에서는 *targetattr=**와 같은 표현식을 허용하지만 구문상 *expression*을 인용 부호(" ")로 묶어야 합니다. 그러나 이후 버전에서 구문 검사가 더 엄격해질 수 있으므로 항상 인용 부호를 사용해야 합니다.

아래 표에는 각 키워드 및 관련된 표현식이 나와 있습니다.

표 6-1 LDIF target 키워드

키워드	유효한 표현식	와일드카드 허용 여부
대상	<i>ldap:///distinguished_name</i>	예
<i>targetattr</i>	속성	예
<i>targetfilter</i>	<i>LDAP_filter</i>	예
<i>targattrfilters</i>	<i>LDAP_operation:LDAP_filter</i>	예

대상 디렉토리 항목 지정

특정 디렉토리 항목과 하위 항목을 대상으로 지정하려면 LDAP URL에서 DN과 target 키워드를 사용합니다. 지정된 DN은 ACI가 정의된 항목의 하위 트리에 위치해야 합니다. 대상 표현식 구문은 다음과 같습니다.

```
(target = "ldap:///distinguished_name")
(target != "ldap:///distinguished_name")
```

고유 이름은 ACI가 정의된 항목이 루트가 되는 하위 트리에 위치해야 합니다. 예를 들어, ou=People,dc=example,dc=com에 대한 ACI에서는 다음과 같은 대상을 사용할 수 있습니다.

```
(target = "ldap:///uid=bjensen,ou=People,dc=example,dc=com")
```

주 항목의 DN은 문자열 표현의 고유 이름이어야 합니다(RFC 2253). 따라서 쉼표와 같이 구문상 DN에 중요한 문자는 역슬래시(\)를 사용하여 이스케이프해야 합니다. 예를 들면 다음과 같습니다.

```
(target="ldap:///uid=cfuentes,o=Example Bolivia\, S.A.")
```

DN에 와일드카드를 사용하여 LDAP URL과 일치하는 항목을 모두 대상으로 지정할 수도 있습니다. 다음은 와일드카드를 올바르게 사용한 예입니다.

- (target="ldap:///uid=*,dc=example,dc=com")

전체 example.com 트리에서 항목 RDN에 uid 속성이 있는 모든 항목이 대상이 됩니다. 깊이에 관계 없이 트리에 있는 모든 항목이 대상이 되며, 예를 들면 다음과 같습니다.

```
uid=tmorris,ou=sales,dc=example,dc=com
uid=yyorgens,ou=marketing,dc=example,dc=com
uid=bjensen,ou=eng,ou=east,dc=example,dc=com
```

- (target="ldap:///uid=*Anderson,ou=People,dc=example,dc=com")

uid가 Anderson으로 끝나는 ou=People 분기의 모든 항목이 대상이 됩니다.

- (target="ldap:/// *Anderson,ou=People,dc=example,dc=com")

이름 지정 속성에 관계 없이 RDN이 Anderson으로 끝나는 ou=People 분기의 모든 항목이 대상이 됩니다.

uid=*,ou=*,dc=example,dc=com에서와 같이 여러 개의 와일드카드를 사용할 수 있습니다. 이 예에서는 example.com 트리에서 uid 및 ou 속성이 포함된 고유 이름을 가진 모든 항목이 대상이 됩니다.

주 고유 이름의 접미어 부분에는 와일드카드를 사용할 수 없습니다. 즉, 디렉토리에 접미어 c=US와 c=GB를 사용한 경우에는 다음과 같은 대상을 사용하여 두 접미어를 참조할 수 *없습니다*.

```
(target="ldap:///dc=example,c=*").
```

```
uid=bjensen,o=*.com과 같은 대상을 사용할 수도 없습니다.
```

대상 속성 지정

디렉토리 항목을 대상으로 지정하는 것 이외에 대상 항목에 있는 하나 이상의 속성 또는 하나 이상의 속성을 제외한 모든 속성을 대상으로 지정할 수도 있습니다. 이 방법은 부분적인 항목 정보에 대한 액세스를 허용하거나 거부할 경우에 유용합니다. 예를 들어 정해진 항목의 이름, 성, 전화 번호 속성에 대해서만 액세스를 허용할 수 있습니다. 또는 개인 데이터와 같은 중요한 정보에 대한 액세스를 거부할 수 있습니다.

targetattr 규칙이 없으면 기본적으로 어떤 속성도 액세스할 수 없습니다. 모든 속성에 액세스하려면 targetattr="*"와 같이 규칙을 설정해야 합니다.

대상으로 지정한 속성이 대상 항목이나 하위 트리에 반드시 포함될 필요는 없지만 ACI는 이 속성이 포함되어 있을 때 적용됩니다. 대상으로 지정한 속성은 스키마에 정의하지 않아도 됩니다. 스키마 검사를 사용하지 않으면 데이터와 스키마를 가져오기 전에 액세스 제어 정책을 구현할 수 있습니다.

대상 속성을 지정하려면 targetattr 키워드를 사용하고 속성 이름을 지정합니다.

targetattr 키워드에 사용되는 구문은 다음과 같습니다.

```
(targetattr = "attribute")
(targetattr != "attribute")
```

targetattr 키워드를 아래 구문으로 사용하면 여러 속성을 대상으로 지정할 수 있습니다.

```
(targetattr = "attribute1 || attribute2... || attributen")
(targetattr != "attribute1 || attribute2... || attributen")
```

예를 들어 항목의 이름, 성, uid 속성을 대상으로 지정하려면 다음과 같은 구문을 사용합니다.

```
(targetattr = "cn || sn || uid")
```

대상 속성에는 지정된 속성의 하위 유형이 모두 포함됩니다. 예를 들어, (targetattr = "locality")는 locality;lang-fr도 대상으로 지정합니다. (targetattr = "locality;lang-fr-ca")와 같이 특별히 하위 유형을 대상으로 지정할 수도 있습니다.

targetattr 규칙에 와일드카드를 사용할 수도 있지만, 특별한 의미가 없으며 성능이 저하될 수 있으므로 사용하지 않는 것이 좋습니다.

대상 항목과 속성 모두 지정

기본적으로 targetattr 키워드가 포함된 ACI는 이 ACI가 위치해 있는 항목을 대상으로 지정합니다. 즉, 다음과 같은 ACI를 가정해 보십시오.

```
aci: (targetattr = "uid") (accessControlRules;)
```

위의 ACI를 ou=Marketing, dc=example, dc=com 항목에 배치하면 전체 Marketing 하위 트리에 ACI가 적용됩니다. 하지만 다음과 같이 target 키워드를 사용하여 명시적으로 대상을 지정할 수도 있습니다.

```
aci: (target="ldap:///uid=*,ou=Marketing,dc=example,dc=com")
(targetattr="uid") (accessControlRules;)
```

target 키워드와 targetattr 키워드를 지정하는 순서는 관계가 없습니다.

LDAP 필터를 사용한 대상 항목 또는 속성 지정

LDAP 필터를 사용하여 특정 조건에 맞는 항목 집합을 대상으로 지정할 수 있습니다. 이렇게 하려면 targetfilter 키워드를 LDAP 필터와 함께 사용합니다. ACI가 포함된 항목의 하위 트리에서 필터와 일치하는 모든 항목에 ACI가 적용됩니다.

targetfilter 키워드 구문은 다음과 같습니다.

```
(targetfilter = "LDAPfilter")
```

여기서 *LDAPfilter*는 표준 LDAP 검색 필터입니다. 필터 구문에 대한 자세한 내용은 [118 페이지의 "LDAP 검색 필터"](#)를 참조하십시오.

예를 들어, 직원을 나타내는 모든 항목에 정규직이나 계약직 상태가 있고, 근무 시간을 나타내는 속성이 상근직 비율로 표시된다고 가정해 보십시오. 계약 직원이나 파트 타임 직원을 나타내는 모든 항목을 대상으로 지정하려면 다음과 같은 필터를 사용합니다.

```
(targetfilter = "(|(status=contractor)(fulltime<=79))")
```

Netscape 확장 필터 구문은 ACI에서 지원되지 않습니다. 예를 들어 다음과 같은 대상 필터는 잘못된 것입니다.

```
(targetfilter = "(locality:fr:=<= Québec)")
```

그러나 국가별 값에 대한 일치 규칙을 설명하는 필터 구문은 지원됩니다. 예를 들면 다음과 같습니다.

```
(targetfilter = "(locality:2.16.840.1.113730.3.3.2.18.1.4:=Québec)")
```

대상 필터는 모든 항목을 ACI의 대상으로 선택합니다. targetfilter 키워드와 targetattr 키워드를 결합하여 대상 항목에 포함된 속성의 부분 집합에 적용되는 ACI를 작성할 수 있습니다.

아래의 LDIF 예를 사용하면 Engineering Admins 그룹의 모든 구성원이 Engineering 업종에 속한 모든 항목의 departmentNumber 및 manager 속성을 수정할 수 있습니다. 이 예에서는 LDAP 필터링을 사용하여 businessCategory 속성이 Engineering으로 설정된 모든 항목을 선택합니다.

```
dn: dc=example,dc=com
objectClass: top
objectClass: organization
aci: (targetattr="departmentNumber || manager")
      (targetfilter="(businessCategory=Engineering)")
      (version 3.0; acl "eng-admins-write"; allow (write)
groupdn ="ldap:///cn=Engineering Admins, dc=example,dc=com";)
```

팁 디렉토리에 분포된 속성과 항목을 대상으로 지정할 경우 LDAP 필터가 유용하긴 하지만 필터는 액세스가 관리되는 객체 이름을 직접 지정하지 않으므로 예측할 수 없는 결과가 발생하기도 합니다. 속성을 추가하거나 삭제하면 필터링된 ACI의 대상 항목 집합이 변경되므로 ACI에서 LDAP 필터를 사용하는 경우에는 ldapsearch 작업에 동일한 필터를 사용하여 필터가 올바른 항목과 속성을 대상으로 지정하는지 확인해야 합니다.

LDAP 필터를 사용한 대상 속성 값 지정

액세스 제어를 사용하여 특정 속성 값을 대상으로 지정할 수 있으므로 속성 값이 ACI에 정의된 조건에 부합되는 경우 속성에 대한 권한을 부여하거나 거부할 수 있습니다. 속성 값을 기준으로 액세스 권한을 부여하거나 거부하는 ACI를 값 기반 ACI라고 합니다.

예를 들어, 조직의 모든 사용자에게 자신의 항목에 있는 nsRoleDN 속성을 수정할 수 있는 권한을 부여할 수 있습니다. 하지만 "최상위 수준의 관리자"와 같은 중요 역할은 자신에게 할당할 수 없도록 설정하려고 합니다. 이 경우 LDAP 필터를 사용하여 속성 값 조건을 만족하는지 검사할 수 있습니다.

값 기반 ACI를 작성하려면 targattrfilters 키워드를 다음과 같은 구문으로 사용해야 합니다.

```
(targattrfilters="Op=attr1:F1 [(&& attr2:F2)*][;Op=attr:F [(&& attr:F)*]"]")
```

여기서,

- Op는 add 또는 delete 작업 중 하나입니다.
- add는 속성 작성 작업을 나타냅니다.
- delete는 속성 삭제 작업을 나타냅니다.
- attr은 대상 속성을 나타냅니다.
- F는 연결된 속성에만 적용되는 필터를 나타냅니다.

항목을 작성하는 경우 필터가 새 항목의 속성에 적용되면 해당 속성의 각 인스턴스가 필터를 만족해야 합니다. 항목을 삭제하는 경우에도 필터가 항목의 속성에 적용되면 해당 속성의 각 인스턴스가 필터를 만족해야 합니다.

항목을 수정하여 속성을 추가하는 경우 해당 속성에 적용되는 추가 필터를 만족해야 하고, 속성을 삭제하는 경우 해당 속성에 적용되는 삭제 필터를 만족해야 합니다. 항목의 개별 속성 값을 바꾸는 경우에는 추가 필터와 삭제 필터를 모두 만족해야 합니다.

예를 들어 다음과 같은 속성 필터를 가정해 보십시오.

```
(targetattrfilters="add=nsroleDN:(!(nsRoleDN=cn=superAdmin)) &&
telephoneNumber:(telephoneNumber=123*)")
```

이 필터를 사용하면 사용자가 superAdmin 역할을 제외한 모든 역할(nsRoleDN 속성)을 자신의 항목에 추가할 수 있으며 123 접두어가 있는 전화 번호를 추가할 수도 있습니다.

주 Directory Server 콘솔에서 값 기반 ACI를 작성할 수는 없습니다.

한 개의 디렉토리 항목을 대상으로 지정

한 개의 항목을 대상으로 지정하는 명시적인 방법은 없지만 다음과 같은 대체 방법을 사용할 수 있습니다.

- 바인드 요청의 사용자 입력을 대상 항목에 저장된 속성 값과 일치시키는 바인드 규칙을 작성합니다. 자세한 내용은 [260페이지의 "값 일치에 따른 액세스 정의"](#)를 참조하십시오.
- targetfilter 키워드를 사용합니다.

targetfilter 키워드를 사용하여 원하는 항목에만 표시되는 속성 값을 지정할 수 있습니다. 예를 들어, Directory Server를 설치하는 동안 다음과 같은 ACI가 작성됩니다.

```
aci: (targetattr="*")(targetfilter=(o=NetscapeRoot))
(version 3.0; acl "Default anonymous access";
allow (read, search) userdn="ldap:///anyone");
```

이 ACI는 `o` 속성이 `NetscapeRoot` 값을 갖는 `o=NetscapeRoot` 항목에만 적용됩니다. 이러한 방법을 사용할 경우 나중에 디렉토리 트리가 변경되면 이에 따라 ACI를 수정해야 하는 단점이 있습니다.

매크로를 사용하여 대상 정의

매크로를 사용하여 ACI의 대상 부분에 DN을 지정함으로써 디렉토리에서 사용되는 ACI의 수를 최적화할 수 있습니다. 자세한 내용은 [305페이지의 "고급 액세스 제어: 매크로 ACI 사용"](#)을 참조하십시오.

권한 정의

권한은 허용 또는 거부되는 액세스 유형을 지정합니다. 디렉토리에서 특정 작업을 수행할 수 있는 권한을 허용하거나 거부할 수 있으며, 지정할 수 있는 다양한 작업도 *권한*이라고 합니다.

권한 설정은 다음 두 단계로 이루어져 있습니다.

- 액세스 허용 또는 거부
- 권한 지정

액세스 허용 또는 거부

디렉토리 트리에 대한 액세스 권한을 명시적으로 허용하거나 거부할 수 있습니다. 액세스를 허용하는 경우와 거부하는 경우에 대한 자세한 지침은 *Directory Server Deployment Planning Guide*의 "Designing Access Control"을 참조하십시오.

권한 지정

권한은 사용자가 디렉토리 데이터에 대해 수행할 수 있는 특정 작업을 자세히 설명합니다. 모든 권한을 허용 또는 거부하거나 다음 중 하나 이상의 권한을 지정할 수 있습니다.

읽기. 사용자가 디렉토리 데이터를 읽을 수 있는지 여부를 나타냅니다. 이 권한은 검색 작업에만 적용됩니다.

쓰기. 사용자가 속성을 추가, 수정 또는 삭제하여 항목을 수정할 수 있는지 여부를 나타냅니다. 이 권한은 수정 작업과 `modrdn` 작업에 적용됩니다.

추가. 사용자가 항목을 작성할 수 있는지 여부를 나타냅니다. 이 권한은 추가 작업에만 적용됩니다.

삭제. 사용자가 항목을 삭제할 수 있는지 여부를 나타냅니다. 이 권한은 삭제 작업에만 적용됩니다.

검색. 사용자가 디렉토리 데이터를 검색할 수 있는지 여부를 나타냅니다. 사용자는 검색 및 읽기 권한이 있어야만 검색 결과의 일부로 반환된 데이터를 볼 수 있습니다. 이 권한은 검색 작업에만 적용됩니다.

비교. 사용자가 자신이 제공한 데이터와 디렉토리에 저장된 데이터를 비교할 수 있는지 여부를 나타냅니다. 비교 권한이 있을 경우 디렉토리에서 조회에 응답하여 성공 또는 실패 메시지를 반환하지만 사용자가 항목이나 속성 값을 볼 수는 없습니다. 이 권한은 비교 작업에만 적용됩니다.

자체 쓰기. 사용자가 대상 항목의 속성에 자신의 DN을 추가하거나 삭제할 수 있는지 여부를 나타냅니다. 이 속성의 구문은 "고유 이름"이어야 합니다. 이 권한은 그룹 관리에만 사용됩니다. 자체 쓰기는 프록시 인증과 함께 작동하여, 바인딩된 사용자의 DN이 아닌 프록시 DN을 그룹 항목에 추가하거나 삭제할 수 있는 권한을 부여합니다.

프록시. 지정된 DN이 다른 항목의 권한을 사용하여 대상에 액세스할 수 있는지 여부를 나타냅니다. 디렉토리 관리자 DN을 제외하고 디렉토리의 모든 사용자 DN을 사용한 프록시 액세스 권한을 부여할 수 있습니다. 디렉토리 관리자에게는 프록시 권한을 부여할 수 없습니다. [294페이지의 "프록시 인증 ACI 예"](#)에 자세한 예가 나와 있습니다.

가져오기. 이 액세스 권한은 수정 DN 작업에서 사용됩니다. 이 권한은 지정된 DN에 항목을 가져올 수 있는지 여부를 나타냅니다.

내보내기. 이 액세스 권한은 수정 DN 작업에서 사용됩니다. 이 권한은 지정된 DN에서 항목을 내보낼 수 있는지 여부를 나타냅니다.

모두. 대상으로 지정한 항목에 읽기, 쓰기, 검색, 삭제, 비교 및 자체 쓰기 등의 권한이 지정된 DN에 있음을 나타냅니다. 이 액세스 권한(모두)은 대상으로 지정한 항목에 프록시, 가져오기 및 내보내기 권한을 제공하지 않습니다.

권한은 서로 독립적으로 부여됩니다. 예를 들어, 추가 권한을 가진 사용자는 항목을 작성할 수 있지만 삭제 권한이 특별히 부여되지 않은 경우 항목을 삭제할 수는 없습니다. 따라서 디렉토리에 대한 액세스 제어 정책을 계획할 때는 사용자에게 합리적인 방식으로 권한을 부여해야 합니다. 예를 들어, 읽기 권한과 검색 권한을 부여하지 않고 쓰기 권한만 부여하는 것은 합리적이지 않습니다.

LDAP 작업에 필요한 권한

이 절에서는 사용자에게 허용하려는 LDAP 작업 유형에 따라 부여해야 하는 권한에 대해 설명합니다.

항목 추가

- 추가할 항목에 대한 추가 권한을 부여합니다.
- 항목의 각 속성 값에 대한 쓰기 권한을 부여합니다. 이 권한은 기본적으로 부여되지만 `targattrfilters` 키워드를 사용하여 제한할 수 있습니다.

항목 삭제

- 삭제할 항목에 대한 삭제 권한을 부여합니다.
- 항목의 각 속성 값에 대한 쓰기 권한을 부여합니다. 이 권한은 기본적으로 부여되지만 `targattrfilters` 키워드를 사용하여 제한할 수 있습니다.

항목의 속성 수정

- 속성 유형에 대한 쓰기 권한을 부여합니다.
- 각 속성 유형 값에 대한 쓰기 권한을 부여합니다. 이 권한은 기본적으로 부여되지만 `targattrfilters` 키워드를 사용하여 제한할 수 있습니다.

항목의 RDN 수정

- 항목에 대한 쓰기 권한을 부여합니다.
- 새 RDN에 사용된 속성 유형에 대한 쓰기 권한을 부여합니다.
- 이전 RDN을 삭제할 수 있는 권한을 부여하려면 이전 RDN에 사용된 속성 유형에 대한 쓰기 권한을 부여합니다.
- 새 RDN에 사용된 속성 유형 값에 대한 쓰기 권한을 부여합니다. 이 권한은 기본적으로 부여되지만 `targattrfilters` 키워드를 사용하여 제한할 수 있습니다.

다른 하위 트리로 항목 이동

- 이동하려는 항목에 내보내기 권한을 부여합니다.
- 이동하려는 항목의 새 상위 항목에 가져오기 권한을 부여합니다.

속성 값 비교

- 속성 유형에 대한 비교 권한을 부여합니다.

항목 검색

- 검색 필터에 사용된 각 속성 유형에 대한 검색 권한을 부여합니다.
- 해당 항목이 반환되는 것을 확인하기 위해 항목에 사용하는 하나 이상의 속성 유형에 대한 읽기 권한을 부여합니다.
- 항목과 함께 반환되는 각 속성 유형에 읽기 권한을 부여합니다.

예를 통해 살펴보면 사용자가 디렉토리를 검색할 수 있도록 허용하기 위해 설정해야 하는 권한을 명확히 이해할 수 있습니다. 아래 검색을 가정해 보십시오.

```
(target="ldap:///dc=example,dc=com")
ldapsearch -h host -p port -D "uid=bjensen,dc=example,dc=com" \
-w password -b "dc=example,dc=com" \
"(objectclass=*)" mail
```

bjensen 사용자에게 자신의 항목 검색을 위한 액세스 권한을 부여할 수 있는지 확인하려면 아래 ACI를 사용합니다.

```
aci: (targetattr = "mail")(version 3.0; acl "self access to \
mail"; allow (read, search) userdn = "ldap:///self");
```

이 ACI는 `objectclass` 속성에 대한 검색 권한을 bjensen에 부여하지 않기 때문에 빈 검색 결과 목록이 표시됩니다. 위에서 설명한 검색 작업이 성공하려면 ACI를 다음과 같이 수정해야 합니다.

```
aci: (targetattr = "mail || objectclass")(version 3.0; acl \
"self access to mail"; allow (read, search) userdn = \
"ldap:///self");
```

권한 구문

ACI 명령문의 권한 구문은 다음과 같습니다.

```
allow|deny (rights)
```

여기서 *rights*는 쉼표로 구분된 키워드의 목록으로 각 키워드는 괄호로 묶여 있습니다. 유효한 키워드는 **read**, **write**, **add**, **delete**, **search**, **compare**, **selfwrite**, **proxy import**, **export** 또는 **all**입니다.

all 액세스 권한은 대상 항목에 **proxy**, **import** 및 **export** 권한을 제공하지 않습니다.

아래 예에서는 바인드 규칙이 **true**일 경우 읽기, 검색 및 비교 액세스가 허용됩니다.

```
aci: (target="ldap:///dc=example,dc=com") (version 3.0;acl \
"example"; allow (read, search, compare) bindRule);
```

바인드 규칙

특정 작업의 경우 디렉토리에 정의된 ACI에 따라 디렉토리에 바인드해야 합니다. 바인드란 바인드 DN과 비밀번호(SSL을 사용하는 경우에는 인증서)를 제공하여 디렉토리에 로그인하거나 자체 인증하는 것을 의미합니다. 바인드 작업에 제공된 자격 증명과 바인드 환경에 따라 디렉토리에 대한 액세스의 허용 여부가 결정됩니다.

ACI에 설정된 각 권한에는 필수 인증서와 바인드 매개 변수를 자세히 설명하는 해당 바인드 규칙이 있습니다.

단순한 바인드 규칙은 디렉토리에 액세스하는 사람이 특정 그룹에 속해야 한다고 지정할 수 있습니다. 복잡한 바인드 규칙은 이 사람이 특정 그룹에 속해야 하며 오전 8시부터 오후 5시 사이에 특정 IP 주소의 시스템에서 로그인해야 한다고 지정할 수 있습니다.

바인드 규칙은 디렉토리에 액세스할 수 있는 사람, 액세스 시기 및 장소를 정의합니다. 보다 구체적으로, 바인드 규칙은 다음과 같은 항목을 지정할 수 있습니다.

- 액세스 권한이 있는 사용자, 그룹 및 역할
- 엔티티의 바인드 위치. 사용자가 인증하는 위치는 위조될 수 있으므로 신뢰할 수 없습니다. 이 정보에만 기반하여 ACI를 작성하지 마십시오.
- 바인드 시간 또는 요일
- 바인드 중에 사용해야 하는 인증 유형

또한 부울 연산자로 이러한 조건을 결합하여 복잡한 바인드 규칙을 구성할 수 있습니다. 자세한 내용은 [269페이지의 "부울 바인드 규칙 사용"](#)을 참조하십시오.

서버는 RFC 2251 *Lightweight Directory Access Protocol(v3)*에 설명된 것처럼 LDAP 필터 평가에 사용되는 논리와 유사한 3개 값 논리에 따라 ACI에 사용된 논리 표현식을 평가합니다. 즉, 자원 제한으로 인해 표현식 평가가 중단된 경우처럼 표현식의 구성 요소가 정의되지 않음으로 평가되어도 서버에서 이를 올바르게 처리합니다. 정의되지 않음 값이 복잡한 부울 표현식에서 발생했으므로 액세스 권한이 잘못 부여되지는 않습니다.

바인드 규칙 구문

액세스 허용 여부는 ACI의 바인드 규칙이 true인지 여부에 따라 결정됩니다. 바인드 규칙은 다음 두 가지 패턴 중 하나를 사용합니다.

```
keyword = "expression";
```

```
keyword != "expression";
```

여기서 같음(=)은 바인드 규칙이 **true**가 되려면 *keyword*와 *expression*이 일치해야 한다는 것을 나타내고, 같지 않음(!=)은 *keyword*와 *expression*이 일치하지 않아야 한다는 것을 나타냅니다.

주 *timeofday* 키워드는 부등식(<, <=, >, >=)도 지원합니다. 부등식은 이 키워드에서만 지원됩니다.

*expression*은 인용 부호(" ")로 묶고 세미콜론(;)으로 구분해야 합니다. 사용할 수 있는 표현식은 관련 *keyword*에 따라 달라집니다.

아래 표에는 각 키워드 및 관련 표현식이 나와 있으며 표현식에 와일드카드를 사용할 수 있는지 여부도 표시되어 있습니다.

표 6-2 LDIF 바인드 규칙 키워드

키워드	유효한 표현식	와일드카드 허용 여부
userdn	ldap:///distinguished_name ldap:///all ldap:///anyone ldap:///self ldap:///parent ldap:///suffix??sub?(filter)	예(DN에서만)
groupdn	[ldap:///DN]	아니요
roledn	[ldap:///DN]	아니요
userattr	attribute#bindType 또는 attribute#value	아니요
ip	IP_address	예
dns	DNS_host_name	예
dayofweek	sun mon tue wed thu fri sat	아니요
timeofday	0 - 2359	아니요
authmethod	none simple ssl sasl authentication_method	아니요

다음 절에서는 각 키워드의 바인드 규칙 구문에 대해 자세히 설명합니다.

사용자 액세스 정의 - userdn 키워드

사용자 액세스는 userdn 키워드를 사용하여 정의됩니다. userdn 키워드에는 다음 형식의 유효한 고유 이름이 하나 이상 있어야 합니다.

```
userdn = "ldap:///dn [| ldap:///dn]..."
userdn != "ldap:///dn [| ldap:///dn]..."
```

여기서 *dn*은 DN일 수도 있고 anyone, all, self 또는 parent 표현식 중 하나일 수도 있습니다. 표현식은 다음과 같은 사용자를 참조합니다.

- userdn = "ldap:///anyone" - 익명 사용자와 인증된 사용자를 모두 참조합니다.
- userdn = "ldap:///all" - 인증된 사용자만 참조합니다.
- userdn = "ldap:///self" - ACI의 대상 항목과 동일한 사용자만 참조합니다.
- userdn = "ldap:///parent" - ACI 대상의 부모 항목만 참조합니다.

userdn 키워드는 다음과 같은 형식의 LDAP 필터로 표현할 수도 있습니다.

```
userdn = ldap:///suffix??sub?(filter)
```

주 접표와 같이 구문상 DN에 중요한 문자는 역슬래시(\)를 사용하여 이스케이프해야 합니다.

익명 액세스(anyone 키워드)

디렉토리에 대한 익명 액세스를 부여하면 바인드 DN이나 비밀번호를 제공하지 않아도 바인드 상황에 관계 없이 모든 사람이 디렉토리에 액세스할 수 있습니다. 익명 액세스를 특정 유형의 액세스(예: 읽기 액세스 또는 검색 액세스)로 제한하거나 디렉토리의 개별 항목이나 특정 하위 트리로 제한할 수 있습니다. anyone 키워드를 사용한 익명 액세스는 인증된 모든 사용자의 액세스도 허용합니다.

예를 들어, 전체 example.com 트리에 대한 익명 읽기 및 검색 액세스를 허용하려면 dc=example,dc=com 노드에 아래 ACI를 작성합니다.

```
aci: (version 3.0; acl "anonymous-read-search";
      allow (read, search) userdn = "ldap:///anyone");
```


일반 액세스(all 키워드)

바인드 규칙을 사용하여 특정 권한이 디렉토리에 성공적으로 바인드한 모든 사람에게 적용됨을 나타낼 수 있습니다. 이 경우 all 키워드는 인증된 모든 사용자의 액세스를 허용하므로 익명 액세스를 방지하는 동시에 일반 액세스를 허용할 수 있습니다.

예를 들어, 모든 인증된 사용자에게 전체 트리에 대한 읽기 액세스 권한을 부여하려면 dc=example,dc=com 노드에 아래 ACI를 작성합니다.

```
aci: (version 3.0; acl "all-read"; allow (read)
      userdn="ldap:///all");
```

자체 액세스(self 키워드)

사용자에게 자신의 항목에 대한 액세스 권한이 부여되거나 거부되도록 지정합니다. 이 경우 바인드 DN이 대상 항목의 DN과 일치하면 액세스 권한이 부여되거나 거부됩니다.

예를 들어, example.com 트리의 모든 사용자에게 userPassword 속성에 대한 쓰기 액세스 권한을 부여하려면 dc=example,dc=com 노드에 아래 ACI를 작성합니다.

```
aci: (targetattr = "userPassword") (version 3.0; acl
      "modify own password"; allow (write) userdn = "ldap:///self");
```

부모 액세스(parent 키워드)

바인드 DN이 대상 항목의 부모인 경우에만 사용자에게 항목에 대한 액세스 권한이 부여되거나 거부되도록 지정합니다. parent 키워드를 사용하려면 서버 콘솔에서 수동으로 ACI를 편집해야 합니다.

예를 들어, 사용자가 바인드 DN의 모든 자식 항목을 수정할 수 있게 하려면 dc=example,dc=com 노드에 아래 ACI를 작성합니다.

```
aci: (version 3.0; acl "parent access";
      allow (write) userdn="ldap:///parent");
```

LDAP URL

다음과 같이 URL을 필터와 함께 사용하여 ACI에서 동적으로 사용자를 지정할 수 있습니다.

```
userdn = "ldap:///<suffix>??sub?(filter) "
```

예를 들어, example.com 트리의 accounting 분기와 engineering 분기에 있는 모든 사용자는 다음 URL을 기준으로 대상 자원에 대한 액세스 권한이 동적으로 부여되거나 거부됩니다.

```
userdn = "ldap:///dc=example,dc=com??sub?(|(ou=eng)(ou=acct)) "
```

주 LDAP URL에 호스트 이름이나 포트 번호를 지정하지 마십시오. LDAP URL은 항상 로컬 서버에 적용됩니다.

LDAP URL에 대한 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.

와일드카드

와일드카드 문자(*)를 사용하여 사용자 집합을 지정할 수도 있습니다. 예를 들어, 사용자 DN을 `uid=b*,dc=example,dc=com`으로 지정하면 바인드 DN이 `b` 문자로 시작하는 사용자에게만 설정된 권한을 기준으로 액세스 권한이 부여되거나 거부됩니다.

LDAP URL의 논리적 OR

사용자 접근에 대한 복잡한 규칙을 작성하려면 여러 개의 LDAP URL이나 키워드 표현식을 지정합니다. 예를 들면 다음과 같습니다.

```
userdn = "ldap:///uid=b*,c=example.com ||
ldap:///cn=b*,dc=example,dc=com";
```

사용자가 DN 패턴 중 하나를 사용하여 바인드하면 바인드 규칙은 `true`가 됩니다.

특정 LDAP URL 제외

같지 않음(!=) 연산자를 사용하여 특정 URL이나 DN을 제외하는 사용자 접근을 정의합니다. 예를 들면 다음과 같습니다.

```
userdn != "ldap:///uid=*,ou=Accounting,dc=example,dc=com";
```

클라이언트가 `accounting` 하위 트리에 있는 UID 기반의 고유 이름으로 바인드하지 않으면 바인드 규칙은 `true`가 됩니다. 이 바인드 규칙은 대상 항목이 디렉토리 트리에서 `accounting` 분기 이외의 위치에 있는 경우에만 적용됩니다.

그룹 액세스 정의 - groupdn 키워드

특정 그룹의 구성원이 대상 자원에 액세스하는 것을 *그룹 액세스*라고 합니다. 사용자가 특정 그룹에 속한 DN을 사용하여 바인드할 경우 대상 항목에 대한 액세스가 부여되거나 거부되도록 지정하려면 groupdn 키워드를 사용하여 그룹 액세스를 정의합니다.

groupdn 키워드에는 한 개 이상 그룹의 고유 이름이 다음과 같은 형식으로 필요합니다.

```
groupdn="ldap:///groupDN [| ldap:///groupDN]..."
```

바인드 DN이 groupDNs으로 지정된 그룹에 속하면 바인드 규칙은 true가 됩니다. 다음 절에서는 groupdn 키워드를 사용한 예를 제공합니다.

주 참표와 같이 구문상 DN에 중요한 문자는 역슬래시(\)를 사용하여 이스케이프해야 합니다.

단일 LDAP URL

```
groupdn = "ldap:///cn=Administrators,dc=example,dc=com";
```

바인드 DN이 Administrators 그룹에 속하면 바인드 규칙은 true가 됩니다. Administrators 그룹에 전체 디렉토리 트리에 대한 쓰기 권한을 부여하려면 dc=example,dc=com 노드에 아래 ACI를 작성합니다.

```
aci: (version 3.0; acl "Administrators-write"; allow (write)
groupdn="ldap:///cn=Administrators,dc=example,dc=com");)
```

LDAP URL의 논리적 OR

```
groupdn = "ldap:///cn=Administrators,dc=example,dc=com [|
ldap:///cn=Mail Administrators,dc=example,dc=com";
```

바인드 DN이 Administrators 그룹이나 Mail Administrators 그룹에 속하면 바인드 규칙은 true가 됩니다.

역할 액세스 정의 - roledn 키워드

특정 역할의 구성원이 대상 자원에 액세스하는 것을 *역할 액세스*라고 합니다. 사용자가 특정 역할에 속한 DN을 사용하여 바인드하는 경우 대상 항목에 대한 액세스 권한이 부여되거나 거부되도록 지정하려면 roledn 키워드를 사용하여 역할 액세스를 정의합니다.

roledn 키워드에는 다음 형식의 유효한 고유 이름이 하나 이상 있어야 합니다.

```
roledn = "ldap:///dn [| ldap:///dn]... [| ldap:///dn]"
```

바인드 DN이 지정된 역할에 속하면 바인드 규칙은 true가 됩니다.

주 접표와 같이 구문상 DN에 중요한 문자는 역슬래시(\)를 사용하여 이스케이프해야 합니다.

roledn 키워드는 groupdn 키워드와 동일한 구문으로 사용됩니다.

값 일치에 따른 액세스 정의

디렉토리에 바인드하는 데 사용되는 항목의 속성 값이 대상 항목의 속성 값과 일치하도록 지정하는 바인드 규칙을 설정할 수 있습니다.

예를 들어, 바인드 DN이 사용자 항목의 manager 속성에 있는 DN과 일치할 경우에만 ACI를 적용하도록 지정할 수 있습니다. 이 경우 사용자의 관리자만 항목에 액세스할 수 있습니다.

이 예는 DN 일치를 기준으로 하지만 바인드에 사용된 항목의 모든 속성을 대상 항목과 일치시킬 수 있습니다. 예를 들어, favoriteDrink 속성이 "beer"인 사용자가 동일한 favoriteDrink 값을 갖는 다른 사용자의 모든 항목을 읽을 수 있도록 허용하는 ACI를 작성할 수 있습니다.

userattr 키워드 사용

userattr 키워드를 사용하여 바인드에 사용된 항목과 대상 항목 간에 일치시킬 속성 값을 지정할 수 있습니다.

다음과 같은 속성 값을 지정할 수 있습니다.

- 사용자 DN
- 그룹 DN
- 역할 DN

- LDAP URL의 LDAP 필터
- 모든 속성 유형

userattr 키워드의 LDIF 구문은 다음과 같습니다.

```
userattr = "attrName#bindType"
```

사용자 DN, 그룹 DN, 역할 DN이나 LDAP 필터 이외의 값이 필요한 속성 유형을 사용하는 경우 LDIF 구문은 다음과 같습니다.

```
userattr = "attrName#attrValue"
```

여기서,

- *attrName*은 값 일치에 사용되는 속성 이름입니다.
- *bindType*은 USERDN, GROUPODN, ROLEDN, LDAPURL 중 하나입니다.
- *attrValue*는 속성 값을 나타내는 문자열입니다.

주 서비스 클래스(CoS) 정의에서 생성된 속성은 userattr 키워드와 함께 사용할 수 없습니다. 이 경우 CoS에서 생성된 속성 값을 사용하는 바인드 규칙이 포함된 ACI가 작동하지 않습니다.

다음 절에는 userattr 키워드를 다양한 바인드 유형과 함께 사용하는 예를 소개합니다.

USERDN 바인드 유형을 사용한 예

사용자 DN을 기준으로 하는 바인드에 사용된 userattr 키워드의 예는 다음과 같습니다.

```
userattr = "manager#USERDN"
```

바인드 DN이 대상 항목의 manager 속성 값과 일치하면 바인드 규칙은 true가 됩니다. 이 기법을 사용하면 사용자의 관리자가 직원의 속성을 수정할 수 있습니다. 이 기법은 대상 항목의 manager 속성이 전체 DN으로 표시된 경우에만 작동합니다.

아래 예에서는 관리자에게 직원 항목에 대한 전체 액세스 권한을 부여합니다.

```
aci: (target="ldap:///dc=example,dc=com") (targetattr="*")
  (version 3.0;acl "manager-write";
  allow (all) userattr = "manager#USERDN");
```

GROUPDN 바인드 유형을 사용한 예

그룹 DN을 기준으로 하는 바인드에 사용된 userattr 키워드의 예는 다음과 같습니다.

```
userattr = "owner#GROUPDN"
```

바인드 DN이 대상 항목의 owner 속성에 지정된 그룹의 구성원이면 바인드 규칙은 true가 됩니다. 예를 들어, 이 기법을 사용하여 그룹이 직원의 상태 정보를 관리하도록 허용할 수 있습니다. 속성에 그룹 항목의 DN이 포함되어 있으면 owner 이외의 속성을 사용할 수도 있습니다.

사용자가 가리키는 그룹은 동적 그룹일 수 있으며, 그룹 DN은 디렉토리의 모든 접미어에 위치할 수 있습니다. 하지만 서버에서 이 유형의 ACI를 평가하려면 많은 자원이 필요합니다.

대상 항목과 동일한 접미어에 있는 정적 그룹을 사용하는 경우 다음과 같은 표현식을 사용할 수 있습니다.

```
userattr = "ldap:///dc=example,dc=com?owner#GROUPDN"
```

이 예에서 그룹 항목은 dc=example,dc=com 접미어에 있습니다. 이 유형의 구문은 이전에보다 서버에서 빨리 처리될 수 있습니다.

ROLEDN 바인드 유형을 사용한 예

역할 DN을 기준으로 하는 바인드에 사용된 userattr 키워드의 예는 다음과 같습니다.

```
userattr = "exampleEmployeeReportsTo#ROLEDN"
```

바인드 DN이 대상 항목의 exampleEmployeeReportsTo 속성에 지정된 역할에 속하면 바인드 규칙은 true가 됩니다. 예를 들어, 회사의 모든 관리자에 대해 증첩된 역할을 작성할 경우 이 기법을 사용하여 모든 수준의 관리자가 자신보다 등급이 낮은 직원에 대한 정보에 액세스할 수 있도록 액세스 권한을 부여할 수 있습니다.

역할 DN은 디렉토리의 모든 접미어에 위치할 수 있습니다. 또한 필터링된 역할을 사용하는 경우 이 유형의 ACI를 평가하려면 많은 서버 자원이 필요합니다.

LDAPURL 바인드 유형을 사용한 예

LDAP 필터를 기준으로 하는 바인드에 사용된 userattr 키워드의 예는 다음과 같습니다.

```
userattr = "myfilter#LDAPURL"
```

바인드 DN이 대상 항목의 myfilter 속성에 지정된 필터와 일치하면 바인드 규칙은 true가 됩니다. myfilter 속성을 LDAP 필터가 포함된 속성으로 바꿀 수 있습니다.

속성 값이 있는 예

속성 값을 기준으로 하는 바인드에 사용된 `userattr` 키워드의 예는 다음과 같습니다.

```
userattr = "favoriteDrink#Beer"
```

바인드 DN과 대상 DN에 포함된 `favoriteDrink` 속성 값이 **Beer**이면 바인드 규칙은 `true`가 됩니다.

상속 기능과 함께 userattr 키워드 사용

`userattr` 키워드를 사용하여 바인드에 사용된 항목을 대상 항목과 연결하면 **ACI**는 지정된 대상에만 적용되고 하위 항목에는 적용되지 않습니다. **ACI**의 응용을 대상 항목보다 몇 수준 아래까지 확장하려는 경우 `parent` 키워드를 사용하고 **ACI**를 상속할 대상 아래의 수준 수를 지정합니다.

`userattr` 키워드를 `parent` 키워드와 함께 사용하는 구문은 다음과 같습니다.

```
userattr = "parent[inheritance_level].attribute#bindType"
```

여기서

- `inheritance_level`은 **ACI**를 상속할 대상 아래의 수준 수를 나타내는 점표로 구분된 목록입니다. 대상 항목 아래에 5개 수준([0, 1, 2, 3, 4])을 포함할 수 있으며 제로(0)는 대상 항목을 나타냅니다.
- `attribute`는 `userattr` 키워드나 `groupattr` 키워드로 지정되는 대상 속성입니다.
- `bindType`은 `USERDN`이나 `GROUPDN`입니다. `LDAPURL`과 `ROLEDN` 바인드 유형에는 상속 기능이 지원되지 않습니다.

예를 들면 다음과 같습니다.

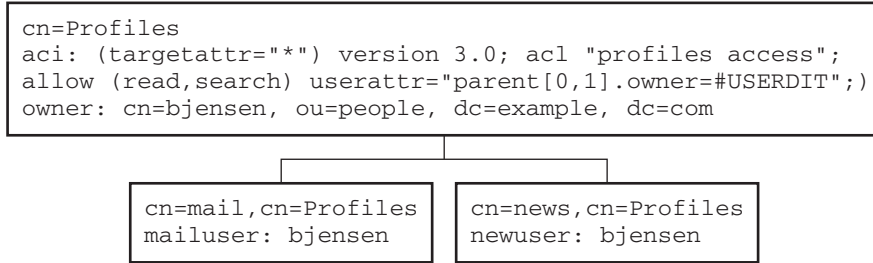
```
userattr = "parent[0,1].manager#USERDN"
```

바인드 DN이 대상 항목의 `manager` 속성과 일치하면 이 바인드 규칙은 `true`가 됩니다. 바인드 규칙이 `true`로 평가될 때 부여된 권한은 대상 항목 및 바로 아래의 모든 항목에 적용됩니다.

userattr 상속을 사용한 예

아래 그림의 예에서는 bjensen 사용자가 cn=Profiles 항목 및 cn=mail과 cn=news가 포함된 첫 수준의 자식 항목을 읽고 검색할 수 있음을 나타냅니다.

그림 6-1 상속 기능과 함께 userattr 키워드 사용



이 예에서 상속을 사용하지 않은 경우 동일한 결과를 얻으려면 다음 중 하나를 수행해야 합니다.

- 디렉토리의 cn=Profiles, cn=mail 및 cn=news 항목에 bjensen 사용자에게 대해 읽기 및 검색 액세스 권한을 명시적으로 설정합니다.
- owner 속성과 다음 ACI를 cn=mail, cn=Profiles 항목과 cn=news, cn=Profiles 항목에 추가합니다.

```
aci: (targetattr="*") (version 3.0; acl "profiles access"; allow
(read,search) userattr="owner#USERDN");)
```

userattr 키워드를 사용한 추가 권한 부여

userattr 키워드를 all 또는 add 권한과 함께 사용하면 서버가 예상과 달리 작동할 수 있습니다. 일반적으로 Directory Server는 디렉토리에 새 항목을 작성할 때 부모 항목이 아닌 작성되는 항목에 대한 액세스 권한을 평가합니다. 하지만 userattr 키워드를 사용하는 ACI의 경우 이 동작으로 인해 보안 허점이 생길 수 있으므로 정상적인 서버 동작이 수정됩니다.

아래 예를 가정에 보십시오.

```
aci: (target="ldap:///dc=example,dc=com") (targetattr="*")
(version 3.0; acl "manager-write"; allow (all)
userattr = "manager#USERDN");)
```


이 ACI는 관리자에게 보고하는 직원 항목에 대한 모든 권한을 관리자에게 부여합니다. 하지만 이 ACI 유형을 사용할 경우 작성되는 항목에 대해서만 액세스 권한이 평가되므로 모든 직원이 자신의 DN으로 설정된 manager 속성을 가진 항목을 작성할 수도 있습니다. 예를 들어, 불만을 품은 직원인 Joe(cn=Joe, ou=eng, dc=example, dc=com)는 트리의 Human Resources 분기에 항목을 작성하여 Human Resources 직원에게 부여된 권한을 사용(또는 남용)할 수 있습니다.

이 경우 다음과 같은 항목만 작성하면 됩니다.

```
dn: cn= Trojan Horse,ou=Human Resources,dc=example,dc=com
objectclass: top
...
cn: Trojan Horse
manager: cn=Joe,ou=eng,dc=example,dc=com
```

이러한 보안 문제를 방지하기 위해 ACI 평가 프로세스는 수준 0, 즉 항목 자체에 대한 추가 권한을 부여하지 않습니다. 하지만 parent 키워드를 사용하여 기존 항목 아래에 추가 권한을 부여할 수 있습니다. 이 경우 추가 권한에 대한 부모 아래의 수준 수를 지정해야 합니다. 예를 들어, 아래 ACI에서는 manager 속성이 바인드 DN과 일치하는 dc=example,dc=com의 모든 항목에 자식 항목을 추가할 수 있습니다.

```
aci: (target="ldap:///dc=example,dc=com") (targetattr="*")
(version 3.0; acl "parent-access"; allow (add)
userattr = "parent[0,1].manager#USERDN");
```

이 ACI를 사용하면 바인드 DN이 부모 항목의 manager 속성과 일치하는 사용자에게만 추가 권한이 부여됩니다.

특정 IP 주소로부터의 액세스 정의

바인드 규칙을 사용하여 바인드 작업이 반드시 특정 IP 주소로부터 시작되도록 지정할 수 있습니다. 이 액세스 정의는 주로 특정 시스템이나 네트워크 도메인에서 모든 디렉토리 업데이트를 수행하도록 강제하는 데 사용됩니다.

IP 주소를 기준으로 바인드 규칙을 설정하는 LDIF 구문은 다음과 같습니다.

```
ip = "IPaddressList" or ip != "IPaddressList"
```

*IPaddressList*는 다음 중 하나 이상의 요소가 쉼표로 구분된 목록입니다.

- 특정 IPv4 주소: 123.45.6.7
- 와일드카드를 사용하여 서브네트워크를 지정한 IPv4 주소: 12.3.45.*

- 서브네트워크 마스크가 있는 IPv4 주소 또는 서브네트워크:
123.45.6.*+255,255,255,192
- RFC 2373(<http://www.ietf.org/rfc/rfc2373.txt>) 및 RFC 2732(<http://www.ietf.org/rfc/rfc2732.txt>)에 정의되어 있는 대로, 정확한 형식으로 되어 있고 대괄호 "[" 및 "]"가 있는 IPv6 주소. 다음은 모두 동일한 주소를 나타냅니다.
 - ldap://[12AB:0000:0000:CD30:0000:0000:0000]
 - ldap://[12AB::CD30:0:0:0:0]
 - ldap://[12AB:0:0:CD30::]
- 서브넷 접두어 길이가 있는 IPv6 주소:
ldap://[12AB::CD30:0:0:0:0]/60

디렉토리에 액세스하는 클라이언트가 지정된 IP 주소에 위치해 있으면 바인드 규칙은 true가 됩니다. 이 기법은 특정 서브넷 또는 시스템으로부터의 특정 디렉토리 액세스만 허용하는 경우에 유용합니다. 사용자가 인증하는 IP 주소는 위조될 수 있으므로 신뢰할 수 없습니다. 이 정보에만 기반하여 ACI를 작성하지 마십시오.

서버 콘솔의 액세스 제어 편집기를 사용하여 ACI가 적용되는 특정 시스템을 정의할 수 있습니다. 자세한 내용은 271페이지의 "콘솔에서 ACI 작성"을 참조하십시오.

특정 도메인으로부터의 액세스 정의

바인드 규칙은 바인드 작업이 반드시 특정 도메인이나 호스트 시스템으로부터 시작되도록 지정할 수 있습니다. 이 액세스 정의는 주로 특정 시스템이나 네트워크 도메인에서 모든 디렉토리 업데이트를 수행하도록 강제하는 데 사용됩니다.

DNS 호스트 이름을 기준으로 바인드 규칙을 설정하는 LDIF 구문은 다음과 같습니다.

```
dns = "DNS_Hostname" or dns != "DNS_Hostname"
```

주의 dns 키워드를 사용하려면 시스템에서 DNS를 이름 지정 서비스로 사용해야 합니다. DNS를 이름 지정 서비스로 사용하지 않는 경우에는 ip 키워드를 사용해야 합니다.

dns 키워드에는 전체 DNS 도메인 이름을 지정해야 합니다. 도메인을 지정하지 않고 호스트에 대한 액세스 권한을 부여하면 보안 문제가 발생할 수 있습니다. 예를 들어, 아래 표현식은 사용할 수는 있지만 바람직하지 않습니다.

```
dns = "legend.eng";
```

이 경우 다음과 같은 정규화된 이름을 사용해야 합니다.

```
dns = "legend.eng.example.com";
```

`dns` 키워드에는 와일드카드를 사용할 수 있습니다. 예를 들면 다음과 같습니다.

```
dns = "*.example.com";
```

디렉토리에 액세스하는 클라이언트가 지정된 도메인에 위치해 있으면 바인드 규칙은 `true`가 됩니다. 이 기법은 특정 도메인으로부터의 액세스만 허용하는 경우에 유용합니다. 시스템에서 DNS 이외의 이름 지정 서비스를 사용하면 와일드카드는 작동하지 않습니다. 이 경우 액세스를 특정 도메인으로 제한하려면 [265페이지의 "특정 IP 주소로부터의 액세스 정의"](#)에 설명된 것처럼 `ip` 키워드를 사용합니다.

특정 시간 또는 요일의 액세스 정의

특정 시간이나 특정 요일에만 바인드할 수 있도록 바인드 규칙을 지정할 수 있습니다. 예를 들어, 월요일부터 금요일까지 오전 8시와 오후 5시 사이에만 액세스를 허용하는 규칙을 설정할 수 있습니다. 액세스 권한 평가에 사용되는 시간은 클라이언트 시간이 아닌 Directory Server 시간입니다.

시간을 기준으로 바인드 규칙을 설정하는 LDIF 구문은 다음과 같습니다.

```
timeofday operator "time"
```

여기서 *operator*는 같음(=), 같지 않음(!=), 보다 큼(>), 크거나 같음(>=), 보다 작음(<), 작거나 같음(<=) 중 하나입니다. `time`은 24시간 시계의 시간과 분을 나타내는 4자리로 표현됩니다(0 - 2359). 예를 들면 다음과 같습니다.

- `timeofday = "1200"`;은 시스템 클럭이 정오를 가리키는 1분 동안 클라이언트가 디렉토리에 액세스할 때 `true`가 됩니다.
- `timeofday != "0100"`;은 오전 1시 이외의 시간에 액세스할 때 `true`가 됩니다.
- `timeofday > "0800"`;은 오전 8:01부터 오후 11:59까지의 시간에 액세스할 때 `true`가 됩니다.
- `timeofday >= "0800"`;은 오전 8:00부터 오후 11:59까지의 시간에 액세스할 때 `true`가 됩니다.
- `timeofday < "1800"`;은 자정 12:00부터 오후 5:59까지의 시간에 액세스할 때 `true`가 됩니다.

주 `timeofday` 및 `dayofweek` 바인드 규칙의 평가에 사용되는 시간과 날짜는 클라이언트 시간이 아닌 디렉토리 서버 시간입니다.

요일을 기준으로 바인드 규칙을 설정하는 LDIF 구문은 다음과 같습니다.

```
dayofweek = "day1, day2 ..."
```

`dayofweek` 키워드에는 요일을 나타내는 3자의 영문 약어(`sun, mon, tue, wed, thu, fri, sat`)를 지정합니다. 액세스를 부여하려는 요일을 모두 지정합니다. 예를 들면 다음과 같습니다.

```
dayofweek = "Mon, Tue, Wed, Thu, Fri";
```

열거된 요일 중 하나에 디렉토리를 액세스하면 바인드 규칙은 `true`가 됩니다.

인증 방법에 따른 액세스 정의

클라이언트가 특정 인증 방법을 사용하여 디렉토리에 바인드하도록 바인드 규칙을 설정할 수 있습니다. 사용할 수 있는 인증 방법은 다음과 같습니다.

- **없음** - 인증이 필요 없습니다. 이것이 기본값입니다. 익명 액세스를 나타냅니다.
- **단순** - 클라이언트가 디렉토리에 바인드하려면 사용자 이름과 비밀번호를 제공해야 합니다.
- **SSL** - 클라이언트가 SSL(Secure Sockets Layer)이나 TLS(Transport Layer Security) 연결을 통해 디렉토리에 바인드해야 합니다.

SSL은 LDAPS 제 2 포트에 연결되고 TLS는 TLS 시작 작업을 통해 연결됩니다. 두 경우 모두 인증서를 제공해야 합니다. SSL 설정에 대한 자세한 내용은 [11장, "인증 및 암호화 관리"](#)를 참조하십시오.

- **SASL** - 클라이언트가 DIGEST-MD5 또는 GSSAPI와 같은 SASL(Simple Authentication and Security Layer) 기법을 사용하여 디렉토리에 바인드해야 합니다.

인증 기반의 바인드 규칙은 액세스 제어 편집기에서 설정할 수 없습니다.

인증 방법에 따라 바인드 규칙을 설정하는 LDIF 구문은 다음과 같습니다.

```
authmethod = "authentication_method"
```

여기서 `authentication_method`는 `none, simple, ssl` 또는 `sasl sasl_mechanism`입니다. 예를 들면 다음과 같습니다.

예

authmethod 키워드의 예는 다음과 같습니다.

- authmethod = "none"; 바인드 규칙을 평가하는 동안 인증을 검사하지 않습니다.
- authmethod = "simple"; 클라이언트가 사용자 이름과 비밀번호를 사용하여 디렉토리에 액세스하면 바인드 규칙은 true가 됩니다.
- authmethod = "ssl"; 클라이언트가 인증서를 사용하여 LDAPS를 통해 디렉토리에 인증하면 바인드 규칙은 true가 됩니다. 클라이언트가 단순 인증(바인드 DN과 비밀번호)을 사용하여 LDAPS를 통해 인증하는 경우에는 true가 아닙니다.
- authmethod = "sasl DIGEST-MD5"; 클라이언트가 SASL DIGEST-MD5 기법을 사용하여 디렉토리에 액세스하면 바인드 규칙은 true가 됩니다. 지원되는 다른 SASL 기법에는 EXTERNAL(모든 플랫폼)과 GSSAPI(Solaris 시스템에서만) 등이 있습니다.

부울 바인드 규칙 사용

부울 연산자인 AND, OR, NOT을 사용하여 세부적인 액세스 규칙을 설정하는 복잡한 바인드 규칙을 작성할 수 있습니다.

Directory Server 콘솔을 사용하여 부울 바인드 규칙을 만들 수 없습니다. LDIF 명령문을 작성해야 합니다.

부울 바인드 규칙을 만들 때는 항상 괄호를 사용하여 평가할 규칙의 순서를 정의해야 합니다.

뒤에 있는 세미콜론(;)은 필수 구분 기호로서 최종 규칙 뒤에 반드시 입력해야 합니다.

예를 들어 bindRuleA를 사용하고, bindRuleB 사용 또는 bindRuleC 및 bindRuleD 사용으로 바인드하려면 다음 구문을 사용하십시오.

```
(bindRuleA and (bindRuleB or (bindRuleC and bindRuleD)));
```

다른 예를 사용하는 경우, 바인드 DN 클라이언트가 example.com 도메인 내에서 액세스될 수 있고 관리자 그룹이나 메일 관리자 및 달력 관리자 그룹의 구성원이라면 다음 바인드 규칙이 true로 평가됩니다.

```
(dns = "*.example.com" and
(groupdn = "ldap:///cn=administrators,dc=example,dc=com" or
(groupdn = "ldap:///cn=mail administrators,dc=example,dc=com" and
groupdn = "ldap:///cn=calendar administrators,dc=example,dc=com")));)
```

명령줄에서 ACI 작성

LDIF 명령문을 사용하여 수동으로 액세스 제어 명령을 작성한 다음 `ldapmodify` 명령을 사용하여 디렉토리 트리에 추가할 수 있습니다. ACI 값은 매우 복잡할 수 있으므로 기존 값을 보고 이를 복사하여 새 값을 작성하는 것이 좋습니다.

ACI 속성 값 보기

ACI는 하나 이상의 `aci` 속성 값으로 항목에 저장됩니다. `aci` 속성은 디렉토리 사용자가 읽고 수정할 수 있는 여러 값을 갖는 작동 가능 속성으로, 이 속성 자체도 ACI로 보호해야 합니다. 대체로 관리 권한이 있는 사용자는 `aci` 속성에 대한 전체 액세스 권한을 가지며 다음 중 한 가지 방법으로 속성 값을 볼 수 있습니다.

일반 편집기에서 다른 값과 같은 방식으로 `aci` 속성 값을 볼 수 있습니다. **Directory Server** 콘솔의 최상위 수준 디렉토리 탭에서 ACI가 있는 항목을 마우스 오른쪽 버튼으로 누르고 일반 편집기로 편집 메뉴 항목을 선택합니다. 하지만 `aci` 값은 대체로 긴 문자열이기 때문에 이 대화 상자에서 보고 편집하기는 어렵습니다.

이 경우에는 디렉토리 트리에서 해당 항목을 마우스 오른쪽 버튼으로 누르고 액세스 권한 설정 메뉴 항목을 선택하여 액세스 제어 편집기를 실행할 수 있습니다. ACI를 선택하고 편집을 누른 다음 수동으로 편집을 눌러 해당 `aci` 값을 표시합니다. ACI 수동 편집기와 시각 편집기를 전환하여 `aci` 값 구문을 구성과 비교할 수 있습니다.

사용 중인 운영 체제에 따라 일반 편집기나 수동 액세스 제어 편집기에서 `aci` 값을 복사하여 LDIF 파일에 붙여넣을 수 있습니다. 관리 권한이 있는 사용자는 아래의 `ldapsearch` 명령을 실행하여 항목의 ACI 속성을 볼 수도 있습니다.

```
ldapsearch -h host -p port -D "cn=Directory Manager" -w password \
  -b entryDN -s base "(objectclass=*)" aci
```

결과는 LDIF 텍스트로 표시되며, 이 텍스트를 새 LDIF ACI 정의에 복사하여 편집할 수 있습니다. ACI 값이 긴 문자열이기 때문에 `ldapsearch` 작업의 출력은 여러 줄에 걸쳐 표시되며, 첫 칸의 공백으로 연속된 줄을 나타냅니다. 이를 고려해서 LDIF 출력을 복사하여 붙여넣습니다.

주 `aci` 값에 따라 부여되거나 거부되는 권한을 확인하려면 [295페이지의 "유효 권한 보기"](#)를 참조하십시오.

콘솔에서 ACI 작성

디렉토리에서 aci 속성이 있는 항목을 표시하도록 Directory Server 콘솔을 구성할 수 있습니다. 이 디스플레이를 전환하려면 보기 > 표시 > ACI 개수 메뉴 항목을 선택하거나 선택 취소합니다. 최상위 수준 디렉토리 탭에 열거된 각 항목 뒤에 해당 aci 속성에 정의된 ACI 수가 표시됩니다. 그런 후에 Directory Server 콘솔을 사용하여 디렉토리에 대한 액세스 제어 명령을 보거나 작성하고 편집 및 삭제할 수 있습니다.

Directory Server 보안 정책에서 일반적으로 사용되는 액세스 제어 규칙 모음과 Directory Server 콘솔을 사용하여 액세스 제어 규칙을 작성하는 단계별 지침은 [276페이지의 "액세스 제어 사용 예"](#)를 참조하십시오.

일부 복잡한 ACI는 시각적 편집 모드의 액세스 제어 편집기에서 구성할 수 없습니다. 특히 액세스 제어 편집기에서 다음과 같은 작업은 수행할 수 없습니다.

- 액세스 거부([253페이지의 "권한 구문"](#) 참조)
- 값 기반의 ACI 작성([248페이지의 "LDAP 필터를 사용한 대상 속성 값 지정"](#) 참조)
- 부모 액세스 정의([257페이지의 "부모 액세스\(parent 키워드\)"](#) 참조)
- 부울 바인드 규칙이 포함된 ACI 작성([269페이지의 "부울 바인드 규칙 사용"](#) 참조)
- 일반적으로 roledn, userattr, authmethod 키워드를 사용하는 ACI 작성

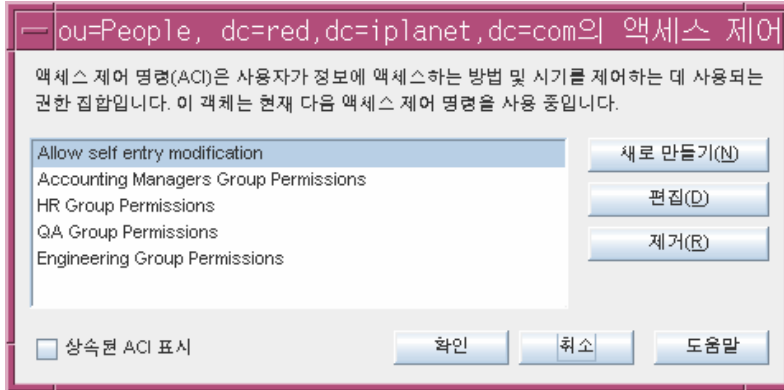
팁 액세스 제어 편집기에서 수동으로 편집 버튼을 누르면 언제든지 그래픽 인터페이스를 통해 변경한 사항을 LDIF 표시로 확인할 수 있습니다.

항목의 ACI 보기

1. Directory Server 콘솔의 최상위 수준 디렉토리 탭에서 디렉토리 트리를 탐색하여 액세스 제어를 설정할 항목을 표시합니다. ACI를 편집하려면 디렉토리 관리자(directory administrator) 또는 디렉토리 관리자(directory manager) 권한이 있어야 합니다.
2. 항목을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 액세스 권한 설정을 선택합니다. 또는 항목을 왼쪽 마우스 버튼으로 눌러 선택한 다음 객체 메뉴에서 액세스 권한 설정을 선택합니다.

아래 그림과 같이 액세스 제어 관리 대화 상자가 표시됩니다. 이 대화 상자에는 선택한 항목에 정의된 모든 ACI에 대한 설정이 표시되며 ACI를 편집하거나 제거하고 새 ACI를 작성할 수 있습니다.

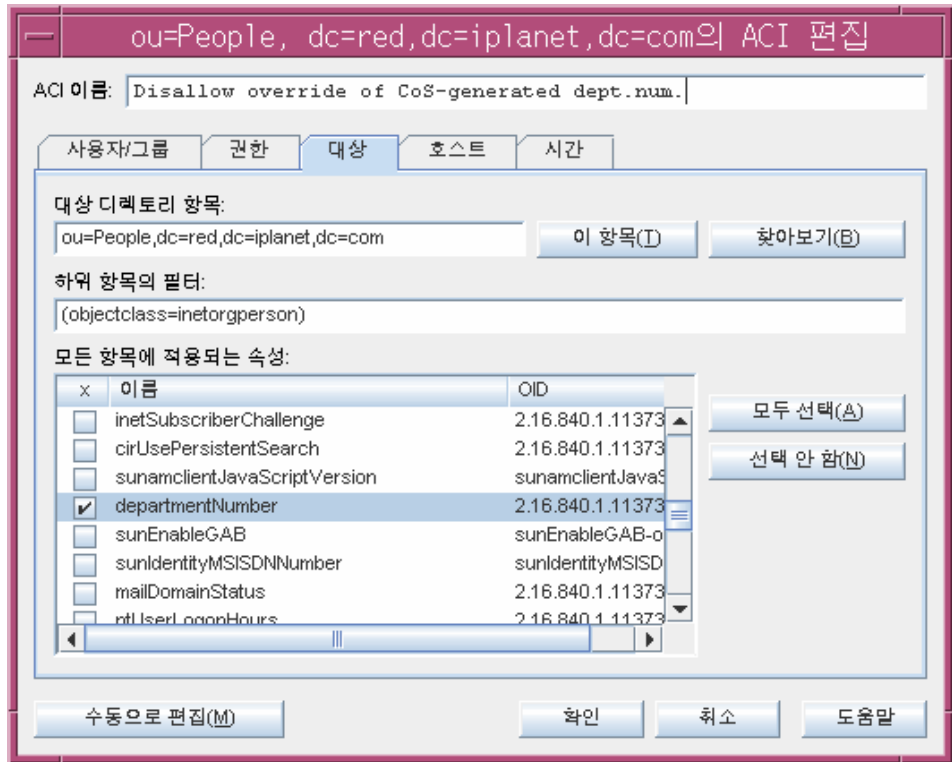
그림 6-2 액세스 제어 관리 대화 상자



상속된 ACI 표시 확인란을 선택하면 선택한 항목의 부모에 의해 정의된 ACI와 선택한 항목에 적용되는 ACI가 모두 표시됩니다. 상속된 ACI는 편집하거나 제거할 수 없으므로 해당 ACI가 정의된 항목에서 관리해야 합니다.

3. 선택한 객체와 전체 하위 트리에 대한 새 액세스 권한을 정의하려면 새로 만들기를 누릅니다. 그림 6-3과 같이 ACI 편집기가 표시됩니다.

그림 6-3 ACI 편집기 대화 상자



이 대화 상자의 맨 위에 있는 ACI 이름은 ACI에 대한 설명으로 액세스 제어 관리 대화 상자에 표시됩니다. ACI를 설명하는 이름을 지정하면 디렉토리에서, 특히 리프 항목에서 상속된 ACI를 볼 때 쉽게 ACI를 관리할 수 있습니다.

액세스 제어 편집기의 탭을 사용하면 액세스 권한이 부여되거나 거부된 사용자, 액세스 중이거나 액세스가 제한된 대상, 허용된 호스트 이름 및 작동 시간과 같은 고급 매개 변수를 지정할 수 있습니다. 액세스 제어 탭의 개별 필드에 대한 자세한 내용은 온라인 도움말을 참조하십시오.

ACI 편집기의 탭은 ACI 값의 내용을 그래픽으로 표시합니다. 텍스트로 ACI 값을 보고 편집하려면 수동으로 편집을 누릅니다. 텍스트 편집기에서는 탭에서 정의할 수 없는 고급 ACI를 정의할 수 있습니다. 하지만 ACI 값을 편집한 후에는 고급 기능을 사용하지 *않더라도* 이상 시각적으로 ACI를 편집할 수 없습니다.

새 ACI 작성

1. 액세스 제어 편집기를 표시합니다.
이 작업에 대해서는 272페이지의 "항목의 ACI 보기"에서 설명합니다.
273페이지의 그림 6-3과 다르게 표시되면 시각적으로 편집 버튼을 누릅니다.
2. ACI 이름 텍스트 상자에 이름을 입력하여 ACI를 지정합니다.
ACI를 고유하게 식별하는 문자열을 이름으로 사용할 수 있습니다. 이름을 입력하지 않으면 서버는 **이름 없는 ACI**를 사용합니다.
3. 사용자/그룹 탭에서 모든 사용자를 강조 표시하거나 추가 버튼을 눌러 디렉토리에서 추가할 사용자를 검색하여 액세스 권한을 부여할 사용자를 선택합니다.
사용자 및 그룹 추가 창에서 다음을 수행합니다.
 - a. 드롭다운 목록에서 검색 영역을 선택하고 검색 필드에 검색 문자열을 입력한 다음 검색 버튼을 누릅니다.
아래 목록에 검색 결과가 표시됩니다.
 - b. 검색 결과 목록에서 원하는 항목을 강조 표시한 다음 추가 버튼을 눌러 액세스 권한이 있는 항목 목록에 추가합니다.
 - c. 확인을 눌러 사용자 및 그룹 추가 창을 닫습니다.
이제 선택한 항목이 ACI 편집기의 사용자/그룹 탭에 표시됩니다.
4. 액세스 제어 편집기에서 권한 탭을 누르고 확인란을 사용하여 부여할 권한을 선택합니다.
5. 대상 탭을 누른 다음 이 항목을 눌러 ACI 대상 노드를 표시합니다.
대상 DN 값을 변경할 수는 있지만 새 DN은 선택한 항목의 직접 또는 간접 자식이어야 합니다.
이 노드의 하위 트리에 있는 일부 항목에만 ACI를 적용하려면 하위 항목의 필터 필드에 필터를 입력해야 합니다.
또한 속성 목록에서 대상 속성을 선택하면 특정 속성에만 ACI가 적용되도록 범위를 제한할 수 있습니다.
6. 호스트 탭을 누른 다음 추가 버튼을 눌러 호스트 필터 추가 대화 상자를 표시합니다.
호스트 이름이나 IP 주소를 지정할 수 있습니다. IP 주소를 지정할 경우 와일드카드 문자(*)를 사용할 수 있습니다.

7. 시간 탭을 눌러 액세스 허용 시간을 나타내는 테이블을 표시합니다.
기본적으로 항상 액세스가 허용됩니다. 테이블을 누른 상태에서 커서를 끌어서 액세스 시간을 변경할 수 있습니다. 시간 블록을 개별적으로 선택할 수는 없습니다.
8. ACI 편집이 끝나면 확인을 누릅니다.
ACI 편집기가 닫히고 ACI 관리자 창에 새 ACI가 표시됩니다.

주 ACI 작성 중에 언제든지 수동으로 편집 버튼을 눌러 입력 내용에 해당하는 LDIF 명령문을 표시할 수 있습니다. 이 명령문을 수정할 수는 있지만 변경 사항이 그래픽 인터페이스에 항상 표시되는 것은 아닙니다.

ACI 편집

ACI를 편집하려면 다음을 수행합니다.

1. 디렉토리 탭의 하위 트리에서 최상위 항목을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 액세스 권한 설정을 선택합니다.
액세스 제어 관리자 창이 표시됩니다. 이 창에는 해당 항목에 속하는 ACI 목록이 포함되어 있습니다.
2. 액세스 제어 관리자 창에서 편집할 ACI를 강조 표시한 다음 편집을 누릅니다.
액세스 제어 편집기가 표시됩니다. 이 대화 상자를 사용하여 편집할 수 있는 정보에 대한 자세한 내용은 온라인 도움말을 참조하십시오.
3. 액세스 제어 편집기의 여러 탭에서 원하는 항목을 변경합니다.
4. ACI 편집이 끝나면 확인을 누릅니다.
ACI 편집기가 닫히고 ACI 관리자 창에 수정된 ACI가 표시됩니다.

ACI 삭제

ACI를 삭제하려면 다음을 수행합니다.

1. 디렉토리 탭의 하위 트리에서 최상위 항목을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 액세스 권한 설정을 선택합니다.

액세스 제어 관리자 창이 표시됩니다. 이 창에는 해당 항목에 속하는 ACI 목록이 포함되어 있습니다.

2. 액세스 제어 관리자 창에서 삭제할 ACI를 선택합니다.
3. 제거를 누릅니다.

해당 ACI가 액세스 제어 관리자에 표시되지 않습니다.

액세스 제어 사용 예

이 절에서는 가상의 ISP 업체인 example.com에서 액세스 제어 정책을 구현하는 예를 제공합니다. 모든 예는 콘솔 및 LDIF 파일을 사용하여 지정된 작업을 수행하는 방법에 대해 설명합니다.

example.com은 웹 호스팅 서비스와 인터넷 액세스를 제공하며 웹 호스팅 서비스의 일부로 클라이언트 기업의 디렉토리를 호스팅합니다. example.com은 실제로 두 개 중소기업(Company333과 Company999)의 디렉토리를 호스팅하고 있으며 일부 관리 작업도 수행합니다. 또한 다수의 개인 가입자에게 인터넷 액세스를 제공합니다.

example.com에서 적용하려는 액세스 제어 규칙은 다음과 같습니다.

- example.com 직원들에게 전체 example.com 트리를 읽고, 검색 및 비교할 수 있는 익명 액세스 권한 부여(277페이지의 "익명 액세스 부여" 참조)
- example.com 직원들에게 homeTelephoneNumber, homeAddress 등의 개인 정보에 대한 쓰기 권한 부여(279페이지의 "개인 항목에 대한 쓰기 액세스 권한 부여" 참조)
- example.com 직원들에게 중요한 특정 역할을 제외한 모든 역할을 자신의 항목에 추가할 수 있는 권한 부여(282페이지의 "중요 역할에 대한 액세스 제한" 참조)
- example.com Human Resources 그룹에 People 분기의 항목에 대한 모든 권한 부여(284페이지의 "그룹에 접미어에 대한 전체 액세스 권한 부여" 참조)
- example.com 직원들에게 디렉토리의 Social Committee 분기에 그룹 항목을 작성할 수 있는 권한과 소유한 그룹 항목을 삭제할 수 있는 권한 부여(285페이지의 "그룹 항목 추가 및 삭제 권한 부여" 참조)

- example.com 직원들에게 디렉토리의 Social Committee 분기의 그룹 항목에 자신을 추가할 수 있는 권한 부여(292페이지의 "사용자가 그룹에 자신을 추가 또는 제거할 수 있도록 허용" 참조)
- SSL 인증, 시간 및 날짜 제한, 지정된 위치와 같은 특정 조건을 사용하여 Company333 및 Company999의 디렉토리 관리자(역할)에게 디렉토리 트리의 해당 분기에 대한 액세스 권한 부여(288페이지의 "그룹이나 역할에 조건부 액세스 권한 부여" 참조)
- 개인 가입자에게 자신의 항목에 대한 액세스 권한 부여(279페이지의 "개인 항목에 대한 쓰기 액세스 권한 부여" 참조)
- 개인 가입자가 자신의 항목에 있는 결제 정보에 액세스하지 못하도록 거부(290페이지의 "액세스 거부" 참조)
- 특별히 목록에 표시하지 않도록 요청한 가입자를 제외하고 개인 가입자 하위 트리에 대한 익명 액세스 권한 부여. 디렉토리에서 이 부분은 방화벽 외부의 슬레이브 서버로 하루에 한 번 업데이트될 수 있습니다. 277페이지의 "익명 액세스 부여" 및 292페이지의 "필터링을 사용한 대상 설정"을 참조하십시오.

익명 액세스 부여

대부분의 디렉토리는 읽기, 검색 또는 비교를 위해 하나 이상의 접미어에 익명으로 액세스할 수 있도록 허용합니다. 예를 들어, 전화 번호부 등 직원들이 검색할 수 있는 인사 디렉토리를 실행하는 경우 이러한 권한을 설정할 수 있습니다. example.com 내부의 경우가 이에 해당하며 [ACI "Anonymous example.com"](#) 예에서 자세히 설명합니다.

또한 example.com은 ISP로서 누구든지 이용할 수 있는 공공 전화 번호부를 작성하여 모든 가입자의 연락처 정보를 제공하려고 합니다. 여기에 대해서는 [ACI "Anonymous World"](#) 예에서 설명합니다.

ACI "Anonymous example.com"

example.com 직원들에게 전체 example.com 트리에 대한 읽기, 검색 및 비교 권한을 부여하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (targetattr !="userPassword")(version 3.0; aci "Anonymous
example"; allow (read, search, compare)
userdn= "ldap:///anyone" and dns="*.example.com");
```

이 예에서는 dc=example,dc=com 항목에 aci를 추가한다고 가정합니다. userPassword 속성은 ACI 범위에서 제외됩니다.

콘솔에서 다음을 수행하여 이 권한을 설정할 수 있습니다.

1. 디렉토리 탭의 왼쪽 탐색 트리에서 **example.com** 노드를 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 액세스 권한 설정을 선택하여 액세스 제어 관리자를 표시합니다.
2. 새로 만들기를 눌러 액세스 제어 편집기를 표시합니다.
3. 사용자/그룹 탭의 ACI 이름 필드에 "**Anonymous example.com**"을 입력합니다. 액세스 권한이 부여된 사용자 목록에 모든 사용자가 표시되는지 확인합니다.
4. 권한 탭에서 읽기, 비교 및 검색 권한 확인란을 선택합니다. 다른 확인란은 모두 선택 취소해야 합니다.
5. 대상 탭에서 이 항목을 눌러 대상 디렉토리 항목 필드에 **dc=example,dc=com** 접미어를 표시합니다. 속성 테이블에서 **userPassword** 속성을 찾아 해당 확인란을 선택 취소합니다.

다른 확인란은 모두 선택해야 합니다. 간편하게 이 작업을 수행하려면 이름 머리글을 눌러 속성 목록을 알파벳순으로 정렬합니다.

6. 호스트 탭에서 추가를 누른 다음 DNS 호스트 필터 필드에 ***.example.com**을 입력합니다. 확인을 눌러 대화 상자를 닫습니다.
7. 액세스 제어 편집기 창에서 확인을 누릅니다.

액세스 제어 관리자 창의 목록에 새 ACI가 추가됩니다.

ACI "Anonymous World"

모든 사람에게 개인 가입자 하위 트리에 대한 읽기 및 검색 권한을 부여하는 동시에 목록에 없는 가입자 정보에 대한 액세스를 거부하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (targetfilter= "(!(unlistedSubscriber=yes))")
(targetattr="homePostalAddress || homePhone || mail")
(version 3.0; acl "Anonymous World"; allow (read, search)
userdn="ldap:///anyone");)
```

이 예에서는 **ou=subscribers, dc=example, dc=com** 항목에 ACI를 추가한다고 가정합니다. 또한 모든 가입자 항목에 예나 아니요로 설정된 **unlistedSubscriber** 속성이 있다고 가정합니다. 대상 정의는 이 속성 값을 기준으로 목록에 없는 가입자를 필터링합니다. 필터 정의에 대한 자세한 내용은 [292페이지의 "필터링을 사용한 대상 설정"](#)을 참조하십시오.

콘솔에서 다음을 수행하여 이 권한을 설정할 수 있습니다.

1. 디렉토리 탭의 왼쪽 탐색 트리에서 `example.com` 노드 아래의 **Subscribers** 항목을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 액세스 권한 설정을 선택하여 액세스 제어 관리자를 표시합니다.
2. 새로 만들기를 눌러 액세스 제어 편집기를 표시합니다.
3. 사용자/그룹 탭의 ACI 이름 필드에 "**Anonymous World**"를 입력합니다. 액세스 권한이 부여된 사용자 목록에 모든 사용자가 표시되어 있는지 확인합니다.
4. 권한 탭에서 읽기 및 검색 권한 확인란을 선택합니다. 다른 확인란은 모두 선택 취소해야 합니다.
5. 대상 탭에서 이 항목을 눌러 대상 디렉토리 항목 필드에 `dc=subscribers, dc=example, dc=com` 접미어를 표시합니다.
 - a. 하위 항목 필터 필드에 아래 필터를 입력합니다.
`!(unlistedSubscriber=yes)`
 - b. 속성 테이블에서 `homePhone`, `homePostalAddress` 및 `mail` 속성에 해당하는 확인란을 선택합니다.

 다른 확인란은 모두 선택 취소해야 합니다. 간편하게 이 작업을 수행하려면 선택 안 함 버튼을 눌러 테이블에 있는 모든 속성의 확인란을 선택 취소한 다음, 이름 머리글을 눌러 속성을 알파벳순으로 정렬하고 해당 속성을 선택합니다.
6. 확인을 누릅니다.

 액세스 제어 관리자 창의 목록에 새 ACI가 추가됩니다.

개인 항목에 대한 쓰기 액세스 권한 부여

대부분의 디렉토리 관리자는 내부 사용자가 자신의 항목에 있는 일부 속성만 변경할 수 있도록 설정합니다. `example.com`의 디렉토리 관리자도 사용자가 자신의 비밀번호, 집 전화 번호 및 집 주소만 변경할 수 있도록 설정하려고 합니다. 여기에 대해서는 [ACI "Write example.com"](#) 예에서 설명합니다.

또한 `example.com`의 정책은 가입자가 SSL을 통해 디렉토리에 연결하면 `example.com` 트리에서 자신의 개인 정보를 업데이트할 수 있도록 허용합니다. 여기에 대해서는 [ACI "Write Subscribers"](#) 예에서 설명합니다.

ACI "Write example.com"

주 이 권한을 설정하면 사용자에게 속성 값을 삭제할 수 있는 권한도 부여하게 됩니다.

example.com 직원들에게 비밀번호, 집 전화 번호 및 집 주소를 업데이트할 수 있는 권한을 부여하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (targetattr="userPassword || homePhone ||
homePostalAddress")(version 3.0; aci "Write example.com";
allow (write) userdn="ldap:///self" and dns="*.example.com");
```

이 예에서는 ou=People, dc=example, dc=com 항목에 ACI를 추가한다고 가정합니다. 콘솔에서 다음을 수행하여 이 권한을 설정할 수 있습니다.

1. 디렉토리 탭의 왼쪽 탐색 트리에서 ou=People, dc=example, dc=com 항목을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 액세스 권한 설정을 선택하여 액세스 제어 관리자를 표시합니다.
2. 새로 만들기를 눌러 액세스 제어 편집기를 표시합니다.
3. 사용자/그룹 탭의 ACI 이름 필드에 "Write example.com"을 입력합니다. 액세스 권한이 부여된 사용자 목록에서 다음을 수행하십시오.
 - a. 모든 사용자를 선택하여 제거한 다음 추가를 누릅니다.
사용자 및 그룹 추가 대화 상자가 표시됩니다.
 - b. 검색 영역을 특수 권한으로 설정하고 검색 결과 목록에서 자신을 선택합니다.
 - c. 추가 버튼을 눌러 액세스 권한이 부여된 사용자 목록에 자신을 추가합니다.
 - d. 확인을 눌러 사용자 및 그룹 추가 대화를 닫습니다.
4. 권한 탭에서 쓰기 권한 확인란을 선택합니다. 다른 확인란은 모두 선택 취소해야 합니다.

5. 대상 탭에서 이 항목을 눌러 대상 디렉토리 항목 필드에 `ou=People,dc=example,dc=com`을 입력합니다. 속성 테이블에서 `homePhone`, `homePostalAddress` 및 `userPassword` 속성에 해당하는 확인란을 선택합니다.
 다른 확인란은 모두 선택 취소해야 합니다. 간편하게 이 작업을 수행하려면 선택 안 함 버튼을 눌러 테이블에 있는 모든 속성의 확인란을 선택 취소한 다음, 이름 머리글을 눌러 속성을 알파벳순으로 정렬하고 해당 속성을 선택합니다.
6. 호스트 탭에서 추가를 눌러 호스트 필터 추가 대화 상자를 표시합니다. DNS 호스트 필터 필드에 `*.example.com`을 입력합니다. 확인을 눌러 대화 상자를 닫습니다.
7. 액세스 제어 편집기 창에서 확인을 누릅니다.
 액세스 제어 관리자 창의 목록에 새 ACI가 추가됩니다.

ACI "Write Subscribers"

주	이 권한을 설정하면 사용자에게 속성 값을 삭제할 수 있는 권한도 부여하게 됩니다.
----------	---

`example.com` 가입자에게 비밀번호, 집 전화 번호를 업데이트할 수 있는 권한을 부여하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (targetattr="userPassword || homePhone")
(version 3.0; acl "Write Subscribers"; allow (write)
userdn= "ldap://self" and authmethod="ssl");
```

이 예에서는 `ou=subscribers,dc=example,dc=com` 항목에 `aci`를 추가한다고 가정합니다.

`example.com` 가입자에게는 집 주소에 대한 쓰기 액세스 권한이 없습니다. 이 속성은 `example.com`의 필수 결제 정보로 업무상 중요하기 때문에 가입자가 삭제하지 못하도록 액세스가 거부됩니다.

콘솔에서 다음을 수행하여 이 권한을 설정할 수 있습니다.

1. 디렉토리 탭의 왼쪽 탐색 트리에서 `example.com` 노드 아래의 **Subscribers** 항목을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 액세스 권한 설정을 선택하여 액세스 제어 관리자를 표시합니다.
2. 새로 만들기를 눌러 액세스 제어 편집기를 표시합니다.
3. 사용자/그룹 탭의 ACI 이름 필드에 "Write Subscribers"를 입력합니다. 액세스 권한이 부여된 사용자 목록에서 다음을 수행합니다.

- a. 모든 사용자를 선택하여 제거한 다음 추가를 누릅니다.
사용자 및 그룹 추가 대화 상자가 표시됩니다.
 - b. 검색 영역을 특수 권한으로 설정하고 검색 결과 목록에서 자신을 선택합니다.
 - c. 추가 버튼을 눌러 액세스 권한이 부여된 사용자 목록에 자신을 추가합니다.
 - d. 확인을 눌러 사용자 및 그룹 추가 대화 상자를 닫습니다.
4. 권한 탭에서 쓰기 확인란을 선택합니다. 다른 확인란은 모두 선택 취소해야 합니다.
 5. 대상 탭에서 이 항목을 눌러 대상 디렉토리 항목 필드에 dc=subscribers, dc=example, dc=com 접미어를 표시합니다.
 - a. 하위 항목 필터 필드에 아래 필터를 입력합니다.
(!(unlistedSubscriber=yes))
 - b. 속성 테이블에서 homePhone, homePostalAddress 및 mail 속성에 해당하는 확인란을 선택합니다.

다른 확인란은 모두 선택 취소해야 합니다. 간편하게 이 작업을 수행하려면 선택 안 함 버튼을 눌러 테이블에 있는 모든 속성의 확인란을 선택 취소한 다음, 이름 머리글을 눌러 속성을 알파벳순으로 정렬하고 해당 속성을 선택합니다.
 6. SSL을 사용하여 사용자를 인증하려면 수동으로 편집 버튼을 눌러 수동 편집으로 전환하고 다음과 같이 LDIF 명령문에 authmethod=ssl을 추가합니다.


```
(targetattr="homePostalAddress || homePhone || mail")
(version 3.0; acl "Write Subscribers"; allow (write)
(userdn= "ldap:///self") and authmethod="ssl");
```

가독성을 위해 분리되었을 뿐 사실은 연속된 줄입니다.
 7. 확인을 누릅니다.
액세스 제어 관리자 창의 목록에 새 ACI가 추가됩니다.

중요 역할에 대한 액세스 제한

디렉토리에서 역할 정의를 사용하여 업무상 중요한 기능이나 네트워크 및 디렉토리 관리 또는 다른 용도를 식별할 수 있습니다.

예를 들어, 특정 시간과 요일에 전세계 기업 사이트에서 사용할 수 있는 시스템 관리자의 부분 집합을 식별하여 superAdmin 역할을 작성할 수 있습니다. 또는 응급 조치 교육을 이수한 특정 사이트의 직원들이 모두 포함된 First Aid 역할을 작성할 수 있습니다. 역할 정의를 작성하는 방법은 [210페이지의 "역할 할당"](#)을 참조하십시오.

중요한 기업 또는 비즈니스 기능에 대한 사용자 권한을 부여하는 역할이 있을 경우 이 역할에 대한 액세스를 제한해야 합니다. 예를 들어, example.com의 직원들은 superAdmin 역할을 제외한 모든 역할을 자신의 항목에 추가할 수 있습니다. 여기에 대해서는 [ACI "Roles"](#) 에에서 설명합니다.

ACI "Roles"

example.com 직원들에게 superAdmin 역할을 제외한 모든 역할을 자신의 항목에 추가할 수 있는 권한을 부여하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (targetattr="*") (targetattrfilters="add=nsRoleDN:
(nsRoleDN !="cn=superAdmin, dc=example, dc=com")")
(version 3.0; acl "Roles"; allow (write)
userdn= "ldap:///self" and dns="*.example.com");
```

이 예에서는 ou=People,dc=example, dc=com 항목에 ACI를 추가한다고 가정합니다. 콘솔에서 다음을 수행하여 이 권한을 설정할 수 있습니다.

1. 디렉토리 탭의 왼쪽 탐색 트리에서 example.com 노드를 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 액세스 권한 설정을 선택하여 액세스 제어 관리자를 표시합니다.
2. 새로 만들기를 눌러 액세스 제어 편집기를 표시합니다.
3. 사용자/그룹 탭의 ACI 이름 필드에 "Roles"를 입력합니다. 액세스 권한이 부여된 사용자 목록에서 다음을 수행합니다.
 - a. 모든 사용자를 선택하여 제거한 다음 추가를 누릅니다.
사용자 및 그룹 추가 대화 상자가 표시됩니다.
 - b. 사용자 및 그룹 추가 대화 상자에서 검색 영역을 특수 권한으로 설정하고 검색 결과 목록에서 자신을 선택합니다.
 - c. 추가 버튼을 눌러 액세스 권한이 부여된 사용자 목록에 자신을 추가합니다.
 - d. 확인을 눌러 사용자 및 그룹 추가 대화 상자를 닫습니다.
4. 권한 탭에서 쓰기 확인란을 선택합니다. 다른 확인란은 모두 선택 취소해야 합니다.

5. 호스트 탭에서 추가를 눌러 호스트 필터 추가 대화 상자를 표시합니다. DNS 호스트 필터 필드에 *.example.com을 입력합니다. 확인을 눌러 대화 상자를 닫습니다.
6. 역할에 대한 값 기반의 필터를 작성하려면 수동으로 편집 버튼을 눌러 수동 편집으로 전환합니다. LDIF 명령문의 시작 부분에 아래 명령을 추가합니다.

```
(targetattrfilters="add=nsRoleDN:
  (nsRoleDN != "cn=superAdmin, dc=example,dc=com") ")
```

LDIF 명령문이 다음과 같이 표시됩니다.

```
(targetattr="*") (targetattrfilters="add=nsRoleDN:
  (nsRoleDN != "cn=superAdmin, dc=example,dc=com") ")
(target = "ldap:///dc=example,dc=com")
(version 3.0; acl "Roles"; allow (write)
  (userdn = "ldap:///self") and (dns="*.example.com");)
```

7. 확인을 누릅니다.

액세스 제어 관리자 창의 목록에 새 ACI가 추가됩니다.

그룹에 접미어에 대한 전체 액세스 권한 부여

대부분의 디렉토리에는 기업의 특정 기능을 식별하는 그룹이 있습니다. 이러한 그룹에 디렉토리의 모두 또는 일부에 대한 전체 액세스 권한을 부여할 수 있습니다. 그룹에 액세스 권한을 적용하면 각 구성원에 대해 개별적으로 액세스 권한을 설정할 필요가 없습니다. 사용자를 그룹에 추가하기만 하면 액세스 권한을 부여할 수 있습니다.

예를 들어, 일반 설치 프로세스를 사용하여 Directory Server를 설치하면 디렉토리에 대한 전체 액세스 권한을 가진 관리자 그룹이 기본적으로 작성됩니다.

example.com의 경우 Human Resources 그룹은 디렉토리의 ou=People 분기에 대한 전체 액세스 권한을 갖고 있으므로 직원 디렉토리를 업데이트할 수 있습니다. 여기에 대해서는 [ACI "HR"](#) 예에서 설명합니다.

ACI "HR"

디렉토리의 employee 분기에 대한 모든 권한을 HR 그룹에 부여하려면 LDIF에 아래 문을 사용합니다.

```
aci: (targetattr="*") (version 3.0; acl "HR"; allow (all)
  groupdn= "ldap:///cn=HRgroup,ou=People,dc=example,dc=com");)
```

이 예에서는 다음 항목에 ACI를 추가한다고 가정합니다.

```
ou=People,dc=example,dc=com
```

콘솔에서 다음을 수행하여 이 권한을 설정할 수 있습니다.

1. 디렉토리 탭의 왼쪽 탐색 트리에서 **example.com** 노드 아래의 **example.com-people** 항목을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 액세스 권한 설정을 선택하여 액세스 제어 관리자를 표시합니다.
2. 새로 만들기를 눌러 액세스 제어 편집기를 표시합니다.
3. 사용자/그룹 탭의 ACI 이름 필드에 "HR"을 입력합니다. 액세스 권한이 부여된 사용자 목록에서 다음을 수행합니다.
 - a. 모든 사용자를 선택하여 제거한 다음 추가를 누릅니다.
사용자 및 그룹 추가 대화 상자가 표시됩니다.
 - b. 검색 영역을 사용자 및 그룹으로 설정하고 검색 대상 필드에 "Hrgroup"을 입력합니다.

이 예에서는 HR 그룹이나 역할이 작성되어 있다고 가정합니다. 그룹 및 역할에 대한 자세한 내용은 5장, "아이디 및 역할 관리"를 참조하십시오.
 - c. 추가 버튼을 눌러 액세스 권한이 부여된 사용자 목록에 HR 그룹을 표시합니다.
 - d. 확인을 눌러 사용자 및 그룹 추가 대화 상자를 닫습니다.
4. 권한 탭에서 모두 선택 버튼을 누릅니다.
프록시 권한을 제외한 모든 확인란이 선택됩니다.
5. 확인을 누릅니다.
액세스 제어 관리자 창의 목록에 새 ACI가 추가됩니다.

그룹 항목 추가 및 삭제 권한 부여

일부 기업은 직원들이 효율성을 높이고 기업의 활력소 역할을 하는 항목을 트리에 작성할 수 있도록 허용합니다.

예를 들어 **example.com**에는 테니스, 수영, 스키, 롤 플레이 등 여러 클럽으로 구성된 활동적인 사교 모임이 있으며 **example.com**의 직원이라면 누구든지 새 클럽을 나타내는 그룹 항목을 작성할 수 있습니다. 여기에 대해서는 ACI "[Create Group](#)" 예에서 설명합니다. **example.com**의 모든 직원은 이러한 그룹 중 하나의 구성원이 될 수 있습니다. 여기에 대해서는 292페이지의 "[사용자가 그룹에 자신을 추가 또는 제거할 수 있도록 허용](#)"의 ACI "[Group Members](#)"에서 설명합니다. 하지만 그룹 소유자만 그룹 항목을 수정하거나 삭제할 수 있습니다. 여기에 대해서는 ACI "[Delete Group](#)" 예에서 설명합니다.

ACI "Create Group"

example.com 직원들에게 ou=Social Committee 분기에 그룹 항목을 작성할 수 있는 권한을 부여하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (target="ldap:///ou=social committee,dc=example,dc=com")
      (targetattr="*") (targetattrfilters="add=objectClass:
      |(objectClass=groupOfNames)(objectClass=top))")
      (version 3.0; acl "Create Group"; allow (read,search,add)
      userdn= "ldap:///uid=*,ou=People,dc=example,dc=com")
      and dns="*.example.com");
```

-
- 주**
- 이 ACI는 쓰기 권한을 부여하지 않기 때문에 항목 작성자가 항목을 수정할 수 없습니다.
 - 서버가 이면에서 "top" 값을 추가하므로 targetattrfilters 키워드에 objectclass=top을 지정해야 합니다.
-

이 예에서는 ou=social committee, dc=example,dc=com 항목에 ACI를 추가한다고 가정합니다.

콘솔에서 다음을 수행하여 이 권한을 설정할 수 있습니다.

1. 디렉토리 탭의 왼쪽 탐색 트리에서 example.com 노드 아래의 Social Committee 항목을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 액세스 권한 설정을 선택하여 액세스 제어 관리자를 표시합니다.
2. 새로 만들기를 눌러 액세스 제어 편집기를 표시합니다.
3. 사용자/그룹 탭의 ACI 이름 필드에 "Create Group"을 입력합니다. 액세스 권한이 부여된 사용자 목록에서 다음을 수행합니다.
 - a. 모든 사용자를 선택하여 제거한 다음 추가를 누릅니다.
사용자 및 그룹 추가 대화 상자가 표시됩니다.
 - b. 검색 영역을 특수 권한으로 설정하고 검색 결과 목록에서 모든 인증된 사용자를 선택합니다.
 - c. 추가 버튼을 눌러 액세스 권한이 부여된 사용자 목록에 모든 인증된 사용자를 추가합니다.
 - d. 확인을 눌러 사용자 및 그룹 추가 대화 상자를 닫습니다.
4. 권한 탭에서 읽기, 검색 및 추가 확인란을 선택합니다. 다른 확인란은 모두 선택 취소해야 합니다.
5. 대상 탭에서 이 항목을 눌러 대상 디렉토리 항목 필드에 ou=social committee, dc=example,dc=com 접미어를 표시합니다.

6. 호스트 탭에서 추가를 눌러 호스트 필터 추가 대화 상자를 표시합니다. DNS 호스트 필터 필드에 *.example.com을 입력합니다. 확인을 눌러 대화 상자를 닫습니다.
7. 직원들이 이 하위 트리에 그룹 항목만 추가할 수 있도록 값 기반의 필터를 작성하려면 수동으로 편집 버튼을 눌러 수동 편집으로 전환합니다. LDIF 명령문의 시작 부분에 아래 명령을 추가합니다.

```
(targetfilters="add=objectClass:(objectClass=groupOfNames)
| (objectClass=top) ")
```

LDIF 명령문이 다음과 같이 표시됩니다.

```
(targetattr = "*")
(targetfilters="add=objectClass:(objectClass=groupOfNames)
| (objectClass=top) ") (target="ldap:///ou=social
committee,dc=example,dc=com) (version 3.0; acl "Create Group";
allow (read,search,add) (userdn= "ldap:///all") and
(dns="*.example.com")); )
```

8. 확인을 누릅니다.

액세스 제어 관리자 창의 목록에 새 ACI가 추가됩니다.

ACI "Delete Group"

example.com 직원들에게 ou=Social Committee 분기에서 그룹 항목을 수정하거나 삭제할 수 있는 권한을 부여하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (target="ou=social committee,dc=example,dc=com)
(targetattr "*")
(targetfilters="del=objectClass:(objectClass=groupOfNames) ")
(version 3.0; acl "Delete Group"; allow (write,delete)
userattr="owner#GROUPDN";)
```

이 예에서는 ou=social committee, dc=example,dc=com 항목에 aci를 추가한다고 가정합니다.

콘솔을 사용할 경우 수동 편집 모드로 대상 필터를 작성하고 그룹 소유권을 확인해야 하기 때문에 이 ACI를 작성하는 데는 효과적인 방법이 아닙니다.

그룹이나 역할에 조건부 액세스 권한 부여

일반적으로 그룹이나 역할에 디렉토리에 대한 액세스 권한을 부여하는 경우 권한이 있는 사용자를 사칭하는 침입자들로부터 해당 권한을 보호해야 합니다. 따라서 그룹이나 역할에 중요한 액세스 권한을 부여하는 액세스 제어 규칙에는 많은 조건이 따릅니다.

예를 들어, `example.com`은 자사가 호스팅 서비스를 제공하는 두 회사인 `Company333`과 `Company999`에 대해 각각 디렉토리 관리자 역할을 작성했습니다. `example.com`은 두 회사가 자사 데이터를 관리하고 액세스 제어 규칙을 구현할 수 있는 동시에 침입자로부터 데이터를 보호할 수 있기를 원합니다. 이런 이유로 `Company333`과 `Company999`는 다음과 같은 조건에 부합될 경우 디렉토리 트리의 해당 분기에 대한 전체 권한을 갖습니다.

- 인증서를 사용하여 SSL을 통해 인증된 연결
- 월요일부터 목요일까지 오전 8시에서 오후 6시 사이에 요청한 액세스
- 각 회사에 지정된 IP 주소로부터 요청한 액세스

이러한 조건에 대해서는 각 회사별 ACI(ACI "`Company333`" 및 ACI "`Company999`")에서 설명합니다. 두 ACI의 내용이 같기 때문에 아래 예에서는 "`Company333`" ACI에 대해서만 설명합니다.

ACI "`Company333`"

위에 명시된 조건을 전제로 `Company333`에 디렉토리의 분기에 대한 전체 액세스 권한을 부여하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (target="ou=Company333,ou=corporate-clients,dc=example,dc=com")
(targetattr = "*" ) (version 3.0; acl "Company333"; allow (all)
(roledn="ldap:///cn=DirectoryAdmin,ou=Company333,
ou=corporate-clients,dc=example,dc=com") and (authmethod="ssl")
and (dayofweek="Mon,Tues,Wed,Thu") and (timeofday >= "0800" and
timeofday <= "1800") and (ip="255.255.123.234"); )
```

이 예에서는 `ou=Company333, ou=corporate-clients,dc=example,dc=com` 항목에 ACI를 추가한다고 가정합니다.

콘솔에서 다음을 수행하여 이 권한을 설정할 수 있습니다.

1. 디렉토리 탭의 왼쪽 탐색 트리에서 `example.com` 노드 아래의 `Company333` 항목을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 액세스 권한 설정을 선택하여 액세스 제어 관리자를 표시합니다.
2. 새로 만들기를 눌러 액세스 제어 편집기를 표시합니다.
3. 사용자/그룹 탭의 ACI 이름 필드에 "`Company333`"을 입력합니다. 액세스 권한이 부여된 사용자 목록에서 다음을 수행합니다.

- a. 모든 사용자를 선택하여 제거한 다음 추가를 누릅니다.
사용자 및 그룹 추가 대화 상자가 표시됩니다.
 - b. 검색 영역을 사용자 및 그룹으로 설정하고 검색 대상 필드에 "DirectoryAdmin"을 입력합니다.
이 예에서는 cn이 DirectoryAdmin인 관리자 역할이 작성되어 있다고 가정합니다.
 - c. 추가 버튼을 눌러 액세스 권한이 부여된 사용자 목록에 관리자 역할을 표시합니다.
 - d. 확인을 눌러 사용자 및 그룹 추가 대화 상자를 닫습니다.
4. 권한 탭에서 모두 선택 버튼을 누릅니다.
 5. 대상 탭에서 이 항목을 눌러 대상 디렉토리 항목 필드에 ou=Company333, ou=corporate-clients, dc=example, dc=com 접미어를 표시합니다.
 6. 호스트 탭에서 추가를 눌러 호스트 필터 추가 대화 상자를 표시합니다. IP 주소 호스트 필터 필드에 255.255.123.234를 입력합니다. 확인을 눌러 대화 상자를 닫습니다.

IP 주소는 Company333 관리자가 example.com 디렉토리에 연결할 때 사용하는 호스트 시스템의 유효한 IP 주소여야 합니다.

7. 시간 탭에서 월요일에서 목요일까지, 오전 8시에서 오후 6시까지에 해당하는 시간 블록을 선택합니다.
선택한 시간 블록을 지정하는 메시지가 테이블 아래에 나타납니다.
8. Company333 관리자로부터의 SSL 인증을 실행하려면 수동으로 편집 버튼을 눌러 수동 편집으로 전환합니다. LDIF 명령문의 끝 부분에 아래 명령을 추가합니다.

and(authmethod="ssl")

LDIF 명령문이 다음과 같이 표시됩니다.

```
aci: (targetattr = "*") (target="ou=Company333,
ou=corporate-clients,dc=example,dc=com") (version 3.0; acl
"Company333"; allow (all) (roledn="ldap:///cn=DirectoryAdmin,
ou=Company333,ou=corporate-clients, dc=example,dc=com") and
(dayofweek="Mon,Tues,Wed,Thu") and (timeofday >= "0800" and
timeofday <= "1800") and (ip="255.255.123.234") and
(authmethod="ssl"); )
```

9. 확인을 누릅니다.

액세스 제어 관리자 창의 목록에 새 ACI가 추가됩니다.

액세스 거부

디렉토리에 업무상 중요한 정보가 있으면 디렉토리에 대한 액세스를 구체적으로 거부할 수 있습니다.

예를 들어, **example.com**은 모든 가입자가 자신의 항목에서 연결 시간이나 계좌 잔고와 같은 결제 정보를 읽을 수 있도록 허용하지만 이 정보에 대한 쓰기 액세스는 명시적으로 거부합니다. 여기에 대해서는 ACI "**Billing Info Read**" 및 ACI "**Billing Info Deny**"에서 각각 설명합니다.

ACI "**Billing Info Read**"

가입자에게 자신의 항목에서 결제 정보를 읽을 수 있는 권한을 부여하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (targetattr="connectionTime || accountBalance")
(version 3.0; acl "Billing Info Read"; allow (search,read)
userdn="ldap:///self");
```

이 예에서는 스키마에 해당 속성이 작성되어 있으며 ou=subscribers,dc=example, dc=com 항목에 ACI를 추가한다고 가정합니다.

콘솔에서 다음을 수행하여 이 권한을 설정할 수 있습니다.

1. 디렉토리 탭의 왼쪽 탐색 트리에서 **example.com** 노드 아래의 **Subscribers** 항목을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 액세스 권한 설정을 선택하여 액세스 제어 관리자를 표시합니다.
2. 새로 만들기를 눌러 액세스 제어 편집기를 표시합니다.
3. 사용자/그룹 탭의 ACI 이름 필드에 "**Billing Info Read**"를 입력합니다. 액세스 권한이 부여된 사용자 목록에서 다음을 수행합니다.
 - a. 모든 사용자를 선택하여 제거한 다음 추가를 누릅니다.
사용자 및 그룹 추가 대화 상자가 표시됩니다.
 - b. 사용자 및 그룹 추가 대화 상자에서 검색 영역을 특수 권한으로 설정하고 검색 결과 목록에서 자신을 선택합니다.
 - c. 추가 버튼을 눌러 액세스 권한이 부여된 사용자 목록에 자신을 추가합니다.
 - d. 확인을 눌러 사용자 및 그룹 추가 대화 상자를 닫습니다.

4. 권한 탭에서 검색 및 읽기 권한 확인란을 선택합니다. 다른 확인란은 모두 선택 취소해야 합니다.
5. 대상 탭에서 이 항목을 눌러 대상 디렉토리 항목 필드에 `ou=subscribers, dc=example, dc=com` 접미어를 표시합니다. 속성 테이블에서 `connectionTime` 속성과 `accountBalance` 속성에 해당하는 확인란을 선택합니다.

다른 확인란은 모두 선택 취소해야 합니다. 간편하게 이 작업을 수행하려면 선택 안 함 버튼을 눌러 테이블에 있는 모든 속성의 확인란을 선택 취소한 다음, 이름 머리글을 눌러 속성을 알파벳순으로 정렬하고 해당 속성을 선택합니다.

이 예에서는 `connectionTime` 속성과 `accountBalance` 속성이 스키마에 추가되어 있다고 가정합니다.

6. 확인을 누릅니다.

액세스 제어 관리자 창의 목록에 새 ACI가 추가됩니다.

ACI "Billing Info Deny"

가입자에게 자신의 항목에서 결제 정보를 수정할 수 있는 권한을 부여하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (targetattr="connectionTime || accountBalance")
      (version 3.0; aci "Billing Info Deny";
      deny (write) userdn="ldap:///self");
```

이 예에서는 스키마에 해당 속성이 작성되어 있으며

`ou=subscribers, dc=example, dc=com` 항목에 ACI를 추가한다고 가정합니다.

콘솔에서 다음을 수행하여 이 권한을 설정할 수 있습니다.

1. 디렉토리 탭의 왼쪽 탐색 트리에서 `example.com` 노드 아래의 `Subscribers` 항목을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 액세스 권한 설정을 선택하여 액세스 제어 관리자를 표시합니다.
2. 새로 만들기를 눌러 액세스 제어 편집기를 표시합니다.
3. 사용자/그룹 탭의 ACI 이름 필드에 "Billing Info Deny"를 입력합니다. 액세스 권한이 부여된 사용자 목록에서 다음을 수행합니다.
 - a. 모든 사용자를 선택하여 제거한 다음 추가를 누릅니다.
사용자 및 그룹 추가 대화 상자가 표시됩니다.
 - b. 사용자 및 그룹 추가 대화 상자에서 검색 영역을 특수 권한으로 설정하고 검색 결과 목록에서 자신을 선택합니다.
 - c. 추가 버튼을 눌러 액세스 권한이 부여된 사용자 목록에 자신을 추가합니다.
 - d. 확인을 눌러 사용자 및 그룹 추가 대화 상자를 닫습니다.

4. 권한 탭에서 쓰기 확인란을 선택합니다. 다른 확인란은 모두 선택 취소해야 합니다.
5. 수동으로 편집 버튼을 누르고 표시된 LDIF 명령문에서 **allow**를 **deny**로 변경합니다.
6. 대상 탭에서 이 항목을 눌러 대상 디렉토리 항목 필드에 `ou=subscribers, dc=example,dc=com` 접미어를 표시합니다. 속성 테이블에서 `connectionTime` 속성과 `accountBalance` 속성에 해당하는 확인란을 선택합니다.

다른 확인란은 모두 선택 취소해야 합니다. 간편하게 이 작업을 수행하려면 선택 안 함 버튼을 눌러 테이블에 있는 모든 속성의 확인란을 선택 취소한 다음, 이름 머리글을 눌러 속성을 알파벳순으로 정렬하고 해당 속성을 선택합니다.

이 예에서는 `connectionTime` 속성과 `accountBalance` 속성이 스키마에 추가되어 있다고 가정합니다.

7. 확인을 누릅니다.

액세스 제어 관리자 창의 목록에 새 ACI가 추가됩니다.

필터링을 사용한 대상 설정

디렉토리에 분산된 여러 항목에 대한 액세스를 허용하는 액세스 제어를 설정하려면 필터를 사용하여 대상을 설정할 수 있습니다. 검색 필터는 액세스가 관리되는 객체의 이름을 직접 지정하지 않으므로 특히 디렉토리가 복잡할수록 실수로 잘못된 객체에 대한 액세스 권한을 부여하거나 거부할 수 있습니다. 또한 필터를 사용할 경우 디렉토리 내의 액세스 제어 문제점을 해결하기 어렵다는 단점이 있습니다.

사용자가 그룹에 자신을 추가 또는 제거할 수 있도록 허용

대부분의 디렉토리는 사용자가 그룹에 자신을 추가하거나 제거할 수 있도록 허용하는 ACI를 설정합니다. 예를 들어, 사용자가 우편 목록에 자신을 추가하거나 제거할 수 있도록 허용할 경우 이러한 ACI를 설정할 수 있습니다.

`example.com` 직원들은 `ou=social committee` 하위 트리의 모든 그룹 항목에 자신을 추가할 수 있습니다. 여기에 대해서는 [ACI "Group Members"](#) 예에서 설명합니다.

ACI "Group Members"

example.com 직원들에게 그룹에 자신을 추가하거나 삭제할 수 있는 권한을 부여하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (targetattr="member")(version 3.0; acl "Group Members";
allow (selfwrite)
(userdn= "ldap:///uid=*,ou=People,dc=example,dc=com") );
```

이 예에서는 ou=social committee, dc=example,dc=com 항목에 ACI를 추가한다고 가정합니다.

콘솔에서 다음을 수행하여 이 권한을 설정할 수 있습니다.

1. 디렉토리 탭의 왼쪽 탐색 트리에서 example.com 노드 아래의 People 항목을 마우스 오른쪽 버튼으로 누른 다음 팝업 메뉴에서 액세스 권한 설정을 선택하여 액세스 제어 관리자를 표시합니다.
2. 새로 만들기를 눌러 액세스 제어 편집기를 표시합니다.
3. 사용자/그룹 탭의 ACI 이름 필드에 "Group Members"를 입력합니다. 액세스 권한이 부여된 사용자 목록에서 다음을 수행합니다.
 - a. 모든 사용자를 선택하여 제거한 다음 추가를 누릅니다.
사용자 및 그룹 추가 대화 상자가 표시됩니다.
 - b. 사용자 및 그룹 추가 대화 상자에서 검색 영역을 특수 권한으로 설정하고 검색 결과 목록에서 모든 인증된 사용자를 선택합니다.
 - c. 추가 버튼을 눌러 액세스 권한이 부여된 사용자 목록에 모든 인증된 사용자를 추가합니다.
 - d. 확인을 눌러 사용자 및 그룹 추가 대화 상자를 닫습니다.
4. 권한 탭에서 자체 쓰기 확인란을 선택합니다. 다른 확인란은 모두 선택 취소해야 합니다.
5. 대상 탭의 대상 디렉토리 항목 필드에 dc=example,dc=com 접미어를 입력합니다. 속성 테이블에서 member 속성에 해당하는 확인란을 선택합니다.
다른 확인란은 모두 선택 취소해야 합니다. 간편하게 이 작업을 수행하려면 선택 안 함 버튼을 눌러 테이블에 있는 모든 속성의 확인란을 선택 취소한 다음, 이름 머리글을 눌러 속성을 알파벳순으로 정렬하고 해당 속성을 선택합니다.
6. 확인을 누릅니다.
액세스 제어 관리자 창의 목록에 새 ACI가 추가됩니다.

쉽표가 있는 DN에 대한 권한 정의

쉽표가 있는 DN은 LDIF ACI 명령문에서 특별히 처리해야 합니다. ACI 명령문의 대상 및 바인드 규칙 부분에서 역슬래시(\)를 사용하여 쉽표를 이스케이프해야 합니다. 아래 예에서는 이 구문에 대해 설명합니다.

```
dn: o=example.com Bolivia\, S.A.
objectClass: top
objectClass: organization
aci: (target="ldap:///o=example.com Bolivia\,S.A.")
      (targetattr="*") (version 3.0; acl "aci 2"; allow (all)
      groupdn = "ldap:///cn=Directory Administrators,
      o=example.com Bolivia\, S.A.");)
```

프록시 인증 ACI 예

프록시 인증 방법은 자신의 아이디를 사용하여 디렉토리에 바인드하는 사용자에게 프록시 인증을 통해 다른 사용자의 권한을 부여하는 특별한 인증 형식입니다.

이 예에서는 다음과 같이 가정합니다.

- 클라이언트 응용 프로그램의 바인드 DN은 uid=MoneyWizAcctSoftware, ou=Applications, dc=example, dc=com입니다.
- 클라이언트 응용 프로그램이 액세스를 요청하는 대상 하위 트리는 ou=Accounting, dc=example, dc=com입니다.
- ou=Accounting, dc=example, dc=com 하위 트리에 대한 액세스 권한을 가진 계정 관리자가 디렉토리에 있습니다.

클라이언트 응용 프로그램이 계정 관리자와 동일한 액세스 권한을 사용하여 Accounting 하위 트리에 액세스하려면 다음과 같은 조건을 만족해야 합니다.

- 계정 관리자에게 ou=Accounting, dc=example, dc=com 하위 트리에 대한 액세스 권한이 있어야 합니다. 예를 들어, 아래 ACI는 계정 관리자 항목에 모든 권한을 부여합니다.

```
aci: (target="ldap:///ou=Accounting,dc=example,dc=com")
      (targetattr="*") (version 3.0; acl "allowAll-AcctAdmin"; allow
      (all) userdn="ldap:///dn:uid=AcctAdministrator,ou=Administrators,
      dc=example,dc=com");)
```

- 클라이언트 응용 프로그램에 프록시 권한을 부여하는 아래 ACI가 디렉토리에 있어야 합니다.

```
aci: (target="ldap:///ou=Accounting,dc=example,dc=com")
(targetattr="*") (version 3.0; acl "allowproxy-
accountingsoftware"; allow (proxy) userdn=
"ldap:///dn:uid=MoneyWizAcctSoftware,ou=Applications,
dc=example,dc=com";)
```

이 ACI가 디렉토리에 있으면 MoneyWizAcctSoftware 클라이언트 응용 프로그램은 디렉토리에 바인드하여 프록시 DN의 액세스 권한이 필요한 `ldapsearch` 또는 `ldapmodify`와 같은 LDAN 명령을 전송할 수 있습니다.

위의 예에서 클라이언트가 `ldapsearch` 명령을 수행하려면 명령에 다음과 같은 컨트롤이 포함됩니다.

```
ldapsearch \
-D "uid=MoneyWizAcctSoftware,ou=Applications,dc=example,dc=com" \
-w password \
-y "uid=AcctAdministrator,ou=Administrators,dc=example,dc=com" \ ...
```

클라이언트는 자신으로 바인드하지만 프록시 항목의 권한이 부여되며 프록시 항목의 비밀번호를 제공할 필요가 없습니다.

주 디렉토리 관리자의 DN은 프록시 DN으로 사용할 수 없으며 디렉토리 관리자에게 프록시 권한을 부여할 수도 없습니다. 또한 Directory Server가 한 개의 바인드 작업에서 프록시 인증 제어를 둘 이상 수신하면 클라이언트 응용 프로그램에 오류가 반환되고 바인드 시도는 실패합니다.

유효 권한 보기

디렉토리 항목에 대한 액세스 정책을 유지 관리할 때 정의한 ACI의 보안에 미치는 영향을 확인할 수 있다면 큰 도움이 됩니다. Directory Server에서는 기존 ACI를 평가하고 지정된 항목의 특정 사용자에게 부여되는 유효 권한을 보고할 수 있습니다.

Directory Server는 검색 작업에 포함할 수 있는 "유효 권한 보기" 컨트롤에 응답하여 항목과 속성에 대한 유효 권한 정보를 검색 결과로 반환합니다. 이 추가 정보에는 각 항목 및 항목에 있는 각 속성에 대한 읽기 및 쓰기 권한이 포함됩니다. 검색에 사용된 바인드 DN이나 임의의 DN에 대한 권한을 요청할 수 있으므로 관리자는 디렉토리 사용자의 권한을 테스트할 수 있습니다.

유효 권한 기능은 LDAP 컨트롤을 사용합니다. 연결 접미어에 대한 유효 권한을 보려면 168페이지의 "연결 정책 구성"에 설명된 것처럼 연결 정책에서 이 컨트롤을 활성화해야 합니다. 원격 서버에 바인드할 때 사용한 프록시 아이디도 유효 권한 속성에 액세스할 수 있어야 합니다.

유효 권한 보기 컨트롤에 대한 액세스 제한

유효 권한 보기는 그 자체가 디렉토리 작업이므로 보호 및 적절한 제한이 필요합니다.

이 정보에 대한 디렉토리 사용자의 액세스를 제한하려면 `getEffectiveRights` 속성에 대한 기본 ACI를 수정하고 `getEffectiveRightsInfo` 속성에 대한 새 ACI를 만들어야 합니다.

예를 들어 아래의 ACI를 사용하면 디렉토리 관리자 그룹의 구성원만 유효 권한을 받을 수 있습니다.

```
aci: (targetattr != "aci")(version 3.0; acl "getEffectiveRights";
allow(all) groupdn = "ldap:///cn=Directory
Administrators,ou=Groups,dc=example,dc=com";)
```

유효 권한 보기 컨트롤 사용

`ldapsearch` 명령을 `-J "1.3.6.1.4.1.42.2.27.9.5.2"` 옵션과 함께 사용하여 "유효 권한 보기" 컨트롤을 지정합니다. 기본적으로 이 컨트롤은 항목과 속성에 대한 바인드 DN 항목의 유효 권한을 검색 결과로 반환합니다. 기본 동작을 변경하려면 다음과 같은 옵션을 사용합니다.

- `-c "dn: DN"` - 지정된 DN을 사용한 사용자 바인드의 유효 권한을 검색 결과로 표시합니다. 관리자는 이 옵션을 사용하면 다른 사용자의 유효 권한을 확인할 수 있습니다. `-c "dn:"` 옵션은 익명 인증에 대한 유효 권한을 표시합니다.
- `-x "attributeName ..."` - 지정된 속성에 대한 유효 권한도 검색 결과에 포함됩니다. 검색 결과에 표시되지 않는 속성을 지정하려면 이 옵션을 사용합니다. 예를 들어, 사용자가 현재 항목에 없는 속성을 추가할 수 있는 권한을 갖고 있는지 확인할 수 있습니다.

-c 및 -x 속성 중 한 개나 둘 모두를 사용할 경우 "유효 권한 보기" 컨트롤의 OID에 -J 옵션이 암묵적으로 지정되므로 별도로 지정할 필요가 없습니다. 유효 권한 컨트롤에 NULL 값을 지정하면 현재 사용자에게 대한 권한과 현재 ldapsearch 작업에서 반환하는 속성 및 항목에 대한 권한이 검색됩니다.

그런 다음, 간단히 권한만 볼 것인지 아니면 이러한 권한이 어떻게 부여 또는 거부되는지 설명하는 자세한 로깅 정보를 볼 것인지 선택합니다. 각각 aclRights 또는 aclRightsInfo를 검색 결과로 반환할 속성에 추가하여 정보 유형을 결정합니다. 간단한 권한 정보는 자세한 로깅 정보의 내용과 중복되지만 두 속성을 모두 요청하여 유효 권한 정보를 모두 받을 수도 있습니다.

주 aclRights 및 aclRightsInfo 속성은 가상의 작동 가능 속성으로 동작합니다. 두 속성은 디렉토리에 저장되지 않으므로 명시적으로 요청해야만 반환됩니다. Directory Server는 "유효 권한 보기" 컨트롤에 응답하여 두 속성을 생성합니다.

이런 이유로 두 속성은 모든 종류의 검색 작업이나 필터에 사용할 수 없습니다.

유효 권한 기능은 검색 작업을 시작한 사용자로부터 액세스 제어에 영향을 주는 기타 매개 변수(시간, 인증 방법, 시스템 주소, 이름 등)를 상속합니다.

아래 예에서는 어떻게 사용자가 디렉토리에서 자신의 권한을 볼 수 있는지 보여줍니다. 결과에서 1은 권한이 부여된 것을, 0은 권한이 거부된 것을 의미합니다.

```
ldapsearch -J "1.3.6.1.4.1.42.2.27.9.5.2" \
-h rousseau.example.com -p 389 \
-D "uid=cfuente,ou=People,dc=example,dc=com" \
-w password -b "dc=example,dc=com" \
"(objectclass=*)" aclRights

dn: dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: ou=Groups, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: cn=Accounting Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: cn=HR Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
```

```
dn: uid=bjensen,ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
```

```
dn: uid=cfuente, ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:1,proxy:0
```

이 결과는 Carla Fuente에게 최소한 읽기 권한이 있는 디렉토리 항목을 보여주며 자신의 항목을 수정할 수 있음을 표시합니다. 유효 권한 컨트롤은 일반 액세스 권한을 무시하지 않으므로 사용자는 읽기 권한이 없는 항목을 볼 수 없습니다. 아래 예에서 디렉토리 관리자 Carla Fuente에게 읽기 권한이 없는 항목을 볼 수 있습니다.

```
ldapsearch -h rousseau.example.com -p 389 \
-D "cn=Directory Manager" -w password \
-c "dn: uid=cfuente,ou=People,dc=example,dc=com" \
-b "dc=example,dc=com" \
"(objectclass=*)" aclRights
```

```
dn: dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
```

```
dn: ou=Groups, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
```

```
dn: cn=Directory Administrators, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:0,write:0,proxy:0
```

```
dn: ou=Special Users,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:0,write:0,proxy:0
```

```
dn: ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
```

```
dn: cn=Accounting Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
```

```
dn: cn=HR Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
```

```
dn: uid=bjensen,ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
```

```
dn: uid=cfuente, ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:1,proxy:0
```

위의 결과에서 디렉토리 관리자는 Carla Fuente가 디렉토리 트리의 Special Users나 Directory Administrators 분기를 볼 수도 없다는 것을 확인할 수 있습니다. 아래 예에서 디렉토리 관리자는 Carla Fuente가 자신의 항목에 있는 mail 속성과 manager 속성을 수정할 수 없다는 것을 확인할 수 있습니다.

```
ldapsearch -h rousseau.example.com -p 389 \
-D "cn=Directory Manager" -w password \
-c "dn: uid=cfuente,ou=People,dc=example,dc=com" \
-b "dc=example,dc=com" \
"(uid=cfuente)" aclRights "*"

version: 1
dn: uid=cfuente, ou=People, dc=example,dc=com

aclRights;attributeLevel;mail: search:1,read:1,compare:1,
write:0,selfwrite_add:0,selfwrite_delete:0,proxy:0
mail: cfuente@example.com

aclRights;attributeLevel;uid: search:1,read:1,compare:1,
write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
uid: cfuente

aclRights;attributeLevel;givenName: search:1,read:1,compare:1,
write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
givenName: Carla

aclRights;attributeLevel;sn: search:1,read:1,compare:1,
write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
sn: Fuente

aclRights;attributeLevel;cn: search:1,read:1,compare:1,
write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
cn: Carla Fuente

aclRights;attributeLevel;userPassword: search:0,read:0,
compare:0,write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
userPassword: {SSHA}wnbWHIq2HPiY/5ECwe6MWBGx2KMiZ8JmjF80Ow==

aclRights;attributeLevel;manager: search:1,read:1,compare:1,
write:0,selfwrite_add:0,selfwrite_delete:0,proxy:0
manager: uid=bjensen,ou=People,dc=example,dc=com

aclRights;attributeLevel;telephoneNumber: search:1,read:1,compare:1,
write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
telephoneNumber: (234) 555-7898
```

```
aclRights;attributeLevel;objectClass: search:1,read:1,compare:1,  
  write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0  
objectClass: top  
objectClass: person  
objectClass: organizationalPerson  
objectClass: inetorgperson  
  
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
```

유효 권한 결과 이해

유효 권한 요청은 지정한 옵션에 따라 다음 정보를 반환합니다.

- [권한 정보](#)
- [Write, Selfwrite_add 및 Selfwrite_delete 권한](#)
- [로깅 정보](#)

권한 정보

유효 권한 정보는 다음 하위 유형에 따라 제공됩니다.

<code>aclRights;entrylevel</code>	항목 수준의 권한 정보 제공
<code>aclRights;attributelevel</code>	속성 수준의 권한 정보 제공
<code>aclRightsInfo;entrylevel</code>	항목 수준의 로깅 정보 제공
<code>aclRightsInfo;attributelevel</code>	속성 수준의 로깅 정보 제공

`aclRights` 문자열의 형식은 `permission:value(permission:value)*`입니다.

사용 가능한 항목 수준 권한은 `add`, `delete`, `read`, `write`, `import`, `export` 및 `proxy`입니다. 가능한 속성 수준 권한에는 `read`, `search`, `compare`, `write`, `selfwrite_add`, `selfwrite_delete` 및 `proxy`가 있습니다.

이러한 권한은 다음 중 하나를 값으로 가질 수 있습니다.

- 0 - 부여됨
- 1 - 부여되지 않음
- ? - 추가, 삭제 또는 바꾸는 속성 값에 따라 권한 부여가 결정되는 경우. ?가 있으면 로깅 정보를 참조하여 권한이 부여되거나 부여되지 않는 이유를 구성합니다.
- -- 해당 속성이 가상 속성이므로 업데이트할 수 없음. 가상 속성을 수정하려면 해당 속성을 생성한 기법을 수정해야 합니다.

Write, Selfwrite_add 및 Selfwrite_delete 권한

Directory Server 5.2에서는 write 속성 수준 권한만 "?" 값을 가질 수 있습니다. add 및 delete 권한을 사용하여 추가하고 삭제할 수 있는 항목은 해당 항목의 속성 값에 따라 달라집니다. "?"를 반환하는 대신, ldapsearch 작업에서와 같이 항목에 대한 권한(0 또는 1)이 반환됩니다.

write 권한 값이 1이면 인증 dn 값을 제외한 모든 값에 대해 추가 및 삭제 ldapmodify 작업을 수행할 수 있습니다. write 권한 값이 0이면 인증 dn 값을 제외한 모든 값에 대해 추가 또는 삭제 ldapmodify 작업을 수행할 수 없습니다. 인증 dn 값에 적용되는 권한은 selfwrite 권한 중 하나인 selfwrite_add 또는 selfwrite_delete를 통해 명시적으로 반환됩니다.

ACI에는 이러한 기능을 수행하는 selfwrite-add 및 selfwrite-delete 속성 수준 권한이 없으므로 ACI 집합은 수정 작업의 오직 추가 부분이나 오직 삭제 부분에 대한 selfwrite 권한만 사용자에게 부여할 수 있습니다. selfwrite 권한을 사용하여 수정하는 속성 값이 인증 dn입니다. write 권한의 경우 수정하는 속성 값이 정의되어 있지 않으므로 이런 방법으로 차별화할 수 없습니다.

targattrfilters ACI에 따라 유효 권한이 결정될 때 "?" 값이 표시되면 로깅 정보에서 자세한 권한 정보를 확인해야 합니다. write, selfwrite_add 및 selfwrite_delete 권한 간의 복잡한 상호 종속성 관계를 고려하여, 표 6-3에서는 세 권한의 각 조합이 의미하는 바를 설명합니다.

표 6-3 유효 권한 상호 종속성

쓰기	selfwrite_add	selfwrite_delete	유효 권한 설명
0	0	0	이 속성의 어떤 값도 추가하거나 삭제할 수 없습니다.
0	0	1	인증 dn 값만 삭제할 수 있습니다.
0	1	0	인증 dn 값만 추가할 수 있습니다.
0	1	1	인증 dn 값만 추가하거나 삭제할 수 있습니다.
1	0	0	인증 dn을 제외한 모든 값을 추가하거나 삭제할 수 있습니다.
1	0	1	인증 dn을 포함 모든 값을 삭제할 수 있으며, 인증 dn을 제외한 모든 값을 추가할 수 있습니다.
1	1	0	인증 dn을 포함 모든 값을 추가할 수 있으며, 인증 dn을 제외한 모든 값을 삭제할 수 있습니다.
1	1	1	이 속성의 모든 값을 추가하거나 삭제할 수 있습니다.
?	0	0	인증 dn 값을 추가하거나 삭제할 수는 없지만, 다른 값을 추가하거나 삭제할 수 있습니다. write 권한에 대한 자세한 내용은 로깅 정보를 참조하십시오.

표 6-3 유효 권한 상호 종속성

쓰기	selfwrite_ add	selfwrite_ delete	유효 권한 설명
?	0	1	인증 dn 값을 삭제할 수만 있고 추가할 수 없으며, 다른 값을 추가하거나 삭제할 수 있습니다. write 권한에 대한 자세한 내용은 로깅 정보를 참조하십시오.
?	1	0	인증 dn 값을 추가할 수만 있고 삭제할 수 없으며, 다른 값을 추가하거나 삭제할 수 있습니다. write 권한에 대한 자세한 내용은 로깅 정보를 참조하십시오.
?	1	1	인증 dn 값을 추가하고 삭제할 수 있으며, 다른 값의 추가를 수정하거나 삭제를 수정할 수 있습니다. write 권한에 대한 자세한 내용은 로깅 정보를 참조하십시오.

로깅 정보

유효 권한 로깅 정보를 사용하여 액세스 제어 문제점을 파악하고 디버그할 수 있습니다. 로깅 정보에는 액세스 제어가 허용 또는 거부된 이유를 나타내는 액세스 제어 요약문이 포함되어 있으며, 이를 `ac1_summary`라고 합니다. 액세스 제어 요약문은 다음 정보를 제공합니다.

- 액세스가 허용 또는 거부되었는지 여부
- 부여된 권한
- 권한의 대상 항목
- 대상 속성의 이름
- 요청되는 권한의 주제
- 프록시에 의해 요청된 것인지 여부, 프록시에 의해 요청된 경우 프록시 인증 DN
- 액세스를 허용 또는 거부하는 이유(디버깅 시 중요함). 가능한 이유는 표 6-4에 열거되어 있습니다.

표 6-4 유효 권한 로깅 정보 이유 및 설명

로깅 정보 이유	설명
no reason available	액세스가 허용 또는 거부된 이유를 설명하기 위해 사용할 수 있는 이유가 없습니다.
no allow acis	ACI가 허용되지 않으므로 액세스가 거부됩니다.
result cached deny	캐시된 정보를 통해 액세스 거부가 결정되었습니다.
result cached allow	캐시된 정보를 통해 액세스 허용이 결정되었습니다.
evaluated allow	ACI 평가를 통해 액세스 허용이 결정되었습니다. ACI 이름은 로그 정보에 포함되어 있습니다.
evaluated deny	ACI 평가를 통해 액세스 거부가 결정되었습니다. ACI 이름은 로그 정보에 포함되어 있습니다.
no acis matched the resource	자원 또는 대상과 일치하는 ACI가 없으므로 액세스가 거부됩니다.
no acis matched the subject	액세스 제어를 요청하는 주제와 일치하는 ACI가 없으므로 액세스가 거부됩니다.
allow anyone aci matched anon user	userdn = "ldap:///anyone" 주제를 가진 ACI가 익명 사용자에게 액세스를 허용했습니다.
no matching anyone aci for anon user	userdn= "ldap:///anyone" 주제를 가진 ACI를 찾을 수 없으므로 익명 사용자의 액세스가 거부되었습니다.
user root	사용자가 루트 DN이므로 액세스가 허용됩니다.

주 가상 속성은 업데이트할 수 없으므로 가상 속성에 대한 **write** 권한이나 관련된 로깅 평가 정보는 제공되지 않습니다.

정확한 로그 파일 형식은 *Directory Server Administration Reference*를 참조하십시오.

고급 액세스 제어: 매크로 ACI 사용

반복 디렉토리 트리 구조를 사용하는 조직에서는 매크로를 사용하여 디렉토리에 사용되는 ACI 수를 최소화할 수 있습니다. 디렉토리 트리의 ACI 수를 줄이면 액세스 제어 정책의 관리가 용이해지며 ACI 메모리를 효율적으로 사용할 수 있습니다.

매크로는 ACI에서 DN 또는 DN의 일부분을 나타내는 자리 표시자입니다. 매크로를 사용하여 ACI의 대상 부분, 바인드 규칙 부분 또는 두 부분에서 모두 DN을 나타낼 수 있습니다. 실제로 LDAP 작업이 수신되면 Directory Server는 ACI 매크로를 LDAP 작업의 대상 자원과 비교하여 일치하는 하위 문자열이 있는지 확인합니다. 일치하는 하위 문자열이 있으면 이 문자열을 사용하여 바인드 규칙측 매크로를 확장하고 확장된 바인드 규칙을 평가하여 자원에 대한 액세스를 결정합니다.

매크로 ACI 예

예를 통해 살펴보면 매크로 ACI의 이점과 작동 방식을 명확히 이해할 수 있습니다. [306페이지의 그림 6-4](#)에서는 매크로 ACI를 사용하여 효과적으로 전체 ACI 수를 줄일 수 있는 디렉토리 트리를 보여줍니다.

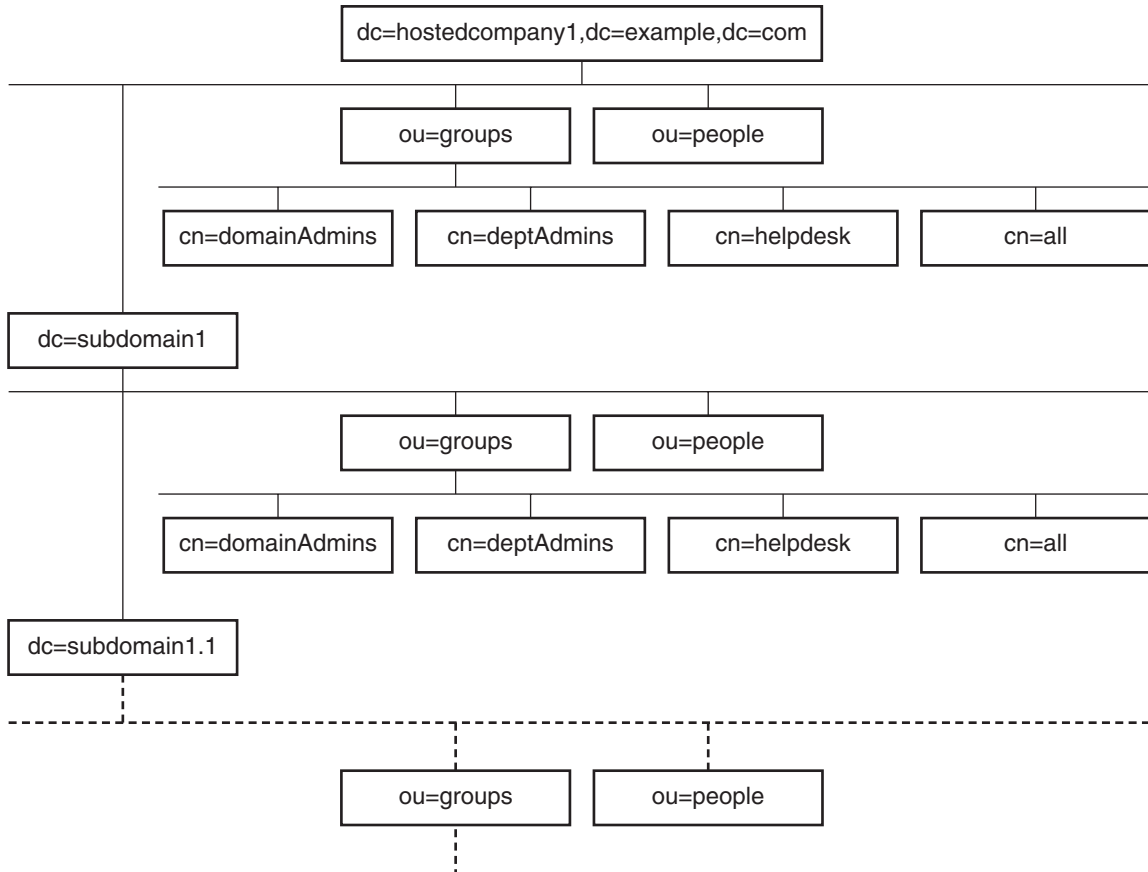
이 그림에서는 하위 도메인의 반복 패턴이 동일한 트리 구조(ou=groups, ou=people)를 갖습니다. example.com 디렉토리 트리에 dc=hostedCompany2, dc=example, dc=com 및 dc=hostedCompany3, dc=example, dc=com 접미어가 저장되기 때문에 이 패턴은 트리 전체에도 반복됩니다.

디렉토리 트리에 적용되는 ACI에도 반복 패턴이 있습니다. 예를 들어 dc=hostedCompany1, dc=example, dc=com 노드에는 아래 ACI가 있습니다.

```
aci: (targetattr="*")
  (targetfilter=(objectClass=nsManagedDomain)) (version 3.0;
  aci "Domain access"; allow (read,search) groupdn=
  "ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany1,
  dc=example,dc=com";)
```

이 ACI는 DomainAdmins 그룹에 dc=hostedCompany1, dc=example, dc=com 트리의 모든 항목에 대한 읽기 및 검색 권한을 부여합니다.

그림 6-4 매크로 ACI의 디렉토리 트리 예



dc=hostedCompany1,dc=example,dc=com 노드에는 아래 ACI가 있습니다.

```

aci: (targetattr="*")
  (targetfilter=(objectClass=nsManagedDomain))
  (version 3.0; acl "Domain access"; allow (read,search)
  groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany1,
dc=example,dc=com";)
  
```

dc=subdomain1,dc=hostedCompany1,dc=example,dc=com 노드에는 아래 ACI가 있습니다.

```
aci: (targetattr="*")
  (targetfilter=(objectClass=nsManagedDomain))
  (version 3.0; acl "Domain access"; allow (read,search)
  groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=subdomain1,
  dc=hostedCompany1,dc=example,dc=com";)
```

dc=hostedCompany2,dc=example,dc=com 노드에는 아래 ACI가 있습니다.

```
aci: (targetattr="*")
  (targetfilter=(objectClass=nsManagedDomain))
  (version 3.0; acl "Domain access"; allow (read,search)
  groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany2,
  dc=example,dc=com";)
```

dc=subdomain1,dc=hostedCompany2,dc=example,dc=com 노드에는 아래 ACI가 있습니다.

```
aci: (targetattr="*")
  (targetfilter=(objectClass=nsManagedDomain))
  (version 3.0; acl "Domain access"; allow (read,search)
  groupdn="ldap:///cn=DomainAdmins,ou=Groups, dc=subdomain1,
  dc=hostedCompany2,dc=example,dc=com";)
```

네 개의 ACI에서 유일한 차이점은 groupdn 키워드에 지정된 DN입니다. 이 경우 DN에 매크로를 사용하여 dc=example,dc=com 노드에서 트리의 루트에 있는 한 개의 ACI로 네 개의 ACI를 대체할 수 있습니다. 이 ACI는 다음과 같이 표시됩니다.

```
aci: (target="ldap:///ou=Groups, ($dn),dc=example,dc=com")
  (targetattr="*") (targetfilter=(objectClass=nsManagedDomain))
  (version 3.0; acl "Domain access"; allow (read,search) groupdn=
  "ldap:///cn=DomainAdmins,ou=Groups, [$dn],dc=example,dc=com";
  )
```

이 경우 앞에서 사용하지 않았던 target 키워드가 사용되었습니다.

위의 예에서는 ACI 수가 4개에서 1개로 줄었습니다. 하지만 전체 디렉토리 트리의 반복 패턴 수가 줄어든다는 것에 가장 큰 이점이 있습니다.

매크로 ACI 구문

이 절에서는 설명을 간소화하기 위해 `userdn`, `roledn`, `groupdn`, `userattr` 등의 바인드 자격 증명을 제공하는 ACI 키워드를 모두 ACI의 *주제*라고 부릅니다. 주제는 ACI의 적용 대상을 결정합니다.

매크로 ACI에는 DN 또는 DN의 일부분을 대체하는 다음과 같은 유형의 표현식이 포함됩니다.

- `($dn)` - 대상 일치 및 주제의 직접 대체에 사용합니다.
- `[$dn]` - 주제의 하위 트리에서 작동하는 여러 RDN을 대체하는 데 사용합니다.
- `($attr.attributeName)` - 대상 항목의 `attributeName` 속성 값을 주제로 대체하는 데 사용합니다.

표 6-5에는 ACI에서 DN 매크로를 사용할 수 있는 부분이 나와 있습니다.

표 6-5 ACI 키워드의 매크로

매크로	ACI 키워드
<code>(\$dn)</code>	<code>target</code> , <code>targetfilter</code> , <code>userdn</code> , <code>roledn</code> , <code>groupdn</code> , <code>userattr</code>
<code>[\$dn]</code>	<code>targetfilter</code> , <code>userdn</code> , <code>roledn</code> , <code>groupdn</code> , <code>userattr</code>
<code>(\$attr.attrName)</code>	<code>userdn</code> , <code>roledn</code> , <code>groupdn</code> , <code>userattr</code>

다음과 같은 제한이 적용됩니다.

- 주제에서 `($dn)` 및 `[$dn]` 매크로를 사용할 경우 `($dn)` 매크로가 포함된 대상을 *받드시* 정의해야 합니다.
- 주제에서 `($dn)` 매크로는 `($attr.attrName)` 매크로와 함께 사용할 수 있지만 `[$dn]` 매크로는 함께 사용할 수 없습니다.

대상의 (\$dn) 일치

ACI의 대상에 있는 `($dn)` 매크로는 LDAP 요청의 대상 항목에 따라 대체 값을 결정합니다. 예를 들어, `cn=all,ou=groups,dc=subdomain1,dc=hostedCompany1,dc=example,dc=com` 항목을 대상으로 하는 LDAP 요청과 대상을 다음과 같이 정의하는 ACI가 있습니다.

```
(target="ldap:///ou=Groups,($dn),dc=example,dc=com")
```

`($dn)` 매크로는 "dc=subdomain1, dc=hostedCompany1"과 일치합니다. 이 경우 ACI 주제에 이 하위 문자열이 대체됩니다.

주제의 (\$dn) 대체

ACI 주제에 있는 (\$dn) 매크로는 대상에서 일치하는 전체 하위 문자열로 대체됩니다. 예를 들면 다음과 같습니다.

```
groupdn="ldap:///cn=DomainAdmins,ou=Groups,($dn),
dc=example,dc=com"
```

위의 ACI 주제는 다음과 같이 표시됩니다.

```
groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=subdomain1,
dc=hostedCompany1,dc=example,dc=com"
```

매크로가 확장되면 Directory Server는 일반 프로세스에 따라 ACI를 평가하여 액세스 권한을 부여할지 여부를 결정합니다.

주 매크로 대체를 사용하는 ACI는 표준 ACI와 달리 반드시 대상 항목의 자식에 대한 액세스 권한을 부여하지는 않습니다. 이는 자식의 DN이 대상일 경우 대체로 인해 주제 문자열에 잘못된 DN이 작성될 수 있기 때문입니다.

주제의 [\$dn] 대체

[\$dn]의 대체 기법은 (\$dn)의 경우와 약간 다릅니다. 일치를 발견할 때까지 대상 자원의 DN을 여러 번 검사하며 매번 가장 왼쪽의 RDN 구성 요소를 삭제합니다.

예를 들어, cn=all,ou=groups, dc=subdomain1,dc=hostedCompany1, dc=example,dc=com 하위 트리를 대상으로 하는 LDAP 요청과 다음과 같은 ACI가 있습니다.

```
aci: (targetattr="*")
(target="ldap:///ou=Groups,($dn),dc=example,dc=com")
(version 3.0; acl "Domain access"; allow (read,search)
groupdn="ldap:///cn=DomainAdmins,ou=Groups,[$dn],
dc=example,dc=com";)
```

서버는 다음과 같이 이 ACI를 확장합니다.

1. 대상의 (\$dn)은 dc=subdomain1,dc=hostedCompany1과 일치합니다.
2. 주제의 [\$dn]을 dc=subdomain1,dc=hostedCompany1로 대체합니다.

따라서 주제는 groupdn="ldap:///cn=DomainAdmins,ou=Groups, dc=subdomain1,dc=hostedCompany1,dc=example,dc=com"이 됩니다. 바인드 DN이 이 그룹의 구성원이어서 액세스 권한이 부여되면 매크로 확장이 중단되고 ACI 평가가 수행됩니다. 바인드 DN이 그룹의 구성원이 아니면 프로세스가 계속됩니다.

3. 주제의 [\$dn]을 dc=hostedCompany1로 대체합니다.

따라서 주제는 `groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany1,dc=example,dc=com"`이 됩니다. 다시 바인드 DN이 이 그룹의 구성원인지 테스트하여, 구성원일 경우 ACI를 완전히 평가합니다. 그룹의 구성원이 아니면 일치한 값의 마지막 RDN에서 매크로 확장이 중단되고 이 ACI에 대한 평가가 완료됩니다.

[\$dn] 매크로를 사용할 경우 도메인 수준 관리자에게 디렉토리 트리의 모든 하위 도메인에 대한 액세스 권한을 유연성 있게 부여할 수 있다는 장점이 있습니다. 따라서 이 매크로는 도메인간 계층 관계를 표시할 때 유용합니다.

예를 들어, 다음과 같은 ACI를 가정해 보십시오.

```
aci: (target="ldap:///ou=*,($dn),dc=example,dc=com")
      (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
      (version 3.0; acl "Domain access"; allow (read,search) groupdn=
        "ldap:///cn=DomainAdmins,ou=Groups,[$dn],dc=example,dc=com";)
```

이 ACI는 `cn=DomainAdmins,ou=Groups,dc=hostedCompany1,dc=example,dc=com` 구성원에게 `dc=hostedCompany1`의 모든 하위 도메인에 대한 액세스 권한을 부여하므로 이 그룹에 속한 관리자는 하위 트리(예: `ou=people,dc=subdomain1.1,dc=subdomain1`)에 액세스할 수 있습니다.

하지만 이와 동시에 `cn=DomainAdmins,ou=Groups,dc=subdomain1.1`의 구성원에게는 `ou=people,dc=subdomain1,dc=hostedCompany1` 및 `ou=people,dc=hostedCompany1` 노드에 대한 액세스가 거부됩니다.

(\$attr.attrName)의 매크로 일치

(\$attr.attrname) 매크로는 항상 DN의 주제 부분에서 사용됩니다. 예를 들어 다음과 같은 roledn을 정의할 수 있습니다.

```
roledn = "ldap:///cn=DomainAdmins,($attr.ou),dc=HostedCompany1,
dc=example,dc=com"
```

이제 서버가 아래 항목을 대상으로 하는 LDAP 작업을 수신한다고 가정해 보십시오.

```
dn: cn=Babs Jensen,ou=People,dc=HostedCompany1,dc=example,dc=com
cn: Babs Jensen
sn: Jensen
ou: Sales
...
```

ACI의 roledn 부분을 평가하기 위해 서버는 대상 항목에 저장된 ou 속성 값을 읽고 주제에 이 값을 대체하여 매크로를 확장합니다. 이 예에서 roledn은 다음과 같이 확장됩니다.

```
roledn = "ldap:///cn=DomainAdmins,ou=Sales,dc=HostedCompany1,
dc=example,dc=com"
```

그런 다음 Directory Server는 일반 ACI 평가 알고리즘에 따라 ACI를 평가합니다.

매크로에 지정된 속성이 여러 값을 갖는 경우 각 값을 차례로 사용하여 매크로를 확장하며 처음 일치하는 항목을 제공한 값이 사용됩니다.

액세스 제어 및 복제

ACI는 항목의 속성으로 저장되므로 ACI가 포함된 항목이 복제된 접미어의 일부일 경우 ACI도 다른 속성과 마찬가지로 복제됩니다.

ACI는 항상 받는 LDAP 요청을 처리하는 Directory Server에서 평가됩니다. 따라서 업데이트 요청이 수신되면 소비자 서버는 마스터 서버에서 이 요청을 처리할 수 있는지 여부를 평가하기 전에 마스터 서버에 대한 참조를 반환합니다.

액세스 제어 및 연결

연결을 사용하여 여러 서버에 디렉토리 트리를 배포하는 경우 액세스 제어 명령문에서 사용할 수 있는 키워드에 몇 가지 제한이 적용됩니다. 자세한 내용은 [240페이지의 "ACI 제한"](#)을 참조하십시오.

인증된 사용자가 연결 접미어에 액세스하면 서버는 이 사용자의 아이디를 원격 서버로 보냅니다. 액세스 제어는 항상 원격 서버에서 평가됩니다. 원격 서버에서 평가되는 모든 LDAP 작업은 프록시 인증 제어를 통해 전달된 클라이언트 응용 프로그램의 원래 아이디를 사용합니다. 사용자에게 원격 서버의 하위 트리에 대한 올바른 액세스 제어가 있을 경우에만 원격 서버 작업이 제대로 수행됩니다. 따라서 다음과 같은 몇 가지 제한을 고려하여 원격 서버에 일반적인 액세스 제어를 추가할 필요가 있습니다. 자세한 내용은 [166페이지의 "연결 접미어를 통한 액세스 제어"](#)를 참조하십시오.

액세스 제어 로깅 정보

오류 로그에서 액세스 제어에 대한 정보를 얻으려면 적절한 로그 수준을 설정해야 합니다.

콘솔에서 오류 로그 수준을 설정하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 수준 디렉토리 탭에서 `cn=config` 노드를 마우스 오른쪽 버튼으로 누른 다음 팝업 메뉴에서 일반 편집기로 편집을 선택합니다.
`cn=config` 항목의 내용이 표시된 일반 편집기가 나타납니다.
2. 속성 값 쌍의 목록을 아래로 스크롤하여 `nsslapd-errorlog-level` 속성을 찾습니다.
3. `nsslapd-errorlog-level` 필드의 값에 128을 더합니다.
예를 들어, 표시된 값이 8192(복제 디버깅)이면 이 값을 8320으로 변경해야 합니다. 오류 로그 수준에 대한 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.
4. 확인을 눌러 변경 사항을 저장하고 일반 편집기를 닫습니다.

이전 릴리스와의 호환성

Directory Server의 이전 릴리스에서 사용되었던 일부 ACI 키워드는 Directory Server 5.2에서 더 이상 사용되지 않습니다. 하지만 이전 버전과의 호환성을 위해 이러한 키워드도 계속 지원됩니다. 예를 들면 다음과 같습니다.

- userdnattr
- groupdnattr

따라서 레거시 공급업체 서버와 소비자 Directory Server 5.2 간에 복제 계약을 설정한 경우에도 ACI 복제 시 문제가 발생하지 않습니다.

하지만 [260페이지의 "값 일치에 따른 액세스 정의"](#)에 설명된 것처럼 이러한 키워드는 userattr 키워드의 기능으로 바꾸는 것이 좋습니다.

사용자 계정 및 비밀번호 관리

사용자가 Directory Server에 연결하면, 사용자가 인증되고 디렉토리는 인증 도중 설정된 아이디에 따라 사용자에게 액세스 권한 및 자원 제한을 부여할 수 있습니다.

이 장에서는 디렉토리에 대한 비밀번호 및 계정 잠금 정책 구성, 디렉토리에 액세스할 수 없도록 계정 또는 사용자 그룹 비활성화, 바인드 DN에 따라 사용자가 사용할 수 있는 시스템 자원 제한 등의 사용자 계정 관리 작업에 대해 설명합니다.

Directory Server는 개별 비밀번호 정책을 지원합니다. 다양한 비밀번호 정책을 정의하여 이 정책 중 하나를 특정 사용자 또는 사용자 그룹에 적용할 수 있으므로 디렉토리에 액세스할 수 있는 사용자 유형을 쉽게 제어할 수 있습니다.

이 장은 다음 내용으로 구성되어 있습니다.

- 비밀번호 정책에 대한 개요
- 전역 비밀번호 정책 구성
- 개별 비밀번호 정책 관리
- 사용자 비밀번호 재설정
- 사용자와 역할 비활성화 및 활성화
- 개별 자원 제한 설정

비밀번호 정책에 대한 개요

보안 **비밀번호 정책**은 다음과 같은 요구 사항을 강제하여 쉽게 추측할 수 있는 비밀번호로 인한 위험을 최소화합니다.

- 사용자는 일정에 따라 비밀번호를 변경해야 합니다.
- 사용자는 일반적이지 않은 비밀번호를 제공해야 합니다.
- 잘못된 비밀번호를 사용하여 여러 번 바인드를 시도하면 해당 계정이 잠길 수 있습니다.

Directory Server는 개별 및 전역 비밀번호 정책을 모두 지원합니다. 개별 비밀번호 정책은 디렉토리 트리의 하위 항목에서 정의되어 해당 정책이 할당된 사용자 항목에서 참조됩니다. 사용자 항목이 개별 정책을 참조하지 않으면 `cn=PasswordPolicy, cn=config`에 있는 전역 비밀번호 정책이 적용됩니다.

비밀번호 정책은 사용자 항목에 있는 `userPassword` 속성에만 적용되며, 이 속성에 기반한 인증 도중에 적용됩니다. SASL GSSAPI 또는 SSL 기반 인증 등의 다른 인증 스키마에는 적용되지 않습니다.

다음 절에서는 비밀번호 정책의 구현 방법과 이 비밀번호 정책을 사용자 및 그룹에 할당하는 방법에 대해 설명합니다. 자세한 내용은 *Directory Server Deployment Planning Guide*를 참조하십시오.

전역 비밀번호 정책 구성

전역 비밀번호 정책은 개별 정책이 정의되어 있지 않은 디렉토리의 모든 사용자에게 적용되지만 **디렉토리 관리자**에게는 적용되지 않습니다.

콘솔에서 비밀번호 정책 구성

Directory Server의 전역 비밀번호 정책을 설정하거나 수정하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드를 선택한 다음 오른쪽 패널에서 비밀번호 탭을 선택합니다.
2. 비밀번호 탭에서 정책의 다음 측면을 설정합니다.

- "사용자가 재설정후 비밀번호를 변경해야 함" 확인란을 선택하여 사용자가 처음 로그인할 때 자신의 비밀번호를 변경하도록 지정합니다.
이 확인란을 선택하면 디렉토리 관리자만 사용자 비밀번호를 재설정할 수 있습니다. 일반 관리자는 사용자가 자신의 비밀번호를 업데이트하도록 강요할 수 없습니다.
- 사용자가 자신의 비밀번호를 변경할 수 있도록 허용하려면 "사용자가 비밀번호를 변경할 수 있음" 확인란을 선택합니다.
- 사용자가 자신의 비밀번호를 변경할 수 있는 빈도를 제한하려면 "X일 내 변경 허용" 텍스트 상자에 일 수를 입력합니다. 사용자가 원하는 대로 비밀번호를 변경할 수 있게 하려면 "제한 없음" 확인란을 선택합니다.
- 사용자가 같은 비밀번호를 계속 사용하는 것을 방지하려면, "비밀번호 기록 유지" 확인란을 선택하고 서버가 각 사용자별로 유지할 비밀번호의 수를 "보존수" 텍스트 상자에 지정합니다. 이 경우 사용자는 목록에 있는 비밀번호를 설정할 수 없습니다. 효과적으로 설정하려면 사용자가 비밀번호를 변경할 수 있는 빈도도 제한해야 합니다.
- 사용자 비밀번호가 만료되지 않도록 하려면 "비밀번호가 만료되지 않음" 라디오 버튼을 선택합니다.
- 그렇지 않으면 "X일 후 비밀번호 만료" 라디오 버튼을 선택하여 사용자가 정기적으로 비밀번호를 변경하도록 설정한 다음, 사용자 비밀번호의 유효 일 수를 입력합니다.
- 비밀번호 만료를 선택한 경우 "비밀번호가 만료되기 X일 전에 경고 보내기" 필드에 비밀번호가 만료되기 며칠 전에 사용자에게 경고를 보낼지 지정할 수 있습니다.
사용자가 경고를 받으면 비밀번호는 원래 날짜에 만료됩니다. 경고를 보낸 후 충분한 경고 기간을 제공하기 위해 만료를 연장하려면 "경고에 관계 없이 만료됨" 확인란을 선택 취소합니다. 한 번의 경고와 1회 연장만 지원됩니다. 사용자가 비밀번호 만료 후에 바인드할 경우 유예 로그인은 허용되지 않습니다.
- 서버에서 사용자 비밀번호 구문을 검사하여 비밀번호 정책에 설정된 최소 요구 사항을 충족하는지 확인하려면 "비밀번호 구문 검사" 확인란을 선택합니다. 그런 다음 "비밀번호의 최소 길이" 텍스트 상자에 허용되는 최소 비밀번호 길이를 지정합니다.
- 기본적으로 디렉토리 관리자는 기록에 있는 비밀번호의 재사용 등과 같이 비밀번호 정책에 위반되는 비밀번호를 재설정할 수 없습니다. 이를 허용하려면 "디렉토리 관리자의 비밀번호 정책 생략 허용" 확인란을 선택합니다.
- 서버에서 비밀번호를 저장할 때 사용할 암호화 방법을 "비밀번호 암호화" 풀다운 메뉴에서 지정합니다.

- 계정 잠금 탭을 누르고 "계정 잠금 기능 사용" 확인란을 선택하여 다음과 같이 계정 잠금 정책을 정의합니다.
 - 잠금을 실행하는 데 필요한 로그인 실패 횟수 및 기간을 입력합니다.
 - 영구히 잠금 라디오 버튼을 선택하여 디렉토리 관리자가 사용자 비밀번호를 재설정할 때까지 영구 잠금을 설정합니다.
 - 그렇지 않으면 잠금 기간 라디오 버튼을 선택하고 사용자 계정을 일시적으로 잠금 시간(분)을 입력합니다.
- 비밀번호 정책 변경이 끝나면 저장을 누릅니다. 새 전역 비밀번호 정책이 즉시 실행됩니다.

명령줄에서 비밀번호 정책 구성

전역 비밀번호 정책은 `cn=Password Policy,cn=config` 항목의 속성에 의해 정의됩니다. 이 항목에 있는 전역 정책을 변경하려면 `ldapmodify` 유틸리티를 사용합니다.

비밀번호 정책에서 가능한 모든 속성에 대한 정의는 *Directory Server Administration Reference*를 참조하십시오.

예를 들어, 비밀번호 구문 및 길이 검사는 기본적으로 사용되지 않으며 계정 잠금도 비활성화됩니다. 구문 검사를 활성화하고, 최소 길이를 8로 설정하고, 잘못된 비밀번호를 5회 입력할 경우 5분 동안 계정을 임시로 잠그려면 아래 명령을 사용합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=Password Policy,cn=config
changetype: modify
replace: passwordCheckSyntax
passwordCheckSyntax: on
-
replace: passwordMinLength
passwordMinLength: 8
-
replace: passwordLockout
passwordLockout: on
-
replace: passwordMaxFailure
passwordMaxFailure: 5
-
```

```
replace: passwordLockoutDuration
passwordLockoutDuration: 300
-
replace: passwordUnlock
passwordUnlock: on
```

개별 비밀번호 정책 관리

개별 비밀번호 정책은 passwordPolicy 객체 클래스가 있는 하위 항목에 정의됩니다. 디렉토리 트리에서 `cn=policy name, subtree` 형식의 DN이 있는 모든 항목에서 정책을 정의할 수 있습니다. Directory Server 콘솔이나 명령줄 유틸리티에서 비밀번호 정책을 정의한 다음 원하는 사용자 항목에 passwordPolicySubentry 속성을 설정하여 할당합니다.

이 절에서는 하위 트리 루트가 `dc=example,dc=com`인 `example.com`의 임시 직원들에 대한 비밀번호 정책 구현 예를 소개합니다.

콘솔에서 정책 정의

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 개별 비밀번호 정책 하위 항목을 정의할 항목을 표시합니다.
2. 항목을 마우스 오른쪽 버튼으로 누르고 새로 만들기 > 비밀번호 정책을 선택합니다. 또는 항목을 왼쪽 마우스 버튼으로 눌러 선택한 다음 객체 메뉴에서 새로 만들기 > 비밀번호 정책을 선택할 수도 있습니다.

비밀번호 정책 항목의 사용자 정의 편집기가 표시됩니다.

3. 일반 필드에 이 정책의 이름 및 설명(선택 사항)을 입력합니다. 이 이름은 정책을 정의하는 하위 항목의 `cn` 이름 지정 속성 값이 됩니다.
4. 비밀번호 탭을 눌러 정책의 다음 측면을 설정합니다.
 - "사용자가 재설정 후 비밀번호를 변경해야 함" 확인란을 선택하여 사용자가 처음 로그인할 때 자신의 비밀번호를 변경하도록 지정합니다. 이 확인란을 선택하면 디렉토리 관리자만 사용자 비밀번호를 재설정할 수 있습니다. 일반 관리자는 사용자가 자신의 비밀번호를 업데이트하도록 강요할 수 없습니다.
 - 사용자가 자신의 비밀번호를 변경할 수 있도록 허용하려면 "사용자가 비밀번호를 변경할 수 있음" 확인란을 선택합니다.

- 사용자가 자신의 비밀번호를 변경할 수 있는 빈도를 제한하려면 "X일 내 변경 허용" 텍스트 상자에 일 수를 입력합니다. 사용자가 원하는 대로 비밀번호를 변경할 수 있게 하려면 제한 없음 확인란을 선택합니다.
 - 사용자가 같은 비밀번호를 계속 사용할 수 없도록 차단하려면 "비밀번호 기록 유지" 확인란을 선택하고 각 사용자별로 저장될 비밀번호의 수를 지정합니다. 이 경우 사용자는 목록에 있는 비밀번호를 설정할 수 없습니다. 효과적으로 설정하려면 사용자가 비밀번호를 변경할 수 있는 빈도도 제한해야 합니다.
 - 사용자 비밀번호가 만료되지 않도록 하려면 "비밀번호가 만료되지 않음" 라디오 버튼을 선택합니다.
 - 그렇지 않으면 "X일 후 비밀번호 만료" 라디오 버튼을 선택하여 사용자가 정기적으로 비밀번호를 변경하도록 설정한 다음, 사용자 비밀번호의 유효 일 수를 입력합니다.
 - 비밀번호 만료를 선택한 경우 비밀번호가 만료되기 며칠 전에 사용자에게 경고를 보낼지 지정할 수 있습니다. "비밀번호가 만료되기 X일 전에 경고 보내기" 텍스트 상자에 경고를 보낼 비밀번호 만료 전의 일 수를 입력합니다.

사용자가 경고를 받으면 비밀번호는 원래 날짜에 만료됩니다. 경고를 보낸 후 충분한 경고 기간을 제공하기 위해 만료를 연장하려면 "경고에 관계 없이 만료됨" 확인란을 선택 취소합니다. 한 번의 경고와 1회 연장만 지원됩니다. 사용자가 비밀번호 만료 후에 바인드할 경우 유예 로그인은 허용되지 않습니다.
 - 서버에서 사용자 비밀번호 구문을 검사하여 비밀번호 정책에 설정된 최소 요구 사항을 충족하는지 확인하려면 "비밀번호 구문 검사" 확인란을 선택합니다. 그런 다음 "비밀번호의 최소 길이" 텍스트 상자에 허용되는 최소 비밀번호 길이를 지정합니다.
 - 기본적으로 디렉토리 관리자는 기록에 있는 비밀번호의 재사용 등과 같이 비밀번호 정책에 위반되는 비밀번호를 재설정할 수 없습니다. 이를 허용하려면 "디렉토리 관리자의 비밀번호 정책 생략 허용" 확인란을 선택합니다.
 - 서버에서 비밀번호를 저장할 때 사용할 암호화 방법을 "비밀번호 암호화" 풀다운 메뉴에서 지정합니다.
5. 잠금 탭을 누르고 "계정 잠금 기능 사용" 확인란을 선택하여 다음과 같이 계정 잠금 정책을 정의합니다.
- 잠금을 실행하는 데 필요한 로그인 실패 횟수 및 기간을 입력합니다.
 - 영구히 잠금 라디오 버튼을 선택하여 디렉토리 관리자가 사용자 비밀번호를 재설정할 때까지 영구 잠금을 설정합니다.

- 그렇지 않으면 잠금 기간 라디오 버튼을 선택하고 사용자 계정을 일시적으로 잠금 시간(분)을 입력합니다.
6. 사용자 정의 편집기에서 확인을 눌러 정책을 저장하고 해당 하위 항목을 작성합니다.

명령줄에서 정책 정의

이 보안 정책에서 임시 직원의 비밀번호는 100일(8 640 000초) 후에 만료되며, 비밀번호 만료 3일(259 200초) 전부터 사용자가 바인드할 때 비밀번호 만료에 대한 경고가 반환된다고 가정해 보십시오. 구문 검사를 활성화하여 비밀번호 보안에 대한 최소 검사를 강제하고, 침입자가 디렉토리 공격을 통해 비밀번호를 해독하지 못하도록 차단하는 잠금을 실행합니다. 정책의 다른 설정은 기본값을 적용합니다.

dc=example,dc=com 아래에 다음 하위 항목을 추가하여 example.com 하위 트리에 이 비밀번호 정책을 정의합니다.

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=TempPolicy,dc=example,dc=com
objectClass: top
objectClass: passwordPolicy
objectClass: LDAPsubentry
cn: TempPolicy
passwordStorageScheme: SSHA
passwordChange: on
passwordMustChange: on
passwordCheckSyntax: on
passwordExp: on
passwordMinLength: 6
passwordMaxAge: 8640000
passwordMinAge: 0
passwordWarning: 259200
passwordInHistory: 6
passwordLockout: on
passwordMaxFailure: 3
passwordUnlock: on
passwordLockoutDuration: 3600
passwordResetFailureCount: 600
```

비밀번호 정책에서 가능한 모든 속성에 대한 정의는 *Directory Server Administration Reference*를 참조하십시오.

주 passwordExpirationTime 작동 가능 속성은 이전 버전의 Directory Server와는 대조적으로 단일 값 속성입니다. 명령줄을 통해 사용자 항목을 추가할 때는 해당 작업이 거부되므로 이 속성에 값을 지정하지 말아야 합니다. 이전 버전의 Directory Server에서 만든 항목의 경우 이 속성에 여러 값을 가지므로 역시 거부됩니다.

비밀번호 정책 할당

개별 비밀번호 정책을 할당하려면 해당 정책 하위 항목을 지정해야 합니다. passwordPolicySubentry 값으로 개별 항목에 정책을 추가하거나 CoS 및 역할을 사용하여 정책을 관리합니다. 사용자가 자신에게 적용되는 비밀번호 정책을 수정할 수 없도록 차단하는 액세스 제어도 설정해야 합니다.

콘솔 사용

Directory Server 콘솔은 사용자 또는 그룹에 할당된 비밀번호 정책을 관리할 수 있는 다음과 같은 인터페이스를 제공합니다.

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 개별 비밀번호 정책을 할당하거나 수정할 사용자 또는 그룹 항목을 표시합니다.
2. 항목을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 비밀번호 정책 설정을 선택합니다. 또는 항목을 왼쪽 마우스 버튼으로 눌러 선택한 다음 객체 메뉴에서 비밀번호 정책 설정을 선택할 수도 있습니다.
3. 이 항목에 적용되는 비밀번호 정책이 비밀번호 정책 대화 상자에 표시됩니다.
 - 전역 정책이 적용되면 할당을 눌러 디렉토리 트리에서 비밀번호 정책 하위 항목을 선택합니다.
 - 개별 정책이 정의되어 있으면 이 정책을 바꾸거나 제거 또는 편집할 수도 있습니다. 정책 편집을 누르면 지정된 정책 하위 항목의 사용자 정의 편집기가 실행됩니다.

비밀번호 정책을 할당하거나 바꾸면 디렉토리 브라우저 대화 상자가 실행되며, 여기서 작은 키 아이콘이 표시된 비밀번호 정책 하위 항목을 찾을 수 있습니다.
4. 정책을 변경한 경우 비밀번호 정책 대화 상자에서 확인을 누릅니다. 새 정책이 즉시 적용됩니다.

명령줄 사용

사용자 또는 그룹 항목에 비밀번호 정책을 할당하려면 비밀번호 정책의 DN을 passwordPolicySubentry 속성 값으로 추가합니다. 예를 들어, 아래 명령은 cn=TempPolicy,dc=example,dc=com을 Barbara Jensen에게 할당합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: passwordPolicySubentry
passwordPolicySubentry: cn=TempPolicy,dc=example,dc=com
```

역할 및 CoS 사용

사용자와 역할을 그룹화하는 경우 CoS를 사용하여 해당 정책 하위 항목을 지정할 수 있습니다. 역할 및 CoS 사용에 대한 자세한 내용은 5장, "아이디 및 역할 관리"를 참조하십시오.

예를 들어, 아래 명령은 example.com의 임시 직원에 대한 필터링된 역할을 작성하고 이 역할이 지정된 직원에게 cn=TempPolicy,dc=example,dc=com을 할당합니다.

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=TempFilter,ou=people,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: TempFilter
nsRoleFilter: (&(objectclass=person)(status=contractor))
description: filtered role for temporary employees

dn: cn=PolTempl,dc=example,dc=com
objectclass: top
objectclass: nsContainer

dn: cn="cn=TempFilter,ou=people,dc=example,dc=com",
   cn=PolTempl,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: LDAPsubentry
objectclass: costemplate
cosPriority: 1
passwordPolicySubentry: cn=TempPolicy,dc=example,dc=com

dn: cn=PolCoS,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
```

```
objectclass: cosClassicDefinition
cosTemplateDN: cn=PolTempl,dc=example,dc=com
cosSpecifier: nsRole
cosAttribute: passwordPolicySubentry operational
```

이제 계약 직원의 상태가 지정된 사용자는 `cn=TempPolicy,dc=example,dc=com` 비밀번호 정책의 적용을 받습니다.

개별 비밀번호 정책 보호

사용자가 자신에게 적용되는 비밀번호 정책을 수정할 수 없도록 차단하려면 루트 항목에 다음과 같은 ACI도 추가해야 합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr != "passwordPolicySubentry")(version 3.0; acl
"Allow self modification except for passwordPolicySubentry";
allow (write) (userdn="ldap:///self");)
```

사용자 비밀번호 재설정

디렉토리는 사용자 항목의 `userPassword` 속성에 비밀번호 값을 저장합니다. 서버에 대한 액세스 제어 설정에 따라 사용자는 `ldapmodify`와 같은 표준 도구를 사용하여 `userPassword` 값을 지정한 비밀번호 정책에 맞게 설정할 수도 있습니다.

영구 계정 잠금이 발생한 경우(비밀번호 정책에서 사용자의 작동 가능 속성인 `accountUnlockTime`이 0이고 `passwordUnlock`이 `off`인 경우) 디렉토리 관리자로 비밀번호를 재설정하여 사용자 계정을 잠금 해제할 수 있습니다. 예를 들어, `example.com`의 디렉토리 사용자인 **Barbara Jensen**이 비밀번호를 잊어버려서 추측 시도한 후에 영구 잠금이 발생했다고 가정하면 아래 명령을 실행합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
replace: userPassword
userPassword: ChAnGeMe
```

비밀번호 정책에서 `passwordMustChange`가 `on`으로 설정되어 있으면 **Barbara**는 다음에 바인드할 때 자신의 비밀번호를 변경해야 합니다. 가능하면 보안 채널을 통해 비밀번호가 `ChAnGeMe`로 변경되었다고 알려주십시오.

사용자와 역할 비활성화 및 활성화

개별 사용자 계정이나 계정 집합을 일시적으로 비활성화할 수 있습니다. 비활성화된 사용자는 디렉토리에 바인드할 수 없으며 인증 작업이 실패합니다.

이 절에 설명된 절차에 따라 사용자와 역할을 동일한 방식으로 비활성화할 수 있습니다. 하지만 역할을 비활성화하는 경우 역할 자체가 아닌 이 **역할**의 구성원이 비활성화됩니다. 역할에 대한 일반적인 내용 및 역할이 특히 액세스 제어와 어떻게 상호 작용하는지에 대한 자세한 설명은 [5장, "아이디 및 역할 관리"](#)를 참조하십시오.

콘솔에서 사용자 및 역할 활성화 설정

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 디렉토리 트리를 탐색하여 비활성화하거나 다시 활성화할 사용자 또는 역할 항목을 표시합니다.
2. 항목을 두 번 눌러 해당 사용자 정의 편집기를 표시하고 왼쪽 열에서 계정 탭을 누릅니다.

오른쪽 패널에 항목의 활성화 상태가 표시됩니다.

3. 이 항목에 해당하는 사용자 또는 역할을 비활성화하거나 활성화하는 버튼을 누릅니다. 편집기의 사용자 또는 역할 아이콘에 빨간색 상자와 막대가 표시되면 해당 항목은 비활성화됩니다.
4. 확인을 눌러 대화 상자를 닫고 항목의 새 활성화 상태를 저장합니다.

사용자 정의 편집기를 쉽게 여는 방법으로는 항목을 선택하고 객체 메뉴에서 비활성화 또는 활성화를 선택할 수도 있습니다.

콘솔의 보기 > 표시 메뉴에서 비활성화 상태를 선택하여 디렉토리 객체의 활성화 상태를 표시할 수 있습니다. 이렇게 하면 모든 비활성화 항목의 아이콘에 빨간색 막대가 표시됩니다. 직접 비활성화되었든 아니면 역할 구성원을 통해 비활성화되었든 관계 없이 사용자 항목의 올바른 활성화 상태가 표시됩니다.

명령줄에서 사용자 및 역할 활성화 설정

사용자 계정 또는 역할 구성원을 비활성화하려면 `directoryserver account-inactivate` 명령을 사용합니다. 사용자 또는 역할을 활성화하거나 다시 활성화하려면 `directoryserver account-activate` 명령을 사용합니다.

```
# /usr/sbin/directoryserver account-inactivate
# /usr/sbin/directoryserver account-activate
```

아래 명령은 이 명령을 사용하여 **Barbara Jensen**의 사용자 계정을 비활성화하거나 다시 활성화하는 방법을 보여줍니다.

```
/usr/sbin/directoryserver account-inactivate -h host -p port -D
"cn=Directory\
  Manager" -w password -I "uid=bjensen,ou=People,dc=example,dc=com"
/usr/sbin/directoryserver account-activate -h host -p port -D "cn=Directory\
  Manager" -w password -I "uid=bjensen,ou=People,dc=example,dc=com"
```

두 명령에서 `-I` 옵션은 활성화 상태를 설정할 사용자 또는 역할의 DN을 지정합니다.

명령에 대한 자세한 내용은 *Directory Server Man Page Reference*를 참조하십시오.

개별 자원 제한 설정

디렉토리에 바인드하는 클라이언트 응용 프로그램의 특수 작업 속성 값을 사용하여 검색 작업에 대한 서버 제한을 제어할 수 있습니다. 다음과 같은 검색 작업 제한을 설정할 수 있습니다.

- 조회 제한은 검색 작업 시 조사할 최대 항목 수를 지정합니다.
- 크기 제한은 서버에서 검색 작업에 응답하여 클라이언트 응용 프로그램에 반환하는 최대 항목 수를 지정합니다.
- 시간 제한은 서버에서 검색 작업 처리에 사용할 수 있는 최대 시간을 지정합니다.
- 유희 시간 초과는 서버에서 연결을 중지할 때까지 클라이언트의 서버 연결을 유희 상태로 유지할 수 있는 시간을 지정합니다.

주 기본적으로 디렉토리 관리자가 사용할 수 있는 자원에는 제한이 없습니다.

특정 사용자에게 대해 설정한 자원 제한은 전역 서버 구성에서 설정한 기본 자원 제한보다 우선합니다. 사용자 항목이 포함된 접미어에 있는 아래 ACI에서 개별 자원 제한이 저장된 속성의 자체 수정을 차단하는지 확인해야 합니다.

```
(targetattr != "nsroledn || aci || nsLookThroughLimit || nsSizeLimit || nsTimeLimit || nsIdleTimeout || passwordPolicySubentry || passwordExpirationTime || passwordExpWarned || passwordRetryCount || retryCountResetTime || accountUnlockTime || passwordHistory || passwordAllowChangeTime")(version 3.0; aci "Allow self entry modification except for nsroledn, aci, resource limit attributes, passwordPolicySubentry and password policy state attributes"; allow (write)userdn ="ldap:///self";)
```

콘솔에서 자원 제한 설정

1. Directory Server 콘솔의 최상위 수준 디렉토리 탭에서 디렉토리 트리를 탐색하여 자원 제한을 설정할 사용자를 표시합니다.
2. 항목을 두 번 눌러 해당 사용자 정의 편집기를 표시하고 왼쪽 열에서 계정 탭을 누릅니다. 오른쪽 패널에 현재 항목에 설정된 제한이 표시됩니다.
3. 위에 설명된 자원 제한에 대한 네 개의 텍스트 필드에 값을 입력합니다. -1의 값은 해당 자원에 대한 제한이 없음을 나타냅니다.
4. 입력이 끝나면 확인을 눌러 새 제한을 저장합니다.

명령줄에서 자원 제한 설정

ldapmodify 명령을 사용하여 다음과 같은 사용자 항목 속성을 설정함으로써 해당 사용자의 자원 사용을 제한할 수 있습니다.

속성	설명
nsLookThroughLimit	검색 작업 시 조사할 수 있는 항목 수를 지정하므로 속성 값은 항목 수가 됩니다. 속성 값이 -1이면 제한 없음을 나타냅니다.
nsSizeLimit	서버에서 검색 작업에 응답하여 클라이언트 응용 프로그램에 반환하는 최대 항목 수를 지정합니다. 속성 값이 -1이면 제한 없음을 나타냅니다.
nsTimeLimit	서버에서 검색 작업 처리에 사용할 수 있는 최대 시간을 지정합니다. 속성 값이 -1이면 시간 제한 없음을 나타냅니다.
nsIdleTimeout	연결이 끊길 때까지 서버 연결을 유휴 상태로 유지할 수 있는 시간을 지정합니다. 값은 초 단위로 설정됩니다. 속성 값이 -1이면 제한 없음을 나타냅니다.

예를 들어, ldapmodify 명령을 다음과 같이 실행하여 항목에 크기 제한을 설정할 수 있습니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: nsSizeLimit
nsSizeLimit: 500
```

위의 ldapmodify 명령문은 nsSizeLimit 속성을 Barbara Jensen의 항목에 추가하고 검색 결과로 반환되는 최대 항목 수를 500개로 제한합니다.

복제 관리

복제란 특정 Directory Server의 디렉토리 내용이 하나 이상의 다른 디렉토리 서버로 자동 복사되는 기법입니다. 항목 추가, 수정 또는 삭제까지 모든 종류의 쓰기 작업은 자동으로 다른 Directory Server에 미러링됩니다. 복제 개념, 복제 시나리오, 디렉토리 배포 시 복제 계획 방법 등에 대한 자세한 내용은 *Directory Server Deployment Planning Guide*를 참조하십시오.

Directory Server 5.2에서는 다음과 같은 새로운 복제 기능을 제공합니다.

- WAN을 통한 여러 마스터 복제(MMR) 기능을 사용하면 멀리 떨어져 있는 마스터 간에 복제 계약을 작성하여 보다 효과적으로 데이터를 배포할 수 있습니다.
- 이제 MMR은 네 개의 완전 연결된 마스터를 동시에 지원하므로 페일오버 기능이 강화되었습니다.
- 이전 복사본을 사용하여 대규모 복제본을 훨씬 빠른 속도로 초기화할 수 있습니다.
- 단편 복제 기능을 사용하여 복제할 속성 집합을 지정함으로써 보다 효과적으로 데이터를 배포할 수 있습니다.
- 새로운 명령줄 도구는 복제 작업을 모니터링할 수 있도록 도와줍니다.

이 장에서는 다양한 복제 시나리오를 설정하는 작업에 대해 설명하며, 다음 내용으로 구성되어 있습니다.

- 소개
- 구성 복제 단계 요약
- 복제 관리자 선택
- 전용 소비자 구성
- 허브 구성
- 마스터 복제본 구성

- 복제 계약 작성
- 단편 복제 구성
- 복제본 초기화
- 참조 무결성 플러그 인 활성화
- SSL을 통한 복제
- WAN을 통한 복제
- 복제 토폴로지 수정
- 이전 릴리스를 사용한 복제
- 레트로 변경 로그 플러그 인 사용
- 복제 상태 모니터
- 일반적인 복제 충돌 해결

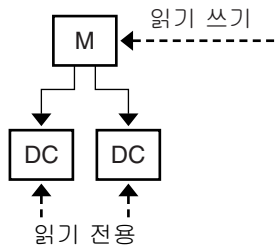
소개

복제 구성은 복잡한 작업입니다. 시작하기 전에 조직에서 어떻게 복제를 배포할지, 즉 단일 마스터, 다중 마스터 또는 허브를 사용한 계단식 복제 중에서 어떤 유형을 선택할지 등을 명백히 이해해야 합니다. 복제 단위는 접미어 또는 하위 접미어로 해당 접미어의 모든 항목이 함께 복제 됩니다. 배포 시에 각 접미어는 포함된 데이터에 대한 마스터, 허브 또는 전용 소비자 중 하나로 식별되어야 합니다.

서버에 있는 복제된 접미어를 *복제본*이라고 합니다. 마스터는 클라이언트의 읽기 및 쓰기 작업을 모두 허용하는 복제본입니다. 허브와 전용 소비자는 읽기 전용 복제본으로, 복제 기법을 통해서만 업데이트를 받습니다. 허브는 마스터나 다른 허브로부터 업데이트를 받아 다른 허브 또는 전용 소비자로 전달합니다. 전용 소비자는 마스터나 허브로부터 업데이트를 받기만 합니다.

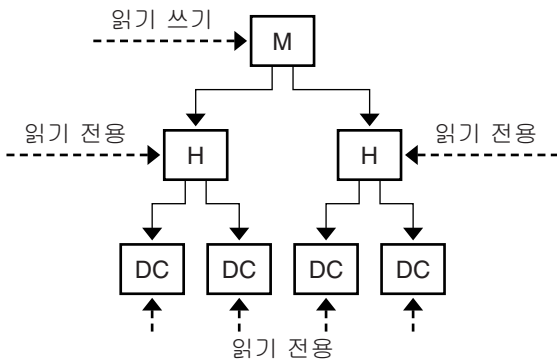
아래 세 개의 다이어그램은 일반적인 복제 시나리오에서 복제본 간의 관계를 보여줍니다.

그림 8-1 단일 마스터 복제



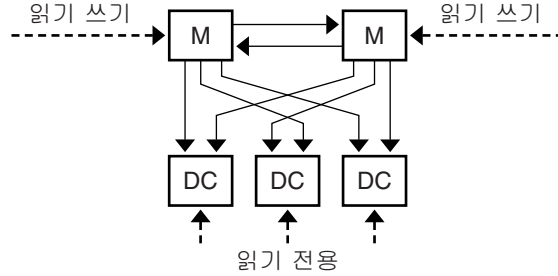
M = 마스터
 DC = 전용 소비자
 ----- 클라이언트 요청
 ————— 복제 업데이트

그림 8-2 허브를 사용한 계단식 복제



M = 마스터
 H = 허브
 DC = 전용 소비자
 ----- 클라이언트 요청
 ————— 복제 업데이트

그림 8-3 다중 마스터 복제



M = 마스터
 DC = 전용 소비자
 ----- 클라이언트 요청
 ————— 복제 업데이트

본 설명서에서는 공급자와 소비자라는 용어를 사용하여 복제 계약에 사용된 두 서버의 역할을 나타냅니다. 공급자는 복제 업데이트를 보내는 서버인 반면, 소비자는 복제 업데이트를 받는 서버입니다. 위의 다이어그램은 다음과 같은 관계를 보여줍니다.

- 단일 마스터는 소비자가 아닌 공급자입니다.
- 다중 마스터 복제 시 마스터는 공급자인 동시에 다른 마스터의 소비자입니다.
- 허브는 항상 공급자이자 소비자입니다.
- 전용 소비자는 소비자일 뿐입니다.

대부분의 복제 설정은 해당 유형에 관계 없이 계약의 공급자 또는 소비자 역할에 설정된 복제본에 적용됩니다.

구성 복제 단계 요약

아래 단계에서는 단일 접미어를 복제한다고 가정합니다. 두 개 이상의 접미어를 복제하는 경우 각 서버에 병렬로 구성할 수 있습니다. 즉, 각 단계를 반복하여 여러 개의 접미어에 대한 복제를 구성할 수 있습니다.

복제 토폴로지를 구성하려면 다음과 같은 순서로 진행해야 합니다.

1. 단일 마스터를 제외한 모든 서버에서 복제 관리자 항목을 정의하거나 모든 서버에서 기본 복제 관리자를 사용합니다.
2. 전용 소비자 복제본이 포함된 모든 서버에서 다음을 수행합니다.
 - a. 소비자 복제본에 대한 빈 접미어를 작성합니다.
 - b. 복제 마법사를 통해 이 접미어에서 소비자 복제본을 활성화합니다.
 - c. 선택 사항으로, 고급 복제본 설정을 구성합니다.
3. 필요한 경우 허브 복제본이 포함된 모든 서버에서 다음을 수행합니다.
 - a. 허브 복제본에 대한 빈 접미어를 작성합니다.
 - b. 복제 마법사를 통해 이 접미어에서 허브 복제본을 활성화합니다.
 - c. 선택 사항으로, 고급 복제본 설정을 구성합니다.
4. 마스터 복제본이 포함된 모든 서버에서 다음을 수행합니다.
 - a. 마스터 중 하나에서 마스터 복제본이 될 접미어를 선택하거나 작성합니다.
 - b. 복제 마법사를 통해 이 접미어에서 마스터 복제본을 활성화합니다.
 - c. 선택 사항으로, 고급 복제본 설정을 구성합니다.
5. 다음 순서로 모든 공급자 복제본에서 복제 계약을 구성합니다.
 - a. 다중 마스터 집합의 마스터 간
 - b. 마스터 및 전용 소비자 간
 - c. 마스터 및 허브 복제본 간

선택 사항으로, 이 단계에서 단편 복제를 구성할 수 있습니다.
6. 허브 복제본과 해당 소비자 간에 복제 계약을 구성합니다.
7. 다중 마스터 복제의 경우 데이터의 원래 복사본이 포함된 마스터 복제본을 사용하여 모든 마스터를 초기화합니다. 허브와 소비자 복제본을 초기화합니다.

주 복제 계약을 작성하기 전에 모든 복제본을 활성화하는 것이 중요합니다. 이렇게 함으로써 복제 계약을 작성한 후에 소비자 복제본을 즉시 초기화할 수 있습니다. 소비자 초기화는 항상 복제 설정의 마지막 단계에 수행해야 합니다.

복제 관리자 선택

복제 설정의 핵심 부분은 공급자가 복제 업데이트를 보내기 위해 소비자 서버에 바인드할 때 사용하는 항목인 *복제 관리자*를 선택하는 것입니다. 업데이트를 받는 접미어가 포함된 모든 서버에 복제 관리자 항목이 적어도 한 개는 있어야 합니다.

Directory Server에는 모든 서버에 사용할 수 있는 기본 복제 관리자가 있습니다. 해당 DN은 `cn=Replication Manager,cn=replication,cn=config`입니다.

단순한 복제 시나리오에서는 기본 복제 관리자를 사용하는 것이 좋습니다. 복제 마법사는 자동으로 이 항목을 사용하여 소비자 복제본을 구성함으로써 복제본의 배포를 간소화합니다.

복제 마법사는 기본 복제 관리자의 비밀번호가 정의되어 있지 않으면 비밀번호를 설정하라는 메시지를 표시합니다. 나중에 비밀번호를 변경하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드를 선택한 다음 오른쪽 패널에서 복제 탭을 선택합니다.
2. 복제 관리자 머리글 아래의 두 텍스트 필드에 새 비밀번호를 입력합니다.
3. 비밀번호 확인이 끝나면 저장을 누릅니다. 입력한 두 비밀번호가 일치하지 않으면 저장 버튼이 활성화되지 않습니다.

기본 복제 관리자를 사용하지 않을 경우, 복제 관리자 역할을 수행할 새 항목을 작성할 수 있습니다. 예를 들어, 여러 개의 복제 관리자 항목이 복제된 각 접미어에 대해 다른 비밀번호를 사용하도록 설정할 수 있습니다. 새로운 복제 관리자를 작성하는 다른 이유는 SSL을 통한 인증서 사용과 같이 다른 복제 인증 모델을 지원하기 위한 것입니다.

복제 관리자 항목에는 복제 계약을 정의할 때 선택한 인증 방법에 필요한 속성이 포함되어 있어야 합니다. 예를 들어, 기본 복제 관리자는 단순한 인증을 위한 `userPassword` 속성을 허용하는 `person` 객체 클래스입니다. 인증서를 사용하여 복제 관리자를 바인드하는 방법은 [363페이지의 "SSL을 통한 복제"](#)를 참조하십시오.

복제 관리자 항목은 소비자 서버의 복제된 접미어에 저장할 수 없습니다. 복제 관리자는 `cn=replication,cn=config`에 정의하는 것이 바람직합니다.

명령줄에서 수동으로 새 복제 관리자를 작성하려면 복제본 구성 항목의 `nsDS5ReplicaBindDN` 속성을 수정하여 소비자에서 바인드 DN을 지정해야 합니다.

레거시 복제를 사용하는 경우 복제 관리자 항목에 대한 제약 조건이 추가됩니다. 자세한 내용은 [378페이지의 "Directory Server 5.2를 Directory Server 4.x의 소비자로 구성하려면"](#)을 참조하십시오.

주의	복제 관리자 항목의 DN과 비밀번호를 사용하여 서버에 대한 작업을 수행하거나 바인드해서는 안 됩니다. 복제 관리자는 복제 기법에서만 사용되며 다른 용도로 사용하려면 복제본을 다시 초기화해야 합니다.
	디렉토리 관리자를 복제 관리자로 사용해서는 안 됩니다.

각 소비자의 복제 관리자를 선택한 후 다음을 수행합니다.

1. 선택 또는 작성한 복제 관리자 DN을 기록해 두거나 기억합니다. 나중에 해당 공급자에서 소비자와의 복제 계약을 작성할 때 이 DN과 비밀번호를 사용해야 합니다.
2. 비밀번호 만료 정책을 정의한 경우 복제 관리자를 제외해야 하며, 그렇지 않으면 비밀번호 만료 시 제대로 복제되지 않습니다. 비밀번호 관리자 항목에 대한 비밀번호 만료를 비활성화하려면 비밀번호가 만료되지 않는 비밀번호 정책을 작성한 다음 복제 관리자 항목에 할당합니다. 자세한 내용은 [319페이지의 "개별 비밀번호 정책 관리"](#)를 참조하십시오.

전용 소비자 구성

전용 소비자는 복제된 접미어의 읽기 전용 복사본으로, 복제 관리자로 바인드하는 서버로부터 업데이트를 받아 항목을 변경합니다. 소비자 서버 구성은 복제본을 저장할 빈 접미어를 준비하는 단계와 복제 마법사를 사용하여 이 접미어에서 복제를 활성화하는 단계로 이루어져 있습니다. 선택 사항인 고급 옵션에는 다른 복제 관리자 선택, 참조 설정, 지연 제거 설정 등이 있습니다.

다음 절에서는 서버에 한 개의 전용 소비자 복제본을 구성하는 단계에 대해 설명합니다. 특정 접미어의 전용 소비자 복제본이 포함될 각 서버에서 모든 절차를 반복합니다.

소비자 복제본에 대한 접미어 작성

원하는 마스터 복제본과 동일한 DN을 가진 접미어가 없으면 이 DN을 사용하여 소비자에 대한 빈 접미어를 작성합니다. 자세한 내용은 141페이지의 "접미어 작성"을 참조하십시오.

내용이 포함된 접미어가 있으면 마스터를 사용하여 복제본을 초기화할 때 이 내용이 손실됩니다.

소비자 복제본 활성화

복제 마법사를 사용하면 간편하게 전용 소비자 복제본을 활성화할 수 있습니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 소비자 복제본으로 사용할 접미어 및 데이터 노드를 모두 확장한 다음 접미어 아래의 복제 노드를 선택합니다.
오른쪽 패널에 복제본 상태 정보가 표시됩니다.
2. 복제 활성화 버튼을 눌러 복제 마법사를 시작합니다.
3. 기본적으로 소비자 복제본 라디오 버튼이 선택됩니다. 다음을 눌러 계속합니다.
4. 기본 복제 관리자의 비밀번호를 입력하지 않은 경우 비밀번호 및 비밀번호 확인 필드에 원하는 비밀번호를 입력하라는 메시지가 표시됩니다. 두 필드에 같은 비밀번호를 입력하고 다음을 눌러 계속합니다.
기본 복제 관리자의 비밀번호가 정의되어 있으면 마법사는 자동으로 이 단계를 건너뛵니다.
5. 복제 마법사는 복제 구성을 업데이트하는 동안 상태 메시지를 표시합니다. 업데이트가 끝나면 단기를 누릅니다.

이제 복제본에서 업데이트를 받을 수 있다는 복제 상태가 표시되고 왼쪽 패널의 아이콘이 이에 따라 바뀝니다.

고급 소비자 구성

기본적으로 마법사는 복제본에서 기본 복제 관리자를 사용하도록 구성합니다. 다른 복제 관리자 항목을 사용하려면 고급 구성을 설정해야 합니다. 이 대화 상자를 사용하여 수정할 참조와 지연 제거를 설정할 수도 있습니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드와 구성할 접미어 노드를 모두 확장한 다음 접미어 아래의 복제 노드를 선택합니다.
2. 오른쪽 패널에서 고급 버튼을 눌러 고급 복제본 설정 대화 상자를 표시합니다.
3. 바인드 DN 탭에서 추가 및 삭제 버튼을 사용하여 유효한 복제 관리자 DN 목록을 작성합니다. 이렇게 하면 공급자가 이 복제본과의 계약에서 목록에 있는 DN 중 하나를 사용할 수 있습니다. 이름을 입력하거나 디렉토리를 탐색하여 새 DN을 추가할 수도 있습니다.

인증서를 사용하여 SSL을 통한 복제를 구성하려면 인증서 항목의 DN을 복제 관리자 중 하나로 입력합니다.

4. 입력이 끝나면 확인을 누르거나, 선택 사항 탭을 선택하여 추가 고급 구성을 설정합니다.
5. 고급 복제본 설정 대화 상자의 선택 사항 탭에 있는 LDAP URL 목록은 이 소비자로 보내진 수정 요청에 대한 추가 참조를 지정합니다. 추가 또는 삭제 버튼을 사용하여 LDAP URL 목록을 작성합니다.

복제 기법은 자동으로 소비자에서 복제 토폴로지의 알려진 모든 마스터에 대한 참조를 반환하도록 구성합니다. 이러한 기본 참조는 클라이언트에서 일반 연결을 통한 단순한 인증을 사용한다고 가정합니다. 보안 연결을 위해 클라이언트에서 SSL을 사용하여 마스터에 바인드할 수 있게 하려면 보안 포트번호를 사용하는 `ldaps://servername:port` 형식의 참조를 추가하십시오. 마스터가 보안 연결에 대해서만 구성되어 있으면 URL이 기본적으로 보안 포트를 가리킵니다.

하나 이상의 LDAP URL을 참조로 추가한 경우 목록 아래의 확인란을 선택하면 소비자는 마스터 복제본이 아닌 LDAP URL에 대한 참조만 보냅니다. 예를 들어, 클라이언트에서 기본 포트가 아닌 마스터 서버에 있는 보안 포트를 항상 참조하게 하려면 보안 포트에 대한 LDAP URL 목록을 작성한 다음 이 확인란을 선택합니다. 특정 마스터 또는 Directory Server 프로시에서 모든 업데이트를 처리하도록 지정하려면 배타적 참조를 사용할 수도 있습니다.

- 또한 선택 사항 탭에서 지연 제거를 변경할 수 있습니다.

소비자 서버는 복제본 내용 업데이트에 대한 내부 정보를 저장하며, 지연 제거 매개 변수는 이 정보의 보관 기간을 지정합니다. 이 매개 변수는 공급자 서버에 있는 변경 로그의 MaxAge 매개 변수와 관련이 있습니다. 두 매개 변수 중에서 작은 값이 두 서버 간의 복제를 비활성화하거나 중단했다가 다시 정상적으로 복구할 수 있는 최대 기간을 지정합니다. 대부분의 경우 이 기간은 기본값 7일이면 충분합니다.

- 확인을 눌러 복제본에 대한 고급 복제 구성을 저장합니다.

허브 구성

허브 복제본은 소비자 및 마스터로서의 기능을 동시에 수행함으로써 복제된 데이터를 다수의 소비자로 배포하며, 공급자로부터 복제 업데이트를 받아 소비자에게 보냅니다. 허브 복제본은 수정을 승인하지 않고 마스터에 대한 참조를 반환합니다.

허브 서버 구성은 복제본을 저장할 빈 접미어를 준비하는 단계와 복제 마법사를 사용하여 이 접미어에서 복제를 활성화하는 단계로 이루어져 있습니다. 선택 사항인 고급 옵션에는 다른 복제 관리자 선택, 참조 설정, 지연 제거 설정, 변경 로그 매개 변수 설정 등이 있습니다.

다음 절에서는 한 개의 허브 서버를 구성하는 단계에 대해 설명합니다. 특정 접미어의 허브 복제본이 포함될 각 서버에서 모든 절차를 반복합니다.

허브 복제본에 대한 접미어 작성

원하는 마스터 복제본과 동일한 DN을 가진 접미어가 없으면 이 DN을 사용하여 허브 서버에 대한 빈 접미어를 작성합니다. 자세한 내용은 [141페이지의 "접미어 작성"](#)을 참조하십시오.

내용이 포함된 접미어가 있으면 마스터를 사용하여 복제본을 초기화할 때 이 내용이 손실됩니다.

허브 복제본 활성화

복제 마법사를 사용하면 간편하게 허브 복제본을 활성화할 수 있습니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 허브 복제본으로 사용할 접미어 및 데이터 노드를 모두 확장한 다음 접미어 아래의 복제 노드를 선택합니다.

오른쪽 패널에 복제본 상태 정보가 표시됩니다.

2. 복제 활성화 버튼을 눌러 복제 마법사를 시작합니다.
3. 허브 복제본 라디오 버튼을 선택하고 다음을 눌러 계속합니다.
4. 변경 로그 파일을 선택하지 않은 경우 파일을 선택하라는 메시지가 표시됩니다. 텍스트 필드에 기본 변경 로그 파일이 표시됩니다. 기본값을 사용하지 않으려면 변경 로그 파일 이름을 입력하거나 찾아보기를 눌러 파일 선택기를 표시합니다.

변경 로그가 활성화되어 있으면 마법사는 이 단계를 건너뜁니다.

5. 다음을 누릅니다. 기본 복제 관리자의 비밀번호를 입력하지 않은 경우 비밀번호 및 비밀번호 확인 필드에 원하는 비밀번호를 입력하라는 메시지가 표시됩니다. 두 필드에 같은 비밀번호를 입력하고 다음을 눌러 계속합니다.

기본 복제 관리자의 비밀번호가 정의되어 있으면 마법사는 자동으로 이 단계를 건너뜁니다.

6. 복제 마법사는 복제 구성을 업데이트하는 동안 상태 메시지를 표시합니다. 업데이트가 끝나면 단기를 누릅니다.

이제 복제본에서 업데이트를 받을 수 있다는 복제 상태가 표시되고 왼쪽 패널의 아이콘이 이에 따라 바뀝니다.

고급 허브 구성

공급자로서의 허브 서버에는 변경 로그가 필요하며, 마법사는 허브 복제본에서 기본 변경 로그 설정을 사용하도록 구성합니다. 이 설정을 수정하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드를 선택한 다음 오른쪽 패널에서 복제 탭을 선택합니다.
2. 변경 로그 사용 확인란을 선택하고 재설정 버튼을 눌러 이 탭의 내용을 갱신해야 할 수도 있습니다. 이렇게 하면 복제 마법사에서 선택한 변경 로그 파일이 표시됩니다.
3. 변경 로그 파일의 이름을 변경하고 다음과 같은 변경 로그 매개 변수를 업데이트할 수 있습니다.

- a. 최대 변경 로그 레코드 수 - 소비자에게 업데이트를 보내기 위해 저장할 수정 사항의 총 수를 지정합니다. 기본적으로 이 값은 제한되지 않습니다. 레코드 수를 제한하면 복제본에서 여러 개의 큰 수정 사항을 받을 경우 디스크 공간이 절약됩니다.
- b. 변경 로그의 최대 수명 - 허브에서 소비자로 보내야 하는 업데이트를 저장하는 시간을 지정합니다. 기본적으로 이 값은 제한되지 않습니다. 최대 수명 매개 변수는 변경 로그의 크기를 제한할 수 있는 좋은 방법입니다.

변경 로그 구성 속성에 대한 자세한 내용은 *Directory Server Administration Reference*에 설명되어 있습니다.

또한 복제 마법사는 기본 복제 관리자를 사용합니다. 사용할 다른 복제 관리자 항목을 작성한 경우에는 고급 구성을 설정해야 합니다. 이 대화 상자를 사용하여 수정할 참조와 지연 제거를 설정할 수도 있습니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드와 구성할 접미어 노드를 모두 확장한 다음 접미어 아래의 복제 노드를 선택합니다.
2. 오른쪽 패널에서 고급 버튼을 눌러 고급 복제본 설정 대화 상자를 표시합니다.
3. 바인드 DN 탭에서 추가 및 삭제 버튼을 사용하여 유효한 복제 관리자 DN 목록을 작성합니다. 이렇게 하면 공급자가 이 복제본과의 계약에서 목록에 있는 DN 중 하나를 사용할 수 있습니다. 이름을 입력하거나 디렉토리를 탐색하여 새 DN을 추가할 수도 있습니다.
인증서를 사용하여 SSL을 통한 복제를 구성하려면 인증서 항목의 DN을 복제 관리자 중 하나로 입력합니다.
4. 입력이 끝나면 확인을 누르거나, 선택 사항 탭을 선택하여 추가 고급 구성을 설정합니다.

5. 고급 복제본 설정 대화 상자의 선택 사항 탭에 있는 LDAP URL 목록은 이 허브로 보내진 수정 요청에 대한 추가 참조를 지정합니다. 추가 또는 삭제 버튼을 사용하여 LDAP URL 목록을 작성합니다.

복제 기법은 자동으로 허브에서 복제 토폴로지의 알려진 모든 마스터에 대한 참조를 반환하도록 구성합니다. 이러한 기본 참조는 클라이언트에서 일반 연결을 통한 단순한 인증을 사용한다고 가정합니다. 보안 연결을 위해 클라이언트에서 SSL을 사용하여 마스터에 바인드할 수 있게 하려면 보안 포트 번호를 사용하는 `ldaps://servername:port` 형식의 참조를 추가하십시오.

하나 이상의 LDAP URL을 참조로 추가한 경우 목록 아래의 확인란을 선택하면 서버는 마스터 복제본이 아닌 LDAP URL에 대한 참조만 보냅니다. 예를 들어, 클라이언트에서 기본 포트가 아닌 마스터 서버에 있는 보안 포트를 항상 참조하게 하려면 보안 포트에 대한 LDAP URL 목록을 작성한 다음 이 확인란을 선택합니다. 특정 마스터 또는 Directory Server 프록시에서 모든 업데이트를 처리하도록 지정하려면 배타적 참조를 사용할 수도 있습니다.

6. 또한 선택 사항 탭에서 지연 제거를 변경할 수 있습니다.

허브 서버는 복제본 내용 업데이트에 대한 내부 정보를 저장하며, 지연 제거 매개 변수는 이 정보의 보관 기간을 지정합니다. 이 매개 변수는 자체 변경 로그가 아닌 업데이트를 공급하는 서버에 있는 변경 로그의 MaxAge 매개 변수와 관련이 있습니다. 두 매개 변수 중에서 작은 값이 두 서버 간의 복제를 비활성화하거나 중단했다가 다시 정상적으로 복구할 수 있는 최대 기간을 지정합니다. 대부분의 경우 이 기간은 기본 값 7일이면 충분합니다.

7. 확인을 눌러 복제본에 대한 고급 복제 구성을 저장합니다.

마스터 복제본 구성

마스터 복제본에는 데이터의 마스터 복사본이 포함되어 있으며 업데이트를 다른 모든 복제본으로 전파하기 전에 모든 수정 사항을 중앙 집중식으로 관리합니다. 마스터는 모든 변경 사항을 기록하고, 소비자 상태를 확인하며, 필요한 경우 소비자에게 업데이트를 보냅니다. 다중 마스터 복제 시 마스터 복제본은 다른 마스터로부터 업데이트를 받기도 합니다.

마스터 서버 구성은 마스터 복제본이 포함된 접미어 정의, 복제 마법사를 사용하여 마스터 복제본 활성화, 필요한 경우 고급 복제를 위한 구성 등의 단계로 이루어져 있습니다.

다음 절에서는 한 개의 마스터 서버를 구성하는 단계에 대해 설명합니다. 특정 접미어의 마스터 복제본이 포함될 각 서버에서 모든 절차를 반복합니다.

마스터 복제본에 대한 접미어 정의

마스터 서버에서 복제할 항목이 포함될 접미어를 선택하거나 새로 작성합니다. 자세한 내용은 [141페이지의 "접미어 작성"](#)을 참조하십시오.

복제 계약을 작성하기 전의 모든 초기 데이터가 접미어에 포함되어 있어야 합니다. 이렇게 하면 이 데이터를 사용하여 즉시 소비자 복제본을 초기화할 수 있습니다. 올바른 다중 마스터를 구성 및 초기화하려면 마스터 중 하나에만 모든 초기 데이터가 포함되고 다른 마스터의 접미어는 비어 있어야 합니다.

마스터 복제본 활성화

복제 마법사를 사용하면 간편하게 마스터 복제본을 활성화할 수 있습니다.

1. **Directory Server** 콘솔의 최상위 구성 탭에서 마스터 복제본으로 사용할 접미어 및 데이터 노드를 모두 확장한 다음 접미어 아래의 복제 노드를 선택합니다.

오른쪽 패널에 복제본 상태 정보가 표시됩니다.

2. 복제 활성화 버튼을 눌러 복제 마법사를 시작합니다.
3. 마스터 복제본 라디오 버튼을 선택하고 다음을 눌러 계속합니다.
4. 복제본 아이디를 입력합니다. 1에서 65534까지의 고유한 정수를 선택하십시오.

복제본 아이디는 개별 접미어에 대한 모든 마스터 복제본에서 고유해야 합니다. 각 복제본의 다른 마스터에서만 고유하면 동일한 서버에 있는 다른 접미어의 마스터 복제본은 같은 복제본 아이디를 사용할 수 있습니다.

5. 다음을 누릅니다. 변경 로그 파일을 선택하지 않은 경우 파일을 선택하라는 메시지가 표시됩니다. 텍스트 필드에 기본 변경 로그 파일이 표시됩니다. 기본값을 사용하지 않으려면 변경 로그 파일 이름을 입력하거나 찾아보기를 눌러 파일 선택기를 표시합니다.

변경 로그가 활성화되어 있으면 마법사는 이 단계를 건너뛵니다.

6. 다음을 누릅니다. 기본 복제 관리자의 비밀번호를 입력하지 않은 경우 비밀번호 및 비밀번호 확인 필드에 원하는 비밀번호를 입력하라는 메시지가 표시됩니다. 단일 마스터 복제 시에는 복제 관리자가 사용되지 않지만 이 경우에도 비밀번호를 입력해야만 계속할 수 있습니다. 두 필드에 같은 비밀번호를 입력하고 다음을 눌러 계속합니다.

기본 복제 관리자의 비밀번호가 정의되어 있으면 마법사는 자동으로 이 단계를 건너뛵니다.

7. 복제 마법사는 복제 구성을 업데이트하는 동안 상태 메시지를 표시합니다. 업데이트가 끝나면 단기를 누릅니다.

이제 복제 상태에 이 마스터의 복제본 아이디가 표시되고 왼쪽 패널의 아이콘이 바뀌어 이 접미어에 대한 복제가 활성화되어 있음을 나타냅니다.

고급 다중 마스터 구성

기본적으로 마법사는 마스터 복제본에서 기본 변경 로그 설정을 사용하도록 구성합니다. 변경 로그 설정을 수정하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드를 선택한 다음 오른쪽 패널에서 복제 탭을 선택합니다.
2. 변경 로그 사용 확인란을 선택하고 재설정 버튼을 눌러 이 탭의 내용을 갱신해야 할 수도 있습니다. 이렇게 하면 복제 마법사에서 선택한 변경 로그 파일이 표시됩니다.
3. 변경 로그 파일의 이름을 변경하고 다음과 같은 변경 로그 매개 변수를 업데이트할 수 있습니다.
 - a. 최대 변경 로그 레코드 수 - 소비자에게 업데이트를 보내기 위해 저장할 수정 사항의 총 수를 지정합니다. 기본적으로 이 값은 제한되지 않습니다. 레코드 수를 제한하면 복제본에서 여러 개의 큰 수정 사항을 받을 경우 디스크 공간이 절약됩니다.
 - b. 변경 로그의 최대 수명 - 마스터에서 소비자로 보내야 하는 업데이트를 저장하는 시간을 지정합니다. 기본적으로 이 값은 제한되지 않습니다. 최대 수명 매개 변수는 변경 로그의 크기를 제한할 수 있는 좋은 방법입니다.

변경 로그 구성 속성의 자세한 내용은 *Directory Server Administration Reference*에 설명되어 있습니다.

또한 복제 마법사는 기본 복제 관리자를 사용합니다. 사용할 다른 복제 관리자 항목을 작성한 경우에는 고급 구성을 설정해야 합니다. 이 대화 상자를 사용하여 수정할 참조와 지연 제거를 설정할 수도 있습니다. 단일 마스터를 구성하는 경우에는 이 절차를 건너뛰어도 됩니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드와 구성할 접미어 노드를 모두 확장한 다음 접미어 아래의 복제 노드를 선택합니다.
2. 오른쪽 패널에서 고급 버튼을 눌러 고급 복제본 설정 대화 상자를 표시합니다.

3. 바인드 DN 탭에서 추가 및 삭제 버튼을 사용하여 유효한 복제 관리자 DN 목록을 작성합니다. 이렇게 하면 공급자가 이 복제본과의 계약에서 목록에 있는 DN 중 하나를 사용할 수 있습니다. 이름을 입력하거나 디렉토리를 탐색하여 새 DN을 추가할 수도 있습니다.

인증서를 사용하여 SSL을 통한 복제를 구성하려면 인증서 항목의 DN을 복제 관리자 중 하나로 입력합니다.

4. 입력이 끝나면 확인을 누르거나, 선택 사항 탭을 선택하여 추가 고급 구성을 설정합니다.
5. 고급 복제본 설정 대화 상자의 선택 사항 탭에 있는 LDAP URL 목록은 이 마스터로 보내진 수정 요청에 대한 추가 참조를 지정합니다. 354페이지의 "다중 마스터 초기화 후의 수립"에 설명된 것처럼 마스터를 초기화하면 자동으로 즉시 참조가 반환됩니다. 추가 또는 삭제 버튼을 사용하여 LDAP URL 목록을 작성합니다.

복제 기법은 자동으로 허브에서 복제 토폴로지의 알려진 모든 마스터에 대한 참조를 반환하도록 구성합니다. 이러한 기본 참조는 클라이언트에서 일반 연결을 통한 단순한 인증을 사용한다고 가정합니다. 보안 연결을 위해 클라이언트에서 SSL을 사용하여 마스터에 바인드할 수 있게 하려면 보안 포트 번호를 사용하는 `ldaps://servername:port` 형식의 참조를 추가하십시오.

하나 이상의 LDAP URL을 참조로 추가한 경우 목록 아래의 확인란을 선택하면 서버는 마스터 복제본이 아닌 LDAP URL에 대한 참조만 보냅니다. 예를 들어, 클라이언트에서 기본 포트가 아닌 마스터 서버에 있는 보안 포트를 항상 참조하게 하려면 보안 포트에 대한 LDAP URL 목록을 작성한 다음 이 확인란을 선택합니다.

6. 또한 선택 사항 탭에서 지연 제거를 변경할 수 있습니다.

마스터 서버는 복제본 내용 업데이트에 대한 내부 정보를 저장해야 하며, 지연 제거 매개 변수는 이 정보의 보관 기간을 지정합니다. 이 매개 변수는 자체 변경 로그가 아닌 업데이트를 공급하는 마스터 서버에 있는 변경 로그의 MaxAge 매개 변수와 관련이 있습니다. 두 매개 변수 중에서 작은 값이 두 서버 간의 복제를 비활성화하거나 중단했다가 다시 정상적으로 복구할 수 있는 최대 기간을 지정합니다. 대부분의 경우 이 기간은 기본값 7일이면 충분합니다.

7. 확인을 눌러 복제본에 대한 고급 복제 구성을 저장합니다.

복제 계약 작성

복제 계약은 특정 소비자에게 업데이트를 보내는 방법을 구성 및 제어하는 공급자의 매개 변수 집합입니다. 복제 계약은 소비자에게 업데이트를 보내는 공급자 복제본에서 업데이트할 모든 소비자에 대해 작성해야 합니다.

다음과 같은 순서로 복제 계약을 작성합니다.

1. 다중 마스터 집합의 마스터 간. 복제할 접미어의 원래 복사본이 포함된 마스터부터 시작합니다.
2. 허브를 통해 복제되지 않은 전용 소비자와 마스터 간
3. 마스터 및 허브 복제본 간
4. 허브 복제본 및 해당 소비자 간

예를 들어, [334페이지의 그림 8-3](#)에 표시된 것처럼 마스터 두 개와 전용 소비자 세 개로 구성된 다중 마스터 복제 토폴로지에서는 다음과 같은 순서로 8개의 복제 계약을 작성하게 됩니다.

- 한 개의 마스터와 다른 마스터 간
- 다른 마스터와 첫 번째 마스터 간
- 한 개의 마스터와 각 전용 소비자 세 개 간
- 다른 마스터와 각 전용 소비자 세 개 간

복제 계약을 작성하려면 다음을 수행합니다.

1. **Directory Server** 콘솔의 최상위 구성 탭에서 데이터 노드와 공급자 접미어 노드를 모두 확장한 다음 접미어 아래의 복제 노드를 선택합니다.
오른쪽 패널에 복제본 상태 정보가 표시됩니다.
2. 정의된 복제 계약 목록 옆에 있는 새로 만들기 버튼을 누릅니다.
3. 복제 계약 대화 상자의 메뉴에서 소비자 복제본이 포함된 기존 서버를 선택하거나 기타 버튼을 눌러 다른 서버를 정의합니다.

기타 버튼을 누른 경우 소비자 서버의 전체 도메인 이름과 LDAP 포트 번호를 입력합니다. 복제 업데이트에 대한 보안 연결을 활성화하기 위해 이 포트에서 SSL을 사용하는 경우 보안 포트 확인란을 선택합니다.

4. 소비자 서버에 있는 복제 관리자 항목의 DN과 비밀번호를 입력합니다. 기본적으로 이 DN은 기본 복제 관리자의 DN입니다.

보안 포트가 있는 소비자를 선택한 경우 옵션 버튼을 눌러 DN 필드의 의미를 지정할 수 있습니다. 비밀번호를 사용하여 연결하면 공급자는 암호화된 SSL 연결을 통한 단순 인증 및 통신을 사용합니다. 인증서를 사용하여 연결하면 DN 필드에는 인증서가 포함된 항목의 DN이 지정되며 비밀번호를 입력할 필요가 없습니다.

5. 선택 사항으로, 이 계약에 대한 설명 문자열을 입력합니다. 이 마스터 복제본에 대한 복제 계약 목록에 소비자 서버 이름과 포트 번호, 그리고 설명 문자열이 표시됩니다.
6. 작업이 끝나면 확인을 누릅니다. 방금 입력한 연결 매개 변수를 테스트할 것인지 확인하는 대화 상자가 표시됩니다.
7. 지정된 복제 관리자 및 비밀번호를 사용하여 특정 서버와 포트 번호에 대한 연결을 테스트하려면 예를 누릅니다. 매개 변수를 올바르게 지정했지만 서버가 오프라인 상태인 경우에는 연결이 실패해도 이 계약을 사용할 수 있습니다.

작업이 끝나면 마스터 복제본에 대한 복제 계약 목록에 이 계약이 표시됩니다.

나중에 다음과 같이 복제 계약을 편집하여 소비자 서버에 있는 복제 관리자의 DN과 비밀번호를 변경할 수 있습니다.

1. 목록에서 복제 계약을 선택하고 편집 버튼을 누릅니다.
2. 복제 계약 대화 상자에서 연결 탭을 선택합니다.
3. 소비자 서버의 복제 관리자 DN 또는 비밀번호를 편집합니다.
4. 선택 사항으로, 계약에 대한 설명 문자열을 편집합니다.
5. 확인을 눌러 새 설정을 저장하면 이 소비자에게 업데이트를 보낼 때 즉시 새 설정이 사용됩니다.

다른 탭의 구성 매개 변수에 대해서는 [352페이지](#)의 "단편 복제 활성화" 및 [364페이지](#)의 "WAN을 통한 복제"에서 설명합니다.

6. 각 복제 계약을 작성한 후 선택 사항으로 이 접미어에 대한 단편 복제를 구성한 다음, 352페이지의 "복제본 초기화"에 설명된 것처럼 즉시 복제본을 초기화할 수 있습니다.

주 복제가 이미 작동한 후 마스터의 포트 번호를 바꾸는 경우에는 서버를 다시 초기화할 필요가 없습니다. 그러나 이전 주소(host:oldport)를 가리키는 이전 복제 계약은 더 이상 사용될 수 없으므로, 포트 변경 전과 같이 복제를 계속하려면 새 주소(host:newport)가 있는 새 계약으로 대체해야 합니다.

단편 복제 구성

기본적으로 복제는 복제된 접미어의 전체 항목을 소비자 복제본에 복사합니다. 단편 복제 기능을 사용하면 복제 중에 복제하거나 제외할 속성의 부분 집합을 지정할 수 있습니다. 단편 복제는 복제 계약에 구성되므로 마스터의 각 소비자 복제본에 대한 속성 집합을 정의할 수 있으며, 배포할 데이터를 제어하고 복제 대역폭과 소비자 자원을 보다 효율적으로 사용할 수 있습니다.

예를 들어, 복제 대역폭을 줄이려면 photo, jpegPhoto, audio와 같이 일반적으로 큰 값을 갖는 속성을 복제하지 않도록 선택할 수 있습니다. 이 경우 소비자에서는 이러한 속성을 사용할 수 없습니다. 다른 예로, uid 및 userpassword 속성만 인증 전용 소비자 서버로 복제하도록 선택할 수도 있습니다.

단편 복제 시 고려 사항

속성의 단편 집합을 활성화하거나 수정하려면 소비자 복제본을 다시 초기화해야 하므로 배포 전에 단편 복제 요구를 결정하여 처음 복제본을 초기화하기 전에 속성 집합을 정의해야 합니다.

특정 속성에 대한 ACI, 역할, CoS 등 복잡한 기능의 종속성을 감안하여 소규모 속성 집합을 복제할 때는 특히 주의해야 합니다. 또한 ACI, 역할 또는 CoS 기법의 지정자나 필터에 명시된 다른 속성을 복제하지 않으면 데이터 보안이 손상되거나 검색 시 다른 속성 집합이 반환될 수 있습니다. 제외할 속성 목록을 관리하는 것이 포함할 속성 목록을 관리하는 것보다 안전하고 실수할 위험이 적습니다.

복제할 속성 집합에서 복제된 항목의 일부만 스키마를 따르도록 허용하는 경우에는 소비자 서버에서 스키마 검사를 비활성화해야 합니다. 복제 기법에서 소비자에 대한 스키마 검사를 생략하기 때문에 비준수 항목을 복제해도 오류가 발생하지는 않지만 소비자에 비준수 항목이 포함되므로 클라이언트에 일관된 상태를 표시하려면 스키마 검사를 비활성화해야 합니다.

단편 복제는 허브 및 전용 소비자와 마스터 복제본 간의 복제 계약에 구성됩니다. 다중 마스터 복제 환경에서 두 마스터 복제본 간의 단편 복제 구성은 지원되지 않습니다. 또한, 여러 개의 마스터가 동일한 복제본과의 복제 계약을 구성하는 경우 모든 계약이 동일한 속성 집합을 복제해야 합니다.

Directory Server 5.2에서 제공하는 단편 복제 기능은 Directory Server의 이전 버전과 호환되지 *않습니다*. 단편 복제 계약을 구성하려면 마스터와 소비자 복제본이 모두 Directory Server 인스턴스 버전 5.2에 있어야 합니다.

속성 집합 정의

속성 집합은 복제본에 대한 단편 복제를 활성화할 때 다른 모든 속성을 제외하고 복제되는 속성 목록입니다. 마스터 서버에서 원하는 대로 속성 집합을 정의한 다음 각 속성 집합을 복제 계약과 연결할 수 있습니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드를 선택한 다음 오른쪽 패널에서 복제 탭을 선택합니다.
2. 복제 탭의 아래쪽에 있는 복제된 속성 집합 관리 버튼을 누릅니다. 이 버튼을 표시하기 위해 아래로 스크롤해야 할 수도 있습니다.

3. 추가를 눌러 새 속성 집합을 정의하거나 목록에서 기존 속성 집합을 선택하고 편집을 눌러 수정합니다. 표시된 속성 집합 대화 상자에서 복제 열의 확인란을 선택하거나 선택 취소하여 집합에서 해당 속성을 제외하거나 포함합니다. 속성 이름 옆의 확인란이 선택되어 있으면 해당 속성이 복제된다는 것을 나타냅니다.

기본적으로 모든 속성이 선택되므로 특별히 복제하지 않으려는 속성만 선택 취소하는 것이 좋습니다. 처음부터 다시 선택 항목을 지정하려면 모두 선택 버튼을 눌러 모든 속성을 다시 선택합니다. 다수의 속성을 선택 취소하면 디렉토리 서버는 선택 취소된 속성을 *제외한 모든 속성*을 복제합니다. 나중에 새 속성을 스키마에 정의하여 복제된 항목에서 사용하면 속성 집합을 편집하여 해당 속성을 선택 취소하지 않을 경우 이러한 새 속성도 복제됩니다.

선택 안 함 버튼을 눌러 모든 속성을 선택 취소한 다음 속성 집합에 포함할 속성을 선택할 수도 있습니다. 선택 안 함 버튼을 누른 후에 원하는 속성 집합을 정의하면 *선택한 속성만* 복제됩니다. 나중에 새 속성을 스키마에 정의하여 복제된 항목에서 사용해도 속성 집합을 편집하여 해당 속성을 선택하지 않을 경우 이러한 새 속성은 복제되지 않습니다.

주

objectClass, nsUniqueId 및 nsDS50ruv 속성과 RDN 이름 지정 속성은 속성 집합에서 제외해도 *항상* 복제됩니다. 이는 LDAP 수정에 objectClass 및 이름 지정 속성이 필요하고, nsUniqueId 및 nsDS50ruv 속성이 있어야만 복제가 제대로 작동하기 때문입니다.

ACI 속성을 제외하면 소비자 복제본의 액세스 제어에 영향을 주게 됩니다. userPassword 속성을 제외하면 어떤 사용자도 소비자 복제본에 대해 인증되지 않습니다.

4. 선택 사항으로, 이 속성 집합에 대한 설명 문자열을 입력하거나 수정합니다. 설명 문자열은 이 속성 집합을 사용하는 복제 계약을 편집할 때와 정의된 속성 집합 목록에 표시되는 텍스트입니다. 설명 문자열을 제공하지 않으면 서버는 포함되거나 제외되는 속성을 기준으로 설명 문자열을 생성합니다.
5. 모두 마쳤으면 저장을 누릅니다.

단편 복제 활성화

단편 복제는 다음과 같이 기존 복제 계약에서만 활성화할 수 있습니다.

1. 347페이지의 "복제 계약 작성"에 설명된 것처럼 복제 계약을 작성하거나 이전에 정의한 계약을 선택하여 수정합니다.
2. 370페이지의 "복제 계약 비활성화"에 설명된 것처럼 복제 계약을 비활성화합니다. 단편 복제 구성을 수정하려면 반드시 계약을 비활성화해야 합니다.
3. 비활성화된 계약을 선택하고 편집을 누릅니다. 표시되는 복제 계약 대화 상자에서 복제된 속성 탭을 선택합니다.
4. "속성 집합만 복제합니다" 확인란을 선택합니다.
5. 드롭다운 목록에서 기존 속성 집합을 선택하거나 새로 만들기를 눌러 350페이지의 "속성 집합 정의"에 설명된 것처럼 새 속성 집합을 정의합니다. 복제된 속성 집합 관리를 눌러 기존 속성 집합 정의를 보고 수정할 수도 있습니다.
단편 복제 시에는 한 개의 속성 집합만 복제 계약에 연결할 수 있습니다. 해당 속성 집합에는 복제할 속성 목록이 포함되어 있어야 합니다.
6. 속성 집합을 선택했으면 확인을 누릅니다. 단편 복제를 구성했으므로 소비자 복제본을 다시 초기화해야 함을 알려주는 메시지가 표시됩니다. 확인을 눌러 메시지를 닫습니다.
7. 활성화를 눌러 복제 계약을 다시 활성화합니다.
8. 복제되는 속성에 따라 소비자 서버에서 스키마 검사를 비활성화해야 합니다.
9. 다른 마스터에서도 이 복제본과 복제 계약을 구성한 경우 이 절차를 반복하여 모든 마스터에서 동일한 속성 집합을 사용하여 단편 복제를 활성화해야 합니다.
10. 이제 소비자 복제본을 초기화하거나, 이미 복제된 경우 다시 초기화해야 합니다. 아래의 "복제본 초기화"를 참조하십시오.

복제본 초기화

복제 계약을 작성한 경우 실제로 복제가 시작되기 전에 소비자 복제본을 초기화해야 합니다. 초기화 중에 공급자 복제본의 데이터가 소비자 복제본으로 복사됩니다.

특정 오류 조건이나 구성 변경 시에는 복제본을 다시 초기화해야 합니다. 다시 초기화하면 소비자에 있는 복제된 접미어의 내용이 삭제되고 마스터에 있는 접미어의 내용으로 교체됩니다. 이렇게 함으로써 복제본이 동기화되어 복제 업데이트를 계속할 수 있습니다. 또한, 여기에 설명된 모든 초기화 방법은 자동으로 소비자 복제본의 색인을 재구성하므로 소비자가 클라이언트 읽기 요청에 대해 적절하게 응답할 수 있습니다.

초기화 시기

복제를 수행하려면 두 복제본이 모두 구성된 후에 복제본을 초기화해야 합니다. 접미어의 데이터가 완전히 소비자에 복사되면 공급자는 소비자에 대한 업데이트 작업을 재생활 수 있습니다.

정상 작동 시에는 소비자를 다시 초기화할 필요가 없지만 어떤 이유로든 백업을 사용하여 단일 마스터 데이터를 복원한 경우에는 백업에서 업데이트하는 모든 복제본을 다시 초기화해야 합니다. 다중 마스터 복제 시 다른 마스터에서 업데이트하지 않은 소비자는 다시 초기화하지 않아도 됩니다.

콘솔에서 온라인으로, 또는 명령줄에서 수동으로 복제본을 초기화할 수 있습니다. 콘솔을 사용한 온라인 초기화는 소규모 소비자를 초기화할 때 편리합니다. 복제 계약을 사용하여 직접 온라인으로 복제본을 초기화할 수도 있습니다. 하지만 각 복제본이 순서대로 초기화되므로 이 방법은 다수의 복제본을 초기화할 때는 적합하지 않습니다. 명령줄을 사용한 수동 초기화는 단일 LDIF 파일을 사용하여 다수의 소비자를 동시에 초기화할 수 있는 효과적인 방법입니다.

마지막으로 숙련된 관리자는 이진 복사 기능을 사용하여 마스터나 소비자 복제본을 복제할 수 있습니다. 이 기능의 특정 제한 사항으로 인해 이진 복사는 수백만 개의 항목이 포함된 복제본과 같이 대규모 데이터베이스 파일이 있는 복제본에만 실용적이며 시간 효율적입니다.

다중 마스터 복제 시 복제본 초기화

다중 마스터 복제의 경우 다음과 같은 순서로 복제본을 초기화해야 합니다.

1. 특정 마스터에 복제할 전체 데이터 집합이 있는지 확인합니다. 이 마스터를 사용하여 다른 마스터의 복제본을 초기화할 수 있습니다.

2. 해당 마스터(357페이지의 "온라인 복제본 초기화" 참조) 또는 마스터 중 하나의 LDIF 파일(359페이지의 "LDIF로 복제본 내보내기" 참조)을 사용하여 소비자 복제본을 초기화합니다.

계단식 복제 시 복제본 초기화

계단식 복제의 경우 항상 다음과 같은 순서로 복제본을 초기화해야 합니다.

1. 다중 마스터 복제도 있는 경우 특정 마스터에 복제할 전체 데이터 집합이 있는지 확인합니다. 이 마스터를 사용하여 다른 마스터의 복제본을 초기화할 수 있습니다.
2. 해당 마스터 복제본을 사용하여 첫 수준의 허브 복제본에 있는 복제본을 초기화합니다.
3. 여러 수준의 허브가 있는 경우 이전에 초기화한 수준의 허브를 사용하여 각 수준을 초기화합니다.
4. 마지막 수준의 허브 복제본에서 전용 소비자에 있는 복제본을 초기화합니다.

다중 마스터 초기화 후의 수렴

다중 마스터 복제의 경우 특정 마스터를 초기화하는 동안 다른 마스터에서 변경 작업을 처리할 수 있습니다. 따라서 초기화가 완료되면 새 마스터는 초기화 데이터에 없는 새 업데이트도 받아야 합니다. 초기화에 상당한 시간이 걸리면 보류 중인 업데이트 수도 증가합니다.

보류 중인 업데이트의 수렴을 허용하기 위해 새로 초기화된 마스터는 초기화 후의 클라이언트 작업 시 자동으로 읽기 전용 모드로 설정됩니다. 이 설정은 명령줄에서 LDIF 파일을 사용하여 초기화하거나 백업을 사용하여 이진 복사를 수행하는 경우에만 적용됩니다.

Directory Server 5.2의 새 기능입니다.

따라서 다중 마스터 구성의 마스터는 초기화 후에 복제 업데이트를 처리하고 읽기 작업을 허용하지만 클라이언트의 모든 쓰기 요청에 대해서는 참조를 반환합니다. 345페이지의 "고급 다중 마스터 구성"에 설명된 것처럼 참조를 정의할 수 있습니다. 다음과 같은 경우에 마스터는 읽기 쓰기 모드로 돌아갑니다.

- ds5BeginReplicaAcceptUpdates 구성 속성을 start로 설정하여 업데이트 작업을 명시적으로 허용한 경우. 업데이트를 활성화하기 전에 새 마스터 복제본이 다른 마스터와 수렴되었는지 확인해야 합니다. 이는 Directory Server 콘솔의 복제 구성 패널이나 명령줄을 통해 확인할 수 있습니다(아래 절차 참조).

수동 작업을 사용하면 업데이트를 허용하기 전에 새 마스터가 다른 마스터와 완전히 동기화되었는지 확인할 수 있으므로 초기화된 마스터에서 업데이트를 활성화하는 데 바람직한 방법입니다.

- 이전에 `ds5referralDelayAfterInit` 속성을 설정한 경우 마스터 복제본은 지정된 지연 후에 자동으로 정상적인 읽기 쓰기 모드로 전환됩니다. 이 속성은 서버에 있는 각 마스터 복제본에 별도로 설정할 수 있습니다.

이 속성을 설정하려면 마스터 복제본이 초기화 후에 다른 마스터와 충분히 수렴될 수 있는 지연을 확인해야 합니다. 이 지연은 예상되는 초기화 크기 및 길이와 다른 마스터에서 동시에 발생하는 변경 속도에 따라 결정됩니다. 마스터에서 초기화 후의 변경 사항을 복제하는 동시에 업데이트 작업을 허용할 경우 알 수 없는 오류가 발생할 수 있습니다. 복제 오류가 발생한 경우에는 *Directory Server Administration Reference*의 오류 참조를 참조하십시오.

주 새 기능으로 인해 마스터 복제본에서 참조를 보내는 경우 쓰기 작업을 수행하려는 클라이언트는 구성된 홑수 제한에 도달할 수 있습니다. 이 경우 클라이언트가 사용 가능한 마스터에 도달할 수 있도록 이 클라이언트의 홑수 제한을 늘려야 합니다. 모든 마스터 복제본을 초기화 또는 다시 초기화하면 클라이언트 업데이트를 허용하는 복제본이 없기 때문에 모든 쓰기 작업이 실패합니다.

항상 초기화된 마스터를 주의해서 모니터하고 참조 속성을 적절하게 설정하여 서버 응답을 최적화해야 합니다.

콘솔을 통해 업데이트 허용

다중 마스터 복제본의 초기화 후에 업데이트 작업을 명시적으로 허용하려면 아래 단계에 따라 수행합니다.

1. **Directory Server** 콘솔의 최상위 구성 탭에서 데이터 노드와 복제된 접미어 노드를 모두 확장한 다음 접미어 아래의 복제 노드를 선택합니다.

콘솔의 오른쪽 패널에 복제본이 초기화되었으며 현재 업데이트 작업에 대해 참조를 반환한다는 메시지가 표시됩니다. 이 메시지에 자동 참조 지연이 활성화되었다는 설명이 있어도 이 절차에 따라 지연을 무시할 수 있습니다.

2. 복제본이 다른 마스터와 수렴되었는지 확인하려면 `insync` 도구를 사용합니다. 모든 서버의 수정 사항 간에 지연이 0(제로)이거나 복제할 변경 사항이 복제본에 없으면 (-1 지연) 복제본은 동기화된 것입니다. 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.
3. 메시지 오른쪽에 있는 버튼을 눌러 업데이트 작업을 즉시 허용합니다.

명령줄을 통해 업데이트 허용

수렴을 확인하고 업데이트 작업을 명시적으로 허용하여 다중 마스터 복제본의 초기화 프로세스를 자동화하는 스크립트에는 다음과 같은 명령을 사용할 수 있습니다.

1. 복제본이 다른 마스터와 수렴되었는지 확인하려면 `insync` 도구를 사용합니다. 모든 서버의 수정 사항 간에 지연이 0(제로)이거나 복제할 변경 사항이 복제본에 없으면 (-1 지연) 복제본은 동기화된 것입니다. 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.
2. 아래 명령을 실행하여 `ds5BeginReplicaAcceptUpdates` 구성 속성을 수정합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn:cn=replica, cn=suffixName, cn=mapping tree, cn=config
changetype: modify
add: ds5BeginReplicaAcceptUpdates
ds5BeginReplicaAcceptUpdates: start
^D
```

복제본을 초기화하면 `ds5BeginReplicaAcceptUpdates` 속성이 자동으로 삭제되어 초기화 후의 업데이트 작업이 다시 거부됩니다.

자동 참조 지연 설정

`ds5ReferralDelayAfterInit` 구성 속성은 복제본에서 참조를 반환할 초기화 후의 시간(초)을 지정합니다. 복제본은 지정된 지연 후에 클라이언트로부터의 업데이트 작업을 자동으로 처리합니다. 이 속성은 [354페이지의 "다중 마스터 초기화 후의 수렴"](#)에 설명된 조건에 따라 각 복제본에 적합한 값으로 설정해야 합니다.

복제본을 최근에 초기화했으며 아직 업데이트를 허용하지 않은 경우 이 속성 값을 변경하면 동적으로 해당 복제본에 영향을 주게 됩니다. 이 값을 수정하여 진행 중인 지연의 길이를 늘리거나 줄일 수 있습니다. 지연이 아직 경과하지 않았으며 복제본에서 업데이트를 허용하는 경우에는 이 속성을 설정해도 영향을 주지 않습니다.

이 속성의 기본값은 -1로, 복제본이 업데이트 작업을 무제한으로 거부합니다. 이 경우 초기화 후에 측정되는 지연이 경과하면 자동으로 업데이트를 허용하도록 지연을 정의할 수도 있습니다. 이미 경과한 지연을 설정하면 복제본이 즉시 업데이트를 허용하게 됩니다.

1. 아래 명령을 실행하여 ds5ReferralDelayAfterInit 속성을 설정합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn:cn=replica, cn=suffixName, cn=mapping tree, cn=config
changetype: modify
replace: ds5ReferralDelayAfterInit
ds5ReferralDelayAfterInit: seconds
^D
```

콘솔에서 복제본 초기화

콘솔을 사용한 온라인 복제본 초기화는 소비자를 초기화 또는 다시 초기화할 수 있는 가장 쉬운 방법입니다. 하지만 다수의 항목(1-2백만 개 이상)을 초기화하는 경우에는 이 프로세스에 많은 시간이 소요될 수 있으므로 명령줄을 사용한 수동 소비자 초기화가 더 효율적일 수 있습니다. 자세한 내용은 [358페이지](#)의 "명령줄에서 복제본 초기화"를 참조하십시오.

콘솔을 사용하여 소비자 복제본을 초기화하는 경우 접미어에 대한 모든 작업(검색 포함)은 초기화 프로세스가 완료될 때까지 마스터 서버로 보내집니다.

Directory Server 콘솔을 사용하면 단편 복제가 구성된 복제본의 초기화를 투명하게 처리할 수 있습니다. 선택한 속성만 초기화 중에 소비자로 보내집니다.

온라인 복제본 초기화

콘솔을 사용하여 복제본을 초기화 또는 다시 초기화하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드와 마스터 복제본의 접미어 노드를 모두 확장한 다음 접미어 아래의 복제 노드를 선택합니다.

오른쪽 패널에 복제본 상태 정보가 표시됩니다.

2. 정의된 계약 목록에서 초기화할 소비자에 해당하는 복제 계약을 선택하고 작업 > 원격 복제본 초기화를 누릅니다.
확인 메시지가 표시되어 소비자 복제본에 저장된 모든 정보가 제거된다고 경고합니다.
3. 확인 대화 상자에서 예를 누릅니다.
온라인 소비자 초기화가 즉시 시작됩니다. 복제 계약의 아이콘에 빨간색 기어가 표시되어 초기화 프로세스의 상태를 나타냅니다.
4. 갱신 > 지금 갱신 또는 갱신 > 계속 갱신을 눌러 소비자 초기화의 상태를 따릅니다.
강조 표시된 계약에 대한 모든 메시지는 목록 아래의 텍스트 상자에 표시됩니다.
복제 및 초기화 상태에 대한 자세한 내용은 [386페이지의 "복제 상태 모니터"](#)를 참조하십시오.

명령줄에서 복제본 초기화

명령줄을 사용한 수동 복제본 초기화는 다수의 항목을 복제하는 배포에 대한 가장 빠른 소비자 초기화 방법입니다. 성능 문제로 인해 온라인 프로세스가 부적절한 경우 수동 프로세스를 사용합니다. 하지만 수동 소비자 초기화 프로세스는 온라인 소비자 초기화 프로세스보다 훨씬 복잡합니다. 먼저 복제 계약을 설치한 후 복제를 초기화하십시오.

복제본을 수동으로 초기화 또는 다시 초기화하려면 먼저 접미어 데이터의 원래 복사본을 LDIF 파일로 내보냅니다. 단편 복제본을 초기화하는 경우에는 파일을 필터링하여 복제된 속성만 유지해야 합니다. 그런 후에 해당 파일을 모든 소비자 서버로 전송하여 가져옵니다. 다중 마스터 복제 배포 시에는 원본 마스터에서 내보낸 LDIF 파일을 사용하여 다른 마스터 및 모든 소비자를 초기화할 수 있습니다. 계단식 복제 환경에서는 동일한 파일을 사용하여 허브 복제본과 해당 소비자를 모두 초기화할 수 있습니다.

항상 구성된 마스터 복제본에서 내보낸 LDIF 파일부터 시작해야 합니다. 복제 데이터가 포함되지 않은 임의의 LDIF를 사용하여 모든 복제본을 초기화할 수는 없습니다. 먼저 LDIF 파일을 마스터 복제본으로 가져온 다음 아래 절차에 따라 내보내야 합니다.

LDIF로 복제본 내보내기

db2ldif -r 또는 db2ldif-task -r 명령을 사용하여 복제본 내용을 LDIF 파일에 저장할 수 있습니다. 자세한 내용은 196페이지의 "명령줄에서 LDIF로 내보내기"를 참조하십시오. 이러한 명령을 사용하여 복제본을 내보내려면 반드시 -r 옵션을 사용해야 합니다.

아래 예에서는 전체 dc=example,dc=com 복제본을 example_master.ldif 파일로 내보냅니다.

```
# /usr/sbin/directoryserver -s example stop
# /usr/sbin/directoryserver db2ldif -r -s "dc=example,dc=com" \
-a /var/ds5/slapd-serverID/ldif/example_master.ldif
# /usr/sbin/directoryserver -s example start
```

필요한 경우 LDIF 파일을 필터링한 다음 소비자 호스트로 전송하여 소비자 복제본을 초기화할 수 있습니다.

단편 복제를 위한 LDIF 파일 필터링

단편 복제를 구성한 경우 내보낸 LDIF 파일을 소비자 서버에 복사하기 전에 사용하지 않는 속성을 필터링하여 제거해야 합니다. Directory Server는 이러한 용도로 fildif 도구를 제공합니다. 이 도구는 특정 LDIF 파일을 필터링하여 복제 계약에 정의된 속성 집합에서 허용하는 속성만 유지합니다.

이 도구는 서버 구성을 읽어 속성 집합 정의를 확인합니다. 구성 파일을 읽으려면 루트나, 프로세스 및 파일을 소유한 사용자(nsslapd-localuser 속성으로 지정됨)로 fildif 도구를 실행해야 합니다. 예를 들어, 아래 명령은 이전 예의 dc=example,dc=com 접미어에서 내보낸 파일을 필터링합니다.

```
# CAMUS=/var/opt/mps/serverroot/slapd-camus
# /var/opt/mps/serverroot/shared/bin/fildif \
-i $CAMUS/ldif/example_master.ldif \
-o $CAMUS/ldif/filtered.ldif -c $CAMUS/config/dse.ldif \
-b "cn=rousseau.example.com:389, cn=replica, \
cn=\"dc=example,dc=com\", cn=mapping tree, cn=config"
```

-i 및 -o 옵션은 각각 입력 파일과 출력 파일을 나타냅니다. -c 옵션은 복제 계약과 속성 집합 정의가 포함된 구성 파일입니다. dse.ldif 파일은 복제 계약 및 속성 집합을 포함한 cn=config 항목의 내용이 저장되는 위치입니다.

-b 옵션은 단편 복제가 정의된 복제 계약의 DN입니다. Directory Server 콘솔에서 디렉토리 관리자로서 cn=config 접미어를 탐색하면 이 항목을 찾을 수 있습니다. 접미어에 대한 cn=replica 항목 아래에서 이 항목을 선택한 다음 편집 > DN 복사 메뉴 항목을 사용하여 이 DN을 클립보드에 복사한 후 명령을 입력할 때 사용할 수 있습니다.

fildif 도구에 대한 전체 명령줄 구문을 보려면 *Directory Server Man Page Reference*를 참조하십시오.

그런 후에 fildif 도구에서 생성된 filtered.ldif 파일을 사용하여 이 복제 계약의 소비자를 초기화할 수 있습니다. 파일을 소비자 서버로 전송하여 다음 절에 설명된 것처럼 가져옵니다.

소비자 복제본으로 LDIF 파일 가져오기

Directory Server 콘솔의 가져오기 기능을 사용하거나, directoryserver ldif2db 명령 또는 directoryserver ldif2db-task 명령을 사용하여 마스터 복제본 내용이 포함된 LDIF 파일을 소비자 복제본으로 가져올 수 있습니다. 모든 가져오기 작업과 마찬가지로 이러한 명령을 사용하여 가져오기를 수행하려면 디렉토리 관리자의 바인드 DN과 비밀번호가 필요합니다. 가져오기 방법에 대해서는 [190페이지의 "명령줄에서 LDIF 가져오기"](#)에서 설명합니다.

아래 예에서는 LDIF 파일을 가져와서 dc=example,dc=com 소비자 복제본을 초기화하는 방법을 보여줍니다.

```
# /usr/sbin/directoryserver -s example stop
# /usr/sbin/directoryserver ldif2db -s "dc=example,dc=com" \
-i example_master.ldif
# /usr/sbin/directoryserver -s example start
```

ldif2db-task를 사용하는 경우 미리 서버를 중지할 필요가 없습니다. 자세한 내용은 *Directory Server Man Page Reference*를 참조하십시오.

이진 복사를 사용한 복제본 초기화

이진 복사 기능은 한 서버의 이진 백업 파일을 사용하여 다른 서버에 동일한 디렉토리 내용을 복원함으로써 전체 서버를 복제합니다. 이 고급 기능은 디렉토리 서버의 데이터베이스 파일과 상호 작용하며 숙련된 관리자만 사용해야 합니다.

이진 복사 제한

이진 복사 기능은 한 시스템의 데이터베이스 파일을 다른 시스템으로 이동하기 때문에 다음과 같은 엄격한 제한이 적용됩니다.

- 두 시스템은 서비스 팩이나 패치를 비롯한 동일한 운영 체제 및 하드웨어를 사용해야 합니다.
- 두 시스템에 설치된 Directory Server는 이진 형식(32비트 또는 64비트), 서비스 팩 및 패치 수준까지 동일해야 합니다.
- 두 서버의 디렉토리 트리와 구성된 접미어는 동일해야 합니다. 반드시 모든 접미어의 데이터베이스 파일을 함께 복사해야 하며 개별 접미어는 복사할 수 없습니다.
- VLV(가상 목록 보기) 색인을 비롯한 두 서버의 각 접미어 색인은 동일하게 구성되어 야 하며, 접미어의 데이터베이스 이름도 같아야 합니다.
- o=NetscapeRoot 접미어가 포함된 Directory Server는 복사할 수 없으므로 Administration Server의 구성 디렉토리가 될 수 없습니다.
- 각 서버에는 동일한 접미어가 복제본으로 구성되어 있어야 하며, 두 서버의 복제본은 같은 역할(마스터, 허브 또는 소비자)을 가져야 합니다. 단편 복제를 구성하는 경우 모든 서버에서 동일하게 구성해야 합니다.
- 속성 암호화는 두 서버에서 모두 사용할 수 없습니다.
- 속성 값 고유성 플러그 인을 사용하는 경우 두 서버에서 동일하게 구성해야 하며, 아래 절차에 설명된 것처럼 새 복사본에서 다시 구성해야 합니다.

위의 조건에 모두 부합되면 다른 마스터 서버의 이진 복사본을 사용하여 마스터를, 또는 다른 소비자 서버의 이진 복사본을 사용하여 소비자를 초기화하거나 다시 초기화할 수 있습니다. 아래의 두 절차에서는 이진 복사를 수행하는 대체 방법에 대해 설명합니다. 하나는 서버를 중지할 필요가 없는 방법이고 다른 하나는 최소 디스크 공간을 사용하는 방법입니다.

서버를 중지하지 않는 이진 복사

아래 절차는 정상적인 백업 기능을 사용하여 서버에 있는 데이터베이스 파일의 복사본을 작성하기 때문에 이진 복사에 권장되는 방법입니다. 정상적인 백업을 수행하면 서버를 중지할 필요 없이 모든 데이터베이스 파일을 일관된 상태로 유지할 수 있습니다.

하지만 이 절차에는 주의해야 하는 특정 제한 사항이 있습니다. 백업 및 복원 작업 시 같은 시스템에 데이터베이스 파일의 복사본이 작성되므로 데이터베이스 파일에 필요한 각 시스템의 디스크 공간이 두 배로 증가합니다. 또한 디렉토리에 기가바이트의 데이터가 포함되어 있을 경우 데이터베이스 파일에 대한 실제 복사 작업에 상당한 시간이 소요될 수 있습니다. 디스크 공간이 제한되어 있거나 데이터베이스 파일이 대규모일 경우 [362페이지](#)의 "[최소 디스크 공간을 사용한 이진 복사](#)"를 참조하십시오.

1. 새 복제본의 대상 시스템에 **Directory Server**를 설치하고, 필요한 경우 서버의 새 인스턴스를 작성한 다음 [360페이지](#)의 "[이진 복사 제한](#)"에 따라 구성합니다.
2. 복제 토폴로지에 이 복제본과 연결된 모든 복제 계약을 작성합니다. 여기에는 공급자와 이 복제본 간의 계약 및 전용 소비자가 아닌 경우 이 복제본과 해당 소비자 간의 계약이 포함됩니다.
3. 초기화하려는 유형(마스터, 허브 또는 소비자)의 완전히 구성 및 초기화된 복제본을 선택하고 [198페이지](#)의 "[콘솔에서 서버 백업](#)"의 절차에 따라 정상적인 백업을 수행합니다.
4. ftp 명령 등을 사용하여 백업 디렉토리의 파일을 대상 시스템의 디렉토리로 복사 또는 전송합니다.
5. [199페이지](#)의 "[백업을 사용한 데이터 복원](#)"의 절차에 따라 파일을 대상 서버에 로드합니다.
6. 다중 마스터 복제 시나리오에서 새 마스터를 초기화한 경우 [354페이지](#)의 "[다중 마스터 초기화 후의 수렴](#)"의 절차에 따라 새 복제본이 클라이언트로부터의 업데이트 작업을 허용하도록 설정합니다.

최소 디스크 공간을 사용한 이진 복사

아래 절차에서는 데이터베이스 파일의 백업 복사본을 만들지 않으므로 최소 디스크 공간을 사용하며, 따라서 소요되는 시간도 단축됩니다. 하지만 이 경우에는 데이터베이스 파일의 일관된 상태를 유지하기 위해 복제되는 서버를 중지해야 합니다.

주의 다중 마스터 복제 시나리오에서 이미 사용되고 있는 마스터를 다시 초기화할 때는 이 절차를 사용할 수 *없습니다*. 이 절차는 소비자 서버를 다시 초기화하거나 새 마스터 서버를 초기화하는 경우에만 사용해야 합니다. 기존 마스터 복제본을 다시 초기화하려면 온라인 초기화를 사용하거나 LDIF 파일을 가져오거나 [361페이지](#)의 "[서버를 중지하지 않는 이진 복사](#)" 절차에 따라 수행합니다.

1. 새 복제본의 대상 시스템에 Directory Server를 설치하고, 필요한 경우 서버의 새 인스턴스를 작성한 다음 [360페이지의 "이진 복사 제한"](#)에 따라 구성합니다.
2. 복제 토폴로지에 이 복제본과 연결된 모든 복제 계약을 작성합니다. 여기에는 공급자와 이 복제본 간의 계약 및 전용 소비자가 아닌 경우 이 복제본과 해당 소비자 간의 계약이 포함됩니다.
3. [35페이지의 "Directory Server 시작 및 중지"](#)에 설명된 것처럼 초기화 또는 다시 초기화할 대상 서버를 중지합니다.
4. 초기화하려는 복제본과 같은 유형(마스터, 허브 또는 소비자)의, 완전히 구성 및 초기화된 접미어를 선택하고 서버도 중지합니다. 다중 마스터 구성의 마스터 복제본을 복제하는 경우 이 복제본을 중지하기 전에 다른 마스터의 최신 변경 사항이 모두 적용되어 있는지 확인해야 합니다.
5. 트랜잭션 로그, 변경 로그 및 지역 파일(__db.xxx 파일)을 포함한 모든 데이터베이스 파일을 대상 서버에서 제거합니다. 파일 위치를 변경하지 않은 경우 데이터베이스 파일과 트랜잭션 로그는 `ServerRoot/slapd-serverID/db` 디렉토리에 위치해 있습니다.
6. ftp 명령 등을 사용하여 트랜잭션 로그를 비롯한 소스 복제본 시스템의 모든 데이터베이스 파일을 대상 시스템으로 복사 또는 전송합니다. 파일 위치를 변경하지 않은 경우 데이터베이스 파일과 트랜잭션 로그는 `ServerRoot/slapd-serverID/db` 디렉토리에 위치해 있습니다.

마스터 또는 허브 복제본을 초기화하는 경우에는 기본적으로 `ServerRoot/slapd-serverID/changeLog`에 위치해 있는 변경 로그의 모든 파일도 복사해야 합니다.
7. 소스 및 대상 서버를 모두 다시 시작합니다.

참조 무결성 플러그인 활성화

참조 무결성 플러그인을 사용하는 경우 모든 마스터 서버에서 이 플러그인을 활성화해야 하지만 허브나 소비자 서버에서는 활성화할 필요가 없습니다. [110페이지의 "복제에 참조 무결성 사용"](#)을 참조하십시오.

SSL을 통한 복제

모든 복제 작업이 SSL 연결을 통해 수행되도록 복제에 사용되는 Directory Server를 구성할 수 있습니다. 이렇게 하려면 다음을 수행합니다.

1. 공급자 및 소비자 서버에서 모두 SSL을 사용하도록 구성합니다.
자세한 내용은 [11장, "인증 및 암호화 관리"](#)를 참조하십시오.

주

- 공급자 서버 인증서가 SSL 핸드셰이크 중에 클라이언트 역할을 할 수 없는 SSL 서버 전용 인증서이면 SSL을 통한 복제가 제대로 수행되지 않습니다.
 - 자체 서명된 인증서를 사용한 SSL을 통한 복제는 현재 지원되지 않습니다.
-

2. 소비자 서버의 접미어에 대한 복제가 구성되어 있지 않으면 [338페이지의 "소비자 복제본 활성화"](#)에 설명된 것처럼 복제를 활성화합니다.
3. [339페이지의 "고급 소비자 구성"](#)의 절차에 따라 소비자 및 다른 복제 관리자의 인증서 항목 DN을 정의합니다.
4. 공급자 서버의 접미어에 대한 복제가 구성되어 있지 않으면 [340페이지의 "허브 복제본 활성화"](#) 또는 [344페이지의 "마스터 복제본 활성화"](#)에 설명된 것처럼 복제를 활성화합니다.
5. 공급자 서버에서 보안 SSL 포트를 통해 소비자에게 업데이트를 보내는 새 복제 계약을 작성합니다. [347페이지의 "복제 계약 작성"](#)의 절차에 따라 수행합니다. 소비자 서버의 보안 포트를 지정하고 비밀번호나 인증서를 사용하는 SSL 옵션을 선택합니다. 복제 관리자나 인증서 중에서 선택한 SSL 옵션의 DN을 입력합니다.

복제 계약의 구성이 끝나면 공급자는 SSL을 통해 모든 복제 업데이트 메시지를 소비자에게 보내며, 해당 옵션을 선택한 경우 인증서를 사용합니다. 콘솔에서 SSL용으로 구성된 계약을 사용하면 소비자 초기화 시에도 보안 연결이 사용됩니다.

WAN을 통한 복제

Directory Server 5.2에서는 WAN을 통해 연결된 시스템 간의 다중 마스터 복제(MMR)를 비롯한 모든 형식의 복제를 수행할 수 있습니다. 복제 기법에 대한 내부 향상 기능을 통해 공급자 서버는 대기 시간이 길고 대역폭이 작은 네트워크에서도 적절한 지연을 유지하여 소비자 초기화 및 업데이트를 수행할 수 있습니다.

주 복제 데이터 전송 속도는 대역폭의 측면에서 사용 가능한 물리적 매체가 허용하는 속도보다 항상 떨어집니다. 복제본 간의 업데이트 용량이 사용 가능한 대역폭에 모두 수용되지 않으면 조정을 통해서도 복제본이 과도한 업데이트 로드로 인해 분산되는 것을 방지할 수 없습니다. 복제 지연 및 업데이트 성능은 수정 속도, 항목 크기, 서버 하드웨어, 평균 대기 시간, 평균 대역폭 등 다양한 요인에 의해 결정됩니다. 사용자 환경에서의 복제에 대한 질문이 있으면 Sun Professional 서비스 담당자에게 문의하십시오.

복제 기법의 내부 매개 변수는 기본적으로 WAN에 최적화되어 있습니다. 하지만 위에 명시된 요인들로 인해 복제 속도가 느려질 경우 실제 테스트를 통해 창 크기 및 그룹 크기 매개 변수를 조정할 수 있습니다. 네트워크 사용량이 많은 시간을 피해 복제를 예약함으로써 전체적인 네트워크 사용을 향상시킬 수도 있습니다. 마지막으로 Directory Server는 대역폭 사용을 최적화하기 위한 복제 데이터 압축 기능을 지원합니다.

네트워크 매개 변수 구성

아래의 두 매개 변수는 네트워크에서 보다 효율적으로 항목을 전송하기 위해 복제 기법에서 항목을 그룹화하는 방법을 결정합니다. 또한 공급자 및 소비자가 복제 업데이트 메시지와 승인을 교환하는 방법에도 영향을 줍니다.

- 창 크기(기본값 10) - 소비자로부터의 즉각적인 응답 없이 보낼 수 있는 최대 업데이트 메시지 수를 나타냅니다.
각각의 메시지 후에 승인을 기다리지 않고 다수의 메시지를 연속해서 빨리 보내는 것이 더욱 효율적입니다. 적절한 창 크기를 사용하면 복제본이 복제 업데이트 또는 승인이 도착할 때까지 기다리는 시간을 줄일 수 있습니다.
소비자 복제본이 공급자보다 지연되는 경우, 창 크기를 기본값보다 높은 값(예: 100)으로 증가시키고 다른 조정 작업을 수행하기 전에 복제 성능을 다시 확인합니다. 복제 업데이트 속도가 빠르고 업데이트 간격이 짧으면 LAN으로 연결된 복제본에서도 창 크기를 높게 설정하는 것이 좋습니다.
- 그룹 크기(기본값 1) - 하나의 업데이트 메시지로 처리할 수 있는 최대 데이터 수정 항목 수를 나타냅니다.

네트워크 연결이 복제 과정에서 병목이 되는 경우, 그룹 크기를 기본값보다 높은 값(예: 10)으로 증가시키고 복제 성능을 다시 확인합니다. 그룹 크기를 증가시킬 때 다음을 수행합니다.

- 창 크기가 그룹 크기보다 훨씬 높게 설정되어 있는지 확인합니다.
- 창 크기/그룹 크기가 소비자의 cn=config에 설정된 nsslapd-maxThreadsPerConn 값보다 훨씬 큰지 확인합니다(일반적으로 두 배).

그룹 크기가 1보다 높게 설정되어 있으면 공급자는 그룹이 채워질 때까지 기다리지 않고 공급자에게 업데이트를 보냅니다. 그러나 보낼 업데이트 목록이 그룹 크기보다 크면 공급자는 소비자가 병렬로 처리할 수 있는 독립적인 수정 항목의 여러 그룹을 보내기 위해 가능한 항목을 그룹화합니다.

수정 결과를 모니터하고 적절하게 조정합니다. 자세한 내용은 [386페이지의 "복제 상태 모니터"](#)를 참조하십시오.

모든 복제 계약에서 두 네트워크 매개 변수를 구성할 수 있으므로 각 소비자의 특정 네트워크 조건에 따라 복제 성능을 조정할 수 있습니다.

다음과 같이 창 크기 및 그룹 크기 매개 변수를 수정하는 경우 복제를 중단할 필요가 없습니다.

1. Directory Server 콘솔의 구성 탭을 선택하고 데이터 노드와 복제된 접미어 노드를 확장합니다.
2. 접미어 아래의 복제 노드를 선택하고 오른쪽 패널에서 구성할 복제 계약을 선택한 다음 편집을 누릅니다.
3. 복제 계약 대화 상자의 네트워크 탭을 선택하고 창 크기(1에서 1000까지) 및 그룹 크기(1에서 100까지)에 새 값을 입력합니다. 그룹 크기는 창 크기보다 작거나 같아야 합니다.
4. 확인을 눌러 새 값을 저장하고 복제 계약 대화 상자를 닫습니다.

다음 복제 업데이트를 해당 소비자로 보낼 때는 새 매개 변수가 적용됩니다.

복제 작업 예약

복제본 간에 즉시 동기화할 필요가 없으면 WAN 상에서 데이터를 복제하는 다른 방법으로 네트워크 사용량이 적은 기간에 업데이트를 예약할 수 있습니다. 네트워크 가용성이 커지면 업데이트 속도도 상당히 증가할 수 있으며, 복제 메시지로 인해 이미 사용량이 많은 네트워크의 혼잡을 가중시키는 것을 방지할 수 있습니다.

다음과 같이 복제 계약을 통해 모든 소비자에 개별적으로 일간 또는 주간 업데이트를 예약할 수 있습니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드와 복제된 접미어 노드를 확장합니다.
2. 접미어 아래의 복제 노드를 선택하고 오른쪽 패널에서 구성할 복제 계약을 선택한 다음 편집을 누릅니다.
3. 복제 계약 대화 상자의 일정 탭을 선택하고 주간 일정 옆에 있는 라디오 버튼을 선택합니다.
4. 다음과 같이 일정을 정의합니다.
 - a. 주간 업데이트의 경우 복제를 수행할 요일 옆에 있는 확인란을 선택합니다. 특정 요일의 복제를 보다 자세히 제한하려면 선택 사항으로 시간 범위(24시간 형식 사용)를 입력할 수도 있습니다.
 - b. 일간 업데이트의 경우 모두를 눌러 매일 복제하도록 선택하고 복제를 수행할 시간 범위(24시간 형식)를 입력합니다.
 자정 전후로 이어진 시간 범위는 지정할 수 없습니다.
5. 확인을 눌러 새 값을 저장하고 복제 계약 대화 상자를 닫습니다.
 새 일정이 즉시 적용되므로 해당 소비자에 대한 다음 복제 업데이트는 일정에 처음 포함될 때까지 지연됩니다.

데이터 압축

복제의 대역폭 사용량을 줄이려면 소비자를 업데이트할 때 데이터를 압축하여 보내도록 복제를 구성할 수 있습니다. 복제 기법은 Zlib 압축 라이브러리를 사용합니다. 압축을 사용하려면 공급자와 소비자 모두 Solaris 또는 Linux 플랫폼에서 실행해야 합니다.

복제 압축을 구성하려면 마스터 서버에 있는 복제 계약 항목의 `ds5ReplicaTransportCompressionLevel` 속성을 설정해야 합니다. 이 속성은 다음 값 중 하나를 가집니다.

- 0 - 압축을 수행하지 않습니다. 이것은 `ds5ReplicaTransportCompressionLevel` 속성이 정의되어 있지 않을 때의 기본 동작입니다.
- 1 - Zlib 라이브러리의 기본 압축 수준을 사용합니다.
- 2 - Zlib 라이브러리의 최소 크기 압축 수준을 사용합니다.
- 3 - Zlib 라이브러리의 최대 속도 압축 수준을 사용합니다.

실제 테스트를 통해 WAN 환경에서 예상되는 복제 사용량에 가장 적합한 압축 수준을 선택해야 합니다. 압축 및 압축 해제를 수행하면 복제 속도가 저하되므로 네트워크 지연이 크지 않은 LAN상에서는 이 매개 변수를 설정하지 마십시오.

예를 들어, `east.example.com`에 있는 소비자에 복제 업데이트를 보낼 때 가장 빠른 압축을 사용하려면 아래의 `ldapmodify` 명령을 실행합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=east.example.com:389,cn=replica,cn="suffixDN",
cn=mapping tree,cn=config
changetype: modify
add: ds5ReplicaTransportCompressionLevel
ds5ReplicaTransportCompressionLevel: 3
^D
```

압축 수준 설정에 대한 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.

복제 토폴로지 수정

이 절에서는 복제 계약 편집 또는 제거, 복제본 수준 올리기, 내리기 또는 비활성화, 소비자에 대한 업데이트 강제, 변경 로그 관리 등 기존 복제 토폴로지를 관리하는 절차에 대해 설명합니다.

복제 계약 관리

마스터 접미어에 대한 복제 패널에서 계약의 인증 정보 변경, 특정 소비자에 대한 복제 중단 또는 토폴로지서에서 소비자 제거 등 복제 계약을 관리할 수 있습니다.

복제 관리자 변경

복제 계약을 편집하여 소비자 서버에 바인드할 때 사용하는 복제 관리자 아이디를 변경할 수 있습니다. 복제 중단을 방지하려면 복제 계약을 수정하기 전에 소비자에 새 복제 관리자 항목 또는 인증서 항목을 정의해야 합니다. 하지만 바인드 실패로 인해 복제가 중단된 경우 복제 기법은 복제 복구 설정의 제한 내에서 오류 수정에 필요한 모든 업데이트를 자동으로 보냅니다(339페이지의 "고급 소비자 구성" 참조).

소비자 인증에 사용되는 복제 관리자를 변경하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드와 복제된 접미어 노드를 모두 확장한 다음 접미어 아래의 복제 노드를 선택합니다.
2. 오른쪽 패널에서 수정할 복제 계약을 선택하고 편집을 누릅니다.
3. 복제 계약 대화 상자에서 연결 탭을 선택합니다.
상태 줄에 소비자 서버의 호스트 이름 및 포트가 표시됩니다.
4. DN 및 비밀번호 필드를 수정하여 다른 복제 관리자 항목의 DN 및 비밀번호나 소비자 서버에 있는 인증서 항목의 DN을 지정합니다.
5. 복제 계약에서 보안 포트를 통한 SSL을 사용하는 경우 옵션 버튼을 눌러 보안 인증 유형을 선택할 수도 있습니다. 비밀번호를 사용하여 연결하면 공급자는 지정된 DN에 대해 암호화된 SSL 연결을 통한 단순 인증을 사용합니다. 인증서를 사용하여 연결하면 DN 필드에는 인증서 항목의 DN이 지정되며 비밀번호를 입력할 필요가 없습니다.
기존 복제 계약을 비보안 인증에서 보안 인증으로, 또는 그 반대로 전환할 수는 없습니다. 다른 보안 설정으로 복제를 활성화하려면 새 복제 계약을 작성해야 합니다.
6. 확인을 눌러 변경 사항을 저장합니다.

복제 계약 복사

복제 계약 복사는 대규모 복제 토폴로지에서 공급자 복제본의 다수 소비자를 구성할 수 있는 간단한 방법입니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드와 복제된 접미어 노드를 모두 확장한 다음 접미어 아래의 복제 노드를 선택합니다.
2. 복제 계약 목록에서 복사할 계약을 선택합니다. 소비자에 대한 보안 연결을 사용하는 새 계약을 작성하려면 보안 포트를 사용하는 기존 계약을 선택해야 합니다. 새로운 비보안 계약을 작성하려면 역시 비보안 계약을 선택해야 합니다.
편집을 누른 다음 복제 계약 대화 상자의 탭을 탐색하여 계약 구성을 확인합니다. 이러한 탭의 구성에 대해서는 해당 절에서 설명합니다.
 - 연결 탭은 369페이지의 "복제 관리자 변경"에서 설명합니다.
 - 일정 및 네트워크 탭은 364페이지의 "WAN을 통한 복제"에서 설명합니다.
 - 복제된 속성 탭은 349페이지의 "단편 복제 구성"에서 설명합니다.
3. 같은 복제 계약이 선택된 상태에서 복사 버튼을 누릅니다.
4. 목록에서 새 소비자의 호스트 이름과 포트 번호를 선택하거나 호스트 추가 버튼을 눌러 다른 호스트와 포트를 사용합니다. 이 목록과 호스트 추가 대화 상자를 사용하면 복사되는 소비자 계약과 동일한 보안 유형의 소비자만 선택할 수 있습니다.
5. 목록에서 호스트 이름을 선택하고 확인을 눌러 해당 소비자 서버에 대한 새 복제 계약을 작성합니다.
6. 새 계약은 기존 계약의 모든 구성 정보를 복사하므로 같은 비밀번호를 사용하여 두 서버에서 동일한 복제 관리자 항목을 정의해야 합니다. 예를 들어, 새 계약의 구성을 수정하여 복제 관리자 DN을 변경하려면 목록에서 해당 DN을 선택하고 편집을 누릅니다.

복제 계약 비활성화

복제 계약을 비활성화하면 마스터에서 지정된 소비자로 업데이트를 보내지 않습니다. 이 경우 해당 서버에 대한 복제가 중지되지만 모든 계약 설정은 그대로 유지됩니다. 나중에 계약을 다시 활성화하면 복제를 계속할 수 있습니다. 중단 후에 복제 기법을 계속하는 방법은 아래의 "복제 계약 활성화"를 참조하십시오.

복제 계약을 비활성화하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드와 복제된 접미어 노드를 모두 확장한 다음 접미어 아래의 복제 노드를 선택합니다.
2. 오른쪽 패널에서 비활성화할 복제 계약을 선택합니다.
3. 계약 목록 아래의 상자에서 작업 > 계약 비활성화를 선택합니다.
4. 예를 눌러 복제 계약 비활성화를 확인합니다.

목록에서 이 계약에 대한 아이콘이 바뀌어 계약이 비활성화되었음을 표시합니다.

복제 계약 활성화

복제 계약을 활성화하면 지정된 소비자부터 복제가 계속됩니다. 하지만 복제 복구 설정에서 허용하는 기간보다 오랫동안 복제가 중단되었으며 다른 공급자가 소비자를 업데이트하지 않은 경우에는 소비자를 다시 초기화해야 합니다. 복제 복구 설정은 공급자 변경 로그의 최대 크기 및 수명과 소비자의 지연 제거로 구성됩니다([339페이지의 "고급 소비자 구성"](#) 참조).

중단 기간이 짧고 복제를 복구할 수 있으면 마스터는 계약이 다시 활성화될 때 자동으로 소비자를 업데이트합니다.

복제 계약을 활성화하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드와 복제된 접미어 노드를 모두 확장한 다음 접미어 아래의 복제 노드를 선택합니다.
2. 오른쪽 패널에서 활성화할 복제 계약을 선택합니다.
3. 계약 목록 아래의 상자에서 활성화 버튼을 누릅니다.
4. 필요한 경우 소비자 복제본을 다시 초기화합니다.

복제 계약 삭제

복제 계약을 삭제하면 해당 소비자에 대한 복제가 중지되고 계약에 대한 모든 구성 정보가 제거됩니다. 나중에 복제를 계속하려면 계약을 삭제하지 말고 [370페이지의 "복제 계약 비활성화"](#)에 설명된 것처럼 계약을 비활성화하십시오.

복제 계약을 삭제하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드와 복제된 접미어 노드를 모두 확장한 다음 접미어 아래의 복제 노드를 선택합니다.
2. 오른쪽 패널에서 삭제할 복제 계약을 선택합니다.
3. 계약 목록 오른쪽에 있는 삭제 버튼을 누릅니다.
4. 예를 눌러 복제 계약 삭제를 확인합니다.

복제본 수준 올리기 또는 내리기

복제본 수준을 올리거나 내리면 복제 토폴로지에서 해당 역할이 변경됩니다. 전용 소비자의 수준을 올리면 허브가 되고 허브의 수준을 올리면 마스터가 됩니다. 이와 마찬가지로 마스터의 수준을 내리면 허브가 되고 허브의 수준을 내리면 전용 소비자가 됩니다. 하지만 마스터 수준을 직접 소비자나 내리거나 소비자 수준을 직접 마스터로 올릴 수는 없습니다.

다중 마스터 복제 기법에서 수준 올리기 및 내리기를 사용하면 토폴로지의 유연성이 크게 증가합니다. 이전에 소비자 복제본으로 처리한 사이트의 규모가 커지면 증가한 로드를 처리하기 위해 여러 개의 복제본이 있는 허브가 필요합니다. 복제본 내용에 대한 다수의 수정이 로드에서 포함되어 있으면 신속한 로컬 변경을 허용하는 마스터가 되어 다른 사이트의 다른 마스터로 변경 사항을 복제할 수 있습니다.

복제본 수준을 올리거나 내리려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드와 복제된 접미어 노드를 모두 확장한 다음 접미어 아래의 복제 노드를 선택합니다.
2. 오른쪽 패널에서 변경 > 복제본 수준 올리기/내리기 메뉴 항목을 선택합니다.
3. 복제 마법사에서 허용되는 새 역할만 선택하여 새 복제본 역할에 대한 구성 절차만 수행할 수 있습니다. 다음과 같은 결과에 주의해야 합니다.
 - 마스터 수준을 허브로 내리면 복제본은 읽기 전용이 되어 나머지 마스터로 참조를 보내도록 구성됩니다. 새 허브는 허브나 전용 소비자 등 해당 소비자를 모두 유지합니다.
 - 단일 마스터 수준을 허브로 내리면 마스터 복제본이 없는 토폴로지가 됩니다. 마법사는 새 마스터가 정의될 것이라는 가정 하에 이 작업을 허용합니다. 하지만 새 마스터를 다중 마스터로 추가하여 다른 마스터의 수준을 내리기 전에 초기화하는 것이 바람직합니다.

- 허브 수준을 소비자로 내리면 모든 복제 계약이 삭제됩니다. 다른 허브나 마스터가 허브 소비자를 업데이트하지 않은 경우 해당 소비자가 더 이상 업데이트되지 않습니다. 나머지 허브나 마스터에서 새 계약을 작성하여 소비자를 업데이트해야 합니다.
- 소비자 수준을 허브로 올리면 해당 변경 로그가 활성화되며 소비자와의 새 계약을 정의할 수 있습니다.
- 허브 수준을 마스터로 올리면 복제본이 수정 요청을 허용하며 다른 마스터, 허브 또는 전용 소비자와의 새 계약을 정의할 수 있습니다.

복제본 비활성화

복제본을 비활성화하면 복제 토폴로지에서 이 복제본이 제거되어 마스터, 허브 또는 소비자 역할에 따라 업데이트되거나 업데이트를 보내지 않습니다. 공급자를 비활성화하면 모든 복제 계약이 삭제되며 다시 복제본을 활성화할 경우 새로 복제 계약을 작성해야 합니다.

복제본을 비활성화하려면 다음을 수행합니다.

1. **Directory Server** 콘솔의 최상위 구성 탭에서 데이터 노드와 복제된 접미어 노드를 모두 확장한 다음 접미어 아래의 복제 노드를 선택합니다.
2. 오른쪽 패널에서 변경 > 복제 비활성화 메뉴 항목을 선택합니다.
3. 확인 대화 상자에서 예를 누릅니다.
4. 선택 사항으로, 이 접미어에 대한 쓰기 권한 및 참조를 재설정합니다. 복제본을 비활성화해도 이 설정은 그대로 유지됩니다. 예를 들어, 비활성화된 소비자도 이전의 마스터 복제본으로 계속 수정 요청을 보냅니다.

쓰기 권한 및 참조를 수정하려면 구성 탭에서 이 접미어 노드를 선택하고 오른쪽 패널의 설정 탭에서 원하는 항목을 수정합니다. 자세한 내용은 [174페이지의 "액세스 권한 및 참조 설정"](#)을 참조하십시오.

변경 로그 이동

변경 로그는 지정된 공급자 복제본에 대한 모든 수정 사항의 내부 기록으로, 서버에서 다른 복제본을 수정할 때 사용됩니다. 변경 로그의 내용은 서버에서 자동으로 관리하며 서버를 다시 시작한 후에도 다중 마스터 업데이트를 통해 업데이트됩니다.

Directory Server의 이전 버전에서는 LDAP를 통해 변경 로그를 액세스할 수 있었지만 지금은 서버에서 내부용으로만 사용됩니다. 변경 로그를 읽어야 하는 응용 프로그램이 있으면 이전 버전과의 호환성을 위해 레트로 변경 로그 플러그인을 사용합니다. 자세한 내용은 [381페이지의 "레트로 변경 로그 플러그인 사용"](#)을 참조하십시오.

관리자는 변경 로그가 저장된 디스크가 가득찬 경우와 같이 파일을 다른 위치로 이동할 필요가 있을 때만 변경 로그를 수정해야 합니다.

변경 로그를 비활성화하거나 새 위치로 이동한 경우에는 다시 초기화해야 합니다. 변경 로그 디렉토리가 바뀔 때 토폴로지의 다른 서버로 복제되지 않은 변경 사항이 있으면 해당 토폴로지의 모든 소비자를 다시 초기화해야 합니다. 변경 로그가 이동되기 전에 모든 변경 사항이 복제된 경우(즉, 모든 서버가 동기화된 상태)에는 다시 초기화할 필요가 없습니다.

운영 체제의 `rename` 또는 `mv` 명령이 아닌 Directory Server 콘솔을 사용하여 변경 로그를 이동해야 합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드를 선택한 다음 오른쪽 패널에서 복제 탭을 선택합니다.
2. 텍스트 필드에 새 위치를 입력합니다. 이 위치는 지금부터 변경 로그를 저장할 새 경로와 디렉토리 이름입니다. 예를 들어 기본 위치인 `ServerRoot/slapd-serverID/changeLogdb`의 변경 로그를 `ServerRoot/slapd-serverID/newchangeLog`로 이동할 수 있습니다.
이전 위치의 기존 변경 로그는 삭제되며 새 위치에 새 변경 로그가 유지됩니다.
3. 복제 탭에서 저장을 누릅니다.
4. Directory Server를 다시 시작합니다.
5. [352페이지의 "복제본 초기화"](#)에 설명된 것처럼 소비자를 다시 초기화합니다.

복제본을 동기화 상태로 유지

일반적인 유지 관리를 위해 복제에 사용된 Directory Server를 중지한 경우 서버가 다시 온라인 상태가 될 때 즉시 복제를 통해 업데이트되도록 해야 합니다. 다중 마스터 환경의 마스터인 경우 다중 마스터 집합의 다른 마스터가 디렉토리 정보를 업데이트해야 합니다. 유지 관리를 위해 오프라인 상태로 설정했던 허브 복제본이나 전용 소비자가 다시 온라인 상태가 되면 마스터 복제본이 이를 업데이트해야 합니다.

이 절에서는 복제 재시도 알고리즘 및 다음 재시도를 기다리지 않고 복제 업데이트를 수행하도록 강제하는 방법에 대해 설명합니다.

주 이 절에 설명된 절차는 복제가 이미 설정되어 있으며 소비자를 초기화한 경우에만 사용할 수 있습니다.

복제 재시도 알고리즘

공급자는 소비자를 성공적으로 복제하지 못한 경우 충분한 시간 간격에 따라 정기적으로 복제를 다시 시도합니다. 재시도는 10, 20, 40, 80, 160초 간격에 한 번, 그 다음 300초 간격 5번의 패턴으로 시도됩니다. 공급자는 같은 재시도 패턴을 계속하여 반복합니다.

항상 공급자 복제본과 소비자 복제본의 동기화를 유지하도록 복제 계약을 구성한 경우에도 5분 이상 오프라인 상태였던 복제본을 즉시 업데이트하기는 어렵습니다.

서버가 온라인 상태가 될 때 디렉토리 정보를 즉시 동기화하려면 Directory Server 콘솔 또는 사용자 정의 스크립트를 사용할 수 있습니다.

콘솔에서 복제 업데이트 강제

소비자 또는 다중 마스터 복제 구성의 마스터가 일정 기간 후에 다시 온라인 상태가 될 때 즉시 복제 업데이트를 보내도록 하려면 최신 버전의 디렉토리 데이터가 포함된 공급자에서 다음 단계를 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드와 마스터 복제본의 접미어 노드를 모두 확장한 다음 접미어 아래의 복제 노드를 선택합니다.
오른쪽 패널에 복제본 상태 정보가 표시됩니다.
2. 목록에서 업데이트할 소비자에 해당하는 복제 계약을 선택한 다음 작업 > 지금 업데이트 보내기를 누릅니다.
이렇게 하면 업데이트할 정보가 포함된 복제본에 대한 복제가 시작됩니다.

명령줄에서 복제 업데이트 강제

업데이트할 소비자에서 다음 스크립트를 실행하면 공급자에게 즉시 복제 업데이트를 보내도록 요청합니다. 이 예를 복사하여 `replicate_now.sh`와 같은 의미 있는 이름을 지정합니다. 이 예에 사용된 변수에는 실제 값을 입력해야 합니다.

주 이 스크립트는 오프라인 상태였던 서버가 다시 온라인 상태가 되면 즉시 자동으로 실행되도록 구성할 수 없기 때문에 관리자가 스크립트를 실행해야 합니다.

```
#!/bin/sh
SUP_HOST=supplier_hostname
SUP_PORT=supplier_portnumber
SUP_MGRDN=supplier_directoryManagerDN
SUP_MGRPW=supplier_directoryManagerPassword
MY_HOST=consumer_hostname
MY_PORT=consumer_portnumber

ldapsearch -l -h ${SUP_HOST} -p ${SUP_PORT} -D "${SUP_MGRDN}" \
-w ${SUP_MGRPW} -b "cn=mapping tree, cn=config" \
"(&(objectclass=nsds5replicationagreement) \
(nsDS5ReplicaHost=${MY_HOST}) \
(nsDS5ReplicaPort=${MY_PORT}))" \
dn nsds5ReplicaUpdateSchedule > /tmp/$$

cat /tmp/$$ |
awk
BEGIN { s = 0 }
/^dn: / { print $0;
print "changetype: modify";
print "replace: nsds5ReplicaUpdateSchedule";
print "nsds5ReplicaUpdateSchedule: 0000-2359 0123456";
print "-";
print "";
print $0;
print "changetype: modify";
print "replace: nsds5ReplicaUpdateSchedule";
}
/^nsds5ReplicaUpdateSchedule: / { s = 1; print $0; }
/^[^$]/ {
if ( $s == 1 )
{ print "-" ; print ""; }
else
{ print "nsds5ReplicaUpdateSchedule: 0000-2359 0123456";
```



```

        print "-" ; print " "; };
    s = 0; }
' > /tmp/ldif.$$

echo "Ldif is in /tmp/ldif.$$"
echo

ldapmodify -c -h ${SUP_HOST} -p ${SUP_PORT} -D "${SUP_MGRDN}" \
-w ${SUP_MGRPW} -f /tmp/ldif.$$

```

SSL 연결을 통해 업데이트 작업을 수행하려면 적절한 매개 변수와 값을 사용하여 스크립트의 `ldapmodify` 명령을 수정해야 합니다. 자세한 내용은 [448페이지의 "LDAP 클라이언트에서 보안을 사용하도록 구성"](#)을 참조하십시오.

이전 릴리스를 사용한 복제

이 절에서는 Directory Server의 이전 릴리스를 사용한 복제 구성 방법에 대해 설명합니다.

Directory Server 5.2 및 Directory Server 5.1 간의 복제

Directory Server 5.1과 5.2는 다음과 같은 점을 제외하고 모든 복제 구성에 대해 완전히 호환됩니다.

- 단편 복제는 Directory Server 5.2를 실행하는 마스터 복제와 Directory Server 5.1을 실행하는 소비자 복제 간에 구성되지 않아야 합니다.
- 5.2 공급자와 5.1 소비자 간의 계약을 구성하기 전에 `cn=config`의 `nsslapd-schema-repl-useronly`를 `on`으로 설정해야 합니다. 그렇지 않으면 5.2의 스키마가 5.1로 복제될 때 충돌이 발생합니다. 이 설정을 사용하면 `99user.ldif` 파일에 저장된 사용자 정의 스키마 요소만 복제됩니다. [405페이지의 "스키마 정의 복제"](#)를 참조하십시오.
- Directory Server 5.2의 스키마 파일인 `11rfc2307.ldif`는 RFC 2307에 따라 변경되었으므로 [380페이지의 "Directory Server 5.1 스키마 업데이트"](#)에 설명된 것처럼 5.1 서버의 해당 파일을 업데이트해야 합니다.

- 허브로 수준이 낮춰진 5.2 마스터도 5.1 소비자의 참조 목록에 표시되지만 수준을 낮추는 내부 기법 때문에 수준이 낮춰진 복제본의 포트 번호는 0(제로)이 됩니다. 이 참조 URL은 사용할 수 없으며 대부분의 클라이언트는 이 참조를 따를 수 없을 경우 자동으로 다른 마스터에 대한 참조를 시도합니다. 그러나 이러한 5.1 복제본에 액세스하는 클라이언트에서 참조에 대한 홑수 제한을 늘려야 할 수도 있습니다. 5.2 소비자 복제본은 사용할 수 없는 참조 URL을 표시하거나 수준이 낮춰진 마스터로 반환하지 않습니다.

Directory Server 5.2 및 Directory Server 4.x 간의 복제

Directory Server 5.2는 다음과 같은 조건에 부합될 경우 Directory Server의 4.x 릴리스와의 복제 시나리오에서 호출할 수 있습니다.

- Directory Server 5.2가 마스터로 구성되어 있지만 소비자로서만 Directory Server 4.x 공급자에 복제됩니다.
- 한 개의 소비자 복제본이 레거시 4.x 공급자와 5.2 공급자 모두의 소비자가 될 수는 없습니다. 하지만 5.2 서버에는 각각 레거시 Directory Server에서 공급하는 복제본과 5.2 Directory Server에서 공급하는 복제본 등 여러 개의 복제본이 포함될 수 있습니다.
- 레거시 4.x 공급자의 소비자로 구성된 Directory Server 5.2 복제본은 토폴로지에서 이 접미어의 허브 복제본 역할을 할 수 없습니다.

Directory Server 5.2를 레거시 Directory Server의 소비자로 사용할 때의 가장 큰 이점은 복제된 환경의 이전이 용이하다는 것입니다. 복제 환경을 이전하기 위해 따라야 할 단계에 대한 자세한 내용은 *Directory Server Installation and Migration Guide*를 참조하십시오.

▶ Directory Server 5.2를 Directory Server 4.x의 소비자로 구성하려면

Directory Server 5.2를 Directory Server 4.x 릴리스의 소비자로 사용하려면 다음과 같이 구성해야 합니다.

1. 344페이지의 "마스터 복제본 활성화"에 설명된 것처럼 복제본을 *마스터* 복제본으로 활성화합니다. 이 복제본은 4.x 공급자의 소비자가 되지만 마스터 복제본으로 구성해야 합니다.
2. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드와 복제된 접미어 노드를 모두 확장한 다음 접미어 아래의 복제 노드를 선택합니다.

3. 오른쪽 패널에서 이 복제본에 대해 변경 > 4.x 호환성 활성화를 선택합니다. 또는 객체 메뉴에서 4.x 호환성 활성화를 선택합니다.
4. 4.x 호환성 활성화 창에서 레거시 공급자 서버가 바인드할 때 사용할 바인드 DN 및 비밀번호를 지정합니다. 여기서 지정한 바인드 DN과 비밀번호는 오직 레거시 복제에 대해서만 사용해야 합니다. 따라서 기존 DN이나 5.x 복제에 사용되는 기본 복제 관리자를 사용해서는 안 됩니다.

복제 마법사를 사용하여 레거시 복제를 구성하면 지정한 바인드 DN과 비밀번호가 레거시 복제 구성 항목에 올바르게 저장됩니다. 명령줄에서 수동으로 레거시 복제를 구성하는 경우, `nsslapd-legacy-updatedn` 속성과 `nsslapd-legacy-updatepw` 속성을 사용하여 레거시 복제 구성 항목에 바인드 DN과 비밀번호를 지정해야 합니다.

레거시 복제는 단순 인증에서만 작동하며, 인증서를 사용한 보안 인증에서는 작동하지 않습니다.

5. 확인을 누릅니다. 이제 소비자 복제본에서 레거시 공급자의 업데이트를 받을 수 있습니다.
6. 4.x 마스터에서 복제할 내용에 사용된 모든 속성 및 객체 클래스가 5.2 복제본 서버의 스키마에 정의되어 있는지 확인합니다.
7. 4.x 마스터에서 작성된 LDIF 복제본 파일을 가져와서 5.2 복제본을 초기화합니다. 이 파일의 첫 항목에는 4.x 복제 기법에 필요한 `copiedfrom` 속성이 포함되어 있습니다.

서버에서 4.x 호환성을 활성화하면 기본적으로 설치된 레거시 복제 플러그 인이 구성됩니다. 이 플러그 인은 레거시 공급자의 업데이트를 처리하고 복제된 접미어 내용에 대한 업데이트를 수행합니다.

주 4.x 호환성을 활성화하면 이 복제본은 클라이언트의 모든 수정 요청에 대해 참조를 반환합니다. Directory Server 5.2는 마스터 복제본으로 구성되어 있지만 이 접미어에 대한 수정 요청을 수행하지 않고 4.x 공급자 서버에 대한 참조를 반환합니다.

레거시 복제 설정을 완료하려면 이제 레거시 공급자가 5.2 Directory Server로 복제되도록 구성해야 합니다. 4.x Directory Server에서 복제 계약을 구성하는 방법은 레거시 Directory Server와 함께 제공된 설명서를 참조하십시오.

Directory Server 5.1 스키마 업데이트

Directory Server 5.2의 스키마 파일인 11rfc2307.ldif는 RFC 2307에 따라 변경되었습니다(<http://www.ietf.org/rfc/rfc2307.txt> 참조). 5.2 서버와 5.1 서버 간의 복제를 구성 또는 활성화하려면 먼저 5.1 서버에서 스키마를 업데이트해야 합니다. 스키마 파일은 두 서버 버전에서 모두 `ServerRoot/slapd-serverID/config/schema/`에 위치해 있습니다.

▶ Directory Server 5.1 스키마를 업데이트하려면

- 5.2 서버의 11rfc2307.ldif 파일을 5.1 서버에 복사합니다.
 - 5.1 서버의 Solaris 패키지가 설치되어 있는 경우 이전의 10rfc2307.ldif 파일도 삭제해야 합니다.
 - 기타 플랫폼에 5.1 서버의 zip 파일이 설치되어 있는 경우에는 기존 11rfc2307.ldif 파일을 덮어씁니다.
- 다음과 같은 스키마 파일도 이 변경에 의해 영향을 받으므로 5.2 서버에서 복사하여 5.1 서버의 기존 파일을 덮어써야 합니다.
 - 20subscriber.ldif
 - 30ns-common.ldif
 - 50ns-admin.ldif
 - 50ns-certificate.ldif
 - 50ns-directory.ldif
 - 50ns-legacy.ldif
 - 50ns-mail.ldif
 - 50ns-mlm.ldif
 - 50ns-msg.ldif
 - 50ns-netshare.ldif
- 5.1 서버를 다시 시작하여 복제 구성 및 복제본 초기화를 계속합니다. 다른 스키마 요소와 동기화될 때 서버 간에 복제되는 스키마 속성도 있지만 이것은 복제 기법의 정상적인 동작입니다.
- 이전 버전의 스키마를 사용하는 모든 응용 프로그램을 업데이트해야 할 수도 있습니다. 새로운 11rfc2307.ldif 파일에서는 다음과 같은 사항이 수정되었습니다.
 - automount 속성과 automountInformation 속성이 제거되었습니다.

- ipHost 객체 클래스의 허용되는 속성 목록에 `o $ ou $ owner $ seeAlso $ serialNumber`가 포함되지 않습니다.
- ieee802Device 객체 클래스의 필수 속성 목록에 `cn`이 포함되지 않습니다.
- ieee802Device 객체 클래스의 허용되는 속성 목록에 `description $ 1 $ o $ ou $ owner $ seeAlso $ serialNumber`가 포함되지 않습니다.
- bootableDevice 객체 클래스의 필수 속성 목록에 `cn`이 포함되지 않습니다.
- bootableDevice 객체 클래스의 허용되는 속성 목록에 `description $ 1 $ o $ ou $ owner $ seeAlso $ serialNumber`가 포함되지 않습니다.
- nisMap 객체 클래스의 OID가 1.3.6.1.1.1.2.9로 변경되었습니다.

레트로 변경 로그 플러그인 사용

레트로 변경 로그 플러그인 설명

레트로 변경 로그는 LDAP 클라이언트에서 사용하는 플러그인으로서, Directory Server 4.x 버전과 응용 프로그램 호환성을 유지하는 데 사용됩니다. 레트로 변경 로그는 Directory Server 변경 로그와는 다른 별도의 데이터베이스에서 `cn=changelog` 접미어 아래에 저장됩니다.

레트로 변경 로그는 독립형 서버나 복제 토폴로지의 각 서버에서 사용할 수 있습니다. 레트로 변경 로그가 서버에서 사용하도록 활성화되어 있는 경우에는 기본적으로 해당 서버의 모든 접미어에 대한 업데이트가 기록됩니다. 레트로 변경 로그는 지정된 접미어의 업데이트만 로그하도록 구성할 수 있습니다.

레트로 변경 로그 추가 정보

복제 토폴로지에서 레트로 변경 로그 플러그인을 사용하는 방법에 대한 자세한 내용은 *Directory Server Deployment Planning Guide*에서 "Replication and the Retro Change Log"를 참조하십시오.

레트로 변경 로그의 항목 속성에 대한 자세한 내용은 *Directory Server Administration Reference*의 "Object Class Reference"에서 `changeLogEntry`를 참조하십시오.

레트로 변경 로그의 속성에 대한 자세한 내용은 *Directory Server Administration Reference*의 "Attribute Reference"를 참조하십시오.

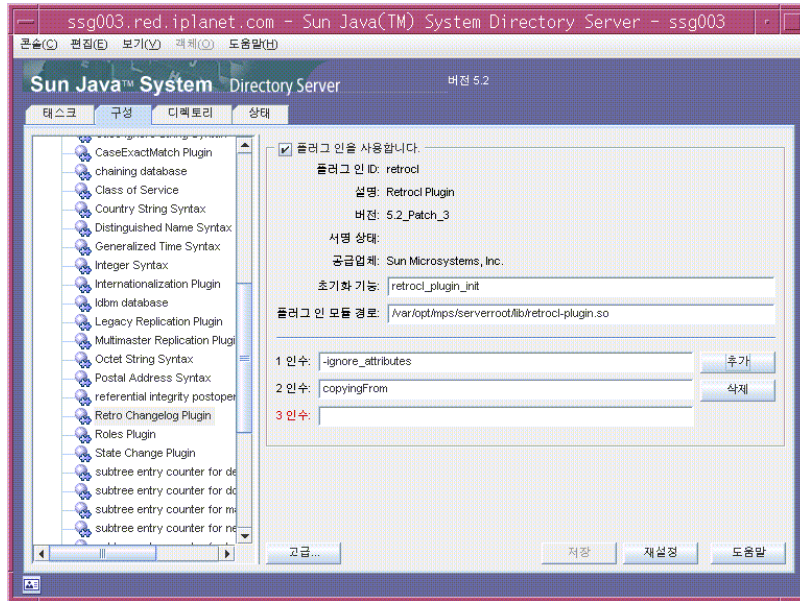
레트로 변경 로그 플러그인 활성화

레트로 변경 로그 플러그인의 구성 정보는 `dse.ldif`의 `cn=Retro Changelog Plugin,cn=plugins,cn=config` 항목에 저장됩니다.

▶ Directory Server 콘솔을 사용하여 레트로 변경 로그 플러그인을 활성화하려면

1. Directory Server 콘솔의 최상위 구성 탭에서 플러그인 노드를 확장한 다음 아래로 스크롤하여 레트로 변경 로그 플러그인을 선택합니다.
2. 오른쪽 패널에서 "플러그인 활성화" 확인란을 선택하고 저장을 누릅니다. 플러그인을 비활성화하려면 이 확인란을 선택 취소하고 저장을 누릅니다.

다음 그림은 레트로 변경 로그 플러그인 구성 화면을 나타냅니다.



3. 플러그인을 활성화 하든 비활성화한 후 Directory Server를 다시 시작합니다.

▶ 명령줄을 사용하여 레트로 변경 로그 플러그인을 활성화하려면

1. 아래 명령을 실행하여 레트로 변경 로그 플러그인의 구성 항목을 수정합니다.


```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=Retro Changelog Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginenabled
nsslapd-pluginenabled: on
^D
```
2. 서버를 다시 시작합니다. 자세한 내용은 35페이지의 "[Directory Server 시작 및 중지](#)"를 참조하십시오.

지정된 접미어에 업데이트를 기록하도록 레트로 변경 로그 구성

서버에 레트로 변경 로그가 활성화되어 있을 때는 기본적으로 레트로 변경 로그는 서버의 모든 접미어에 대한 업데이트를 기록합니다. 이 절에서는 지정된 접미어의 업데이트만 기록하도록 레트로 변경 로그를 구성하는 방법을 설명합니다.

▶ **Directory Server** 콘솔을 사용하여 지정된 접미어에 대한 업데이트를 기록하도록 레트로 변경 로그를 구성하려면

1. 382페이지의 "[Directory Server 콘솔을 사용하여 레트로 변경 로그 플러그인을 활성화하려면](#)"에 설명된 대로 레트로 변경 로그 플러그인을 활성화합니다.
2. 화면의 오른쪽에서 추가를 누릅니다.

Argument라는 이름의 새 필드가 나타납니다.
3. 다음 구문을 사용하여 기록하려는 접미어의 이름을 입력합니다.


```
suffixes="suffix1", "suffix2"
```
4. 저장을 누릅니다.
5. 35페이지의 "[Directory Server 시작 및 중지](#)"에 설명된 것처럼 서버를 다시 시작합니다.

▶ 명령줄을 사용하여 지정된 접미어에 대한 업데이트를 기록하도록 레트로 변경 로그를 구성하려면

1. 아래 명령을 실행하여 레트로 변경 로그 플러그인의 구성 항목을 수정합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password  
dn: cn=Retro Changelog Plugin,cn=plugins,cn=config  
changetype: modify  
add: nsslapd-pluginarg2  
nsslapd-pluginarg2: suffixes="suffix1", "suffix2"  
^D
```

2. 서버를 다시 시작합니다. 자세한 내용은 35페이지의 "Directory Server 시작 및 중지"를 참조하십시오.

삭제된 항목의 속성을 기록하도록 레트로 변경 로그 구성

이 절에서는 항목이 삭제될 때 해당 항목의 지정된 속성을 기록하도록 레트로 변경 로그를 구성하는 방법을 설명합니다.

▶ Directory Server 콘솔을 사용하여 삭제된 항목의 속성을 기록하도록 레트로 변경 로그를 구성하려면

1. 382페이지의 "Directory Server 콘솔을 사용하여 레트로 변경 로그 플러그인을 활성화 하려면"에 설명된 대로 레트로 변경 로그 플러그인을 활성화합니다.

2. 화면의 오른쪽에서 Add를 누릅니다.

Argument라는 이름의 새 필드가 나타납니다.

3. 다음 구문을 사용하여 기록하려는 속성의 이름을 입력합니다.

```
deletedentryattributes=description,givenname
```

4. 저장을 누릅니다.

5. 35페이지의 "Directory Server 시작 및 중지"에 설명된 것처럼 서버를 다시 시작합니다.

- ▶ 명령줄을 사용하여 삭제된 항목의 속성을 기록하려면 레트로 변경 로그를 구성하려면
1. 아래 명령을 실행하여 레트로 변경 로그 플러그인의 구성 항목을 수정합니다.


```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=Retro Changelog Plugin,cn=plugins,cn=config
changetype: modify
add: nsslapd-pluginarg3
nsslapd-pluginarg3: deletedEntryAttributes=attribute1,attribute2
^D
```
 2. 서버를 다시 시작합니다. 자세한 내용은 35페이지의 "Directory Server 시작 및 중지"를 참조하십시오.

레트로 변경 로그 지우기

변경 로그 항목은 지정된 기간 후에 자동으로 제거할 수 있습니다. 변경 로그의 항목이 자동으로 삭제되기까지의 기간을 구성하려면 `cn=Retro Changelog Plugin`, `cn=plugins`, `cn=config` 항목의 `nsslapd-changelogmaxage` 구성 속성을 설정해야 합니다. 이 속성은 다음과 같이 명령줄에서만 설정할 수 있습니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=Retro Changelog Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-changelogmaxage
nsslapd-changelogmaxage: IntegerTimeunit
^D
```

여기서 *Integer*는 숫자를 나타내고 *Timeunit*은 s(초), m(분), h(시간), d(일) 또는 w(주) 중 하나로 지정될 수 있습니다. *Integer* 변수와 *Timeunit* 변수 사이에는 공백을 사용할 수 없습니다. 예를 들면 다음과 같습니다.

```
nsslapd-changelogmaxage: 2d
```

레트로 변경 로그는 변경 로그에 대한 다음 작업을 수행할 때 지워집니다.

레트로 변경 로그 액세스

변경 로그는 검색 작업을 지원하며, 다음과 같은 형식의 필터를 사용한 검색에 최적화되어 있습니다.

```
(&(changeNumber>=X) (changeNumber<=Y))
```

변경 로그 크기를 줄이기 위해 항목을 삭제할 수는 있지만 일반적으로 레트로 변경 로그 항목에 대한 추가 또는 수정 작업은 허용되지 않습니다. 기본 액세스 제어 정책을 수정하는 경우에만 레트로 변경 로그에 대한 수정 작업을 수행해야 합니다.

레트로 변경 로그가 작성되면 기본적으로 다음과 같은 액세스 제어 규칙이 적용됩니다.

- 레트로 변경 로그 상위 항목인 cn=changelog에 대해 인증된 모든 사용자 (userdn=all인 익명 액세스가 아닌 userdn=anyone)에게 읽기, 검색 및 비교 권한이 부여됩니다.
- 디렉토리 관리자에게 암묵적으로 부여되는 경우를 제외하고 쓰기 및 삭제 액세스 권한은 부여되지 않습니다.

변경 로그 항목에는 비밀번호와 같은 중요한 정보의 수정 사항이 포함될 수 있으므로 익명 사용자에게 읽기 액세스 권한을 부여해서는 안 됩니다. 인증된 사용자도 레트로 변경 로그 내용을 볼 수 없게 하려면 이 로그 내용에 대한 액세스를 추가로 제한할 수 있습니다.

레트로 변경 로그에 적용되는 기본 액세스 제어 정책을 수정하려면 cn=changelog 항목의 aci 속성을 수정해야 합니다. aci 속성 설정에 대한 자세한 내용은 [6장, "액세스 제어 관리"](#)를 참조하십시오.

복제 상태 모니터

새로운 명령줄 도구와 Directory Server 콘솔을 사용하여 복제 상태를 모니터할 수 있습니다.

명령줄 도구

다음 세 개의 명령줄 도구를 사용하여 복제 배포를 모니터할 수 있습니다.

- repldisc - 복제 배포 시 알려진 모든 서버를 "찾아서" 테이블을 구성합니다.
- insync - 공급자와 하나 이상의 소비자 복제본 간의 동기화 상태를 나타냅니다.
- entrycmp - 두 개 이상의 복제본에 있는 동일 항목을 비교합니다.

위의 세 도구는 아래 디렉토리에 위치해 있습니다.

```
ServerRoot/shared/bin
```

*Directory Server Man Page Reference*는 이러한 도구의 사용 예와 전체 명령줄 구문을 제공합니다.

복제 상태 탭

Directory Server 콘솔에서 복제 상태 요약을 보려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 상태 탭에서 복제 노드를 선택합니다.
오른쪽 패널에는 이 서버에 구성된 각 복제 계약에 대한 정보가 포함된 테이블이 표시됩니다.
2. 복제 상태를 모니터링하려면 계속 갱신 확인란을 선택합니다. 예를 들어, 언제 복제본의 초기화가 끝났는지 확인할 수 있습니다.
3. 아직 소비자에 복제되지 않은 마스터의 최종 수정 사항을 확인하려면 보류 중인 변경 번호 버튼을 누릅니다. 이 작업에 오랜 시간이 소요될 수 있다는 경고와 확인하라는 메시지가 표시됩니다. 보류 중인 변경 번호를 확인하려면 소비자의 업데이트 레코드를 다운로드하여 이를 마스터의 변경 로그와 비교해야 합니다. 로그가 클 경우 이 작업에 많은 시간과 서버 자원이 소요될 수 있습니다.
4. 열 머리글을 누르고 크기를 조정하여 테이블 레이아웃을 수정할 수 있습니다. 보기 옵션 버튼을 누르고 표시하려면 열만 선택하여 테이블 내용을 수정할 수도 있습니다. [표 8-1](#)에는 서버의 각 계약에 대해 테이블에 표시할 수 있는 복제 매개 변수가 나와 있습니다.

표 8-1 Directory Server 콘솔 상태 탭의 복제 매개 변수

테이블 머리글	설명
접미어	복제되는 접미어 또는 하위 접미어의 이름을 지정합니다.
원격 복제본	소비자 서버의 호스트 이름 및 포트가 포함됩니다.
설명	이 복제 계약에서 제공하는 설명 문자열이 포함됩니다.
상태	계약이 비활성화되었는지, 소비자를 초기화하는 중인지 또는 증분 업데이트를 통해 정상적으로 복제하는지 여부를 나타냅니다.

표 8-1 Directory Server 콘솔 상태 탭의 복제 매개 변수(계속)

테이블 머리글	설명
요약	최신 이벤트(초기화 또는 업데이트의 시작이나 끝)와 마지막으로 수신한 메시지가 포함됩니다.
전송된 업데이트	복제를 활성화하거나 서버를 다시 시작한 후 소비자에게 전송된 개별 업데이트의 총 수를 누적합니다.
시작된 마지막 업데이트	최근 복제 업데이트가 시작된 시기를 나타냅니다.
종료된 마지막 업데이트	최근 복제 업데이트가 종료된 시기를 나타냅니다.
마지막 업데이트 메시지	최근 복제 업데이트의 상태를 제공합니다.
마지막 초기화 메시지	소비자에 대한 마지막 초기화 상태를 제공합니다.
시작된 마지막 초기화	소비자 복제본에 대한 최근 초기화가 시작된 시기를 나타냅니다.
종료된 마지막 초기화	소비자 복제본에 대한 최근 초기화가 종료된 시기를 나타냅니다.

일반적인 복제 충돌 해결

다중 마스터 복제 시에는 일관성이 낮은 복제 모델이 사용되므로 여러 서버에서 동일한 항목을 동시에 수정할 수 있습니다. 따라서 두 서버 간에 업데이트를 전송하는 경우 충돌하는 변경 사항을 해결해야 합니다. 대체로 이러한 충돌은 각 서버의 변경에 연결된 타임스탬프에 따라 자동으로 해결되며 최근 변경 사항이 우선합니다.

하지만 변경 충돌을 해결하기 위해 수동 작업이 필요한 경우도 있습니다. 복제 프로세스에서 자동으로 해결할 수 없는 변경 충돌이 있는 항목에는 작동 가능 속성인 `nsds5ReplConflict`가 충돌 표식으로 포함되어 있습니다.

충돌 문제가 있는 항목을 찾으려면 이 속성이 포함된 항목을 정기적으로 검색합니다. 예를 들어 아래의 `ldapsearch` 명령을 사용할 수 있습니다.

```
% ldapsearch -h host -p port -D "cn=Directory Manager" -w password \
-b "dc=example,dc=com" "(nsds5ReplConflict=*)"
```

`nsds5ReplConflict` 속성은 기본적으로 색인화됩니다.

이름 지정 충돌 해결

서버가 변경 사항을 서로에게 복제하기 전에 항목이 작성되면 동일한 DN을 가진 항목이 각각의 마스터에 존재할 수 있습니다. 복제 시 충돌 해결 기법은 두 번째로 작성된 항목의 이름을 자동으로 바꿉니다.

DN 이름이 충돌하는 항목은 작동 가능 속성인 `nsuniqueid`로 지정된 고유 식별자가 해당 DN에 추가된 새 이름으로 바뀝니다. 예를 들어, `uid=bjensen,ou=People,dc=example,dc=com` 항목을 두 마스터에서 동시에 작성하면 복제 후에 다음 두 항목이 두 서버에 모두 포함됩니다.

- `uid=bjensen,ou=People,dc=example,dc=com`
- `nsuniqueid=66446001-1dd211b2+uid=bjensen,dc=example,dc=com`

두 번째 항목은 고유 DN을 갖도록 이름을 바꿔야 합니다. 충돌하는 항목을 삭제하고 충돌하지 않는 이름을 사용하여 항목을 다시 추가할 수도 있지만 항목을 처음 작성된 상태로 유지하는 가장 확실한 방법은 이름을 바꾸는 것입니다. 이름 변경 절차는 이름 지정 속성이 한 개 값을 갖는지 또는 여러 값을 갖는지에 따라 달라집니다. 각 절차에 대해서는 아래에서 설명합니다.

여러 값을 갖는 이름 지정 속성이 있는 항목의 이름 변경

여러 값을 갖는 이름 지정 속성이 있는 충돌 항목의 이름을 바꾸려면 다음을 수행합니다.

1. 이전 RDN 값을 유지하여 항목의 이름을 바꿉니다. 예를 들면 다음과 같습니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: nsuniqueid=66446001-1dd211b2+uid=bjensen,dc=example,dc=com
changetype: modrdn
newrdn: uid=NewValue
deleteoldrdn: 0
^D
```

삭제할 수 없는 작동 가능 속성인 `nsuniqueid`도 포함되어 있으므로 이 단계에서는 이전 RDN 값을 삭제할 수 없습니다.

2. 이름 지정 속성의 이전 RDN 값과 충돌 표시 속성을 제거합니다. 예를 들면 다음과 같습니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=NewValue,dc=example,dc=com
changetype: modify
delete: uid
uid: bjensen
-
delete: nsds5ReplConflict
^D
```

한 개의 값을 갖는 이름 지정 속성이 있는 항목의 이름 변경

이름 지정 속성이 한 개의 값을 가지면(예: dc) 항목의 이름을 단순히 동일한 속성의 다른 값으로 바꿀 수 없으므로 임시 이름을 지정해야 합니다.

1. 다른 이름 지정 속성을 사용하여 항목의 이름을 바꾸고 이전 RDN을 유지합니다. 예를 들면 다음과 같습니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: nsuniqueid=66446001-1dd211b2+dc=HR,dc=example,dc=com
changetype: modrdn
newrdn: o=TempName
deleteoldrdn: 0
^D
```

삭제할 수 없는 작동 가능 속성인 nsuniqueid도 포함되어 있으므로 이 단계에서는 이전 RDN 값을 삭제할 수 없습니다.

2. 원하는 이름 지정 속성을 고유한 값으로 변경하고 충돌 표식 속성을 제거합니다. 예를 들면 다음과 같습니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: o=TempName,dc=example,dc=com
changetype: modify
replace: dc
dc: uniqueValue
-
delete: nsds5ReplConflict
^D
```

3. 항목의 이름을 다시 해당 이름 지정 속성으로 바꿉니다. 예를 들면 다음과 같습니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: o=TempName,dc=example,dc=com
changetype: modrdn
newrdn: dc=uniqueValue
deleteoldrdn: 1
^D
```

deleteoldrdn 속성 값을 1로 설정하여 임시 속성 값 쌍인 o=*TempName*을 삭제합니다. 이 속성을 유지하려면 deleteoldrdn 속성 값을 0으로 설정할 수 있습니다.

고아 항목 충돌 해결

삭제 작업을 복제할 때 소비자 서버에서 삭제할 항목에 자식 항목이 있음을 발견하면 충돌 해결 절차는 연결 항목을 작성하여 디렉토리에 고아 항목이 발생하지 않도록 방지합니다.

이와 마찬가지로 추가 작업을 복제할 때 소비자 서버에서 부모 항목을 찾을 수 없으면 충돌 해결 절차는 새 항목이 고아 항목이 되지 않도록 부모가 될 연결 항목을 작성합니다.

연결 항목은 glue 객체 클래스와 extensibleObject 객체 클래스를 포함하는 임시 항목으로, 다음과 같은 여러 가지 방법으로 작성할 수 있습니다.

- 충돌 해결 절차에서 일치하는 고유 식별자를 가진 삭제된 항목을 발견한 경우에는 해당 항목에 glue 객체 클래스와 nsds5ReplConflict 속성이 추가되어 연결 항목을 작성합니다.

이 경우 정상적인 항목으로 유지하기 위해 연결 항목을 수정하여 glue 객체 클래스와 nsds5ReplConflict 속성을 제거하거나 연결 항목 및 해당 자식 항목을 삭제할 수 있습니다.

- 서버는 glue 객체 클래스와 extensibleObject 객체 클래스가 있는 최소 항목을 작성합니다.

이 경우 항목을 수정하여 의미 있는 항목으로 설정하거나 항목 및 해당 자식 항목을 모두 삭제해야 합니다.

잠재적 상호 운용성 문제 해결

우편 서버처럼 속성 고유성에 의존하는 응용 프로그램과의 상호 운용성을 위해 nsds5ReplConflict 속성이 포함된 항목에 대한 액세스를 제한해야 할 수도 있습니다. 이러한 항목에 대한 액세스를 제한하지 않으면 한 개의 속성만 필요한 응용 프로그램이 nsds5ReplConflict가 포함된 충돌 해결 항목과 원래 항목을 모두 받게 되므로 작업이 실패합니다.

액세스를 제한하려면 아래 명령을 실행하여 익명 읽기 액세스를 부여하는 기본 ACI를 수정해야 합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: dc=example,dc=com
changetype: modify
delete: aci
aci: (target ="ldap:///dc=example,dc=com")
      (targetattr !="userPassword"
      (version 3.0;acl "Anonymous read-search access";
```

```
allow (read, search, compare)(userdn = "ldap:///anyone");)
-
add: aci
aci: (target="ldap:///dc=example,dc=com")
(targetattr!="userPassword")
(targetfilter="(!(nsds5ReplConflict=*))") (version 3.0;acl
"Anonymous read-search access";allow (read, search, compare)
(userdn="ldap:///anyone");)
^D
```

새 ACI는 `nsds5ReplConflict` 속성을 포함한 항목이 검색 결과로 반환되지 않도록 필터링합니다.

디렉토리 스키마 확장

Directory Server에는 수백 개의 객체 클래스 및 속성이 포함된 표준 스키마가 미리 구성되어 있습니다. 표준 객체 클래스 및 속성만으로도 대부분의 요구 사항을 충족시킬 수 있지만, 새로운 객체 클래스 및 속성을 작성하여 스키마를 확장해야 하는 경우도 있습니다. 표준 스키마의 개요와 배포에 적합한 스키마 설계에 대한 지침은 *Directory Server Deployment Planning Guide*를 참조하십시오.

이 장의 다음 절에서는 스키마 확장 방법을 설명합니다.

- 스키마 검사
- 스키마 확장에 대한 개요
- 속성 정의 관리
- 객체 클래스 정의 관리
- 스키마 정의 복제

스키마 검사

스키마 검사를 활성화하면 Directory Server는 모든 가져오기, 추가 및 수정 작업이 현재 정의된 디렉토리 스키마에 맞는지 확인합니다.

- 각 항목의 객체 클래스와 속성이 스키마에 맞는지 여부
- 정의된 모든 객체 클래스에 필요한 속성이 항목에 모두 포함되어 있는지 여부
- 객체 클래스에서 허용하는 속성만 항목에 포함되어 있는지 여부

주 항목을 수정하면 **Directory Server**는 수정되는 항목만이 아닌 전체 항목에 대해 스키마 검사를 수행합니다. 따라서 항목의 객체 클래스 또는 속성이 스키마에 맞지 않으면 작업이 실패할 수 있습니다. 하지만 스키마 검사는 구분과 관련하여 속성 값의 유효성을 검사하지는 않습니다.

스키마 검사는 기본적으로 활성화되며, **Directory Server**를 실행할 때는 항상 스키마 검사를 사용해야 합니다. 대부분의 클라이언트 응용 프로그램은 스키마 검사를 활성화하면 모든 항목이 스키마에 맞을 것이라고 가정합니다. 하지만 스키마 검사를 활성화해도 디렉토리에 있는 기존 항목은 확인되지 않습니다. 모든 디렉토리 내용이 스키마에 맞도록 하려면 첫 항목을 추가하기 전이나 모든 항목을 다시 초기화하기 전에 스키마 검사를 활성화해야 합니다.

스키마에 맞는 **LDAP** 파일의 가져오기 작업 속도를 향상시키기 위해 예외적으로 스키마 검사를 비활성화할 수도 있습니다. 하지만 스키마에 맞지 않는 항목을 가져오게 될 위험이 있으며 이러한 오류를 감지할 수 없습니다.

항목이 스키마에 맞지 않으면 해당 항목을 검색할 수 없고 이 항목에 대한 수정 작업이 실패할 수 있습니다. 스키마에 맞도록 항목을 수정하려면 다음을 수행해야 합니다.

1. 생산 환경의 서버에서는 먼저 전체 서버를 읽기 전용으로 설정하여 스키마 검사를 비활성화한 동안 수정을 방지하는 것이 좋습니다. [51페이지의 "전역 읽기 전용 모드 설정"](#)을 참조하십시오.
2. 아래에 설명된 것처럼 스키마 검사를 비활성화합니다.
3. 항목을 검색한 다음 현재 정의된 스키마와 수동으로 비교하여 항목이 스키마에 맞지 않는 이유를 확인합니다. [398페이지의 "속성 보기"](#) 및 [402페이지의 "객체 클래스 보기"](#)를 참조하십시오.
4. 스키마에 맞도록 항목을 수정합니다.
맞지 않는 항목이 많으며, 이러한 항목이 새로운 데이터 형식이나 패턴을 나타낼 경우 스키마를 수정할 수도 있습니다. 하지만 배포 전에 스키마를 계획하여 스키마에 대한 변경을 최소화해야 합니다. 자세한 내용은 *Directory Server Deployment Planning Guide*를 참조하십시오.
5. 아래에 설명된 것처럼 스키마 검사를 활성화합니다.
6. 전역 읽기 전용 모드를 활성화한 경우 이를 해제합니다.

콘솔에서 스키마 검사 설정

1. Directory Server 콘솔의 최상위 구성 탭에 있는 구성 트리에서 스키마 노드를 선택합니다.

오른쪽 패널에는 스키마 정의가 포함되어 있습니다.

2. 패널 맨 위의 상태 메시지에 현재 스키마 검사의 활성화 여부가 표시됩니다. 오른쪽에 있는 버튼을 누르면 스키마 검사가 비활성화 또는 활성화됩니다.
 - 버튼 레이블이 비활성화이면 스키마 검사를 비활성화할 수 있습니다.
 - 버튼 레이블이 활성화이면 스키마 검사를 활성화할 수 있습니다.

새 스키마 검사 정책이 즉시 적용됩니다.

명령줄에서 스키마 검사 설정

다음과 같이 `cn=config` 항목의 `nsslapd-schemacheck` 속성을 설정하여 스키마 검사를 활성화 및 비활성화할 수도 있습니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=config
changetype: modify
replace: nsslapd-schemacheck
nsslapd-schemacheck: on or off
^D
```

서버는 즉시 새 스키마 검사 정책을 실행합니다.

스키마 확장에 대한 개요

스키마에 새 속성을 추가하는 경우 이 속성이 포함될 새로운 객체 클래스를 작성해야 합니다. 필요한 대부분의 속성이 포함되어 있는 기존의 객체 클래스에 속성을 추가하는 것이 더 편리할 수도 있지만 LDAP 클라이언트와의 상호 운용성이 저하되는 단점이 있습니다.

Directory Server와 기존 LDAP 클라이언트와의 상호 운용성은 표준 LDAP 스키마에 기반을 두고 있으므로 표준 스키마를 변경하면 서버를 업그레이드할 때 문제가 발생합니다. 표준 스키마 요소를 삭제할 수 없는 것도 이 때문입니다.

객체 클래스와 속성 및 디렉토리 스키마에 대한 자세한 내용과 스키마 확장에 대한 지침은 *Directory Server Deployment Planning Guide*를 참조하십시오. 표준 속성과 객체 클래스에 대한 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.

Directory Server 스키마는 cn=schema 항목의 속성에 저장됩니다. 구성 항목과 마찬가지로 이 항목은 서버 시작 중에 파일에서 읽은 스키마의 LDAP 뷰입니다. 스키마 파일은 아래 위치에 있는 LDIF 파일입니다.

`ServerRoot/slapd-serverID/config/schema`

이 디렉토리에는 Directory Server 및 Directory Server에 기반을 둔 다른 Sun Java System 서버에서 사용하는 표준 스키마 파일이 포함되어 있습니다. 이러한 파일 및 표준 스키마는 *Directory Server Administration Reference*에 설명되어 있습니다.

스키마 파일 수정

서버는 시작 시에만 한 번 스키마 파일을 읽습니다. 파일의 LDIF 내용은 cn=schema에 있는 스키마의 메모리 내장 LDAP 뷰에 추가됩니다. 스키마 정의의 순서가 중요하기 때문에 스키마 파일 이름 앞에는 번호가 붙으며 영숫자순으로 로드됩니다. 이 디렉토리에 있는 스키마 파일은 설치 중에 정의된 시스템 사용자만 쓸 수 있습니다.

파일의 스키마 정의를 수정하려면 원하는 파일을 작성 또는 수정한 다음 서버를 다시 시작해야 합니다. 스키마 파일의 정의 구문에 대해서는

RFC 2252(<http://www.ietf.org/rfc/rfc2252.txt>)에서 설명합니다.

LDIF 파일에 스키마를 직접 정의하는 경우 x-ORIGIN 필드에 'user defined' 값을 사용해서는 안 됩니다. 이 값은 cn=schema의 LDAP 뷰를 통해 정의되며 99user.ldif 파일에 표시되는 스키마 요소에 예약된 값입니다.

99user.ldif 파일에는 cn=schema 항목에 대한 추가 ACI 및 명령줄이나 콘솔에서 추가된 모든 스키마 정의가 포함됩니다. 새 스키마 정의가 추가되면 99user.ldif 파일을 덮어쓰기 때문에 이 파일을 수정하려는 경우에는 즉시 서버를 다시 시작하여 변경 사항을 영구 저장해야 합니다.

다른 스키마 파일에 정의된 표준 스키마를 수정할 수는 없지만 새 파일을 추가하여 새 속성과 객체 클래스를 정의할 수 있습니다. 예를 들어, 여러 서버에 새 스키마 요소를 정의하려면 98mySchema.ldif 파일에 스키마 요소를 정의한 다음 이 파일을 모든 서버의 스키마 디렉토리에 복사할 수 있습니다. 그런 후에 모든 서버를 다시 시작하여 새 스키마 파일을 로드해야 합니다.

명령줄에서 스키마 수정

스키마는 `cn=schema`에 있는 LDAP 뷰에서 정의되기 때문에 `ldapsearch` 및 `ldapmodify` 유틸리티를 사용하여 온라인으로 스키마를 보고 수정할 수 있습니다. 하지만 `X-ORIGIN` 필드에 'user defined' 값이 있는 스키마 요소만 수정할 수 있습니다. 다른 정의에 대한 수정은 서버에서 모두 거부합니다.

`attributeTypes` 속성과 `objectClasses` 속성의 개별 값을 추가 및 삭제하려면 `ldapmodify`를 사용합니다. 두 속성은 여러 값을 가지므로 값 중 하나를 수정하려면 특정 값을 삭제한 다음 새 값으로 추가해야 합니다(86페이지의 "여러 값을 갖는 속성의 값 하나만 수정" 참조). 또한 RFC 2252(<http://www.ietf.org/rfc/rfc2252.txt>)에 설명된 스키마 요소 정의 구문을 사용해야 합니다.

사용자 정의 요소에 대한 변경 사항과 새로운 요소 정의는 모두 `99user.ldif` 파일에 저장됩니다.

명령줄에서 스키마 정의를 수정하는 경우 긴 값을 정확하게 입력해야 하기 때문에 오류가 발생할 가능성이 큽니다. 하지만 디렉토리 스키마를 업데이트해야 하는 스크립트에 이 기능을 사용할 수 있습니다.

콘솔에서 스키마 수정

디렉토리 스키마를 사용자 정의하려면 다음 절에 설명된 **Directory Server** 콘솔 인터페이스를 사용하는 것이 좋습니다. 콘솔을 사용하여 표준 스키마를 볼 수 있으며, 새 속성과 객체 클래스를 정의하고 정의한 요소를 편집할 수 있는 그래픽 인터페이스를 제공합니다.

사용자 정의 요소에 대한 변경 사항과 새로운 요소 정의는 모두 `99user.ldif` 파일에 저장됩니다.

디렉토리 스키마를 확장하려면 다음과 같은 순서로 진행해야 합니다.

1. 400페이지의 "속성 작성"에 설명된 것처럼 먼저 새 속성을 작성합니다.
2. 그런 후에 새 속성이 포함될 객체 클래스를 작성하여 속성을 추가합니다. 자세한 내용은 403페이지의 "객체 클래스 작성"을 참조하십시오.

속성 정의 관리

Directory Server 콘솔은 스키마의 모든 속성을 볼 수 있으며 자신의 속성 정의를 작성, 편집 및 삭제할 수 있는 인터페이스를 제공합니다.

속성 보기

현재 디렉토리 스키마에 있는 모든 속성에 대한 정보를 보려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에 있는 구성 트리에서 스키마 노드를 선택한 다음 오른쪽 패널에서 속성 탭을 선택합니다.

이 탭에는 스키마에 있는 모든 표준(읽기 전용) 속성 및 사용자 정의 속성 테이블이 포함되어 있습니다. 테이블에서 개별 줄 위에 마우스를 놓으면 해당 속성에 대한 설명 문자열이 표시됩니다.

아래 표에는 속성 테이블의 필드가 나와 있습니다.

표 9-1 속성 탭의 테이블 열

열 머리글	설명
이름	속성 이름, 속성 유형이라고도 합니다.
OID	속성의 객체 식별자. OID는 스키마 객체를 고유하게 식별하는 문자열이며 대체로 점으로 구분된 10진수로 구성됩니다. OID에 대한 자세한 내용을 보거나 기업용 접두어를 요청하려면 iana@iana.org 로 IANA(Internet Assigned Number Authority)에 전자 메일을 보내거나 IANA 웹 사이트(http://www.iana.org/)를 방문하십시오.
구문	구문 열에서는 이 속성에 허용되는 값의 형식에 대해 설명합니다. 사용할 수 있는 구문은 398페이지의 표 9-2 에 나와 있습니다.
여러 값	이 열의 확인란은 속성이 여러 값을 가질 수 있는지 여부를 지정합니다. 여러 값을 갖는 속성은 한 항목에서 여러 번 사용할 수 있지만 값이 하나인 속성은 두 번 이상 사용할 수 없습니다.

표 9-2 속성 구문 정의

구문 이름	정의
Binary(이전에는 bin)	속성 값이 이진 데이터로 처리된다는 것을 나타냅니다.

표 9-2 속성 구문 정의 (계속)

구문 이름	정의
Boolean	이 속성에는 True 또는 False의 두 값 중 하나만 있음을 나타냅니다.
Country String	속성 값이 ISO 3166에서 지정한 두 문자 국가 코드(예: FR)로 엄격하게 제한된다는 것을 나타냅니다.
DN(이전에는 dn)	속성 값이 DN(고유 이름)임을 나타냅니다.
DirectoryString (이전에는 cis)	이 속성 값이 UTF-8로 인코딩된 모든 문자를 포함할 수 있으며 대소문자를 구분하지 않음을 나타냅니다.
GeneralizedTime	속성 값이 인쇄 가능한 문자열로 인코딩된다는 것을 나타냅니다. 표준 시간대를 지정해야 하며, 가능하면 GMT를 사용하는 것이 좋습니다.
IA5String(이전에는 ces)	이 속성 값이 ASCII 문자의 부분 집합만 포함할 수 있으며 대소문자를 구분한다는 것을 나타냅니다.
Integer(이전에는 int)	유효한 속성 값이 숫자임을 나타냅니다.
OctetString	이진과 같은 동작을 합니다.
Postal Address	속성 값이 아래와 같이 인코딩된다는 것을 나타냅니다. <i>dstring</i> [\$ <i>dstring</i>]* 여기서 각각의 <i>dstring</i> 구성 요소는 DirectoryString 구문의 값으로 인코딩됩니다. <i>dstring</i> 에 있는 역슬래시와 달러 기호는 줄 구분 기호로 잘못 인식되지 않도록 인용 부호로 묶어야 합니다. 대부분의 서버에서는 주소를 6줄 최대 30자로 제한합니다. 예를 들면 다음과 같습니다. 1234 Main St.\$Anytown, CA 12345\$USA
TelephoneNumber(이전에는 tel)	속성 값이 전화 번호 형식임을 나타냅니다. 국제 표준 형식의 전화 번호를 사용하는 것이 좋습니다.
URI	속성 값이 http://, https://, ftp://, ldap://, ldaps:// 등의 선택적 접두어가 있는 URL을 포함한다는 것을 나타냅니다. URI 값은 IA5String과 같은 동작을 합니다(RFC 2396, http://www.ietf.org/rfc/rfc2396.txt 참조).

속성 작성

스키마에 자신의 속성 정의를 추가하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에 있는 구성 트리에서 스키마 노드를 선택한 다음 오른쪽 패널에서 속성 탭을 선택합니다.
2. 만들기를 눌러 속성 작성 대화 상자를 표시합니다.
3. 텍스트 필드에 다음과 같은 정보를 입력하여 새 속성을 정의합니다. 속성 이름과 구분만 필수입니다.
 - 속성 이름 - 속성 유형이라고도 하는 속성의 고유 이름을 입력합니다. 속성 이름은 문자로 시작해야 하며 ASCII 문자, 숫자 및 하이픈만 사용할 수 있습니다.

주 속성 이름에 대문자가 포함될 수는 있지만 LDAP 클라이언트에서 대문자를 사용해서는 안 됩니다. 속성 이름은 RFC 2251 섹션 4.1.4에 따라 대소문자를 구분하지 않고 처리해야 합니다 (<http://www.ietf.org/rfc/rfc2251.txt>).

- 속성 OID(선택 사항) - 속성의 객체 식별자를 입력합니다. OID에 대해서는 [398 페이지의 표 9-1](#)에서 설명합니다. OID를 지정하지 않으면 Directory Server에서 자동으로 *attributeName-oid*를 사용합니다. LDAP v3을 엄격히 준수하려면 유효한 숫자 OID를 입력해야 합니다.
 - 속성 별명(선택 사항) - 속성의 대체 이름을 쉼표로 구분된 목록에 입력합니다.
 - 속성 설명(선택 사항) - 속성의 용도를 설명하는 짧은 설명 텍스트를 입력합니다.
 - 구분 - 드롭다운 메뉴에서 속성에 저장할 데이터를 설명하는 구문을 선택합니다. 사용할 수 있는 구문에 대해서는 [398페이지의 표 9-2](#)에서 설명합니다.
 - 여러 값 - 기본적으로 속성은 여러 값을 갖습니다. 속성이 항목 당 한 개의 값만 가져야 하는 경우에는 이 확인란을 선택 취소합니다.
4. 속성 작성 대화 상자에서 확인을 눌러 새 속성을 정의합니다. 사용자 정의 속성 테이블에 새 속성이 표시됩니다.

디렉토리 항목에 이 속성 값을 정의하기 전에 [402페이지의 "객체 클래스 정의 관리"](#)에 설명된 것처럼 속성을 필요로 하거나 허용하는 객체 클래스를 작성 또는 편집해야 합니다.

속성 편집

사용자 정의 속성만 콘솔에서 편집할 수 있습니다. 속성의 이름, 구문, 여러 값 정의를 수정하기 전에 현재 이 속성을 사용하는 디렉토리 항목이 없도록 해야 합니다. 속성을 사용하는 항목이 있을 경우 클라이언트에서 액세스할 수 없게 됩니다.

속성의 스키마 정의를 수정하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에 있는 구성 트리에서 스키마 노드를 선택한 다음 오른쪽 패널에서 속성 탭을 선택합니다.
2. 사용자 정의 속성 테이블에서 편집할 속성을 선택하고 편집을 누릅니다.
3. 속성 편집 대화 상자의 필드를 수정하여 속성을 다시 정의합니다.
OID 문자열은 속성 이름에 따라 지정되므로 이름을 변경하는 경우 OID도 변경해야 합니다. OID에 대해서는 [398페이지의 표 9-1](#)에서 설명합니다. 사용할 수 있는 구문에 대해서는 [398페이지의 표 9-2](#)에서 설명합니다.
4. 속성 편집이 끝나면 확인을 눌러 변경 사항을 저장합니다.

속성 삭제

사용자 정의 속성만 콘솔에서 삭제할 수 있습니다. 속성 정의를 삭제하기 전에 현재 이 속성을 사용하는 디렉토리 항목이 없도록 해야 합니다. 속성을 사용하는 항목이 있을 경우 클라이언트에서 액세스할 수 없게 됩니다.

속성의 스키마 정의를 삭제하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에 있는 구성 트리에서 스키마 노드를 선택한 다음 오른쪽 패널에서 속성 탭을 선택합니다.
2. 사용자 정의 속성 테이블에서 속성을 선택하고 삭제를 누릅니다.
3. 확인 메시지가 표시되면 삭제를 확인합니다.
속성 정의는 즉시 삭제되며 이 작업은 실행 취소할 수 없습니다.

객체 클래스 정의 관리

Directory Server 콘솔은 스키마의 모든 객체 클래스를 볼 수 있으며 자신의 객체 클래스 정의를 작성, 편집 및 삭제할 수 있는 인터페이스도 제공합니다.

객체 클래스 보기

현재 정의된 모든 객체 클래스에 대한 정보를 보려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에 있는 구성 트리에서 스키마 노드를 선택한 다음 오른쪽 패널에서 객체 클래스 탭을 선택합니다.

이 탭에는 스키마에 있는 모든 표준(읽기 전용) 및 사용자 정의 객체 클래스 목록이 포함되어 있습니다.

2. 두 목록 중 하나에서 보려는 객체 클래스를 선택합니다.

탭의 다른 필드에는 선택한 객체 클래스에 대한 다음과 같은 정보가 표시됩니다.

표 9-3 객체 클래스 탭의 필드

필드	설명
필수 속성	이 객체 클래스를 사용하는 항목에 반드시 필요한 속성 목록이 포함되어 있습니다. 이 목록에는 상속된 속성도 포함됩니다.
허용되는 속성	이 객체 클래스를 사용하는 항목에 허용되는 속성 목록이 포함되어 있습니다. 이 목록에는 상속된 속성도 포함됩니다.
부모	부모는 한 객체 클래스가 속성과 구조를 상속 받는 특정 객체 클래스를 식별합니다. 객체 클래스는 부모 객체 클래스의 필수 속성과 허용되는 속성을 자동으로 상속합니다.
OID	객체 클래스의 객체 식별자. OID는 스키마 객체를 고유하게 식별하는 문자 열이며 대체로 점으로 구분된 10진수로 구성됩니다. OID에 대한 자세한 내용을 보거나 기업용 접두어를 요청하려면 iana@iana.org 로 IANA(Internet Assigned Number Authority)에 전자 메일을 보내거나 IANA 웹 사이트(http://www.iana.org/)를 방문하십시오.

객체 클래스 작성

다른 객체 클래스로부터 서로 상속 받는 여러 객체 클래스를 작성하는 경우 먼저 부모 객체 클래스를 작성해야 합니다. 새 객체 클래스에 사용자 정의 속성이 사용되면 이러한 속성도 먼저 정의해야 합니다.

주 콘솔에서는 구조적 객체 클래스만 작성할 수 있습니다. 이러한 객체 클래스는 부모로부터 상속 받아야 합니다. 보조 및 추상적 객체 클래스를 정의하려면 명령줄 유틸리티를 사용해야 합니다.

스키마에 자신의 객체 클래스 정의를 추가하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에 있는 구성 트리에서 스키마 노드를 선택한 다음 오른쪽 패널에서 객체 클래스 탭을 선택합니다.
2. 만들기를 눌러 객체 클래스 작성 대화 상자를 표시합니다.
3. 텍스트 필드에 다음과 같은 정보를 입력하여 새 객체 클래스를 정의합니다.

- 이름 - 객체 클래스의 고유 이름을 입력합니다.
- 부모 - 부모가 될 기존의 객체 클래스를 선택합니다. 기본적으로 top이 선택되며, 다른 객체 클래스로부터 상속 받지 않는 경우 이 객체 클래스를 사용해야 합니다. 부모로부터 상속된 필수 속성과 허용되는 속성이 해당 목록에 표시됩니다.

일반적으로 사용자 항목에 새 속성을 추가하려는 경우에는 inetOrgPerson 객체 클래스가 부모입니다. 기업 항목에 새 속성을 추가하려는 경우에는 대체로 organization 또는 organizationalUnit가 부모입니다. 그룹 항목에 새 속성을 추가하려는 경우에는 대체로 groupOfNames 또는 groupOfUniqueNames가 부모입니다.

- OID(선택 사항) - 객체 클래스의 객체 식별자를 입력합니다. OID에 대해서는 [402페이지의 표 9-3](#)에서 설명합니다. OID를 지정하지 않으면 Directory Server는 자동으로 `objectClassName-oid`를 사용합니다. LDAP v3을 엄격히 준수하려면 유효한 숫자 OID를 입력해야 합니다.
4. 새 객체 클래스를 사용하는 항목에 포함될 속성을 정의합니다.
 - 필수 속성을 정의하려면 사용 가능한 속성 목록에서 속성을 하나 이상 선택한 다음 필수 속성 상자 왼쪽에 있는 추가 버튼을 누릅니다.

- *허용되는* 속성을 정의하려면 사용 가능한 속성 목록에서 속성을 하나 이상 선택한 다음 허용되는 속성 상자 왼쪽에 있는 추가 버튼을 누릅니다.
 - 이전에 추가한 속성을 제거하려면 두 목록 중 하나에서 제거할 속성을 선택한 다음 해당 제거 버튼을 누릅니다. 부모 객체 클래스로부터 상속 받은 허용되는 속성 또는 필수 속성은 제거할 수 없습니다.
5. 객체 클래스 작성 대화 상자에서 확인을 눌러 새 객체 클래스를 정의합니다. 사용자 정의 객체 클래스 테이블에 새 객체 클래스가 표시되며, 이 객체 클래스를 사용하여 항목을 정의할 수 있습니다.

객체 클래스 편집

사용자 정의 객체 클래스만 콘솔에서 편집할 수 있습니다. 객체 클래스 정의를 수정하기 전에 현재 이 객체 클래스를 사용하는 디렉토리 항목이 없도록 해야 합니다. 객체 클래스를 사용하는 항목이 있을 경우 클라이언트에서 액세스할 수 없게 됩니다.

객체 클래스의 스키마 정의를 수정하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에 있는 구성 트리에서 스키마 노드를 선택한 다음 오른쪽 패널에서 객체 클래스 탭을 선택합니다.
2. 사용자 정의 객체 클래스 목록에서 편집할 객체 클래스를 선택하고 편집을 누릅니다.
3. 객체 클래스 편집 대화 상자의 필드를 수정하여 객체 클래스를 다시 정의합니다.

객체 클래스의 이름을 바꾸거나 해당 OID를 변경할 수는 없습니다. 이름과 OID를 수정하려면 기존 객체 클래스를 삭제하고 새 객체 클래스를 작성합니다.

- 부모 - 부모가 될 기존의 객체 클래스를 선택합니다. 부모로부터 상속된 필수 속성과 허용되는 속성이 해당 목록에 표시됩니다.
- 필수 속성을 정의하려면 사용 가능한 속성 목록에서 속성을 하나 이상 선택한 다음 필수 속성 상자 왼쪽에 있는 추가 버튼을 누릅니다.
- *허용되는* 속성을 정의하려면 사용 가능한 속성 목록에서 속성을 하나 이상 선택한 다음 허용되는 속성 상자 왼쪽에 있는 추가 버튼을 누릅니다.

- 이전에 추가한 속성을 제거하려면 두 목록 중 하나에서 제거할 속성을 선택한 다음 해당 제거 버튼을 누릅니다. 부모 객체 클래스로부터 상속 받은 허용되는 속성 또는 필수 속성은 제거할 수 없습니다.

4. 객체 클래스 편집이 끝나면 확인을 눌러 변경 사항을 저장합니다.

객체 클래스 삭제

사용자 정의 객체 클래스만 콘솔에서 삭제할 수 있습니다. 객체 클래스 정의를 삭제하기 전에 현재 이 객체 클래스를 사용하는 디렉토리 항목이 없도록 해야 합니다. 객체 클래스를 사용하는 항목이 있을 경우 클라이언트에서 액세스할 수 없게 됩니다.

객체 클래스의 스키마 정의를 삭제하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에 있는 구성 트리에서 스키마 노드를 선택한 다음 오른쪽 패널에서 객체 클래스 탭을 선택합니다.
2. 사용자 정의 객체 클래스 목록에서 객체 클래스를 선택하고 삭제를 누릅니다.
3. 확인 메시지가 표시되면 삭제를 확인합니다.

객체 클래스 정의는 즉시 삭제되며 이 작업은 실행 취소할 수 없습니다.

스키마 정의 복제

두 서버 간에 하나 이상의 접미어 복제를 구성하면 스키마도 자동으로 복제됩니다. 이렇게 해서 모든 복제본은 소비자로 복제될 수 있는 모든 객체 클래스 및 속성을 정의하는 동일한 스키마를 갖게 되며 마스터 서버에도 마스터 스키마가 있습니다.

모든 복제본에서 스키마를 실행하려면 모든 마스터에서 스키마 검사를 활성화해야 합니다. LDAP 작업이 수행되는 마스터에서 스키마를 검사하기 때문에 소비자를 업데이트할 때는 스키마를 검사할 필요가 없습니다. 성능을 향상시키기 위해 복제 기법은 소비자 복제본에 대한 스키마 검사를 무시합니다.

주 허브 및 전용 소비자에서는 스키마 검사를 비활성화하지 마십시오. 스키마 검사는 소비자 성능에 영향을 주지 않으므로 복제본 내용이 스키마에 맞다는 것을 표시하기 위해 계속 활성화 상태로 유지해야 합니다.

마스터 서버는 콘솔이나 명령줄 도구를 통해 스키마가 수정될 때 및 소비자 초기화 중에 자동으로 스키마를 해당 소비자에 복제합니다. 기본적으로 전체 스키마가 복제되며, 소비자에 없는 추가 스키마 요소는 새로 작성되어 `99user.ldif` 파일에 저장됩니다.

예를 들어, 마스터 서버를 시작하면 `98mySchema.ldif` 파일에 스키마 정의가 저장되며 다른 서버(마스터, 허브 또는 전용 소비자)에 대한 복제 계약을 정의한다고 가정해 보십시오. 나중에 이 마스터를 사용하여 복제본을 초기화하면 복제된 스키마는 `98mySchema.ldif`의 정의를 포함하지만 복제본 서버의 `99user.ldif`에 저장됩니다.

스키마가 소비자 초기화 중에 복제된 경우 마스터의 `cn=schema`에 있는 스키마를 수정해도 전체 스키마가 소비자에 복제되므로 명령줄 유틸리티나 콘솔을 통한 마스터 스키마의 모든 수정 사항이 소비자에 복제됩니다. 이러한 수정 사항은 마스터의 `99user.ldif`에 저장되며, 위와 같은 기법에 의해 소비자의 `99user.ldif`에도 저장됩니다.

복제된 스키마 파일 수정

스키마가 포함된 LDIF 파일에서 직접 변경한 사항은 복제 기법에서 감지할 수 없으므로 [396페이지의 "스키마 파일 수정"](#)에 설명된 것처럼 스키마를 업데이트하는 경우 마스터를 다시 시작해도 변경 사항이 소비자에 복제되지 않습니다.

Directory Server 5.2는 스키마 파일의 변경 사항을 소비자에게 "밀어넣는" 다음과 같은 스크립트를 제공합니다.

```
# ServerRoot/slapd-serverID/schema_push.pl
```

마스터 서버의 스키마 파일을 수정하려면 아래 절차를 사용합니다.

1. 스키마 디렉토리에 새 스키마 파일을 추가하거나 기존 스키마 파일을 수정합니다.
`ServerRoot/slapd-serverID/config/schema`
 이 디렉토리에 있는 스키마 파일은 설치 중에 정의된 시스템 사용자만 쓸 수 있습니다. 자세한 내용은 [396페이지의 "스키마 파일 수정"](#)을 참조하십시오.
2. 위의 설명대로 `schema_push.pl` 스크립트를 실행합니다. 이 스크립트는 실제로 스키마를 복제본에 "밀어넣는" 것이 아니라 스키마 파일이 로드되면 즉시 복제되도록 특수 속성을 스키마 파일에 작성합니다.
3. 서버를 다시 시작합니다. 모든 스키마 파일이 로드되며, 복제 기법이 새 스키마를 해당 소비자에 복제합니다.

스키마 복제 제한

기본적으로 복제 기법은 스키마를 복제할 때 항상 전체 스키마를 소비자에게 보냅니다. 하지만 다음 두 가지 경우에는 이러한 기본 설정이 바뀔지 않습니다.

- 콘솔이나 명령줄에서 `cn=schema`를 수정하면 사용자 정의 스키마 요소만 변경되고 표준 스키마는 변경되지 않습니다. 자주 스키마를 수정하면 변경되지 않는 스키마 요소의 대규모 집합을 매번 보내야 하기 때문에 성능이 저하됩니다. 이 경우 사용자 정의 스키마 요소만 복제하여 복제 및 서버 성능을 향상시킬 수 있습니다.
- Directory Server 5.2의 마스터를 Directory Server 5.1의 소비자에 복제하면 두 버전의 구성 속성에 대한 스키마가 다르기 때문에 충돌이 발생합니다. 이 경우 아래에 설명된 것처럼 반드시 사용자 정의 스키마 요소만 복제해야 합니다.

사용자 정의 스키마만 복제되도록 스키마 복제를 제한하려면 아래 명령을 실행합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=config
changetype: modify
replace: nsslapd-schema-repl-useronly
nsslapd-schema-repl-useronly: on
^D
```

기본값 `off`를 설정하면 필요한 경우 전체 스키마가 복제됩니다.

스키마 정의 복제

디렉토리 데이터 색인화

책 색인과 마찬가지로 **Directory Server** 색인은 검색 문자열을 디렉토리 내용에 대한 참조와 연결하여 검색 속도를 향상시킵니다. 색인은 별도의 데이터베이스 파일에 저장된 속성 값의 테이블로서, 디렉토리의 각 접미어에 대해 개별적으로 작성 및 관리됩니다. 접미어 구성에 색인을 작성하면 서버에서 자동으로 색인을 유지 관리합니다.

색인화 소개, 색인화의 손실과 이점, `nsslapd-allidsthreshold` 속성에 대한 설명, **Directory Server**의 성능 향상 방법 등에 대해서는 *Directory Server Performance Tuning Guide*를 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

- [색인화에 대한 개요](#)
- [색인 관리](#)
- [찾아보기 색인 관리](#)

색인화에 대한 개요

각 접미어에 대한 색인은 해당 데이터베이스 디렉토리의 파일에 저장됩니다. 각 색인 파일에는 지정된 속성에 대해 접미어에 정의된 모든 색인이 포함됩니다. 예를 들어, 일반 이름 (`cn`) 속성에 대해 유지 관리되는 모든 색인은 `databaseName_cn.db3` 파일에 저장됩니다.

접미어를 초기화하거나 이 장에 설명된 명령을 사용하면 색인 파일이 작성됩니다. 클라이언트 검색 작업 및 내부 작업 중에 서버는 색인에 액세스하여 보다 신속하게 디렉토리 항목을 찾을 수 있습니다. 수정 작업 중에는 디렉토리에서 디렉토리 내용을 업데이트한 후에 색인 파일을 업데이트하여 색인을 유지 관리합니다.

Directory Server는 다음과 같은 유형의 색인을 지원합니다.

- 있음 색인(pres) - 속성 값에 관계 없이 특정 속성이 있는 모든 항목이 포함됩니다.
- 동일 색인(eq) - 특정 속성 값이 있는 항목을 효율적으로 검색할 수 있습니다.
- 근사 색인(approx) - ~= 필터 연산자를 사용하여 효율적인 "유사 발음" 검색을 제공합니다. 예를 들어, 근사 색인은 이름의 일부나 맞춤법이 틀린 이름을 검색하는 데 유용합니다. Directory Server는 Metaphone Phonetic 알고리즘의 변형을 사용하여 근사 색인에 대한 검색을 수행합니다.

주 Directory Server의 Metaphone Phonetic 알고리즘은 US-ASCII 문자만 지원하기 때문에 영어 값에만 근사 색인화를 사용해야 합니다.

- 하위 문자열 색인(sub) - cn=*john* 또는 cn=john* 같은 속성 값 하위 문자열을 효율적으로 검색합니다. 하지만 각각의 값이 수많은 하위 문자열을 갖기 때문에 유지 관리 손실이 매우 큰 색인입니다.

Directory Server는 색인에서 두문자의 하위 문자열을 검색할 수 있도록 하위 문자열을 색인화합니다. 따라서 색인을 사용하면 (sn=*ab) 검색 속도는 향상될 수 있지만 (sn=*a) 검색 속도는 향상되지 않습니다. Directory Server는 와일드카드 앞에 한 문자뿐인 이니셜 하위 문자열 검색을 허용하여 최적화를 향상시킵니다. 따라서 하위 문자열 색인이 있는 경우 (sn=a*) 검색 속도는 향상될 수 있지만 (sn=*a) 검색 속도는 향상되지 않습니다. 하위 문자열 색인에 대한 자세한 내용은 *Directory Server Performance Tuning Guide*를 참조하십시오.

- 일치 규칙 색인 - 현지화된 일치 규칙(조합 순서라고도 함)의 OID를 색인화할 속성과 연결하여 다국어 디렉토리의 검색 속도를 향상시킵니다.
- 찾아보기 색인 - 가상 목록 보기(VLV) 컨트롤을 사용한 검색의 응답 시간을 향상시킵니다. 데이터가 가득 채워진 하위 트리(예: ou=People, dc=example, dc=com)의 표시 성능을 높이기 위해 분기점에 대한 찾아보기 색인을 작성할 수도 있습니다.

시스템 색인

시스템 색인은 삭제하거나 수정할 수 없으며 Directory Server가 효율적으로 정상 작동하는 데 필요합니다. 아래 표에는 모든 접미어에 자동으로 작성되는 시스템 색인이 나와 있습니다.

표 10-1 모든 접미어의 시스템 색인

속성	동일	있음	용도
aci	X		디렉토리에 유지 관리된 액세스 제어 정보를 신속하게 가져올 수 있게 합니다.
ancestorid	X		모든 항목에 대한 상위 목록이 포함됩니다.
entrydn	X		DN 검색에 의한 항목 검색 속도를 향상시킵니다.
id2entry	X		디렉토리 항목의 실제 데이터베이스가 포함됩니다. 이 파일을 사용하여 다른 모든 데이터베이스 파일을 다시 작성할 수 있습니다.
nsUniqueid	X		특정 항목을 검색하는 데 사용됩니다.
nscpEntryDN	X		Directory Server에서 복제를 위해 내부적으로 사용됩니다.
nsds5ReplConflict	X	X	복제 충돌을 찾는 데 사용됩니다.
numsubordinates		X	Directory Server 콘솔에서 디렉토리 탭의 표시 성능을 향상시키는 데 사용됩니다.
objectClass	X		디렉토리 내의 하위 트리 검색을 가속화하는 데 사용됩니다.
parentID	X		한 수준 검색 중에 디렉토리 성능을 향상시킵니다.

기본 색인

디렉토리에 새 접미어를 작성하면 서버는 해당 데이터베이스 디렉토리에 기본 색인 집합을 구성합니다. 기본 색인은 사용자의 색인화 요구에 따라 수정할 수 있습니다. 단, 기업의 서버 플러그인 또는 다른 서버에서 색인화된 특정 속성을 사용하지 않을 경우에만 해당 색인의 구성을 해제할 수 있습니다.

새 접미어를 작성할 때 사용할 기본 색인 집합을 수정하려면 [421페이지의 "기본 색인 집합 수정"](#)을 참조하십시오.

아래 표에는 Directory Server에 사전 구성된 기본 색인이 나와 있습니다.

표 10-2 모든 새 접미어의 기본 색인

속성	동일	있음	하위 문자 열	용도
cn	X	X	X	가장 일반적인 유형의 사용자 디렉토리 검색 성능을 향상시킵니다.
givenName	X	X	X	가장 일반적인 유형의 사용자 디렉토리 검색 성능을 향상시킵니다.
mail	X	X	X	가장 일반적인 유형의 사용자 디렉토리 검색 성능을 향상시킵니다.
mailAlternateAddress	X			Sun Java System Messaging Server에서 사용됩니다.
mailHost	X			Sun Java System Messaging Server에서 사용됩니다.
member	X			Sun Java System 서버 성능을 향상시킵니다. 이 색인은 참조 무결성 플러그 인에서도 사용됩니다. 자세한 내용은 108페이지의 "참조 무결성 유지"를 참조하십시오.
nsCalXItemId	X	X	X	Sun Java System Calendar Server에서 사용됩니다.
nsLIProfileName	X			Sun Java System Messaging Server의 로밍 기능에 사용됩니다.
nsRoleDN	X			역할 기반의 작업 성능을 향상시킵니다.
nswcalCALID	X			Sun Java System Calendar Server에서 사용됩니다.
owner	X			Sun Java System 서버 성능을 향상시킵니다. 이 색인은 참조 무결성 플러그 인에서도 사용됩니다. 자세한 내용은 108페이지의 "참조 무결성 유지"를 참조하십시오.
pipstatus	X			Sun Java System 서버에서 사용됩니다.
pipuid		X		Sun Java System 서버에서 사용됩니다.
seeAlso	X			Sun Java System 서버 성능을 향상시킵니다. 이 색인은 참조 무결성 플러그 인에서도 사용됩니다. 자세한 내용은 108페이지의 "참조 무결성 유지"를 참조하십시오.
sn	X	X	X	가장 일반적인 유형의 사용자 디렉토리 검색 성능을 향상시킵니다.
telephoneNumber	X	X	X	가장 일반적인 유형의 사용자 디렉토리 검색 성능을 향상시킵니다.
uid	X			Sun Java System 서버 성능을 향상시킵니다.

표 10-2 모든 새 접미어의 기본 색인(계속)

속성	동일 있음	하위 문자 열	용도
uniquemember	X		Sun Java System 서버 성능을 향상시킵니다. 이 색인은 참조 무결성 플러그 인에서도 사용됩니다. 자세한 내용은 108페이지의 "참조 무결성 유지"를 참조하십시오.

속성 이름 빠른 참조 테이블

아래 표에는 기본 이름(실제 이름)과 별명을 가진 모든 속성이 나와 있습니다. 색인을 작성할 때는 반드시 기본 이름을 사용하십시오.

표 10-3 속성의 기본 이름 및 별명

기본 속성 이름	속성 별명
authorCn	documentAuthorCommonName
authorSn	documentAuthorSurname
c	countryName
cn	commonName
co	friendlyCountryName
dc	domainComponent
dn	distinguishedName
drink	favoriteDrink
facsimileTelephoneNumber	fax
l	localityName
labeledUri	labeledUrl
mail	rfc822mailbox
mobile	mobileTelephoneNumber
o	organizationName
ou	organizationalUnitName
pager	pagerTelephoneNumber
sn	surname
st	stateOrProvinceName

표 10-3 속성의 기본 이름 및 별명

기본 속성 이름	속성 별명
street	streetAddress
ttl	timeToLive
uid	userId

색인 관리

이 절에서는 Directory Server 콘솔과 명령줄에서 특정 속성에 대한 있음, 동일, 근사, 하위 문자열 및 국가별 색인을 작성 및 제거하는 방법에 대해 설명합니다. 가상 목록 보기 (VLV) 작업 전에 필요한 별도의 절차에 대해서는 [422페이지의 "찾아보기 색인 관리"](#)를 참조하십시오.

주	<p>색인은 각 접미어별로 작성되기 때문에 모든 접미어 구성에서 새 색인을 작성해야 합니다.</p> <p>콘솔을 사용하여 새 접미어를 작성하는 경우 기존 접미어의 색인 구성을 복제할 수 있습니다.</p>
----------	---

새 색인을 작성하기 전에 색인을 유지 관리함으로써 얻을 수 있는 이점과 손실을 비교해 보십시오. 다음과 같은 점에 주의합니다.

- 전화 번호와 같이 주로 숫자가 포함되는 속성에 근사 색인을 사용하는 것은 효율적이지 않으므로 피해야 합니다.
- 이진 속성에 대한 하위 문자열 색인은 제대로 작동하지 않습니다.
- 이진 데이터가 포함될 속성(예: jpegPhoto) 등의 큰 값에 동일 색인을 사용하는 것은 피해야 합니다.
- 색인을 유지 관리하려면 많은 자원이 필요하므로 빈번하게 검색되는 속성만 색인화해야 합니다. 항목을 작성하려면 서버에서 색인화된 모든 속성을 조사하여 새 항목에 포함된 각 속성에 대해 새 색인 항목을 생성해야 하므로 많은 CPU 시간이 소요됩니다.
- 각 색인 파일의 크기는 디렉토리 내용에 비례합니다.
- 색인화되지 않은 속성도 검색 요청에 지정할 수 있지만, 이 경우 검색 유형에 따라 검색 성능이 색인화된 검색보다 훨씬 떨어집니다.

콘솔에서 색인 관리

많은 속성에 대한 색인을 수정 또는 추가하려는 경우 먼저 접미어를 읽기 전용으로 설정한 후에 접미어 내용을 LDIF로 내보내야 합니다. 이렇게 하면 LDIF 파일을 사용하여 접미어를 다시 초기화함으로써 신속하게 접미어를 다시 색인화할 수 있습니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드를 확장하여 색인화할 접미어를 선택합니다. 그런 후에 오른쪽 패널에서 색인 탭을 선택합니다.

시스템 색인 테이블은 수정할 수 없습니다. 추가 색인 테이블에서 속성에 대한 색인을 추가, 수정 또는 제거합니다.

2. 아직 색인화되지 않은 속성에 대한 색인을 추가하려면 속성 추가 버튼을 누릅니다. 표시되는 대화 상자에서 색인화할 속성을 하나 이상 선택하고 확인을 누릅니다.

추가 색인 테이블에 새 속성이 표시됩니다.

3. 특정 속성의 색인을 수정하려면 추가 색인 테이블에서 해당 속성에 대해 유지 관리할 각 색인 유형에 해당하는 확인란을 선택하거나 선택 취소합니다.

4. 영어 이외의 언어로 지정된 값을 포함하는 속성에 대한 색인을 작성하려면 사용할 조합 순서 OID를 일치 규칙 필드에 입력합니다.

여러 개의 OID를 공백 없이 쉼표로 구분하여 속성을 여러 언어로 색인화할 수 있습니다. 지원되는 로캘의 목록과 관련 조합 순서의 OID를 보려면 *Directory Server Administration Reference*를 참조하십시오.

5. 속성에 대한 모든 접미어를 제거하려면 테이블에서 해당 행을 선택하고 속성 삭제 버튼을 누릅니다.

6. 저장을 눌러 새 색인 구성을 저장합니다.

속성에 대한 색인을 모두 제거하면 해당 속성에 대한 색인 파일이 제거되고 구성이 완료됩니다. 속성에 대한 색인을 수정했거나 새 색인을 추가한 경우에는 아래 단계에 따라 수행합니다.

7. 새 색인을 사용하려면 데이터베이스 파일을 업데이트해야 한다는 경고 대화 상자가 표시됩니다. 접미어를 다시 색인화하거나 다시 초기화할 수 있습니다.

- 한두 개의 색인만 추가 또는 수정했거나 접미어를 계속 사용할 수 있어야 하는 경우에는 접미어를 다시 색인화해야 합니다. 접미어 다시 색인화 버튼을 눌러 다시 색인화 대화 상자를 표시합니다. 기본적으로 색인 구성에 추가했거나 수정한 속성이 선택됩니다. 확인을 눌러 이러한 속성을 다시 색인화합니다. 수백만 개의 항목이 있는 디렉토리의 많은 속성을 다시 색인화하려면 몇 시간이 걸릴 수도 있지만 다시 색인화 중에는 접미어가 항상 온라인 상태를 유지합니다.
- 여러 속성에 대한 색인을 추가 또는 수정했으며 이 접미어에서 내보낸 최근 LDIF 파일이 있는 경우 접미어 초기화 버튼을 누릅니다. 접미어 초기화 대화 상자에서 LDIF 파일의 경로와 이름을 입력하거나 탐색한 다음 확인을 누릅니다. LDIF 파일을 사용하여 접미어가 다시 초기화되고 새 구성에 따라 모든 색인이 작성됩니다. 디렉토리 크기에 따라 대체로 접미어를 다시 초기화하는 것이 두 개 이상의 속성을 다시 색인화하는 것보다 속도가 더 빠르지만 초기화 중에는 접미어를 사용할 수 없다는 단점이 있습니다.
- 접미어를 다시 색인화하거나 다시 초기화하지 않아도 모든 데이터를 계속 사용할 수는 있지만 새 색인이 작성되지 않으며 디렉토리 액세스 성능이 향상되지 않습니다.

접미어를 다시 색인화하거나 다시 초기화하면 새 색인이 디렉토리에 추가한 새 데이터 및 디렉토리의 기존 데이터에 대해 즉시 활성화되므로 서버를 다시 시작할 필요가 없습니다.

명령줄에서 색인 관리

명령줄에서 색인을 작성하거나 수정하려면 다음 두 단계를 수행해야 합니다.

- `ldapmodify` 명령줄 유틸리티를 사용하여 색인 구성 항목을 추가하거나 수정합니다. 색인은 각 접미어에 별도로 구성되며 색인 구성 항목은 해당 데이터베이스 구성과 함께 저장됩니다.
- `directoryserver db2index-task` 명령을 실행하면 서버에서 유지 관리하는 새로운 색인 집합이 생성됩니다.

주의 시스템 색인은 삭제하지 마십시오. 시스템 색인을 삭제하면 Directory Server 성능이 크게 저하될 수 있습니다. 시스템 색인은 다음 항목에 있습니다.

```
cn=index, cn=databaseName, cn=ldb database, cn=plugins, cn=config
cn=default indexes, cn=config, cn=ldb database,
cn=plugins, cn=config
```

기본 색인을 삭제하는 경우에도 Directory Server 작동에 영향을 줄 수 있으므로 주의해야 합니다.

색인 구성 항목 작성

아직 색인화되지 않은 속성에 대한 색인을 작성하려면 해당 데이터베이스의 구성에 이 속성에 대한 새 항목을 작성해야 합니다.

색인 구성 항목의 DN은 다음과 같습니다.

```
cn=attributeName, cn=index, cn=databaseName, cn=ldb database,
cn=plugins, cn=config
```

여기서 *databaseName*은 색인을 작성할 접미어에 해당하는 데이터베이스 이름입니다. 예를 들어, 아래 명령은 프랑스어의 sn(성) 속성 값에 대한 있음, 동일, 하위 문자열 및 "유사 발음" 색인을 작성합니다.

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=sn, cn=index, cn=databaseName, cn=ldb database,
cn=plugins, cn=config
objectClass: top
objectClass: nsIndex
cn: sn
nsSystemIndex: false
nsIndexType: pres
nsIndexType: eq
nsIndexType: sub
nsIndexType: approx
nsMatchingRule: 1.3.6.1.4.1.42.2.27.9.4.76.1
^D
```

색인 구성 항목에는 nsIndex 객체 클래스가 있으며, nsSystemIndex 속성이 존재하고 해당 속성 값이 false여야 합니다. 새 시스템 색인은 작성할 수 없습니다. Directory Server에서 내부적으로 정의한 기존의 시스템 색인만 유지 관리됩니다.

nsIndexType 속성 값은 지정된 속성에 대해 유지 관리될 색인을 열거합니다. 위에 제공된 값 중 하나를 사용하여 해당 색인을 정의합니다.

속성 색인화를 일시적으로 비활성화하기 위해 none 값만 사용하여 해당 속성에 대한 색인을 명시적으로 비활성화할 수도 있습니다. 색인 구성 항목에 nsIndexType 속성을 추가하지 않으면 기본적으로 모든 색인이 유지 관리됩니다.

선택 사항 nsMatchingRule 속성에는 서버에서 인식하는 일치 규칙의 OID가 들어 있습니다. 여기에는 국가별 색인에 대한 언어 조합의 OID가 들어 있지만 CaseExactMatch와 같은 기타 일치 규칙도 있습니다. 지원되는 로케일의 목록과 관련 조합 순서의 OID를 보려면 *Directory Server Administration Reference*를 참조하십시오.

색인 구성 속성에 대한 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.

주 색인을 작성할 때는 항상 속성의 별명이 아닌 기본 이름을 사용해야 합니다. 기본 속성 이름은 스키마에서 해당 속성에 지정된 이름입니다(예: userid 속성의 경우 uid). 속성의 기본 이름 및 별명 목록은 [413페이지의 표 10-3](#)을 참조하십시오.

색인 구성 항목 수정

속성에 대해 이미 정의된 색인을 구성하려면 해당 색인 항목을 수정합니다. 예를 들어, 이전에 정의한 sn 색인 구성에 대해 아래 명령을 실행하면 "유사 발음" 색인이 제거되고 언어가 캐나다 프랑스로 변경됩니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=sn,cn=index,cn=databaseName,cn=ldb database,
  cn=plugins,cn=config
changetype: modify
delete: nsIndexType
nsIndexType: approx
-
replace: nsMatchingRule
nsMatchingRule: 1.3.6.1.4.1.42.2.27.9.4.78.1
^D
```

db2index-task 실행

색인화 항목을 새로 작성했거나 기존 색인화 항목에 다른 색인 유형을 추가했거나 색인화 항목의 조합 순서를 수정한 경우 `directoryserver db2index-task` 명령을 실행하여 새 색인을 생성합니다. 이 명령은 접미어 내용을 읽은 후에 해당 구성 항목에 따라 지정된 속성을 다시 색인화합니다.

명령을 실행하는 동안에도 서버를 통해 계속 접미어 내용을 사용할 수 있지만 명령이 완료될 때까지는 검색이 색인화되지 않습니다. 다시 색인화하려면 상당한 자원이 필요하므로 서버의 다른 작업 성능이 저하될 수 있습니다. 디렉토리 크기에 따라 대체로 접미어를 다시 초기화하는 것이 두 개 이상의 속성을 다시 색인화하는 것보다 속도가 더 빠르지만 초기화 중에는 접미어를 사용할 수 없다는 단점이 있습니다. 자세한 내용은 [421페이지](#)의 "접미어 다시 초기화"를 참조하십시오.

아래 예에서는 `databaseName`에 해당하는 접미어의 `sn` 색인을 다시 생성합니다.

```
# /usr/sbin/directoryserver db2index-task
-D "cn=Directory Manager" -w password -n databaseName -t sn
```

자세한 내용은 *Directory Server Man Page Reference*를 참조하십시오.

특정 속성에 대한 모든 색인 삭제

특정 속성에 대해 구성된 모든 색인을 제거하려면 해당 구성 항목과 데이터베이스 파일을 제거할 수 있습니다. 예를 들어, 아래 명령은 `databaseName` 데이터베이스의 `sn` 속성에 대한 모든 색인의 구성을 해제합니다.

```
ldapdelete -h host -p port -D "cn=Directory Manager" -w password \
"cn=sn,cn=index,cn=databaseName,cn=ldbm database,cn=plugins, \
cn=config"
```

이 항목을 삭제하면 `databaseName` 데이터베이스에 해당하는 접미어의 `sn` 속성에 대한 색인이 유지 관리되지 않습니다. 디스크 공간을 절약하려면 서버에서 더 이상 사용하지 않는 해당 색인 파일을 삭제할 수도 있습니다. 이 예에서는 아래 파일을 삭제할 수 있습니다.

```
ServerRoot/slapd-serverID/db/databaseName/databaseName_sn.db3
```

접미어 다시 색인화

색인 파일이 손상되면 접미어를 다시 색인화하여 해당 데이터베이스 디렉토리에 색인 파일을 다시 작성해야 합니다. Directory Server 콘솔을 사용하여 접미어를 다시 색인화하려면 다시 색인화하거나 다시 초기화하는 두 가지 방법이 있습니다.

접미어 다시 색인화

접미어를 다시 색인화하면 서버는 포함된 모든 항목을 조사하여 색인 파일을 다시 구성합니다. 다시 색인화하는 동안에도 접미어 내용을 읽기 및 쓰기 작업에 사용할 수 있습니다. 하지만 전체 접미어에서 다시 색인화되는 각 속성을 검사해야 하므로 구성하는 색인에 따라 수백만 개의 항목이 있는 접미어의 경우 작업에 몇 시간이 걸릴 수도 있습니다. 또한 다시 색인화하는 동안에는 색인을 사용할 수 없으며 서버 성능이 저하됩니다.

콘솔을 사용하여 접미어를 다시 색인화하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 데이터 노드를 확장하여 다시 색인화할 접미어를 표시합니다.
2. 접미어 구성 노드를 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 다시 색인화를 선택합니다. 또는 노드를 마우스 왼쪽 버튼으로 눌러 선택한 다음 객체 메뉴에서 다시 색인화를 선택할 수도 있습니다.

선택한 접미어에 색인화된 모든 속성 목록이 있는 접미어 다시 색인화 대화 상자가 표시됩니다.
3. 다시 색인화할 각 속성 옆에 있는 확인란을 선택합니다. 모두 선택 및 선택 안 함 버튼을 사용하면 원하는 항목을 쉽게 선택할 수 있습니다. 지정된 속성에 대한 모든 색인은 한 개의 데이터베이스 파일에 저장되므로 모든 색인을 동시에 다시 색인화해야 합니다.
4. 확인을 누릅니다. 예상치 못한 검색 결과 및 다시 색인화하는 동안 성능에 미치는 영향에 대한 확인 메시지가 콘솔에 표시됩니다.
5. 예를 눌러 다시 색인화를 시작합니다.

다시 색인화에 대한 모든 메시지가 포함된 대화 상자가 콘솔에 표시됩니다. 작업이 끝나면 대화 상자를 닫습니다.

명령줄에서 접미어를 다시 색인화하려면 [419페이지의 "db2index-task 실행"](#)에 설명된 지침에 따라 색인 파일을 다시 구성할 모든 속성을 지정합니다.

접미어 다시 초기화

접미어를 다시 초기화하는 경우 새 내용을 가져오면 해당 내용이 바뀌고 새 색인 파일이 작성됩니다. 항목을 로드하면 모든 속성이 한 번에 색인화되므로 대체로 접미어를 다시 초기화하는 것이 두 개 이상의 속성을 다시 색인화하는 것보다 속도가 더 빠르지만 다시 초기화하는 동안에는 접미어를 사용할 수 없다는 단점이 있습니다.

아래 단계는 모두 Directory Server 콘솔이나 명령줄에서 수행할 수 있습니다.

1. 151페이지의 "액세스 권한 및 참조 설정"에 설명된 것처럼 접미어를 읽기 전용으로 설정합니다. 내보내기 후에 내용이 수정되지 않도록 먼저 접미어에 쓰기 금지를 설정해야 합니다.
2. 196페이지의 "콘솔에서 개별 접미어를 LDIF로 내보내기"에 설명된 것처럼 전체 접미어를 LDIF 파일로 내보냅니다.
3. 191페이지의 "접미어 초기화"에 설명된 것처럼 이 LDIF 파일을 가져와서 접미어를 다시 초기화합니다.

초기화하는 동안에는 접미어를 사용할 수 없습니다. 초기화가 끝나면 구성된 모든 색인을 사용할 수 있습니다.
4. 151페이지의 "액세스 권한 및 참조 설정"에 설명된 것처럼 접미어에 대한 쓰기를 다시 허용합니다.

기본 색인 집합 수정

새 접미어를 작성할 때 사용되는 기본 색인 집합은 아래 항목에 정의됩니다.

```
cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config
```

콘솔이나 명령줄에서 접미어를 작성하면 기본 색인 정의 항목이 있는 그대로 복사되어 해당 데이터베이스의 초기 색인 구성이 됩니다.

기본 색인 집합은 명령줄 유틸리티에서만 구성할 수 있습니다. 기본 색인 항목은 416페이지의 "명령줄에서 색인 관리"에 설명된 색인 구성 항목과 동일한 구문을 사용합니다. 예를 들어, 기본 색인 구성 항목을 추가하려면 아래의 `ldapmodify` 명령을 사용할 수 있습니다.

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=drink,cn=default indexes,cn=config,cn=ldb database,
   cn=plugins,cn=config
objectClass: top
objectClass: nsIndex
cn: drink
nsSystemIndex: false
nsIndexType: eq
nsIndexType: sub
nsMatchingRule: 1.3.6.1.4.1.42.2.27.9.4.76.1
^D
```

이 항목을 추가하면 프랑스어의 동일 검색 및 하위 문자열 검색에 대해 색인화된 drink 속성 값이 모든 새 접미어에 포함됩니다.

기본 색인 항목을 수정 또는 삭제하려면 `ldapmodify` 또는 `ldapdelete` 명령을 사용하여 `cn=default indexes,cn=config,cn=ldb database,cn=plugins,cn=config`에 있는 색인 집합을 편집합니다.

찾아보기 색인 관리

찾아보기 색인은 서버측 정렬이나 가상 목록 보기(VLV) 결과를 요청하는 검색 작업에만 사용되는 특수 색인입니다. 찾아보기 색인을 사용하면 다수 검색 결과의 서버측 정렬을 요청하는 검색 성능을 향상시킬 수 있습니다. 찾아보기 색인이 정의되어 있지 않을 경우 디렉토리 구성에 따라 서버에서 정렬을 요청하는 검색을 수행하지 않는 경우도 있습니다. 이렇게 하면 대규모 정렬 작업으로 인한 서버 자원의 소모를 방지할 수 있습니다.

찾아보기 색인은 검색 기본 항목에 적용되며 정렬 요청에 사용되는 각 검색 필터에 대해 별도의 색인을 작성해야 합니다. 예를 들어, 클라이언트 응용 프로그램이 모든 사용자의 정렬 목록을 자주 요청하는 경우 클라이언트가 사용한 필터 문자열의 `ou=People`에 대해 찾아보기 색인을 작성할 것입니다.

다른 색인과 마찬가지로 찾아보기 색인을 유지 관리하기 위한 업데이트 작업 중에도 성능 손실이 발생합니다. 찾아보기 색인을 배포할 경우 신중하게 계획하고 테스트해야 합니다.

콘솔에 대한 찾아보기 색인

Directory Server 콘솔은 전체 디렉토리에 대한 검색을 자주 수행하여 패널 내용을 갱신합니다. 47페이지의 "디렉토리 트리 보기 옵션"에 설명된 것처럼 디렉토리 트리 항목을 정렬하도록 콘솔을 구성한 경우 콘솔에 대한 찾아보기 색인을 작성해야 합니다.

콘솔에 대한 찾아보기 색인은 콘솔에서 수행하는 각 검색별로 작성되며 콘솔을 사용하여 작성할 수도 있습니다. 콘솔에 대한 찾아보기 색인을 작성하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 디렉토리 트리를 탐색하여 정렬해야 하는 대규모 하위 트리의 부모(예: 수천 개의 사용자 항목이 있는 `ou=People,dc=example,dc=com`)를 표시합니다.
2. 부모 항목을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 찾아보기 색인 만들기를 선택합니다. 또는 항목을 왼쪽 마우스 버튼으로 눌러 선택한 다음 객체 메뉴에서 찾아보기 색인 만들기를 선택합니다.

찾아보기 색인 만들기 대화 상자가 표시되어 색인 작성 상태를 보여줍니다. 콘솔은 아래에 표시된 찾아보기 색인 구성 항목을 작성하고 색인 파일의 내용을 생성합니다.

3. 단기를 눌러 찾아보기 색인 만들기 대화 상자를 닫습니다.

새 색인은 콘솔 갱신 작업 시 즉시 활성화되고 디렉토리에 새 데이터가 추가될 때마다 유지 관리되므로 서버를 다시 시작할 필요가 없습니다.

콘솔에 대한 찾아보기 색인 구성은 다음과 같은 항목으로 이루어져 있습니다.

`vlvSearch` 항목은 색인화할 검색의 기본 항목, 범위 및 필터를 정의합니다. `vlvIndex` 항목의 `vlvSort` 속성은 정렬이 지원되는 속성을 정렬되는 순서대로 표시합니다.

```
dn: cn=MCC entryDN, cn=databaseName, cn=ldbm database,
   cn=plugins, cn=config
objectClass: top
objectClass: vlvSearch
cn: MCC entryDN
vlvBase: "entryDN"
vlvScope: 1
vlvFilter: (|(objectclass=*)(objectclass=ldapsubentry))
```

```
dn: cn=by MCC entryDN, cn=MCC entryDN, cn=databaseName,
   cn=ldbm database, cn=plugins, cn=config
```

```
objectClass: top
objectClass: vlvIndex
cn: by MCC entryDN
vlvSort: cn givenname o ou sn uid
```

Directory Server 콘솔에 대한 찾아보기 색인을 삭제하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 디렉토리 탭에서 디렉토리 트리를 탐색하여 찾아보기 색인을 작성한 항목을 표시합니다.
2. 항목을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 찾아보기 색인 삭제를 선택합니다. 또는 항목을 왼쪽 마우스 버튼으로 눌러 선택한 다음 객체 메뉴에서 찾아보기 색인 삭제를 선택합니다. 이 메뉴 항목은 콘솔에 대한 찾아보기 색인이 선택한 항목에 있는 경우에만 활성화됩니다.
3. 색인 삭제를 확인하는 찾아보기 색인 삭제 경고 대화 상자가 표시됩니다. 예를 눌러 찾아보기 색인을 삭제합니다.

클라이언트 검색에 대한 찾아보기 색인

클라이언트 검색 결과 정렬에 대해 사용자 정의된 찾아보기 색인은 수동으로 정의해야 합니다. 명령줄에서 찾아보기 색인, 즉 가상 목록 보기(VLV) 색인을 작성하려면 다음 두 단계를 수행해야 합니다.

- `ldapmodify` 유틸리티 또는 Directory Server 콘솔의 디렉토리 탭을 사용하여 새 찾아보기 색인 항목을 추가하거나 기존의 찾아보기 색인 항목을 편집합니다.
- `directoryserver vlvindex` 명령을 실행하여 서버에서 유지 관리할 새 찾아보기 색인 집합을 생성합니다.

찾아보기 색인 항목 지정

찾아보기 색인은 지정된 기본 항목 및 해당 하위 트리별로 작성됩니다. 또한 찾아보기 색인 구성은 이 항목이 포함된 접미어의 데이터베이스 구성에 정의됩니다.

주 찾아보기 색인은 연결 접미어가 아닌 로컬 접미어 및 하위 접미어에만 작성할 수 있습니다.

찾아보기 색인은 두 개의 항목으로 구성됩니다. 첫 번째 항목은 `vlvSearch` 객체 클래스를 사용하며 색인화할 검색의 기본 항목, 범위 및 필터를 지정합니다. 두 번째 항목은 첫 번째 항목의 자식으로, `vlvIndex` 객체 클래스를 사용하여 정렬할 속성 및 정렬 순서를 지정합니다.

아래 예에서는 `ldapmodify` 유틸리티를 사용하여 두 개의 찾아보기 색인 구성 항목을 작성합니다.

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=Browsing ou=People, cn=databaseName,
   cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: vlvSearch
cn: Browsing ou=People
vlvbase: ou=People,dc=example,dc=com
vlvscope: 1
vlvfilter: (objectclass=inetOrgPerson)

dn: cn=Sort rev employeenumber, cn=Browsing ou=People,
   cn=databaseName,cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: vlvIndex
cn: Sort rev employeenumber
vlvSort: -employeenumber
^D
```

`vlvscope`는 기본 항목만 검색하는 경우 0, 기본 항목의 직계 자식의 경우 1, 기본 항목을 루트로 하는 전체 하위 트리의 경우 2로 지정됩니다. `vlvfilter`는 클라이언트 검색 작업에 사용할 LDAP 필터입니다. 찾아보기 색인 항목은 모두 같은 위치에 있으므로 항목을 설명하는 `cn` 값을 사용하여 찾아보기 색인 이름을 지정해야 합니다.

각각의 `vlvSearch` 항목에는 `vlvIndex` 항목이 한 개 이상 있어야 합니다. `vlvSort` 속성은 정렬 기준으로 사용할 속성 및 정렬 순서를 정의하는 속성 이름 목록입니다. 속성 이름 앞에 대시(-)가 있으면 역순서를 나타냅니다. 여러 개의 `vlvIndex` 항목을 정의하여 검색에 하나 이상의 색인을 정의할 수도 있습니다. 이전 예에서는 아래 항목을 추가할 수 있습니다.

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=Sort sn givenname uid, cn=Browsing ou=People,
   cn=databaseName,cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: vlvIndex
cn: Sort sn givenname uid
vlvSort: sn givenname uid
^D
```

찾아보기 색인 구성을 수정하려면 해당 `vlvSearch` 항목이나 `vlvIndex` 항목을 편집합니다. 서버에서 유지 관리하지 않도록 찾아보기 색인을 제거하려면 개별 `vlvIndex` 항목을 제거하거나, 이 항목이 하나뿐인 경우 `vlvSearch` 항목과 `vlvIndex` 항목을 모두 제거합니다. `vlvIndex` 항목을 제거할 때 해당 데이터베이스 파일도 제거할 수 있습니다. 예를 들면 다음과 같습니다.

```
ServerRoot/slapd-serverID/db/dbName/dbName_vlv#Sortsngivennameuid.db3
```

vlvindex 명령 실행

찾아보기 색인 항목을 새로 작성했거나 기존 항목을 수정한 경우 `directoryserver vlvindex` 명령을 실행하여 새 찾아보기 색인 집합을 생성해야 합니다. 이 명령은 디렉토리 내용을 검사하여 찾아보기 색인용 데이터베이스 파일을 작성합니다.

찾아보기 색인을 생성하려면 아래 명령을 실행합니다.

```
# /usr/sbin/directoryserver vlvindex
```

아래 예에서는 이전 절에서 정의한 찾아보기 색인을 생성합니다.

```
# /usr/sbin/directoryserver vlvindex -n databaseName -T "Browsing ou=People"
```

표 10-4 위의 예에 사용된 `vlvindex` 옵션에 대한 설명

옵션	설명
-n	색인화할 항목이 있는 데이터베이스 이름을 지정합니다.
-T	해당 찾아보기 색인의 <code>vlvSearch</code> 항목에 포함된 이름 지정 속성 값을 지정합니다. 지정된 <code>vlvSearch</code> 항목의 <code>vlvIndex</code> 항목에 해당하는 모든 색인이 생성됩니다.

명령에 대한 자세한 내용은 *Directory Server Man Page Reference*를 참조하십시오.

인증 및 암호화 관리

Directory Server는 네트워크를 통한 안전하고 신뢰할 수 있는 통신을 제공하기 위한 몇 가지 기법을 제공합니다. LDAPS는 SSL(Secure Sockets Layer)에 추가로 실행되는 표준 LDAP 프로토콜로, 데이터를 암호화하고 인증서를 사용하여(선택 사항) 인증합니다.

Directory Server는 Start TLS(Start Transport Layer Security) 확장 작업을 지원하므로 기존에 암호화되지 않은 LDAP 연결에서 TLS를 사용할 수 있습니다.

Directory Server에서는 SASL(Simple Authentication and Security Layer)을 통한 GSSAPI(Generic Security Services API)도 지원합니다. Solaris 운영 체제에서 Kerberos 버전 5 보안 프로토콜을 사용할 수 있습니다. 이 경우 아이디 매핑 기법이 커버로스 사용자를 디렉토리 아이디와 연결합니다.

추가 보안 정보에 대해서는 다음 위치의 NSS 웹 사이트를 참조하십시오.

<http://www.mozilla.org/projects/security/pki/nss/>

이 장은 다음 내용으로 구성되어 있습니다.

- [Directory Server의 SSL 소개](#)
- [SSL 활성화 단계 요약](#)
- [서버 인증서 얻기 및 설치](#)
- [SSL 활성화](#)
- [클라이언트 인증 구성](#)
- [아이디 매핑](#)
- [LDAP 클라이언트에서 보안을 사용하도록 구성](#)

Directory Server의 SSL 소개

SSL(Secure Sockets Layer)은 Directory Server와 클라이언트 간에 암호화된 통신과 인증(선택 사항)을 제공합니다. LDAP와 DSML-over-HTTP 프로토콜 모두에 대해 SSL을 활성화하여 모든 서버 연결에 보안을 제공할 수 있습니다. 또한 복제 및 연결 접미어 기법을 구성하여 서버 간의 보안 통신에 SSL을 사용할 수도 있습니다.

SSL을 단순 인증(바인드 DN과 비밀번호)과 함께 사용하면 서버 간에 전송되는 모든 데이터가 암호화되므로 기밀성과 데이터 무결성이 보장됩니다. 선택 사항으로, 클라이언트는 인증서를 사용하여 Directory Server 또는 SASL(Simple Authentication and Security Layer)을 통한 타사 보안 기법에 인증할 수 있습니다. 인증서 기반 인증에서는 클라이언트나 서버를 사칭하지 못하도록 공개 키 암호화를 사용합니다.

Directory Server는 별도의 포트에서 SSL 통신과 비SSL 통신을 동시에 지원합니다. 보안상의 이유로 모든 통신을 보안 포트에 제한할 수도 있습니다. 클라이언트 인증은 구성이 가능하며, 필수 또는 선택 사항으로 구성되어 실행할 보안 수준을 결정할 수 있습니다.

SSL을 사용하면 일반 LDAP 연결에 보안을 제공하는 Start TLS 확장 작업도 지원됩니다. 클라이언트는 비SSL 포트에 바인드한 후에 TLS(Transport Layer Security) 프로토콜을 사용하여 SSL 연결을 시작할 수 있습니다. Start TLS 작업은 클라이언트의 유연성을 높이며 포트 할당을 용이하게 합니다.

SSL에서 제공하는 암호화 기법은 속성 암호화에도 사용됩니다. SSL을 사용할 경우 접미어에 속성 암호화를 구성하여 디렉토리에 저장된 데이터를 보호할 수 있습니다. 자세한 내용은 [103페이지의 "속성 값 암호화"](#)를 참조하십시오.

클라이언트의 SSL 및 인증서 사용에 따라 디렉토리 내용에 대한 액세스 제어를 구성하여 보안을 강화할 수 있습니다. 특정 인증 방법이 필요한 액세스 제어 명령(ACI)을 정의할 수 있으며, 이 경우 보안 채널을 통해서만 데이터가 전송됩니다. 자세한 내용은 [254페이지의 "바인드 규칙"](#)을 참조하십시오.

SSL, 인터넷 보안, 인증서 및 Administration Server에서 SSL을 구성하는 방법 등에 대한 자세한 내용은 *Administration Server Administration Guide*를 참조하십시오.

SSL 활성화 단계 요약

각각의 단계에 대해서는 이 장의 후속 절에서 설명합니다.

1. 인증서를 얻어 Directory Server에 설치하고 Directory Server에서 인증 기관의 인증서를 트러스트하도록 구성합니다. 이 절차에는 다음과 같은 작업이 포함됩니다.
 - a. 필요한 경우 인증서 데이터 베이스 작성
 - b. 서버에서 인증서 요청을 생성하여 서버 인증서를 제공할 인증 기관으로 전송
 - c. 서버에 새 인증서 설치
 - d. 인증 기관과 인증 기관이 발급한 모든 인증서 트러스트
2. LDAP 및 DSML 작업을 위한 보안 포트를 포함하여 디렉토리에서 SSL을 활성화하고 구성합니다. Directory Server 콘솔에서 SSL을 사용하여 서버에 액세스하도록 구성할 수도 있습니다.
3. 선택 사항으로, 다음 클라이언트 인증 기법 중에서 하나 이상을 서버에 구성합니다.
 - a. 인증서 기반의 기본 인증 기법
 - b. SASL을 통한 DIGEST_MD5 인증 기법
 - c. Kerberos V5 보안 기법의 사용을 허용하는 SASL을 통한 GSSAPI 인증

이 단계를 완료한 다음에는 선택 사항으로 사용할 인증 기법을 포함하여 클라이언트에서 Directory Server와의 통신에 SSL을 사용하도록 구성합니다. `ldapsearch` 및 `ldapmodify`와 같은 도구를 참조하십시오.

명령줄을 통해 인증서를 관리하려면 `certutil` 도구를 사용하여 위의 단계 중 일부를 수행할 수도 있습니다. 이 도구는 `SUNWt1su` 패키지에 포함되어 있습니다.

서버 인증서 얻기 및 설치

이 절에서는 인증서 데이터베이스를 작성하고, Directory Server에서 사용할 인증서를 얻어서 설치하며, Directory Server에서 인증 기관(CA)의 인증서를 트러스트하도록 구성하는 절차에 대해 설명합니다.

인증서 데이터베이스 작성

서버에 처음 SSL을 구성할 때 보안 장치 비밀번호를 설정해야 합니다. 외부 하드웨어 보안 장치를 사용하지 않는 경우 아래 파일에 저장된 인증서 및 키 데이터베이스가 내부 보안 장치가 됩니다.

```
ServerRoot/alias/slaped-serverID-cert8.db  
ServerRoot/alias/slaped-serverID-key3.db
```

*serverID*에 대문자가 포함되어 있으면 아래의 명령줄 절차를 사용하여 인증서 데이터베이스를 작성해야 합니다.

콘솔 사용

콘솔에서 인증서 관리자 대화 상자를 처음 실행하면 인증서 데이터베이스 파일이 자동으로 작성됩니다.

1. Directory Server 콘솔의 최상위 태스크 탭에서 인증서 관리 버튼을 누릅니다. 또는 태스크 탭을 표시하고 콘솔 > 보안 메뉴에서 인증서 관리 항목을 선택합니다.
2. 서버에서 인증서 및 키 데이터베이스를 자동으로 작성하며, 보안 장치 비밀번호를 설정하라는 메시지가 표시됩니다. 이 비밀번호는 서버에 저장된 인증서의 개인 키를 보호합니다. 비밀번호를 두 번 입력하여 확인한 다음 확인을 누릅니다.

명령줄 사용

명령줄에서 인증서 데이터베이스 파일을 작성하는 경우 서버가 이 파일을 찾을 수 있도록 아래 절차에 제공된 경로 및 파일 이름 접두어를 사용해야 합니다.

1. 서버 호스트 시스템에서 아래 명령을 실행하여 인증서 데이터베이스를 작성합니다.
`certutil -N -d ServerRoot/alias -P slaped-LCserverID`

여기서 *LCserverID*는 모두 소문자로 지정된 서버 이름입니다.

인증서 키를 보호할 비밀번호를 입력하라는 메시지가 표시됩니다.

인증서 요청 생성

PEM 형식으로 PKCS #10 인증서 요청을 생성하려면 아래 절차 중 하나를 사용합니다. PEM은 RFC 1421부터 1424 (<http://www.ietf.org/rfc/rfc1421.txt>)까지에 지정된 Privacy Enhanced Mail 형식으로, base64 인코딩된 인증서 요청을 US-ASCII 문자로 나타내는 데 사용됩니다. 요청 내용은 아래 예와 같이 표시됩니다.

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBrjCCARcCAQAwbjELMAkGA1UBhMCVXMxEzARBgNVBAgTCkNBElGT1JOSUExLD
AqBgVBAoTl25ldHNjYXB1IGNvb11bmljYXRpb25zIGNvcnBvcnF0aWwMRwwGgYDV
QQDExNtZWxsb24umV0c2NhcGUuY29tMIGfMA0GCSqGSIb3DQEBAUAA4GNADCBiQK
BgCwAbskGh6SKY0gHy+UCSLnm3ok3X3u83Us7u0EfgSLR0f+K41eNqgWRftGR83e
mqPLDOf0ZLTLjVVGJaHJn411gG+Jdf/n/zMyahxtV7+T8GOFffigFfuxJaxMjr2j7I
vELlxQ4IfZgwgCm4qQecv3G+N9YdbjveMVXW0v4XwIDAQABAADQYJKoZIhvcNAQ
EEBQADgYEAZyZAm8UmP9PQYwNy4Pmypk79t2nvzKbwKVb97G+MT/gw1pLRsuBoKi
nMfLgKp1Q38K5Py2VGW1E47/rhm3yVQrIiwV+Z8Lcc=
-----END NEW CERTIFICATE REQUEST-----
```

콘솔 사용

1. Directory Server 콘솔의 최상위 태스크 탭에서 인증서 관리 버튼을 누릅니다. 또는 태스크 탭을 표시하고 콘솔 > 보안 메뉴에서 인증서 관리 항목을 선택합니다.
인증서 관리 대화 상자가 표시됩니다.
2. 서버 인증서 탭을 선택하고 요청 버튼을 누릅니다.
인증서 요청 마법사가 표시됩니다.
3. 서버에서 CA와 직접 통신할 수 있도록 지원하는 플러그 인을 설치한 경우 지금 선택할 수 있습니다. 그렇지 않으면 생성된 요청을 전자 메일이나 웹 사이트를 통해 전송하여 수동으로 인증서를 요청해야 합니다. 다음을 눌러 계속합니다.
4. 빈 텍스트 필드에 다음과 같은 요청자 정보를 입력합니다.

서버 이름. DNS 조회 시 사용되는 Directory Server의 전체 호스트 이름을 입력합니다(예: east.example.com).

조직. 회사나 기관의 공식 이름을 입력합니다. 대부분의 CA는 사업자 등록증 사본과 같은 공식 문서를 사용하여 이 정보를 확인하도록 요청합니다.

조직 구성 단위. (선택 사항) 회사 부서 또는 사업부를 잘 나타내는 이름을 입력합니다.

구/군/시. (선택 사항) 회사가 있는 구/군/시 이름을 입력합니다.

시/도. 약어를 사용하지 말고 회사가 있는 시/도의 전체 이름을 입력합니다.

국가. 두 자로 된 ISO 형식의 국가 이름 약어를 선택합니다. 미국의 국가 코드는 US입니다. *Directory Server Administration Reference*에는 ISO 국가 코드 목록이 있습니다.

다음을 눌러 계속합니다.

5. 보안 장치 비밀번호를 입력하고 다음을 누릅니다. 이 비밀번호는 [430페이지의 "인증서 데이터베이스 작성"](#)에서 설정한 비밀번호입니다.
6. 클립보드로 복사 또는 파일에 저장을 선택하여 인증 기관에 보내야 하는 인증서 요청 정보를 저장합니다.
7. 완료를 눌러 인증서 요청 마법사를 닫습니다.

명령줄 사용

1. 아래 명령을 실행하여 서버 인증서 요청을 작성합니다.

```
certutil -R \  
-s "cn=serverName,ou=division,o=company,l=city,st=state,c=country" \  
-a -d ServerRoot/alias -P slapd-serverID-
```

-s 옵션은 요청된 서버 인증서의 DN을 지정합니다. 대체로 인증 기관은 서버를 정확하게 식별하기 위해 이 예에 사용된 속성을 모두 필요로 합니다. 각 속성에 대한 자세한 내용은 위의 [단계 4](#)를 참조하십시오.

2. certutil 도구로부터 서버의 키 데이터베이스 비밀번호를 입력하라는 메시지가 표시됩니다. 이 비밀번호는 [430페이지의 "인증서 데이터베이스 작성"](#)에서 설정한 비밀번호입니다. 그런 후에 이 도구는 PEM 인코딩된 텍스트 형식으로 PKCS #10 인증서 요청을 생성합니다.

서버 인증서 설치

인증 기관의 절차에 따라 이전 절에서 생성한 인증서 요청을 인증 기관으로 전송합니다. 예를 들어, 전자 메일로 인증서 요청을 보내야 하는 경우도 있고 CA 웹 사이트에서 요청을 입력할 수 있는 경우도 있습니다.

요청을 보낸 후에는 CA에서 이 요청에 대한 응답으로 인증서를 보내줄 때까지 기다려야 합니다. 요청에 대한 응답 시간은 경우에 따라 달라집니다. 예를 들어, 회사 내부의 CA인 경우 하루나 이틀이면 요청에 대한 응답을 받을 수 있습니다. 회사 외부의 CA를 선택하면 요청에 대한 응답을 받을 때까지 몇 주가 걸릴 수도 있습니다.

CA로부터 받은 응답 정보는 반드시 텍스트 파일에 저장해야 합니다. PEM 형식의 PKCS #11 인증서는 아래 예와 같이 표시됩니다.

```
-----BEGIN CERTIFICATE-----
MIICjCCA ZugAwIBAgICCEEWdQYJKoZIhKqvcNAQFBQAwfDELMakGA1UEBhMCMVVMx
IzAhBgNVBAoGlBhbG9a2FWaWxsZGwSBXaWRnZXRzLCBjb2MwMR0wGwYDVQQLEExRX
aWRnZXQgTW3FrZXJzICdSjyBVczEpMCCGAX1UEAxgVGVzdCBUXN0IFRlc3QgVGVz
dCBUZXR0IFRlc3QgQ0EswHhcNOTgwMzEyMDIzMDIzMDIzMDIzMDIzMDIzMDIzMDIz
MQswCYDDVQQGEWJVUzEoMCMYGA1UEChMfTmV0c2NhcGUgRGlyZn0b3J5VlFB1YmXP
Y2F0aW9uczEwMB4QGA1UEAxMNZHVh49dq2tLnVbjTBaMA0GCsGSIb3DQEBAQUA
A0kAMEYkCQksMR/aLgdfp4m00iGgijG5Kg0syRNvWGYW7kfw+8mmijDtZarjYnj
jcgpf3VnlbxbclX9LVjjNLC5737XZdAgEDozYwpNDARBg1ghkgBhvCEAQEEBAMC
APAwHkYDVR0jBBgwFAU67URjwCaGqZHUUpSpdLxlzWJKiMwDQYJKoZIhKqvcNAQEF
BQADgYEAJ+BfVem3vBOPBvenDLGfj1b9hucgmaMcQa9FA/db8qimKT/ue9UGOJqL
bwbMKBBopsDn56p2yV3PLIsBgrcuSoBCuFFnxBngSiTS7YiYgCWqWauA0ExJFmD6
6hBLseqkSWulk+hXHN7L/NrViO+7zNtKcaZLlFPf7d7j2MgX4Bo=
-----END CERTIFICATE-----
```

인증서 데이터도 안전한 장소에 백업해 두어야 합니다. 시스템에 저장된 인증서 데이터가 손실될 경우 백업 파일을 사용하여 인증서를 다시 설치할 수 있습니다.

서버 인증서가 있으면 즉시 서버의 인증서 데이터베이스에 설치할 수 있습니다.

콘솔 사용

1. Directory Server 콘솔의 최상위 태스크 탭에서 인증서 관리 버튼을 누릅니다. 또는 태스크 탭을 표시하고 콘솔 > 보안 메뉴에서 인증서 관리 항목을 선택합니다.

인증서 관리 창이 표시됩니다.

2. "서버 인증서" 탭을 선택하고 "설치"를 누릅니다.

인증서 설치 마법사가 표시됩니다.

3. 다음 인증서 위치 옵션 중에서 하나를 선택합니다.

인증서가 있는 파일. 이 필드에는 인증서의 절대 경로를 입력합니다.

다음 인코딩된 텍스트 블록. 사용자가 작성한 텍스트 파일 또는 인증 기관의 텍스트를 복사하여 이 필드에 붙여넣습니다. 예를 들면 다음과 같습니다.

다음을 눌러 계속합니다.

4. 표시된 인증서 정보가 올바른지 확인하고 다음을 누릅니다.
5. 인증서 이름을 지정하고 다음을 누릅니다. 이 이름이 인증서 테이블에 표시됩니다.
6. 개인 키를 보호하는 비밀번호를 입력하여 인증서를 확인합니다. 이 비밀번호는 [430 페이지의 "인증서 데이터베이스 작성"의 단계 2](#)에서 입력한 비밀번호와 같아야 합니다. 작업이 끝나면 완료를 누릅니다.

서버 인증서 탭의 목록에 새 인증서가 표시됩니다. 이제 서버에서 SSL을 사용할 수 있습니다.

명령줄 사용

1. 아래 명령을 실행하여 인증서 데이터베이스에 새 서버 인증서를 설치합니다.

```
certutil -A -n "certificateName" -t "u,," -a -i certFile \  
-d ServerRoot/alias -P slapd-serverID-
```

여기서 *certificateName*은 인증서를 식별하기 위한 이름이고, *certFile*은 PEM 형식으로 PKCS #11 인증서에 포함된 텍스트 파일입니다. `-t "u,,"` 옵션은 이 인증서가 SSL 통신용 서버 인증서임을 나타냅니다.

2. 선택 사항으로, 아래의 certutil 명령을 실행하여 설치된 인증서를 확인할 수도 있습니다.

```
certutil -L -d ServerRoot/alias -P slapd-serverID-
```

`u, ,` 트러스트 속성을 사용하면 서버 인증서가 열거됩니다.

인증 기관 트러스트

Directory Server에서 인증 기관을 트러스트하도록 구성하려면 인증서를 얻어 서버의 인증서 데이터베이스에 설치해야 합니다. 이 프로세스는 사용하는 인증 기관에 따라 달라집니다. 인증서를 자동으로 다운로드할 수 있는 웹 사이트를 제공하는 민간 CA도 있고, 요청 시 전자 메일로 인증서를 보내주는 CA도 있습니다.

콘솔 사용

CA 인증서가 있으면 인증서 설치 마법사를 사용하여 Directory Server에서 인증 기관을 트러스트하도록 구성할 수 있습니다.

1. Directory Server 콘솔의 최상위 태스크 탭에서 인증서 관리 버튼을 누릅니다. 또는 태스크 탭을 표시하고 콘솔 > 보안 메뉴에서 인증서 관리 항목을 선택합니다.
인증서 관리 창이 표시됩니다.
2. CA 인증서 탭을 선택하고 설치를 누릅니다.
인증서 설치 마법사가 표시됩니다.
3. CA 인증서를 파일에 저장한 경우 제공된 필드에 파일 경로를 입력합니다. 전자 메일을 통해 CA 인증서를 받은 경우 헤더를 포함한 인증서 전체를 복사하여 제공된 텍스트 필드에 붙여넣습니다. 다음을 누릅니다.
4. 표시된 인증서 정보가 인증 기관에 맞는지 확인하고 다음을 누릅니다.
5. 인증서 이름을 지정하고 다음을 누릅니다.
6. 이 CA를 트러스트하는 목적을 선택합니다. 다음 중 하나 또는 둘 모두를 선택할 수 있습니다.

클라이언트의 연결 허용(클라이언트 인증). LDAP 클라이언트가 이 CA에서 발급한 인증서를 제공하여 인증서 기반의 클라이언트 인증을 수행하는 경우 이 확인란을 선택합니다.

다른 서버에 연결(서버 인증). 서버가 이 CA에서 발급한 인증서가 있는 다른 서버에 대해 SSL을 통한 복제 공급자 또는 연결 멀티플렉서 역할을 할 경우 이 확인란을 선택합니다.

7. 완료를 눌러 마법사를 닫습니다.

명령줄 사용

1. 아래 명령을 실행하여 트러스트할 수 있는 CA 인증서를 설치할 수도 있습니다.

```
certutil -A -n "CAcertificateName" -t "trust,," -a -i certFile \  
-d ServerRoot/alias -P slapd-serverID-
```

여기서 *CAcertificateName*은 트러스트할 수 있는 CA를 식별하기 위해 사용자가 지정한 이름이고 *certFile*은 PEM 인코딩된 텍스트 형식의 PKCS #11 CA 인증서가 포함된 텍스트 파일이며, *trust*는 다음 코드 중 하나입니다.

- o T- 이 CA는 클라이언트 인증서를 발급하도록 트러스트되었습니다. LDAP 클라이언트가 이 CA에서 발급한 인증서를 제공하여 인증서 기반의 클라이언트 인증을 수행하는 경우 이 코드를 사용합니다.

- C-이 CA는 서버 인증서를 발급하도록 트러스트되었습니다. 서버가 이 CA에서 발급한 인증서가 있는 다른 서버에 대해 SSL을 통한 복제 공급자 또는 연결 멀티플렉서 역할을 할 경우 이 코드를 사용합니다.
 - CT-이 CA는 클라이언트 인증서와 서버 인증서를 모두 발급하도록 트러스트되었습니다. 위의 두 경우가 모두 적용되면 이 코드를 사용합니다.
2. 선택 사항으로, 아래의 certutil 명령을 실행하여 설치된 인증서를 확인할 수도 있습니다.

```
certutil -L -d ServerRoot/alias -P slapd-serverID-
```

u,, 트러스트 속성을 사용하면 서버 인증서가 열거되고 CT,, 속성을 사용하면 트러스트할 수 있는 CA 인증서가 열거됩니다.

SSL 활성화

서버 인증서 설치와 CA 인증서 트러스트가 완료되면 SSL을 활성화할 수 있습니다. 대체로 SSL을 사용하여 서버를 실행하는 것이 좋습니다. 일시적으로 SSL을 비활성화한 경우 기밀성, 인증 또는 데이터 무결성이 필요한 작업을 처리하기 전에 SSL을 다시 활성화해야 합니다.

SSL을 활성화하려면 먼저 [429페이지](#)의 "서버 인증서 얻기 및 설치"에 설명된 것처럼 인증서 데이터베이스를 작성하고 서버 인증서를 얻어서 설치한 후에 CA 인증서를 트러스트해야 합니다.

그런 다음, 아래 절차에 따라 Directory Server에서 SSL 통신을 활성화하고 암호화 체계를 사용합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 이 서버 이름이 포함된 루트 노드를 선택한 다음 오른쪽 창에서 암호화 탭을 선택합니다.
현재의 서버 암호화 설정이 탭에 표시됩니다.
2. "이 서버에 SSL 사용" 확인란을 선택하여 암호화를 사용하도록 지정합니다.
3. "사용할 암호 패밀리" 확인란을 선택합니다.
4. 드롭다운 메뉴에서 사용할 인증서를 선택합니다.
5. 암호 설정을 눌러 암호 기본 설정 대화 상자에서 사용할 암호를 선택합니다. 특정 암호에 대한 자세한 내용은 [438페이지](#)의 "암호화 암호 선택"을 참조하십시오.

6. 클라이언트 인증에 대한 기본 설정을 설정합니다.

클라이언트 인증 허용 안 함. 이 옵션을 사용하면 서버는 클라이언트의 인증서를 무시하고 해당 인증서에 기반한 인증을 거부합니다.

클라이언트 인증 허용. 이 옵션이 기본 설정입니다. 이 옵션을 사용하면 클라이언트 요청에 대한 인증이 수행됩니다. 인증서 기반의 인증에 대한 자세한 내용은 [440페이지](#)의 "[클라이언트 인증 구성](#)"을 참조하십시오.

주 복제 시 인증서 기반의 인증을 사용하는 경우 소비자 서버에서 클라이언트 인증을 허용하거나 필요로 하도록 구성해야 합니다.

클라이언트 인증 필요. 이 옵션을 사용하면 클라이언트가 서버의 인증 요청에 응답하지 않을 경우 클라이언트 연결이 거부됩니다.

주 서버 콘솔이 SSL을 통해 Directory Server에 연결하는 경우 "클라이언트 인증 필요"를 선택하면 클라이언트 인증에 사용할 인증서가 서버 콘솔에 없기 때문에 통신이 비활성화됩니다. 명령줄에서 이 속성을 수정하려면 [440페이지](#)의 "[클라이언트 인증 허용](#)"을 참조하십시오.

7. 선택 사항으로, 콘솔이 SSL을 통해 Directory Server와 통신하도록 설정하려면 "서버 콘솔에 SSL 사용"을 선택합니다.
8. 작업이 끝나면 저장을 누릅니다.
9. 선택 사항으로, 서버에서 LDAP 및 DSML-over-HTTP 프로토콜 모두의 SSL 통신에 사용할 보안 포트를 설정합니다. 자세한 내용은 [49페이지](#)의 "[Directory Server의 포트 번호 변경](#)"을 참조하십시오.
- 보안 포트에 대한 모든 연결은 SSL을 사용해야 합니다. SSL을 활성화하면 보안 포트 구성 여부에 관계 없이 클라이언트에서 Start TLS 작업을 사용하여 비보안 포트를 통한 SSL 암호화를 수행할 수 있습니다.
10. Directory Server를 다시 시작합니다.
- 자세한 내용은 [36페이지](#)의 "[SSL을 활성화하여 서버 시작](#)"을 참조하십시오.

암호화 암호 선택

암호는 데이터를 암호화하고 암호를 해독하는 데 사용하는 알고리즘입니다. 일반적으로 암호는 암호화 중에 사용하는 비트 수가 많을수록 더 강력하거나 안전합니다. SSL 암호는 사용된 메시지 인증 유형으로도 식별할 수 있습니다. 메시지 인증은 데이터 무결성을 보장하는 체크섬을 계산하는 별개의 알고리즘입니다. 암호 알고리즘 및 각 장점에 대한 자세한 내용은 *Administration Server Administration Guide*를 참조하십시오.

클라이언트가 서버와의 SSL 연결을 시작하려면 클라이언트 및 서버가 정보 암호화에 사용할 암호에 동의해야 합니다. 양방향 암호화 프로세스의 경우 클라이언트와 서버 모두 지원되는 가장 강력한 암호를 동일하게 사용해야 합니다.

Directory Server는 다음과 같은 SSL 3.0 및 TLS용 암호를 제공합니다.

표 11-1 Directory Server에서 제공하는 암호

암호 이름	설명
None	암호화 없음. MD5 메시지 인증만 사용(rsa_null_md5)
RC4(128비트)	128비트 암호화 및 MD5 메시지 인증을 사용하는 RC4 암호 (rsa_rc4_128_md5)
RC4(Export)	40비트 암호화 및 MD5 메시지 인증을 사용하는 RC4 암호 (rsa_rc4_40_md5)
RC2(Export)	40비트 암호화 및 MD5 메시지 인증을 사용하는 RC2 암호 (rsa_rc2_40_md5)
DES 또는 Export DES	56비트 암호화 및 SHA 메시지 인증을 사용하는 DES(rsa_des_sha)
DES(FIPS)	56비트 암호화 및 SHA 메시지 인증을 사용하는 FIPS DES. 이 암호는 암호화 모듈 구현을 위한 미국 정부 표준인 FIPS 140-1(rsa_fips_des_sha)을 준수합니다.
Triple-DES	168비트 암호화 및 SHA 메시지 인증을 사용하는 Triple DES(rsa_3des_sha)
Triple-DES(FIPS)	168비트 암호화 및 SHA 메시지 인증을 사용하는 FIPS Triple DES. 이 암호는 암호화 모듈 구현을 위한 미국 정부 표준인 FIPS 140-1(rsa_fips_3des_sha)을 준수합니다.
Fortezza	80비트 암호화 및 SHA 메시지 인증을 사용하는 Fortezza 암호
RC4(Export)	128비트 암호화 및 SHA 메시지 인증을 사용하는 Fortezza RC4 암호
없음(Fortezza)	암호화 없음, Fortezza SHA 메시지 인증만 사용

서버 콘솔에서 계속 SSL을 사용하려면 최소한 다음 암호 중 하나를 선택해야 합니다.

- 40비트 암호화 및 MD5 메시지 인증을 사용하는 RC4 암호
- 암호화 없음, MD5 메시지 인증만 사용(권장되지 않음)
- 56비트 암호화 및 SHA 메시지 인증을 사용하는 DES
- 128비트 암호화 및 MD5 메시지 인증을 사용하는 RC4 암호
- 168비트 암호화 및 SHA 메시지 인증을 사용하는 Triple DES

서버에서 사용할 암호를 선택하려면 아래 절차에 따라 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 이 서버 이름이 포함된 루트 노드를 선택한 다음 오른쪽 창에서 암호화 탭을 선택합니다.
현재의 서버 암호화 설정이 탭에 표시됩니다. [436페이지의 "SSL 활성화"](#)에 설명된 것처럼 서버에서 SSL을 활성화해야 합니다.
2. 암호 설정을 누릅니다.
암호 기본 설정 대화 상자가 표시됩니다.
3. 암호 기본 설정 대화 상자에서 이름 옆에 있는 확인란을 선택하거나 선택 취소하여 서버에서 사용할 암호를 지정합니다.
보안상 특정 암호를 사용하지 않는 경우가 아니면 none, MD5를 제외한 모든 암호를 선택해야 합니다.

주의 암호화 없음, MD5 메시지 인증만 사용 암호 옵션은 선택하지 마십시오. 서버는 클라이언트에서 다른 암호를 사용할 수 없을 때 이 옵션을 사용합니다. 이 경우 암호화가 사용되지 않으므로 연결이 안전하지 않습니다.

4. 암호 기본 설정 대화 상자에서 확인을 누른 다음 암호화 탭에서 저장을 누릅니다.

클라이언트 인증 허용

Directory Server에 클라이언트 인증이 *필요*하고 서버 콘솔이 SSL을 사용하여 연결하도록 구성한 경우에는 더 이상 서버 콘솔을 사용하여 Sun Java System 서버를 관리할 수 없습니다. 이 경우 해당 명령줄 유틸리티를 사용해야 합니다.

서버 콘솔을 사용할 수 있도록 디렉토리 구성을 변경하려면 다음 단계에 따라 클라이언트 인증을 *필수*가 아닌 *선택 사항*으로 설정해야 합니다.

1. 아래 명령을 실행하여 cn=encryption, cn=config 항목을 수정합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=encryption, cn=config
changetype: modify
replace: nsSSLClientAuth
nsSSLClientAuth: allowed
^D
```

2. 35페이지의 "명령줄에서 서버 시작 및 중지"에 설명된 것처럼 Directory Server를 다시 시작합니다.

이제 서버 콘솔을 시작할 수 있습니다.

클라이언트 인증 구성

클라이언트 인증은 서버에서 클라이언트 아이디를 확인하는 기법입니다. 클라이언트가 제공한 인증서를 사용하거나 DIGEST-MD5 같은 SASL 기반의 기법을 통해 클라이언트 인증을 간단하게(DN과 비밀번호를 제공하여) 수행할 수 있습니다. Solaris 운영 체제의 경우 Directory Server에서 SASL을 통한 GSSAPI 기법을 지원하므로 Kerberos V5를 통해 클라이언트 인증을 허용할 수 있습니다.

인증서 기반의 인증 시에는 SSL 프로토콜을 통해 얻은 클라이언트 인증서를 사용하여 식별할 사용자 항목을 찾습니다. 이 기법은 낮은 계층이 이미 설정되어 있는 인증 기법에 의존하기 때문에 EXTERNAL이라고도 합니다. (차후 릴리스에서는 외부 인증을 IP 보안 프로토콜(ipsec)과 함께 사용할 수 있습니다.)

인증서 기반의 인증은 *Administration Server Administration Guide*에 자세히 설명되어 있습니다.

다음 절에서는 Directory Server에 두 개의 SASL 기법을 구성하는 방법에 대해 설명합니다. 448페이지의 "LDAP 클라이언트에서 보안을 사용하도록 구성"을 참조하십시오.

주 아이디 매핑은 연결 접미어를 지원하지 않습니다.

DIGEST-MD5를 통한 SASL 인증

DIGEST-MD5 기법은 클라이언트에서 보낸 해시 값을 사용자 비밀번호의 해시와 비교하여 클라이언트를 인증합니다. 하지만 이 기법은 사용자 비밀번호를 읽어야 하기 때문에 DIGEST-MD5를 통해 인증을 받으려는 사용자는 디렉토리에 {CLEAR} 비밀번호가 있어야 합니다. 디렉토리에 {CLEAR} 비밀번호를 저장하는 경우 6장, "액세스 제어 관리"에서 설명하는 것처럼, 비밀번호에 대한 액세스가 ACI를 통해 적절하게 제한되도록 해야 합니다. 103페이지의 "속성 값 암호화"에서 설명하는 것처럼 해당 접미어에 속성 암호화를 구성하여 {CLEAR} 비밀번호에 대한 보호를 더 강화할 수 있습니다.

DIGEST-MD5 기법 구성

아래 절차에서는 Directory Server에서 DIGEST-MD5를 사용하도록 구성하는 방법에 대해 설명합니다.

1. 콘솔이나 `ldapsearch` 명령을 사용하여 DIGEST-MD5가 루트 항목의 `supportedSASLMechanisms` 속성 값인지 확인합니다. 예를 들어 아래 명령은 현재 활성화되어 있는 SASL 기법을 보여줍니다.

```
ldapsearch -h host -p port -D "cn=Directory Manager" -w password \
-s base -b "" "(objectclass=*)" supportedSASLMechanisms
```

```
dn:
supportedSASLMechanisms: EXTERNAL
supportedSASLMechanisms: DIGEST-MD5
supportedSASLMechanisms: GSSAPI
^D
```

2. DIGEST-MD5가 활성화되어 있지 않으면 아래의 `ldapmodify` 명령을 사용하여 활성화합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=SASL, cn=security, cn=config
changetype: modify
add: dsSaslPluginsEnable
dsSaslPluginsEnable: DIGEST-MD5
-
replace: dsSaslPluginsPath
dsSaslPluginsPath: ServerRoot/lib/sasl
^D
```

3. 442페이지의 "DIGEST-MD5 아이디 매핑"에 설명된 것처럼 DIGEST-MD5에 기본 아이디 매핑을 사용하거나 새로운 아이디 매핑을 작성합니다.
4. DIGEST-MD5를 사용하여 SSL을 통해 서버에 액세스하는 모든 사용자의 비밀번호가 {CLEAR}에 저장되어 있는지 확인합니다. 비밀번호 저장 체계를 구성하는 방법은 7장, "사용자 계정 및 비밀번호 관리"를 참조하십시오.
5. SASL 구성 항목이나 DIGEST-MD5 아이디 매핑 항목 중 하나를 수정한 경우 Directory Server를 다시 시작합니다.

DIGEST-MD5 아이디 매핑

SASL 기법의 아이디 매핑은 SASL 아이디의 자격 증명을 디렉토리에 있는 사용자 항목과 일치시킵니다. 이 기법에 대한 자세한 내용은 446페이지의 "아이디 매핑"을 참조하십시오. 매핑 중에 SASL 아이디에 해당하는 DN을 찾을 수 없으면 인증은 실패합니다.

SASL 아이디는 각 기법의 고유 형식으로 사용자를 나타내는 사용자 문자열입니다. DIGEST-MD5의 경우, dn: 접두어와 LDAP DN 또는 u: 접두어 뒤에 클라이언트가 지정한 텍스트가 있는 사용자를 클라이언트에서 작성하는 것이 좋습니다. 클라이언트가 보낸 사용자는 매핑 중에 \${Principal} 자리 표시자에서 사용할 수 있습니다.

DIGEST-MD5의 기본 아이디 매핑은 서버 구성의 아래 항목에서 지정합니다.

```
dn: cn=default,cn=DIGEST-MD5,cn=identity mapping,cn=config
objectClass: top
objectClass: nsContainer
objectClass: dsIdentityMapping
objectClass: dsPatternMatching
cn: default
dsMatching-pattern: ${Principal}
dsMatching-regexp: dn:(.*)
dsMappedDN: $1
```

이 아이디 매핑에서는 사용자 dn 필드에 기존 디렉토리 사용자의 DN이 포함되어 있다고 가정합니다.

DIGEST-MD5에 대한 사용자 정의 아이디 매핑을 정의하려면 다음을 수행합니다.

1. 기본 매핑 항목을 편집하거나 `cn=DIGEST-MD5,cn=identity mapping,cn=config`에 새 매핑 항목을 작성합니다. 아이디 매핑 항목의 속성 정의에 대해서는 [446페이지의 "아이디 매핑"](#)을 참조하십시오. DIGEST-MD5 매핑 예는 아래 파일에 포함되어 있습니다.

```
ServerRoot/slapd-serverID/ldif/identityMapping_Examples.ldif
```

이 예에서는 정식 이름이 아닌 사용자 텍스트 필드에 원하는 아이디의 사용자 이름이 포함되어 있다고 가정합니다. 아래 명령은 이 매핑이 어떻게 정의되는지 보여줍니다.

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=unqualified-username,cn=DIGEST-MD5,cn=identity mapping,
   cn=config
objectclass: dsIdentityMapping
objectclass: dsPatternMatching
objectclass: nsContainer
objectclass: top
cn: unqualified-username
dsMatching-pattern: ${Principal}
dsMatching-regexp: u:(.*)@(.*).\com
dsSearchBaseDN: dc=$2
dsSearchFilter: (uid=$1)
```

2. Directory Server를 다시 시작하여 새 매핑을 적용합니다.

GSSAPI를 통한 SASL 인증(Solaris에만 해당)

SASL을 통한 GSSAPI(Generic Security Services API)를 사용하면 Kerberos V5와 같은 타사 보안 시스템을 통해 클라이언트를 인증할 수 있습니다. GSSAPI 라이브러리는 Solaris SPARC 기반의 플랫폼에서만 사용할 수 있습니다. SEAM(Sun Enterprise Authentication Mechanism) 1.0.1 서버에 Kerberos V5 구현을 설치하는 것이 좋습니다.

서버는 이 API를 사용하여 사용자 아이디를 검증합니다. 그런 후에 SASL 기법에서 GSSAPI 매핑 규칙을 적용하여 이 연결이 유지되는 동안 모든 작업의 바인드 DN으로 지정될 DN을 얻습니다.

커버로스 시스템 구성

제조업체의 지침에 따라 커버로스 소프트웨어를 구성합니다. SEAM 1.0.1 서버를 사용하는 경우 아래 단계도 수행해야 합니다.

1. `/etc/krb5`에 있는 파일을 구성합니다.

2. 사용자와 서비스를 저장할 커버로스 데이터베이스를 작성한 다음, 여기에 LDAP 서비스 사용자를 작성합니다. LDAP 서비스 사용자는 다음과 같습니다.

`ldap/serverFQDN@REALM`

여기서 `serverFQDN`은 Directory Server의 정규화된 도메인 이름입니다.

3. 커버로스 데몬 프로세스를 시작합니다.

DNS는 호스트 시스템에 구성해야 합니다.

자세한 단계별 지침은 소프트웨어 설명서를 참조하십시오. [458페이지의 "SASL이 있는 GSSAPI를 사용하여 커버로스 인증 구성: 절차 예"](#) 예도 함께 참조하십시오.

GSSAPI 기법 구성

아래 절차에서는 Solaris 플랫폼의 Directory Server에서 GSSAPI를 사용하도록 구성하는 방법에 대해 설명합니다.

1. [444페이지의 "GSSAPI 아이디 매핑"](#)에 설명된 것처럼 GSSAPI에 대한 기본 아이디 매핑과 사용자 정의 매핑을 작성합니다.

2. LDAP 서비스 키 등의 서비스 키를 저장할 키 탭을 작성합니다.

- 키 탭은 Directory Server 사용자만 읽을 수 있어야 합니다.
- 파일 이름을 기본 `/etc/krb5/krb5.keytab`과 다르게 지정하십시오.

`start-slapd` 스크립트에서 환경 변수 `KRB5_KTNAME`을 설정하여 기본값 대신 새로운 키 탭을 사용하도록 합니다.

3. SASL 구성 항목이나 GSSAPI 아이디 매핑 항목 중 하나를 수정한 경우 Directory Server를 다시 시작합니다.

DNS는 호스트 시스템에 구성해야 합니다.

GSSAPI 아이디 매핑

SASL 기법의 아이디 매핑은 SASL 아이디의 자격 증명을 디렉토리에 있는 사용자 항목과 일치시킵니다. 이 기법에 대한 자세한 내용은 [446페이지의 "아이디 매핑"](#)을 참조하십시오. 매핑 중에 SASL 아이디에 해당하는 DN을 찾을 수 없으면 인증은 실패합니다.

SASL 아이디는 각 기법의 고유 형식으로 사용자를 나타내는 *사용자* 문자열입니다. GSSAPI를 사용하는 커버로스에서 사용자는 `uid[/instance][@realm]` 형식의 아이디로 나타납니다. 여기서 `uid`에는 `instance` 식별자가 선택 사항으로 포함될 수 있으며 이 식별자 뒤에 선택 사항인 `realm`이 옵니다. `realm`에는 일반적으로 도메인 이름이 포함됩니다. 예를 들어, 다음은 모두 유효한 사용자입니다.

```
bjensen
bjensen/Sales
bjensen@EXAMPLE.COM
bjensen/Sales@EXAMPLE.COM
```

처음에는 디렉토리에 GSSAPI 매핑이 정의되어 있지 않습니다. 클라이언트에서 사용자를 정의하는 방법에 따라 기본 매핑 및 필요한 사용자 정의 매핑을 모두 정의해야 합니다.

GSSAPI에 대한 아이디 매핑을 정의하려면 다음을 수행합니다.

1. `cn=GSSAPI`, `cn=identity mapping`, `cn=config`에 새 매핑 항목을 작성합니다. 아이디 매핑 항목의 속성 정의에 대해서는 [446페이지](#)의 "아이디 매핑"을 참조하십시오.

GSSAPI 매핑 예는 아래 파일에 포함되어 있습니다.

```
ServerRoot/slapd-serverID/ldif/identityMapping_Examples.ldif
```

이 파일에 제공된 기본 GSSAPI 매핑에서는 사용자에게 사용자 아이디만 포함되어 있으며, 이 아이디로 디렉토리의 고정 분기에 있는 사용자를 결정한다고 가정합니다.

```
dn: cn=default,cn=GSSAPI,cn=identity mapping,cn=config
objectclass: dsIdentityMapping
objectclass: nsContainer
objectclass: top
cn: default
dsMappedDN: uid=${Principal},ou=people,dc=example,dc=com
```

이 파일의 다른 예에서는 알려진 영역이 지정된 사용자에게 포함할 사용자 아이디를 결정하는 방법을 보여줍니다.

```
dn: cn=same_realm,cn=GSSAPI,cn=identity mapping,cn=config
objectclass: dsIdentityMapping
objectclass: dsPatternMatching
objectclass: nsContainer
objectclass: top
cn: same_realm
dsMatching-pattern: ${Principal}
dsMatching-regexp: (.*)@EXAMPLE.COM
dsMappedDN: uid=$1,ou=people,dc=EXAMPLE,dc=COM
```

2. Directory Server를 다시 시작하여 새로운 매핑을 적용합니다.

아이디 매핑

Directory Server에는 다른 프로토콜의 자격 증명을 디렉토리의 DN으로 매핑해야 하는 여러 가지 인증 매커니즘이 있습니다. DSML-over-HTTP 프로토콜과 DIGEST-MD5 및 GSSAPI SASL 기법도 이러한 경우에 해당합니다. 각 기법에서는 아이디 매핑을 사용하여 클라이언트가 제공한 프로토콜별 자격 증명에 따라 바인드 DN을 결정합니다.

아이디 매핑 중에는 `cn=identity mapping`, `cn=config` 구성 분기의 항목이 사용됩니다. 이 분기에는 아이디 매핑을 수행해야 하는 각 프로토콜의 컨테이너가 포함되어 있습니다.

- `cn=HTTP-BASIC`, `cn=identity mapping`, `cn=config-DSML-over-HTTP` 연결에 대한 매핑이 포함되어 있습니다.
- `cn=DIGEST-MD5`, `cn=identity mapping`, `cn=config-DIGEST-MD5 SASL` 기법을 사용하는 클라이언트 인증에 대한 매핑이 포함되어 있습니다.
- `cn=GSSAPI`, `cn=identity mapping`, `cn=config-GSSAPI SASL` 기법을 사용하는 클라이언트 인증에 대한 매핑이 포함되도록 작성해야 합니다.

매핑 항목은 디렉토리 검색에서 사용하기 위해 프로토콜별 자격 인증 요소를 추출하는 방법을 정의합니다. 검색 결과 한 개의 사용자 항목이 반환되면 매핑이 성공한 것이며, 연결이 유지되는 동안 이 항목을 모든 작업의 바인드 DN으로 사용합니다. 검색 결과 0 또는 두 개 이상의 항목이 반환되면 매핑이 실패하고 다른 매핑이 적용됩니다.

각 분기마다 해당 프로토콜의 기본 매핑과 사용자 정의 매핑이 포함되어야 합니다. 기본 매핑에는 `cn=default RDN`이 있고, 사용자 정의 매핑에는 이름 지정 속성으로 `cn`을 사용하는 다른 RDN이 있을 수 있습니다. 사용자 정의 매핑 중 하나가 성공할 때까지 모든 사용자 정의 매핑이 비결정적 순서로 먼저 평가됩니다. 모든 사용자 정의 매핑이 실패하면 마지막으로 기본 매핑이 적용됩니다. 기본 매핑도 실패하면 클라이언트 인증은 실패합니다.

매핑 항목에는 `top`, `Container` 및 `dsIdentityMapping` 객체 클래스가 있어야 하며, 이 경우 항목에는 다음과 같은 속성이 포함될 수 있습니다.

- `dsMappedDN`: DN - 디렉토리에 DN을 정의하는 리터럴 문자열입니다. 매핑을 수행할 때 이 DN이 있으면 해당 DN이 바인드에 사용됩니다. 해당 DN이 없을 때 검색을 수행할 다른 속성을 정의할 수도 있습니다.
- `dsSearchBaseDN`: DN - 검색할 기본 DN입니다. 이 속성을 지정하지 않으면 매핑에서 전체 디렉토리 트리의 모든 루트 접미어(모든 이름 지정 컨텍스트 포함, 단, `cn=config`, `cn=monitor` 및 `cn=schema` 제외)를 검색하게 됩니다.

- `dsSearchScope`: `base|one|sub` - 검색 기본 항목 자체, 기본 항목 아래의 하나의 하위 수준, 기본 항목 아래의 전체 하위 트리 중 하나인 검색 범위입니다. 이 속성을 지정하지 않으면 매핑 검색의 기본 범위는 전체 하위 트리입니다.
- `dsSearchFilter`: `filterString` - 매핑 검색을 수행할 필터 문자열입니다. LDAP 검색 필터는 RFC 2254(<http://www.ietf.org/rfc/rfc2254.txt>)에 정의되어 있습니다.

또한 매핑 항목에는 `dsPatternMatching` 객체 클래스가 포함되어 다음과 같은 속성의 사용을 허용할 수도 있습니다.

- `dsMatching-pattern`: `patternString` - 패턴 일치를 수행할 문자열을 지정합니다.
- `dsMatching-regexp`: `regularExpression` - 패턴 문자열에 적용할 정규 표현식을 지정합니다.

`dsSearchScope`를 제외한 위의 모든 속성 값에는 `${keyword}` 형식의 자리 표시자가 포함될 수 있습니다. 여기서 `keyword`는 프로토콜별 자격 증명에 있는 요소 이름입니다. 매핑 중에 자리 표시자는 클라이언트가 제공한 요소의 실제 값으로 대체됩니다.

모든 자리 표시자가 대체된 후에 정의된 패턴 일치기가 수행됩니다. 일치하는 패턴은 정규 표현식과 비교됩니다. 정규 표현식이 패턴 문자열과 일치하지 않으면 이 매핑은 실패합니다. 일치하면 괄호 안에 있는 정규 표현식 조건에 일치하는 값을 번호가 매겨진 다른 속성 값의 자리 표시자로 사용할 수 있습니다. 예를 들어, SASL에 아래 매핑을 정의할 수 있습니다.

```
dsMatching-pattern: ${Principal}
dsMatching-regexp: (.*)@(.*).\.(.*)
dsMappedDN: uid=$1,ou=people,dc=$2,dc=$3
```

클라이언트가 `bjensen@example.com` 사용자로 인증하면 이 매핑은 `uid=bjensen,ou=people,dc=example,dc=com` 바인드 DN을 정의합니다. 디렉토리에 이 DN이 있으면 매핑이 성공하여 클라이언트 인증이 이루어지고, 연결이 유지되는 동안 수행된 모든 작업은 이 바인드 DN을 사용합니다.

`dsMatching-pattern`은 Posix regexec (3C) 및 regcomp (3C) 함수 호출을 사용하여 `dsMatching-regexp`와 비교됩니다. Directory Server는 확장된 정규 표현식을 사용하며, 비교 시 대소문자를 구분하지 않습니다. 자세한 내용은 *Directory Server Man Page Reference*에서 이러한 함수에 대한 설명서 페이지를 참조하십시오.

자리 표시자를 포함할 수 있는 속성 값에서 자리 표시자로 사용되지 않은 \$, { 및 } 문자는 다른 자리 표시자를 사용하지 않아도 반드시 인코딩해야 합니다. 즉, 문자 \$는 \24로, {는 \7B로, }는 \7D로 인코딩해야 합니다.

자리 표시자와 대체를 사용하면 프로토콜별 자격 증명에서 사용자 이름과 다른 값을 추출하는 매핑을 작성하고 이 값을 사용하여 매핑된 DN을 정의하거나 디렉토리에서 해당 DN을 검색할 수 있습니다. 디렉토리 클라이언트가 제공한 예상 자격 증명을 추출하는 매핑을 정의하고 특정 디렉토리 구조에 이를 매핑해야 합니다.

주의 제대로 정의되지 않은 매핑을 작성하면 보안에 허점이 생길 수 있습니다. 예를 들어, 패턴 일치를 수행하지 않고 하드 코딩된 DN에 매핑하면 항상 매핑이 성공하기 때문에 디렉토리 사용자가 아닌 클라이언트를 인증할 수도 있습니다.

지나치게 광범위하고 허용 범위가 넓은 한 개의 매핑을 작성하는 것보다 다양한 클라이언트 자격 증명 형식을 처리하는 여러 개의 매핑을 정의하는 것이 더 안전합니다. 항상 클라이언트 자격 증명에 따라 클라이언트 연결을 특정 사용자에게 매핑해야 합니다.

LDAP 클라이언트에서 보안을 사용하도록 구성

다음 절에서는 Directory Server와 보안 연결을 구성하려는 LDAP 클라이언트에서 SSL을 구성 및 사용하는 방법에 대해 설명합니다. SSL 연결에서 서버는 해당 인증서를 클라이언트로 보냅니다. 클라이언트는 먼저 인증서를 트러스트하여 서버를 인증해야 합니다. 그런 후에 클라이언트는 두 SASL 기법(DIGEST-MD5 또는 Kerberos V5)를 사용하는 GSSAPI 중 하나에 대한 정보나 자신의 인증서를 보내 클라이언트 인증 기법 중 하나를 선택적으로 시작할 수 있습니다.

다음 절에서는 SSL을 사용하는 LDAP 클라이언트의 예로 `ldapsearch` 도구를 사용합니다. Directory Server와 함께 제공된 `ldapmodify`, `ldapdelete` 및 `ldapcompare` 도구는 모두 동일한 방법으로 구성됩니다. 이러한 디렉토리 액세스 도구는 Directory SDK for C에 기반을 두고 있으며, *Directory Server Resource Kit Tools Reference*에서 자세히 설명합니다.

다른 LDAP 클라이언트에서 SSL 연결을 구성하려면 응용 프로그램과 함께 제공된 설명서를 참조하십시오.

주 일부 클라이언트 응용 프로그램은 SSL만 구현하고 서버에 트러스트할 수 있는 인증서가 있는지 확인하지 않으므로 SSL 프로토콜을 사용하여 데이터 암호화는 제공하지만 기밀성이나 사칭에 대한 보호는 보장할 수 없습니다.

클라이언트에 서버 인증 구성

클라이언트는 서버와 SSL 연결을 구성할 때 서버에서 제공한 인증서를 트러스트해야 합니다. 이렇게 하려면 클라이언트는 다음과 같은 작업을 수행해야 합니다.

- 인증서 데이터베이스를 작성합니다.
- 서버 인증서를 발급하는 인증 기관(CA)을 트러스트합니다.
- LDAP 클라이언트의 SSL 옵션을 지정합니다.

Mozilla는 SSL을 사용하여 HTTP 프로토콜을 통해 웹 서버와 통신하는 클라이언트 응용 프로그램입니다. Mozilla를 사용하여 LDAP 클라이언트에서도 사용할 인증서를 관리할 수 있습니다. 또는 `certutil` 도구를 사용하여 인증서 데이터베이스를 관리할 수 있습니다.

Mozilla를 통한 클라이언트 인증서 관리

아래 절차에서는 Mozilla를 사용하여 클라이언트 시스템에서 인증서 데이터베이스를 관리하는 방법에 대해 설명합니다.

1. Mozilla는 시작과 동시에 인증서 데이터베이스가 있는지 확인하며, 필요한 경우 새 인증서 데이터베이스를 작성합니다. 인증서 데이터베이스는 예를 들어 `.mozilla/username/string.slt/cert8.db`와 같이 다른 Mozilla 기본 설정과 함께 파일에 저장됩니다.

이 절차를 사용할 경우 Mozilla에서 작성된 인증서 데이터베이스를 찾아 클라이언트 응용 프로그램에 사용할 경로를 기억해 둡니다.

2. Mozilla를 사용하여 액세스할 Directory Server 인증서를 발급한 인증 기관의 웹사이트를 탐색합니다. Mozilla에서 인증 기관의 인증서를 자동으로 검색하며, 트러스트 여부를 묻는 메시지를 표시합니다.

예를 들어, 내부용으로 배포된 Sun Java System Certificate Server를 사용하는 경우 `https://hostname:444` 형식의 URL을 방문합니다.

3. 인증서를 트러스트하라는 메시지가 Mozilla에 표시되면 인증 기관의 인증서를 트러스트합니다. 서버 인증을 위해 CA 인증서를 트러스트해야 합니다.

CA 웹사이트에 따라 이 단계를 수행할 수 없는 경우도 있습니다. CA 인증서를 트러스트하라는 메시지가 Mozilla에 자동으로 표시되지 않으면 아래 절차를 수행하여 인증서를 수동으로 트러스트합니다.

명령줄을 통한 클라이언트 인증서 관리

명령줄을 통해 인증서를 관리하려면 `certutil` 도구를 사용합니다. 이 도구는 `SUNWt1su` 패키지에 포함되어 있습니다.

1. 아래 명령을 실행하여 클라이언트 호스트 시스템에 인증서 데이터베이스를 작성합니다.

```
certutil -N -d path -P prefix
```

이 도구는 인증서를 보호하는 비밀번호를 입력하라는 메시지를 사용자에게 표시한 다음 `path/prefixcert8.db` 및 `path/prefixkey3.db` 파일을 생성합니다.

LDAP 클라이언트 응용 프로그램 사용자는 자신들만 액세스할 수 있는 위치(예: 홈 디렉토리의 보호된 하위 디렉토리)에 개별적으로 인증서 데이터베이스를 작성해야 합니다.

2. 액세스하려는 Directory Server의 인증서를 발급한 인증 기관에 연락하여 CA 인증서를 요청합니다. 전자 메일을 보내거나 해당 웹 사이트에 액세스하여 PEM 인코딩된 텍스트 버전 형식의 PKCS #11 인증서를 얻을 수 있습니다. 이 인증서를 파일에 저장합니다.

예를 들어, 내부용으로 배포된 Sun Java System Certificate Server를 사용하는 경우 `https://hostname:444` 형식의 URL을 방문합니다. 최상위 검색 탭에서 CA 인증서 체인 가져오기를 선택하고 인코딩된 인증서를 복사합니다.

또는, 동일한 CA에서 클라이언트 인증서와 서버 인증서를 얻는 경우 [434페이지의 "인증 기관 트러스트"](#) 절차를 통해 얻은 CA 인증서를 다시 사용할 수도 있습니다.

3. SSL 연결에 사용할 서버 인증서를 발급하기 위해 CA 인증서를 트러스트할 수 있는 CA로 가져옵니다. 아래 명령을 실행합니다.

```
certutil -A -n "certificateName" -t "C,," -a -i certFile -d path -P prefix
```

여기서 `certificateName`은 이 인증서를 식별하기 위해 사용자가 지정한 이름이고 `certFile`은 PEM 인코딩된 텍스트 형식의 PKCS #11 CA 인증서가 있는 텍스트 파일이며, `path`와 `prefix`는 [단계 1](#)과 같습니다.

LDAP 클라이언트 응용 프로그램의 모든 사용자는 CA 인증서를 자신의 인증서 데이터베이스로 가져와야 하며 모든 사용자가 `certFile`에 있는 동일한 인증서를 가져올 수 있습니다.

서버 인증에 대한 SSL 옵션 지정

ldapsearch 도구를 사용하여 SSL에서 서버 인증을 수행하려면 사용자는 인증서 데이터베이스 경로만 지정하면 됩니다. 서버는 보안 포트를 통해 SSL 연결을 구성할 때 자체 인증서를 보냅니다. 그런 후에 ldapsearch 도구는 사용자의 인증서 데이터베이스에서 서버 인증서를 발급한 CA의 트러스트할 수 있는 CA 인증서를 찾습니다.

아래 명령은 Mozilla에서 인증서 데이터베이스를 작성한 경우 사용자가 자신의 인증서 데이터베이스를 지정하는 방법을 보여줍니다.

```
ldapsearch -h host -p securePort \
-D "uid=bjensen,dc=example,dc=com" -w bindPassword \
-Z -P .mozilla/bjensen/string.slt/cert8.db \
-b "dc=example,dc=com" "(givenname=Richard)"
```

클라이언트에 인증서 기반의 인증 구성

클라이언트 인증의 기본 기법은 인증서를 사용하여 Directory Server에서 사용자를 안전하게 식별하는 것입니다. 인증서 기반의 클라이언트 인증을 수행하려면 다음과 같은 작업을 해야 합니다.

- 모든 디렉토리 사용자에게 대한 인증서를 얻어 클라이언트 응용 프로그램이 액세스할 수 있는 위치에 설치합니다.
- 동일한 인증서의 이진 복사본을 사용하여 사용자 디렉토리 항목을 구성합니다. 인증을 수행하는 동안 서버는 클라이언트 응용 프로그램이 제공한 인증서를 이 복사본과 비교하여 사용자를 정확하게 식별합니다.
- *Administration Server Administration Guide*에 설명되어 있는 대로 인증서 기반의 인증을 위해 서버를 구성하십시오.
- 인증서 기반의 인증에 대한 LDAP 클라이언트의 SSL 옵션을 지정합니다.

이 절차에서는 certutil 도구를 사용하여 명령줄을 통해 인증서를 관리해야 합니다. 이 도구는 SUNWt1su 패키지에 포함되어 있습니다.

사용자 인증서 얻기 및 설치

인증서 기반의 인증을 사용하여 디렉토리에 액세스하려는 각 사용자는 클라이언트 인증서를 요청하여 설치해야 합니다. 이 절차에서는 [450페이지의 "클라이언트에 서버 인증 구성"](#)에 설명된 것처럼 사용자가 이미 인증서 데이터베이스를 구성했다고 가정합니다.

1. 아래 명령을 실행하여 사용자 인증서 요청을 작성합니다.

```
certutil -R \
-s "cn=Babs Jensen,ou=Sales,o=example.com,l=city,st=state,c=country" \
-a -d path -P prefix
```

-s 옵션은 요청한 인증서의 DN을 지정합니다. 대체로 인증 기관은 인증서 소유자를 정확하게 식별하기 위해 이 예에 사용된 속성을 모두 필요로 합니다. 단계 9의 인증서 매핑 기법을 통해 인증서 DN이 사용자 디렉토리 DN에 매핑됩니다.

path 및 prefix가 사용자의 인증서와 키 데이터베이스를 찾습니다. certutil 도구는 사용자에게 키 데이터베이스 비밀번호를 입력하라는 메시지를 표시한 다음 PEM 인코딩된 텍스트 형식으로 PKCS #10 인증서 요청을 생성합니다.

2. 해당 절차에 따라 인코딩된 인증서 요청을 파일에 저장하여 인증 기관으로 보냅니다. 예를 들어, 전자 메일로 인증서 요청을 보내야 하는 경우도 있고 CA 웹사이트에서 요청을 입력할 수 있는 경우도 있습니다.
3. 요청을 보낸 후에는 CA에서 이 요청에 대한 응답으로 인증서를 보내줄 때까지 기다려야 합니다. 요청에 대한 응답 시간은 경우에 따라 달라집니다. 예를 들어, 회사 내부의 CA인 경우 하루나 이틀이면 요청에 대한 응답을 받을 수 있습니다. 회사 외부의 CA를 선택하면 요청에 대한 응답을 받을 때까지 몇 주가 걸릴 수도 있습니다.
4. CA에서 응답을 보내면 새 인증서의 PEM 인코딩된 텍스트를 다운로드하거나 텍스트 파일에 복사합니다.
5. 아래 명령을 실행하여 인증서 데이터베이스에 새 사용자 인증서를 설치합니다.

```
certutil -A -n "certificateName" -t "u,," -a -i certFile -d path -P prefix
```

여기서 *certificateName*은 이 인증서를 식별하기 위해 사용자가 지정한 이름이고 *certFile*은 PEM 형식의 PKCS #11 CA 인증서가 있는 텍스트 파일이며, *path*와 *prefix*는 단계 1과 같습니다.

또는 Mozilla를 통해 인증서 데이터베이스를 관리하는 경우 CA 웹사이트에 있는 링크를 사용하여 인증서를 직접 설치할 수도 있습니다. 이 링크를 누른 다음 Mozilla에서 제공하는 대화 상자에 따라 단계별로 수행합니다.

6. 아래 명령을 실행하여 인증서 이진 복사본을 작성합니다.

```
certutil -L -n "certificateName" -d path -r > userCert.bin
```

여기서 *certificateName*은 설치할 때 인증서에 지정한 이름이고 *path*는 인증서 데이터베이스의 위치이며, *userCert.bin*은 이진 형식의 인증서가 포함될 출력 파일의 이름입니다.

7. Directory Server에서 클라이언트 인증서를 소유한 사용자의 디렉토리 항목에 `userCertificate` 속성을 추가합니다.
 - 콘솔을 통해 인증서를 추가하려면 다음을 수행합니다.
 - a. Directory Server 콘솔의 최상위 디렉토리 탭의 디렉토리 트리에서 사용자 항목을 찾아 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 일반 편집기로 편집을 선택합니다.
 - b. 일반 편집기에서 속성 추가를 누릅니다.
 - c. 팝업 대화 상자에서 `userCertificate` 속성을 선택하고 하위 유형 드롭다운 목록에서 `binary`를 선택합니다. `binary` 하위 유형을 지정하지 않으면 인증서 매핑이 실패합니다.
 - d. 일반 편집기에서 새 `userCertificate` 필드를 찾습니다. 해당 값 설정 버튼을 눌러 이 속성에 이진 값을 설정합니다.
 - e. 값 설정 대화 상자에 **단계 6**에서 작성한 `userCert.bin` 파일의 이름을 입력하거나 찾아보기를 눌러 해당 파일을 찾습니다.
 - f. 값 설정 대화 상자에서 확인을 누른 다음 일반 편집기에서 저장을 누릅니다.
 - 명령줄에서 인증서를 추가하려면 아래 예와 같이 `ldapmodify` 명령을 사용합니다. 이 명령은 SSL을 사용하여 보안 연결을 통해 인증서를 보냅니다.

```
ldapmodify -h host -p securePort \
           -D "uid=bjensen,dc=example,dc=com" -w bindPassword \
           -Z -P .mozilla/bjensen/string.slt/cert8.db
```

```
version: 1
dn: uid=bjensen,dc=example,dc=com
changetype: modify
add: userCertificate
userCertificate;binary: < file:///path/userCert.bin
```

`binary` 하위 유형을 포함하지 않으면 인증서 매핑이 실패합니다. < 앞뒤의 공백은 중요하므로 표시된 대로 정확하게 사용해야 합니다. < 구문을 사용하여 파일 이름을 지정하려면 LDIF 명령문을 `version: 1`로 시작해야 합니다. `ldapmodify`는 이 명령문을 처리할 때 속성을 지정된 파일의 전체 내용에서 읽은 값으로 설정합니다.

8. 필요한 경우 사용자 인증서를 발급한 CA 인증서를 Directory Server에 설치하여 트러스트합니다. 클라이언트 연결을 허용하려면 이 CA를 트러스트해야 합니다. [434페이지](#)의 "인증 기관 트러스트"를 참조하십시오.

9. *Administration Server Administration Guide*에 설명되어 있는 대로 인증서 기반의 인증을 위해 Directory Server를 구성합니다. 이 절차에서는 서버가 LDAP 클라이언트를 통해 제공된 사용자 인증서를 해당 사용자 DN에 매핑할 수 있도록 `certmap.conf` 파일을 편집합니다.

`certmap.conf` 파일의 `verifyCert` 매개 변수가 `on`으로 설정되어 있는지 확인합니다. 이렇게 하면 서버가 사용자 항목에 동일한 인증서가 있는지 확인하여 사용자를 증명합니다.

인증서 기반의 클라이언트 인증에 대한 SSL 옵션 지정

`ldapsearch` 도구를 사용하여 SSL에서 인증서 기반의 클라이언트 인증을 수행하려는 경우 사용자는 자신의 인증서를 사용하는 여러 가지 명령줄 옵션을 지정해야 합니다. 보안 포트를 통해 SSL 연결을 설정하면 이 도구는 서버 인증서를 인증한 다음 사용자 인증서를 서버로 보냅니다.

아래 명령은 사용자가 Mozilla에서 작성된 자신의 인증서 데이터베이스에 액세스하는 옵션을 지정하는 방법을 보여줍니다.

```
ldapsearch -h host -p securePort \
-Z -P .mozilla/bjensen/string.slt/cert8.db \
-N "certificateName" \
-K .mozilla/bjensen/string.slt/key3.db -W keyPassword \
-b "dc=example,dc=com" "(givenname=Richard)"
```

`-Z` 옵션은 인증서 기반의 인증을 나타내고 `certificateName`은 보낼 인증서를 지정하며, `-K` 및 `-W` 옵션은 클라이언트 응용 프로그램이 인증서에 액세스하여 보낼 수 있도록 합니다. `-D` 및 `-w` 옵션을 지정하지 않으면 인증서 매핑을 통해 바인드 DN이 결정됩니다.

클라이언트에 SASL DIGEST-MD5 사용

클라이언트에 DIGEST-MD5 기법을 사용하는 경우에는 사용자 인증서를 설치할 필요가 없습니다. 하지만 암호화된 SSL 연결을 사용하려면 [450페이지의 "클라이언트에 서버 인증 구성"](#)에 설명된 것처럼 서버 인증서를 트러스트해야 합니다.

영역 지정

영역은 선택한 인증 아이디가 속하는 이름 공간을 정의합니다. DIGEST-MD5 인증에서는 특정 영역에 인증해야 합니다.

Directory Server는 시스템의 정규화된 호스트 이름을 DIGEST-MD5의 기본 영역으로 사용하며, `nsslapd-localhost` 구성 속성에 있는 호스트 이름의 소문자 값을 사용합니다.

영역을 지정하지 않으면 서버에서 제공하는 기본 영역이 사용됩니다.

환경 변수 지정

UNIX 환경에서 LDAP 도구가 DIGEST-MD5 라이브러리를 찾으려면 SASL_PATH 환경 변수를 설정해야 합니다. DIGEST-MD5 라이브러리는 SASL 플러그 인에 의해 동적으로 로드되는 공유 라이브러리이므로 Korn 셸의 경우처럼 SASL_PATH 변수를 다음과 같이 설정해야 합니다.

```
export SASL_PATH=ServerRoot/lib/sasl
```

이 경로에서는 Directory Server가 LDAP 도구를 실행할 호스트에 설치되어 있다고 가정합니다.

ldapsearch 명령 예

SSL을 사용하지 않고 DIGEST-MD5 클라이언트 인증을 수행할 수 있습니다. 아래 예에서는 기본 DIGEST-MD5 아이디 매핑을 사용하여 바인드 DN을 결정합니다.

```
ldapsearch -h host -p nonSecurePort -D "" -w bindPassword \
-o mech=DIGEST-MD5 [-o realm="hostFQDN"] \
-o authid="dn:uid=bjensen,dc=example,dc=com" \
-o authzid="dn:uid=bjensen,dc=example,dc=com" \
-b "dc=example,dc=com" "(givenname=Richard)"
```

위 예에서는 -o(소문자 o) 옵션을 사용하여 SASL 옵션을 지정합니다. Realm은 선택 사항이지만 지정할 경우 서버 호스트 시스템의 정규화된 도메인 이름을 제공해야 합니다. 프록시 작업용 authzid를 사용하지 않는 경우에도 authid와 authzid는 둘 다 있어야 하며 동일한 값을 가져야 합니다.

authid 값은 아이디 매핑에 사용되는 사용자입니다. authid는 dn: 접두어가 사용되고 뒤에 디렉토리의 유효한 사용자 DN이 오거나 u: 접두어가 사용되고 뒤에 클라이언트에서 결정되는 문자열이 오는 것이 좋습니다. 이렇게 하면 [442페이지의 "DIGEST-MD5 아이디 매핑"](#)에 설명된 매핑을 사용할 수 있습니다.

일반적으로 SSL 연결을 사용하여 보안 포트를 통해 암호화를 제공하고 DIGEST-MD5를 사용하여 클라이언트 인증을 제공합니다. 아래 예에서는 SSL을 통해 동일한 작업을 수행합니다.

```
ldapsearch -h host -p securePort \
-Z -P .mozilla/bjensen/string.slt/cert8.db \
-N "certificateName" -W keyPassword \
-o mech=DIGEST-MD5 [-o realm="hostFQDN"] \
-o authid="dn:uid=bjensen,dc=example,dc=com" \
-o authzid="dn:uid=bjensen,dc=example,dc=com" \
-b "dc=example,dc=com" "(givenname=Richard)"
```


이 예에서 `-N` 및 `-w` 옵션은 `ldapsearch` 명령에 필요하며 클라이언트 인증에는 사용되지 않습니다. 대신, 서버는 `authid` 값의 사용자에게 대해 다시 DIGEST-MD5 아이디 매핑을 수행합니다.

클라이언트에 커버로스 SASL GSSAPI 사용

클라이언트에 GSSAPI 기법을 사용하는 경우 사용자 인증서를 설치할 필요는 없지만 Kerberos V5 보안 시스템을 구성해야 합니다. 또한 암호화된 SSL 연결을 사용하려면 [450 페이지](#)의 "클라이언트에 서버 인증 구성"에 설명된 것처럼 서버 인증서를 트러스트해야 합니다.

클라이언트 호스트에 Kerberos V5 구성

LDAP 클라이언트를 실행할 호스트 시스템에 Kerberos V5를 구성해야 합니다.

1. 설치 지침에 따라 Kerberos V5를 설치합니다. Sun Enterprise Authentication Mechanism(SEAM) 1.0.1 클라이언트 소프트웨어를 설치할 것을 권장합니다.
2. 커버로스 소프트웨어를 구성합니다. SEAM을 사용하면 `/etc/krb5`에 있는 파일을 구성하여 `kdc` 서버를 설정하고, 기본 영역 및 커버로스 시스템에 필요한 기타 구성을 정의할 수 있습니다.
3. 필요한 경우 `/etc/gss/mech` 파일을 수정하여 `kerberos_v5`를 첫 번째 값으로 표시합니다.

커버로스 인증에 대한 SASL 옵션 지정

1. GSSAPI를 사용하는 클라이언트 응용 프로그램을 사용하기 전에 먼저 아래 명령을 실행하여 커버로스 보안 시스템을 사용자로 초기화해야 합니다.

```
kinit userPrincipal
```

`userPrincipal`은 사용자의 SASL 아이디입니다(예: `bjensen@EXAMPLE.COM`).

2. 커버로스를 사용하도록 SASL 옵션을 지정합니다.

UNIX 환경에서 정확한 라이브러리를 찾으려면 SASL_PATH 환경 변수를 `ServerRoot/lib/sasl`으로 설정해야 합니다. Korn 셸의 예는 다음과 같습니다.

```
export SASL_PATH=ServerRoot/lib/sasl
```

이 경로에서는 Directory Server가 LDAP 도구를 실행할 호스트에 설치되어 있다고 가정합니다.

아래의 `ldapsearch` 도구 예에서는 `-o`(소문자 `o`) 옵션을 사용하여 커버로스 사용에 대한 SASL 옵션을 지정하는 방법을 보여줍니다.

```
ldapsearch -h host -p Port \
           -o mech=GSSAPI \
           -o authid="bjensen@EXAMPLE.COM" \
           -o authzid="bjensen@EXAMPLE.COM"] \
           -b "dc=example,dc=com" "(givenname=Richard) "
```

`authid`는 `kinit` 명령으로 초기화된 커버로스 캐시에 있기 때문에 생략할 수 있습니다. `authid`가 있는 경우, 프록시 작업에 대한 `authzid`를 사용하지 않는 경우에도 `authid`와 `authzid` 값은 동일한 값을 가져야 합니다. `authid` 값은 아이디 매핑에 사용되는 사용자입니다. 사용자는 영역을 포함하는 완전한 사용자여야 합니다. 자세한 내용은 [444페이지의 "GSSAPI 아이디 매핑"](#)을 참조하십시오.

SASL이 있는 GSSAPI를 사용하여 커버로스 인증 구성: 절차 예

Directory Server에 대한 커버로스 구성은 복잡한 작업이 될 수 있습니다. 일차적으로 커버로스 설명서를 참조하십시오.

아래의 절차 예를 사용하여 어떤 단계를 따를 것인지 고려해 보면 도움이 될 것입니다. 그러나 이러한 절차는 예로 제시된 것이므로 사용자 자신의 구성과 환경에 맞게 수정해야 합니다.

Solaris에서 커버로스를 구성 및 사용하는 방법에 대한 자세한 내용은 Solaris 설명서 세트의 일부인 *Security Services System Administration Guide*에서 볼 수 있습니다. 또한 설명서 페이지를 참조할 수도 있습니다.

이 절차 예의 단계는 다음과 같습니다.

1. "가정"
2. "모든 컴퓨터 - 커버로스 클라이언트 구성 파일 편집"
3. "모든 컴퓨터 - Administration Server ACL 구성 파일 편집"

4. "KDC 컴퓨터 - KDC 서버 구성 파일 편집"
5. "KDC 컴퓨터 - KDC 데이터베이스 만들기"
6. "KDC 컴퓨터 - 관리 사용자 및 키 탭 만들기"
7. "KDC 컴퓨터 - 커버로스 데몬 시작"
8. "KDC 컴퓨터 - KDC 및 Directory Server 컴퓨터에 대한 호스트 사용자 추가"
9. "KDC 컴퓨터 - Directory Server에 대한 LDAP 사용자 추가"
10. "KDC 컴퓨터 - KDC에 테스트 사용자 추가"
11. "Directory Server 컴퓨터 - Directory Server 설치"
12. "Directory Server 컴퓨터 - GSSAPI를 활성화하도록 Directory Server 구성"
13. "Directory Server 컴퓨터 - Directory Server 키 탭 만들기"
14. "Directory Server 컴퓨터 - Directory Server에 테스트 사용자 추가"
15. "Directory Server 컴퓨터 - 테스트 사용자로서 커버로스 티켓 받기"
16. "클라이언트 컴퓨터 - GSSAPI를 통한 Directory Server 인증"

가정

이 절차 예에서는 컴퓨터 한 대가 KDC로 작동하고 보조 컴퓨터는 Directory Server를 실행하여 사용자가 GSSAPI를 통해 커버로스 인증을 수행하도록 구성하는 프로세스를 설명합니다.

KDC와 Directory Server를 같은 컴퓨터에서 실행할 수 있습니다. 이렇게 되도록 선택한 경우 같은 절차를 사용하지만 Directory Server 컴퓨터에서 이미 KDC 컴퓨터에 대해 수행된 단계는 생략하도록 합니다.

이러한 절차는 사용될 환경에 대해 여러 가지 가정을 전제로 합니다. 이는 아래에 나열되어 있습니다. 절차 예를 사용할 때는 사용자의 환경에 맞게 적절히 값을 수정하십시오.

- 이 시스템에는 최신 권장 패치 클러스터가 설치되어 있는 Solaris 9 운영 환경이 설치됩니다. 적합한 Solaris 패치가 설치되어 있지 않아 Directory Server에 대한 커버로스 인증이 실패하는 경우가 있으므로 최신 권장 패치 클러스터가 설치되는 것은 중요합니다. 또한 문서화된 절차가 대체로 Solaris 10의 절차와 같을지라도 구성 파일 형식은 약간 다르며 일부 명령의 출력은 다를 수 있습니다.

- 커버로스 데몬을 실행하는 컴퓨터는 `kdc.example.com`이라는 정규화된 도메인 이름을 갖게 됩니다. 이 컴퓨터는 이름 지정 서비스로서 DNS를 사용하도록 구성되어야 합니다. (이는 커버로스의 요구 사항으로서 `file` 같은 기타 이름 지정 서비스가 대신 사용되는 경우에는 일부 작업이 실패할 수도 있습니다.)
- Directory Server를 실행하는 컴퓨터는 `directory.example.com`이라는 정규화된 도메인 이름을 갖게 됩니다. 이 컴퓨터 역시 이름 지정 서비스로서 DNS를 사용하도록 구성되어야 합니다.
- Directory Server 컴퓨터는 커버로스를 통해 Directory Server에 인증하는 작업에 대해 클라이언트 시스템의 역할을 수행합니다. 이러한 인증은 Directory Server 및 커버로스 데몬 둘 다와 통신할 수 있는 모든 시스템에서 수행할 수 있지만 이 예에 필요한 구성 요소가 모두 Directory Server에 제공되므로 인증도 해당 시스템에서 수행됩니다.
- Directory Server의 사용자는 `uid=username, ou=People, dc=example, dc=com`이라는 형식의 DN을 가지며 해당되는 커버로스 사용자는 `username@EXAMPLE.COM`이 됩니다. 다른 이름 지정 스키마를 사용하는 경우에는 다른 GSSAPI 아이디 매핑을 사용해야 합니다.

모든 컴퓨터 - 커버로스 클라이언트 구성 파일 편집

`/etc/krb5/krb5.conf` 구성 파일은 KDC와 통신하기 위해 커버로스 클라이언트에서 필요로 하는 정보를 제공합니다.

KDC 컴퓨터, Directory Server 컴퓨터 및 커버로스를 사용하여 Directory Server에 인증할 클라이언트 컴퓨터에서 `/etc/krb5/krb5.conf` 구성 파일을 다음과 같이 편집하십시오.

- `"__default_realm__"`가 나올 때마다 `"EXAMPLE.COM"`으로 바꿉니다.
- `"__master_kdc__"`가 나올 때마다 `"kdc.example.com"`으로 바꿉니다.
- 커버로스 서버가 하나만 있게 되므로 `"__slave_kdcs__"`가 포함된 행을 제거합니다.
- `"__domain_mapping__"`를 `".example.com = EXAMPLE.COM"`(`example.com`의 처음 마침표에 주의)으로 바꿉니다.

업데이트된 `/etc/krb5/krb5.conf` 구성 파일의 내용은 코드 예 11-1과 같이 나타납니다.

코드 예 11-1 편집된 커버로스 클라이언트 구성 파일 /etc/krb5/krb5.conf

```

#pragma ident    "@(#)krb5.conf  1.2    99/07/20  SMI"
# Copyright (c) 1999, by Sun Microsystems, Inc.
# All rights reserved.
#
# krb5.conf template
# In order to complete this configuration file
# you will need to replace the __<name>__ placeholders
# with appropriate values for your network.
#

[libdefaults]
    default_realm = EXAMPLE.COM
[realms]
    EXAMPLE.COM = {
        kdc = kdc.example.com
        admin_server = kdc.example.com
    }
[domain_realm]
    .example.com = EXAMPLE.COM
[logging]
    default = FILE:/var/krb5/kdc.log
    kdc = FILE:/var/krb5/kdc.log
    kdc_rotate = {

# How often to rotate kdc.log. Logs will get rotated no more
# often than the period, and less often if the KDC is not used
# frequently.
        period = 1d

# how many versions of kdc.log to keep around (kdc.log.0, kdc.log.1, ...)
        versions = 10
    }

[appdefaults]
    kinit = {
        renewable = true
        forwardable = true
    }
    gkadmin = {
        help_url =
http://docs.sun.com:80/ab2/coll.384.1/SEAM/@AB2PageView/1195
    }

```

모든 컴퓨터 - Administration Server ACL 구성 파일 편집

/etc/krb5/kadm5.ac1 구성 파일에서 "____default_realm____"를 "EXAMPLE.COM"으로 바꿉니다. 업데이트된 /etc/krb5/kadm5.ac1 파일은 [코드 예 11-3](#)과 같이 나타납니다.

코드 예 11-2 편집된 Administration Server ACL 구성 파일

```
#
# Copyright (c) 1998-2000 by Sun Microsystems, Inc.
# All rights reserved.
#
#pragma ident  "@(#)kadm5.ac1  1.1      01/03/19 SMI"
*/admin@EXAMPLE.COM *
```

KDC 컴퓨터 - KDC 서버 구성 파일 편집

/etc/krb5/kdc.conf 파일을 편집하여 "____default_realm____"를 "EXAMPLE.COM"으로 바꿉니다.

업데이트된 /etc/krb5/kdc.conf 파일은 [코드 예 11-3](#)과 같이 나타납니다.

코드 예 11-3 편집된 KDC 서버 구성 파일 /etc/krb5/kdc.conf

```
# Copyright 1998-2002 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident  "@(#)kdc.conf  1.2      02/02/14 SMI"
```

```
[kdcdefaults]
    kdc_ports = 88,750

[realms]
    EXAMPLE.COM = {
        profile = /etc/krb5/krb5.conf
        database_name = /var/krb5/principal
        admin_keytab = /etc/krb5/kadm5.keytab
        acl_file = /etc/krb5/kadm5.ac1
        kadmind_port = 749
        max_life = 8h 0m 0s
        max_renewable_life = 7d 0h 0m 0s
        default_principal_flags = +preauth
    }
```

KDC 컴퓨터 - KDC 데이터베이스 만들기

코드 예 11-4에 사용된 명령으로 KDC 데이터베이스를 만듭니다.

코드 예 11-4 KDC 데이터베이스를 만드는 명령

```
bash-2.05# /usr/sbin/kdb5_util create -r EXAMPLE.COM -s
Initializing database '/var/krb5/principal' for realm 'EXAMPLE.COM',
master key name 'K/M@EXAMPLE.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key: master-password
Re-enter KDC database master key to verify: master-password
bash-2.05#
```

KDC 컴퓨터 - 관리 사용자 및 키 탭 만들기

다음 명령을 사용하여 관리 데몬에서 사용될 kws/admin@EXAMPLE.COM 사용자 및 서비스 키가 있는 관리 사용자를 만드십시오.

코드 예 11-5 관리 사용자 및 키 탭을 만드는 명령

```
bash-2.05# /usr/sbin/kadmin.local
kadmin.local: add_principal kws/admin
Enter password for principal "kws/admin@EXAMPLE.COM": kws-password
Re-enter password for principal "kws/admin@EXAMPLE.COM": kws-password
Principal "kws/admin@EXAMPLE.COM" created.
kadmin.local: ktadd -k /etc/krb5/kadm5.keytab kadmin/kdc.example.com
Entry for principal kadmin/kdc.example.com with kvno 3, encryption type
DES-CBC-CRC added to keytab WRFILE:/etc/krb5/kadm5.keytab.
kadmin.local: ktadd -k /etc/krb5/kadm5.keytab changepw/kdc.example.com
Entry for principal changepw/kdc.example.com with kvno 3, encryption type
DES-CBC-CRC added to keytab WRFILE:/etc/krb5/kadm5.keytab.
kadmin.local: ktadd -k /etc/krb5/kadm5.keytab kadmin/changepw
Entry for principal kadmin/changepw with kvno 3, encryption type
DES-CBC-CRC added to keytab WRFILE:/etc/krb5/kadm5.keytab.
kadmin.local: quit
bash-2.05#
```

KDC 컴퓨터 - 커버로스 데몬 시작

다음 명령을 실행하여 KDC 및 관리 데몬을 시작합니다. KDC 프로세스는 프로세스 목록에 `/usr/lib/krb5/krb5kdc`와 같이 나타나고 관리 데몬은 `/usr/lib/krb5/kadmind`와 같이 나타납니다.

코드 예 11-6 Solaris 9에서 커버로스 데몬을 시작하는 명령

```
bash-2.05# /etc/init.d/kdc start
bash-2.05# /etc/init.d/kdc.master start
bash-2.05#
```

Solaris 10의 경우 커버로스 데몬은 SMF 프레임워크에서 관리합니다. Solaris 10에서 다음과 같이 데몬을 시작하십시오.

코드 예 11-7 Solaris 10에서 커버로스 데몬을 시작하는 명령

```
bash-2.05# svcadm disable network/security/krb5kdc
bash-2.05# svcadm enable network/security/krb5kdc
bash-2.05# svcadm disable network/security/kadmin
bash-2.05# svcadm enable network/security/kadmin
bash-2.05#
```

KDC 컴퓨터 - KDC 및 Directory Server 컴퓨터에 대한 호스트 사용자 추가

다음 명령 시퀀스를 사용하여 KDC 및 Directory Server 컴퓨터에 대한 커버로스 데이터베이스에 호스트 사용자를 추가하십시오. 호스트 사용자는 `klist`와 같은 특정 커버로스 유틸리티에서 사용됩니다.

코드 예 11-8 호스트 사용자를 추가하는 명령

```
bash-2.05# /usr/sbin/kadmin -p kws/admin
Enter Password: kws-password
kadmin: add_principal -randkey host/kdc.example.com
Principal "host/kdc.example.com@EXAMPLE.COM" created.
kadmin: ktadd host/kdc.example.com
Entry for principal host/kdc.example.com with kvno 3, encryption type
DES-CBC-CRC added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin: add_principal -randkey host/directory.example.com
Principal "host/directory.example.com@EXAMPLE.COM" created.
kadmin: ktadd host/directory.example.com
```



```
Entry for principal host/directory.example.com with kvno 3, encryption type
DES-CBC-CRC added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin: quit
bash-2.05#
```

KDC 컴퓨터 - Directory Server에 대한LDAP 사용자 추가

Directory Server에서 인증 사용자가 보유한 커버로스 티켓을 확인할 수 있도록 하기 위해서는 해당 Directory Server에 자체 사용자가 있어야 합니다. 현재, Directory Server는 하드 코딩되어 있으므로 `ldap/fqdn@realm`의 사용자가 필요합니다. 여기서 `fqdn`은 Directory Server의 정규화된 도메인 이름(Directory Server를 설치할 때 제공된 정규화된 이름과 일치해야 함)이며 `realm`은 커버로스 영역입니다. 이 경우, Directory Server의 사용자는 `ldap/directory.example.com@EXAMPLE.COM`이 됩니다.

다음 명령 시퀀스를 사용하여 Directory Server에 대한 LDAP 사용자를 만드십시오.

코드 예 11-9 KDC에 LDAP 사용자를 추가하는 명령

```
bash-2.05# /usr/sbin/kadmin -p kws/admin
Enter Password: kws-password
kadmin: add_principal -randkey ldap/directory.example.com
Principal "ldap/directory.example.com@EXAMPLE.COM" created.
kadmin: quit
bash-2.05#
```

KDC 컴퓨터 - KDC에 테스트 사용자 추가

커버로스 인증을 수행할 수 있도록 하기 위해서는 사용자 인증이 커버로스 데이터베이스에 있어야 합니다. 이 예에서 사용자는 `kerberos-test`라는 사용자 이름을 갖게 되며 이는 커버로스 사용자가 `kerberos-test@EXAMPLE.COM`이 된다는 의미입니다.

[코드 예 11-10](#)에 제공된 명령 시퀀스를 사용하여 사용자를 만드십시오.

코드 예 11-10 KDC에 테스트 사용자를 추가하는 명령

```
bash-2.05# /usr/sbin/kadmin -p kws/admin
Enter Password: kws-password
kadmin: add_principal kerberos-test
Enter password for principal "kerberos-test@EXAMPLE.COM": test-password
Re-enter password for principal "kerberos-test@EXAMPLE.COM": test-password
Principal "kerberos-test@EXAMPLE.COM" created.
kadmin: quit
bash-2.05#
```

Directory Server 컴퓨터 - Directory Server 설치

Directory Server 5.2 및 패치를 설치하십시오. 설정 예는 아래와 같습니다.

정규화된 컴퓨터 이름	directory.example.com
설치 루트	/data/ds52p2
서버 사용자	unixuser
서버 그룹	unixgroup
서버 식별자	directory
서버 포트	389
접미어	dc=example,dc=com
관리자 아이디	admin
관리자 비밀번호	dsadmin-password
관리 도메인	example.com
디렉토리 관리자 DN	cn=Directory Manager
디렉토리 관리자 비밀번호	dm-password
관리 포트	390

Directory Server 컴퓨터 - GSSAPI를 활성화하도록 Directory Server 구성

먼저, Directory Server에서 사용자를 기반으로 인증하는 커버로스 사용자를 식별하는 데 사용할 매핑을 정의하는 /data/ds52p2/shared/bin/gssapi.ldif 파일을 만드십시오. 코드 예 11-11에 나타난 것과 같이 파일 내용을 만듭니다.

코드 예 11-11 gssapi.ldif 파일 내용

```
n: cn=GSSAPI,cn=identity mapping,cn=config
changetype: add
objectClass: top
objectClass: nsContainer
cn: GSSAPI
dn: cn=default,cn=GSSAPI,cn=identity mapping,cn=config
changetype: add
```

```
objectClass: top
objectClass: nsContainer
objectClass: dsIdentityMapping
objectClass: dsPatternMatching
cn: default
dsMatching-pattern: ${Principal}
dsMatching-regexp: (.*)@EXAMPLE.COM
dsMappedDN: uid=$1,ou=People,dc=example,dc=com
```

```
dn: cn=SASL,cn=security,cn=config
changetype: modify
replace: dsSaslPluginsPath
dsSaslPluginsPath: /data/ds52p2/lib/sasl
```

다음으로, `ldapmodify` 명령을 사용하여 코드 예 11-12에 나타난 것처럼 적합한 매핑을 사용하여 GSSAPI를 활성화하도록 Directory Server를 업데이트하십시오.

코드 예 11-12 GSSAPI를 활성화하도록 Directory Server를 업데이트하는 명령

```
bash-2.05# ./ldapmodify -D 'cn=Directory Manager' -w dm-password -a -f
gssapi.ldif
adding new entry cn=GSSAPI,cn=identity mapping,cn=config
adding new entry cn=default,cn=GSSAPI,cn=identity mapping,cn=config
modifying entry cn=SASL,cn=security,cn=config
bash-2.05#
```

Directory Server 컴퓨터 - Directory Server 키 탭 만들기

앞서 언급한 바와 같이 Directory Server는 GSSAPI를 통해 커버로스 사용자를 인증하기 위해 KDC에 자체 사용자를 두어야 합니다. 이 작업이 올바르게 작동하도록 하려면 이 사용자 정보가 Directory Server 컴퓨터의 커버로스 키 탭에 Directory Server가 작동하는 사용자 계정에서 읽을 수 있는 파일 형식으로 있어야 합니다.

다음 명령 시퀀스를 사용하여 정확한 속성을 갖도록 키 탭을 만드십시오.

코드 예 11-13 Directory Server에 대한 키 탭을 만드는 명령

```
bash-2.05# /usr/sbin/kadmin -p kws/admin
Enter Password: kws-password
kadmin: ktadd -k /data/ds52p2/slapd-directory/config/ldap.keytab
ldap/directory.example.com
Entry for principal ldap/directory.example.com with kvno 3, encryption type
DES-CBC-CRC added to keytab
WRFIL: /data/ds52p2/slapd-directory/config/ldap.keytab.
kadmin: quit
bash-2.05#
```

Directory Server를 실행하는 데 사용되는 사용자 계정에서 소유하고 해당 사용자만 읽을 수 있도록 이 사용자 정의 키 탭의 사용 권한과 소유권을 변경하십시오.

코드 예 11-14 키 탭 사용 권한 및 소유권을 바꾸는 명령

```
bash-2.05# chown unixuser:unixgroup
/data/ds52p2/slapd-directory/config/ldap.keytab
bash-2.05# chmod 600 /data/ds52p2/slapd-directory/config/ldap.keytab
bash-2.05#
```

기본적으로 Directory Server는 /etc/kerberos5/krb5.keytab 파일의 표준 커버로스 키 탭을 사용하려고 합니다. 그러나 이 파일을 Directory Server 사용자가 읽을 수 있도록 만들면 보안상 위험이 따르므로 Directory Server에 사용자 정의 키 탭을 만드는 것입니다.

새 사용자 정의 키 탭을 사용하도록 Directory Server를 구성하십시오. 이는 **코드 예 11-15**에서 보는 바와 같이 start-slapd 스크립트를 편집하여 KRB5_KTNAME 환경 변수를 사용하는 방법으로 수행할 수 있습니다.

코드 예 11-15 사용자 정의 커버로스 키 탭을 사용하도록 업데이트된 start-slapd 스크립트

```
#!/bin/sh

# Configure the server to use a custom Kerberos keytab.
KRB5_KTNAME=/data/ds52p2/slapd-directory/config/ldap.keytab
export KRB5_KTNAME

unset LD_LIBRARY_PATH

# Script that starts the ns-slapd server.
# Exit status can be:
#      0: Server started successfully
#      1: Server could not be started
#      2: Server was already started
```

```
NETSITE_ROOT=/data/ds52p2
export NETSITE_ROOT
```

{The remainder of this file has been omitted to conserve space}

마지막으로, 이러한 변경 사항이 적용되도록 Directory Server를 다시 시작하십시오.

코드 예 11-16 Directory Server를 다시 시작하는 명령

```
bash-2.05# cd /data/ds52p2/slaped-directory
bash-2.05# ./stop-slaped
bash-2.05# ./start-slaped
bash-2.05#
```

Directory Server 컴퓨터 - Directory Server에 테스트 사용자 추가

Directory Server에 커버로스 사용자를 인증하기 위해서는 해당 사용자의 커버로스 사용자에게 해당하는 사용자에 대한 디렉토리 항목이 있어야 합니다.

이전 단계에서 `kerberos-test@EXAMPLE.COM` 사용자가 있는 커버로스 데이터베이스에 테스트 사용자가 추가되었습니다. 아이디 매핑 구성이 디렉토리에 추가되었기 때문에 그 사용자에게 해당하는 디렉토리 항목에는 `uid=kerberos-test,ou=People,dc=example,dc=com`이라는 DN이 있어야 합니다.

먼저 다음 내용이 있는 `/data/ds52p2/shared/bin/testuser.ldif` 파일을 작성하여 디렉토리에 사용자를 추가하십시오.

코드 예 11-17 새 파일 `testuser.ldif`

```
dn: uid=kerberos-test,ou=People,dc=example,dc=com
changetype: add
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: kerberos-test
givenName: Kerberos
sn: Test
cn: Kerberos Test
description: An account for testing Kerberos authentication through GSSAPI
```

다음으로, `ldapmodify`를 사용하여 서버에 이 항목을 추가하십시오.

코드 예 11-18 서버에 테스트 항목을 추가하는 명령

```
bash-2.05# ./ldapmodify -D 'cn=Directory Manager' -w dm-password -f
testuser.ldif
adding new entry uid=kerberos-test,ou=People,dc=example,dc=com
bash-2.05#
```

Directory Server 컴퓨터 - 테스트 사용자로서 커버로스 티켓 받기

커버로스 데이터베이스와 Directory Server에 테스트 사용자가 존재하므로 KDC는 GSSAPI를 사용하여 커버로스에 대해 그 사용자로서 Directory Server에 인증할 수 있습니다.

먼저, **코드 예 11-19**에 나와 있는 대로 kinit 명령을 사용하여 이 사용자에 대한 커버로스 티켓을 받으십시오.

코드 예 11-19 커버로스 티켓을 받는 명령

```
bash-2.05# kinit kerberos-test
Password for kerberos-test@EXAMPLE.COM: test-password
bash-2.05#
```

그런 다음 klist 명령을 사용하여 이 티켓에 관한 정보를 확인하십시오.

코드 예 11-20 테스트 티켓에 대한 정보를 보는 명령

```
bash-2.05# klist
Ticket cache: /tmp/krb5cc_0
Default principal: kerberos-test@EXAMPLE.COM

Valid starting          Expires                Service principal
Sat Jul 24 00:24:15 2004 Sat Jul 24 08:24:15 2004  krbtgt/EXAMPLE.COM@EXAMPLE.COM
    renew until Sat Jul 31 00:24:15 2004
bash-2.05#
```

클라이언트 컴퓨터 - GSSAPI를 통한 Directory Server 인증

마지막 단계로서, GSSAPI를 사용하여 Directory Server를 인증하십시오. Directory Server에 제공된 ldapsearch 유틸리티는 GSSAPI, DIGEST-MD5 및 EXTERNAL 기법을 포함하여 SASL 인증에 대한 지원을 제공합니다. 그러나 이를 위해서는 클라이언트가 사용할 SASL 라이브러리에 대한 경로가 제공되어야 합니다. SASL_PATH 환경 변수를 Directory Server 설치 루트 아래 lib/sasl 디렉토리로 설정함으로써 이 작업을 수행할 수 있습니다.

코드 예 11-21 SASL_PATH 환경 변수를 설정하는 명령

```
bash-2.05# SASL_PATH=/data/ds52p2/lib/sasl
bash-2.05# export SASL_PATH
bash-2.05#
```

ldapsearch를 사용하여 Directory Server에 대한 커버로스 기반의 인증을 수행하려면 -o mech=GSSAPI 및 -o authzid=*principal* 인수를 포함해야 합니다. 다음 예는 이전에 만든 커버로스 테스트 사용자 계정으로 인증되는 동안 dc=example, dc=com 항목을 검색합니다.

코드 예 11-22 GSSAPI를 통한 인증 후 dc=example, dc=com 항목 검색

```
bash-2.05# ./ldapsearch -h directory.example.com -p 389 -o mech=GSSAPI -o
authzid="kerberos-test@EXAMPLE.COM" -b "dc=example,dc=com" -s base
i(objectClass=*)i
version: 1
dn: dc=example,dc=com
dc: example
objectClass: top
objectClass: domain
bash-2.05#
```

Directory Server 액세스 로그를 검사하여 인증이 예상한 대로 처리되었는지 확인하십시오.

코드 예 11-23

```
bash-2.05# tail -12 /data/ds52p2/slaped-directory/logs/access
[24/Jul/2004:00:30:47 -0500] conn=0 op=-1 msgId=-1 - fd=23 slot=23 LDAP connection from
1.1.1.8 to 1.1.1.8
[24/Jul/2004:00:30:47 -0500] conn=0 op=0 msgId=1 - BIND dn="" method=sasl version=3
mech=GSSAPI
[24/Jul/2004:00:30:47 -0500] conn=0 op=0 msgId=1 - RESULT err=14 tag=97 nentries=0
etime=0, SASL bind in progress
[24/Jul/2004:00:30:47 -0500] conn=0 op=1 msgId=2 - BIND dn="" method=sasl version=3
mech=GSSAPI
[24/Jul/2004:00:30:47 -0500] conn=0 op=1 msgId=2 - RESULT err=14 tag=97 nentries=0
etime=0, SASL bind in progress
[24/Jul/2004:00:30:47 -0500] conn=0 op=2 msgId=3 - BIND dn="" method=sasl version=3
mech=GSSAPI
[24/Jul/2004:00:30:47 -0500] conn=0 op=2 msgId=3 - RESULT err=0 tag=97 nentries=0 etime=0
dn="uid=kerberos-test,ou=people,dc=example,dc=com"
[24/Jul/2004:00:30:47 -0500] conn=0 op=3 msgId=4 - SRCH base="dc=example,dc=com" scope=0
filter="(objectClass=*)" attrs=ALL
```

```
[24/Jul/2004:00:30:47 -0500] conn=0 op=3 msgId=4 - RESULT err=0 tag=101 nentries=1 etime=0
[24/Jul/2004:00:30:47 -0500] conn=0 op=4 msgId=5 - UNBIND
[24/Jul/2004:00:30:47 -0500] conn=0 op=4 msgId=-1 - closing - U1
[24/Jul/2004:00:30:48 -0500] conn=0 op=-1 msgId=-1 - closed.
bash-2.05#
```

이로부터 바인드는 3단계 프로세스로서, 처음 두 단계는 LDAP 결과 14(처리 중인 SASL 바인드)를 반환하고 세 번째 단계는 해당 바인드가 성공적으로 처리되었음을 표시한다는 사실을 확인할 수 있습니다. `method=sasl` 및 `mech=GSSAPI` 태그는 이 바인드가 GSSAPI SASL 기법을 사용했음을 나타내고, 성공적인 바인드 응답의 끝 부분에 있는 `dn="uid=kerberos-test,ou=people,dc=example,dc=com"`은 적합한 사용자로서 바인드가 수행되었음을 나타냅니다.

Pass Through Authentication 구현

PTA(Pass-Through Authentication)는 한 디렉토리 서버에서 다른 디렉토리 서버를 참조하여 바인드 요청을 인증하는 기법입니다. 디렉토리 서버는 PTA 플러그인의 기능을 사용하여 로컬 접미어에 저장되지 않은 항목에 대한 단순한 바인드 작업(비밀번호 기반)을 허용할 수 있습니다.

Directory Server에서는 PTA를 사용하여 Directory Server의 여러 인스턴스에 있는 사용자 및 구성 디렉토리를 관리할 수 있습니다.

주 한 개의 서버를 사용자 디렉토리 및 구성 디렉토리에 모두 사용하는 경우에는 Directory Server 콘솔에 PTA 플러그인이 표시되지 않지만 새로 작성하여 PTA를 사용할 수 있습니다.

이 장의 다음 절에서는 PTA 플러그인에 대해 설명합니다.

- [Directory Server의 PTA 사용 방법](#)
- [PTA 플러그인 구성](#)

Directory Server의 PTA 사용 방법

Directory Server의 각 인스턴스에 구성 디렉토리와 사용자 디렉토리를 따로 설치하면 설치 프로그램이 자동으로 PTA를 설정하여 구성 관리 권한이 있는 사용자(일반적으로 admin)가 관리 업무를 수행할 수 있도록 허용합니다.

이 경우 admin 사용자 항목이 구성 디렉토리의 o=NetscapeRoot에 저장되기 때문에 PTA가 필요하며 admin으로 사용자 디렉토리에 바인드를 시도하면 일반적으로 실패하게 됩니다. PTA를 통해 사용자 디렉토리는 자격 증명을 구성 디렉토리로 전송하여 확인 받을 수 있으며, 이 확인 결과에 따라 admin 사용자의 바인드를 허용합니다.

이 예에서 사용자 디렉토리는 PTA 서버, 즉 바인드 요청을 다른 디렉토리 서버로 전달하는 서버 역할을 합니다. 구성 디렉토리는 *인증 서버*, 즉 항목이 포함되어 있으며 요청하는 클라이언트의 바인드 자격 증명을 확인하는 서버 역할을 합니다.

이 장에서는 *PTA 하위 트리*란 용어도 사용합니다. PTA 하위 트리는 PTA 서버에 존재하지 않는 하위 트리입니다. 사용자의 바인드 DN에 이 하위 트리가 포함되어 있으면 사용자 자격 증명은 인증 디렉토리로 전달됩니다.

아래의 단계 시퀀스는 PTA가 어떻게 작동하는지 보여줍니다.

1. PTA 하위 트리 o=NetscapeRoot가 포함된 구성 디렉토리 서버(인증 디렉토리)를 configdir.example.com 호스트에 설치합니다.
2. dc=example,dc=com 접미어의 데이터가 포함된 사용자 디렉토리 서버(PTA 디렉토리)를 userdir.example.com 호스트에 설치합니다.
3. 사용자 디렉토리 설치 중에 구성 디렉토리를 가리키는 LDAP URL을 지정하라는 메시지가 표시됩니다. 예를 들면 다음과 같습니다.
 ldap://configdir.example.com/o=NetscapeRoot
4. 설치 프로그램이 지정된 LDAP URL을 사용하여 사용자 디렉토리에 PTA 플러그인을 구성 및 활성화합니다.

이제 사용자 디렉토리가 PTA 디렉토리로 구성되어 o=NetscapeRoot가 포함된 DN을 가진 항목에 대한 모든 바인드 요청을 구성 디렉토리 configdir.example.com으로 보냅니다.

5. 설치가 완료되면 admin 사용자는 사용자 디렉토리에 바인드를 시도하여 사용자 데이터를 작성하기 시작합니다.
 admin 항목은 구성 디렉토리에 uid=admin,
 ou=Administrators,ou=TopologyManagement,o=NetscapeRoot로 저장되므로, PTA 플러그인 구성에 정의된 것처럼 사용자 디렉토리는 이 바인드 요청을 구성 디렉토리로 전달합니다.
6. 구성 디렉토리는 비밀번호를 포함한 바인드 자격 증명을 인증하여 다시 사용자 디렉토리로 확인을 보냅니다.
7. 이 확인 결과에 따라 admin 사용자의 바인드를 허용합니다.

PTA 플러그인 구성

PTA 플러그인 구성 정보는 PTA 서버의 `cn=Pass Through Authentication,cn=plugins,cn=config` 항목에 지정됩니다.

사용자 디렉토리 및 구성 디렉토리를 별개의 서버 인스턴스에 설치하면 사용자 디렉토리의 구성에 PTA 플러그인 항목이 자동으로 추가됩니다. 두 디렉토리를 하나의 인스턴스에 설치한 경우 다른 디렉토리 및 PTA를 수행하려면 먼저 플러그인 구성 항목을 작성해야 합니다.

플러그인 구성 항목 작성

1. 아래 명령을 실행하여 플러그인 구성 항목을 작성합니다.

```
ldapmodify -a -h PTAhost -p port -D "cn=Directory Manager" -w password
dn: cn=Pass Through Authentication,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: Pass Through Authentication
nsslapd-pluginPath: ServerRoot/lib/passthru-plugin.soextension
nsslapd-pluginInitfunc: passthruauth_init
nsslapd-pluginType: preoperation
nsslapd-plugin-depends-on-type: database
nsslapd-pluginId: passthruauth
nsslapd-pluginVersion: 5.2
nsslapd-pluginVendor: Sun Microsystems, Inc.
nsslapd-pluginDescription: pass through authentication plugin
nsslapd-pluginEnabled: on or off
nsslapd-pluginarg0: ldap[s]://authenticatingHost[:port]/PTAsubtree options
```

여기서 *ServerRoot*는 설치에 따라 결정되며 확장자는 HP-UX의 경우 `.sl`, 기타 모든 UNIX 플랫폼의 경우 `.so`, Windows의 경우 `.dll`이 됩니다.

플러그인 인수는 인증 디렉토리 서버의 호스트 이름을 식별하는 LDAP URL, 선택 사항 포트 및 PTA 하위 트리를 지정합니다. 포트를 지정하지 않으면 389(LDAP)와 636(LDAPS)이 기본 포트가 됩니다. 다음 절에 설명된 연결 매개 변수를 선택 사항으로 설정할 수도 있습니다. *PTAsubtree*가 *PTAhost*에 있으면 플러그인이 바인드 요청을 *authenticatingHost*로 전달하지 않으므로 바인드는 PTA 없이 로컬로 처리됩니다.

2. 35페이지의 "Directory Server 시작 및 중지"에 설명된 것처럼 서버를 다시 시작합니다.

보안 연결을 사용하도록 PTA 구성

PTA 플러그 인은 비밀번호를 포함한 바인드 자격 증명을 인증 디렉토리로 보내야 하므로 보안 연결을 사용하는 것이 좋습니다. PTA 디렉토리가 SSL을 통해 인증 디렉토리와 통신하도록 구성하려면 다음을 수행합니다.

- 11장, "인증 및 암호화 관리"에 설명된 것처럼 PTA 디렉토리와 인증 디렉토리에서 모두 SSL을 구성 및 활성화합니다.
- LDAP URL에 LDAPS 및 보안 포트를 사용하도록 PTA 플러그 인 구성을 새로 작성하거나 수정합니다. 예를 들면 다음과 같습니다.

```
ldaps://configdir.example.com:636/o=NetscapeRoot
```

연결 매개 변수(선택 사항) 설정

PTA 플러그 인 인수는 LDAP URL 뒤에 오는 연결 매개 변수 집합을 선택 사항으로 허용합니다. 예를 들면 다음과 같습니다.

```
ldap[s]://host[:port]/subtree [maxconns,maxops,timeout,ldapver,connlife]
```

매개 변수는 표시된 순서대로 지정해야 합니다. 이러한 매개 변수는 선택 사항이지만 한 개만 개별적으로 지정할 수는 없습니다. 한 개의 매개 변수를 지정하려면 모두 지정해야 합니다. 일부 매개 변수만 사용자 정의하려면 아래에 제공된 기본값을 지정하십시오.

subtree 매개 변수와 선택 사항 매개 변수 사이에는 공백이 있어야 합니다.

각각의 LDAP URL에 대해 다음과 같은 선택 사항 매개 변수를 구성할 수 있습니다.

- *maxconns* - PTA 서버에서 인증 서버에 대해 동시에 열 수 있는 최대 연결 수. 이 매개 변수는 인증 서버로 전달할 수 있는 동시 바인드 수를 제한합니다. 기본값은 3개입니다.
- *maxops* - PTA 디렉토리 서버에서 단일 연결 내에 인증 디렉토리 서버로 동시에 보낼 수 있는 최대 바인드 요청 수. 이 매개 변수는 동시 PTA 수를 추가로 제한합니다. 기본값은 5개입니다.
- *timeout* - PTA 서버에서 인증 서버의 응답을 기다리는 최대 지연 시간(초). 기본값은 300초(5분)입니다.
- *ldapver* - PTA 서버에서 인증 서버에 연결할 때 사용할 LDAP 프로토콜 버전. 허용되는 값은 LDAPv2의 경우 2, LDAPv3의 경우 3입니다. 기본값은 3개입니다.

- *connlife* - PTA 서버에서 인증 서버에 대한 연결을 다시 사용할 시간 제한(초). 이 시간이 만료된 후 클라이언트에서 PTA 하위 트리의 바인드를 요청하면 서버는 PTA 연결을 닫고 새 연결을 엽니다. 바인드 요청이 시작되어 서버에서 시간 제한이 초과되었다는 것을 확인한 경우에만 연결이 닫힙니다. 이 옵션을 지정하지 않거나 LDAP URL에 한 개의 인증 서버만 열거되어 있으면 시간 제한은 실행되지 않습니다. 두 개 이상의 호스트가 열거되어 있으면 기본값은 300초(5분)입니다.

PTA 플러그 인 인수에 대한 아래 예에서는 연결 수를 10개로 늘리고 시간 제한을 1분(60초)으로 줄입니다. 다른 모든 매개 변수에 대해서는 기본값이 지정됩니다.

```
ldaps://configdir.example.com:636/o=NetscapeRoot 10,5,60,3,300
```

여러 개의 서버 및 하위 트리 지정

PTA 플러그 인에 여러 개의 인수를 구성하여 여러 인증 서버, 여러 PTA 하위 트리 또는 둘 모두를 지정할 수 있습니다. 각 인수에는 한 개의 LDAP URL이 포함되며 자체 연결 옵션 집합을 가질 수 있습니다.

한 개의 PTA 하위 트리에 여러 개의 인증 서버가 있는 경우 이러한 인증 서버는 페일오버 서버 역할을 합니다. PTA 연결이 시간 제한에 도달하면 플러그 인은 열거된 순서대로 인증 서버에 연결하며, 모든 연결이 시간 초과되면 인증이 실패합니다.

여러 개의 PTA 하위 트리가 정의되어 있으면 플러그 인은 바인드 DN에 따라 해당 서버로 인증 요청을 전달합니다. 아래 예에서는 두 개의 PTA 하위 트리를 정의하는 PTA 플러그 인 인수 네 개를 보여줍니다. 각 하위 트리에는 인증용 페일오버 서버와 서버별 연결 매개 변수가 있습니다.

```
nsslapd-pluginarg0: ldaps://configdir.example.com/o=NetscapeRoot
10,10,60,3,300
nsslapd-pluginarg1: ldaps://configbak.example.com/o=NetscapeRoot
3,5,300,3,300
nsslapd-pluginarg2: ldaps://east.example.com/ou=East,ou=People,
dc=example,dc=com 10,10,300,3,300
nsslapd-pluginarg3: ldaps://eastbak.example.com/ou=East,ou=People,
dc=example,dc=com 3.5,300,3,300
```

다음 예처럼 호스트 이름을 공백으로 구분하여 여러 서버를 지정할 수도 있습니다.

```
nsslapd-pluginarg0: ldaps://configdir.example.com:636
  configbak.example.com:636/o=NetscapeRoot 10,10,60,3,300
```

PTA 플러그 인 구성 수정

언제든지 PTA 플러그 인을 다시 구성하여 활성화 또는 비활성화하거나 인증 호스트나 PTA 하위 트리를 변경할 수 있습니다.

1. PTA 플러그 인 구성 항목(`cn=Pass Through Authentication,cn=plugins,cn=config`)을 편집하여 `nsslapd-pluginenabled` 속성과 `nsslapd-pluginargN` 속성을 수정합니다. `콘솔`이나 `ldapmodify` 유틸리티 중 하나를 사용하여 구성을 편집할 수 있습니다.

예를 들어, 아래 명령은 위에 열거된 연결 매개 변수와 SSL을 사용하여 PTA 플러그 인을 활성화합니다.

```
dn: cn=Pass Through Authentication,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginenabled:
nsslapd-pluginenabled: on
-
replace: nsslapd-pluginarg0:
nsslapd-pluginarg0: ldaps://configdir.example.com:636/
  o=NetscapeRoot 10.100.60,3,300
-
replace: nsslapd-pluginarg1
nsslapd-pluginarg1: ldaps://configbak.example.com:636/
  o=NetscapeRoot 3,5,300,3,300
^D
```

2. 35페이지의 "[Directory Server 시작 및 중지](#)"에 설명된 것처럼 서버를 다시 시작합니다.

로그 파일을 사용하여 Directory Server 모니터

이 장에서는 로깅 정책을 구성하고 서버에서 유지 관리하는 상태 정보를 분석하여 Directory Server를 모니터링하는 방법에 대해 설명합니다.

Directory Server는 다음 세 가지 유형의 로그를 제공합니다.

- 액세스 로그 - 서버에 연결하는 클라이언트와 요청한 작업을 열거합니다.
- 오류 로그 - 서버 오류에 대한 정보를 제공합니다.
- 감사 로그 - 접미어 및 구성의 수정 사항에 대한 세부 정보를 제공합니다.

서버의 상태 정보에는 연결 및 캐시 작업에 대한 통계도 포함되어 있습니다. 이 정보는 Directory Server 콘솔 및 항목 모니터 시 LDAP 명령줄 도구에서 사용할 수 있습니다. SNMP를 사용하여 서버를 모니터링하는 방법은 [14장](#), "[SNMP를 사용하여 Directory Server 모니터](#)"를 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

- [로그 파일 정책 정의](#)
- [액세스 로그](#)
- [오류 로그](#)
- [감사 로그](#)
- [서버 작업 모니터](#)

로그 파일 정책 정의

다음 절에서는 로그 파일 작성 및 삭제 정책을 정의하는 방법에 대해 설명합니다.

로그 파일 순환 정책 정의

디렉토리에서 현재 로그를 정기적으로 아카이브하고 새 로그를 시작하도록 설정하려면 Directory Server 콘솔에서 로그 파일 순환 정책을 정의할 수 있습니다. 다음과 같은 매개 변수를 구성할 수 있습니다.

- 디렉토리에서 유지 관리할 총 로그 수. 이 로그 수에 도달하면 디렉토리는 새 로그를 작성하기 전에 폴더에서 가장 오래된 로그 파일을 삭제합니다. 기본값은 10개입니다. 이 값을 1로 설정하지 마십시오. 이렇게 하면 디렉토리에서 로그를 순환하지 않으므로 로그가 무제한 증가하게 됩니다.
- 각 로그 파일의 최대 크기(MB). 최대 크기를 설정하지 않으려면 이 필드에 -1을 입력합니다. 기본값은 100MB입니다. 로그 파일이 이 최대 크기(또는 다음 단계에서 정의한 최대 수명)에 도달하면 디렉토리는 해당 파일을 아카이브하고 새 파일을 시작합니다. 최대 로그 수를 1로 설정하면 디렉토리는 이 속성을 무시합니다.
- 디렉토리에서 현재 로그 파일을 아카이브하고 새 로그 파일을 작성하는 빈도. 분, 시간, 일, 주 또는 월을 입력합니다. 기본값은 매일입니다. 최대 로그 수를 1로 설정하면 디렉토리는 이 속성을 무시합니다.

로그 파일 삭제 정책 정의

디렉토리에서 아카이브된 이전 로그를 자동 삭제하도록 설정하려면 Directory Server 콘솔에서 로그 파일 삭제 정책을 정의할 수 있습니다. 로그 삭제 정책은 이전에 로그 파일 순환 정책을 정의한 경우에만 적용됩니다. 로그 파일이 하나뿐이면 로그 파일 삭제는 작동하지 않습니다.

서버는 로그 순환 시 로그 파일 삭제 정책을 평가하여 적용합니다.

다음과 같은 매개 변수를 구성할 수 있습니다.

- 아카이브된 로그 전체의 최대 크기. 최대 크기에 도달하면 아카이브된 가장 오래된 로그부터 자동으로 삭제됩니다. 최대 크기를 설정하지 않으려면 이 필드에 -1을 입력합니다. 기본값은 500MB입니다. 로그 파일 수를 1로 설정하면 이 매개 변수는 무시됩니다.

- 사용 가능한 최소 디스크 공간. 사용 가능한 디스크 공간이 최소값에 도달하면 아카이브된 가장 오래된 로그부터 자동으로 삭제됩니다. 기본값은 5MB입니다. 로그 파일 수를 1로 설정하면 이 매개 변수는 무시됩니다.
- 로그 파일의 최대 수명. 최대 수명에 도달한 로그 파일은 자동으로 삭제됩니다. 기본값은 1개월입니다. 로그 파일 수를 1로 설정하면 이 매개 변수는 무시됩니다.

수동 로그 파일 순환

자동 로그 파일 작성 또는 삭제 정책을 설정하지 않은 경우 수동으로 로그 파일을 순환할 수 있습니다. 기본적으로 액세스, 오류 및 감사 로그 파일은 아래 디렉토리에 위치해 있습니다.

`ServerRoot/slapd-serverID/logs`

로그 파일을 수동으로 순환하려면 다음을 수행합니다.

1. 서버를 종료합니다. 자세한 내용은 [35페이지의 "Directory Server 시작 및 중지"](#)를 참조하십시오.
2. 이전 로그 파일을 나중에 참조하려면 순환하는 로그 파일을 이동하거나 이름을 바꿉니다.
3. 서버를 다시 시작합니다. 자세한 내용은 [35페이지의 "Directory Server 시작 및 중지"](#)를 참조하십시오.

서버는 각 로그 구성에 따라 자동으로 새 파일을 작성합니다.

액세스 로그

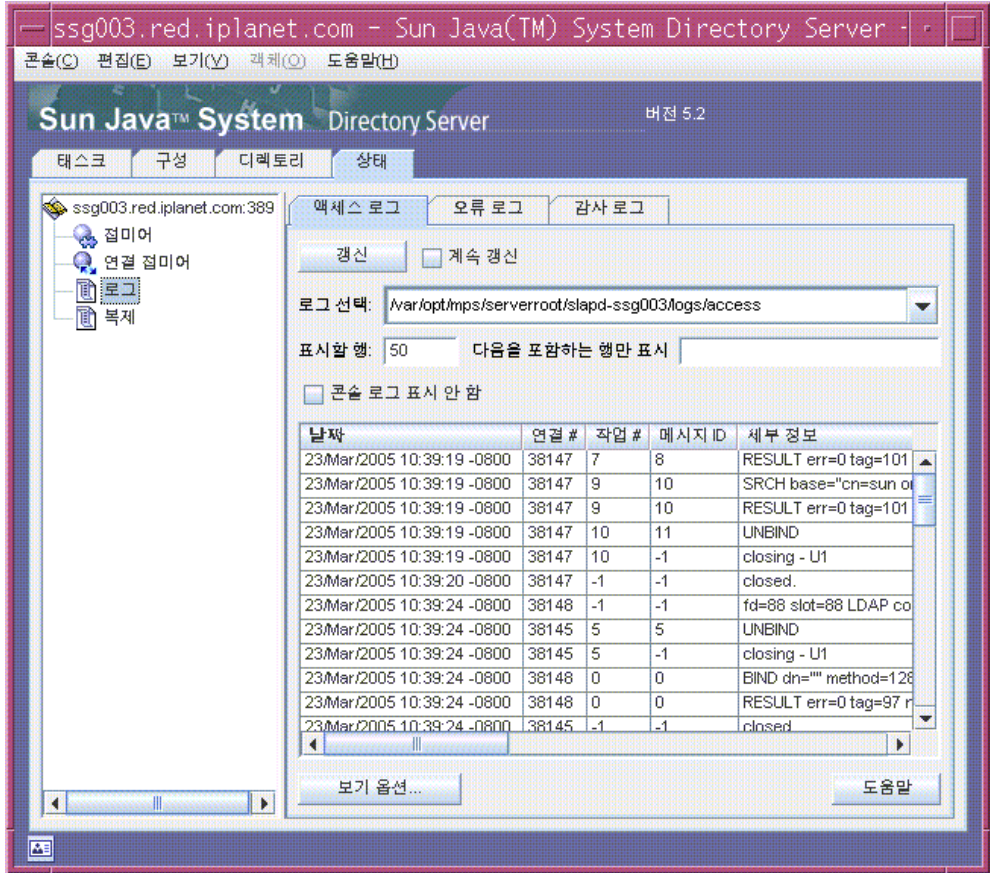
액세스 로그에는 디렉토리에 대한 클라이언트 연결 정보가 포함되어 있습니다. Directory Server Resource Kit에서는 Directory Server 액세스 로그를 분석할 수 있게 해주는 로그 분석기 도구인 `logconv.pl`을 제공합니다. 로그 분석기 도구는 사용 통계를 추출하여 중요한 이벤트의 발생 수를 계산합니다. 이 도구에 대한 자세한 내용은 *Directory Server Resource Kit Tools Reference*를 참조하십시오.

액세스 로그 보기

1. Directory Server 콘솔의 최상위 상태 탭에서 로그 아이콘을 선택한 다음 오른쪽 패널에서 액세스 로그 탭을 선택합니다.

1. 아래 그림과 같이 이 탭에는 선택한 액세스 로그의 최신 항목이 포함된 테이블이 표시됩니다. 액세스 메시지에 대한 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.

그림 13-1 로그 내용 보기



2. 현재 디스플레이를 갱신하려면 갱신을 누릅니다. 디스플레이가 10초마다 자동으로 갱신되게 하려면 계속 확인란을 선택합니다.

3. 다른 액세스 로그 파일을 보려면 로그 선택 드롭다운 메뉴에서 해당 파일을 선택합니다.
4. 다른 번호의 메시지를 표시하려면 표시할 행 텍스트 상자에 보려는 번호를 입력한 다음 갱신을 누릅니다.
5. 로그 메시지를 필터링하려면 다음을 포함하는 행만 표시 텍스트 상자에 문자열을 입력한 다음 갱신을 누릅니다. 또한 콘솔 로그 표시 안 함 확인란을 선택하여 서버에 대한 콘솔 연결에서 작성된 모든 메시지를 필터링할 수도 있습니다.
6. 로그 항목 테이블의 열을 수정하려면 보기 옵션을 누릅니다. 보기 옵션을 대화 상자의 컨트롤을 사용하여 열 순서 변경, 열 추가 또는 제거, 테이블 정렬 기준으로 사용할 열 선택 등의 작업을 수행할 수 있습니다.

액세스 로그 구성

다양한 설정을 구성하여 디렉토리의 액세스 로그 저장 위치, 작성 및 삭제 정책 등 액세스 로그를 사용자 정의할 수 있습니다.

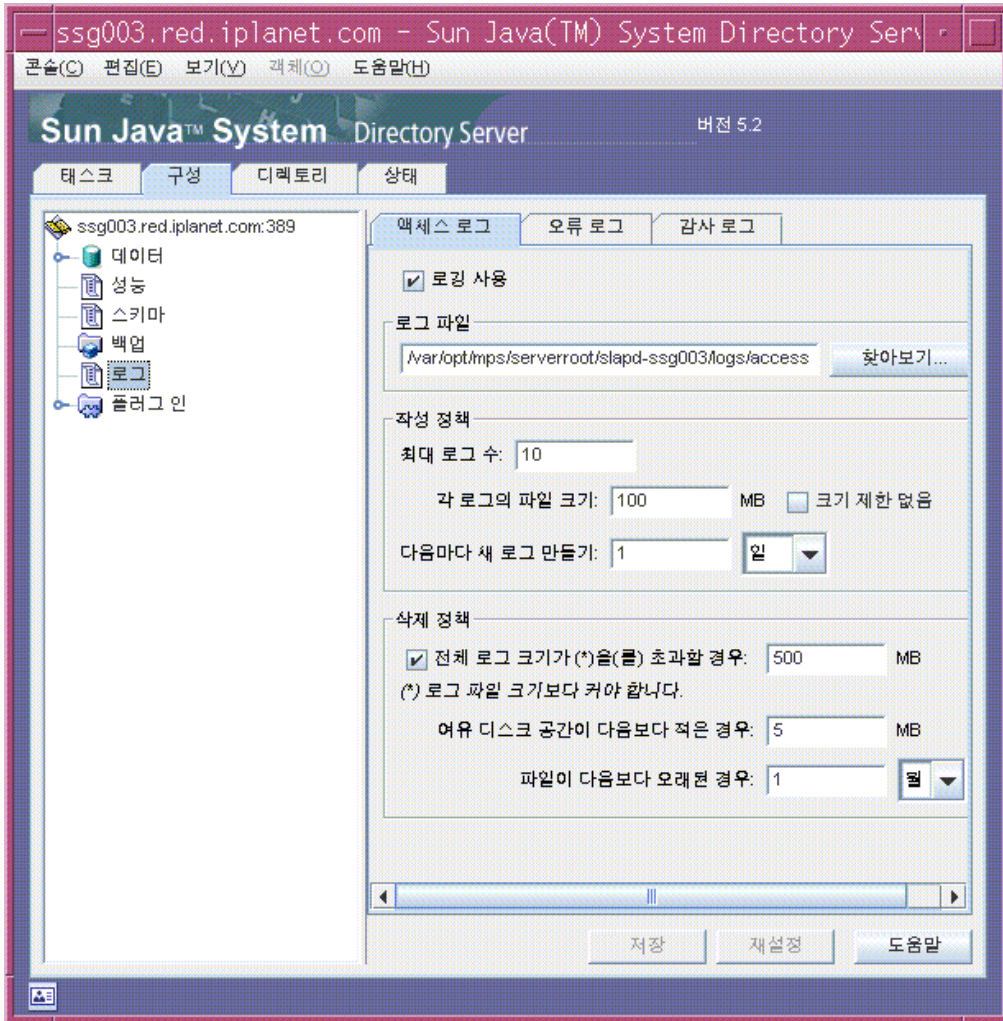
디렉토리에 대한 액세스 로깅을 비활성화할 수도 있습니다. 액세스 로그는 매우 빠른 속도로 증가할 수 있으므로(디렉토리에 대한 2,000회의 액세스마다 액세스 로그가 약 1MB 증가) 비활성화해야 하는 경우가 있습니다. 하지만 액세스 로깅을 비활성화하기 전에 액세스 로그가 유용한 문제 해결 정보를 제공한다는 것을 충분히 고려하십시오.

액세스 로그를 구성하려면 다음을 수행합니다.

1. **Directory Server** 콘솔의 최상위 구성 탭에서 로그 아이콘을 선택한 다음 오른쪽 패널에서 액세스 로그 탭을 선택합니다.

[그림 13-2](#)와 같이 이 탭에는 액세스 로그 구성 설정이 포함되어 있습니다.

그림 13-2 로그 파일 순환 및 삭제 구성 패널



2. 액세스 로깅을 사용하려면 로깅 사용 확인란을 선택합니다.
디렉토리에서 액세스 로그를 유지 관리하지 않도록 설정하려면 이 확인란을 선택 취소합니다.
액세스 로깅은 기본적으로 사용됩니다.
3. 로그 파일 필드에 액세스 로그에 사용할 파일 이름과 전체 경로를 입력합니다. 기본 파일은 다음과 같습니다.
`ServerRoot/slapd-serverID/logs/access`

4. 최대 로그 수, 로그 크기 및 아카이브 빈도를 설정합니다.
이러한 매개 변수에 대한 자세한 내용은 480페이지의 "로그 파일 순환 정책 정의"를 참조하십시오.
5. 아카이브된 전체 로그의 최대 크기, 사용 가능한 최소 디스크 공간 및 로그 파일의 최대 수명을 설정합니다.
이러한 매개 변수에 대한 자세한 내용은 480페이지의 "로그 파일 삭제 정책 정의"를 참조하십시오.
6. 설정 변경이 끝나면 저장을 누릅니다.

오류 로그

오류 로그에는 정상적인 작동 중에 디렉토리에 발생하는 오류 및 이벤트에 대한 자세한 메시지가 포함되어 있습니다.

오류 로그 보기

1. Directory Server 콘솔의 최상위 상태 탭에서 로그 아이콘을 선택한 다음 오른쪽 패널에서 오류 로그 탭을 선택합니다.

482페이지의 그림 13-1과 같이 이 탭에는 선택한 오류 로그의 최신 항목이 포함된 테이블이 표시됩니다. 오류 메시지에 대한 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.
2. 현재 디스플레이를 갱신하려면 갱신을 누릅니다. 디스플레이가 10초마다 자동으로 갱신되게 하려면 계속 확인란을 선택합니다.
3. 아카이브된 오류 로그를 보려면 로그 선택 풀다운 메뉴에서 해당 로그를 선택합니다.
4. 다른 번호의 메시지를 지정하려면 "표시할 행" 텍스트 상자에 보려는 번호를 입력한 다음 갱신을 누릅니다.
5. 로그 메시지를 필터링하려면 "다음에 포함하는 행만 표시" 텍스트 상자에 문자열을 입력한 다음 갱신을 누릅니다. 또한 콘솔 로그 표시 안 함 확인란을 선택하여 서버에 대한 콘솔 연결에서 작성된 모든 오류 메시지를 필터링할 수도 있습니다.
6. 로그 항목 테이블의 열을 수정하려면 보기 옵션을 누릅니다. 보기 옵션 대화 상자의 컨트롤을 사용하여 열 순서 변경, 열 추가 또는 제거, 테이블 정렬 기준으로 사용할 열 선택 등의 작업을 수행할 수 있습니다.

오류 로그 구성

디렉토리의 로그 저장 위치, 로그에 포함할 내용 등 오류 로그에 대한 여러 가지 설정을 변경할 수 있습니다.

오류 로그를 구성하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 로그 아이콘을 선택한 다음 오른쪽 패널에서 오류 로그 탭을 선택합니다.

484페이지의 그림 13-2와 같이 이 탭에는 오류 로그 구성 설정이 포함되어 있습니다.

2. 오류 로깅을 사용하려면 로깅 사용 확인란을 선택합니다.

디렉토리에서 오류 로그를 유지 관리하지 않도록 설정하려면 이 확인란을 선택 취소합니다. 오류 로깅은 기본적으로 사용됩니다.

3. 오류 로그의 세부 정보 수준을 설정하려면 로그 수준 버튼을 눌러 오류 로그 수준 대화 상자를 표시합니다. 자세한 오류 및 디버깅 정보를 표시할 내부 제품 구성 요소를 하나 이상 선택합니다. 선택 사항으로, 세부 정보 표시 확인란을 선택하여 사소한 메시지를 포함한 최대 런타임 출력을 반환할 수도 있습니다.

이러한 설정의 기본값을 변경하면 오류 로그가 매우 빠른 속도로 증가할 수 있으므로 충분한 디스크 공간을 보유해야 합니다. Sun Java System 고객 지원 담당자가 요청한 경우가 아니면 로깅 수준은 변경하지 않는 것이 좋습니다.

4. 로그 파일 필드에 오류 로그에 사용할 파일 이름과 전체 경로를 입력합니다. 기본 파일은 다음과 같습니다.

`ServerRoot/slapd-serverID/logs/error`

5. 최대 로그 수, 로그 크기 및 아카이브 빈도를 설정합니다.

이러한 매개 변수에 대한 자세한 내용은 480페이지의 "로그 파일 순환 정책 정의"를 참조하십시오.

6. 아카이브된 전체 로그의 최대 크기, 사용 가능한 최소 디스크 공간 및 로그 파일의 최대 수명을 설정합니다.

이러한 매개 변수에 대한 자세한 내용은 480페이지의 "로그 파일 삭제 정책 정의"를 참조하십시오.

7. 설정 변경이 끝나면 저장을 누릅니다.

감사 로그

감사 로그에는 각 접미어와 서버 구성의 변경 사항에 대한 자세한 정보가 포함되어 있습니다. 액세스로그 및 오류 로그와 달리 감사 로그는 기본적으로 사용되지 않습니다. 감사 로그를 보려면 먼저 로그를 활성화해야 합니다.

감사 로그 구성

Directory Server 콘솔을 사용하여 감사 로깅을 활성화 및 비활성화하고 감사 로그 파일의 저장 위치를 지정할 수 있습니다.

감사 로그를 구성하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 로그 아이콘을 선택한 다음 오른쪽 패널에서 감사 로그 탭을 선택합니다.

[484페이지의 그림 13-2](#)와 같이 이 탭에는 감사 로그 구성 설정이 포함되어 있습니다.

2. 감사 로깅을 사용하려면 로깅 사용 확인란을 선택합니다.

감사 로깅을 비활성화하려면 확인란을 선택 취소합니다. 기본적으로 감사 로깅은 사용되지 않습니다.

3. 로그 파일 필드에 감사 로그에 사용할 파일 이름과 전체 경로를 입력합니다. 기본 파일은 다음과 같습니다.

```
ServerRoot/slapd-serverID/logs/audit
```

4. 최대 로그 수, 로그 크기 및 아카이브 빈도를 설정합니다.

이러한 매개 변수에 대한 자세한 내용은 [480페이지의 "로그 파일 순환 정책 정의"](#)를 참조하십시오.

5. 아카이브된 전체 로그의 최대 크기, 사용 가능한 최소 디스크 공간 및 로그 파일의 최대 수명을 설정합니다.

이러한 매개 변수에 대한 자세한 내용은 [480페이지의 "로그 파일 삭제 정책 정의"](#)를 참조하십시오.

6. 설정 변경이 끝나면 저장을 누릅니다.

감사 로그 보기

1. Directory Server 콘솔의 최상위 상태 탭에서 로그 아이콘을 선택한 다음 오른쪽 패널에서 감사 로그 탭을 선택합니다.

[482페이지의 그림 13-1](#)과 같이 이 탭에는 선택한 감사 로그의 최신 항목이 포함된 테이블이 표시됩니다.

2. 현재 디스플레이를 갱신하려면 갱신을 누릅니다. 디스플레이가 10초마다 자동으로 갱신되게 하려면 계속 확인란을 선택합니다.
3. 아카이브된 감사 로그를 보려면 로그 선택 폴다운 메뉴에서 해당 로그를 선택합니다.
4. 다른 번호의 메시지를 표시하려면 "표시할 행" 텍스트 상자에 보려는 번호를 입력한 다음 갱신을 누릅니다.
5. 로그 메시지를 필터링하려면 "다음에 포함하는 행만 표시" 텍스트 상자에 문자열을 입력한 다음 갱신을 누릅니다.

서버 작업 모니터

서버는 모든 접미어에 대한 캐시 작업, 연결 및 작업 수 등 자체 작업에 대한 카운터와 통계를 항상 유지 관리합니다. 이 정보는 오류를 해결하고 서버 성능을 확인하는 데 유용합니다. Directory Server 콘솔 또는 명령줄에서 Directory Server의 현재 작업을 모니터링할 수 있습니다.

모니터할 수 있는 대부분의 매개 변수는 Directory Server 성능을 나타내며 구성 및 조정 에 의해 영향을 받을 수 있습니다. 구성 가능한 속성 및 각 속성의 조정 방법에 대한 자세한 내용은 *Directory Server Performance Tuning Guide*를 참조하십시오.

콘솔에서 서버 모니터

1. Directory Server 콘솔의 최상위 상태 탭에서 상태 트리의 루트에 있는 서버 아이콘을 선택합니다.
오른쪽 패널에 서버 작업에 대한 현재 정보가 표시됩니다. 서버를 실행하고 있지 않으면 성능 모니터 정보는 표시되지 않습니다.
2. 현재 디스플레이를 갱신하려면 갱신을 누릅니다. 표시되는 정보를 서버에서 계속 업데이트하도록 설정하려면 계속 확인란을 선택합니다.
서버 상태 패널에는 다음과 같은 정보가 표시됩니다.
 - 서버를 시작한 날짜 및 시간
 - 서버의 현재 날짜 및 시간. 복제를 활성화한 경우 각 서버의 날짜가 동일한지 정기적으로 확인해야 합니다.

- 자원 요약 테이블. 이 테이블에는 다음과 같은 자원의 시작 후 총 수와 시작 후 분 당 평균이 각각 표시됩니다.

표 13-1 자원 요약 테이블

자원	시작 후 총 수와 분 당 평균
연결	구성된 클라이언트 연결 수
시작한 작업	클라이언트에서 요청한 작업 수
완료한 작업	클라이언트에서 중단하지 않고 서버가 결과를 반환한 작업 수
클라이언트에게 전송한 항목	검색 결과로 반환된 항목 수
클라이언트에게 전송한 바이트	클라이언트 요청에 대한 응답으로 전송된 총 바이트 수

- 현재 자원 사용 테이블. 이 테이블에는 패널을 마지막으로 갱신했을 때 사용 중이었던 다음과 같은 자원이 표시됩니다.

표 13-2 현재 자원 사용

자원	최근 실시간 사용
활성 스레드	요청 처리에 사용된 스레드 수. 복제나 연결과 같은 내부 서버 기법에서 추가 스레드를 작성할 수도 있습니다.
연결 열기	현재 열려 있는 연결 수. 각 연결은 다수의 작업과 여러 개의 스레드를 차지할 수 있습니다.
사용 가능한 남은 연결	서버에서 동시에 열 수 있는 남은 연결의 총 수. 이 개수는 현재 열려 있는 연결 수와 서버에서 열 수 있는 동시 연결의 총 수에 따라 결정됩니다. 동시 연결의 총 수는 일반적으로 운영 체제에 의해 결정되며 작업에 사용할 수 있는 파일 설명자 수로 표시됩니다.
클라이언트에서 읽기 대기 중인 스레드	서버에서 클라이언트 요청을 받기 시작한 후 어떤 이유로든 해당 요청의 전송이 중지된 경우에는 스레드가 읽기 대기할 수 있습니다. 일반적으로 읽기 대기 중인 스레드는 네트워크 또는 클라이언트의 속도가 느리다는 것을 나타냅니다.
사용 중인 데이터베이스	이 서버에서 호스트하는 접미어 수. 연결 접미어는 이 개수에 포함되지 않습니다.

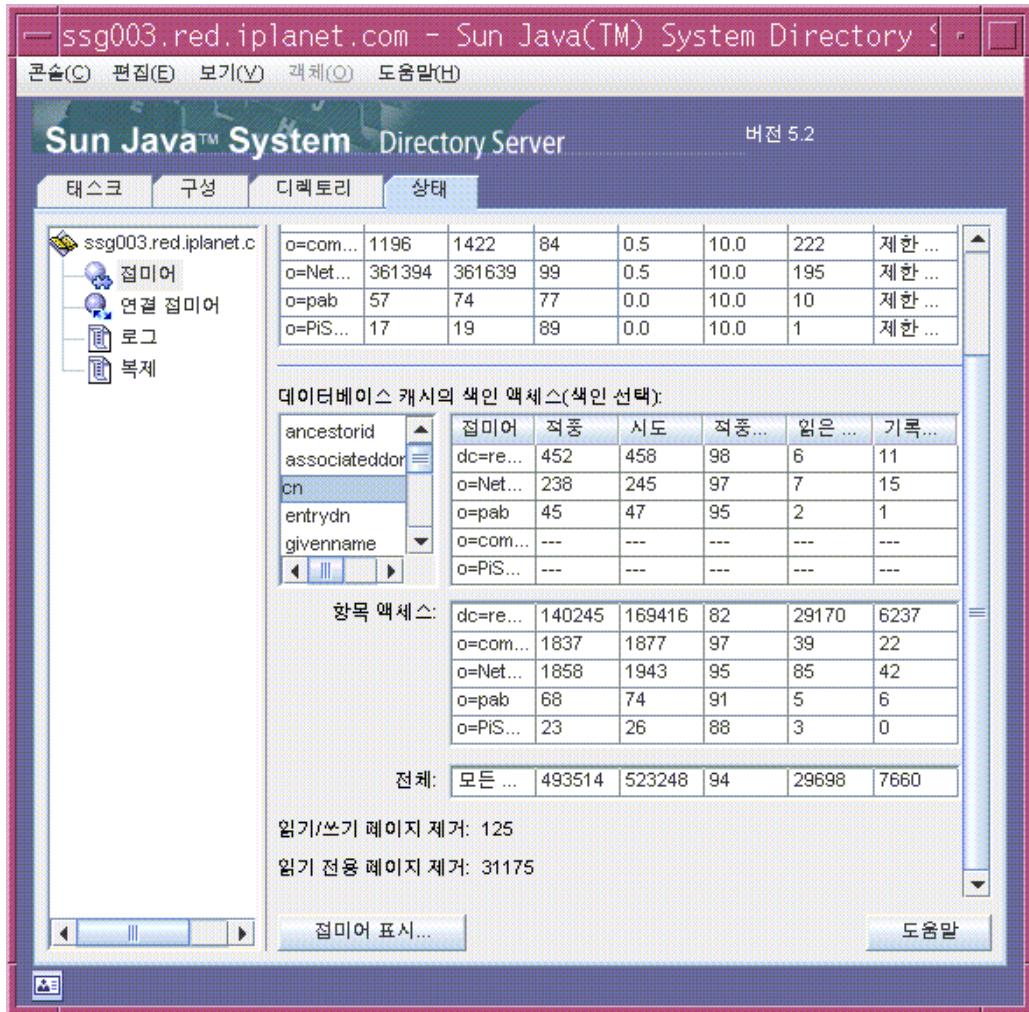
- 연결 상태 테이블. 이 테이블에는 현재 열려 있는 각 연결에 대한 다음과 같은 정보가 표시됩니다.

표 13-3 연결 상태 테이블

열 머리글	설명
열린 시간	연결이 구성된 서버 시간
시작됨	연결 중에 요청된 작업 수
완료됨	클라이언트에서 중단하지 않아 연결 중에 서버에서 완료한 작업 수
다른 이름으로 바인드됨	클라이언트에서 서버에 바인드할 때 사용한 고유 이름을 제공합니다. 클라이언트가 서버에 인증하지 않은 경우에는 <code>not bound</code> 열이 표시됩니다.
상태	<ul style="list-style-type: none"> • <code>Not Blocked</code> - 서버가 유휴 상태이거나 연결을 통해 활발하게 데이터를 송수신하고 있음을 나타냅니다. • <code>Blocked</code> - 서버가 연결을 통한 데이터 읽기 또는 쓰기 대기 중임을 나타냅니다. 이는 네트워크 또는 클라이언트의 느린 속도 때문일 수 있습니다.
유형	LDAP 연결인지 또는 DSML-over-HTTP 연결인지 표시합니다.

3. 왼쪽 상태 트리에서 접미어 노드를 누릅니다. 아래 그림과 같이 이 패널에는 각 접미어 데이터베이스 캐시의 색인 사용 및 항목 캐시에 대한 모니터 정보가 표시됩니다.

그림 13-3 접미어 모니터 패널



원할 경우 갱신 모드를 설정합니다. 패널 아래쪽에 있는 접미어 표시를 눌러 테이블에 표시할 접미어를 선택합니다.

- 첫 번째 테이블에는 각 항목 캐시에 대한 다음과 같은 정보가 표시됩니다.

표 13-4 항목 캐시 사용

열 머리글	설명
접미어	접미어의 기본 DN

표 13-4 항목 캐시 사용(계속)

열 머리글	설명
적중	디스크가 아닌 캐시로부터 읽은 항목 수
시도	캐시로부터 요청된 항목 수
적중률(%)	시도에 대한 적중률(백분율)
크기(MB)	지정된 접미어에서 읽은 항목 캐시 내용의 현재 크기
최대 크기(MB)	현재 구성의 최대 캐시 크기
크기(항목)	지정된 접미어에서 읽은 현재 캐시 항목 수
최대 크기(항목)	현재 구성의 최대 캐시 항목 수

다음 테이블에서는 각 접미어의 데이터베이스 캐시에 대한 액세스를 보여줍니다.

- 첫 번째 테이블은 구성된 색인을 통한 데이터베이스 캐시 액세스를 보여줍니다. 속성 이름 목록에서 색인 통계를 보려는 속성을 선택합니다. 선택한 속성이 색인화되어 있는 접미어의 데이터만 테이블에 표시됩니다.
- 항목 액세스 테이블은 항목 검색을 위한 데이터베이스 캐시 액세스를 보여줍니다.
- 마지막 테이블의 전체는 모든 데이터베이스 캐시에 대한 전체 액세스를 보여줍니다. 세 테이블에는 모두 다음과 같은 열이 있습니다.

표 13-5 데이터베이스 캐시 액세스

열 머리글	설명
접미어	접미어의 기본 DN
적중	색인을 통해 읽은 항목 수
시도	색인을 통해 요청된 항목 수
적중률(%)	시도에 대한 적중률(백분율)
읽은 페이지	디스크에서 접미어 캐시로 읽어온 페이지 수
기록한 페이지	캐시에서 디스크로 다시 기록한 페이지 수. 읽기 쓰기 페이지를 수정한 후에 새 페이지 공간을 확보하기 위해 캐시에서 제거하면 접미어 페이지가 디스크에 기록됩니다.

- 테이블 아래에 있는 다음과 같은 페이지 제거는 모든 데이터베이스 캐시에 누적됩니다. 캐시에서 제거된 페이지는 디스크에 기록해야 하므로 서버 성능에 영향을 줄 수 있습니다. 페이지 제거 수가 작을수록 더 바람직합니다.
 - 읽기/쓰기 페이지 제거 - 새 페이지 공간을 확보하기 위해 캐시에서 제거된 읽기/쓰기 페이지 수를 표시합니다. 수정되지 않은 제거된 읽기/쓰기 페이지라는 점에서 이 값은 기록한 페이지와 다릅니다.
 - 읽기 전용 페이지 제거 - 새 페이지 공간을 확보하기 위해 캐시에서 제거된 읽기 전용 페이지 수를 표시합니다.
- 4. 필요한 경우 왼쪽 상태 트리에서 연결 접미어 노드를 누릅니다. 이 패널에는 디렉토리에 구성된 연결 접미어에 대한 액세스 정보가 표시됩니다. 원할 경우 갱신 모드를 설정합니다.

목록에서 연결 접미어 DN을 선택하면 해당 상태를 볼 수 있습니다. 오른쪽 테이블에는 이 연결 접미어에 대해 수행된 모든 작업 횟수가 표시됩니다.

명령줄에서 서버 모니터

아래 항목에 대한 검색 작업을 수행하여 LDAP 클라이언트에서 Directory Server의 현재 작업을 모니터할 수 있습니다.

- `cn=monitor`
- `cn=monitor, cn=ldbm database, cn=plugins, cn=config`
- `cn=monitor, cn=dbName, cn=ldbm database, cn=plugins, cn=config`
- `cn=monitor, cn=dbName, cn=chaining database, cn=plugins, cn=config`

여기서 `dbName`은 모니터할 접미어의 데이터베이스 이름입니다. 기본적으로 각 연결 정보를 제외한 `cn=monitor` 항목은 익명으로 바인드된 클라이언트를 포함한 모든 사람이 읽을 수 있습니다.

아래 예에서는 일반 서버 통계를 보는 방법을 보여줍니다.

```
ldapsearch -h host -p port -D "cn=Directory Manager" -w password \
-s base -b "cn=monitor" "(objectclass=*)"
```

이 항목에서 사용 가능한 모든 모니터링 속성에 대한 자세한 내용은 *Directory Server Administration Reference*를 참조하십시오.

SNMP를 사용하여 Directory Server 모니터

SNMP(Simple Network Management Protocol)는 장치와 응용 프로그램을 실시간으로 모니터 및 관리하기 위한 표준화된 관리 프로토콜입니다. Directory Server는 SNMP 관리자 응용 프로그램에서 모니터할 수 있도록 하위 에이전트 인터페이스를 제공합니다. 이러한 인터페이스를 통해 네트워크 응용 프로그램은 디렉토리 서버 상태를 확인하고 작업 정보를 얻을 수 있습니다.

Directory Server SNMP 하위 에이전트에는 읽기 전용 값이 포함되어 있습니다. SNMP 관리자 응용 프로그램은 서버에서 작업을 수행할 수 없습니다.

일반적으로 13장, "로그 파일을 사용하여 Directory Server 모니터"에 설명된 액세스와 오류 로그가 서버에 대한 훨씬 자세한 정보를 제공하며 서버 구성에 안전하게 액세스하여 수정하려는 경우에는 주로 LDAP 프로토콜이 사용됩니다. 하지만 SNMP 하위 에이전트를 사용하면 Directory Server 인스턴스를 기존의 네트워크 관리 시스템에 통합할 수 있습니다.

이 장은 다음 내용으로 구성되어 있습니다.

- Sun Java System 서버의 SNMP
- Directory Server MIB의 개요
- SNMP 설정
- Directory Server에서 SNMP 구성
- SNMP 하위 에이전트 시작 및 중지

Sun Java System 서버의 SNMP

관리 응용 프로그램은 SNMP를 사용하여 에이전트 또는 하위 에이전트 응용 프로그램을 실행하는 응용 프로그램과 장치를 쿼리할 수 있습니다. SNMP 에이전트 또는 하위 에이전트는 SNMP 관리자의 쿼리에 응답하여 응용 프로그램 또는 장치로부터 정보를 수집합니다. 이 정보는 에이전트에 대한 MIB(Management Information Base)에 정의된 테이블의 변수로 구성됩니다.

일반적으로 네트워크 관리자는 하위 에이전트의 SNMP 변수를 쿼리하고 하위 에이전트는 요청된 값을 반환합니다. 또한 SNMP는 에이전트가 트랩 메시지를 모든 네트워크 관리자에게 보내 이벤트를 보고할 수 있도록 하는 기법을 정의합니다. Directory Server 데몬이 실행되기 전에 하위 에이전트 및 마스터 에이전트가 실행되고 있는 경우, Directory Server 하위 에이전트는 Directory Server 시작 또는 종료 시 마스터 에이전트에 SMUX 트랩을 전송합니다. 마스터 에이전트는 이것을 SNMP 트랩으로 변환합니다.

호스트 시스템에 여러 개의 하위 에이전트를 설치할 수 있습니다. 예를 들어 한 개의 호스트에 Directory Server, Application Server 및 Messaging Server가 모두 설치되어 있는 경우 각 서버의 하위 에이전트는 동일한 마스터 에이전트와 통신합니다.

UNIX 환경에서는 Administration Server가 설치되어 있는 경우 Directory Server 및 기타 Sun Java System 제품에 SNMP 마스터 에이전트(MAGT)가 제공됩니다.

SNMP를 사용하여 서버를 모니터링하는 방법에 대한 자세한 내용은 *Administration Server 관리 설명서*를 참조하십시오.

SNMP를 통해 모니터링할 서버를 설정하는 일반 절차는 다음과 같습니다.

1. Directory Server MIB을 컴파일하여 SNMP 관리 시스템에 통합합니다. 시스템 설명서를 참조하십시오.
2. [498페이지의 "SNMP 설정"](#)에 설명되어 있는 대로 컴퓨터에 SNMP를 설치하십시오.
3. Directory Server 콘솔을 통해 SNMP 하위 에이전트를 구성합니다. [506페이지의 "Directory Server에서 SNMP 구성"](#)을 참조하십시오.
4. Directory Server 콘솔을 통해 SNMP 하위 에이전트를 시작합니다. [507페이지의 "SNMP 하위 에이전트 시작 및 중지"](#)를 참조하십시오.
5. MIB에 정의되어 에이전트를 통해 제공된 SNMP 관리 대상 객체에 액세스합니다. 이 단계는 전적으로 SNMP 관리 시스템에 따라 결정됩니다.

Directory Server 구성별 단계에 대해서는 다음 절에서 설명합니다.

Directory Server MIB의 개요

Directory Server MIB에는 다음의 객체 식별자가 있습니다.

```
iso.org.dod.internet.private.enterprises.netscape.nslldap
(nslldapd OBJECT IDENTIFIER ::= { 1.3.6.1.4.1.1450.7 })
```

아래 파일에 정의되어 있습니다.

```
ServerRoot/plugins/snmp/netscape-ldap.mib
```

MIB은 SNMP를 통해 모니터할 수 있는 변수와 각 변수에 포함된 값 유형을 정의합니다. 디렉토리 MIB은 네 개의 관리 대상 객체 테이블로 구분됩니다.

- 작업 테이블 - 디렉토리의 바인드, 작업, 참조 및 오류에 대한 통계가 포함되어 있습니다. 이러한 변수 값은 `cn=snmp`, `cn=monitor` 디렉토리 항목의 속성에도 있습니다. *Directory Server Administration Reference*를 참조하십시오.
- 항목 테이블 - 디렉토리 항목 및 항목 캐시 적중 수가 포함되어 있습니다. 이러한 변수 값은 작업 변수와 함께 `cn=snmp`, `cn=monitor` 디렉토리 항목의 속성에 사용됩니다. *Directory Server Administration Reference*를 참조하십시오.
- 상호 작용 테이블 - 이 디렉토리 서버에서 마지막으로 통신한 5개 디렉토리 서버에 대한 통계가 포함되어 있습니다. *Directory Server Administration Reference*를 참조하십시오.
- 엔티티 테이블 - *Directory Server*의 인스턴스를 설명하는 변수(예: 서버 아이디 및 버전)가 포함되어 있습니다. *Directory Server Administration Reference*를 참조하십시오.

디렉토리의 MIB을 사용하려면 먼저 아래 디렉토리에 있는 MIB과 함께 컴파일해야 합니다.

```
ServerRoot/plugins/snmp/mibs
```

MIB 컴파일 방법에 대한 자세한 내용은 SNMP 제품 설명서를 참조하십시오.

SNMP 설정

Solaris 플랫폼의 경우

권장되는 Solaris 구성은 Solaris 기본 에이전트 프레임워크에 Directory Service 마스터 에이전트(MPS/MAGT)를 플러그하는 것입니다. 이 구성에서 Solaris 에이전트 `snmpdx` 는 MPS 에이전트를 시작하는 일과 해당 MPS 에이전트에서 Directory Server SNMP 요청을 전달하는 일을 담당합니다. 즉, 요청이 Directory Server와 관련되어 있는지 여부에 관계 없이 모든 SNMP 관리 클라이언트가 표준 포트 161에서 Solaris 에이전트와 연락할 수 있습니다.

MPS 에이전트가 다른 포트(1161이 권장됨)에서 실행되도록 다시 구성해야만 Solaris 에이전트와 충돌을 일으키지 않게 됩니다. 아래 제공된 절차 예에는 이러한 단계가 들어 있습니다. MPS 에이전트를 구성하는 자세한 지침은 *Administration Server 관리 설명서*에 나와 있습니다. 그러나 이 구성에서 MPS 에이전트는 Solaris 에이전트에 의해 시작되어야 하므로 MPS 에이전트를 시작 또는 중지하는 데 Administration Server를 사용하지 않는다는 점을 주의하십시오.

▶ `snmpdx` SNMP 마스터 에이전트로 플러그 - Solaris 예

이 절차를 수행하여 Solaris `snmpdx` 에이전트(`man snmpdx(1M)`)로 플러그하십시오. Solaris `snmpdx` 에이전트는 표준 포트 161에서 실행될 수 있으며 Directory SNMP 요청을 Directory Server로 전달합니다.

`slapd-instance`를 사용자의 디렉토리 인스턴스 이름으로 바꾸십시오.

1. 루트로 로그인합니다.

2. MPS 에이전트에 대한 등록 파일(/etc/snmp/conf/mps.reg)을 만듭니다.
다음 내용을 파일에 추가합니다.

```
#
# Registration file for MPS Agent.
#

#####
# agents #
#####

agents = {
    {
        name = "mps"
        subtrees = {
            # Directory Server
            1.3.6.1.4.1.1450.7,
            # Web Server
            1.3.6.1.4.1.1450.1.60
        }
        timeout = 2000000
        watch-dog-time = 86400
        port = 1161
    }
}
```

3. MPS 에이전트에 대한 리소스 파일(/etc/snmp/conf/mps.rsrc)을 만듭니다.
다음 내용을 파일을 추가합니다.

```
#
# Resource file for MPS Agent.
#

resource =
{
    {
        registration_file = "/etc/snmp/conf/mps.reg"
        policy = "spawn"
        type = "legacy"
        command = "install-dir/plugins/snmp/magt/magt
install-dir/plugins/snmp/magt/CONFIG
install-dir/plugins/snmp/magt/INIT"
    }
}
```

4. *install-dir/plugins/snmp/magt/CONFIG*의 끝에 다음 코드 행을 추가합니다.

```
TRANSPORT extraordinary SNMP
OVER UDP SOCKET
AT PORT 1161
```

5. 다시 부팅한 다음 `magt` 프로세스가 포트 1161에서 실행되고 있는지 확인합니다. 예를 들면 다음과 같습니다.

```
$ pgrep magt
```

```
21954
```

```
$ netstat -a | grep 1161
```

```
*.1161
```

```
Idle
```

```
*.1161
```

```
Idle
```

6. 506페이지의 "Directory Server에서 SNMP 구성"에 설명된 대로 Directory Server 하위 에이전트를 구성합니다. 기본 포트 199를 사용하십시오.

부트 시 자동으로 하위 에이전트를 시작하도록 다음 행을 포함하는 `/etc/init.d`에 시작 스크립트를 작성할 수 있습니다. 하위 에이전트는 MPS 에이전트가 이미 시작되었을 때만 시작됩니다.

```
$ cd install-dir/bin/slapd/server
```

```
[ -x ./64/ns-slapd ] && [ "`/bin/isainfo -b 2> /dev/null`" = 64 ] && cd ./64
```

```
./ns-ldapagt -d install-dir/slapd-instance
```

7. 하위 에이전트가 실행되고 있는지 확인합니다. 예를 들면 다음과 같습니다.

```
$ pgrep ns-ldapagt
```

```
22411
```

Linux 플랫폼의 경우

이 절의 절차는 마스터 에이전트 기본 공개 커뮤니티를 사용하여 Red Hat Linux Advanced Server에 SNMP를 설치하는 데 쓰이도록 개발되었습니다. 다른 Linux 버전을 사용하거나 다른 커뮤니티를 사용하는 경우에는 이 절차를 적절하게 수정하시기 바랍니다.

1. Red Hat Linux Advanced Server에서는 기본 snmpd가 실행되고 있지 않도록 해야 합니다. 아래 명령을 사용하십시오.

```
$ pgrep snmpd
```

기본 snmpd가 실행 중이면 다음 명령을 사용하여 중지하십시오.

```
/etc/rc.d/init.d/snmpd stop
```

2. 기본 snmpd를 실행하려는 경우에는 기본 snmpd의 포트 번호를 바꿉니다. 이는 기본적으로 Directory Server MPS(MAGT) 마스터 에이전트와 기본 snmpd가 모두 포트 161을 사용하기 때문입니다.

포트 번호를 바꾸려면 /etc/rc.d/init.d/snmpd에서 OPTIONS 변수를 다음과 같이 수정하십시오.

```
OPTIONS="-s -l /dev/null -P /var/run/snmpd.pid -a -p 1161"
adding -p 1161
```

이렇게 하면 snmpd 포트가 1161로 바뀝니다.

3. 다음을 실행하여 기본 snmpd를 시작합니다.

```
/etc/rc.d/init.d/snmpd start
```

Linux의 SNMP 관련 힌트

Linux에서 SNMP를 구성할 때는 다음 사항을 고려해야 합니다.

- 506페이지의 "Directory Server에서 SNMP 구성"에 설명되어 있는 대로 Directory Server SNMP 하위 에이전트를 구성할 때는 기본 포트 199를 사용합니다.
- Directory Server와 그 MAGT(MPS) SNMP 마스터 에이전트, SNMP 하위 에이전트를 시작, 중지, 다시 시작하는 스크립트를 사용하려면 코드 예 14-1의 코드를 사용하십시오. directorysnmp라는 파일 이름을 사용하여 /etc/init.d에 스크립트를 만드십시오.

코드 예 14-1
#!/bin/sh

SNMP Linux 스크립트 directorysnmp

```
DIR=""
```

```
BASEDIR='/bin/rpm -qa --queryformat '%{INSTPREFIXES}' sun-directory\* |
head -1'
```

```

[ -r "$BASEDIR/directory-server/5.2/shared/config/serverroot.conf" ] ||
exit 0

VAR=`/bin/cat
$BASEDIR/directory-server/5.2/shared/config/serverroot.conf`

for di in $VAR/slaped-*
do
    [ -d "$di" ] && DIR="$DIR $di"
done

start() {
    [ ! -z "$DIR" ] || exit 0
    # Start the Directory Server or Servers
    for di in $DIR; do
        $di/start-slaped
    done

    # Start the master agent
    cd $VAR/plugins/snmp/magt
    ./magt ./CONFIG ./INIT >/dev/null 2>&1 &

    # Give the master agent a chance to start.
    #
    loop_counter=1
    max_count=10
    while [ $loop_counter -le $max_count ]; do
        n=`/usr/bin/pgrep -f magt | /usr/bin/wc -l`
        if [ $n -gt 0 ]; then break; fi
        loop_counter=`expr $loop_counter + 1`
        sleep 1
    done
}

```

```

done
if [ $n -eq 0 ]; then
    echo "Directory Server agents not started."
    exit 1
fi

# Start the subagent
cd $VAR/bin/slaped/server
for di in $DIR; do
    ./ns-ldapagt -d $di >/dev/null 2>&1 &
done
}

stop() {
    [ ! -z "$DIR" ] || exit 0
    # Stop the subagent
    for di in $DIR; do
        lkfile=${di}/logs/NSLDAPAGT.LK
        if [ -r $lkfile ]; then
            pid=`/usr/bin/cut -f 1 -d' ' $lkfile`
            kill $pid >/dev/null 2>&1
        fi
    done
    # Stop the master agent
    /usr/bin/pkill magt

    # Stop the Directory Server
    for di in $DIR; do
        $di/stop-slaped
    done
}

```



```

}

# See how we were called.
case "$1" in
  start)
    start
    ;;
  stop)
    stop
    ;;
  restart)
    stop
    start
    ;;
  *)
    echo $"Usage: $0 {start|stop|restart}"
    exit 1
esac

exit 0

```

- Directory Server 및 두 개의 SNMP 에이전트가 부트 시 자동으로 시작되도록 하려면 /etc/rc3.d 및 /etc/rc5.d 모두에 다음과 같이 심볼릭 링크를 만드십시오.

```

cd /etc/rc3.d
ln -s ../init.d/directorysnmp S98directorysnmp
cd /etc/rc5.d
ln -s ../init.d/directorysnmp S98directorysnmp

```

이렇게 하면 `directorysnmp`가 실행 수준 3과 5(복수 사용자 실행 수준)에서 모두 실행됩니다. X-Windows를 사용하는 시스템(GUI 로그인)은 레벨 5에서 실행되며 레벨 3은 X-Windows를 사용하지 않는 시스템을 위한 것입니다.

- 재부트한 후에는 다음 명령을 사용하여 Directory Server 및 그 SNMP 에이전트가 실행되고 있는지 확인하십시오.

```
pgrep slapd
```

```
pgrep magt
```

```
pgrep ns-ldapag
```

- SNMP 설치를 완료하고 나면 `snmpwalk`를 사용하여 SNMP를 통한 Directory Server 관련 정보를 확인할 수 있습니다.

```
snmpwalk hostname community enterprises.1450.7.1.1.1
```

예를 들면 다음과 같습니다.

```
snmpwalk auseng026.example.com public enterprises.1450.7.1.1.1
```

Directory Server에서 SNMP 구성

사용하는 플랫폼에 SNMP 에이전트 또는 서비스를 설정한 후에는 Directory Server 인스턴스에 SNMP 매개 변수를 구성해야 합니다. Directory Server 콘솔에서 SNMP 설정을 구성하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에 있는 구성 트리의 루트에서 서버 노드를 선택한 다음 오른쪽 패널에서 SNMP 탭을 선택합니다.
2. "통계 모음 사용" 확인란을 선택합니다. 자원 사용을 향상시키기 위해 기본적으로 SNMP 변수에 대한 통계는 수집되지 않습니다. SNMP를 사용하지 않으며 LDAP를 통해 `cn=snmp, cn=monitor` 항목의 속성을 모니터링하지 않는 경우에는 이 확인란을 선택 취소된 상태로 그대로 두어야 합니다.
3. 마스터 에이전트의 호스트 이름과 포트 번호를 해당 텍스트 필드에 입력합니다.
기본값은 각각 `localhost`와 포트 `199`입니다.
4. 설명 등록정보 상자의 텍스트 필드에 정보를 입력합니다. 다음과 같은 값은 이 서버에서 제공하는 SNMP 엔티티 테이블에 반영됩니다.
 - 설명 - 디렉토리 서버에 대한 설명을 입력합니다. 이 필드는 서버 콘솔의 토폴로지 트리에 있는 이 인스턴스의 설명 필드와 유사합니다.
 - 조직 - 디렉토리 서버가 속하는 회사 또는 내부 조직 이름을 입력합니다.

- 위치 - 디렉토리 서버 호스트의 지리적 위치를 입력합니다.
 - 연락처 - 디렉토리 서버 관리자의 전자 메일 주소나 연락처 정보를 입력합니다.
5. 저장을 눌러 변경 사항을 저장합니다.
 6. 다음 절의 설명에 따라 SNMP 에이전트를 시작하거나 다시 시작합니다.

SNMP 하위 에이전트 시작 및 중지

다음 절차는 Directory Server 콘솔에서 SNMP 하위 에이전트를 시작, 재시작 또는 중지하는 방법입니다.

주 동일한 호스트에서 다른 서버 인스턴스를 추가하고 이 인스턴스가 SNMP 네트워크의 일부가 되도록 하려면 SNMP 하위 에이전트를 다시 시작해야 합니다.

SNMP 하위 에이전트를 시작, 중지 및 다시 시작하려면 다음을 수행합니다.

1. Directory Server 콘솔의 최상위 구성 탭에 있는 구성 트리의 루트에서 서버 노드를 선택한 다음 오른쪽 패널에서 SNMP 탭을 선택합니다.
2. 설명 등록정보 상자 아래에 있는 하위 에이전트 제어 버튼을 사용하여 하위 에이전트를 시작, 중지 또는 다시 시작합니다.

디렉토리를 중지해도 디렉토리 하위 에이전트는 중지되지 않습니다. 하위 에이전트를 중지하려면 이 탭에서 명시적으로 지정해야 합니다.

SNMP 하위 에이전트 시작 및 중지

속성 값 고유성 강화

UID 고유성 플러그인은 지정된 속성 값이 디렉토리 또는 하위 트리의 모든 항목에서 고유한지 확인하며, 기존의 속성 값이 포함된 항목을 추가하려는 작업이나 속성을 디렉토리에 있는 값으로 수정하려는 작업을 모두 중지합니다.

UID 고유성 플러그인은 기본적으로 비활성화되어 있습니다. 이 플러그인을 활성화하면 기본적으로 uid 속성의 고유성이 유지됩니다. 플러그인의 새 인스턴스를 작성하여 다른 속성 값에 대한 고유성을 적용할 수도 있습니다. UID 고유성 플러그인은 단일 서버의 속성 값 고유성만 제한적으로 확인합니다.

이 장은 다음 내용으로 구성되어 있습니다.

- 개요
- uid 속성에 대한 고유성 실행
- 다른 속성에 대한 고유성 실행
- 복제 시 고유성 플러그인 사용

개요

UID 고유성 플러그인은 사전 작업 플러그인으로, 서버에서 디렉토리를 업데이트하기 전에 모든 LDAP 작업을 검사하여 이 작업으로 동일한 속성 값을 가진 두 개의 항목이 작성되는지 여부를 확인합니다. 이 경우 서버는 작업을 종료하고 오류 19, LDAP_CONSTRAINT_VIOLATION을 클라이언트로 반환합니다.

디렉토리에 있는 하나 이상의 하위 트리 또는 특정 객체 클래스 항목에서 고유성을 적용하도록 플러그인을 구성할 수 있습니다. 이 구성은 속성 값에 대한 고유성을 적용할 항목 집합을 결정합니다. 이 집합의 항목을 대상으로 하고 속성 값이 이 집합의 모든 항목에서 고유하지 않을 경우에만 작업을 종료할 수 있습니다.

다른 속성에 대한 고유성을 적용하려면 UID 고유성 플러그인의 인스턴스를 여러 개 정의할 수 있습니다. 값이 고유해야 하는 각 항목 및 속성 집합에 하나의 플러그인 인스턴스를 정의합니다. 동일한 속성에 여러 개의 플러그인 인스턴스를 정의하여 각 항목 집합에 "별도의" 고유성을 적용할 수도 있습니다. 지정된 속성 값은 각 집합에 한 번만 허용됩니다.

기존 디렉토리에서 속성 고유성을 활성화해도 기존 항목에서의 고유성은 검사되지 않습니다. 고유성은 항목을 추가하거나 속성을 추가 또는 수정한 경우에만 실행됩니다.

UID 고유성 플러그인은 다중 마스터 복제 작업에 영향을 주기 때문에 기본적으로 비활성화됩니다. 복제 사용 시 UID 고유성 플러그인을 사용할 수도 있지만 [514페이지의 "복제 시 고유성 플러그인 사용"](#)에 설명된 동작에 주의해야 합니다.

uid 속성에 대한 고유성 실행

이 절에서는 uid 속성에 대한 기본 고유성 플러그인을 사용 및 구성하는 방법에 대해 설명합니다. 다른 속성에 대한 고유성을 적용하려면 [513페이지의 "다른 속성에 대한 고유성 실행"](#)을 참조하십시오.

콘솔에서 플러그인 구성

콘솔을 사용하는 경우 다른 속성에 대한 고유성을 적용하기 위해 기본 uid 고유성 플러그인을 수정해서는 안 됩니다. uid 고유성 플러그인을 사용하지 않으려면 비활성화된 상태로 두고 [513페이지의 "다른 속성에 대한 고유성 실행"](#)에 설명된 것처럼 다른 속성에 대한 새 플러그인 인스턴스를 작성합니다.

1. Directory Server 콘솔의 최상위 구성 탭에서 플러그인 노드를 확장하여 uid uniqueness 플러그인을 선택합니다.
2. 오른쪽 패널에서 확인란을 선택하여 플러그인을 활성화합니다.
초기화 기능이나 플러그인 모듈 경로 필드는 수정하지 마십시오.

3. 고유성을 적용할 하위 트리의 지정 방법에 따라 플러그 인 인수를 수정합니다.
 - 단일 하위 트리의 기본 DN을 지정하려면 인수 2의 값을 편집합니다. 두 개 이상의 하위 트리를 지정하려면 "추가"를 눌러 인수를 추가하고 새 텍스트 필드에 각 하위 트리의 기본 DN을 입력합니다.
 - 기본 항목의 객체 클래스로 하위 트리를 지정하려면 인수를 아래 값으로 설정합니다.

인수 1: attribute=uid

인수 2: markerObjectClass=*baseObjectClass*

플러그 인은 지정된 *baseObjectClass*를 가진 모든 디렉토리 항목 아래의 하위 트리에서 uid 고유성을 적용합니다. 예를 들어, ou=Employees 및 ou=Contractors와 같은 많은 분기에 사용자 항목이 있는 경우 markerObjectClass=organizationalUnit를 지정합니다.

표식 객체 클래스 아래의 분기 범위는 상당히 클 수 있으므로 해당 객체 클래스에 따라 속성 고유성 실행을 특정 항목으로 더욱 제한할 수 있습니다. 추가를 눌러 세 번째 플러그 인 인수를 추가하고 아래 값으로 설정합니다.

인수 3: requiredObjectClass=*entryObjectClass*

플러그 인은 *baseObjectClass*가 포함된 항목의 하위 트리에서 *entryObjectClass*가 있는 항목을 대상으로 하는 작업에만 고유성을 적용합니다. 예를 들어, 일반 사용자 항목이 있는 경우 requiredObjectClass=inetorgperson을 지정합니다.

4. uid 고유성 플러그 인의 편집이 끝나면 저장을 누릅니다. 변경 사항을 적용하려면 서버를 다시 시작해야 한다는 메시지가 표시됩니다.
5. 서버를 다시 시작하여 uid 속성 값에 대한 고유성을 적용합니다.

명령줄에서 플러그 인 구성

아래 절차에서는 ldapmodify 명령을 사용하여 uid 고유성 플러그 인을 활성화 및 구성하는 방법에 대해 설명합니다. 플러그 인 구성 항목의 DN은 cn=uid uniqueness, cn=plugins, cn=config입니다.

1. 아래 명령에서 nsslapd-pluginEnabled 속성을 on 또는 off로 설정하여 플러그 인을 각각 활성화하거나 비활성화합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=uid uniqueness,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on 또는 off
^D
```

2. 고유성을 적용할 하위 트리의 지정 방법에 따라 플러그 인 인수를 수정합니다.

- 단일 하위 트리의 기본 DN을 지정하려면 아래 명령을 실행하여 `nsslapd-pluginArg1` 값을 수정합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=uid uniqueness,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginArg1
nsslapd-pluginArg1: subtreeBaseDN
^D
```

두 개 이상의 하위 트리를 지정하려면 아래 명령을 실행하여 하위 트리의 전체 기본 DN이 각 인수 값으로 지정된 인수를 추가합니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=uid uniqueness,cn=plugins,cn=config
changetype: modify
add: nsslapd-pluginArg2
nsslapd-pluginArg2: subtreeBaseDN
-
add: nsslapd-pluginArg3
nsslapd-pluginArg3: subtreeBaseDN
-
...
^D
```

- 기본 항목의 객체 클래스에 따라 하위 트리를 지정하려면 인수를 아래 값으로 설정합니다. `uid` 속성에 대한 고유성은 `baseObjectClass`가 포함된 항목 아래의 하위 트리에서 실행됩니다. 이 객체 클래스가 있는 항목을 대상으로 하는 작업에만 고유성을 적용하도록 선택 사항으로 세 번째 인수에 `entryObjectClass`를 지정할 수도 있습니다.

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=uid uniqueness,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginArg0
nsslapd-pluginArg0: attribute=uid
-
replace: nsslapd-pluginArg1
```



```
nsslapd-pluginArg1: markerObjectClass=baseObjectClass
-
replace: nsslapd-pluginArg2
nsslapd-pluginArg2: requiredObjectClass=entryObjectClass
^D
```

3. 서버를 다시 시작하여 변경 사항을 적용합니다.

다른 속성에 대한 고유성 실행

UID 고유성 플러그 인을 사용하여 모든 속성에 대한 고유성을 적용할 수 있습니다. 디렉토리에서 `cn=plugins, cn=config` 아래에 새 항목을 작성하여 플러그 인의 새 인스턴스를 작성해야 합니다.

1. `ldapmodify` 명령을 사용하여 새 플러그 인 인스턴스의 구성 항목을 추가합니다. 명령의 첫 부분은 다음과 같습니다. 나머지 부분은 다음 단계에 나와 있습니다.

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=plug-in_name,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: plug-in_name
nsslapd-pluginDescription: Enforce unique attribute values
nsslapd-pluginType: preoperation
nsslapd-plugin-depends-on-type: database
nsslapd-pluginPath: ServerRoot/lib/uid-plugin.soextension
nsslapd-pluginVersion: 5.2
nsslapd-pluginVendor: Sun Microsystems, Inc.
nsslapd-pluginId: NSUniqueAttr
nsslapd-pluginInitfunc: NSUniqueAttr_Init
nsslapd-pluginEnabled: on 또는 off
...
^D
```

명령의 첫 부분에서 `plug-in_name`은 속성 이름이 포함된, 자신을 잘 나타내는 짧은 이름(예: `cn=mail uniqueness`)이어야 합니다. `ServerRoot` 및 라이브러리 `extension`은 사용하는 플랫폼에 따라 결정됩니다. 서버를 다시 시작할 때 새 인스턴스의 활성화된 상태를 `on` 또는 `off`로 지정합니다.

2. 명령의 나머지 부분은 고유성을 적용할 하위 트리의 지정 방법에 따라 플러그 인 인수를 지정합니다.

- 기본 DN에 따라 하나 이상의 하위 트리를 정의하려면 첫 번째 인수는 고유한 속성 이름이고 이후의 인수는 하위 트리에 있는 기본 항목의 전체 DN이어야 합니다.

```
nsslapd-pluginarg0: attribute_name  
nsslapd-pluginarg1: subtreeBaseDN  
nsslapd-pluginarg2: subtreeBaseDN  
...  
^D
```

- 기본 항목의 객체 클래스에 따라 하위 트리를 정의하려면 첫 번째 인수에 `attribute=attribute_name`을 포함하여 고유한 속성 이름을 지정해야 합니다. 두 번째 인수는 고유성을 적용할 하위 트리의 기본 항목을 지정하는 `baseObjectClass`여야 합니다. 이 객체 클래스가 있는 항목을 대상으로 하는 작업에만 고유성을 적용하도록 선택 사항으로 세 번째 인수에 `entryObjectClass`를 지정할 수도 있습니다.

```
nsslapd-pluginarg0: attribute=attribute_name  
nsslapd-pluginarg1: markerObjectClass=baseObjectClass  
nsslapd-pluginarg2: requiredObjectClass=entryObjectClass  
^D
```

모든 플러그인 인수에서 = 부호의 앞뒤에는 공백이 없어야 합니다.

3. 서버를 다시 시작하여 고유성 플러그인의 새 인스턴스를 서버에 로드합니다.

복제 시 고유성 플러그인 사용

UID 고유성 플러그인은 복제 작업의 일부로 업데이트를 수행하는 경우 속성 값을 검사하지 않으므로 단일 마스터 복제에는 영향이 없지만 다중 마스터 복제 시 속성 고유성을 자동으로 실행할 수 없습니다.

단일 마스터 복제 시나리오

클라이언트 응용 프로그램은 항상 마스터 복제본을 수정하므로 마스터 서버에서 UID 고유성 플러그인을 사용해야 합니다. 복제된 접미어에서 고유성을 적용하도록 플러그인을 구성해야 합니다. 마스터에서 원하는 속성 값이 고유한지 확인하기 때문에 소비자 서버에서 플러그인을 사용할 필요는 없습니다.

단일 마스터의 소비자에서 UID 고유성 플러그 인을 사용해도 복제나 정상적인 서버 작업을 방해하지는 않지만 성능이 약간 저하될 수 있습니다.

다중 마스터 복제 시나리오

UID 고유성 플러그 인은 다중 마스터 복제 시나리오에 적합하지 않습니다. 다중 마스터 복제 시에는 느슨하게 일관적인 복제 모델을 사용하기 때문에 두 서버에서 모두 플러그 인을 사용해도 같은 속성 값이 두 서버에 동시에 추가되는 것을 감지하지 못합니다.

하지만 다음과 같은 조건을 만족할 경우 UID 고유성 플러그 인을 사용할 수 있습니다.

- 이름 지정 속성에 대한 고유성 검사를 수행하는 경우
- 모든 마스터에서 동일한 하위 트리의 동일한 속성에 대해 고유성 플러그 인을 사용하는 경우

이러한 조건을 만족하면 복제 시 고유성 충돌이 이름 지정 충돌로 보고됩니다. 이름 지정 충돌은 수동으로 해결해야 합니다. 복제 충돌 해결에 대한 자세한 내용은 [388페이지의 "일반적인 복제 충돌 해결"](#)을 참조하십시오.

Directory Server 문제 해결

이 장에서는 Directory Server 설치 및 로그인에 대한 기본적인 문제 해결 정보를 제공합니다.

설치 및 로그인 문제 해결

표 16-1 솔루션 사용시 일반 설치 및 로그인 문제

문제	가능한 해결책
라이브러리가 없다는 메시지가 나타납니다.	idsktune을 실행하고 모든 권장 패치를 설치하여 모든 ERROR 조건을 수정합니다.
설치가 수행되지 않았는데 제거도 되지 않습니다. 어떻게 해야 하나요?	<p><i>다른 제품에 부정적 영향을 미치지 않는 한도에서</i> 제품 레지스트리 파일을 제거합니다.</p> <ul style="list-style-type: none"> • 슈퍼 유저로 설치할 경우 Solaris 시스템에서는 /var/sadm/install/productregistry • AIX 및 HP-UX 시스템에서는 /var/tmp/sw/productregistry • Red Hat 시스템에서는 /var/tmp/productregistry • Windows에서는 %SYSTEM_DIR%\system32\productregistry <p>그런 다음 부분적으로 설치된 파일을 직접 제거한 후 다시 설치합니다.</p>
설치가 실패한 이유를 알 수 없습니다. 설치 로그를 볼 수 있습니까?	<p>예. 로그는 다음 위치에 있습니다.</p> <ul style="list-style-type: none"> • Solaris 시스템에서는 /var/sadm/install/logs(슈퍼유저로 설치) 또는 /var/tmp(일반 사용자로 설치) • 다른 UNIX 시스템에서는 /var/tmp • Windows 시스템에서는 %TEMP% 폴더

표 16-1 솔루션 사용시 일반 설치 및 로그인 문제(계속)

문제	가능한 해결책
클라이언트가 서버를 찾지 못합니다.	<p>dirserv와 같은 호스트 이름을 사용해 보십시오.</p> <p>그래도 찾지 못하는 경우 서버가 DNS 등과 같은 사용 중인 이름 서비스에 나열되어 있는지 확인하고, dirserv.example.com과 같은 정규화된 도메인 이름을 사용해 보십시오.</p> <p>그래도 문제가 지속되면 192.168.0.30과 같은 호스트의 IP 주소를 사용해 보십시오.</p>
포트가 사용 중입니다.	<p>업그레이드하는 경우, 업그레이드하기 전에 Directory Server를 종료하지 않았을 수 있습니다. 기존 서버를 종료한 다음 업그레이드된 서버를 수동으로 시작하십시오.</p> <p>그렇지 않은 경우에는 다른 서버가 포트를 사용하고 있을 수 있습니다. UNIX 시스템의 경우 netstat(1M) 유틸리티에 -a 옵션을 사용하는 등의 적절한 도구를 통해 현재 사용 중인 포트를 확인하여, 사용 가능한 포트를 결정하십시오.</p>
LDAP 인증 오류 때문에 설치가 실패합니다.	<p>dirserv.example.com 대신 dirserv.nisDomain.Example.COM을 입력하는 등과 같이, 설치 도중 잘못된 정규화된 도메인 이름을 입력했을 수 있습니다.</p>
Directory Manager DN 및 비밀번호를 잊어버렸습니다.	<p>Directory Manager DN은 <code>ServerRoot/slapd-serverID/config/dse.ldif</code>에 <code>nsslapd-rootdn</code>의 값으로 기록됩니다.</p> <p>Directory Manager 비밀번호는 <code>dse.ldif</code>에 <code>nsslapd-rootpw</code>의 값으로 기록됩니다. 비밀번호가 암호화되지 않은 경우(암호화할 것을 권장합니다), 비밀번호는 {SSHA} 등과 같은 암호화 방법 식별자가 앞에 추가되지 않고 <code>dse.ldif</code>에 일반 텍스트로 표시됩니다.</p> <p>비밀번호가 암호화된 경우 문제를 수동으로 해결해야 합니다.</p> <ol style="list-style-type: none"> 1. Directory Server를 중지합니다. 2. <code>dse.ldif</code>에서 <code>nsslapd-rootpw</code>의 값을 변경합니다. 이 경우 뒤에 공백이 추가되지 않도록 주의하십시오. 3. <code>dse.ldif</code>를 저장하고 닫습니다. 4. 서버를 다시 시작합니다. 5. <code>nsslapd-rootpw</code>에 할당된 값을 사용하여 Directory Manager로 로그인합니다. 6. 319페이지의 "개별 비밀번호 정책 관리"에 설명되어 있는 대로 디렉토리 관리자 비밀번호의 암호화 체계를 설정한 다음, 비밀번호를 다시 변경합니다.

표 16-1 솔루션 사용시 일반 설치 및 로그인 문제(계속)

문제	가능한 해결책
관리자 비밀번호를 잊어버렸습니다.	<p>서버 설치 시 사용한 관리자 비밀번호가 생각나지 않을 때는 디렉토리 관리 권한이 있는 사용자 아이디(기본값: cn=Directory Manager)와 디렉토리 관리자 비밀번호를 사용하여 서버 콘솔을 통해 로그인할 수 있습니다. 디렉토리 관리 권한이 있는 사용자 아이디나 비밀번호가 생각나지 않을 때는 이전 항목을 참조하십시오.</p> <p>이미 디렉토리 관리자로 로그인할 수 있는 경우 다음을 수행하십시오.</p> <ol style="list-style-type: none"> 1. 서버 콘솔을 시작합니다. 2. 디렉토리 관리자 로 로그인합니다. 3. 사용자의 구성 정보가 들어 있는 디렉토리 서버에 대해 Directory Server 콘솔을 엽니다. 디렉토리 서버 인스턴스가 하나만 설치되어 있을 때는 구성 정보가 그 인스턴스에 있습니다. 4. 디렉토리 탭을 누르고 o=NetscapeRoot > TopologyManagement 아래의 관리자 노드로 갑니다. 5. 비밀번호를 바꾸려는 관리 권한이 있는 사용자를 두 번 누른 다음 나타나는 팝업 창에서 비밀번호를 변경(하고 확인)합니다. <p>이제, 관리 권한이 있는 사용자 아이디와 방금 바꾼 비밀번호를 사용하여 서버 콘솔로 다시 로그인할 수 있을 것입니다.</p>
실수로 Directory Server의 32비트 버전을 설치했습니다. 64비트 버전을 실행하려면 어떻게 해야 하나요?	<ol style="list-style-type: none"> 1. LDIF로 모든 접미어를 내보냅니다. 2. 모든 데이터베이스 파일을 제거합니다. 데이터베이스 파일은 cn=config,cn=ldbm database,cn=plugins,cn=config에서 nsslapd-directory의 값으로 지정된 경로 아래에 있습니다. 3. 아직 설치하지 않았으면 64비트 구성 요소를 설치합니다. 4. <code>ServerRoot/bin/slapd/server/64/ns-slapd</code>를 실행 가능하도록 만듭니다. 5. 운영 체제가 32비트 모드로 실행되고 있으면 64비트 모드로 재부트합니다. 6. 필요한 경우 캐시 크기 설정을 32비트 모드에서 작동하도록 변경합니다. 7. 내보낸 LDIF가 있는 모든 접미어를 초기화합니다. 8. 서버를 다시 시작합니다.
실수로 Directory Server의 64비트 버전을 설치했습니다. 32비트 버전을 실행하려면 어떻게 해야 하나요?	<ol style="list-style-type: none"> 1. LDIF로 모든 접미어를 내보냅니다. 2. 모든 데이터베이스 파일을 제거합니다. 데이터베이스 파일은 cn=config,cn=ldbm database,cn=plugins,cn=config에서 nsslapd-directory의 값으로 지정된 경로 아래에 있습니다. 3. <code>ServerRoot/bin/slapd/server/64/ns-slapd</code>가 실행 가능하지 않도록 모드를 변경합니다. 4. 내보낸 LDIF가 있는 모든 접미어를 초기화합니다. 5. 서버를 다시 시작합니다.

표 16-1 솔루션 사용시 일반 설치 및 로그인 문제(계속)

문제	가능한 해결책
설치를 처리할 스크립트를 작성했습니다. 이 스크립트를 사용하여 설치를 시도할 때 설치 프로그램이 0이 아니라 73을 반환했습니다. 무슨 의미입니까?	설치 프로그램 반환 코드는 다음과 같습니다. 0 - SUCCESS 1 - WARNING_REBOOT_REQUIRED 2 - WARNING_PLATFORM_SUPPORT_LIMITED 3 - WARNING_RESOURCE_NOT_FOUND 4 - WARNING_CANNOT_WRITE_LOG 5 - WARNING_LOCALE_NOT_SUPPORTED 50 - ERROR_FATAL 51 - ERROR_ACCESS 52 - ERROR_PLATFORM_NOT_SUPPORTED 53 - ERROR_NO_WINDOWING_SYSTEM_AVAILABLE 54 - ERROR_RESOURCE_NOT_FOUND 55 - ERROR_TASK_FAILURE 56 - ERROR_USER_EXIT 57 - ERROR_CANNOT_UPGRADE 58 - ERROR_NOTHING_TO_DO 59 - ERROR_IN_SERIALIZATION 60 - ERROR_ABNORMAL_EXIT 61 - ERROR_INCOMPATIBLE_STATEFILE 62 - ERROR_UNKNOWN_COMMANDLINE_OPTION 70 - ERROR_NOT_INSTALLED 71 - PARTIALLY_UNINSTALLED 72 - FULLY_UNINSTALLED 73 - INSTALLED 74 - ERROR_FAILED 75 - ERROR_STOPPED 76 - ERROR_STOPPED_ON_ERROR 77 - PARTIALLY_INSTALLED 즉, 73은 성공적으로 설치되었음을 의미합니다.

Sun Crypto Accelerator 보드 사용

이 부록에서는 Directory Server와 함께 Sun Crypto Accelerator 보드를 사용하여 인증서 기반의 인증과 SSL(Secure Sockets Layer) 프로토콜을 사용한 연결의 성능 향상 방법에 대해 설명합니다.

시작 전 주의 사항

[표 A-1](#)에는 Sun Crypto Accelerator 보드를 사용하여 SSL 연결 성능을 향상시키기 위해 먼저 수행해야 하는 항목이 설명되어 있습니다.

표 A-1 보드 사용을 위한 필수 사항

필수 사항	주의
보드 설치	호스트에 하드웨어, 드라이버, 패치 및 관리 유틸리티를 설치하는 경우 보드용으로 제공된 제품 설명서를 참조하십시오.
Directory Server 설치	자세한 내용은 <i>Java Enterprise System 설치 설명서</i> 를 참조하십시오.
서버 인증서(PKCS#12 형식)	Directory Server에 대한 서버 인증서를 .p12 파일로 받습니다.
CA 인증서(PEM 형식)	인증 기관(CA)에 대한 CA 인증서를 PEM(Privacy Enhanced Mail) 형식의 파일로 받습니다.

SSL 프로토콜과 SSL 인증서에 대한 자세한 설명 및 서버 콘솔을 통한 관리를 지원하는 Sun Java System 서버와 함께 프로토콜을 사용하는 방법은 [11장, "인증 및 암호화 관리"](#)를 참조하십시오.

토큰 작성

Directory Server는 토큰과 비밀번호를 사용하여 가속 보드에 있는 해당 암호화 키 자료에 액세스합니다. 토큰은 `user@realm` 형식을 사용하며, 여기서 `user`는 가속 보드의 사용자로서, 암호화 키 생성 자료의 소유자를 나타내고 `realm`은 가속 보드의 영역으로, 사용자 및 해당 키 생성 자료의 논리적 분할 영역입니다. 가속 보드 `user`는 시스템의 사용자 계정과 아무런 관계를 가질 필요가 없으며 보드에만 적용됩니다. 사용자 및 영역에 대한 자세한 내용은 가속 보드 제품 설명서를 참조하십시오.

보드용으로 제공된 `secadm(1M)` 유틸리티를 사용하여 토큰 사용자와 영역을 작성할 수 있습니다. 가속 보드에서는 다수의 `slots`을 작성하여 여러 응용 프로그램에 대한 토큰을 관리할 수도 있습니다. 여기서는 성능을 위해 호스트를 Directory Server 전용으로 설정하여 기본값인 한 개의 슬롯을 사용한다고 가정합니다. 여러 소프트웨어 응용 프로그램에 대해 보드를 사용하는 방법은 가속 보드 제품 설명서를 참조하십시오.

기본 슬롯에 액세스할 토큰 사용자와 영역을 작성하려면 다음 단계를 수행합니다.

1. `secadm` 유틸리티를 시작합니다.

```
$ CryptoPath/bin/secadm
```

기본 `CryptoPath`는 `/opt/SUNWconn/crypto`입니다.

2. 토큰 영역을 작성합니다.

```
secadm> create realm=dsrealm
System Administrator Login Required
Login: super-user
Password:
Realm dsrealm created successfully.
```

3. 작성하려는 사용자가 포함될 영역을 설정합니다.

```
secadm> set realm=dsrealm
secadm{dsrealm}> su
System Administrator Login Required
Login: super-user
Password:
secadm{root@dsrealm}#
```

4. SSL이 구성된 Directory Server를 다시 시작할 때 사용되는 비밀번호를 제공하여 기본 슬롯을 사용할 `nobody` 사용자를 작성합니다.

```
secadm{root@dsrealm}# create user=nobody
Initial password: password
Confirm password: password
User nobody created successfully.
secadm{root@dsrealm}# exit
```

이제 nobody@dsrealm 토큰의 사용자와 영역이 작성되었으며 Directory Server를 다시 시작할 때 사용되는 비밀번호를 제공했습니다.

보드 바인딩 생성

가속 보드의 바인딩은 Directory Server가 보드에 바인드할 수 있도록 생성하는 외부 보안 모듈의 형식을 사용합니다. 다양한 SSL 알고리즘을 지원하는 Directory Server 인증서 데이터베이스와 외부 보안 모듈 간의 바인딩을 생성하려면 다음 단계를 수행합니다.

1. modutil을 사용하기 전에 LD_LIBRARY_PATH를 설정합니다.


```
$ set LD_LIBRARY_PATH=ServerRoot/lib ; export LD_LIBRARY_PATH
```
2. 보안 모듈 데이터베이스가 없을 경우 아래 명령을 실행하여 작성합니다.


```
$ cd ServerRoot/shared/bin
$ ./modutil -create -dbdir ../../alias -dbprefix "slapd-serverID"
```
3. 외부 보안 모듈을 보안 모듈 데이터베이스에 추가합니다.


```
$ ./modutil -add "Crypto Mod" -dbdir ../../alias -nocertdb \
-libfile CryptoPath/lib/libpkcs11.so \
-mechanisms "RSA:DSA:RC4:DES" -dbprefix "slapd-serverID"
```

기본 *CryptoPath*는 /opt/SUNWconn/crypto입니다.
4. 보안 모듈을 나열하여 모듈이 제대로 추가되었는지 확인합니다.


```
$ ./modutil -list -dbdir ../../alias -dbprefix "slapd-serverID"
```

단계 3에서 추가한 Crypto Mod 항목이 표시되어야 합니다.
5. 이 외부 보안 모듈을 RSA, DSA, RC4 및 DES의 기본값으로 설정합니다.


```
$ ./modutil -default "Crypto Mod" -dbdir ../../alias \
-mechanisms "RSA:DSA:RC4:DES" -dbprefix "slapd-serverID"
```

이렇게 하면 기본 보안 모듈이 제대로 변경됩니다.

이제 가속 보드에 대한 바인딩을 생성했으므로 인증서를 가져올 수 있습니다.

인증서 가져오기

SSL을 구성하기 전에 먼저 [521페이지의 표 A-1](#)에 설명된 대로 받은 서버 인증서와 CA 인증서를 가져와야 합니다. 인증서를 가져오려면 다음 단계를 수행합니다.

1. 서버 인증서 .p12 파일을 가져옵니다.

```
$ cd ServerRoot/shared/bin
$ ./pk12util -i ServerCert.p12 -d ../../alias -P "slapd-serverID" \
-h "nobody@dsrealm"
Enter Password or Pin for "nobody@dsrealm": password
Enter Password for PKCS12 file: password
```

2. CA 인증서를 가져옵니다.

```
$ ./certutil -A -n "Crypto CA Cert" -t CT -i CACert.txt \
-d ../../alias -P "slapd-serverID" -h "nobody@dsrealm"
```

3. 토큰과 관련된 인증서를 나열하여 인증서를 제대로 가져왔는지 확인합니다.

```
$ ./certutil -L -d ../../alias -P "slapd-serverID" \
-h "nobody@dsrealm"
```

[단계 1](#) 및 [단계 2](#)에서 추가한 인증서 항목이 표시되어야 합니다.

이제 인증서를 가져왔으므로 Directory Server에서 SSL 연결을 수신하도록 구성할 수 있습니다.

SSL 구성

작성한 토큰과 비밀번호, 외부 보안 모듈과 Directory Server 인증서 데이터베이스 간에 생성한 바인딩, 그리고 가져온 인증서를 사용하여 Directory Server가 보안 모드로 시작 되도록 구성할 수 있습니다. SSL을 구성하고 Directory Server를 보안 모드로 다시 시작하려면 다음 단계를 수행합니다.

1. SSL 관련 Directory Server 구성 항목을 변경하기 위한 수정 작업이 요약된 `ssl.ldif` 파일을 작성합니다.

코드 예 A-1

보드를 사용하여 SSL을 활성화하기 위한 수정 작업(ssl.ldif)

```

dn: cn=RSA,cn=encryption,cn=config
changetype: add
objectclass: top
objectclass: nsEncryptionModule
cn: RSA
nsSSLToken: nobody@dsrealm
nsSSLPersonalitySSL: ServerCertNickname1
nsSSLActivation: on

dn: cn=encryption,cn=config
changetype: modify
replace: nsSSL3
nsSSL3: on
-
replace: nsSSLClientAuth
nsSSLClientAuth: allowed
-
replace: nsSSL3Ciphers
nsSSL3Ciphers: -rsa_null_md5,+rsa_rc4_128_md5,+rsa_rc4_40_md5,
+rsa_rc2_40_md5,+rsa_des_sha,+rsa_fips_des_sha,+rsa_3des_sha,
+rsa_fips_3des_sha,+fortezza,+fortezza_rc4_128_sha,
+fortezza_null,+tls_rsa_export1024_with_rc4_56_sha,
+tls_rsa_export1024_with_rc4_56_sha,
+tls_rsa_export1024_with_des_cbc_sha
-
replace: nsCertfile
nsCertfile: alias/slapd-serverID-cert8.db
-
replace: nsKeyFile
nsKeyFile: alias/slapd-serverID-key3.db

dn: cn=config
changetype: modify
replace: nsslapd-secureport
nsslapd-secureport: port
-
replace: nsslapd-security
nsslapd-security: on

```

1. 이 별칭은 Directory Server 인증서에 포함되어 있습니다.

여기서 nsslapd-secureport 값인 *port*는 보안 모드로 시작된 Directory Server에서 SSL 연결을 수신하는 포트입니다.

2. 수정 작업을 적용하여 Directory Server 구성을 변경합니다.

```
$ ldapmodify -p currPort -D "cn=directory manager" -w password -f ssl.ldif
```

여기서 *currPort*는 Directory Server에서 현재 클라이언트 요청을 수신하는 포트 번호입니다.

3. Directory Server를 보안 모드로 다시 시작합니다.

```
$ ServerRoot/slapd-serverID/restart-slapd
Enter PIN for nobody@dsrealm: password
```

여기서 *password*는 nobody@dsrealm 토큰을 작성할 때 입력한 nobody 사용자 비밀번호입니다.

이제 Directory Server는 지정한 포트에서 SSL 트래픽을 수신합니다. 이 포트에서 SSL을 통해 Directory Server에 액세스하도록 Sun Java System Administration Server와 클라이언트 응용 프로그램을 구성할 수 있습니다. 자세한 내용은 11장, "인증 및 암호화 관리"를 참조하십시오.

타사 사용권 내용

이 제품에는 다음과 같은 저작권 고지 사항의 적용을 받는 소프트웨어가 포함되어 있습니다. 여기에 명시된 모든 상표 및 등록 상표는 해당 소유자의 자산입니다.

Copyright (c) 1990-2000 Sleepycat Software. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Redistributions in any form must be accompanied by information on how to obtain complete source code for the DB software and any accompanying software that uses the DB software. The source code must either be included in the distribution or be available for no more than the cost of distribution plus a nominal fee, and must be freely redistributable under reasonable conditions. For an executable file, complete source code means the source code for all modules it contains. It does not include source code for modules or files that typically accompany the major components of the operating system on which the executable file runs.

THIS SOFTWARE IS PROVIDED BY SLEEPYCAT SOFTWARE "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT, ARE DISCLAIMED. IN NO EVENT SHALL SLEEPYCAT SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 1990, 1993, 1994, 1995 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 1995, 1996 The President and Fellows of Harvard University. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY HARVARD AND ITS CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL HARVARD OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2001 Carnegie Mellon University. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name "Carnegie Mellon University" must not be used to endorse or promote products derived from this software without prior written permission. For permission or any other legal details, please contact Office of Technology Transfer Carnegie Mellon University 5000 Forbes Avenue Pittsburgh, PA 15213-3890 (412) 268-4387, fax: (412) 268-7395 tech-transfer@andrew.cmu.edu
4. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by Computing Services at Carnegie Mellon University (<http://www.cmu.edu/computing/>)."

CARNEGIE MELLON UNIVERSITY DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL CARNEGIE MELLON UNIVERSITY BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 1997, 1998 Kungliga Tekniska Högskolan (Royal Institute of Technology, Stockholm, Sweden). All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by Kungliga Tekniska Högskolan and its contributors.
4. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 1987, 1988 Student Information Processing Board of the Massachusetts Institute of Technology. Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of M.I.T. and the M.I.T. S.I.P.B. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. M.I.T. and the M.I.T. S.I.P.B. make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Copyright (c) 1996 Regents of the University of Michigan. All rights reserved. Redistribution and use in source and binary forms are permitted provided that this notice is preserved and that due credit is given to the University of Michigan at Ann Arbor. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission. This software is provided "as is" without express or implied warranty.

Copyright 1992 Network Computing Devices, Inc. Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Network Computing Devices may not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Network Computing Devices makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

NETWORK COMPUTING DEVICES DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL NETWORK COMPUTING DEVICES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 2001-2002 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
4. The names "Xerces" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

COPYRIGHT Copyright (c) 1997-2000 Messaging Direct Ltd. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY MESSAGING DIRECT LTD. "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL MESSAGING DIRECT LTD. OR ITS EMPLOYEES OR AGENTS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. END COPYRIGHT

COPYRIGHT Copyright (c) 2000 Fabian Knittel. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. END COPYRIGHT

The source code to the Standard Version of Perl can be obtained from CPAN sites, including <http://www.perl.com/>.

This product incorporates compression code by the Info-ZIP group. There are no extra charges or costs due to the use of this code, and the original compression sources are freely available from <ftp://ftp.cdrom.com/pub/infozip/> on the Internet.

용어집

이 설명서 세트에서 사용된 용어 목록에 대한 자세한 내용은 *Java Enterprise System 용어집*(<http://docs.sun.com/doc/819-1934>)을 참조하십시오.

색인

가

가상 속성

서비스 클래스(CoS)에서 생성된
역할로 생성된 210

간접 CoS, CoS 참조

감사 로그, 로그 참조

값 기반 ACI 248

객체 식별자(OID)

일치 규칙에서 126

객체 클래스

스키마 참조

콘솔에서 항목 관리 75

cosClassicDefinition 233

cosIndirectDefinition 232

cosPointerDefinition 231

cosSuperDefinition 226

cosTemplate 221

dsIdentityMapping 446

nsComplexRoleDefinition 219

nsFilteredRoleDefinition 219

nsIndex 417

nsManagedRoleDefinition 218

nsNestedRoleDefinition 220

nsRoleDefinition 217

nsSimpleRoleDefinition 218

passwordPolicy 321

referral 102

검색 112

근사 120

동일 120, 124, 129

범위 지정 114

보다 작음 128

있음 120, 123

작거나 같음 120, 129

크거나 같음 120, 129, 130

하위 문자열 120, 130

검색 권한 251

검색 유형

지원되는 127

검색 유형, 다수 120

검색 필터 115, 118

구문 119

복합 121

복합 사용 121

부울 연산자 121

속성 지정 119

여러 개 사용 121

연산자 120

연산자 사용 120

예 118, 123

일치 규칙 125

파일에 포함 117

파일을 사용하여 지정 122

해당 속성 사용 119

계단식 복제, 복제 참조

다

계정, 사용자 계정 참조

고유 속성 플러그인

구성 510

관리된 역할, 역할 참조

구문

검색 필터 119

일치 규칙 필터 125

국제적 검색 128

동일 129

보다 작음 128

보다 큼 130

예 128

일치 규칙 필터 구문 125

작거나 같음 129

크거나 같음 129

하위 문자열 130

OID 사용 126

국제화 124

일치 규칙 필터 125

항목 수정 85

권한

개요 250

권한 지정 250

목록 250

액세스 허용 또는 거부 250

우선 순위 규칙 240

ACI 구문 243

그룹 208

그룹 정의 수정 210

그룹 정의 제거 210

동적 그룹 208

디렉토리에 대한 액세스 259

액세스 제어 256

액세스 제어 예 284

작성

동적 그룹 209

정적 그룹 208

정적 그룹 208

참조 무결성 관리 108

근사 검색 120

근사 색인, 색인화 참조

근사 색인화의 Metaphone Phonetic 알고리즘 410

기본 DN, ldapsearch 116

다

다중 마스터 복제, 복제 참조

단순 인증 268

대상

개요 243

속성 246

속성 값 248

섬표가 있는 DN 245, 294

ACI 구문 242

ACI의 키워드 244

LDAP 검색 필터 사용 247

LDAP URL 사용 257

대상 지정

디렉토리 항목 245

데이터 백업 197

기본 디렉토리 위치 198

명령줄 198

콘솔 사용 198

dse.ldif 서버 구성 파일 199

데이터베이스 캐시

모니터 492

동일 검색 120

국제적 예 129

예 124

동일 색인, 색인화 참조

동적 그룹, 그룹 참조

디렉토리 관리자

구성 49

권한 49

디렉토리 서버

개요 34

검색 111

구성 49

로그인 45

모니터 488

바인드 45

바인드 DN 변경 45

- 성능 카운터 488
- 시작 및 중지 35
- 액세스 제어 237
- 콘솔에서 항목 관리 64
- 콘솔에서 항목 삭제 76
- 콘솔에서 항목 수정 70
- MIB 497
- SNMP를 사용하여 모니터 495
- 디렉토리 서버 시작 35
- SSL 활성화 36
- 디렉토리 서버 중지 35
- 디렉토리 항목
 - 명령줄에서 관리 77
- 디렉토리 항목, 항목 참조

라

- 레거시 서버
 - 복제 378
- 레트로 변경 로그
 - 지우기 385
 - ACI 386
- 레트로 변경 로그 플러그인
 - 개요 381
 - 활성화 382
- 로그 479
 - 감사 로그 487
 - 구성
 - 감사 로그 487
 - 액세스 로그 483
 - 오류 로그 486
 - 보기
 - 감사 로그 487
 - 액세스 로그 481
 - 오류 로그 485
 - 수동 파일 순환 481
 - 액세스 로그 481
 - 액세스 로그의 디스크 공간 사용 483
 - 오류 로그 485
 - 파일 순환 정책 480
- 루트 접미어, 접미어 참조

- 루트 DN, 디렉토리 관리자 참조
- 루트 DSE 115

마

- 마스터 복제본
 - 구성 344
- 마스터 에이전트
 - Unix 496
- 매크로 ACI
 - 개요 305
 - 구문 308
 - 예 305
- 명령줄 유틸리티
 - ldapmodify 81
 - ldapsearch 118
 - start-slapd 35
 - stop-slapd 35
- 모니터
 - 데이터베이스 캐시 492
 - 로그 파일 479
 - 명령줄 493
 - 복제 상태 386
 - 연결 490
 - 연결 접미어 사용 493
 - 자원 사용 489
 - 콘솔 사용 488
 - 항목 캐시 491
 - SNMP 495
- 문자 이스케이프 123
- 문제 해결 517

바

- 바인드 규칙
 - 값 일치에 따른 액세스
 - 개요 260
 - 개요 254
 - 그룹 액세스 259

- 그룹 액세스 예 284
- 사용자 액세스
 - 부모 257
 - 자체 257
- 사용자 액세스 예 279
- 역할 액세스 260
- 익명 액세스 256
 - 예 277
- 인증 방법에 따른 액세스 268
 - LDIF 예 269
- 일반 액세스 257
- 특정 시간 또는 요일의 액세스 267
- ACI 구문 243
- all 키워드 257
- anyone 키워드 256
- authmethod 키워드 268
- Boolean 269
- dayofweek 키워드 268
- dns 키워드 266
- groupdn 키워드 259
- ip 키워드 265
- LDAP URL 257
- LDIF 키워드 255
- parent 키워드 257
- roledn 키워드 260
- self 키워드 257
- timeofday 키워드 267
- userattr 키워드 260
- userdn 키워드 256
- 바인드 DN
 - 콘솔에서 변경 45
 - 현재 보기 45
- 백업 복원
 - 명령줄 203, 204
 - 복제 시 고려 사항 200
 - 콘솔 사용 203
 - dse.ldif 서버 구성 파일 204
- 변경 로그 373
- 보다 작음 검색
 - 국제적 예 128
- 보안 427
 - 클라이언트 인증 440
- 복제 331
 - 계단식 복제본 초기화 354
 - 다중 마스터 복제본 초기화 353
 - 동기화 보장 374
 - 레거시 복제 구성 378
 - 마스터 복제본 구성 344
 - 명령줄에서 소비자 초기화 358
 - 변경 로그 373
 - 복제 계약 작성 347
 - 복제 관리자 항목 선택 336
 - 복제본 아이디 344
 - 상태 모니터 386
 - 소비자 참조 339
 - 액세스 제어 311
 - 이전 버전과의 호환성 377
 - 전용 소비자 복제본 구성 338
 - 지연 제거 340
 - 참조 무결성 구성 110
 - 허브 복제본 구성 340
 - ACI 311
 - replicate_now.sh script 376
 - SSL 활성화 363
 - WAN 364
- 복합 검색 필터 121
- 부모 액세스 257
- 부울 바인드 규칙
 - 개요 269
- 부울 연산자, 검색 필터 121
- 비교 권한 251
- 비밀번호
 - 비밀번호 정책 참조
 - 사용자 비밀번호 재설정 325
- 비밀번호 정책
 - 명령줄에서 개별 정책 작성 321
 - 명령줄에서 전역 비밀번호 정책 구성 318
 - 복제 시 고려 사항 337
 - 사용자에게 할당 322
 - 콘솔에서 개별 정책 작성 319
 - 콘솔에서 전역 비밀번호 정책 구성 316
 - ACI를 사용하여 보호 325

사

사용자 계정

- 개별 자원 제한 설정 327
- 비활성화 326

사용자 계정 비활성화 326

사용자 비밀번호 재설정 325

사용자 액세스 256

- 예 279
- 자식 항목 257
- 자체 항목 257

사용자에 대한 자원 제한 327

삭제

- ACI 276

삭제 권한 251

색인화 409

- 근사 색인 410
- 기본 색인 보기 411
- 기본 색인 수정 421
- 동일 색인 410
- 명령줄에서 색인 작성 416
- 색인 파일 삭제 419
- 시스템 색인 411
- 일치 규칙 색인 410
- 있음 색인 410
- 접미어 다시 색인화 420
- 접미어를 다시 초기화하여 다시 색인화 421
- 찾아보기 색인 422
- 콘솔에 대한 찾아보기 색인 작성 423
- 콘솔에서 색인 작성 415
- 클라이언트 검색에 대한 찾아보기 색인 작성 424
- 하위 문자열 색인 410

서비스 클래스, CoS 참조

설명서 27

성능

- 콘솔 42, 410

성능 카운터

- 서버 모니터 488

소비자 복제본

- 구성 338

속성

- 검색 119

대상 지정 246

명령줄에서 이진 값 추가 84

참조 무결성 사용 108

콘솔에서 값 제거 74

콘솔에서 항목에 추가 73

하위 유형

- 서비스 클래스(CoS)에서 지원되지 않음 222

ACI 238, 239

속성 값

대상 지정 248

속성 고유성, UID 고유성 플러그인 참조

속성 유형

스키마 참조

- cosAttribute 228
- cosIndirectSpecifier 232
- cosPriority 230
- cosSpecifier 233
- cosTemplateDN 233
- ds5BeginReplicaAcceptUpdates 356
- ds5ReferralDelayAfterInit 356
- ds5ReplicaTransportCompressionLevel 368
- dsMappedDN 446
- dsMatching-pattern 447
- dsMatching-regex 447
- dsSearchBaseDN 446
- dsSearchFilter 447
- dsSearchScope 447
- nsIdleTimeout 329
- nsIndexType 418
- nsLookThroughLimit 329
- nsMatchingRule 418
- nsRole 212
- nsRoleDN 218, 220
- nsRoleFilter 219
- nsRoleScopeDN 220
- nsSizeLimit 329
- nsSystemIndex 417
- nsTimeLimit 329
- passwordCheckSyntax 318
- passwordLockout 318
- passwordLockoutDuration 318
- passwordMaxFailure 318
- passwordMinLength 318
- passwordMustChange 325
- passwordUnlock 318
- ref 102

아

- 덤프, DN 79
 - ACI 대상 245, 294
 - 스키마 393
 - 객체 클래스 정의 보기 402
 - 객체 클래스 정의 삭제 405
 - 객체 클래스 정의 수정 404
 - 객체 클래스의 선택 사항(MAY) 속성 404
 - 객체 클래스의 속성 삭제 404
 - 객체 클래스의 필수(MUST) 속성 403
 - 검사 393
 - 검색 116
 - 속성 유형 정의 보기 398
 - 속성 유형 정의 삭제 401
 - 속성 유형 정의 편집 401
- 스키마 검사 393
- 액세스 제어 246
- 쓰기 권한 250

아

- 아이디 매핑 446
- 암호 438
- 암호화 438
- 액세스 허용 250
- 액세스 거부 250
 - 우선 순위 규칙 240
- 액세스 로그, 로그 참조
- 액세스 제어
 - 값 일치 260
 - 개요 237, 238
 - 권한 250
 - 단순 인증 268
 - 대상 속성 값 지정 248
 - 대상 속성 지정 246
 - 대상 지정 243
 - 대상 항목 지정 245
 - 동적 대상 257
 - 로깅 정보 312
 - 및 복제 311
 - 및 스키마 검사 246

- 바인드 규칙 254
 - 값 일치에 따른 액세스 260
 - 사용자 및 그룹 액세스 256
 - 일반 액세스 257
 - 특정 시간 또는 요일의 액세스 267
 - 부울 바인드 규칙 269
 - 덤프가 있는 대상 DN 245, 294
 - 액세스 제어 편집기 사용 271
 - 액세스 허용 또는 거부 250
 - 이전 버전과의 호환성 313
 - 익명 액세스 256, 268, 277
 - 콘솔에서 작성 271
 - 특정 도메인 266
 - 특정 IP 주소 265
 - 필터를 사용하여 대상 지정 247
 - ACI 구문 242
 - ACI 구조
 - ACI 배치 239
 - ACI 속성 238
 - SASL 인증 268
 - SSL 인증
- 액세스 제어 명령(ACI). ACI 참조
- 액세스 제어 설정 271
 - 액세스 제어 편집기
 - 표시 272
 - 언어 태그 126
 - 국제적 검색에서 126
 - 에이전트
 - 마스터 에이전트
 - Unix 496
 - 하위 에이전트
 - 구성 506
 - 활성화 506
 - Unix에서 시작 및 중지 507
 - 여러 개의 검색 필터 121
 - 역할 210
 - 개별 비밀번호 정책의 할당에 사용된 324
 - 객체 클래스 및 속성 217
 - 관리된 역할 211
 - 구성원 비활성화 326
 - 디렉토리에 대한 액세스 260
 - 역할 기반의 서비스 클래스(CoS) 234

- 역할 정의 삭제 216
- 역할 정의 수정 216
- 역할 정의 편집 215
- 작성
 - 명령줄에서 관리된 역할 217
 - 명령줄에서 중첩된 역할 219
 - 명령줄에서 필터링된 역할 218
 - 콘솔에서 관리된 역할 213
 - 콘솔에서 중첩된 역할 215
 - 콘솔에서 필터링된 역할 213
- 중첩된 역할 211
- 필터링된
 - 예 218
- 필터링된 역할 211
- 항목의 역할 구성원 보기 215
- 항목의 역할 구성원 정의 215
- 연결
 - 개요 155
 - 계단식 연결 구성 182
 - 계단식을 위한 프록시 인증 184
 - 명령줄에서 연결 접미어 작성 162
 - 모니터 490
 - 서버 구성 요소 170
 - 서비스 클래스(CoS) 템플릿은 연결할 수 없음 223
 - 액세스 제어 평가 166
 - 연결 접미어 관리 168
 - 연결 접미어 사용 모니터 493
 - 연결 접미어 삭제 181
 - 일시적으로 연결 접미어 비활성화 173
 - 컨트롤 및 구성 요소에 대한 연결 정책 설정 171
 - 콘솔에서 연결 접미어 작성 160
 - LDAP 컨트롤 168
 - SSL 구성 167
- 연결 접미어, 연결 참조
- 연산자
 - 검색 필터 120
 - 국제적 검색 및 127
 - 접미어 128
 - Boolean 121
- 영역
 - inSASL DIGEST-MD5 455
- 오류 로그
 - 액세스 제어 정보 312
 - 오류 로그, 로그 참조
 - 와일드카드
 - 대상 245
 - 일치 규칙 필터에서 127
 - LDAP URL 258
 - 우선 순위 규칙
 - ACI 240
 - 을 444
 - 익명 액세스 268
 - 개요 256
 - 예 277
 - 인증
 - 바인드 DN 45
 - 액세스 제어 268
 - 인증 방법
 - 프록시 인증 294
 - 인증서 기반의 인증 440
 - 인증서, SSL 참조
 - 일반 액세스
 - 개요 257
 - 일치 규칙
 - 언어 태그 및 접미어 사용 127
 - 언어 태그 사용 126
 - 와일드카드 사용 127
 - OID 및 접미어 사용 127
 - OID 사용 126
 - 일치 규칙 색인, 색인화 참조
 - 일치 규칙 필터 125
 - 읽기 권한 250
 - 읽기 전용 모드
 - 접미어 188
 - 있음 검색
 - 구문 120
 - 예 123
 - 있음 색인, 색인화 참조

자

- 자원
 - 모니터 489
- 자원 제한
 - 설정
 - 명령줄 사용 329
- 자체 쓰기 권한 251
 - 예 292
- 자체 액세스 257
- 작거나 같음 검색
 - 구문 120
 - 국제적 예 129
- 접미어 421
 - 명령줄에서 작성 147
 - 명령줄에서 접미어 초기화 192, 193
 - 명령줄에서 LDIF로 내보내기 196
 - 연결, 연결 참조
 - 일시적 비활성화 150
 - 읽기 전용 모드 188
 - 전체 디렉토리 백업 197
 - 접미어 다시 색인화 420
 - 접미어 삭제 153
 - 접미어 수준의 참조 설정 151
 - 콘솔에서 개별 접미어 내보내기 196
 - 콘솔에서 루트 접미어 작성 142
 - 콘솔에서 전체 디렉토리 내보내기 195
 - 콘솔에서 접미어 초기화 191
 - 콘솔에서 하위 접미어 작성 144
 - 항목 캐시 및 데이터베이스 캐시 사용 모니터 490
 - LDIF로 데이터 내보내기 194
 - LDIF의 항목 가져오기 189
- 접미어를 다시 초기화하여 다시 색인화 421
- 정의
 - 액세스 제어 정책 271
- 정적 그룹, 그룹 참조
- 조정
 - SSL 521-526
- 조합 순서
 - 검색 필터 124
- 조합 순서, 일치 규칙을 사용한 색인화 참조
- 중첩된 역할, 역할 참조

차

- 참조
 - 기본 참조 100
 - 스마트 참조 작성 101
 - 전역 참조 100
 - 접미어 수준의 참조 설정 151
- 참조 무결성
 - 개요 108
 - 로그 파일 108
 - 복제 110, 363
 - 비활성화 109
 - 속성 108
 - 활성화 109
- 찾아보기 색인, 색인화 참조
- 추가 권한 250

카

- 커버로스, SASL 참조
- 콘솔
 - 성능 42, 410
- 크거나 같음 검색 120
 - 국제적 예 129, 130
- 클래식 CoS, CoS 참조

타

- 특수 문자 113, 123

파

- 포인터 CoS, CoS 참조
- 포트 번호
 - 디렉토리 서버 구성 49
- 프록시 권한 251
- 프록시 인증 294

- 계단식 연결 184
- ACI 예 294
- 프록시 DN 295
- 필터
 - 일치 규칙 125
- 필터링 118
- 필터링된 역할
 - 예 218
- 필터링된 역할, 역할 참조

하

- 하위 문자열 검색 120
 - 국제적 예 130
- 하위 문자열 색인, 색인화 참조
- 하위 에이전트
 - 구성 506
 - 활성화 506
 - Unix에서 시작 및 중지 507
- 하위 유형
 - 이진 속성 84
 - LDIF 업데이트 명령문의 언어 85
- 하위 접미어, 접미어 참조
- 하위 집합 117
- 하위 트리
 - 이름 바꾸기 또는 이동 96
- 항목
 - 대상 지정 245
 - 명령줄에서 관리 77
 - 명령줄에서 삭제 87
 - 명령줄에서 수정 81
 - 역할 구성원 보기 215
 - 역할 구성원 정의 215
 - 이름 바꾸기 또는 이동 96
 - 일반 편집기에서 수정 70
 - 찾기 111, 112
 - 콘솔에서 객체 클래스 관리 75
 - 콘솔에서 관리 64
 - 콘솔에서 작성 64

- 콘솔에서 항목 삭제 76
- 콘솔에서 항목 추가 73
- LDIF 파일에서의 순서 80
- LDIF를 사용한 대량 작업 77
- 항목 캐시
 - 모니터 491
- 허브 복제본
 - 구성 340
- 호환성
 - ACI 313

A

- ACI
 - 값 기반 248
 - 구문 242
 - 구조
 - 권한 243, 250
 - 대상 242
 - 대상 개요 243
 - 대상에 있는 와일드카드 245
 - 레트로 변경 로그 386
 - 매크로 ACI 사용 305
 - 바인드 규칙 243, 254
 - 복제 311
 - 비밀번호 정책 보호 325
 - 상속 263
 - 속성 239
 - 쉽표가 있는 대상 DN 245, 294
 - 연결 접미어 166
 - 와일드카드 258
 - 용례 276
 - 우선 순위 규칙 240
 - 이름 243
 - 콘솔에서 삭제 276
 - 콘솔에서 작성 274
 - 콘솔에서 편집 275
 - 평가 240
 - 프록시 권한 예 294
 - authmethod 키워드 268
 - dayofweek 키워드 268

B

- dns 키워드 266
- groupdn 키워드 259
- ip 키워드 265
- roledn 키워드 260
- targetfilters 키워드 248
- target 키워드 244
- targetattr 키워드 246
- targetfilter 키워드 247
- userattr 및 parent 263
- userattr 키워드 260

ACI 배치 239

ACI 속성

- 개요 238

ACL, ACI 참조

all 키워드 257

anyone 키워드 256

authmethod 키워드 268

B

- b 옵션 113
- bak2db 유틸리티 203
- bak2db.pl Perl 스크립트 204

C

CoS 220

- 간접 CoS 221
- 개별 비밀번호 정책의 할당에 사용된 324
- 실제 속성 값 무시 228
- 여러 값을 갖는 속성(merge-schemes) 229
- 역할 기반의 CoS 234
- 작동 가능 속성 생성 228
- 작성
 - 명령줄에서 간접 CoS 232
 - 명령줄에서 클래식 CoS 233
 - 명령줄에서 템플릿 항목 230
 - 명령줄에서 포인터 CoS 231
 - 콘솔에서 모든 유형의 CoS 224

- 콘솔에서 포인터 및 클래식 CoS 템플릿 항목 223
- 제한 222
- 클래식 CoS 221
- 템플릿 우선 순위 230
- 템플릿 항목 221
- 포인터 CoS 221
- CoS 정의 삭제 226
- CoS 정의 편집 225

cosAttribute 속성 유형 228

cosClassicDefinition 객체 클래스 233

cosIndirectDefinition 객체 클래스 232

cosIndirectSpecifier 속성 유형 232

cosPointerDefinition 객체 클래스 231

cosPriority 속성 유형 230

cosSpecifier 속성 유형 233

cosSuperDefinition 객체 클래스 226

cosTemplate 객체 클래스 221

cosTemplateDN 속성 유형 233

D

- D 옵션 113
- dayofweek 키워드 268
- db2bak 유틸리티 198
- db2index.pl Perl 스크립트 419
- db2ldif 유틸리티 196
 - 복제본 내보내기 359
- DES 암호 439
- DIGEST-MD5, SASL 참조
- DN의 쉼표 123
- dns 키워드 266
- ds5BeginReplicaAcceptUpdates 속성 유형 356
- ds5ReferralDelayAfterInitattribute 유형 356
- ds5ReplicaTransportCompressionLevel 속성 유형 368
- dse.ldif 파일
 - 백업 199

백업을 사용하여 복원 204
 dsIdentityMapping 객체 클래스 446
 dsMappedDN 속성 유형 446
 dsMatching-pattern 속성 유형 447
 dsMatching-regexp 속성 유형 447
 dsSearchBaseDN 속성 유형 446
 dsSearchFilter 속성 유형 447
 dsSearchScope 속성 유형 447

F

Fortezza 438

G

groupdn 키워드 259
 groupdnattr 키워드 260
 GSSAPI, SASL 참조

H

-h 옵션 113

I

ip 키워드 265

L

-l 옵션 114
 LDAP 검색 필터
 대상 247

예 247, 292
 쉽표가 있는 DN 123
 LDAP 컨트롤
 연결 168
 LDAP 클라이언트
 SSL을 통한 인증 448
 LDAP URL
 액세스 제어 257
 LDAP_BASEDN 116
 ldapdelete 유틸리티
 쉽표가 있는 DN 79
 항목 삭제 87
 ldapmodify
 항목의 이름을 바꾸거나 이동하기 위해 사용 96
 ldapmodify 유틸리티
 쉽표가 있는 DN 79
 항목 수정 81
 ldapsearch 유틸리티 112
 검색 필터 118
 기본 DN 116
 명령줄 구문 112
 반환되는 속성 제한 117
 쉽표가 있는 DN 123
 예 115
 옵션 113
 특수 문자 113
 파일 지정 117
 필터 118
 LDIF
 액세스 제어 키워드
 groupdnattr 260
 userattr 260
 콘솔을 사용한 대량 작업 77
 항목 순서 80
 LDAP Data Interchange Format 29
 LDIF 가져오기 188
 명령줄 190
 콘솔 사용 189
 콘솔에서 접미어 초기화 191
 ldif2db.pl을 사용하여 접미어 초기화 193
 ldif2db를 사용하여 접미어 초기화 192
 LDIF 내보내기 194

M

- 명령줄 196
- 콘솔 사용 195
- LDIF 입력에서 파일 포식의 끝 78
- LDIF 입력의 EOF 포식 78
- ldif2db 유틸리티 192
- ldif2db.pl Perl 스크립트 193
- ldif2ldap 유틸리티 190

M

MIB

- 디렉토리 서버 497
- netscape-ldap.mib 497

N

- netscape-ldap.mib 497
- nsComplexRoleDefinition 객체 클래스 219
- nsFilteredRoleDefinition 객체 클래스 219
- nsIdleTimeout 속성 유형 329
- nsIndex 객체 클래스 417
- nsIndexType 속성 유형 418
- nsLookThroughLimit 속성 유형 329
- nsManagedRoleDefinition 객체 클래스 218
- nsMatchingRule 속성 유형 418
- nsNestedRoleDefinition 객체 클래스 220
- nsRole 속성 유형 212
- nsRoleDefinition 객체 클래스 217
- nsRoleDN 속성 유형 218, 220
- nsRoleFilter 속성 유형 219
- nsRoleScopeDN 속성 유형 220
- nsSimpleRoleDefinition 객체 클래스 218
- nsSizeLimit 속성 유형 329
- nsSystemIndex 속성 유형 417
- nsTimeLimit 속성 유형 329

O

- OID 125

P

- p 옵션 114
- parent 키워드 257
- passwordCheckSyntax 속성 유형 318
- passwordLockout 속성 유형 318
- passwordLockoutDuration 속성 유형 318
- passwordMaxFailure 속성 유형 318
- passwordMinLength 속성 유형 318
- passwordMustChange 속성 유형 325
- passwordPolicy 객체 클래스 321
- passwordUnlock 속성 유형 318
- PTA(Pass-Through Authentication) 473
 - 연결 매개 변수 476
 - 페일오버 서버 지정 477
 - 플러그인 구성 475
 - SSL 사용 476

R

- RC4 암호 439
- ref 속성 유형 102
- referral 객체 클래스 102
- replicate_now.sh script 376
- roledn 키워드 260

S

- s 옵션 114
- SASL 427
 - 서버에 커버로스 구성 443
 - 서버에 DIGEST-MD5 구성 441

- 서버에 GSSAPI 구성 444
 - 아이디 매핑 기법 446
 - 커버로스 443
 - 클라이언트에 커버로스 사용 457
 - 클라이언트에 DIGEST_MD5 구성 455
 - DIGEST-MD5 영역 455
 - DIGEST-MD5에 대한 아이디 매핑 442
 - GSSAPI 443
 - GSSAPI 및 커버로스에 대한 아이디 매핑 444
 - SASL 인증 268
 - SASL(Simple Authentication and Security Layer).
 - SASL 인증 참조
 - Secure Sockets Layer, SSL 참조 36
 - self 키워드 257
 - SNMP
 - 개요 496
 - 디렉토리 서버 모니터 495
 - 마스터 에이전트
 - Unix 496
 - 에이전트 496
 - 하위 에이전트
 - 구성 506
 - 마스터 포트 구성 506
 - 마스터 호스트 구성 506
 - 활성화 506
 - Unix에서 시작 및 중지 507
 - SSL 427
 - 가속 521-526
 - 복제 363
 - 서버 인증서 429
 - 서버 인증서 설치 432
 - 암호화 암호 선택 438
 - 연결 접미어 167
 - 인증 기관 트러스트 434
 - 인증서 데이터베이스 작성 430
 - 인증서 요청 생성 431
 - 콘솔에 클라이언트 인증 허용 440
 - 클라이언트 인증 440
 - 클라이언트에 서버 인증 구성 450
 - 클라이언트에 인증서 기반의 인증 구성 452
 - 클라이언트에서 SSL을 사용하도록 구성 448
 - 클라이언트의 사용자 인증서 452
 - pin 파일을 사용하여 서버 시작 36
 - PTA(Pass-Through Authentication) 플러그 인 사용 476
 - SSL 구성 436
 - SSL 활성화 429
 - SSL 인증
 - SSL(Simple Sockets Layer). SSL 참조
 - start-slapd 스크립트 35
 - stop-slapd 스크립트 35
- ## T
- targattrfilters 키워드 248
 - target 키워드 245
 - targetattr 키워드 246
 - targetfilter 키워드 247
 - timeofday 키워드 267
 - TLS 427
 - Triple DES 암호 439
- ## U
- UID 고유성 플러그 인 509
 - Unix
 - 마스터 에이전트 496
 - userattr 키워드 260
 - 추가에 대한 제한 264
 - userdn 키워드 256
- ## V
- VLV 색인, 찾아보기 색인을 사용한 색인화 참조
 - vlvindex 유틸리티 426

w

W

-w 옵션 [114](#)

X

-x 옵션 [114](#)

Z

-z 옵션 [114](#)